Measuring Risk in Stealth Games

Jonathan Tremblay
School of Computer Science
McGill University, Montréal
Québec, Canada
jtremblay@cs.mcgill.ca

Pedro Andrade Torres
Ciência da Computação
Universidade Salvador, Salvador
Bahia, Brasil
pedro@mail.mcgill.ca

Clark Verbrugge
School of Computer Science
McGill University, Montréal
Québec, Canada
clump@cs.mcgill.ca

ABSTRACT

Level design for stealth games requires the ability to explore and understand the possible paths players may take through a given scenario and how they are impacted by different design choices. Good tool support can help by demonstrating the existence of such paths, but for rapid, interactive design, the relative difficulty of possible solutions also needs to be quantified, in a way that correlates well with human perception of risk. Here we propose and evaluate three different metrics for defining and quantifying the risk of stealthy paths. We validate and compare these measures through a small human study, showing that a simple path-distance measure correlates best with human judgement. An evaluation of a non-trivial stealth scenario demonstrates the practicality of our approach, and shows how such measures can be useful in understanding a level design.

Keywords

Artificial Intelligence, Video Games, Stealth

1. INTRODUCTION

Stealthy behaviour, sneaking past or up to enemies, is an interesting and popular mechanism in many First Person Shooter (FPS) and Role Playing Game (RPG) games, with the design approach even becoming a sub-genre on its own [20]. Developing stealth scenarios, however, is non-trivial—a designer must select locations and parameters for static and dynamic enemies, the placement of different stealth obstacles or enhancers (sound, light/dark), and also consider the impact of level geometry [14]. The success of a design with respect to ahead-of-time gameplay goals then depends on the behaviour of actual players within the scenario, something which is traditionally available only as a result of extensive beta-testing, well after the design process should be completed.

Better stealth level design can be facilitated through appropriate interactive design tools, which can use artificial intelligence to expose the existence of different possible player paths through a stealth scenario [18]. This gives designers the ability to dynamically evaluate the relative existence of a stealthy path solution. Qualitative evaluation of level difficulty or game complexity, though, depends strongly on how players perceive possible solutions—during immersive gameplay, players will view different stealthy path choices as more or less likely to result in exposure and thus more or less dangerous or difficult. Design of a stealth metric that quantifies this perception is thus an important aspect of stealth level design as such definition is non-existent in the literature. A reliable measure of stealth difficulty would enable possible stealth paths to be algorithmically analyzed during and interactive with the level design, improving the ability of designers to appropriately scale game difficulty, and avoiding the long round trip otherwise required for human evaluation.

We propose three different, intuitively appropriate metrics for measuring stealth danger. We use a human study to evaluate these metrics, showing that while our measures all have a reasonably close correspondence to human perception, the (conceptually) simple measure of path-distance correlates best with human judgement of relative risk. Our approach is integrated into a non-trivial, *Unity3D*-based design tool, illustrating technical feasibility of our metric evaluation, and allowing us to further demonstrate application of a measurement-based approach to a non-trivial stealth scenario taken from a realistic computer game.

Specific contributions of our work include:

- We propose and describe 3 different metrics for measuring player perception of risk in a stealth game context. These metrics consider intuitively appropriate factors such as distance to enemy (Dist), line of sight (LOS), and the presence of "near misses" in being seen (NM).
- Using a human study, we evaluate how well our metrics correlate with a human ranking of the relative risk of different generated stealth paths. In this we find that (path) distance to an enemy is likely a dominant factor in evaluating risk, more important than more complex line-of-sight oriented features.
- Finally, we demonstrate application of our metrics to evaluation of the risk distribution of a realistic game level from *Metal Gear Solid.*

2. BACKGROUND

The stealth genre is characterized by games that emphasize stealthy movement (avoiding detection by enemies) as a fundamental mechanism. A number of examples exist, such as *Mark of the Ninja* [8] or *Dishonored* [2], although the approach is also popular in many combat-oriented, First Person Shooter (FPS) and Role-Playing Games (RPG), giving the player an additional interesting, alternative gameplay style that can also be used to save resources (ammo, health, magic, *etc.*). In this section we will explore the definition of a stealth game. From this definition we then introduce a high-level algorithm that finds undetected paths from a to b.

2.1 Stealth Games

Stealth games imply presentation of a challenge to the player of moving from one location to another, while avoiding detection by static and mobile enemy entities. In order to mitigate the challenge different mechanics are provided to the player. Designs that allow players to hide at different points, exploiting occlusion or shadow, and abilities such as invisibility, teleportation, *etc.,* help the player. Challenge is increased by various environmental factors: snow may leave visible movement traces, metal floors or loose objects may produce noise when walked on, alarming enemies in the level, and so forth. A level is created by combining these different structures together with basic enemy positioning, movements and detection abilities. This complex relationship between components will then generate the player's experience. Heuristically, a level can be considered to be *stealth friendly* if the in-game tools that reduce detection are greater than the environmental challenges that increase it [14].

Stealth is also encountered in FPSs and RPGs as part of normal combat preparation, where players seek to scout out the environment in order to gain knowledge about enemy movements and placements. This usage constitutes less of a stealth game in itself, but does not change the main techniques involved.

