A Cognitive Model for Routing in Agent-based Modelling

Master Thesis

Author(s):
Grübel, Jascha

Publication date:
2018-02

Permanent link:
https://doi.org/10.3929/ethz-b-000291815

Jascha Grübel

A Cognitive Model for Routing in Agent-based Modelling

Master's Thesis

JASCHA GRÜBEL

A COGNITIVE MODEL FOR ROUTING IN AGENT-BASED MODELLING

To Tanja

ABSTRACT

Routing is an essential process for pedestrian Agent-Based Modelling (ABM). ABM is a computational tool to model and analyse human behaviour. The process of routing is well-studied in both Computer Science and Cognitive Science. However, routing in ABM is often taken for granted and both its impact and its implementation are disregarded. In this work, I unpack the blackbox of routing in ABM and take insights from Cognitive Science to improve the realism of routing.

In particular, I focus on the agent's mental representation of the environment and typical errors in encoding this information. I propose to deviate from classical Computer Science paradigm of optimality to capture human behaviour more accurately. The resulting model produces routes that are less prone to typical computational artefacts such as *ziggzagging*, i. e. turning more often than humans would, and *bottlenecks*, i. e. always routing through one particular node because it is minimally more efficient.

ZUSAMMENFASSUNG

Routing (Wegfindung) ist ein grundlegender Prozess für agentenbasierte Modellierung (AM) von Fussgängern. AM ist ein Werkzeug für wissenschaftliches Rechnen, um menschliches Verhalten zu modellieren und zu analysieren. Der Prozess des Routing ist gut erforscht, sowohl in der Informatik als auch in den Kognitivwissenschaften. Dennoch wird Routing in AM für selbstverständlich erachtet und daher bleiben sowohl der Einfluss als auch die Umsetzung oft unberücksichtigt. In dieser Arbeit öffne ich die Blackbox Routing in AM und verwende Erkenntnisse aus den Kognitivwissenschaften, um den Realismus von Routing zu verbessern.

Insbesondere betrachte ich dabei die mentale Repräsentation der Umgebung und typische Fehler, die beim Abrufen dieser Informationen entstehen. Ich schlage vor, von dem klassischen Informatik-Paradigma der Optimalität abzuweichen, um menschliches Verhalten zutreffender zu erfassen. Das resultierende Modell produziert Routen, die weniger anfällig sind, Berechnungs-Artefakte wie *Zickzack-Muster* und *Nadelöhre* aufzuweisen. Zickzack-Muster stellen mehr Abbiegungen in Routen dar, als Menschen nehmen würden, und Nadelöhre stellen Routen dar, die gezwungenermassen durch einen bestimmten Knoten gehen, da dieser unerheblich mehr effizient ist.

ACKNOWLEDGEMENTS

I would like to thank Dirk Helbing and Christoph Hölscher for enabling me to work on this interdisciplinary topic and write this thesis. My special thanks also extend to their postdoctoral researchers Tyler Thrash and Leonel Aguilar for the excellent supervision and the invaluable feedback for both aspects of Computer Science and aspects of Cognitive Science. Our discussions helped me to formulate my hypothesis, formalise the theoretical framework, and stopped me from straying to far into interesting but unrelated topics.

I would also like to thank Sarah Wise and Chris Rouly for introducing me to Agent-Based Modelling and its curiosities, pitfalls, and potential. Discussions with you helped me to define the topic of this thesis and eventually propose it. Without your input I would not have written this.

With deep gratitude I thank Tanja, without her I would not be where I am today. She always encourages me to continue on my path and it is easier to walk our path together. She also helped proof-reading and without her input on structure this work would be less coherent.

Lastly, but surely not least, I would like to repeat how grateful I am to my parents Elvira and Felix and all my grandparents who gave me the possibility to study to my heart's content and pursue one degree after another. We are at three now and still counting.

CONTENTS

NOTATION

FREQUENTLY USED ABBREVIATIONS

ABM$^\diamond$	Agent-Based Modelling
GIS$^\diamond$	Geographical Information Systems
CSS†	Computational Social Science
KDE*	Kernel Density Estimation
JRD$^\circ$	Judgment of Relative Direction

($\diamond$ Castle & Crooks [1])
($\dagger$ Cioffi-Revilla [2])
($\star$ Duong *et al.* [3])
($\circ$ Klatzky *et al.* [4])

1

INTRODUCTION

Millions saw the apple fall, but Newton asked why.

— Bernard Baruch

Modern Agent-based Modelling (ABM) traces its roots back to cellular automata[5] and the Game of Life[6]. From this mathematical backdrop it found early application in social sciences such as in the segregation model[7,8]. Today, ABM can be seen from two perspectives, Computer Science and Social Sciences, see Fig. 1.1, and are often considered to be part of the interdisciplinary Computational Social Science field[2]. They are also considered to be a complex system encompassing "heterogeneous subsystems or autonomous entities, which often feature non-linear relationships and multiple interactions[9]." ABM is considered a tool to study bottom-up phenomena and to understand the complexity associated with the interaction of bottom-up processes[10,11].

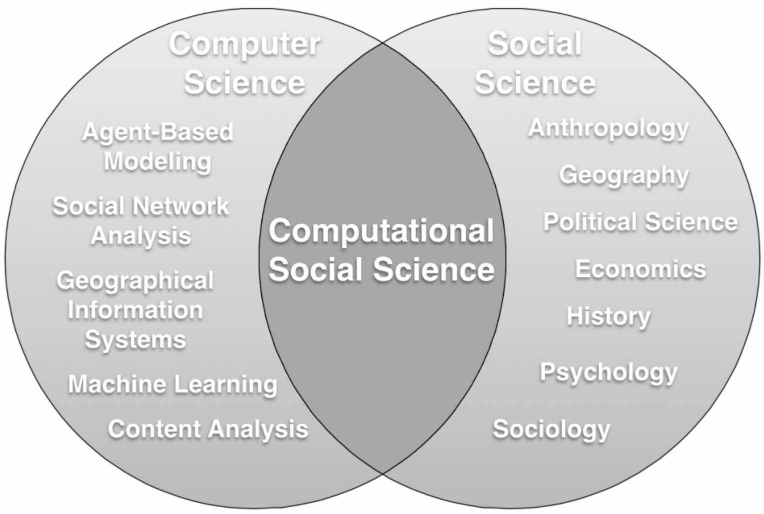

Figure 1.1: **Computation Social Science** – An interdisciplinary intersection between Computer Science and Social Sciences. Source: `https://cos.gmu.edu/cds/computational-social-science/`

ABM also exhibits useful properties when assessing theoretical models due to its inherent internal validity[12,13]. This is especially useful when a theoretical framework has been established and matches some observed behaviour in the real world but the question remains whether it is a correct model or some other unaccounted factor would better explain the observed[14,15]. ABM can put a theoretical framework to the test and run according to its specification and *only* its specification[13]. When simulation results match collected data, then internal validity of the model can be established.

Since the advent of Geographic Information Systems (GIS), the application of ABMs on human navigation have become a common interest[16,17]. ABMs are used to analyse human navigation on a multitude of scales from large regions[18,19], cities[20,21] over small neighbourhoods[22] and to buildings[23]. Previous ABMs have addressed important aspects of realistically modelling human navigation behaviour on a strategic, tactical, and operational level[24–34], see Fig. 1.2. This is known as an instance of the *means-end problem-solving* approach[35–37] and has been suggested as a conceptualisation for subdivision of tasks in ABM[38].

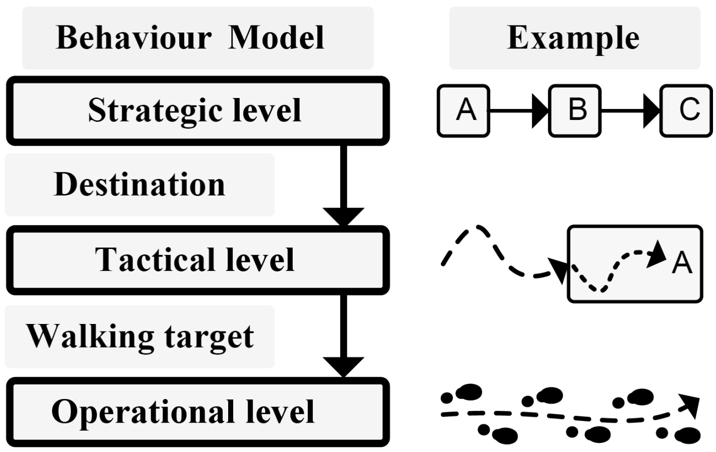

FIGURE 1.2: **Pedestrian Behavioural Model** – The subdivision of the behavioural model into levels of action labelled strategic, tactical and operational. Additionally, the mapping of actions to a symbolic representation of the execution is shown. On a strategic level, the destinations are chosen. On a tactical level, intermediate waypoints are chosen. On an operational level, movement is performed. The figure has been adapted from Kielar & Borrmann [39] and is printed with the authors' permission. The figure has been simplified and relabelled for this work.

On the strategic level, agents make decisions on what kind of locations they wish to visit and why. Strategic level modelling is often the domain of Spatial Cognition where Landmark Knowledge and Survey Knowledge are explored, defined and analysed[40–45]. Landmark knowledges covers distinct objects or scenes recognition[43], whereas Survey Knowledge is linked to the configuration of space[43]. On the tactical level, agents may choose modes of transportations and particular types of behaviour, such as crowd evasion[46]. This is also the domain where most often routing occurs[47], however, it is not limited to this level. On the operational level, agents make locomotion decisions[47] and types of behaviour like queuing and collision evasion are performed[48]. A common operational model to simulate interaction between the agent, the environment and other agents is the Social Forces model[32,49,50]. Other research in the operational level focuses on phenomena like local pushing in crowded situtations[51].

However, there is one component on the tactical level neglected in most models despite being a core component of any pedestrian ABM simulation—routing. A route is an abstract representation of space along which an agent moves from a start location to a goal location by the means of visiting a sequence of intermediate locations[52]. From a cognitive perspective, a route encodes knowledge of a sequence of decisions, which are triggered by the perception of landmarks[53], in a particular location[52].

This component is often taken for granted and is not further explored or explained both in research[54–57] and introductory literature[58,59]. For many models it is only stated which algorithm is used, if they explain the underlying assumptions of their routing at all. More often routing is assumed to be obvious and trivial. Sometimes, routing is discussed when the algorithm is optimised for large scale, but it still follows the same classical paradigms[18].

