# #164
# ASSISTED AGENT-BASED SIMULATIONS:
## Fusing Non-Player Character Movement With Space Syntax

PETROS KOUTSOLAMPROS
Space Syntax Laboratory, University College London, London, United Kingdom
petros.koutsolampros.11@ucl.ac.uk

TASOS VAROUDIS
Space Syntax Laboratory, University College London, London, United Kingdom
t.varoudis@ucl.ac.uk

## ABSTRACT

Agent-based simulation is one of the core tools of spatial analysis utilised to provide an understanding of space when complex parameters come into play, such as how the visible space changes while traversing a building, or what happens when there is a destination to be reached.

This type of simulation has a lot in common with techniques used in video games to create movement trajectories for non-player characters. Although these techniques have been developed over the years to provide more realistic and more "human-like" behaviour, they are rarely woven back into analytical and simulation tools. As a first step to remedy that, we developed a new methodology that fuses non-player character movement from computer games with simulation techniques traditionally used for agent-based analysis in Space Syntax. This first attempt utilises a different type of underlying representation of space, known as a navigation mesh.

We first examine in detail two traditional techniques utilised in depthmapX agent-based analysis and highlight their strengths and limitations. We then describe how this technique differs from the classic space syntax methods, as well as how it can be combined to create hybrid analytical models of movement. The hybrid model developed in this case is that of a classic space syntax agent assisted by the aforementioned technique. We then tested and evaluated the traditional and new models for their capacity to explore two gallery spaces.

The results extracted from the new hybrid simulation model depict agents with more capacity to explore, a significant addition to the traditional space syntax agent based methods.

## KEYWORDS

Space syntax, agent-based simulation, navigation mesh

## 1. INTRODUCTION

Within the field of Space syntax agent-based models have been employed to provide a stochastic alternative to the deterministic Visibility Graph Analysis, to study how human movement can be approximated, and to act as an alternative evaluation tool to observations for understanding the relationship of movement to the various metrics that describe space. An agent-based model is an example of a simulation in which autonomous agents are left to wander a virtual space having only a small set of simple rules to follow. Through this set of rules and the interactions between the agents a complex system emerges. Batty (2001) specifically highlighted agent-based models as alternatives to large aggregative models which are built from the top down.

In parallel, video games have been utilising agent-simulation as a way to allow non-player characters to move through the virtual game world, and thus provide interactivity to the narrative of the medium. A common method used to allow for this kind of movement is path-finding through a navigation mesh. A navigation mesh is an underlying (non-visible to the player) representation of the 'walkable' space in the game world, and it is comprised of a set of convex polygons that are inter-connected. As mentioned by Snook (2000), its main purpose is to allow for a crude approximation of the virtual world which can in turn be used to calculate paths quickly. He states that "the added bonus is that our replacement for the 2D grid can have cells of irregular shape and size, wind up and down stairs and hills and even overlap itself on things like bridges and catwalks". While this approach is especially beneficial as a 3D representation, the majority of the Space Syntax research focuses on two dimensions and thus we only utilise 2D instances.

In the next two sections we examine the development and general background of the two representations, as well as how they are used today. The next section describes agent-based analysis in general and where the direction decision making process happens, while the two after that examine the two Space Syntax 'look' methods, and the new suggested 'look' respectively. Section number seven describes the method of evaluation for the three techniques as well as the results of this evaluation and the last section discusses the results and provides possible directions for future research.

## 2. BACKGROUND: AGENT-BASED RESEARCH WITHIN SPACE SYNTAX

The Space Syntax field had a long history before agent analysis was introduced, with roots at Hillier and Hanson's 'The social logic of space' (1984) which considered the elements of space (rooms, lines of sight) as parts of a graph, an abstraction that allowed mapping on it different concepts such as social and physical behaviour. This abstraction was combined with Gibson's theory of affordances (1986) and Benedikt's isovists (1979) to create what is known today as Visibility Graph Analysis (VGA). VGA is a framework that allows the analysis space by dividing it into cells and connecting these cells in a graph if they are inter-visible. VGA is deterministic, in that the same spatial configuration will always produce the same result.

Turner and Penn (2002) identified the need for a model of human movement that does not follow a grand theory about the underlying space, but which regards "the environment as the provider of possibilities rather than a place to be rationalised" (Turner and Penn 2002, p.473) and set out to identify whether it is possible, and to what extend, to use configuration to explain movement by using one such model. The authors suggested that other examples of agent-based models which depended purely on the physical displacement of humans (Helbing and Molnar 1998) or those that treated the problem as one of least-cost paths (Hoogendoorn et al. 2002) created agents that lacked a basic driver for natural movement, the ability to see.

Turner and Penn considered this an omission and created an example of agent-based analysis that incorporated this ability. They initially followed a specific idea, that when engaging in natural movement, a human will move towards further available space as determined by his or her current visual field. The researchers thus developed a model with an agent that had a specific visual field which in its turn depends on an EVA (Exosomatic Visual Architecture), in this case a Visibility Graph as those suggested by Turner et al. (2001). The underlying graph had a 0.75 x 0.75 m resolution grid to approximate the average human step length: 0.77m (Sutherland et al. 1994). The EVA allowed the agents to pick a location out of the ones in their visual field, take a step towards that location, change direction and repeat. The implementation favoured the availability of space, thus the agents were more likely to turn and walk towards the areas within their visual field that had more space. The algorithm that made this selection will be referred in this paper as the 'standard look".