2.2 Finding Undetected Paths

Our work presumes the existence of some collection of stealthy paths through a game level. Such paths may be determined by observing and recording successful human players, or through specialized path-finding algorithms that search the state space for undetected routes. For use in interactive design the second option is of course necessary.

Our approach to calculating stealthy paths is fully described in previous work [18], but may be sketched out as follows. First, we make assumptions reasonable to stealth games, that enemy movement is deterministic, and that our path-finding goal is to guarantee the player's path does not intersect an enemy's field of view (FOV). More complex scenarios where the player clears parts of an area by enticing some enemies away by being temporarily seen are thus disallowed.

Given an enemy's deterministic path and a geometry with a start and goal position, we can formalize the problem. We begin with an overall space χ consisting of the 2D level design extruded over time. Within this space we can then define χ_{free} to be the space where an agent may move freely, not colliding with obstacles or enemy FOVs: $\chi_{free} = \chi - (\chi_{obstacles} \cup \chi_{FOV})$. In order to easily search this space we map χ to a discretized environment using $\triangle p$ for the steps in precision. A slice of this space is shown in figure 1. This view shows the level at one point in time, using red cells to represent obstacles (as a screenshot of the tool output there is some minor perspective), orange the enemies' FOVs, green representing χ_{free}, and the spheres the starting and ending player locations. As this viewpoint is moved through the time dimension, enemies will move according to their deterministic motions, changing which cells are contained in their FOV. The result is a full 3D space that represents the complete level dynamics.

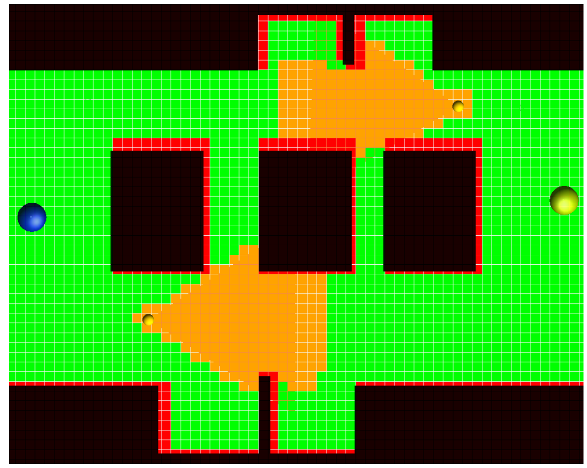


Figure 1: Level discretization: green represents walkable areas, red represents obstacles, orange the field of view of enemies, and blue and yellow spheres the start and goal location respectively.

Within this 3D space, we discover possible stealth paths through a basic path-finding algorithm, beginning at the start position and reaching the goal position, while avoiding obstacles, FOVs, and constrained by forward time-movement and feasible movement velocities. As our goal is to enable quick analysis in an interactive context, we use a heuristic search based on the Rapidly exploring Random Tree (RRT) algorithm [11]. While not optimal, this approach allows us to rapidly generate a large number of paths, with the further advantage that the randomization properties of RRT better approximates human search of the problem space. Other path-finding approaches are possible of course; the main requirement, however, is that we receive a feasible path, $q \subseteq \chi_{free}$, consisting of a sequence of points in our 3D (x, y, time) space. To inspect this path, we define a function g that takes as arguments a path q and a time t to return a 2D position in the plane: $g(q, t) = \alpha$, where α is a 2D vector. We access a specific coordinate value of the α tuple using an appropriate subscript, *e.g.* α_x or α_y.

3. RELATED WORK

Our work is intended to better understand and be able to model player behaviour in stealth games. A significant amount of previous work has been directed at analyzing and modelling player behaviours. Informal, early work was done by Bartle as a categorization of player types [4], he defined four basic kinds of players, *Achievers, Socializers, Explorers,* and *Killers*, with each type having different motives to interact with a game. A game's success can then be partly explained by how much it appeals to the different types, as well as the ecology of types a game attracts. Subsequent, more formal studies have since confirmed this rough categorization [1].

A different approach is taken by Sweetser and Wyeth [15], who introduced the *Gameflow* model to evaluate player enjoyment in games regardless of type. This model borrows from the *flow* theory in psychology originally proposed by Csikszentmihalyi, on which they added a level of abstraction specifically oriented to games. Like Bartle's work and other statistical studies, however, this does not offer a clear formal model at how a machine could, without the help of a human being, evaluate a human player's experience/trace.

A few precise models have been defined to measure player experience. In the game series *Left 4 Dead* 1 and 2 from *Valve*, an "intensity" metric is used to represent player game experience. The metric varies positively when a player is injured by an opponent, is incapacitated by an opponent, pushed or pulled off a ledge by an opponent, and when nearby opponents die, and is reduced by less stressful stretches of gameplay. The *AI Director* uses this knowledge about the player's experience to create a tailored game [6]. Measuring player experience in this fashion has since been applied in a number of different contexts, including an analysis of difficulty in World of Warcraft [3], and in more recent work by Tremblay *et al.,* where the authors used an intensity metric to dynamically adapt companion behaviour [19].

