Computers & Graphics (2021)

Contents lists available at ScienceDirect

Computers & Graphics

journal homepage: www.elsevier.com/locate/cag

Towards a human-like approach to path finding

Vahid Rahmani[a], Nuria Pelechano[b,*]

[a]Deutsches Elektronen-Synchrotron (DESY), Hamburg, 22607, Germany
[b]Universitat Politecnica de Catalunya (UPC), Barcelona, 08034, Spain

ARTICLE INFO

Article history:
Received September 2, 2021

Keywords: Path finding, neuroscience based simulation, autonomous agents

ABSTRACT

Path finding for autonomous agents has been traditionally driven by finding optimal paths, typically by using A* search or any of its variants. When it comes to simulating virtual humanoids, traditional approaches rarely consider aspects of human memory or orientation. In this work, we propose a new path finding algorithm, inspired by current research regarding how the brain learns and builds cognitive maps. Our method represents the space as a hexagonal grid with counters, based on brain research that has investigated how memory cells are fired. Our path finder then combines a method for exploring unknown environments while building such a cognitive map, with an A* search using a modified heuristic that takes into account the cognitive map. The resulting paths show how as the agent learns the environment, the paths become shorter and more consistent with the optimal A* search. Moreover, we run a perceptual study to demonstrate that the viewers could successfully identify the intended level of knowledge of the simulated agents. This line of research could enhance the believability of autonomous agents' path finding in video games and other VR applications.

1. Introduction

Path planning for autonomous agents and robots has been widely applied for many decades in fields such as robotics, video games and populated virtual realty. In autonomous agents and multi-agent simulation, the problem is focused on finding an optimal path between two points, although often sub-optimal solutions are accepted if time performance is critical [1].

Currently, most path planning methods aim at planning an optimization model that considers one or more features (e.g: distance, energy, or smoothness) and then conducting a minimization procedure to achieve an optimal path [2].

While existing techniques give possible solutions for practical applications, they do not typically take into account human factors to closely simulate how humans behave in the real world, which could enhance the overall realism of video games. There are many aspects of human behavior that affect route choice during navigation, such as: memory, mental maps, or visibility. Some of these aspects, have been included in previous models, such as visibility [3] or memory [4]. Studies from neuroscience research have observed that mental maps are built as we physically move through an environment [5]. The method that we present in this paper is inspired by research from neuroscience, and its novelty lies in building agents' mental maps, following the human brain navigation research. According to the theory known as "the GPS of the brain" [5] mental maps are built by cells being fired in the hippocampus region of the brain as we physically move through the environment. Collectively these cells create a coordinate system, with a hexagonal grid shape, that allows for spatial navigation.

When a human is looking for a path within a large environment, such as a city, there can be two opposite scenarios: (i) The person knows very well the city, or (ii) the person has never been in the city before. Of course there can be many situations in between, such as the person knows very well a part of the city

*Corresponding author: Tel.: +34-93-413-7858;
e-mail: npelechano@cs.upc.edu (Nuria Pelechano)

but has no information about other parts. We will first focus on providing algorithms for cases (i) and (ii), and then combine both algorithms to fit any situation in between those two cases. In the first situation, the person has a mental map in his memory and will choose a path based on the previous knowledge. In our work, we will build this mental map based on the human GPS of the brain theory [5, 6]. In the second situation, when the person does not know the environment, he can either try to find the given goal position randomly or follow some vague knowledge (for example to simulate how humans move around an unknown city after looking at a map or asking for directions). We thus simulate how humans wander through an unknown environment, having some vague idea of where the goals could be roughly located. A completely unknown environment would require an exhaustive search such as Breath First Search or Depth First Search which is rarely used by humans, as we would expect humans to ask for guidance or else have a quick glance at a map.

In this paper we propose a novel path finding method for intelligent agents that better simulates humans by implementing methods based on the human brain research. We build mental maps considering how humans learn about the environment and memorize spatial information following the GPS of the brain theory. In order to navigate known environments, we propose a path finding method that uses such mental maps. To explore unknown environments, we propose a naive path finding method which considers the confidence level regarding goal direction, while attempting to walk along the line of sight, to better mimic humans' decision making [7]. Then we combine known and unknown areas to propose a novel path finding model that better resembles what we would expect humans to do, and show the resulting paths based on the levels of knowledge assigned to the agents. Finally, we present a user study which demonstrates that the resulting paths are perceived as being consistent with the level of knowledge assigned to the agents. We focus exclusively on path finding within a navigation mesh, since collision avoidance could be handled with any local movement method [8].

This paper is structured as follows: in the next section, we introduce the theory of the GPS of the brain, followed by related work on path finding. The next section explains our approach in detail, and finally we present results and conclusions.

2. Related work

There are many aspects that are relevant when simulating autonomous agents, and thus this research gathers knowledge from biology, neurosience, human behavior, artificial intelligence and robotics. We will classify thus the related work, according to these areas.

2.1. Human brain navigation

In 1971, John O'Keefe [9] discovered the first key to the inner GPS in the mammal's brain which is called place cells. He recorded nerve activity in the hippocampus region of the brain in unobstructively moving rats. He obtained single cells that

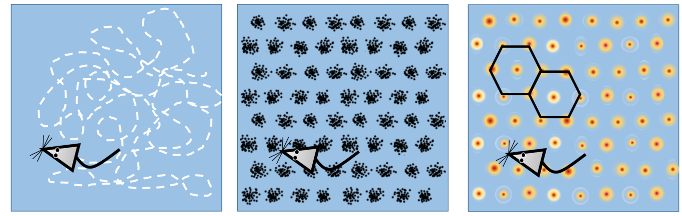

Fig. 1. A schematic drawing of grid cell firing as the rat moves through a square [5]. On the left, rat with trajectory, in the middle grid cells firing as the rat moves on the floor, and on the right the hexagonal pattern formed, which gives high spatial resolution that allows the animal to recognize its locations and orientation.

just activated when the rat was in a certain location in the environment. These places cells were active for different locations, generating an inner map in the hippocampus region of the animal brain showing the animal where it is in the environment.

In 2005 May Britt and Edvard Moser [5], observed cells in the Entorhinal cortex of rat brain. Here, they obtained nervecells that were not active in just one location but fired when the rats passed multiple locations.

Each of these cells were fired in a single spatial pattern and collectively these grid cells create a coordinate system, with a hexagonal grid shape, that allows for spatial navigation. Even though this studies have been mostly performed with rats, this nerve cell system has been observed also in rodents, bats, monkey and humans. Studies using functional magnetic resonance imaging (fMRI) have provided evidence for grid-cell-like representations in humans, implicating a specific type of neural representation in a network of regions which supports spatial cognition [6]. In 2014 the research carried out by E. Moser, J. O'Keefe and M. Moser was awarded the Nobel Prize for the discovery of what is known as the inner GPS of the brain. Their findings provided an explanation regarding how the brain builds cognitive maps, by firing neurons in a hexagonal pattern as we move through an environment. Therefore, from a human simulation perspective, it is key to consider such hexagonal formation, with cells being fired based on movement as a more plausible model to simulate human spatial memory and path finding.

2.2. Biologically inspired path finding

A large number of robots have been built that explicitly simulate biological navigation behaviors for obstacle avoidance, such as the ones simulating the migration of seabirds [10] and ant colony behavior navigation model [11]. In these cases, animals work collectively to determine the final navigation formation. Inspired by the social interplays in human crowds or animal swarms, Savkin and Wang [12] have proposed an efficient obstacle avoidance method in dynamic environments by combining representation of the information about the environment. Compared to ant colony optimization were all ants contribute to the map with pheromones, our maps are built for each individual following the GPS of the brain theory.

2.3. Path finding by minimizing cost functions

Artificial potential fields (APF) [13] are generated by considering repulsive and attractive spaces. The obstacles will be considered as repulsive areas, and the goal position is considered as an attractive area in the artificial potential field. The APF provides smooth paths, but the main disadvantage of APF is that it suffers from local minima problems. The concept of potential fields has been combined with path planning to impose constraints when searching for a path [14]. Influence maps and potential fields build maps that are shared by all the agents in the simulation, as opposed to our maps which are build individually for each agent.