On the computational side, for most models it is merely stated that they use Dijkstra for routing. However, when inspecting the code, this is often inaccurate. In reality, most models use the A*-algorithm in Dijkstra's stead, to increase performance. This inaccuracy may be explained by people using a framework where routing is already implemented. Most frameworks note that their routing results are equivalent to Dijkstra's routing algorithm and hence contribute to the confusion on the user side. Dijkastra's work is taken as the baseline and features most prominently in Theoretical Computer Science when discussing routing properties[60–63]. Nonetheless, this behaviour by scientists reduces the routing to a black box that outputs a path to be taken by agents without fully understanding why or how.

On the modelling side, routing is usually delegated to middle layers—i. e. tactical level—of functioning within agents[64]. Based on the conceptual division into strategy, tactics and operations[64], routing could be understood in different distinct layers. On a strategic level, routing consists of choosing a destination, but not how to get there.

On a tactical level, routing consists of find connecting intermediate steps to reach the goal. On an operational level, routing consists of an agent's locomotion system to move towards the next intermediate goal, see Fig. 1.2.

While this delegation allows for a more effective implementation of higher level reasoning such strategic routing or lower-level locomotion such operational routing it also separates intertwined cognitive processes of understanding the environment and making decisions about routing through it[65,66]. In particular, the algorithmic solutions to the shortest path problem usually assume either perfect knowledge or no knowledge at all which is not the case in human navigation. This distinction has introduced a bias when resolving a route. This lower-level task of computing the route is either optimally solved or purely exploratory. In contrast, we would expect the agents to exhibit similar cognitive mistakes that humans make.

On the cognitive side, multiple aspects of routing warrant closer attention and indeed have been widely studied outside the ABM context. Interesting topics include uncertainty in wayfinding, perception and environmental cues, learning of spatial knowledge, and memory retrieval.

Uncertainty in wayfinding is an aspect of routing that has been studied before[33,67]. This branch of research focusses more on uncertainty in the agents understanding of a route description, i. e. whether the route was appropriately communicated to the agent. The concepts are hence implemented on a strategic level and not on the tactical level. It influences the general planning behaviour but ultimately still relies on perfect routing when choosing the path.

Using the agent's perception to make routing more human-like is the approach of the Unified Pedestrian Routing Model (UPRM)[68]. This model addresses group-behaviour in routing and routing choice based on cognitive principles, such as small angles between route segments[69–71] and beeline distance[70–72]. It influences the agents' behaviour on an operational level and may force agents to reroute on a tactical level based on the decisions.

There is a branch of modelling cognitive processes that tries to capture the learning process that transforms Landmark Knowledge into Survey Knowledge[73]. The process influences all levels of the pedestrian behaviour model. However, the process of acquiring the mental representation is not the goal of this study and is not further discussed.

Mental representation and its errors are well-studied in Cognitive Science, but yet have to make an impact on ABM. In this work, I will explore both computational and cognitive aspects of routing in ABM. My contribution is to show how important it is to account for errors in mental representation in navigation. It is not sufficient to solve the routing problem, good ABM also needs to account for the cognitive processes that (mis)guide real human behaviour.

The routing model presented in this paper transforms the perfect knowledge of the graph into an agent's mental representation with appropriate errors. These mental representations are based on findings in Cognitive Science that allow us to simulate the expected error that humans make when encoding the real world into their mental representations. The resulting model reduces the accuracy of agents to a degree observed in humans, while maintaining their ability to route through the environment.

The core of my thesis is split into three parts—background, methods, and results—and closed by a conclusion. In the background Ch. 2, I give an overview over the possible ways of solving the shortest path algorithm from a computational perspective. With a clear idea of possible technical solutions, I explore the Social Sciences side by using Cognitive Science, in particular Spatial Cognition, to justify changes to the routing model from a mental representation perspective. This brings us more closely to human-like behaviour in agents, a declared goal of ABM[58]. Human-like behaviour is a vague term that entails not only finding a technical solutions but also to consider humans' possible action space and solving the problem accordingly.

In the methods Ch. 3, I present both algorithm for routing and the error modelling from cognitive perspective followed by a discussion of the implementation.

In the results Ch. 4, I show visual and statistical evidence for the difference in the routing algorithms and the mental representation error.

2

BACKGROUND

Computer science is no more about computers than astronomy is about telescopes.

— Edsger Dijkstra

Due to the interdisciplinary work of this thesis, the background is split in two sections. The main work will draw from both strains of literature, but they are split for the sake of consistently presenting them. I will first present the technical background of solving routing and then will elaborate how routing can be understood in the context of Cognitive Science.

2.1 ROUTING IN COMPUTER SCIENCE

Finding an optimal route between two points A and B is a well-known problem in Computer Science and coined as the *Shortest-Path Problem*. The first efficient solution and today's usual point of departure for the problem has been postulated by Dijkstra [74]. Albeit the publication took three years [1] and there are preceding publications that respond to his initial drafts widely circulated by his contemporaries. These graph theoretical formulations abstracts the environment into nodes (decision points) and edges (distances) in order to compute the shortest path. Whereas Dijkstra's work was revolutionary at the time it has since been superseded by more efficient formulations. Nonetheless, the underlying idea has remained the same and the additional sophistication of newer algorithm has lead to a common-held belief that all routing algorithms are *"doing Dijkstra"*. This unfortunate notion can partially be blamed for the lack of description of routing in most ABM publications as often the topic is assumed to be "completed".

To overcome this issue, I will first explore the common routing algorithm families and their approach to improving upon Dijkstra's initial suggestion. The main improvement is grouped together as Heuristic Incremental Search algorithms[75] and has been studied

1 Turing Award Ceremony Transcript: `https://amturing.acm.org/award_winners/dijkstra_1053701`

since shortly after Dijkstra's initial publication. ABM applications seem to limit themselves to this family of algorithms due to familiarity as well as problem specification, which I will explore in the following subsections.

2.1.1 *Shortest-Path Problem*

More formally, for a graph $G = (V, E)$ a path is defined as a sequence of vertices $P_{v_1,v_n} = (v_1, v_2, ..., v_n) \in V \times V \times ... \times V$ such that there is a connecting edge $e_{i,i+1} = (v_i, v_{i+1}) \in E$ for $1 \leq i \leq n$ between vertices along the sequence. The *shortest path* $P_{s,t}$ is defined between two vertices $s, t \in V$ such that under an edge weight function $d : E \rightarrow \mathbb{R}$ the sum of weights along the path is minimal, see Eq. 2.1. For a more detailed discussion, see Korte *et al.* [76, p. 151].

$$P_{s,t} \text{ s.t. } \underset{e_{i,i+1} \in P_{s,t}}{\arg\min} \sum_{i=1}^{n-1} d(e_{i,i+1}) \tag{2.1}$$

For pedestrian paths we can introduce the additional assumption that $d(e) \geq 0 \ \ \forall e \in E$, since paths of negative lengths are not possible. This additional assumption allows us to use algorithms for non-negative weights.

2.1.2 *Dijkstra's Algorithm*

The classical solution to the Shortest-Path Problem improved upon previous solution[77,78] by reducing complexity from $O(|V|^3)$ to $O(|V|^2)$[74] where $|V|$ is the number of vertices. Dijkstra's Algorithm starts at the start vertex s and iteratively adds new new shortest paths to all neighbouring vertices u. In its working memory the algorithm only keeps for each vertex v the last node on the path v' that leads up to it as well as the length of $P_{s,v}$. If a new path from u' leads to v such that the total length is shorter, the shortest path gets updated. The pseudo-code is shown in Alg. 2.1.

The algorithm's inefficiency lies in the data structure of Q which may need $O(|V|)$ time to find the next minimal distance. Fredman & Tarjan [79] showed that when using a Fibonacci Heap as Q, the complexity can be reduced to $O(|E| + |V|log|V|)$ where $|E|$ is the number of edges in G.

Algorithm 2.1 Dijkstra's Algorithm

```
Require: set Q contains all v; arrays distance[v] = ∞, previous[v]= NULL ∀v
Q.Add(s)
while ¬ Q.IsEmpty do
    u ← MinimalDistance(Q,s)                    ▷ Vertex with shortest distance to s
    Q.Remove(u)
    for all v in Neighbourhood(u) do
        altDist ← distance[u] + Length(u, v)
        if altDist ≤ distance[v] then
            distance[v]← altDist
            previous[v]← u
        end if
    end for
end while
ReconstructPath(t,previous)                     ▷ Returns P_{s,t}
```

2.1.3 *A* Algorithm*

The A* algorithm[80,81] is the first algorithm in the family of incremental heuristic search algorithms[60]. It improves upon Alg. 2.1 by adding a heuristic to guide the search for the goal. In particular, a cost function $c_t : V \rightarrow \mathbb{R}$ is introduced that approximates the remaining cost to reach the goal vertex t. In response, the cost function for the shortest-path problem is expanded to incorporate the heuristic, seen in Eq. 2.2.

$$P_{s,t} \text{ s.t. } \arg\min_{e_{i,i+1} \in E} \sum_{i=1}^{n-1} d(e_{i,i+1}) + c_t(v_{i+1}) \tag{2.2}$$

If the heuristic cost function c is consistent or monotone, i. e. it satisfies the inequality $c(v_i) \leq d(e_{i,i+1}) + c(v_{i+1})$, then A* does not need to revisit nodes to converge. A typical heuristic function is the *straight line distance*, *beeline distance*, or *Euclidean distance*.

A further optimisation compared to Alg. 2.1 is to use a priority queue as the data structure of Q. This allows to cheaply retrieve the shortest heuristic distance from Q. Hence A* is also a *best-first search* algorithm as it relies on "extra knowledge about the problem domain," Pearl [82, p. 48].

A* is structured very similarly to Dijkstra's Algorithm, compare Alg. 2.1 to Alg. 2.2. The main difference is that additionally the set of visited vertices Q_{closed} is maintained and that the simple distance function, see Eq. 2.1, is expanded with the heuristic function, see Eq. 2.2.

The heuristic, in contrast to Dijkstra's Algorithm allows us to stop early and remain optimal, once the goal is found. However, this only holds for monotone heuristic function. The caveat can be overcome by expanding Alg. 2.2 in line 13 to re-evaluate the heuristic function in light of the new score.