Geometric models have also been applied to game analysis. Liapis *et al.,* for example, translated some game design patterns [5] into simple algorithms that describe *symmetry, area control* and *exploration* [10]. They showed how these different metrics could be used to optimally evolve different levels. Metrics algorithms are often closely linked to generative methods for game content [7]; many generative processes follow a core structure of generating some content, *e.g.* a game level, followed by using a metric-based utility function on the content to determine the resulting quality [17]. For example, in the work of Togelius and *et al.,* they presented generative methods to evolve race tracks, measuring the result using neural network-based agents [16]. This allowed them to evaluate the quality of the track, measuring amount of progress, variation in progress, and difference between maximum and average speeds.

Perhaps the closest prior work to our presented work is by Shi and Crawfis, who presented a design tool that computes metrics on the optimal path a player may find to get through a level, given obstacles and enemy distribution [13]. They considered properties such as the minimum damage cover, longest path, and standard deviation of cover points. We are concentrating on metrics relevant to the stealth games genre, but incorporation of these kinds of FPS metrics would be interesting in more complex situations, where stealth and combat combine. Both their work and ours fall under Nelson's *state space characterization* strategy [12].

4. METRICS

The amount of danger or risk inherent in a stealthy path has a close relation to the potential for discovery by the enemy. Relative proximity of enemies is thus important, as is the direction in which enemies face—if an enemy directly looks at a player the risk of failure is also increased. We first describe a metric that relies on path distance, followed by one that focuses on the enemy's relative angle of sight, and finally a more complex metric that tries to measure how close a player came to being discovered. The different metrics presented should be looked as propositions for a formal definition of risk and will be validated in section 5.

4.1 Dist: Distance to enemy

Heuristically, a player close to an enemy risks discovery more than one far away. Since discovery is typically predicated on visual contact, however, this distance measure also needs to take into account game obstacles—an enemy behind a thin wall represents less of a danger than one equally close but not occluded. Our distance to an enemy measure (Dist) is thus defined in terms of path-distance using an A* search within χ_{free} rather than simple Euclidean or Manhattan distance. In order to compute this metric we need to consider proximity to each enemy at each point in time. We thus define $d^*(\alpha, \beta)$ to be the path-distance between two planar position, α and β. This gives us the equation,

$$\text{Dist}(p) = \sum_{t=1}^{\mathtt{T}} \left[\sum_{e \in \mathtt{E}} \frac{1}{d^*(g(p,t), g(e,t))^3} \right] \tag{1}$$

In equation 1 p represents a player's path, and e is an enemy path from the set of all enemy paths $\mathtt{E}$. T is defined as the maximum t value in our path p. In the above equation we use the reciprocal of the distance cubed as a non-linear means of scaling intensity as the player gets closer to an enemy. This is based on our need to have the function weight closer enemies as much more dangerous than distant ones, and our observations during prototyping that other exponents tend to either over-value or under-value proximity.

Variable movement speeds imply a further scaling factor must also be applied to produce meaningful values. Suppose there are two paths, initially identical, and deviating only during the last unit of distance, where despite being well away from any enemies one path ends up taking takes twice as much time as the other. Intuitively, such paths should have very similar values in terms of relative danger, but the accumulation of terms over time in equation 1 can give the slower path arbitrarily higher danger values, depending on the relative movement speeds.

To reduce the impact of these extra factors, we normalize the Dist value by the total length, L, of the path in three dimensions (2D$\times$time).

$$\widehat{\text{Dist}}(p) = \frac{\text{Dist}(p)}{L}$$

A drawback to $\widehat{\text{Dist}}$ is that it is relatively expensive to calculate. Even though A* is an efficient search algorithm for path planning, an A* search is done for every enemy in E and repeated for every time step t. Incremental approaches and cached searches would improve this, as agents typically only move small distances between time steps. Since our focus is on evaluating the metric itself rather than optimizing efficiency this remains future work.

4.2 LOS: Considering enemy view

The risk of discovery is also increased when enemies look toward a player—the more directly a player is in the line of sight of an enemy the higher the risk. Our second metric focuses on this factor, emphasizing the relative angle between the enemy's direct line-of-sight (LOS) and the player.

The first step in this computation is to know if there exists a direct LOS from the player to any enemies. If so, the angle, Angle(v_1, v_2), between the vector formed by $r = g(p,t) - g(e,t)$ and the direction the enemy is facing, $f(e,t)$ is calculated. For these we define helper functions,

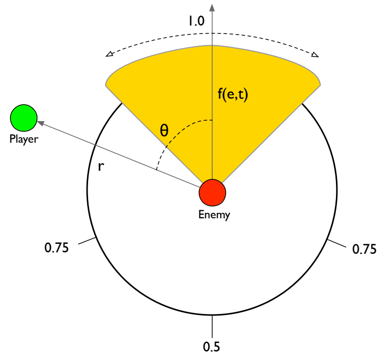


Figure 2: Threshold-based angular cost calculation. This figure illustrates how the Cost function maps angles to $[0.5, 1.0]$.

$$\mathrm{Vis}(p,e,t) = \begin{cases} 1 & \text{if } r \subseteq \chi_{free} \\ 0 & \text{otherwise} \end{cases}$$

$$\theta(g(p,t), g(e,t)) = \mathrm{Angle}(r, f(e,t))$$

This angle is then weighted according to the Cost function, considering how far outside the field of view of the enemy the player is, as illustrated in figure 2.