The original implementation introduced some limitations to account for the corporeal nature of human beings, i.e. it did not allow two agents to be on the same cell of the graph, or walk or see through walls. These limitations introduced possible gridlocks where the agent had no available space in the visual field. In this case the visual field of the agent was expanded to 360° and a new pixel was chosen through the whole isovist. Turner and Penn proceeded to test this new method against observed movement captured earlier (Turner and Penn 1999) with the aim of finding the combination between number of steps and field of view that had the best correlation with the observed data. The combination that best correlated with human movement was found to be 3 steps before a change of direction and a field of view of 170 degrees. Turner and Penn also tested an implementation of the algorithm that chose directions at random but found that it did not correlate well with the observed data.

In a later study (Turner and Penn 2007) the authors suggested that an agent following a natural movement schema may opt to follow specific paths that open new possibilities for exploration . In order to encode this type of affordance into their model, the researchers extended the agents' available input to include line-of-sight (LoS) information. The agents could thus identify, within their immediate environment, in what direction they could see further and choose to turn and walk in that direction. This algorithm will be referred in the rest of this paper as 'LoS look'.

Both methods were tested for correlation against data collected through observation in the Tate Britain Gallery in London. Both the 'standard look' and the 'LoS look' were found to correlate well with the observed data with coefficients (R2) of 0.79 and 0.78 respectively for 3-step movement. Turner and Penn also calculated two extra measures, the 'total coverage of rooms' which were visited by at least one agent and the 'per-agent cumulative isovist', the mean fraction of building area that could have been viewed by an agent during its visit, had it had 360 degree vision. They suggested that the cumulative isovist gave an idea about how optimised an agent is in terms of explorative ability.

In a study by Penn and Turner (2001) a similar agent model was tested, only this time the agents had access to the 'clustering coefficient' metric of the underlying visibility graph. This provided them with a way to distinguish junctions in the space and move towards them. The new model was tested, along with the 'standard look' one, against observed trails in a department store. The 'standard look' model outperformed the new model most likely due to the fact that the next random step is chosen from the field of view and is thus more likely to stay along long lines of sight.

Most of these models have been tested on exhibition or retail spaces for which the words of Gibson properly describe human behaviour: "When no constraints are put on the visual system, we look around, walk up to something interesting and move around it so as to see it from all sides, and go from one vista to another. That is natural vision..." (Gibson 1986, p.1). Some studies provided the agents with the knowledge of origins and destinations to simulate this effect.

Ferguson et al. (2012) reported two limitations with previous models. The original EVA models used the amount of visible space in front of the agent to determine the next destination and would change direction every three steps. The agents in this model would therefore favour big open spaces simply because an agent standing at the edge of such a space would have the biggest amount of visible space toward the centre. A change to the model suggested by Turner (2004) partially solved that problem by requiring the agents to reach a visible destination before changing direction, but since open spaces are a large percentage of all spaces in a building most destinations would still be in them. Therefore Ferguson et al. (2012) suggested to complement the EVA model with a second lookup table of origins and destinations that reflect each activities at those locations. An agent can then use this information to choose a direction based on how closer it takes him or her to the assigned destination, making this destination a form of attractor.

Implementation of many of the above mentioned methods can currently be found in an open source software tool called depthmapX (Varoudis 2012). This tool is currently used by the Space Syntax community to carry out agent analysis and encapsulates many of the ideas for this and other types of analysis. The original implementation was developed within UCL by Turner (2007) and was eventually re-engineered and open sourced by Varoudis (2012).

## 3. BACKGROUND: VIDEO GAME PATHFINDING AND NAVIGATION MESHES

Like the above-highlighted research, video game development required ways to interact with virtual space. Especially with the advent of three-dimensional computer graphics, space became the primary environment a player navigates, and in order to build believable stories non-player characters were eventually introduced. For the stories to be immersive, video games also required these characters to move naturally so that they can lead players through a space, or act against them. In contrast to research though, video games required this to happen in real time and could not depend on grid-based solutions like the ones developed in the field of Space Syntax. Therefore, other solutions were developed that were more lightweight and provided good approximations of natural movement.

One of these solutions was a navigation mesh (Figure 1, left). First mentioned by Snook (2000) a navigation mesh is a set of convex polygons that signifies the space an agent might walk on. Snook specifically suggested this as a way to provide a crude representation of a 'walkable' area, which is not as detailed as the virtual world and can be used to determine where an agent can walk.