The algorithm has a worst-case complexity of $O\left(\overline{deg(v)}^{|P_{s,t}|}\right)$ where $\overline{deg(v)}$ is the average degree (i. e. number of edges connected to the vertex) and $|P_{s,t}|$ is the length of the shortest path[83]. However, with a good heuristic, the complexity can become near constant in the shortest path length $\lim_{\overline{n}\to 1} O\left(\overline{n}^{|P_{s,t}|}\right) = O(1)$ if only ever one outgoing edge would be selected by the heuristic. While this is clearly optimistic, it demonstrates how useful a good heuristic is and how damaging a bad heuristic can be.

A* is the *de facto* standard and is usually implemented, even if the paper only mentions Dijkstra. The similarity between Alg. 2.1 and Alg. 2.2 together with the optimality has lead many outside of Computer Science to take the topic as resolved.

2.1.4 *Other Algorithms*

The number of algorithms solving the shortest-path problem is large and I will limit the review to the most prominent algorithms which are in use outside of ABM.

A* VARIANTS The A* Algorithm provides a multitude of opportunities to be improved upon. These range from technical innovations such as Iterative Deepening (IDA*[84], Fringe Search[85]) and memory efficiency (MA*/SMA*[86]) over generalisations (RBFS[87]) to handling dynamically changing graphs (LPA*[88]). In common for all of those improvements is that they usually change a minor aspect for a specific problem formulation and require deep understanding of the inner workings of the algorithms to decide the trade-off. In light of these difficulties, it is understandable why they have not become popular in ABM.

Algorithm 2.2 A* Algorithm

Require: empty set Q_{open}, Q_{closed}; arrays score[v] $= \infty$,heuristic[v] $= \infty$, previous[v]= NULL $\forall v$

Q_{open}.ADD(s)
score[s] $\leftarrow 0$
heuristic[s] $\leftarrow$ HEURISTIC(s, t)
while $\neg$ Q_{open}.ISEMPTY **do**
    $u \leftarrow$ MINIMALHEURISTIC(Q_{open},t) ▷ Vertex with lowest heuristic to t
    if $u = t$ **then**
        RECONSTRUCTPATH(t,previous) ▷ Returns $P_{s,t}$
    end if
    Q_{open}.REMOVE(u)
    Q_{closed}.ADD(u)
    for all v in NEIGHBOURHOOD(u) **do**
        if $v \in Q_{closed}$ **then**
            CONTINUE ▷ Only for monotone heuristic functions
        end if
        if $v \notin Q_{open}$ **then**
            Q_{open}.ADD(u)
        end if
        $altScore \leftarrow$ score[u] + LENGTH(u, v)
        if $altDist \leq$ score[v] **then**
            score[v]$\leftarrow$ $altDist$
            heuristic[v]$\leftarrow$score[v]+ HEURISTIC(v, t)
            previous[v]$\leftarrow$ u
        end if
    end for
end while

D* AND VARIANTS The D* Algorithm family is derived from the A* algorithm family, but does not assume knowledge about the complete graph and obtains data through sensory input. It comes from the branch of robotics, where it is successfully used to map the environment for robot navigation on the fly[89–91]. The algorithms may appear to solve a problem that is too complex as the environment is well-known in most ABM specifications. Nonetheless, a robot can be considered an agent and the ability to move on unknown terrain certainly allows for a more flexible routing model. The complexity of implementing D* and its variants together with the possible overkill for many ABM applications has so far ruled out the use, but see Aguilar *et al.* [18] for a complex approach in the footsteps of D*.

BELLMAN-FORD The naming of the algorithm is disputed as the algorithm has been developed concurrently by Ford Jr [78], Shimbel [92], Bellman [93] and Moore [94]. In contrast to Dijkstra-derivatives, all edges are visited per iteration and if one node has a shortest distance, it propagates it to the other. This results in at most $|V|$ iterations and therefore a complexity of $O(|V^2|)$ similar to Dijkstra. More recently, this approach has been used to handle heterogenous dynamic changes in graphs (edge insertions, edge deletions, and edge-length changes) giving it a new competitive edge to Dijkstra-derivatives[95]. The capacity to handle complex changes is beyond most ABM interests and mapping the model to common ABM tasks my be strenuous for non-Computer Scientists.

FLOYD- WARSHALL The naming of the algorithm is, again, disputed as it also has been developed concurrently by Kleene [96], Roy [97], Floyd [98], Ingerman [99] and Warshall [100]. In contrast to Dijkstra-derivatives, the algorithm computes paths $P_{s,t}$ by only using an increasing subset of all nodes in V. It has complexity $O(|V|^3)$, but it finds the paths between all nodes. The original implementations only computed path length, but with a small addition, the path itself may be computed as a shortest-path tree[101]. The non-agent-centric approach of this algorithm is probably the reason, why it has not been considered for routing in ABM.

2.2 WAYFINDING IN COGNITIVE SCIENCE

Cognitive Science itself is already considered an interdisciplinary science as contributions range from psychology, philosophy, linguistics, anthropology, neuroscience and artificial intelligence[102]. The subfield of spatial cognition specialises in spatial learning and spatial behaviour and is most strongly associated with cognitive psychology and neuroscience[103]. Today's research focusses on the acquisition of spatial knowledge, the format of such knowledge within human memory and the application of such knowledge during real and imagined movement[103].

In the traditional approach to navigation, three pillar form the mental representation: Landmark Knowledge, Route Knowledge and Survey Knowledge[43]. According to Siegel & White [104] landmarks are unique views which are remembered but are not associated with spatial information other than local context themselves[43]. Route Knowledge emerges when landmarks are sequentially chained by experiencing a path through the environment. In Route Knowledge landmarks are annotated with routing information such as "turn left". Survey Knowledges is considered to emerge as (separate) Route Knowledge is integrated into single representation[43]. It is usually described as a map-like or configurational knowledge[104], but see Thrash *et al.* [105].

The strict division into the three types of knowledge has been replaced with a more incremental approach[43]. According to Montello [43] the processes run concurrently and influence each other. Metric information is immediately gathered and not only derived in a last step when creating survey knowledge. Recent research postulates Graph Knowledge that sits in between Route and Survey Knowledge as an alternative representation to the concurrency of the Route and Survey Knowledge[106].

In this theoretical frame of less separated forms of knowledge, wayfinding is the process of acquiring new spatial knowledge, processing it and finally applying it[107]. Wayfinding is generally not least effort, shortest path or distance minimising[108]. Route knowledge is acquired via procedural rules[109] and thus routing can be understood as the production of a route. This has been modelled early on by Kuipers [24] as the TOUR model where *views* and *actions* allow the actor to act upon the viewed to obtain new views. This model focusses on "knowledge in the head"[29] whereas later models such as the *wayfinding graph*[29] also add "knowledge in the world"[29].

Those models still do not fully account for the process of storing the knowledge and retrieving it. There are two contesting models on how the the knowledge is learned. On the traditional side, it is assumed that mental spatial representation is build up by a sequence from egocentric to allocentric[110]. On the other side, recent research suggests that mental representations using difference frames of reference can develop in tandem[65,66].

In particular, there is a lot of literature about errors that are introduced in the process and that lead to objectively wrong, but subjectively appropriate, decisions in the subsequent steps. For instance, Loomis & Knapp [111] have observed that participants in their study consistently underestimate the goal location in a virtual environment by up to a factor of two. In an unpublished study[105] at the Chair of Cognitive Science at ETH Zürich[2] the directional encoding error for the mental representation was captured probabilistically as a von Mises distribution.

Briggs [112] analysed configurational knowledge and built the foundation for later observations by Hayashi *et al.* [113] that error terms are cumulative as more segments along the same route increase error. Based on these works, Golledge [109] hypothesises that humans do create something akin to a minimising criteria in the mental representation to produce a route.

As a whole, Spatial Navigation research has allowed us to gain insight into the process of wayfinding. In particular, we have gained a better understanding of the steps necessary in wayfinding and the accompanied errors that may be introduced. However, few have tried to develop computational models of spatial behaviour from a cognitive science point-of-view[24,29,47]. It is in this context that I apply ABM to test contemporary theories on the structure of knowledge within human memory that underlie the wayfinding process.

2.3 SYNTHESIS OF ROUTING AND WAYFINDING

As shown by the previous sections, wayfinding and routing are two perspectives on the same issue. However, the computational approach focusses more an a mechanical solution how to obtain a route from knowledge irrespective of human cognitive processes. In contrast, wayfinding also incorporates the acquisition process and the knowledge

2 Available from the author upon request.

storage under the consideration of a human actor. In ABMs in the wild, this distinction is usually not understood and consequentially the wayfinding process is separate from the routing. This unjustified, but commonly practised, separation can be considered another reason why routing in ABM has not been improved upon in recent decades other than technical optimisations.

The cognitive conceptualisation of wayfinding gives us new opportunities to rethink traditional Computer Science routing and bring it closer to reality. I intend to connect the two by explicitly modelling the error processes that accumulate when human agents transform mental representations.

The recent approach of the MomenTUM framework[34,47] tries to provide a holistic approach to modelling pedestrian movement in ABM under consideration of cognitive processes[64], not only spatial cognition. This allows to model more clearly a difference between strategic, tactical and operational actions by the agents and correctly associate them with different cognitive processes and biases. The MomenTUM framework will also be the backdrop of this work and allows me to apply a state of the art ABM for pedestrian movement while solely focusing on the spatial cognition component under consideration.

In this work, I demonstrate that a cognitive routing model provides distinct routing behaviour compared to previous shortest path solutions exemplified with the A* algorithm. The cognitive routing model's algorithm C* is an adaptation of A* that uses different heuristics. Instead of optimising for the best (i. e. shortest) solution, I optimise to match human error as close as possible. This distinction allows agents to explore space more naturally (see Ch. 4.3.3).

3

METHODS

> *The true logic of this world is in the calculus of probabilities.*
>
> — James C. Maxwell

I develop a new model for routing in ABM based on findings in Cognitive Science. This work is not about reinventing the wheel (routing) for ABM and therefore I use the MomenTUM framework as a baseline. As most other ABM implementations, MomenTUM provides an implementation of A* algorithm but it is mislabelled as a simple Dijkstra implementation[64]. However, MomenTUM, due to its modular design, only offers "Dijkstra" as one of many options and allows end users to devise their own algorithms. MomenTUM provides an extension format for agents that I used to implement the cognitive routing model.

I propose the hypothesis that in aggregate cognitive routing produces a different density distribution than classical routing. Based on error models established in Cognitive Science those agents under cognitive routing work less optimally, but more similar to humans.

3.1 COGNITIVE ROUTING MODEL

In this section, the theoretical background for the model is provided. The different findings from Cognitive Science are put into context with the Computer Science algorithms and a model for cognitive routing is derived.