Considering just the angular proximity of the player to the enemy is of course not enough in itself. An enemy that is very far away, outside the range of vision, is not much of a threat, even if they look directly at the player. We also have the same concern as with Dist, that the variations in duration of the path have a significant impact on the metric value. Thus we scale the accumulated angular values by dividing by both the Euclidean distance (cubed), and the total path length, L. Note that in this case Euclidean distance (d) rather than path-distance (d^*) is usable, as we have already determined a straight line-of-sight exists.

$$\mathrm{LOS}(p) = \frac{\sum_{t=1}^{\mathrm{T}} \left[\sum_{e \in \mathrm{E}} \frac{\mathrm{Cost}(\theta(g(p,t),g(e,t)))}{d(g(p,t),g(e,t))^3} \mathrm{Vis}(p,e,t) \right]}{L} \tag{2}$$

4.3 NM: Measuring nearly misses

The last two metrics miss some important information about the player's behaviour, in that they do not account for the player's past or future behaviour. A situation that involves a near-miss in terms of discovery, barely avoiding being seen, is more risky than one where the player has ample lattitude to easily avoid detection.

Our last metric, LOS, attempts to capture the presence of such risky manoeuvres. At a given point in time, we look at the states of the last n positions of our player, as well as the next m positions in the future using a fixed time-step $\triangle\, t$. If these prior or future positions are exposed to enemy view, as shown in figure 3, the player experienced (or will experience) a near-miss in terms of detection, suggesting a risky, stressful movement.

The full, window-based risk calculation is shown in equation 3. The function $\mathrm{Seen}(\alpha, \tau)$ take as argument a planar tuple and a specific time τ. The whole equation 3 is intended to capture the idea that the closer a player passes (or will pass) to an enemy's FOV, the greater the risk. Note that unlike the previous two metrics NM is an unscaled value, not normalized to the total length of the path. The risky, near-miss behaviours we are trying to capture in this calculation imply singularly stressful events that make a path dangerous, even if the rest of it is relatively safe.

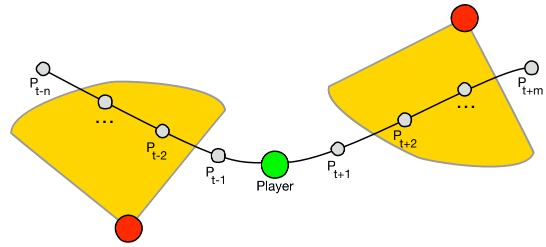


Figure 3: A representation of the NM metric time-window. Here enemies can see both near-past and near-future player positions, and so even if the path is successful due to changing enemy FOVs, the player is undertaking a *risky* movement.

$$\mathrm{Seen}(\alpha,\tau) = \begin{cases} 1 & \text{if } (\alpha_x, \alpha_y, \tau) \in \chi_{FOV} \\ 0 & \text{otherwise} \end{cases}$$

$$W^-(p,t,n) = \sum_{i=1}^{n} (n-i)^2 \cdot \mathrm{Seen}\left(g(p,t-i),t\right)$$

$$W^+(p,t,m) = \sum_{i=1}^{m} (m-i)^2 \cdot \mathrm{Seen}\left(g(p,t+i),t\right)$$

$$\mathrm{NM}(p) = \sum_{t=1}^{\mathrm{T}} \left(W^-(p,t,n) + W^+(p,t,m)\right) \tag{3}$$

An obvious final direction for these metrics would be consider a hybrid form, combining the individual metrics in some fashion. Our interests in this work, however, is to first understand the efficacy of the our metrics in isolation before addressing the significant complexity of tuning the weights of each metric's contribution in a hybrid. Efficiency is also expected to be a concern, and a hybrid form would need to demonstrate a sufficient trade-off between cost and improvement. The experimental work we now present shows that the individual metrics already have good predictive power.

5. EXPERIMENTAL RESULTS

Our experimental work is aimed at demonstrating and comparing the value of our metrics with respect to measuring stealth difficulty or danger, which will allow us to gain deeper knowledge of the definition of risk. In this section we first discuss how the metrics behaves differently in a particular scenario, and then use a human study to argue for and compare validity. Finally, we will see a simple example application of the metrics to understand an in-game stealth level.

5.1 Metric Behaviours

Our metrics have interesting qualitative differences in how they measure danger, and thus tend to identify risky behaviour at different points. Consider an example taken from level 14 of our human study from section 5.2, and shown in figure 5. This figure shows three key-frame moments in

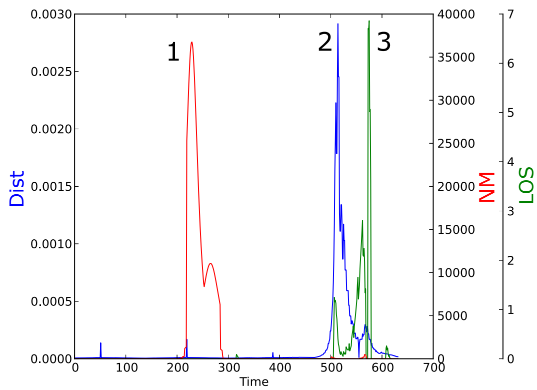


Figure 4: Evolution of the presented metrics over time (s) on level 14. The numbers refer to key-frames in Figure 5.

the level. The player is represented by a small blue sphere, and proceeds along the gray trace from the the large blue sphere (bottom right of the center) to the large green sphere (bottom left of the center) positions, while the black bars indicate walls, and the orange areas the (discretized) FOVs of 3 mobile enemy guards.