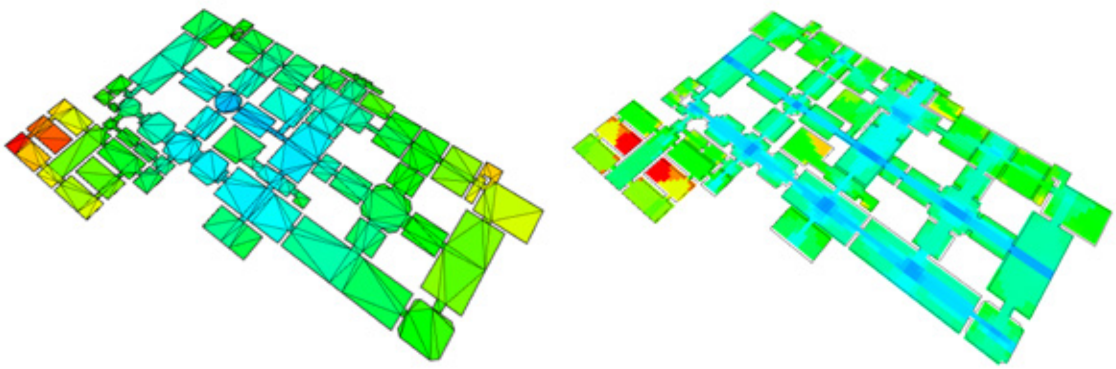

Figure 1 - Depth on a navigation mesh (left) and Visual Mean Depth on a grid (right). From deeper (segregated) areas shown in red, to shallower (integrated) shown in blue

This mesh of polygons can also be thought of as a graph. Each polygon is a node and if two are adjacent, they are connected through an edge. This allows for creation of metrics similar to VGA, such as the mean depth from any polygon to all others, which can be seen in Figure 1 (left) in comparison to the VGA Visual mean depth (right). A navigation mesh also acts as an EVA for non-player characters to utilise, traditionally a path-finding method. The underlying traversal algorithm, typically A* (Hart et al. 1968), is used to identify a series of consecutive polygons (nodes) to go through in order for the agent to reach another area in the map. Another algorithm then provides an actual line-path through these polygons, such as a 'Funnel algorithm'. Cui and Shi (2012) provide an example that describes the whole process. In their simplest implementations most of these algorithms create paths that join the centres of polygons, or the midpoints of the edges.

The lack of regularity of the polygons in a navigation mesh provides specific strengths and weaknesses. It allows us to describe the space in more detail, for example it is much easier to determine where an opening is, given that its vertices are used to generate the mesh. The previously mentioned grid representation suffers from two related problems. If a small cell size is chosen that provides a lot of detail, analysing that grid will require a lot of computational power due to the sheer number of cells. If that is not available, then a larger cell size can be selected which will require less computational power but will also provide less detail, and thus more potential for error. On the other hand, the perfect regularity of the grid allows for much cleaner analysis, since all the elements are the same, while analysis of a navigation mesh will need to control for their size and shape.

## 4. AGENT-BASED ANALYSIS: OVERVIEW

Agent-based analysis in the Space Syntax field is currently done using the software tool depthmapX (Varoudis 2012), although for this study we duplicated the algorithms in a different application that allows for more flexibility in the display of the metrics. The simulation is typically run with a large number of 'agents' that perform a specific process. Each agent is given a certain field of visibility and is left to roam a space for a specific amount of time. Roaming involves the agent deciding a direction, 'walking' in that direction by taking a predefined number of steps and then repeating the process.

Responsible for choosing a direction is a 'look' algorithm. This algorithm takes the agent's position, orientation and environment (EVA) into account and decides what the new direction should be. The agent's visibility system uses a set of 32 'bins', a radial assortment of the pixels around a specific cell according to the angle they are found, as seen in Figure 2. The look algorithm selects a bin and a random direction within that bin to 'step' towards. The various look algorithms help choose the bin by weighting each one of them according to different parameters. This study is interested in the direction decision-making process and thus all other parameters are kept the same throughout all tests.

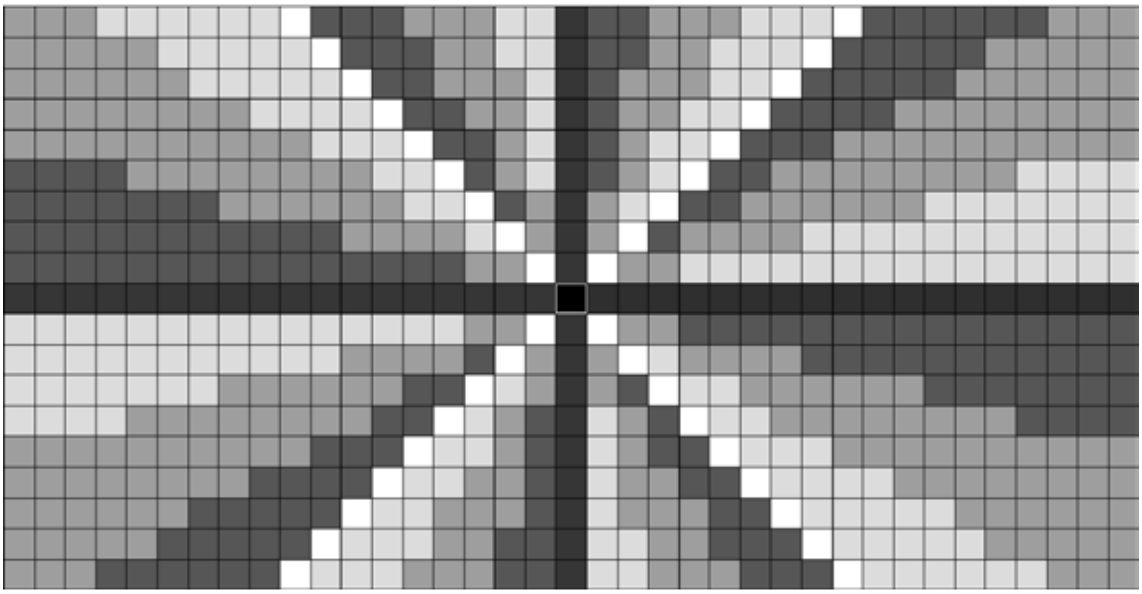

Figure 2 - Cells around a central cell (black) grouped in bins. Adjacent bins in different shades of grey. Counter-clockwise in top-right quadrant, bins: 32 (horizontal left), 1, 2, 3, 4 (top-right diagonal), 5, 6, 7, 8 (vertical top).