Sampling Based Planning (SBP) methods [15] are the most important improvement in path planning Main benefits of SBP methods are that they have very low computational cost and also they have applicability to high dimensional problems with better success rate for complex queries. SBPs are probabilistically complete, and the paths created by these algorithms for the same problem are not unique.

The A* search algorithm [16] is the most popular algorithm for path finding, since it has many beneficial properties. First, it provides an optimal path between the given start and end positions in a scenario. Secondly, it has the ability to return a result in a finite time even in the case that there is no solution for the problem. Thirdly, a suitable admissible function can lead to an acceptable time-consuming even for a big scenario. Today, there are many variants of A* search algorithm to deal with different problems and tasks, such as D* Lite[17] any-angle A*[18], Phi*[19] and Field D*[20], and hierarchical approaches [21, 22].

Gradient-based methods [23] use a direct line to connect the start and goal points (even if it goes through obstacles), and then eliminate the obstacle points in the gradient directions. Since this method creates non-smooth paths, a post-smoothing process is required.

Considering current path finding methods, A* search algorithm offers the possibility of being deterministic and highly adaptable, mostly by altering the heuristic function. A* will find an optimal solution, however, humans are not always likely of finding an optimal solution, specially when not all the environment is fully known. Therefore, to carry out path finding in partly known environments, we will use A* with a new heuristic function that considers human-like cognitive maps and explores the environment based on the reliability of the acquired knowledge.

2.4. Path finding in partly known environments

Path finding in unknown environments has been frequently used in robotics, where an agent gather information from sensors while navigating the space. In video games, typically agents have access to all the information about the environment. However some methods dealing with partly known environments have been introduced to avoid having autonomous agents with the "super powers" [4]. D* is an incremental search algorithm, which plans the shortest paths under the assumption that there are no obstacles between the agent position and the goal [24]. When obstacles are encountered, the algorithm can replan the route. There have been many variants of D* to optimize the new search and to deal with dynamic environments [25].

Simultaneous localization and mapping (SLAM) uses the data captured by the externally perceived sensors to self-locate and build a map of the surrounding environment at the same time. Currently, there are many methods for locating and mapping (SLAM) that allow a single agent or a group of agents to gather knowledge from their environment and generate the map by employing different sensors like laser telemetry sensor or LIDAR [26, 27, 28, 29].

2.5. Path finding based on human behavior

There has been a large number of studies on human wayfinding both in open and close spaces. Although there are still many unknowns about the functioning of the human brain, there have been some interesting findings regarding humans' navigation.

Dalton [7] observed that humans appear to be attempting to conserve linearity throughout their paths, choosing the straightest possible routes instead of the more meandering ones. According to Dalton, people may unconsciously prefer a straight path as a complexity-minimizing strategy. Previous research by Hillier [30] had already observed that people tend to follow the longest line of sight that approximates their heading. Other researchers support that humans try to minimize the angle change in a route [31, 32], which corresponds to the phenomenon of considering a route to be longer if it has many changes in direction as opposed to a straighter one of identical length [7]. Garling and Garling [33] observed that all pedestrians may not adopt path minimization (in terms of time or distance), but it appears to be a dominant characteristic, and that minor changes in direction tend to be preferred over great changes in direction, possibly due to an innate human tendency to avoid getting lost [34].

Human path finding is the result of combining many different elements, such as visibility, memory, mental maps, landmarks, etc. Some authors have made the effort to include some or several of these aspects in their path finding models. For example, Space syntax [35] simulates human trajectories based on visibility, and then achieve simulation models that quite accurately represent the real use of space in the city of London. Reiter and Lebiere [3] proposed a twocomponent cognitive model that combines retrieval of knowledge about the environment with search guided by visual perception. Sohn et al. [4] presented a wayfinding model to simulate more human-like agents, by having agents with imperfect memory that could build a mental map from landmarks. Their cognitive map is based on a spring-mass system, where landmark memorability is encoded by the mass.

Another interesting theory is the hypothesis of cognitive graphs by Warren [36]. A cognitive graph is a network of paths between places labeled with approximate local distances and angles. The main difference with a metric cognitive map, is that this local information is not embedded in a global coordinate system, so spatial knowledge is often geometrically inconsistent. This closely represents the fact that humans can make direction estimates with large angular errors, and distance estimates which are largely biased by landmarks. This hypothesis has been evaluated in virtual reality where participants navigate an environment wearing a head mounted display and have to perform a series of tasks to find evidence of whereas human memory works over Euclidean maps or cognitive maps [37]. The results of their experiments suggest that: either spatial knowledge is Euclidean, but too imprecise to support metric shortcuts to discriminable locations; or else that spatial knowledge is non-Euclidean, and best described as a labeled graph. This theory is still being investigated by running experiments with humans in VR. It will be interesting to know how direction and distance accuracy improves with experience. Our method shares some characteristics with Warren's hypothesis, such as the concept of humans making direction estimates (during our naive search), and distance errors being reproduced with the cell counters (during A* with modified heuristic).

Another aspect of human decision making that has been incorporated in path finding, is the fact that humans tend to plan in an abstract manner and then break the problem down into its smaller pieces [38]. There have been several approaches simulating this type of hierarchical path finding, where the agent plans a high level path and then computes the coarser details of the trajectory for each high level node [39, 40, 22].

3. Human-like path finding model

Our goal is to create a path finding model to better resemble human behavior. Most previous models focus on finding optimal paths, smooth paths, and/or finding solutions within certain time constraints. Some previous models have included visibility, landmarks, memory or mental maps as representations of parts of the navigation mesh. Our model focuses on building the first path finder method that resembles the human brain memory as a hexagonal map which is iteratively built based on previously visited places. The proposed method consists of two phases: (1) Exploration to generate the cognitive map and (2) path finding based on the current map information.

Most navigation maps in the literature used for path finding, consists of either a regular grid of squared cells, or a polygonal mesh. Hexagonal grids are rare, although sometimes used because they have the advantage that moving to an neighbouring cell has the same cost regardless of the direction [41]. Our work proposes a navigation mesh consisting of regular hexagons, with the goal of resembling the brain grid cells structure.

For the case of searching for the goal in an unknown environments, the agent needs to search with a naive approach while building such map. This is the human equivalent to wandering an unknown environment, choosing directions at each intersection based on naive knowledge (i.e. approximate directions), and learning about the environment as we walk. Since we know that human cognitive maps are created by firing the neurons of the hippocampus region of the brain when the person visits a location in the environment, our agents will thus fire the cells corresponding to the location that the agent walks by. This cell firing is implemented with a counter that increases as the agent walks repetitively through a location. Therefore, the value of the cell counter is an indicator of how well the agent knows that corresponding location in the virtual space.

Our path finding algorithm inspired by the GPS of the brain, will then use the value of the counters to introduce a new heuristic function for the A* search. We give higher priority to choosing paths that move through cells with higher counter value. This situation is the human equivalent to thinking about the best known path and then following it. Figure 3 shows the counter values of each hexagon when the agent searches the environment.

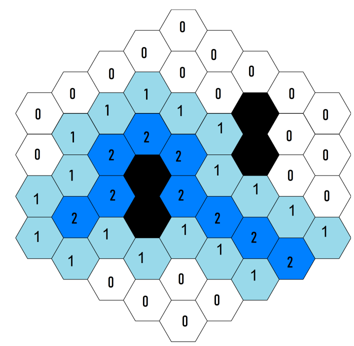

Fig. 2. Hexagon grid cell with counters. Black cells are obstacles, and the intensity of blue represents the counter value.

Since mental maps are built from the agents' movement, we first explain how our agents explore an unknown environment while building such maps, and then we propose a method that uses those mental maps to carry out an A* search with a new heuristic, driven by the cell counters.

3.1. naive exploration for unknown environments

When a human is located in an unknown environment for the first time, he would have no mental map of the environment in his brain. In order to search for a path between S and G, he would have to fully explore the environment. There are well known algorithms to perform a full exploration of an unknown environment such as Depth First Search (DFS) or Breadth First Search (BFS). However, humans do not typically perform such a blind exhaustive search. Whereas we are inside a large building or outdoors in a city, it would be reasonable to consider that humans would either have a glance at a map, or else ask for basic directions. So, in order to simulate human path finding behavior in unknown environments, we have developed an algorithm that performs a naive search with some basic knowledge of directions. Our method also assumes that humans prefer to walk along the longest sight-line towards the goal direction, which is a behavior that has been reported on real humans [42]. Our algorithm is based on DFS but neighboring cells are not explored in random order, instead, a specific order is followed to prioritize two concepts: (1) choose first cells that are closer to the goal direction according to the agents' naive knowledge, and (2) once a direction has been chosen, follow it for a several steps to reduce the number of turns in the path.