The first key observation is based on configurational knowledge. This structure of knowledge encodes relative position and direction between points to establish a layout[112] and thus operations like routing are power functions thereof.

The second key observations is that variations in memory are systematic (shown in the next two subsections), and whereas the structure of memory is yet to be determined, experimental observations have shown consistent properties of the error terms. In the following subsections, those structured error terms will be explored.

3.1.1 *Distance Deviation*

Distance estimation is a crucial step in wayfinding[114]. Early studies showed that the perceived distance is usually shorter than the actual distance[115]. This effect is moderated by different measures such as training[116], age[117], and environmental framing[118,119], but some moderators, such as feedback on relative distance judgements, have no effect[120].

Stevens [121] offers the first quantitative formalisation, see Eq. 3.1, that captures the over- or under-estimation. A long line of research[122–126] has since computed the coefficient and found it to be in the range of $\beta = 0.95 \pm 0.2$.

$$d(x) = x^{\beta} \tag{3.1}$$

Vista space describes the visually perceptible space from a single location without locomotion[127]. Systematic analysis has questioned the uniformity of the vista space and has brought forth different divisions of the vista space[128–130]. There are systemic differences between perception in frontal and sagittal vista space. Frontal vista space is measured as an ego-centric distance, whereas sagittal vista space is defined as the visual distance between objects in view[131]. Studies have found that the former is compressed and the latter is overextended[128]. Another transformation takes place between near and far vista space[130] with a tendency to underestimate short distances and overestimate large distances, see Eq. 3.2.

$$d(x) = \begin{cases} x^{\beta_1} & \text{, if } x \leq 100 \\ x^{\beta_2(x)} & \text{, otherwise.} \end{cases} \tag{3.2}$$

Daum & Hecht [130] note that the threshold of $100m$ is relatively arbitrary as the number of measurements they took in the range is limited. Furthermore, the exact coefficient β_2 was not computed, but is assumed to be larger than β_1 and a function of the distance.

3.1.2 *Direction Deviation*

Previous research has found a range of common direction errors that differ slightly, but overall stay within the same range. Early research shows direction errors of up to 35° for judgment of relative direction (JRD) tasks[4,132] . More recent research found direction erros of 20° for onsite JRDs and up to 40° for offsite JRDs[133] where onsite refers to judging within the environment and offsite refers to judging from memory only. In an experiment with binned direction (each octant counts 45°), participants exhibited a mean success rate of 0.2[134] or only 20% of the trials were off by more than 5° to 50°. The original work does not provide more detailed numbers, but with a 5° overlap to the next bin, these numbers are the upper and lower bound for the error. Several studies found that learning reduces error significantly[133,135,136], but ultimately levelled off without further gains.

Following the error modelling of Thrash *et al.* [105], I encode the error probability as a von Mises distribution[137] which is also known as the circular normal distribution. This allows me to specify the probability of an agent remembering the correct direction with a mean $\mu = 0$ and a concentration κ expressing a reciprocal measure of dispersion. Alternatively, $\frac{1}{\kappa}$ can be analogously understood to σ^2. For comparison, the function assumes a uniform distribution if $\kappa = 0$ and concentrates around μ as $\kappa \to \infty$. The von Mises distribution for radian $x \in [0, 2\pi]$ is defined as

$$p(x|\mu,\kappa) = \frac{e^{\kappa \cos(x-\mu)}}{2\pi I_0(\kappa)} \tag{3.3}$$

where I_0 is the modified Bessel function. In Eq. 3.4 it is defined as a Riemann-sum[138] and in Eq. 3.5 as an integral[139]:

$$I_\alpha(x) = \sum_{m=0}^{\infty} \frac{1}{m!\Gamma(m+\alpha+1)} \left(\frac{x}{2}\right)^{2m+\alpha} \tag{3.4}$$

$$I_\alpha(x) = \frac{1}{\pi} \int_0^{\pi} e^{x\cos\theta} \cos\alpha\theta d\theta - \frac{\sin\alpha\pi}{\pi} \int_0^{\infty} e^{-x\cosh t - \alpha t} dt \tag{3.5}$$

The integral form of 0 order modified Bessel function thus collapses to Eq. 3.6:

$$I_0(x) = \frac{1}{\pi} \int_0^{\pi} e^{x\cos\theta} d\theta \tag{3.6}$$

3.1.3 *C* Algorithm*

While typical routing algorithms such as A* focus on shortest paths under objective criteria (i.e. Euclidean distance), I suggest an algorithm that transforms the objective criteria based on the cognitive encoding error to obtain a minimising criteria in the sense of Golledge [109].

To model the influence of human estimation errors, I follow the two-tiered reality and belief representation[140] where the reality (*facts* about the environment) and the agent's cognition (*beliefs* about the environment) are represented separately. The category *beliefs* derive from the Artificial Intelligence literature, where they denote that an agent's representation differs from reality[141].

At the onset, C* (for Cognitive) is a derivative of A* and if an agent's *beliefs* were to match *facts*, they would be equivalent. The algorithm is differing from A* as it uses an adapted distance estimation function that represent rather the *beliefs* than the *facts*. This changes the agents' behaviour in two ways. First, the agent may have incorrect beliefs about the location of the goal. Second, the agent may have incorrect beliefs about the location of other nodes.

Beliefs about the goal location matter most as they lead the agent towards a wrong point in space. Nonetheless, the error will get small when the agent approaches the goal. This manifests itself in the form of the heuristics function and eventually the agent converges towards the goal.

Beliefs about other nodes influence the agents choice directly, if some nodes are remembered to be closer to the goal than they actually are, agents may choose a longer route which they believe to be shorter. However, as they draw closer to the destination, this error minimises and eventually the agent converges towards the goal.

The C* algorithm consists of two components that work independently, but influence one another. The *routing* component and the *memory* component. Unlike classical A*, C* requires for each agent to maintain a memory of all known locations which introduces an increased memory footprint in the order of $O(|V|)$.

3.1.3.1 *Routing Component*

The implementation of the actual routing exhibits only two differences compared to A*, see Alg. 2.2 and Alg. 3.1. The heuristic function is more complex and instead of the actual length the expected length is used. While the changes seem minor, they bring the algorithm closer to human behaviour. The complexity of the algorithm remains the same at $O\left(\overline{deg(v)}^{|P_{s,t}|}\right)$, but average performance is slightly worse as the heuristic more often will choose a less efficient solution, see discussion in Ch. 2.1.3.

3.1.3.2 *Memory Component*

The implementation of the memory component is kept simple. For each known location, the algorithm keeps the distorted coordinates in memory. This means that in every iteration, for each agent, the perceived distortion is computed and stored. In practise, I shift the coordinates along a polar coordinate system with the agent in its centre. The distance to a location x is computed based on Eq. 3.1 (but could be modelled on Eq. 3.2 if more accurate data was available, both regarding the threshold and the functional form of $\beta_2(x)$). The direction error is computed by drawing a random sample from the von Mises distribution as shown in Eq. 3.3. The new location x' is computed by scaling x according to function d and rotation $\mathbf{R}$ by the draw from the von Mises distribution.

$$x' = \frac{|d(x)|}{|x|}\mathbf{R} \cdot x \tag{3.7}$$

As random draws each iteration would average out towards the mean, the new draws only contribute at a rate of $\delta = 0.2$ to the remembered location, see Eq. 3.8. The equation is based on prototype weighting[142] but without empirical data the value is arbitrary. The chosen value of $\delta = 0.2$ allowed for acceptable simulation results. Nonetheless, fine-tuning with real world data would be desirable to perform validation and verification[143].

$$x'_{update} = \delta * x' + (1 - \delta)x'_{old} \tag{3.8}$$

Available parameters for the memory component are κ parameter of the von Mises distribution, the standard deviation σ_β of the compression parameter β of Eq. 3.1 and the update rate δ. The κ parameter allows to reduce/increase the spread of the von Mises distribution and therefore expresses the familiarity that an agent has with the destination in terms of direction error. The σ_β expresses the deviation from the population distance error mean and therefore also expresses familiarity with the destination. Lastly, the δ

Algorithm 3.1 C* Algorithm

Require: empty set Q_{open}, Q_{closed}; arrays score$[v] = \infty$,heuristic$[v] = \infty$, previous$[v]=$ NULL $\forall v$

Q_{open}.Add(s)
score$[s] \leftarrow 0$
heuristic$[s] \leftarrow$ CognitiveHeuristic(s, t)
while $\neg$ Q_{open}.IsEmpty **do**
    $u \leftarrow$ MinimalCognitiveHeuristic(Q_{open},t) ▷ Vertex with lowest heuristic to t
    if $u = t$ **then**
        ReconstructPath(t,previous) ▷ Returns $P_{s,t}$
    end if
    Q_{open}.Remove(u)
    Q_{closed}.Add(u)
    for all v in Neighbourhood(u) **do**
        if $v \in Q_{closed}$ **then**
            Continue ▷ Only for monotone heuristic functions
        end if
        if $v \notin Q_{open}$ **then**
            Q_{open}.Add(u)
        end if
        $altScore \leftarrow$ score$[u]$ + ExpectedLength(u, v)
        if $altDist \leq$ score$[v]$ **then**
            score$[v] \leftarrow$ $altDist$
            heuristic$[v] \leftarrow$score$[v]+$ CognitiveHeuristic(v, t)
            previous$[v] \leftarrow$ u
        end if
    end for
end while

parameter expresses the drift rate at which the remembered location changes compared to the last recall. Drift is a known phenomena[142], but is little studied and no empirical data was available.

The computational complexity does not increase as the linear traversal of all nodes $O(|V|)$ to compute the distortion is well below the quadratic complexity $O(|V|^2)$ of the routing computations.

3.2 MOMENTUM FRAMEWORK

The MomenTUM Framework is a pedestrian ABM simulator[34]. The agents follow a cognitive model called *Spice* which stands for (Sp)atial sequent(i)al choi(ce)[39]. It is modelled following the design of ACT-R (Adaptive Control of Thought-Rational)[144,145] and relies on the structuring into an external world, perception, intention, motor, knowledge, and production rules[39,145]. It extends previous methodology for pedestrian modelling[146] by explicitly modelling perception and memory[39], see Fig. 3.1.