## 5. REVIEW OF EXISTING 'LOOK' METHODS

This section will focus on the two 'look' methods already mentioned in the literature 'standard' and 'LoS'. As we described above, each method utilises the underlying EVA to make a decision about which bins to favour more out of the 32 available.

### 5.1 STANDARD LOOK

In the 'standard look' (Turner and Penn 2002) algorithm (Figure 3) before every step a pool of choices is created that contains as many bin choices, as there are cells within that bin. A selection is made at random from this pool that indicates which direction to follow. The bins that contain many cells are thus more likely to be selected given that they were placed in the pool more times. This has the effect of driving the agents towards the largest spaces within their visibility field.

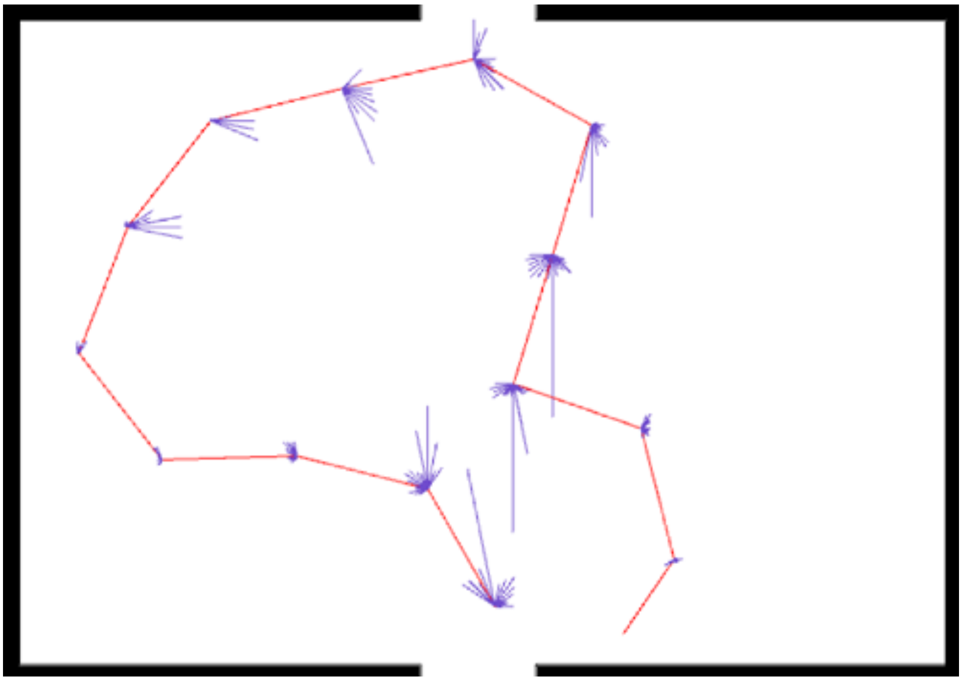

Figure 3 - The path (in red) of a single agent utilising 'standard look' bin weighting in a room with two doors. Decision points noted as radial sets of blue lines representing each bin and each line's length the weight given by the algorithm to that specific bin

The effect of this look technique can be seen at Figure 4. As expected, cells that are close to the walls point towards the centre of the room, while ones closer to the centre have an effect that spreads in all directions. The existence of an opening in this case that leads to more space beyond eventually pulls the relevant bins in that direction. In Figure 4 (right) we can see that this algorithm suffers from a low resolution in the underlying grid. Directions that are equidistant to the walls can have very different multipliers due to the underlying representation.

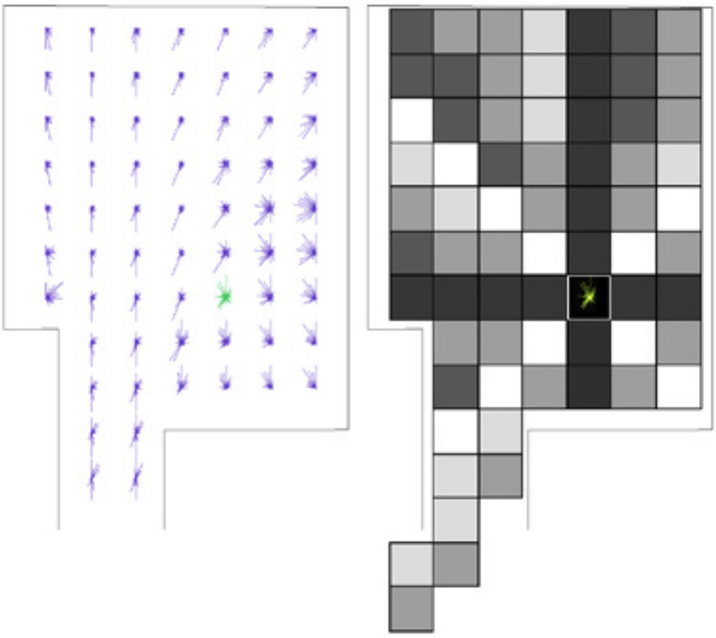

Figure 4 - 'Standard look' at all positions (left) and visualisation of bins for a specific point (right)

### 5.2 LINE OF SIGHT LOOK

The LOS (Turner and Penn 2007) algorithm on the other hand weighs the bins according to the maximum distance visible from within the bin. This distance is calculated by taking the distance of the cell the agent is on to the cell that is furthest away from it within each bin. The bin selection process is the same as the one for the 'standard look' but in this case the algorithm makes the bins that have cells furthest away more likely to be selected.