Therefore, our agents' naive exploration algorithm is based on two principles: Firstly, humans are likely to perform exploration in a sequential manner moving towards a goal G, and thus our search is based on DFS with a greedy heuristic based on a confidence level of the goal direction, and secondly humans are likely to walk along the line of sight [7], and only reconsider direction if they feel that they are not moving towards the goal (or encounter an obstacle). The idea of using visibility to walk along the longest line of sight, has been used in space syntax methods such as the work by Penn et al. [35], our model differs in that we do not choose the longest line of sight, but a direction that is approximately aligned with the goal orientation, and then use the concept of walking along the chosen line of sight (thus reducing the number of turns in the trajectory) as long as the agent does not sense that it is no longer moving towards the goal direction.

When the agent is located in an unknown environment, all cell counters are set to zero (no cognitive map). The agent will then move towards the goal position with our naive approach. The first step is thus to compute the forward direction for the agent. As shown in figure 3, each hexagon in the grid cell has at most 6 neighbors. In order to calculate the movement direction, we calculate the vectors $\vec{u_i}, i \in [1, 6]$ which point from the current cell towards each neighbor, and keep those that are approximately in the desired direction of movement, as the set $\mathbb{D}_{(G,\delta)}$. Note that vectors pointing towards an obstacle cell will be discarded since they do not represent a valid agent movement.

The set of possible directions of movement is thus:

$\mathbb{D}_{(G,\delta)} = \bigcup \vec{u_i}$, such that $cos(\angle(\vec{u_i}, \vec{u}_G)) > \delta$

$\vec{u}_G$ is the unit vector from the current cell pointing towards the goal G. The confidence threshold, δ, is a user defined value in the range $[-1, 0.8]$, and will determine the number of vectors in $\mathbb{D}_{(G,\delta)}$. The next direction $\vec{v}$ chosen for graph exploration is randomly picked from the set of directions $\mathbb{D}_{(G,\delta)}$ (note that since both $\vec{u_i}$ and $\vec{u}_G$ are unit vectors, the cosine can be efficiently computed using the dot product). Figure 3 shows a representation of the vector pointing towards the goal $\vec{u}_G$, and the set of vectors that will be inserted in $\mathbb{D}_{(G,\delta)}$ if $\delta = 0.5$. Note that when $\delta = -1$ we would have a completely uninformed search similar to DFS. The key in our method is that δ represents the level of confidence regarding the goal direction. The larger the δ the more directly the agent will explore the straight line towards the goal direction. The maximum possible value of δ is 0.8 which corresponds to an angle of 30^o with respect to the goal direction $\vec{v}_G$ and guarantees that there will be at least one possible direction in our hexagonal grid. Having a large value of δ will give fewer possible directions pointing towards the goal direction. If $\mathbb{D}_{(G,\delta)} = \emptyset$ then the next direction $\vec{v}$ is chosen randomly among the possible directions of movement (towards obstacle-free cells).

The agent will follow direction $\vec{v}$ until it either hits an obstacle or its trajectory appears to be moving away from its desired direction. This second case is done by triggering a new computation of preferred direction every $\mathcal{S}$ cells. We have empirically found that $\mathcal{S} = 5$ provides good perceptual results for our naive search exploration.

This parameter can be defined by the user, or can be set to a range of values to provide more heterogeneous graph exploration in the case of crowd path finding. If $\mathcal{S} = 1$ the agent would recompute its direction at every cell, whereas $\mathcal{S} = \infty$ implies recomputing direction only after a collision. The larger the value of $\mathcal{S}$, the longer the agent walks along the previously chosen direction (simulating walking along the longest line of sight or minimizing turns). However, large values of $\mathcal{S}$ can also make the agent move further away from the goal direction.

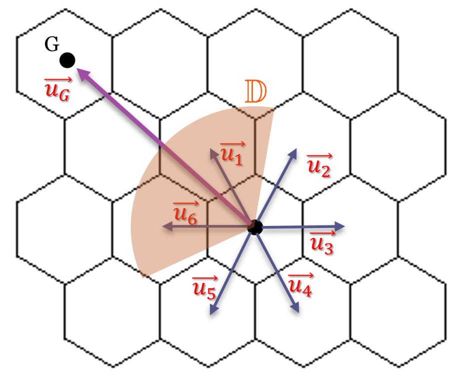

Fig. 3. Representation of the goal direction ($\vec{u}_G$), with the six possible directions of movement ($\vec{u}_i$) and the circular sector indicating the directions included in $\mathbb{D}_{(G,\delta)}$ for a given confidence threshold δ.

Figure 4 shows on the left an example path of our naive path exploration algorithm (in pink) in an unknown area of the environment, with $\mathcal{S} = 5$, and $\delta = -0.4$, against the A* solution (in green). Note that our naive exploration algorithm has the agent walking along the cells whilst exploring, and thus there can be some small loops in the trajectory, which can resemble when a human needs to walk back to a street junction after realizing he may not be in the right path towards the destination. On the right side of Figure 4 we can see an example path with with $\mathcal{S} = \infty$, meaning that the agent only turns after finding an obstacle cell and and $\delta = -1$, meaning that any non-obstacle neighbour can be picked for such turn. In both cases, as the agent walks by the environment, the cells counters will be increased to build its mental map.

3.2. Known environment

Path finding in a known environment can be done using the agent's cognitive map that has been built by performing many searches following our naive exploration algorithm. Every time an agent visits a hexagonal cell, the counter value of the corresponding cell is increased. By increasing the counter values of the cells, the agent's knowledge about the environment is also increased. In order to find a path in a familiar environment, we use the A* search algorithm, but with a modified heuristic function that takes into account the agent's knowledge.

Given a goal position G and the starting location of the agent S, we need to find the path that can get the agent from S to G avoiding the static obstacles in the environment. Note that obstacle cells will not appear in the mental map, as the agent will not have walked through them previously. The agent needs to find a path $\pi = \langle S, p_1, p_2, ..., G\rangle$ by running the A* search algorithm with a modified heuristic. The main key for the A* search method is to define an admissible heuristic function $h(p_i)$, so that it avoids overestimating the actual cost to arrive at the goal location. In this paper, we define the heuristic function $h(p_i)$ from a point p_i to a goal position G as follows:

$$h(p_i) = \|p_i - G\| + \lambda_i \tag{1}$$

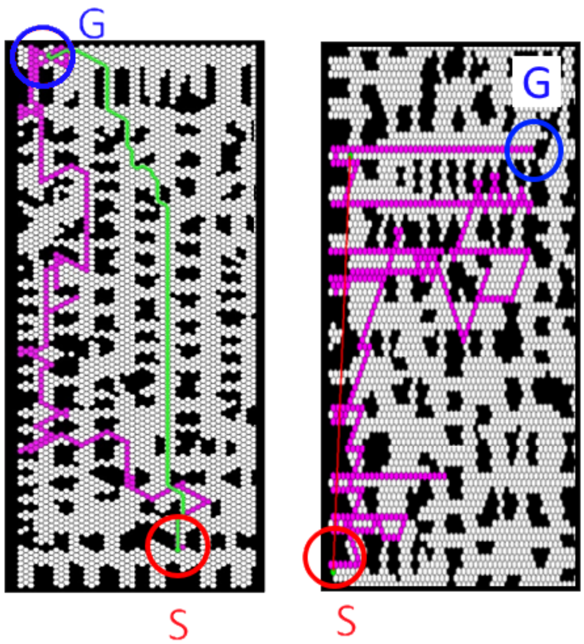

Fig. 4. On the left, comparison of the path obtained with the naive exploration algorithm when $\mathcal{S} = 5$ (in pink), and the A* optimal solution (in green). On the right, path obtained with the naive exploration when $\mathcal{S} = \infty$, and thus the agent only replans its direction when finding an obstacle.

where $||p_i - G||$ is the 2D Euclidean distance from the current position to the goal G, and the term λ_i is defined as:

$$\lambda_i = \begin{cases} 2 \times C_{max} & if\ c_i = 0 \\ \frac{C_{max}}{c_i} & if\ c_i > 0 \end{cases} \quad (2)$$

C_{max} is a user defined value, which sets an upper limit to the level of knowledge about a cell. Therefore, the largest heuristic would be assigned to those cells with counter, c_i, equal to 0 (unknown cells). For known cells, the heuristic value becomes smaller as the counter increases, and thus known cells have higher priority in the A* algorithm to be selected for exploration. When all cells have the highest counter value, our algorithm is equivalent to a basic A* search.