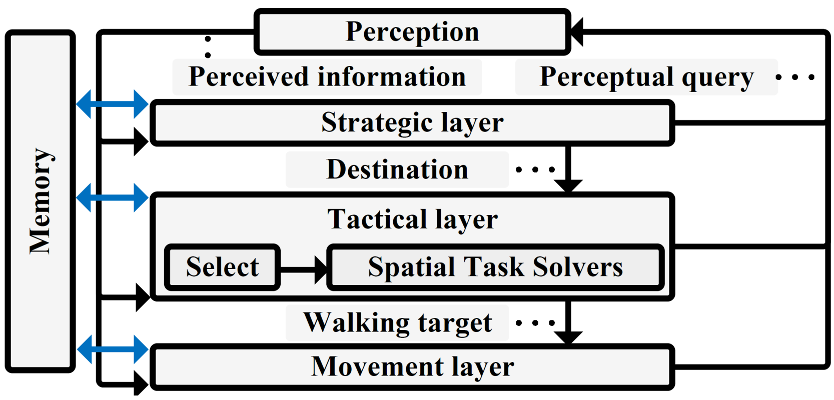

FIGURE 3.1: **Complete Pedestrian Behavioural Model** – The simple behavioural model, see Fig. 1.2 is extended with memory and perception and completed the internal model of the agent. The figure has been taken from Kielar & Borrmann [39] and is printed with the authors' permission.

Each module of the *Spice* architecture is generic containers for cognitive processes and MomenTUM allows users to specify their own processes[39]. The breadth of available modules is large, however, I will focus on a subset that is sufficient to compare the different types of routing.

3.2.1 *Cognitive Routing Extension for MomenTUM*

The Cognitive Routing Model is implemented in the `Cognitive Routing Extension` for MomenTUM, available on `https://github.com/jugdemon/MomenTUM`. The code consists of a new `Extension` and `Tactical Model` that provide the computation of the new heuristics and the management of the agent's memory as described in the previous sections.

The von Mises distribution is sampled with the modified Bessel function as described in Eq. 3.3. As the modified Bessel function would be computationally heavy to correctly compute, a numerical approximation is used as derived in Press [147, p. 274]. The implementation is adapted from the Taylor expansion by Vogelaar [148], shown in Eq. 3.11.

$$y := \frac{|x|}{3.75} \tag{3.9}$$

$$z := \frac{3.75}{|x|} \tag{3.10}$$

$$I_0(x) \approx \begin{cases} 1 + 3.52y^2 + 3.09y^4 + 1.21y^6 + & \text{if } |x| < 3.75 \\ \quad 0.27y^8 + 0.04y^{10} + 0.005y^{12} & \\ \frac{\exp(|x|)}{\sqrt{|x|}}(0.40 + 0.013z + 0.002z^2 + & \text{otherwise.} \\ \quad -0.002z^3 + 0.009z^4 - 0.021z^5 + & \\ \quad 0.026z^6 + 0.016z^7 + 0.004z^8) & \end{cases} \tag{3.11}$$

3.2.1.1 *MomenTUM Configuration*

The MomenTUM Configuration, shown in App. A, is based on the basic example configuration provided with MomenTUM in the documentation. No settings have been changed except for the routing, the layout and graph generation and where explicitly stated. The main features of the basic example are outlined to present expected behaviour. The following models based on previous research are in the basic example: a perception model, walking model, standing model and a staying model. The perception model is based on Bresenham [149] and allows agents to perceive visible nodes in the graph[39]. The perception model is set to a spatial resolution of $0.1m$ with a maximal perception distance of 500m. The walking model is an implementation of the Social Forces model[150] with some adaptations to improve numerical issues[151]. The Johansson Standing Model is set within the Social Forces model[152] and allows pedestrians to remain idle. Lastly, the staying model is the Shifted Participating Model[153]. Standard values for the above models have been used and can be reviewed in App. A. The simulation is set to run for 20 minutes at a time step of 0.05 seconds.

The layout, shown in Fig. 3.2, consists of a 305m $\times$ 310m area with 100 25m $\times$ 25m blocks arranges as a 10 $\times$ 10. Blocks represent buildings or open squares (pink; B, C, E, F in Fig. 3.2). Squares are also intermediate locations for strategic decisions in the MomenTUM framework. Agents will enter and exit in two protruded areas to the North (A, D in Fig. 3.2). The graph generation algorithm I use was suggested by Kielar [154] and its exact configuration can be read in App. A. Its result is shown in Fig. 3.3.

For the purpose of demonstrating the Cognitive Routing model, the strategy of the agents is defined by an Origin-Destination-Matrix[34]. Such a matrix defines the probability of chosen any other goal location as the next target. A random matrix was generated based on the MomenTUM example by Grübel [156], see Tab. 3.1. For intermediate destinations B, C (see. Fig. 3.2) the probability of visiting was set to 0 to simplify subsequent analyse of pedestrian flows. Agents were generated at the start location such that on average 1 to 1000 agents were in the simulation. After an initial up-ramping phase, the average number of agents was maintained throughout the simulation.

3.3 STATISTICAL ANALYSIS

To analyse the resulting movement patters, 2d-histograms over the layout are generated for both the standard routing and the cognitive routing. To evaluate the two-sample comparisons of multivariate data, we use kernel density estimates (KDE)[3]. A classical t-test is a parametric density-based comparison. Duong *et al.* [3] describe how to compare two vectorised images by replacing the parametric with a non-parametric counterpart. The non-parametric density estimation is obtained by kernel smoothing due to its intuitive construction and interpretation[157]. In this section, the test statistic and its motivation is explained.

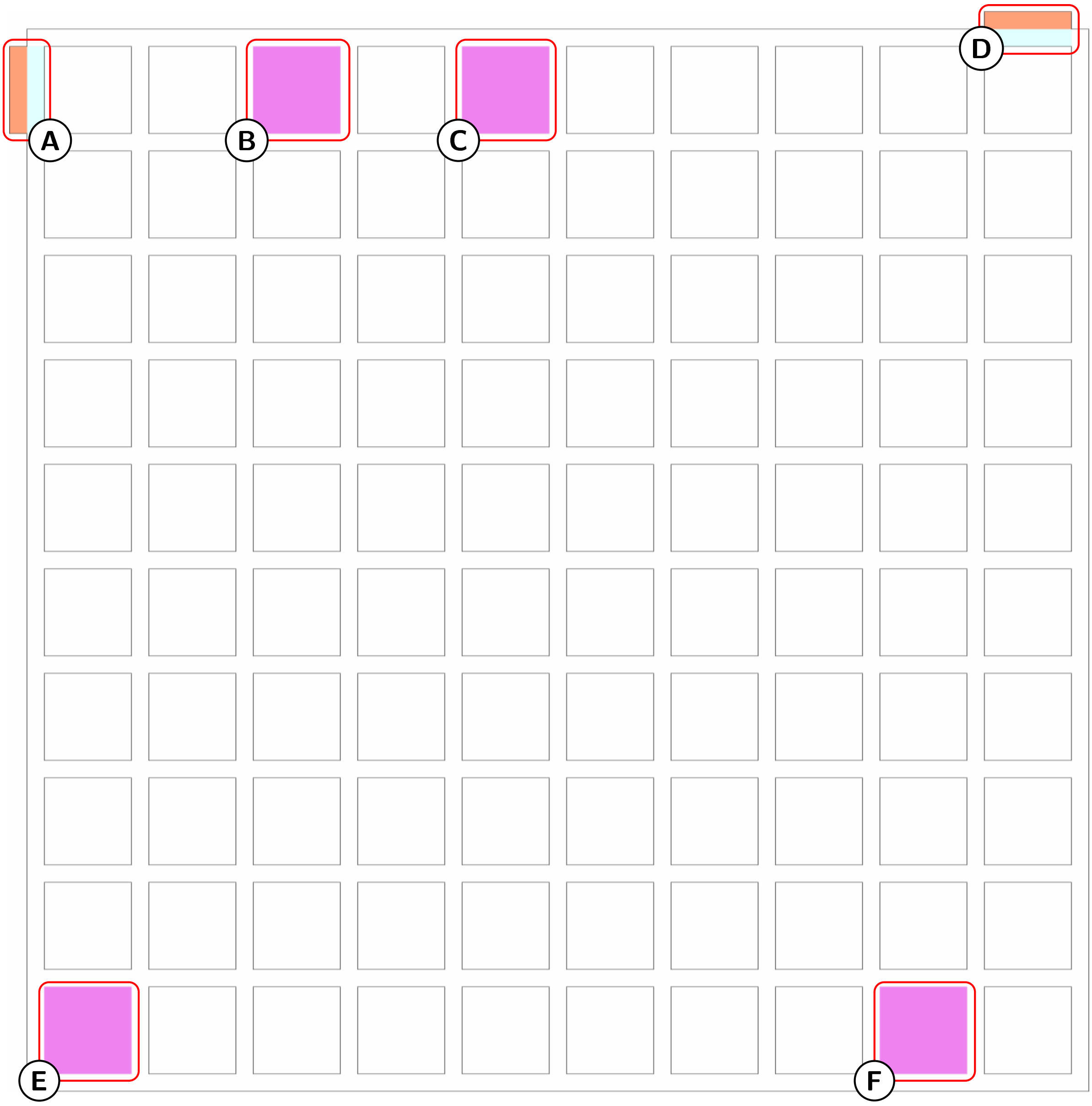

FIGURE 3.2: **Layout Overview** – The layout consists of a 10 x 10 grid of quadratic blocks with a side length of 25m. Blocks are symbolic buildings and cannot be passed or seen through. The paths between blocks are 10m wide. (A, D) mark start (turquoise) and exit (orange) locations. (B, C, E, F) mark intermediate locations (pink), walkable and see-through. Visualisation generated from the MomenTUM Visualisation Tool[34] and overlay from Latex Overlay Generator[155].