The overall effect can be seen in Figure 5. Once again, the cells that are closer to the wall tend to have bins that point toward the centre, while cells that approach the centre tend to have more distributed bin weighting. In contrast to the 'standard look' algorithm the 'LoS look' algorithm does not suffer from low resolution as much, although it can still skew the results dramatically as soon as new pixels are visible.

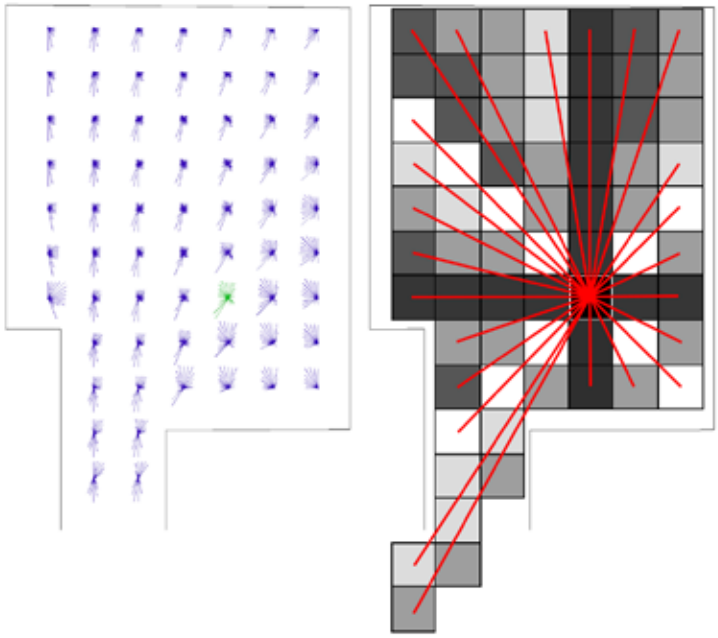

Figure 5 - Line of Sight (LoS) look at all positions (left) and visualisation of bins and distances from a specific point (right)

## 6. UTILISING THE NAVIGATION MESH IN DECIDING A DIRECTION

One of the core features of the Navigation Mesh is that it can describe the surroundings of the virtual world in terms of convex polygons. This allows us to find whether a specific triangle side belongs to a wall (not connected to another triangle), is part of the room (connected to another fully visible triangle) or a passage (connected to a non-fully-visible triangle). Using this categorisation we can identify where a specific agent may enter another room. This information can act as an intermediary step to the previously mentioned 'look' methods. It is information local to the room which points to possible ways out but deals with the actual visible surface instead of the amount of space or longest line of sight. Knowledge of this information (doors or 'gates' between spaces) can act as another 'affordance' used to navigate, as the ones described by Gibson(1977). The navigation mesh representation is not discrete (does not use a grid) but each point lies on continuous space.

### 6.1 ASSISTED STANDARD LOOK

The weighing is based on the angle a bin has from a specific point. If there are multiple passages then all of them contribute to the weighting. Given that this representation is continuous we found it fitting to use a continuous method of weighting to also avoid the pitfalls of discretisation such as slight inaccuracies or problems created from low resolution.

Specifically, a bin is weighted more heavily if a radial line starting from the cell the agent is standing on and passing through the middle of the bin is at a closer angle to the line with the same starting point, but an ending at the middle of the passage. The power of the effect is linear (a bin at twice the angle from the passage will gain half the weighting), but any other function could be used to achieve more or less focus towards a passage.

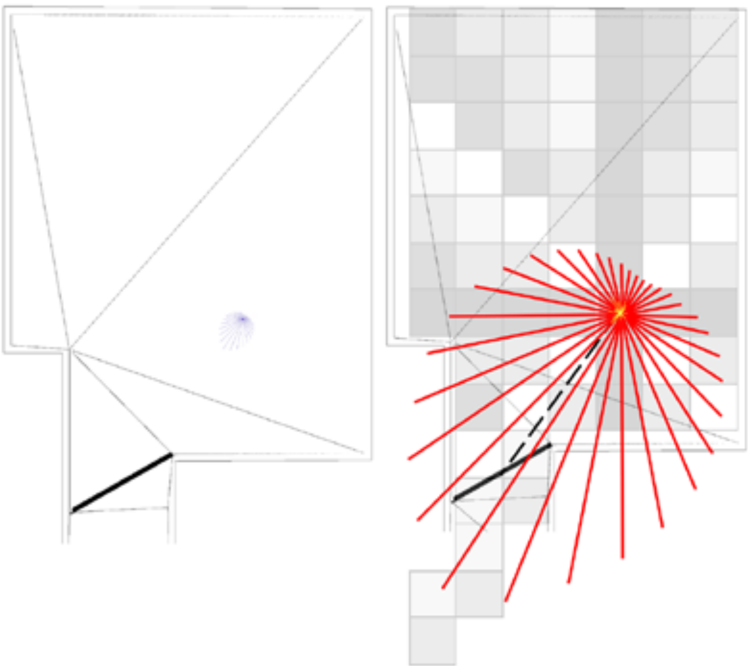

Figure 6 - Passage (heavy black line, left) and effect on each bin (right)