Figure 4 shows the evolution over time of each of our metric values. Note that here these values are not normalized by the total length of the path. This way we can clearly see at which key moment the metrics are activated. Each metric indicates points of maximal danger at different times in the player's progress. Although the Dist metric shows some activity in key-frames 5a and 5c, the peak of the Dist metric occurs in key-frame 5b, where proximity to an enemy is greatest, and the overall accumulated path-distances from the player to all the enemies ends up maximized. The LOS metric, however, considers the point of maximal danger to be at key-frame 5a, when the player moves directly through the line-of-sight of an enemy, if just outside the visual range. The NM metric finds risky behaviour only at key-frame 5c, when the enemy has just turned to look at the earlier path of the player.

Arguments can be made for each of these that they accurately represent the element of risk in this stealthy path. Maximal points, however, clearly differ. In the following section subsection we thus look at how these metrics compare to human rankings, in order to determine which best corresponds to human judgement.

5.2 Human Study

Our human study asked participants to evaluate different player paths (produced by the algorithm presented in section 2) and determine which was safer. The degree of correlation between participant choices and which paths our metrics determined safer would thus show whether our metrics matched human perception, and thus tell us which factors most contributed to what players felt was risky.

Participants were presented with an animated gif image on endless loop showing the movements of two players, including enemy movements and FOVs. Levels were based on different, but game-realistic geometries and enemy arrangements, attempting to provide a cross-section of level complexity and path safety. Figure 6 shows 3 examples of the 15 levels[1] presented—the small red and blue spheres represented the two players, both beginning and ending at the same points (choice of red or blue for the different paths was randomized). After viewing a level, the participant had to click on a button to select which path was the overall safest, allowing us to compare the resulting ranking with our individual metric ranks. The study was designed to take about 15–20 minutes to complete (avoiding participant burn-out). The study was conducted under unsupervised conditions (*i.e.,* on the participant's web browsers, at their leisure) and consisted of 27 anonymous participants, mainly drawn from the graduate and undergraduate population of our department.

Table 1 summarizes the raw data from our study. For each of the 15 levels, and for humans and each metric, we indicate a 1 if the participants or metric selected player 1's path as safest, 2 if they selected player 2's path as safest, and 0 if there was no consistent choice. The latter value only occurred for human players, where we imposed a 75% threshold on agreement between participants to establish a ranking. This count can be recuperated from the first line of table 1.

Table 1: Human rankings *vs.* metrics. 1 refers to blue player and 2 to the red player. 0 represents that no agreement was reached by the humans.

Level	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
# voted 1	22	1	18	27	23	12	7	7	0	0	4	22	4	22	8
Human	1	2	0	1	1	0	0	0	2	2	2	1	2	1	0
Dist	1	2	1	1	1	2	2	2	2	2	2	1	2	1	1
LOS	1	2	1	1	2	1	2	2	2	2	2	2	2	2	2
NM	0	2	2	1	1	1	1	2	2	2	2	1	2	2	1

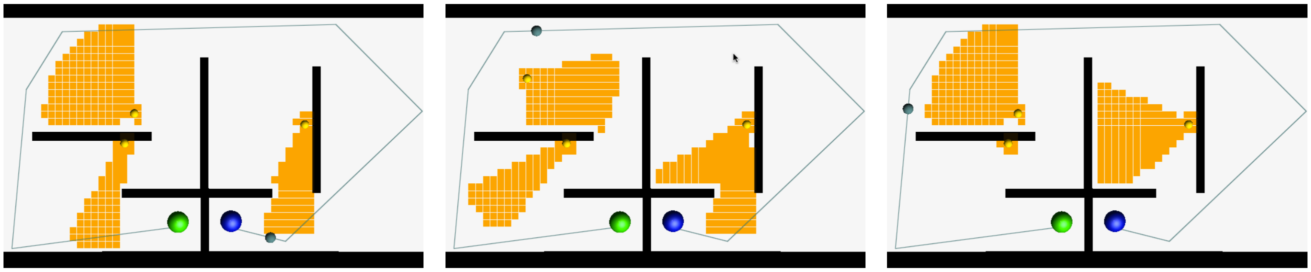
As an initial and basic observation, we observe that all metrics have quite good agreement with the humans. Out of the 10 levels for which human agreement reached our 75% threshold LOS agrees 7 times, and NM agrees 7. Surprisingly, given that it is conceptually the simplest, the Dist metric stands out as achieving perfect agreement with the humans. Even with just 10 of our 15 human judgements considered definitive this is very unlikely if due to random chance, suggesting that path-distance might be a more important factor than others.