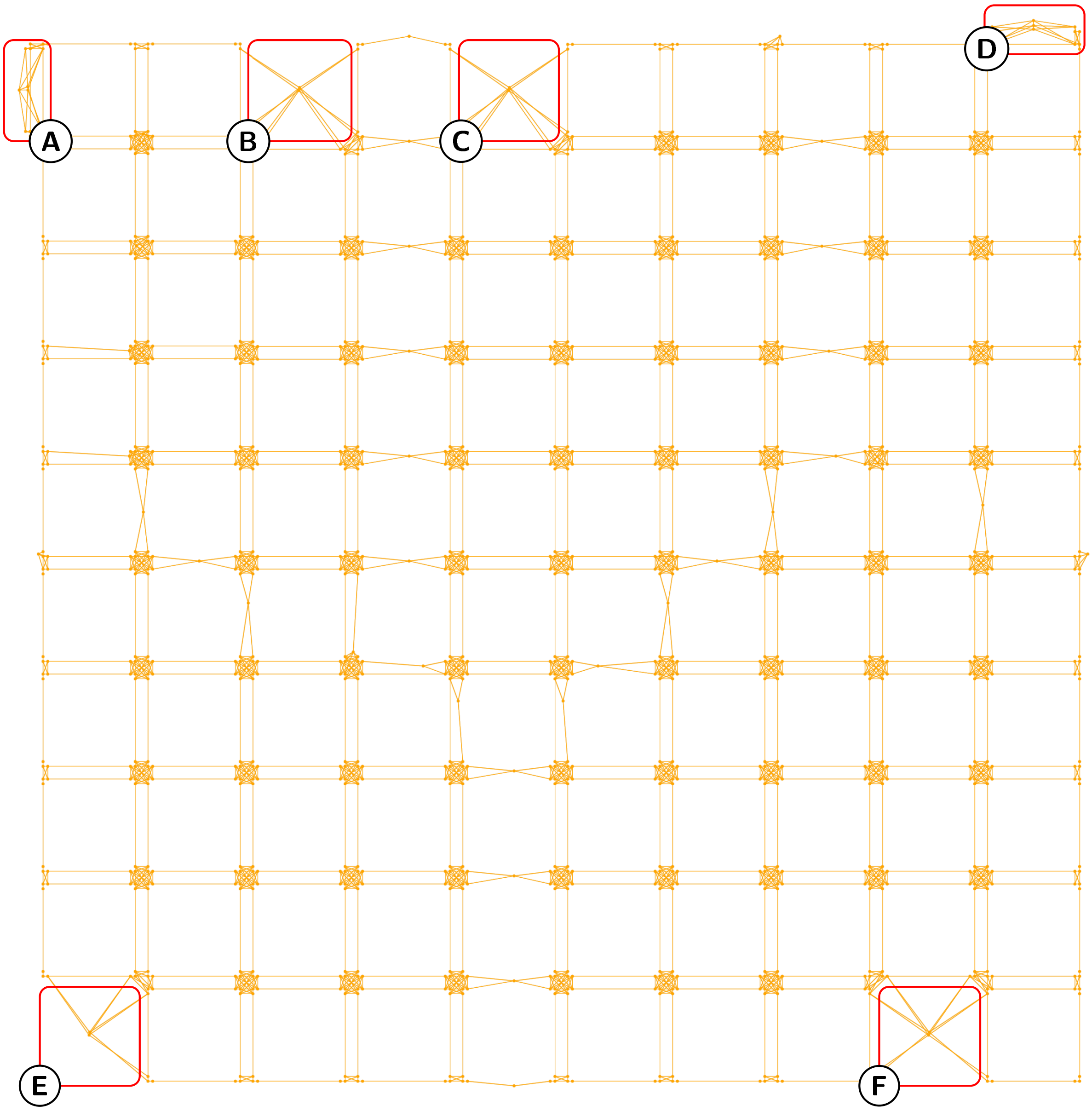

FIGURE 3.3: **Graph Overview** – The automatically generated graph based on the layout in Fig. 3.2. (A, D) mark start and exit nodes. (B, C, E, F) mark intermediate nodes. Visualisation generated from the MomenTUM Visualisation Tool and overlay from Latex Overlay Generator[155].

	A	D	E	F
A	0.487	0	0.513	0
D	0.608	0	0.392	0
E	0	0.606	0.345	0.049
F	0.286	0.467	0	0.247

Table 3.1: **Origin Destination Matrix** – Destinations B and C are omitted for readability. The rows denote the probability at a location to visit any other location in the columns and sums to 1. For instance, if agents are at location D, they may choose A with a probability of 0.608 or E with a probability of 0.392, but will never choose D or F as their next target. Note that self-references are explicitly allowed and mean that agents have a chance to remain at the same location.

Based on the statistical framework in Schauer *et al.* [158], for common densities f_1 and f_2 d-variate random samples $X_1, X_2, ..., X_{n_1}$ and $Y_1, Y_2, ..., Y_{n_2}$ are drawn respectively. For a Kernel K,see Eq. 3.14, the density estimates of f_1 and f_2 can be constructed as in Eq. 3.12 and 3.13 where H_l is a bandwidth matrix for $l = 1, 2$.

$$\hat{f}_1(\mathbf{x}, \mathbf{H_1}) = \frac{1}{n_1} \sum_{i=1}^{n_1} K_{\mathbf{H_1}}(\mathbf{x} - \mathbf{X_i}) \tag{3.12}$$

$$\hat{f}_2(\mathbf{x}, \mathbf{H_2}) = \frac{1}{n_2} \sum_{i=}^{n_2} K_{\mathbf{H_2}}(\mathbf{x} - \mathbf{Y_i}) \tag{3.13}$$

$$K_{\mathbf{H_l}}(\mathbf{x}) = |\mathbf{H_l}|^{-\frac{1}{2}} K(\mathbf{H_l}^{-\frac{1}{2}}\mathbf{x}) \tag{3.14}$$

To test the null hypothesis

$$H_0 : f_1 = f_2, \tag{3.15}$$

a test statistic is derived from a discrepancy measure[159], shown in Eq. 3.16, and can be rewritten in terms of the density estimates[3], shown in Eq. 3.17 where $\psi_l = \int f_l(\mathbf{x})^2 d\mathbf{x}$ for $l = 1, 2$ and $\psi_{i,j} = \int f_i(\mathbf{x}) f_j(\mathbf{x}) d\mathbf{x}$.

$$T = \int [f_1(\mathbf{x}) - f_2(\mathbf{x})]^2 d\mathbf{x} \tag{3.16}$$

$$T = \psi_1 + \psi_2 - (\psi_{1,2} + \psi_{1,2}) \tag{3.17}$$

The test statistic, shown in Eq. 3.18 is composed based on density estimates Eq. 3.12 and 3.13 and yields the terms shown in Eq. 3.19,3.20,3.21, and 3.22.

$$\hat{T} = \hat{\psi}_1 + \hat{\psi}_2 - (\hat{\psi}_{1,2} + \hat{\psi}_{2,1}) \tag{3.18}$$

$$\hat{\psi}_1 = \frac{1}{n_1^2} \sum_{i_1=1}^{n_1} \sum_{i_2=1}^{n_1} K_{\mathbf{H_1}}(\mathbf{X_{i_1}} - \mathbf{X_{i_2}}) \tag{3.19}$$

$$\hat{\psi}_2 = \frac{1}{n_2^2} \sum_{j_1=1}^{n_2} \sum_{j_2=1}^{n_2} K_{\mathbf{H_2}}(\mathbf{Y_{j_1}} - \mathbf{Y_{j_2}}) \tag{3.20}$$

$$\hat{\psi}_{1,2} = \frac{1}{n_1 n_2} \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} K_{\mathbf{H_1}}(\mathbf{X_i} - \mathbf{Y_j}) \tag{3.21}$$

$$\hat{\psi}_{2,1} = \frac{1}{n_1 n2} \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} K_{\mathbf{H_2}}(\mathbf{X_i} - \mathbf{Y_j}) \tag{3.22}$$

$$\tag{3.23}$$

According to Duong *et al.* [3], the test statistic describes the intrasample pairwise differences (Eq. 3.19 and 3.20) to the intersample pairwise differences (Eq. 3.21 and 3.22). That is, if the intersample differences are larger than the intrasample differences, the two distributions are different. Furthermore, Duong *et al.* [3] establish asymptotic normality under the null hypothesis.

3.3.1 *Convergence*

ABMs and this simulation in particular are stochastic processes and therefore require Monte Carlo sampling to achieve statistical robustness for hypotheses testing and varying experimental parameters[160]. I vary two parameters of the overall simulation, the routing algorithm and the average number of agents in the system. The routing algorithm is either the new cognitive routing with the C* algorithm and the classical routing with the A* algorithm. The average number of agents in the environment has been varied systematically with the following values: 1 agent, 10 agents, 100 agents, and 1000 agents.

Large sample counts are generally desirable. However, with too large samples, the sensitivity of statistical tests can lead to the exposure of minuscule and inconsequential differences. Lee *et al.* [160] suggest to compute the minimum sample size to counter such effects. The recommended optimal minimum sample size is dependent on the objective goal as output distributions are usually *a priori* unknown. A subjective criteria for stability is unavoidable as there are currently no established statistical methods to compute these objectively[160].

I determine minimum sample size based on two considerations. First, as the KDE test is my tool of choice, I compute the minimum sample size based on the robustness of the KDE test results. To that end, I establish two kinds of robustness checks based on my parameter variation. As all simulations with the same routing algorithm are drawn from the same distribution, see Eq. 3.12 and 3.13, I expect the KDE test to correctly identify the draws from the same distribution at the same level of average number of agents. I define convergence to be the point where the KDE test reliably identifies draws from the same distribution as equal.

Second, I observe individual behaviour and I expect that adding more agents acts as if I repeat a single agents behaviour more often as long as the density of agents does not cause excessive crowdedness. Excessive crowdedness demarks the level of other agents at the environment at which the behaviour deviates from simple routing to crowd-evasion. To establish that I can aggregate accordingly I compare draws from the same distribution where one draw consists of all agents in one simulation and the other draw consists of repeated simulations with less agents until the number of agents is equivalent in both scenarios. The aggregated scenario is compared with multiple scenarios single simulation of the next larger average agent number scenario and the minimum is taken. The minimum specifies the the largest difference between scenarios.

The two algorithms are then compared at the level of agents that provides stability according to my criteria. Additionally, I will provide the results for the other levels at well, but my discussion will revolve around the stable values.

3.3.2 *Implementation*

The test is run with the **R**-Package `ks`[161] that implements Kernel Smoothing techniques. In particular, the function `kde.test` is used that implements the kernel density estimation test[3]. The parameters for `kde.test` are automatically estimated by the `ks` package based on work of Wand & Jones [162] and Chacón & Duong [163].

4

RESULTS

Essentially, all models are wrong, but some are useful.

— George E. P. Box

The simulation shows that introducing uncertainty into the routing process creates more diverse routing behaviour in otherwise equal agents. I will show the results in three steps.

In a first step, I will visually inspect the density of pedestrians under the different routing processes to qualitatively assess differences between the models. In a second step, I will visualise the mental representation of the environment to further comprehension of the differences between the routing options and to explain the behavioural differences observed in the first step. In a third and last step, I will apply statistical comparisons to quantify the difference.