Instead of applying this weighting on its own we tested a fused methodology which takes into account both the 'standard look' weighting and the one described above. We will refer to this approach as 'assisted standard look'. Like the 'standard look' algorithm, each bin is added to a selection pool as many times as it has cells, but this time for each bin this number is multiplied by the weighting. Therefore, in a single-passage example bins that are almost perpendicular to the door will keep about the same weight, the ones that point towards the passage will have their weight increased while the ones that point away from the door will have their weight decreased. The effect is shown more clearly in Figure 7 where the 'standard look' in the previously shown room is affected. The overall effect of the room remains (pull to the centre) but the bins that point to the door are now more heavily weighted.

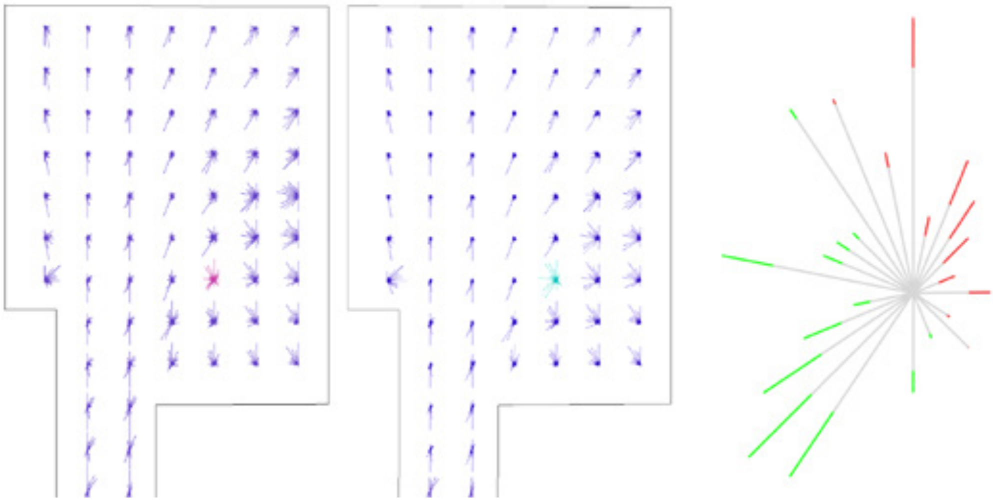

Figure 7 - 'Standard look' (left), effect of angle-to-passage on 'standard look' (centre), and closer inspection of the effect (right) with positive effect (lines made longer, green) and negative effect (lines made shorter, red)

This hybrid look algorithm allows the passage to function like a local attractor no matter how much empty space there is behind it. It can therefore specifically address what Turner and Penn (2002) identified as a possible limitation of the methods that use line-of-sight or walkable surface as affordances: "Our model uses infinite sight, and therefore an infinitely long corridor with respect to side corridors would drive all movement continuously along that corridor, whereas we might expect a human to take an exit some way along the corridor" (Turner and Penn 2002, p.481)

## 7. EVALUATION: AIMING FOR EXPLORATION

In order to build some preliminary confidence about the new methodology we developed two new visual-agent metrics to measure the success of the algorithm, the speed of exploration and the percentage of stuck steps. In the first case we defined an agent as successful if they managed to explore a large amount of space, much like Turner and Penn's (2002) per-agent cumulative isovist. The metric is specifically defined as the number of unique cells 'seen', that is, new cells that appeared within an agent's visual field as they took a new step, divided by the number of steps taken.

We tested four implementations, the standard look, LoS, assisted standard and assisted standard with power of effect at 2, across two buildings, the National Gallery and the Tate Britain Gallery both in London. The field of view and number of steps of the agents before a new 'look' were set as in the original studies by Turner and Penn (2002; 2007) at 170° and 3 steps respectively. In contrast to the original studies, we are not comparing the agents to real people therefore we did not fully take into account the humans' corporeal nature. While the agents could still not pass or see through walls, we did not disallow them to step on a cell already occupied by another agent. Therefore this simulation can only be thought of as a test of the agents' navigating and exploring abilities and may only provide insights about human behaviour in an abstract sense.