With our proposed heuristic function, agents will find paths towards a goal position based on their previous knowledge of visited places (i.e. their cognitive map). This heuristic makes agents more likely to move within familiar areas of the environment, and only when knowing the entire environment would they be able to find an optimal path. Unknown cells are thus avoided.

3.3. Combining the known and unknown

Even though we have presented so far two different algorithms for path exploration, based on whether the environment is known or not, it is of course possible to encounter scenarios with partly known areas. In such case, both algorithms can be combined, so that, if the agents has knowledge about the area covering the space between the start and goal positions, then algorithm A* with modified heuristic is applied, but when

Algorithm 1: Neuroscience based path finding

Procedure `NeuroSciPathFinder`$(x, G, steps, \vec{dir})$
  if $x = G$ **then**
    `// has reached the goal G`
    return
  end if
  if $x \neq currentCell()$ **then**
    `// walk from current cell back to x to continue exploring`
    $path.walkBackTo(x)$
  end if
  $c \leftarrow get\_counter(x)$
  $c.increaseCounter(x)$
  $x.visited(TRUE)$
  $path.addCell(x)$
  if $c = 0$ **then**
    `// Exploration with naive direction`
    $\mathbb{D} \longleftarrow getDirsNotVisitedNeighbors(x)$
    if $steps < \mathcal{S}$ ***and*** $\vec{dir} \in \mathbb{D}$ **then**
      `// continue with current dir`
      $\mathbb{D} \longleftarrow \mathbb{D} - \{\vec{dir}\}$
      $steps++$
      $NeuroSciPathFinder(x, G, steps, \vec{dir})$
    end if
    $steps = 0$
    for $i \leftarrow 0\ to\ |\mathbb{D}|$ **do**
      `// explore first directions in` $\mathbb{D}_{(G,\delta)}$
      if $cos(\vec{u}_i, \vec{u}_G) > \delta$ **then**
        $\mathbb{D} \longleftarrow \mathbb{D} - \{\vec{u}_i\}$
        $NeuroSciPathFinder(x, G, steps, \vec{u}_i)$
      end if
    end for
    for $i \leftarrow 0\ to\ |\mathbb{D}|$ **do**
      `// explore the rest of directions`
      $NeuroSciPathFinder(x, G, steps, \vec{u}_i)$
    end for
  else
    `// A* with GPS counter heuristic`
    $cell = A^*\_withGPSHeuristic(x, G)$
    $NeuroSciPathFinder(cell, G, \mathcal{S}, \vec{0})$
  end if
end

the agent is in an unknown cell, the naive search algorithm is executed to further explore the environment. By alternating between both algorithms, the agent will gradually increase its internal cognitive map based on the GPS of the brain. The details of our path planning method are shown in Algorithm 1.

4. Experimental Results

In this section we present the results achieved for both informed and uninformed path finding, and compare against A*. We show the visual aspects of the paths, as well as paths length. We have also run a perceptual study to evaluate whether the resulting paths correctly represent the levels of knowledge of the agents.

4.1. Evaluation of path lengths based on knowledge

In order to evaluate the effect of the percentage of agent's knowledge, $\mathcal{P}$, on the resulting path, we have tested the proposed path finder on $\mathcal{P} = [25, 50, 75, 100]\%$ of known cells which were previously visited by the agent. Figure 5 shows results for different percentages of agent's knowledge. By increasing the percentage of agent's knowledge, the length of the informed path (in pink) will get shorter and the path will get closer to the result provided by the A* search (in green).

For the completely unknown parts of the environments (white cells), we should expect our agents to choose a path that appears rather random and is far from optimal. The random appearance is the result of agents needing to explore and thus checking more locations.

In order to see the agent's behavior as we gradually move from unknown to known areas, we have computed paths in different regions of an environment which is only partly known. Figure 6 shows the resulting paths (in pink) for both our methods combined (naive exploration and A* with GPS heuristic) against the A* solution (in green). As we can see in Figure 6 (left), when the agent is located in an unknown area , it first explores a path moving roughly towards the goal direction. Our naive exploration method avoids the agent from moving too far off the goal direction by using a large confidence threshold, $\delta = 0.4$, and number of steps along the line of sight, $\mathcal{S} = 6$, which overall shows high confidence on the goal direction, but not knowing its exact position. Figure 6 (middle) shows how the path shape and length gets closer to A* when searching on areas of the environment that the agent knows very well (darker blue indicates higher counter values). Figure 6 (right) shows the combination of exploring initially an unknown area and then performing A* with GPS heuristic when entering a known part of the environment.

To further understand how the algorithm parameters affects the visual results, Figure 7, shows paths followed by an agent based on the values of $\mathcal{P}$, δ and $\mathcal{S}$. Notice that $\mathcal{P}$ only affects the path on known areas, whereas δ and $\mathcal{S}$ only affect the path on unknown areas.

To show the paths followed by an agent based on the percentage of known cells, we show results for which we fix δ and $\mathcal{S}$, while increasing $\mathcal{P}$. Figure 7 top row shows the paths for a give environment with the levels of $\mathcal{P}$ increasing from left to right (map 0:{$\mathcal{P} = 0\%$}, map 1:{$\mathcal{P} = 35\%$}, map 2:{$\mathcal{P} = 75\%$}, map 3:{$\mathcal{P} = 100\%$}), while all four maps have $\delta = 0$ and $\mathcal{S} = 5$. As we can see, with increasing values of $\mathcal{P}$, the resulting paths become shorter. The chosen values for δ and $\mathcal{S}$ steer the agent towards the goal direction quite efficiently, and that is why even for low values of $\mathcal{P}$, the agent appears to explore the path quite efficiently.

The bottom row of Figure 7 had values for δ and $\mathcal{S}$ that better matched the level of confidence with the levels of knowledge, meaning that low knowledge is accompanied by less confidence on the goal direction, whereas as we increase the knowledge, we also increase the confidence regarding the goal direction. This is achieved with the following configurations: map 0: $\{\mathcal{P} = 0\%, \delta = -1, \mathcal{S} = 10\}$, map 1: $\{\mathcal{P} = 35\%, \delta = -0.7, \mathcal{S} = 8\}$, map 2: $\{\mathcal{P} = 70\%, \delta = 0, \mathcal{S} = 6\}$, map 3: $\{\mathcal{P} = 100\%, \delta = 0.4, \mathcal{S} = 4\}$.

In this case, the level of knowledge affects both the path finding in known areas, but also the search on unknown areas, where the agent's ability to guess the direction towards the goal also varies. As we can see in the resulting paths, the first image shows a path that demonstrates very little knowledge of both the map and the goal direction, and as we move towards the right maps, the resulting paths exhibit more knowledge for both the counter based path finder, and the naive exploration with increasing confidence on the goal direction.

In order to show the results of the exploration part of the algorithm, we have searched for paths in another environment with varying values of δ and $\mathcal{S}$, but with the $\mathcal{P} = 0$, so that the algorithm cannot run the A* with GPS counters (see Figure 8). The top row shows paths with $\delta = -1$, meaning that the agent can choose any random direction because it has no confidence on the goal direction; and $\mathcal{S} = 10$, meaning once a direction has been chosen, it will continue with the same direction for 10 steps, as it has no confidence on how far it may be steering away from the goal direction. The bottom row shows paths with $\delta = 0.4$ and $\mathcal{S} = 5$ which steers the agent quite confidently towards the goal even for the exploration part of the algorithm. As we can see in the figure, we can obtain paths that look organized (bottom) when the confidence on the goal direction is high, or chaotic (top) when there is no information regarding the goal direction. Note that the core of the exploration method is based on DFS (but altering the order in which neighbours are visited), thus it keeps track of visited and unvisited cells, so that it will find a solution despite finding deadlocks such as the ones that appear on the right most map of the bottom row in Figure 8.