Our expectation after examining the paths found in the levels we had designed was that levels 2–4 and 9–13 represented situations in which one path was clearly safer than the other, while levels 1, 5–8, 14, and 15 were more ambiguous. We thus now explore levels 1, 3, 5, 12, and 14 as example situations where the outcome either did not match our expectation, or where our metrics disagreed.

Level 1 - As shown in figure 6a, this level consisted of a central occlusion, with a single guard blocking one route around the obstacle and two paths going the other way. Both paths easily avoid the enemy guard, with the only significant difference being that the red path arrived at the goal at the same time as the enemy was rotating on the right.

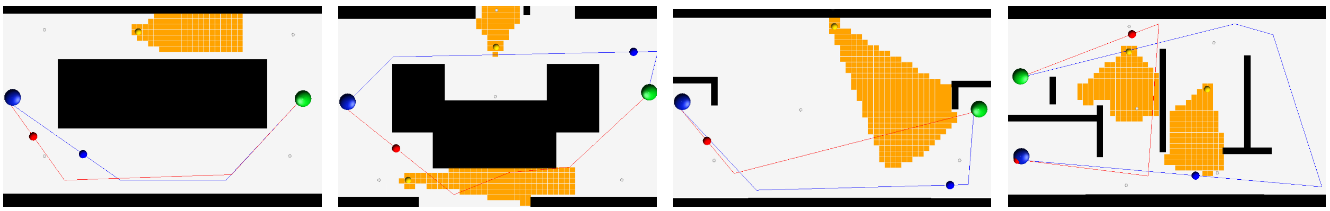
The metrics for level 1 shown in table 2 indicate that Dist and LOS ranked the blue path safer than the red path, although with a very small difference in value. We will revisit this concept for larger values, but in this case, since we are

[1]The levels and study are available at `http://goo.gl/fGg3pR`.



(a) Key 1, $t \simeq 200$s. (b) Key 2, $t \simeq 500$s. (c) Key 3, $t \simeq 560$s.

Figure 5: Different key-frames of level 14.



(a) Level 1 (b) Level 3 (c) Level 5 (d) Level 12

Figure 6: Examples of levels used in the study.

Table 2: Level 1 Metrics

	Dist	LOS	NM
blue path (1)	0.00002	0	0
red path (2)	0.00003	0.00003	0

interested in ranking the paths, this is a valid ranking. LOS measures some danger for the red path whereas no line of sight exists between the blue player and the enemy. In the case of NM, since the players' paths do not cross the enemy's path, they both were measured as zero. Since the humans clearly identify the blue player being the safest, this suggests our NM metric may be too coarse, and that a focus on unusually dangerous events is not sufficient.

Level 3 - This is a tricky level, in that there are two possible ways to get to the goal, and the players proceed through one or the other; see figure 6b. The north corridor has a fast moving enemy walking north/south, and the south corridor has a slow moving enemy walking from east to west. Here the blue player quickly raced through the north corridor, while the red player slowly sneaked behind the enemy on the south corridor and then dashed to the goal once near the end. Metric data for this level is shown in table 3.

Table 3: Level 3 Metrics

	Dist	LOS	NM
blue path (1)	0.00115	0.00362	428203
red path (2)	0.00242	0.01243	204

This represents a corner case for NM. The red player received a small value, mostly because despite closely following the enemy, the player is far enough away that little cost was attributed as she dashed out and was walking outside the cost window. The distance-based weighting the other two metrics, however, tends to give a higher value to the red player. Since humans did not achieve good agreement themselves, however, we can see this as a trade-off in risk—a brief, close call in the blue player's path is roughly equivalent to the long, slow, moderately dangerous progress by the red player.

Level 5 - This level only has a rotating camera with a long, narrow FOV; see figure 6c. The blue player walks fully outside of this field of view, whereas the red player waits for an opening in the rotation to dash directly to the goal. Metrics for this level are shown in table 4.

Table 4: Level 5 Metrics

	Dist	LOS	NM
blue path (1)	0.00002	0.01798	0
red path (2)	0.00003	0.00223	6155

The humans here again prioritized the distance to enemy, even though the blue player actually spent more time within the enemy's viewpoint. Our different metrics of course prioritize this differently: LOS gives greater weight to the angular proximity, while the path-intersection of the red player results in higher NM.

Level 12 - In this level, shown in figure 6d, the blue player goes around the long way, but at one point, walks in front of the FOV of the east enemy. The red enemy takes more risks by taking the short cut through the middle of the level.

Table 5: Level 12 Metrics

	Dist	LOS	NM
blue path (1)	0.00127	28.565	276620
red path (2)	0.00191	0.005677	682589

The results in table 5 show that the LOS metric ranks the red player safer than the blue player. This is mainly because the blue player walks in front of FOV of the east enemy, causing the metric to add a high cost to that path. Dist and

NM assess the situation better and more like the humans, attributing greater danger to the red player who both comes closer the enemy, and much closer to being seen.

Level 14 - Similar to level 12, the blue player in this level follows a roundabout route, trying to avoid contact with enemies, as shown in the key-frames of figure 5. The red player (not shown) walks through the enemies' FOV space quite quickly.