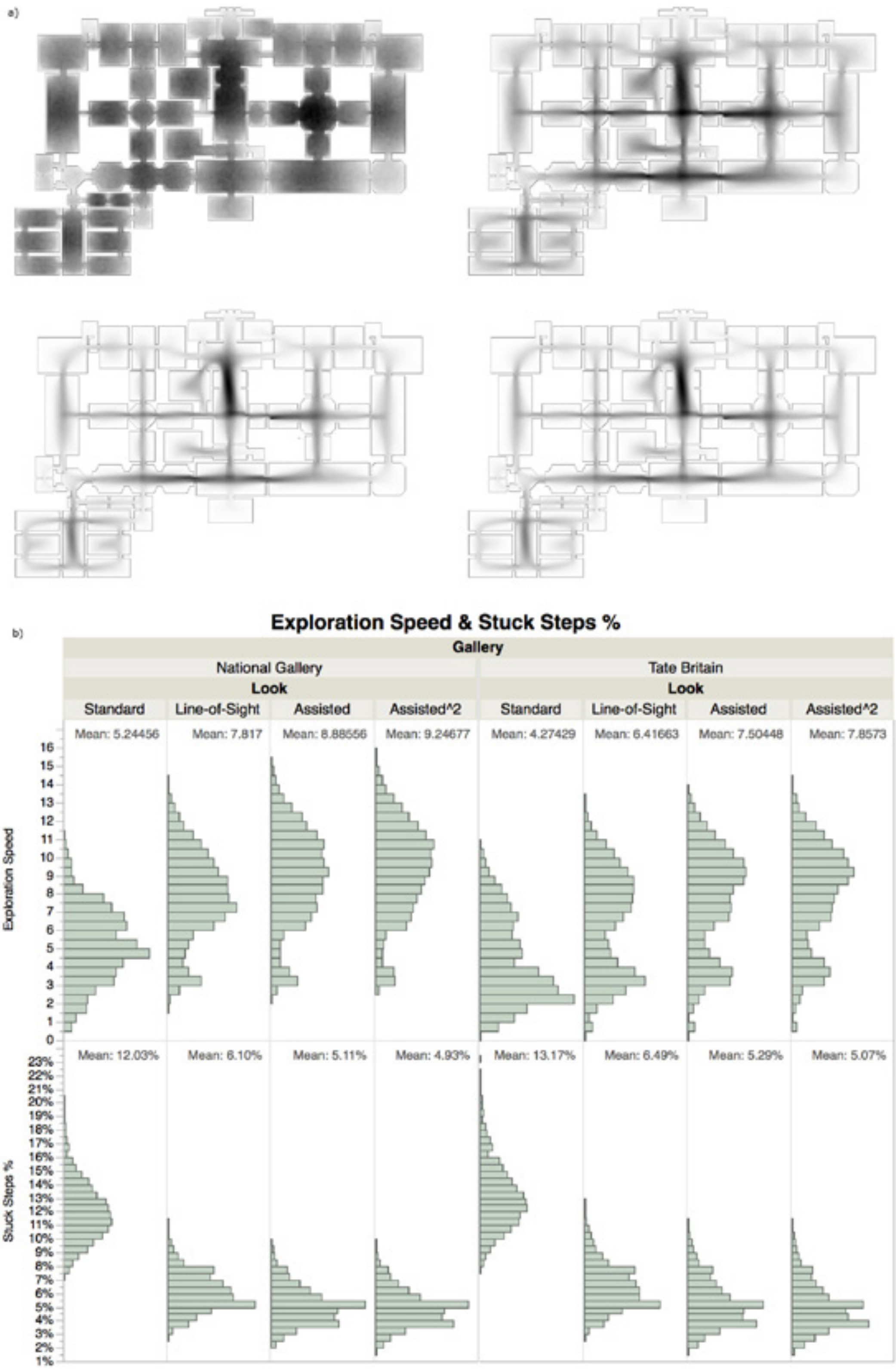


Figure 8 - a) Left to right: standard and LoS (top) portals and portals^2 (bottom) for National Gallery b) Distribution of exploration speed (top) and percentage of stuck cells (bottom) metric for the (left to right) Standard, LoS, Assisted standard, Assisted standard (2) for the National Gallery (left four) and Tate Britain (right four)

tendency for the agents to roam the same rooms or get stuck more. The other three solutions had much higher means, and smaller differences between them. It seems that 'LoS look' is indeed an effective exploration strategy, although not as effective as the new assisted standard, or the slightly more extreme assisted standard with a power of two.

The second metric examined was the number of times the agents got 'stuck' and had to open their field of view to 360° to continue. This can happen if an agent hits a wall or walks into a niche in such a way that its full field of vision is blocked and has nowhere to go. Apart from the fact that such a behaviour would be counter-intuitive for a human, it is not a strategy that optimises for exploration and thus unlikely to allow the agent to view more space. Therefore an agent and in extension the simulation that had less failed steps was deemed more successful in exploring the available environment.

The overall mean percentage of stuck steps was 7.0% for the National Gallery and 7.5% for the Tate Britain Gallery. In both galleries examined the effects are the same. Agents with the 'standard look' algorithm tended to get stuck a lot more than the LoS and the two assisted standard looks, missing on average 12% / 13% of their steps in contrast to the rest which were around 5-6%. Once again, the 'assisted standard look' algorithm (around 5%) fared slightly better than the 'LoS look' (6%), but the effect was lowest at when the power of the algorithm was increased to 2, although marginally. This effect is likely to also cause the lower values observed in the first metric. Agents that get stuck have to reset their field of vision (turn around) due to the fact that they have no available choices to walk towards, meaning that most likely for the past few steps they have been narrowing their field of view moving towards a wall or corner.

## 8. DISCUSSION AND FUTURE PLANS

This study was a fragment of exploratory research that aimed to understand the algorithms used for agent-based analysis within the field of Space Syntax and identify potential ways to expand them, taking into consideration the advances in computer-games research. Relevant developments in video game research were discussed that offer new perspectives to the overarching aim, to simulate natural human movement. We presented an evolution of the Space Syntax methods with techniques from these new perspectives in the form of an 'assisted standard look' algorithm.