4.1 DENSITY VISUALISATION

A 2D-histogram over the traces of all agents, shown in Fig. 4.1, reveals more diverse route choice under the Cognitive Routing Model. Instead of only walking along the shortest path, the agents choose more often non-optimal routes and also distribute more. The centre of the environment is highly frequented but does not have a main route (compare row 4 left versus right). However, highly unlikely routes (shown in shades green in row 2 to 4) are still rarely chosen.

Under classical routing, the agents rarely leave the optimal path, usually only to evade other agents. While evasion is not part of A*, it is part of the behaviour model of the MomenTUM framework discussed before. We can see that with an increasing number of agents, the classical routing algorithm diverges more due to some minor congestion. However, the diffusion never takes up a degree similar to the cognitive routing algorithm, compare left to right column in Fig. 4.1. The shortest paths remain highly frequented compared to secondary routes to evade other agents (compare orange and green routes in row 4).

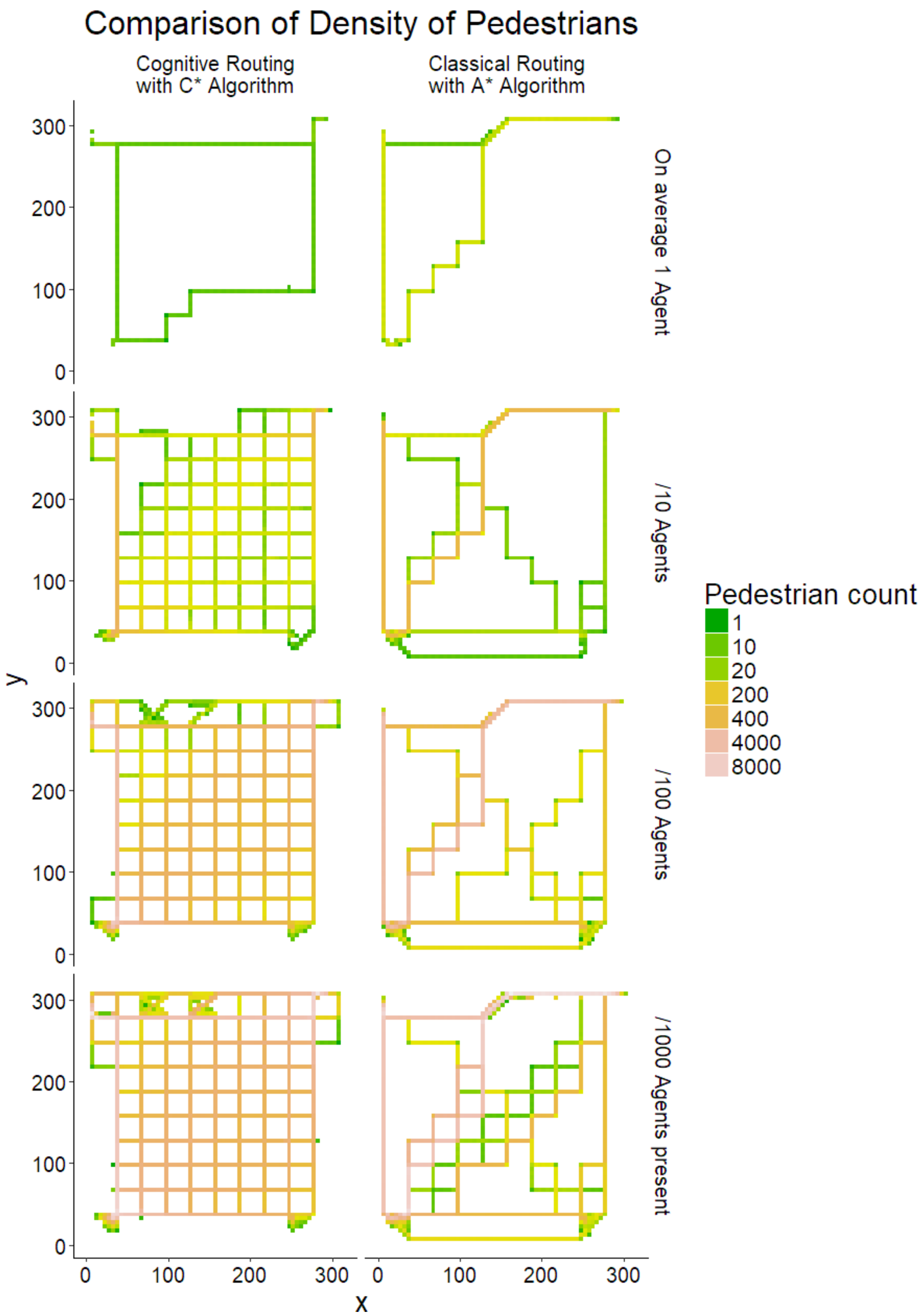

FIGURE 4.1: **Fine-grained Path Density** – All visualisations show $5m \times 5m$ bins over the layout shown in Fig. 3.2. The destinations can be clearly seen at the bottom of each image. *To the left:* the Cognitive Routing is applied. Even paths that are not on the optimal route are highly frequented. *To the right:* the classical routing is applied. Mostly the optimal routes are used, only some interaction with other agents drive agents off their path. *From top to bottom:* the average number of agents in the environment is increased.

Nonetheless, we need to keep in mind, that technically, we operate on space which nearly exhibits a L1 distance metric (Manhattan Distance)[164]. On a perfect grid (with no path width), multiple paths are equally optimal under Manhattan Distance. However, the road width of $10m$ allows agents to move on diagonals along one segment, thus optimising the path slightly more (which shows up in the classical routing as *zigzagging*[165–167]). In particular, the open space at intermediate location C allows agents to move along a diagonal instead of following the Manhattan Distance. Consequently, under classical routing, all agents at A and E on the way to D try to pass through C as it optimises the path length and creates a *bottleneck*. In general, agents under classical routing consider even minimal differences in the configuration space to decide for the optimal route. In contrast, the agents under cognitive routing prefer more straight lines as shown in the first row of Fig. 4.1. The distortion in their mental representation is large enough to subdue *zigzagging* as can be seen in Fig. 4.1 rows 2 to 4. The *bottleneck* does not arise as cognitive agents do not account for the minimal improvement provided by the diagonal path.

4.2 MEMORY VISUALISATION

To understand the agents' routing behaviour, I visualise the difference between actual location and remembered location for a single agent. First, I show the snapshot at the start of the simulation in Fig. 4.2. The agent is located at A in Fig. 3.2. The distance underestimation is visible as the length of the segments gets longer the further they are away from the agent. Similarly, the direction distortion is shifting the location of each node radially. The agent exhibits the expected mental representation errors of distance underestimation and rotational distortion.

Fig. 4.3 shows the evolution of an agent on the route from A over E to D. At the beginning, the knowledge of the locations is fuzzy and very distorted, see Fig. 4.3.1. At Fig. 4.3.6 the agent reaches location E and at Fig. 4.3.20 reaches D. The error is lower close to the agent, and larger further away. The initial direction error makes some points drift very far to either side of its location, cf. yellow lines in Fig. 4.3, but they reduce as the agent moves around more.

Throughout the simulation, the quality of the memory improves as the agent passes through the area. Most locations show distance underestimation with slight displace-

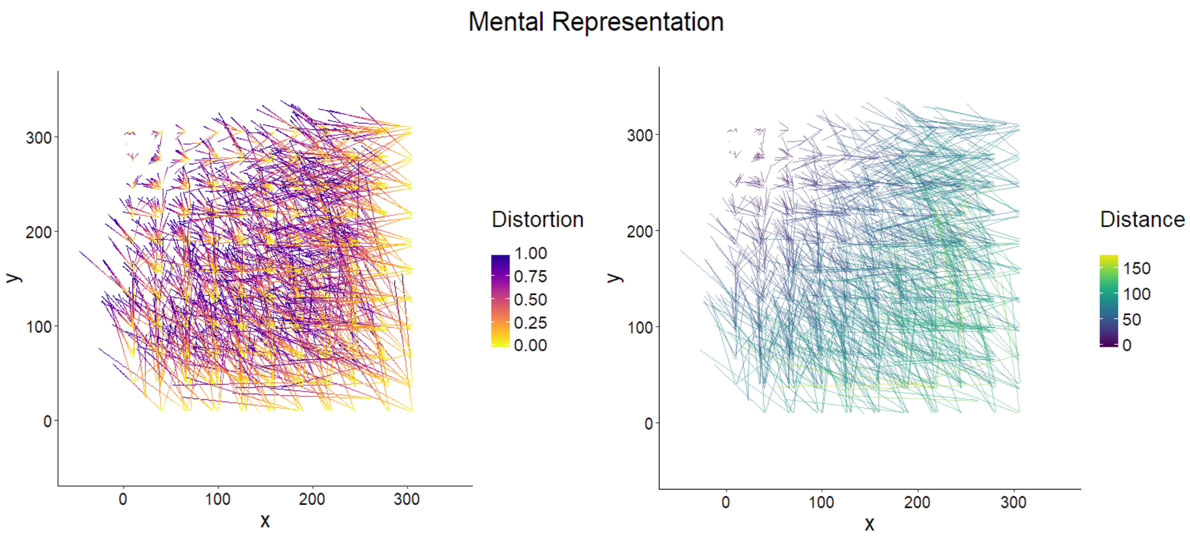

FIGURE 4.2: **Representation Distortion of an Agent** – Each line connects the real location of a node in the graph, see Fig. 3.3, with the mental representation of the same node. *To the left:* The lines are colour-coded such that at the original node location is bright and the distorted node location is dark. The grid can be recognised from the bright spots. *To the right:* The lines are colour-coded according to the length of the distance. Darker colour means more accurate representation whereas brighter colours signify larger distortions.

ments that increase with the distance to the agent. As the agent moves between A and E, we note that error term close to the agent is reduced. Error at already visited locations that are further away is getting worse again, but not at the same rate as the error of locations on the other side, that the agent has not visited yet. This is due to the drift rate discussed previously.

Visual inspection of a randomly selected agents shows that the agent's memory works as expected and provides the kind of encoding errors that previous research has found in human environment encoding up to the specification of our model. Validating this data more precisely than visualising the error terms is difficult as no real world data has been gathered. Most experiments in Spatial Cognition are content with getting a snapshot of the memory in the present and there is little work beyond theoretical contributions[142]. It would be beneficial to obtain time-series of memory states but this remains an open question in Spatial Cognition and is beyond the scope of this thesis.