Table 6: Level 14 Metrics

	Dist	LOS	NM
blue path (1)	0.00025	2.36677	1196323
red path (2)	0.00242	0.127234	645539

As opposed to level 12, however, the results in table 6 show that both LOS and NM rank the red player as being the safest, since despite the proximity to enemies the path manages to almost never cross in front of the enemy, even within a window of past and future positions. Dist agrees more with the humans, ranking blue safer since its route takes it well away from the enemies.

Each of our metrics tries to measure different properties of what might be considered risky behaviour in a stealth context. Comparison with human perception verifies that these factors are indeed important concerns for players as well, and while confirmation of our results in a larger human study is necessary, suggests there may be a useful ranking of these factors. Human results best correspond to a pure (geometric) distance measure, and our measures based on near-misses and angular proximity match less well. This may also depend on context, however, and a similar exploration using a first-person perspective visualization (rather than overhead view) would be interesting—near-misses and being apparently within an enemy's FOV may be more important concerns if the player is more immersed in the game context, and is also less able to easily determine the extent of enemy FOV.

5.3 Metrics for level analysis

An interesting application of stealth metrics is with respect to level analysis. Measurement of the different paths through a level gives us an overall measure of level difficulty, as well as distribution of solution difficulty, both of which can then be used in tuning level design.

As an initial demonstration of the technique, we thus applied our metrics to the first level of *Metal Gear Solid* [9], a highly regarded stealth game. In this level the player has to sneak around two moving enemies within a cargo dock in an attempt to get to the surface exit on the other side. Figure 7 shows the initial (blue) and goal (green) player positions, along with the probability distribution of the player being seen by 2 patrolling enemy guards. This is a complex scenario with multiple path choices. In figure 8 we show a sample of 1500 paths found by the stealth tool, clustering them into 4 main groupings (colours), based on which corridor a path traverses (indicated by the circles in Figure 8). Note that this clustering is heuristic and based on our observations; it is also possible to have one path associated with more than one cluster.

Table 7 shows the average and median metric values of the different clusters. From this simple data we can see that the red cluster seems to have safer paths than the rest,

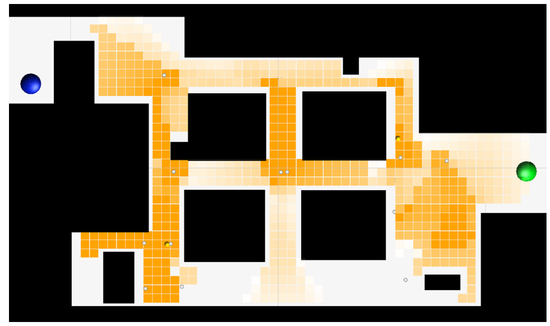


Figure 7: Metal Gear Solid's first level with a static probability distribution of getting seen by the enemy. The darker the orange, the higher the chance of getting seen.

especially if we look at the Dist metric. This was not obvious just from the simple, time-flattened probabilities shown in figure 7, but does mirror our informal observations, as paths in this cluster seemed to do quite well at avoiding enemies.

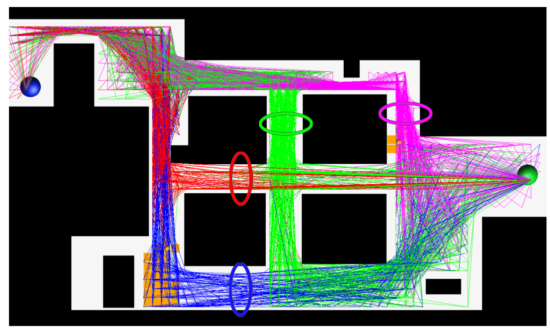


Figure 8: Clustering of feasible paths within the level.

We present both average and median since outlier paths can easily skew these simple statistics. It is, however, important to investigate these since they may represent extremal or unusual strategies. Figure 9 shows a detailed view of the distribution of the Dist metrics for the red and green clusters as the average and mean are widely different. From the distribution, one can observe that even though it is not the trend, it is possible to achieve the same level of safety in both the green and red clusters. Smaller peaks in the green cluster suggest sub-clusters of higher risk, likely due to differences in timing that result in more closely encountering enemies.

Table 7: Clustering result, showing the average and median for each metric.

Metrics	Red		Blue		Green		Magenta	
	Avg	Med	Avg	Med	Avg	Med	Avg	Med
Dist ($\times 10^{-3}$)	0.6	0.2	3.7	0.9	1.8	1.0	0.2	0.9
LOS ($\times 10^{-2}$)	0.7	0.02	13.8	0.4	1.6	0.01	4.7	0.3
NM ($\times 10^{5}$)	2.0	1.6	2.8	2.4	2.5	1.9	1.9	0.7

Informed by this kind of data, a designer has a better understanding of her level design, and can use the relative risk value, as well as the distribution of risk values over different path clusters as a guide for manipulating the level