As a quantitative evaluation, we can show how by increasing the percentage of agent's knowledge, the total path length decreases. Figure 9 shows the comparison of total path length of our method combining A* with GPS heuristic and naive exploration, against A*. The image shows how the path length of our algorithm decreases as the familiarity with the environment increases. The paths provided by our method are almost as optimal as A* when $\mathcal{P} > 75\%$.

Figure 10 shows the average path ratios as the level of knowledge increases. We can observe how for no knowledge ($\mathcal{P} = 0\%$) the path length we obtain is up to $3.6x$ longer, which demonstrates that our naive exploration provides a solution that

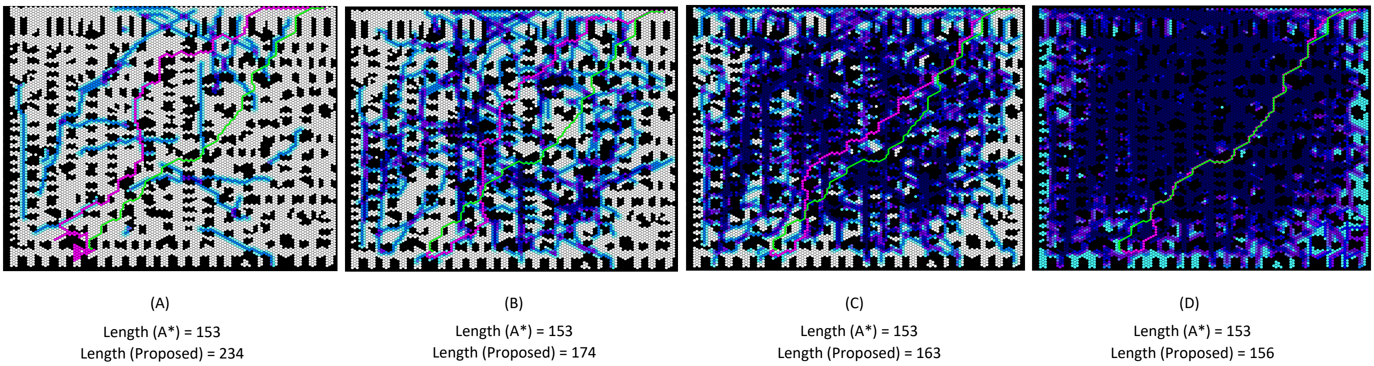

Fig. 5. Path length depending on percentage of agent's knowledge, $\mathcal{P}$ (in pink our method and in green the A* path). $\mathcal{P}$ = (A)25% , (B)50% , (C)75% and (D)100%. Blue intensity indicates level of knowledge based on how many times it has been visited before.

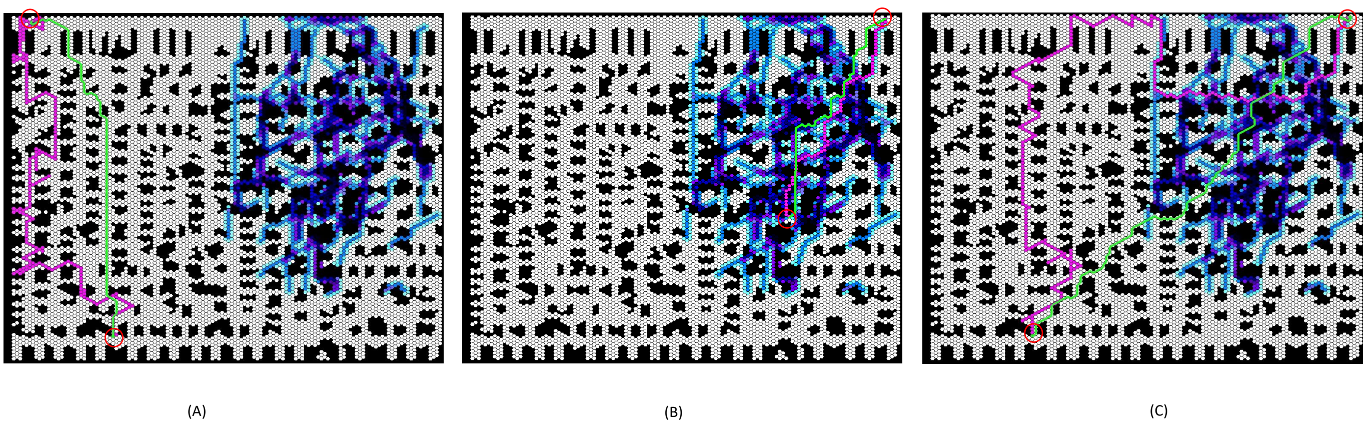

Fig. 6. Illustration of our method (in pink) and A* path (in green). Blue intensities represent agent's level of knowledge. (A) path planing between two points in an unknown area, (B) Path between two points in a known area and (C) path from an unknown area to a location in a known area.

is longer than the optimal solution but not too far off from it. When the knowledge is up to 75% the ratio is $1.07x$ which indicates that is very close in length to the optimal solution (ratio 1).

When an agent is not familiar with the environment, it will only run the naive search algorithm while looking for paths between random S and G positions. Table 1 shows the average number of searches needed to acquire knowledge equivalent to $25\%, 50\%, 75\% and 100\%$ for each of the two scenarios used in this paper. Scenario 1 appears in figures 5, 6, 7, and 11 (top), and scenario 2 appears in figures 8 and 11 (bottom).

Initially our algorithm will only run naive search, and then as knowledge increases it will combine naive search with the modified A* based on the GPS of the brain heuristics. naive search requires less computation and thus can be calculated very efficiently, whereas the modified A* search with our new heuristic takes longer to compute. So, as the number of known cells increases, the length of the path decreases, but the performance increases due to a larger number of searches in the known areas. Performance depends on many factors, such as the size of the environment or the distance between goal and search cells. Table 1 shows a performance analysis of average times taken on searches in the two scenarios that appear in this paper as the knowledge level increases. T1 represents the time taken by the naive search algorithm, T2 the time taken by the modified GPS-A* and T (total) de sum of both (all times shown in miliseconds).

Our algorithm combines two types of searches: (1) naive exploration and (2) a path finder with a heuristic based the GPS of the brain theory, thus on the counters in the mental map. The former (1) can exhibit different levels of confidence on the direction of the goal (higher values of δ and smaller values of $\mathcal{S}$ can move the agent quicker towards the goal), and the latter (2) will get closer to A* as $\mathcal{P}$ increases. Having such a variety of paths based on knowledge, could also be interesting to exhibit more heterogeneity of paths in games. Currently, game engines apply the same pathfinder over the navigation mesh for all agents, which leads to unrealistic simulations when having a large number of agents. Our work could benefit crowd simulation and multi-agent path finding by providing a variety of paths for the same scenario.

4.2. User study

Even though simulating human behavior is a huge challenge, and there is still a lot of work to be done, we have presented a method that attempts to imitate more closely how humans