We examined and described the inner details of two traditional algorithms described by Turner and Penn (2002; 2007), highlighted their potential and limitations and provided possible ways to introduce novel methods from video game research. We used an underlying navigation mesh representation to allow for the identification of passages when an agent is located in a room, and a new, hybrid look algorithm that affects the 'standard look' with an angle-to-passage weighting.

We then presented a new algorithmic methodology tied specifically to a task -to explore more space- and tested the various 'look' methods against it. We found that while the results are not extremely different between the various look algorithms, we could observe substantial differences, especially between the 'standard look' algorithm and the rest. The differences identified were that agents with the 'assisted standard look' tended to explore more cells, and get stuck less than the ones with the traditional algorithms. This reminding us more the feeling humans have while exploring new, never seen before, topo-geometric layouts and constantly looking for the thresholds to new areas for exploration.

On the other hand, the similarity of the results when using the new algorithm and LOS does not clearly point to one of them as the best. If the discussed methods of evaluation are to be used in a different analysis, the choice of algorithm may need to come down to other factors, such as the available hardware, size of the model or other data. The underlying grid used in the LOS algorithm can make calculation extremely heavy for larger models but it is a simpler representation and thus likely to be followed by other datasets. Other validation methods such as comparisons with real gate counts and movement traces will also be tested in the future to further highlight differences and similarities between these methods.

The ‘standard look’ and ‘line-of-sight look’ algorithms are just two of the many algorithms in depthmapX utilised by the Space Syntax community. In the future we plan to create detailed descriptions of these techniques to gain insights on their possibilities and limitations. We also plan to create implementations of the traditional techniques that avoid the pitfalls of the discretisation, by substituting the grid for visibility in continuous space. These implementations will then be used as bases for new hybrid techniques that take into account other elements of space such as passages, or even transparencies.

This study relied on evaluation techniques that aimed solely for exploration. This type of evaluation is useful for a specific subset of spaces (i.e. galleries) and thus more evaluation metrics need to be examined, in fields beyond Space Syntax. An example of this would be applying agent-simulation in workspaces where the staff is familiar with the space and they are more likely to aim to reach their destination (kitchens, toilets) as fast as possible. Thus, a future implementation will involve the development of techniques that allow the agents to travel through space in search of specific destinations.

## REFERENCES

Batty, M., (2001). Agent-Based Pedestrian Modeling. *Environment and Planning B: Planning and Design*, 28(3), pp.321–326.

Benedikt, M.L., (1979). To take hold of space: isovists and isovist fields. , 6(1), pp.47–65.

Cui, X. and Shi, H., (2012). An overview of pathfinding in navigation mesh. *IJCSNS*, 12(12), p.48.

Ferguson, P., Friedrich, E. and Karimi, K., (2012). Origin-destination weighting in agent modelling for pedestrian movement forecasting. In *Proceedings of the international space syntax symposium*. p. 8153.

Gibson, J.J., (1986). *The Ecological Approach To Visual Perception* New edition., Psychology Press.

Gibson, J.J., (1977). The Theory of Affordances. *Perceiving, acting, and knowing: Toward an ecological psychology*, pp.67–82.

Hart, P., Nilsson, N. and Raphael, B., (1968). A Formal Basis for the Heuristic Determination of Minimum Cost Paths. , 4(2), pp.100–107.

Helbing, D. and Molnar, P., (1998). Self-Organization Phenomena in Pedestrian Crowds. *Physics*.

Hillier, B. and Hanson, J., (1984). The social logic of space.

Hoogendoorn, S.P., Bovy, P.H. and Daamen, W., (2002). Microscopic pedestrian wayfinding and dynamics modelling. , 123, p.154.

Penn, A. and Turner, A., (2001). Space syntax based agent simulation.

Snook, G., (2000). Simplified 3D movement and pathfinding using navigation meshes. In *Game Programming Gems*. pp. 288–304.

Sutherland, D., Kaufman, K. and Moitoza, J., (1994). Kinematics of normal human walking. *Human walking*, 2, pp.23–44.

Turner, A. et al., (2001). From Isovists to Visibility Graphs: A Methodology for the Analysis of Architectural Space. , 28(1), pp.103–121.

Turner, A., (2007). UCL Depthmap 7: From isovist analysis to generic spatial network analysis. , pp.43--51.

Turner, A., Mottram, C. and Penn, A., (2004). An Ecological Approach to Generative Design. In J. S. Gero, ed. *Design Computing and Cognition '04*. Dordrecht: Springer Netherlands, pp. 259–274.

Turner, A. and Penn, A., (2002). Encoding Natural Movement as an Agent-Based System: An Investigation into Human Pedestrian Behaviour in the Built Environment. *Environment and Planning B: Planning and Design*, 29(4), pp.473–490.

Turner, A. and Penn, A., (2007). Evolving Direct Perception Models of Human Behavior in Building Systems. In N. Waldau et al., eds. Pedestrian and Evacuation Dynamics 2005. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 411–422.

Turner, A. and Penn, A., (1999). Making isovists syntactic: isovist integration analysis.

Varoudis, T., (2012). *depthmapX–Multi-platform Spatial Network Analyses Software*, OpenSource.