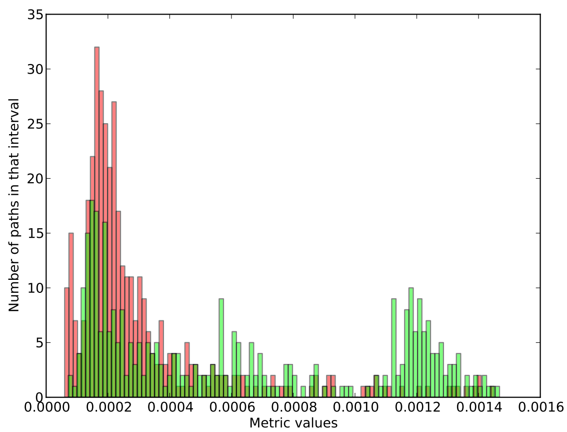


Figure 9: Histogram of red and green clusters for the Dist metric.

design to better balance overall risk, or introduce interesting variation in strategy choices for players.

6. CONCLUSIONS AND FUTURE WORK

Quantitative metrics for measuring player behaviours are an important element in improving and even automating aspects of game design. In this work we described 3 non-trivial metrics for measuring risk in a stealth game context. We showed that our metrics correlate with human perception (to varying degrees), and how the resulting data may be used to understand a level design.

For future work we are interested in further validating and extending our metrics. An investigation of different game contexts and presentations would help determine whether the importance of the distance metric that we find in our study is indeed general. Optimizing the efficiency of metric calculation is also a concern. Our main interest, however, is in applying metrics to improve game design, providing interactive input to the designer, using metrics to quantify design patterns or principles, and in using dynamically computed risk values to identify player styles for improved adaptivity in gameplay and NPC behaviours.

7. ACKNOWLEDGEMENTS

This research was supported by the Fonds de recherche du Québec - Nature et technologies, and the Natural Sciences and Engineering Research Council of Canada.

8. REFERENCES

[1] A. Alix. Beyond P-1: Who plays online? In *Proceedings of DiGRA 2005 Conference*, 2005.

[2] Arkane Studios. Dishonored. `http://www.dishonored.com/`, 2012.

[3] M. Ashton and C. Verbrugge. Measuring cooperative gameplay pacing in World of Warcraft. In *FDG'11: Proceedings of the Sixth International Conference on Foundations of Digital Games*, pages 77–83, June 2011.

[4] R. A. Bartle. Hearts, clubs, diamonds, spades: Players who suit muds. *Journal of MUD Research*, 1, 1996.

[5] S. Bjork and J. Holopainen. *Patterns in Game Design (Game Development Series)*. Charles River Media, 2004.

[6] M. Booth. AI systems of Left 4 Dead. `http://www.valvesoftware.com/publications/2009/ai_systems_of_l4d_mike_booth.pdf`, 2009.

[7] K. Compton, J. C. Osborn, and M. Mateas. Generative methods. In *The Fourth Procedural Content Generation in Games workshop*, PCG, 2013.

[8] Klei Entertainment. Mark of the Ninja. `http://www.markoftheninja.com/`, 2012.

[9] Konami Digital Entertainment. Metal Gear Solid, 1998. `www.metalgearsolid.com`.

[10] A. Liapis, G. N. Yannakakis, and J. Togelius. Towards a generic method of evaluating game levels. In *Proceedings of the AAAI Artificial Intelligence for Interactive Digital Entertainment Conference*, pages 30–36, 2013.

[11] S. Morgan and M. Branicky. Sampling-based planning for discrete spaces. In *Proceedings of the International Conference on Intelligent Robots and Systems (IROS 2004)*, volume 2, pages 1938–1945, 2004.

[12] M. Nelson. Game metrics without players: Strategies for understanding game artifacts, 2011.

[13] Y. Shi and R. Crawfis. Optimal cover placement against static enemy positions. In *Proceedings of the 8th International Conference on Foundations of Digital Games*, FDG 2013, pages 109–116, 2013.

[14] R. Smith. Level-building for stealth gameplay - Game Developer Conference. `http://www.roningamedeveloper.com/Materials/RandySmith_GDC_2006.ppt`, 2006.

[15] P. Sweetser and P. Wyeth. GameFlow: a model for evaluating player enjoyment in games. *Comput. Entertain.*, 3(3):3–3, July 2005.

[16] J. Togelius, R. De Nardi, and S. Lucas. Towards automatic personalised content creation for racing games. In *Computational Intelligence and Games*, CIG 2007, pages 252–259, 2007.

[17] J. Togelius, G. Yannakakis, K. Stanley, and C. Browne. Search-based procedural content generation: A taxonomy and survey. *Computational Intelligence and AI in Games, IEEE Transactions on*, 3(3):172–186, 2011.

[18] J. Tremblay, P. A. Torres, N. Rikovitch, and C. Verbrugge. An exploration tool for predicting stealthy behaviour. In *Proceedings of the 2013 AIIDE Workshop on Artificial Intelligence in the Game Design Process*, IDP 2013, 2013.

[19] J. Tremblay and C. Verbrugge. Adaptive companions in FPS games. In *FDG'13: Proceedings of the 8th International Conference on Foundations of Digital Games*, pages 229–236, May 2013.

[20] Wikipedia. Stealth game. `https://en.wikipedia.org/wiki/Stealth_game`, 2013.