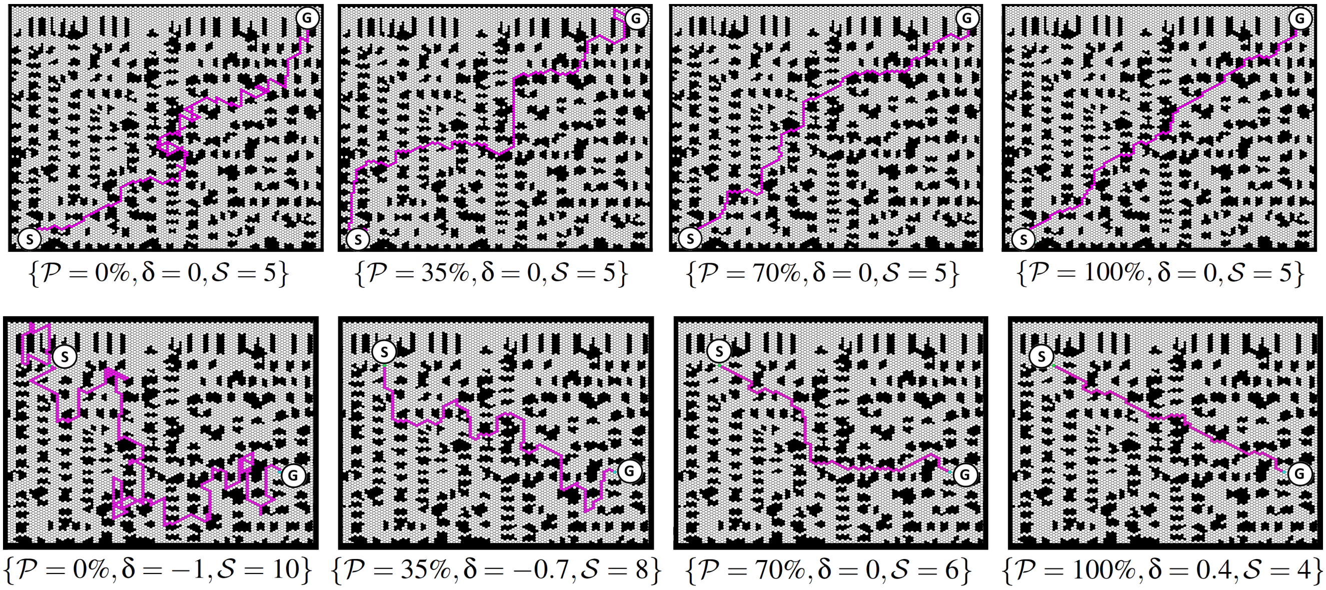

Fig. 7. Two example maps, each with 4 paths corresponding to increasing levels of knowledge (from left to right). The top map has a fixed δ and $\mathcal{S}$, while $\mathcal{P}$ increases from left to right. The bottom map modifies all three parameters to exhibit knowledge not only in terms of known cells, but also in terms of confidence on the goal direction.

	Scenario 1 (13,236 cells)				Scenario 2 (9,028 cells)			
$\mathcal{P}$	avg. # searches	avg. T1 Unknown naive	avg. T2 Known GPS-A*	Total (ms) T1+T2	avg. # searches	avg. T1 Unknown naive	avg. T2 Known GPS-A*	Total (ms) T1+T2
0%	0	2.01	0	2.01	0	1.77	0	1.77
25%	36	1.87	0.34	2.21	22	1.08	0.73	1.81
50%	61	0.97	1.36	2.33	37	0.41	1.61	2.02
75%	102	0.32	2.48	2.80	72	0.18	2.10	2.28
100%	141	0	3.11	3.11	127	0	2.63	2.63

Table 1. Average number of searches needed to acquire different levels of knowledge about the environment for the two scenarios used in the paper. For each scenario and level of knowledge, $\mathcal{P}$, we also show how the search time (in milliseconds) is distributed between the naive search (T1) and the A* with the modified GPS heuristics (T2).

find paths in the real world. The goal of our work was to have autonomous agents exhibiting a wider variety of path searches consistent with their level of knowledge. Therefore, besides evaluating path length as knowledge increases, it is also important to study whether the resulting paths would be perceived by the viewer as being more or less knowledgeable. We have run two perceptual studies to test whether participants can correctly identify the intended level of knowledge of our agents based on their paths.

For the first user study, we set a fixed δ and $\mathcal{S}$, while increasing $\mathcal{P}$. For the second study, we use varying values for the three parameters. We had a total of 40 participants, 20 doing each study. Each study was done as a within-subjects experiment, and consisted of 2 environments, 4 configurations of start and goal positions per map, and 4 configurations of agents' knowledge. In this study each user would see a total of 32 maps with a path drawn from the start position S to the goal position G, and was asked to determine the perceived level of knowledge about the environment by an agent following such trajectory. The values that the participant had to assigned were: 0 meaning "very little", 1 "a bit", 2 "quite well" or 3 "extremely well". Figure 11 shows the 2 environments used in the study, with 4 different paths corresponding to different levels of agent knowledge. For each map we gathered 160 responses (20 participants x 2 environments x 4 configurations of S and G) which can be seen in the 4 bars above each map in figure 12. Results were analyzed running a Chi-square test.

Experiment 1 had 4 configurations: map 0:$\{\mathcal{P} = 0\%\}$, map 1:$\{\mathcal{P} = 35\%\}$, map 2:$\{\mathcal{P} = 75\%\}$, map 3:$\{\mathcal{P} = 100\%\}$, all four maps with $\delta = -0.5$ and $\mathcal{S} = 5$. Participants saw a total of 32 paths, and were asked to look at the path and rank the agent's knowledge, $\mathcal{K}$, as: 0 meaning "very little", 1 "a bit", 2 "quite well" or 3 "extremely well". As we show in the top graph in figure 12, participants ranked map 0 with mostly $\mathcal{K} = \{0, 1\}$, map 1 with $\mathcal{K} = \{1, 2\}$, map 2 with $\mathcal{K} = \{2, 3\}$, and map 3 with mostly $\mathcal{K} = \{3\}$. We ran a χ^2 and obtained a *p–value* ≈ 0 (4.23×10^{-156}) indicating that there is a statistically significant relationship between the map configuration and the user's per-

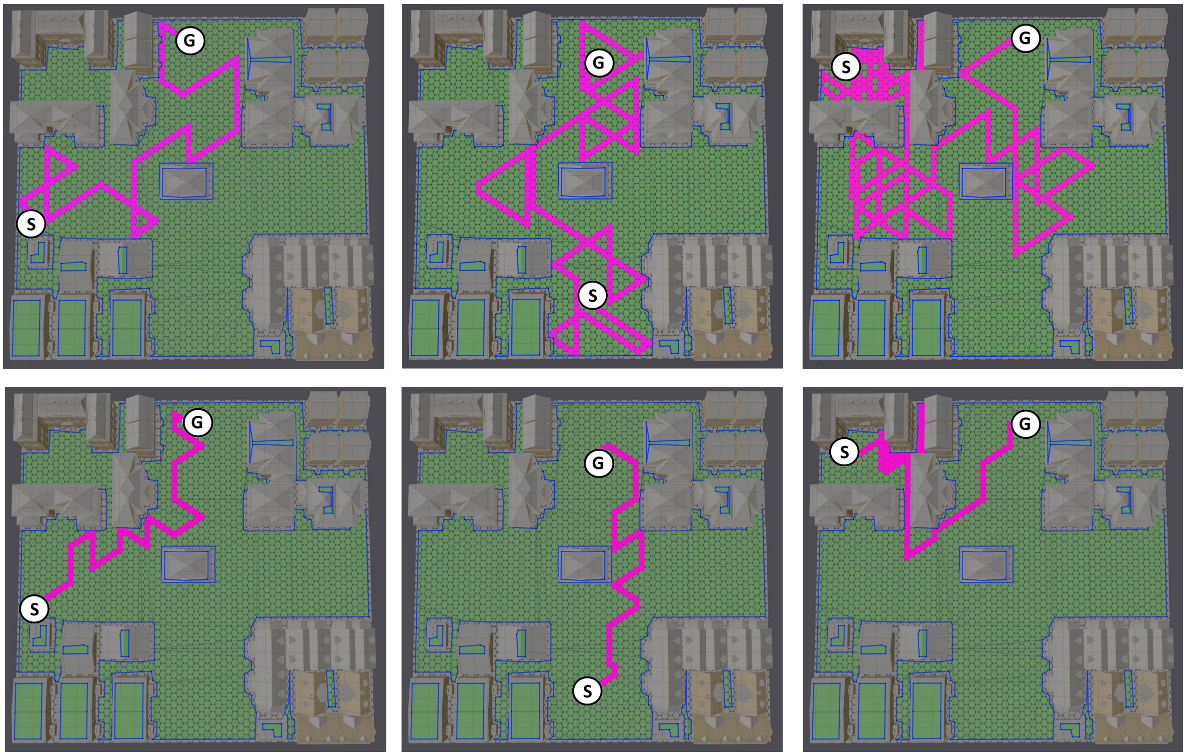

Fig. 8. Example map with green cells representing empty space, and grey representing obstacles. All paths correspond to $\mathcal{P} = 0$, meaning that there is only exploration (no A* with GPS heuristic). The top row shows paths with $\delta = -1$ and $\mathcal{S} = 10$, meaning that the agent chooses random directions and then follows the decision for 10 steps, as it has no confidence on the goal direction. The bottom row shows paths with $\delta = 0.4$ and $\mathcal{S} = 5$ which steers the agent quite confidently towards the goal.

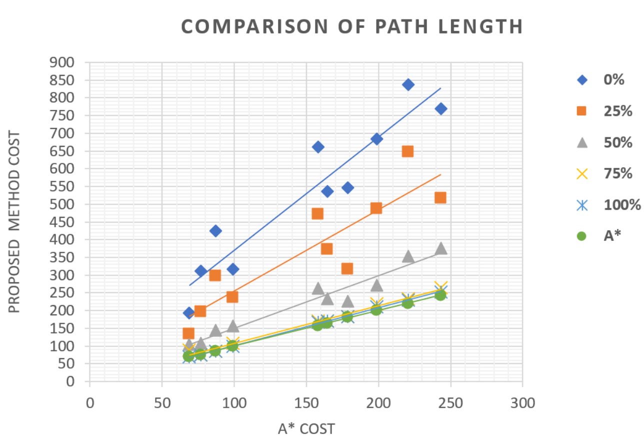

Fig. 9. Comparison of path length for increasing levels of $\mathcal{P}$

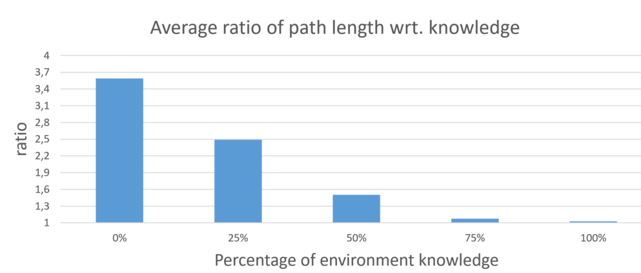

Fig. 10. Ratios of average path length as $\mathcal{P}$ increases $\{0\%, 25\%, 50\%, 75\%, 100\%\}$, $\delta = 0$ and $\mathcal{S} = 10$, with respect to the A* path length (using the map that appears in Figure ??).

ceived level of knowledge. This means that participants either guessed correctly the level of knowledge for each map, or else they slightly overestimated it. The reason for this, is that the naive search made the agents move quite well towards the goal direction.

We then run experiment 2, trying to assign δ and $\mathcal{S}$ with values that better matched level of confidence with levels of knowledge. Therefore, we had the following map configurations: map 0: $\{\mathcal{P} = 0\%, \delta = -1, \mathcal{S} = 10\}$, map 1: $\{\mathcal{P} = 35\%, \delta = -0.8, \mathcal{S} = 8\}$, map 2: $\{\mathcal{P} = 70\%, \delta = -0.6, \mathcal{S} = 6\}$, map 3: $\{\mathcal{P} = 100\%, \delta = -0.4, \mathcal{S} = 4\}$. The χ^2 test gave us a *p–value* ≈ 0 (6.26×10^{-173}) indicating again that there is a statistically significant relationship between the map configuration and the user's perceived level of knowledge. The bottom graph of Figure 12, show that for the second experiment, users perceived the resulting paths as being closer to our intended configuration, therefore each map level got the highest number of answers matching the corresponding knowledge level intended for each map. The Pearson rank correlation between the map knowledge and the user's perceived agent knowledge was $r_s = 0.86$, indicating a strong relationship between them.

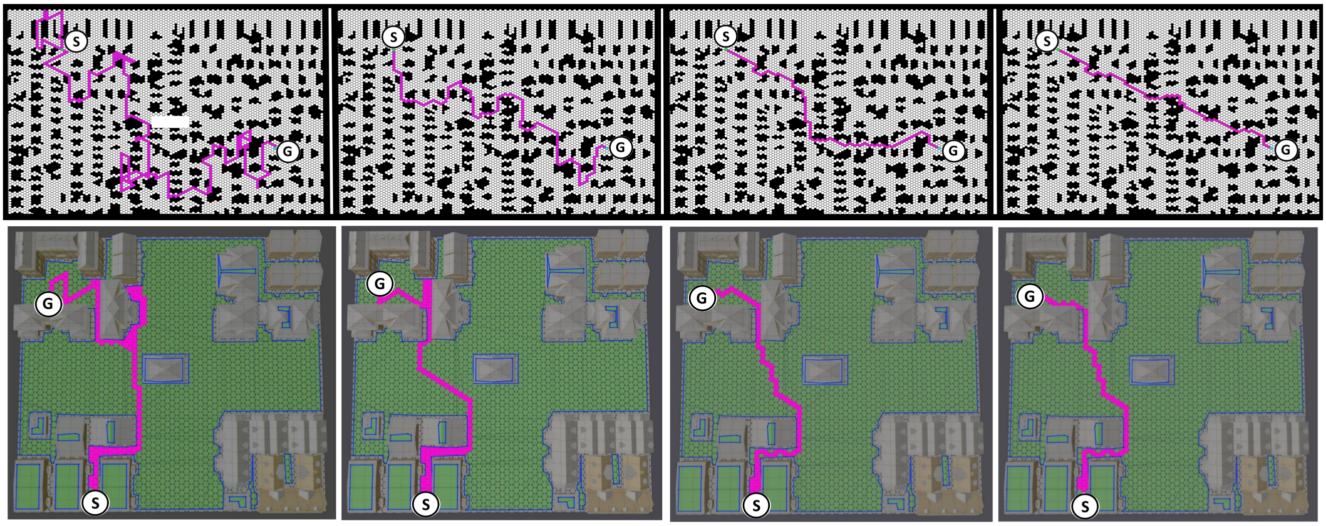

Fig. 11. Two example maps used for the user study, each with 4 paths corresponding to increasing levels of knowledge. The specific values for each path finding are those described for user study 2.

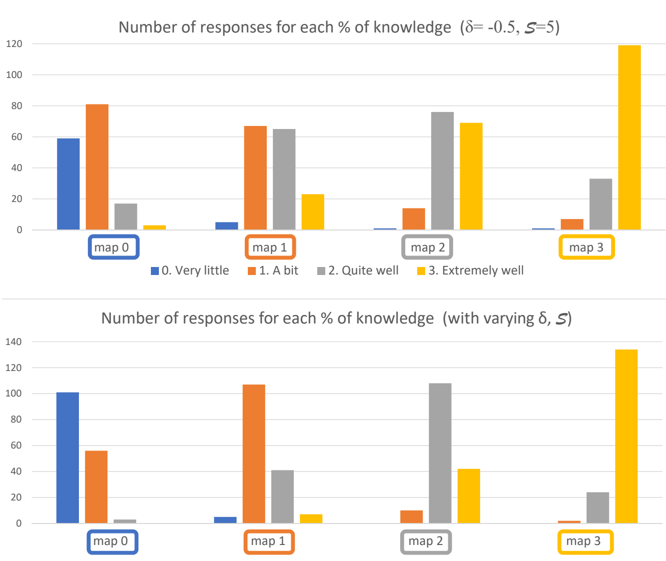

Fig. 12. Perceptual Evaluation. The top graph shows the perceived level of familiarity for maps of increasing $\mathcal{P}$, with $\delta = 0.5$ and $\mathcal{S} = 5$. The bottom shows also maps of increasing $\mathcal{P}$, but varying δ and $\mathcal{S}$ to also exhibit increasing levels of confidence on the goal direction.

5. Discussion and conclusion

Even though simulating human behavior is a huge challenge, and there is still a lot of work to be done, we have presented a method that attempts to imitate more closely how humans find paths in the real world. In this paper we have proposed a path finding method that attempts to resemble the human's brain navigation system to simulate more human-like autonomous agents. We also propose a more human-like exploration method for unknown environments with vague knowledge of goal direction. We believe that this is the first attempt towards simulating more human-like path finding.

Our method can work with known, unknown and mixed environments. The hexagonal grid navigation mesh mimics the humans' brain grid cell. Cell counters simulate the way our brain keeps track of visited places as agent's memory. The proposed naive exploration uses a variation of the Depth First Search (DFS) algorithm to consider vague information of the environment (rough knowledge of goal direction), and builds a cognitive map for the agent as it wanders the environment. Path finding in known environment, is carried out by applying a modified heuristic to A*. The new heuristic considers the cognitive map counters as the agents' memory.

Even though the alternating nature between exploration and GPS path finding may resemble reinforcement learning, please note that RL uses exploration to learn a policy, which changes actions for each state based on a reward function. Our method differs from this, because exploration is used to gather knowledge about the environment, which then improves the pathfinder by being more informed and thus having a more accurate heuristic.

Our experimental results show that path length for the proposed method converges towards the traditional A* search as the agent acquires more knowledge of the environment.

As future work we would also like to consider memory decay and other aspects of human perception that may affect the way we remember places (for example based on their saliency or uniqueness). The concept of line of sight could also be used to include memory of landmarks that can be seen from a distance, even if the agent does not physically walk by them.

Acknowledgments

This work was partly funded by the Spanish Ministry of Economy, Industry and Competitiveness under Grant No. TIN2017-88515-C2-1-R.

References

[1] Kallmann, M, Kapadia, M. Geometric and Discrete Path Planning for Interactive Virtual Worlds. Morgan and Claypool; 2016.

[2] Galceran, E, Carreras, M. A survey on coverage path planning for robotics. Robotics and Autonomous systems 2013;61(12):1258–1276.

[3] Reitter, D, Lebiere, C. A cognitive model of spatial path-planning. Computational and Mathematical Organization Theory 2010;16(3):220–245.

[4] Sohn, SS, DeStefani, S, Kapadia, M. Dynamic cognitive maps for agent landmark navigation in unseen environments. In: Proceedings of the 11th Annual International Conference on Motion, Interaction, and Games. 2018, p. 1–6.

[5] Hafting, T, Fyhn, M, Molden, S, Moser, MB, Moser, EI. Microstructure of a spatial map in the entorhinal cortex. Nature 2005;436(7052):801.

[6] Doeller, CF, Barry, C, Burgess, N. Evidence for grid cells in a human memory network. Nature 2010;463(7281):657–661.

[7] Dalton, RC. The secret is to follow your nose: Route path selection and angularity. Environment and Behavior 2003;35(1):107–131.

[8] Pelechano, N, Allbeck, JM, Badler, NI. Virtual crowds: Methods, simulation, and control. Synthesis lectures on computer graphics and animation 2008;3(1):1–176.

[9] O'Keefe, J, Dostrovsky, J. The hippocampus as a spatial map: preliminary evidence from unit activity in the freely-moving rat. Brain research 1971;.

[10] Franz, MO, Mallot, HA. Biomimetic robot navigation. Robotics and autonomous Systems 2000;30(1-2):133–153.

[11] Milford, M, Schulz, R. Principles of goal-directed spatial robot navigation in biomimetic models. Philosophical Transactions of the Royal Society B: Biological Sciences 2014;369(1655):20130484.

[12] Savkin, AV, Wang, C. Seeking a path through the crowd: Robot navigation in unknown dynamic environments with moving obstacles based on an integrated environment representation. Robotics and Autonomous Systems 2014;62(10):1568–1580.

[13] Khansari-Zadeh, SM, Khatib, O. Learning potential functions from human demonstrations with encapsulated dynamic and compliant behaviors. Autonomous Robots 2017;41(1):45–69.

[14] Kapadia, M, Ninomiya, K, Shoulson, A, Garcia, F, Badler, N. Constraint-aware navigation in dynamic environments. In: Proceedings of Motion on Games. ACM; 2013, p. 111–120.

[15] Elbanhawi, M, Simic, M. Sampling-based robot motion planning: A review. Ieee access 2014;2:56–77.

[16] Hart, PE, Nilsson, NJ, Raphael, B. A formal basis for the heuristic determination of minimum cost paths. IEEE transactions on Systems Science and Cybernetics 1968;4(2):100–107.

[17] Koenig, S, Likhachev, M. Dˆ* lite. Aaai/iaai 2002;15.

[18] Yap, PKY, Burch, N, Holte, RC, Schaeffer, J. Any-angle path planning for computer games. In: Seventh Artificial Intelligence and Interactive Digital Entertainment Conference. 2011,.

[19] Nash, A, Koenig, S, Likhachev, M. Incremental phi*: Incremental any-angle path planning on grids. In: Twenty-First International Joint Conference on Artificial Intelligence. 2009,.

[20] Ferguson, D, Stentz, A. Field d*: An interpolation-based path planner and replanner. In: Robotics research. Springer; 2007, p. 239–253.

[21] Botea, A, Müller, M, Schaeffer, J. Near optimal hierarchical pathfinding. Journal of game development 2004;1(1):7–28.

[22] Rahmani, V, Pelechano, N. Multi-agent parallel hierarchical path finding in navigation meshes (ma-hna*). Computers & Graphics 2020;86:1–14.

[23] Schulman, J, Duan, Y, Ho, J, Lee, A, Awwal, I, Bradlow, H, et al. Motion planning with sequential convex optimization and convex collision checking. The International Journal of Robotics Research 2014;33(9):1251–1270.

[24] Stentz, A. Optimal and efficient path planning for partially known environments. In: Intelligent unmanned ground vehicles. Springer; 1997, p. 203–220.

[25] Kapadia, M, Beacco, A, Garcia, F, Reddy, V, Pelechano, N, Badler, NI. Multi-domain real-time planning in dynamic environments. In: Proceedings of the 12th ACM SIGGRAPH/Eurographics symposium on computer animation. 2013, p. 115–124.

[26] Jiménez, AC, García-Díaz, V, González-Crespo, R, Bolaños, S. Decentralized online simultaneous localization and mapping for multi-agent systems. Sensors 2018;18(8):2612.

[27] López, E, García, S, Barea, R, Bergasa, LM, Molinos, EJ, Arroyo, R, et al. A multi-sensorial simultaneous localization and mapping (slam) system for low-cost micro aerial vehicles in gps-denied environments. Sensors 2017;17(4):802.

[28] Alismail, H, Baker, LD, Browning, B. Continuous trajectory estimation for 3d slam from actuated lidar. In: 2014 IEEE International Conference on Robotics and Automation (ICRA). IEEE; 2014, p. 6096–6101.

[29] Li, J, Zhong, R, Hu, Q, Ai, M. Feature-based laser scan matching and its application for indoor mapping. Sensors 2016;16(8):1265.

[30] Hillier, B. Moving diagonally: Some results and some conjectures. London: University College 1997;.

[31] Sadalla, EK, Montello, DR. Remembering changes in direction. Environment and Behavior 1989;21(3):346–363.

[32] Montello, DR. Spatial orientation and the angularity of urban routes: A field study. Environment and Behavior 1991;23(1):47–69.

[33] Gärling, T, Gärling, E. Distance minimization in downtown pedestrian shopping. Environment and Planning A 1988;20(4):547–554.

[34] Zacharias, J. Pedestrian behavior pedestrian behavior and perception in urban walking environments. Journal of Planning Literature 2001;16(1):3–18.

[35] Penn, A, Turner, A. Space syntax based agent simulation. Springer-Verlag; 2002,.

[36] Warren, WH. Non-euclidean navigation. Journal of Experimental Biology 2019;222(Suppl_1):jeb187971.

[37] Ericson, JD, Warren, WH. Probing the invariant structure of spatial knowledge: Support for the cognitive graph hypothesis. Cognition 2020;200:104276.

[38] Hirtle, SC, Jonides, J. Evidence of hierarchies in cognitive maps. Memory & cognition 1985;13(3):208–217.

[39] Thomas, R, Donikian, S. A model of hierarchical cognitive map and human-like memory designed for reactive and planned navigation. In: Proceedings of the 4th International Space Syntax Symposium. 2003, p. 1–18.

[40] Zhong, C, Liu, S, Lu, Q, Zhang, B, Yang, SX. An efficient fine-to-coarse wayfinding strategy for robot navigation in regionalized environments. IEEE transactions on cybernetics 2016;46(12):3157–3170.

[41] Saian, PON, et al. Optimized a-star algorithm in hexagon-based environment using parallel bidirectional search. In: 2016 8th International Conference on Information Technology and Electrical Engineering (ICITEE). IEEE; 2016, p. 1–5.

[42] Gibson, JJ. The ecological approach to visual perception: classic edition. Psychology Press; 2014.