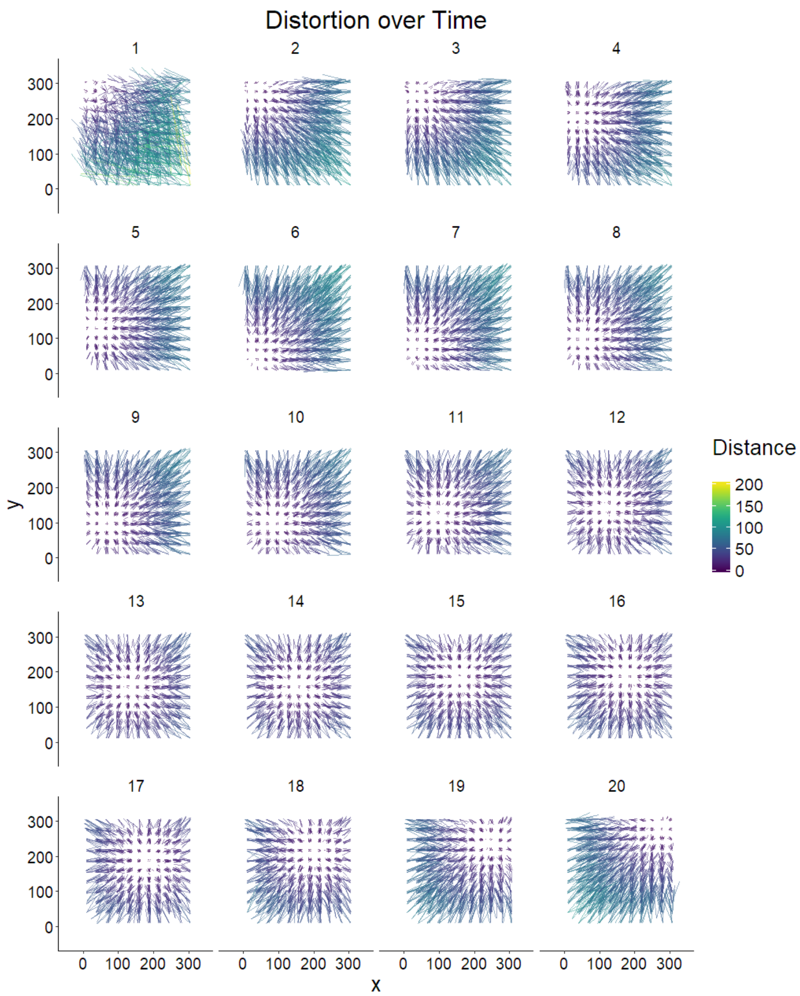

FIGURE 4.3: **Distortion over Time** – Each figure represents the mental representation as in Fig. 4.2. The change of the mental representation over time is shown in 20 snapshots over the course of an agent through the environment. The agent moves from A to E to D as labelled in Fig. 3.2

4.3 STATISTICAL ANALYSIS

As introduced in Ch. 3.3, the 2d-histograms are compared. The comparison will be performed with $5m \times 5m$ bins and $30m \times 30m$ bins. The former is used to capture the space precisely as the bins are aligned with all walls in the environment, see Fig. 4.1. The latter is used to abstract routing decision due to the irregular structure that is introduced through the blocks. The larger bin outsizes the blocks such that the topology is simplified to a regular von Neumann neighbourhood (4-connected)[168], while at the same time maintaining information about the decision points of an agent's path, see Fig. 4.4.

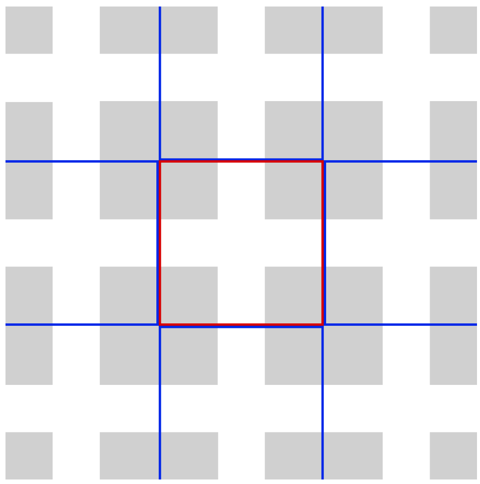

FIGURE 4.4: **Neighbourhood Topology** – The roads are white and the blocks are gray. The red bin neighbours the four blue bins (von Neumann neighbourhood; 4-connected[168]). The bin topology encapsulates all possible paths and preserves the decision points.

Based on the larger bin, I recreate the path density to get a visual impression of the data, see Fig. 4.5. This data representations maintains connectivity information about the agents route choice but abstracts the holes in density produced by the blocks. When comparing Fig. 4.1 and Fig. 4.5, we can see that the most frequented paths are visible in both and that lowly frequented areas are similar with some lowly frequented areas no longer visible due to being grouped into the highly frequented area.

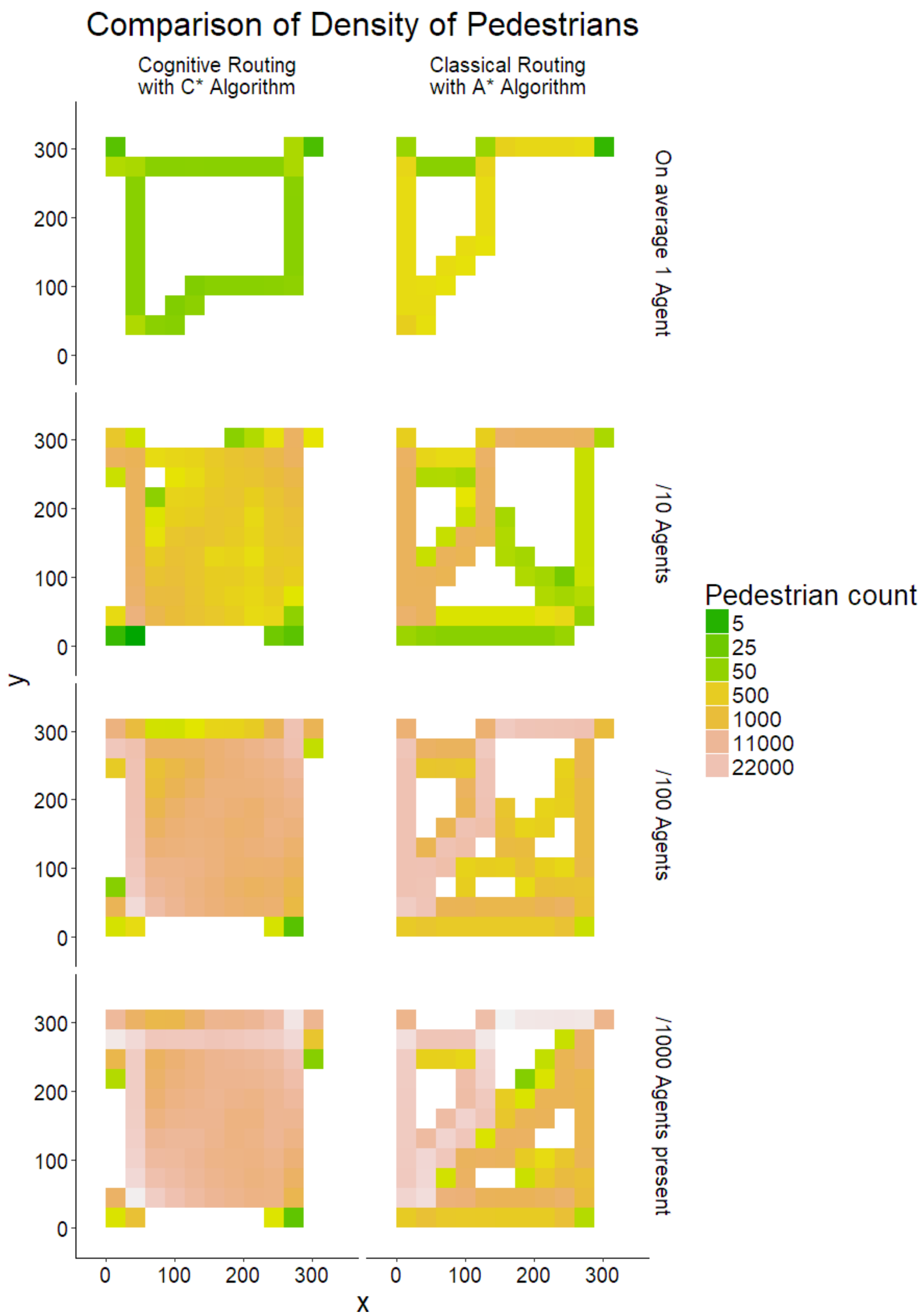

FIGURE 4.5: **Coarse-grained Path Density** – All visualisations show $30m \times 30m$ bins over the layout shown in Fig. 3.2 aligned as shown in Fig. 4.4. *To the left:* the Cognitive Routing is applied. *To the right:* the classical routing is applied. From top to bottom the average number of agents in the environment is increased.

4.3.1 *Convergence Test*

The convergence is tested by applying the KDE[3] test within each routing type. As discussed in Ch. 3.3.1, the goal is to establish that, under the KDE metric, and given the asymptotic normality, instances of one routing type are statistically equivalent. This holds for both within the same level of agents and between different levels of agents with the same number of overall agents. For the same level of agents, 10 simulations were compared with one another and only the minimal p-value among all comparisons is shown in Tab. 4.1. The t-values are difficult to read and do not translate into p-values with the usual table due to the non-parametric form[3]. Therefore, z-values are also provided with the interpretation that the larger the absolute z-value, the more more likely the two distributions are different[3].

Two simulations of the C* algorithm can be concluded to be from the same distribution if there are more than 10 agents involved in a simulation. The simulations of A* have a higher threshold and can only be identified in simulations with 100 or more agents.

To establish that crowdedness does not alter the routing behaviour, the different levels of average agents in the system are compared. For comparing the different levels, the level with less agents is aggregated and then it was compared with each the 10 simulations at the higher level and only the minimal p-value is shown in Tab. 4.2.

Crowdedness has an strong impact on algorithm distribution. While instances with the same level of crowdedness are correctly attributed, we see that for A* algorithm trying to sum densities does not yield the same distribution. The result is less strong for the C* algorithm, where only at very high densities this can be observed. From this sensitivity, I conclude that very low (i. e. on average 1 or 10 agents) and very high densities (i. e. on average 1000) produce corner cases for KDE tests and should be avoided based on the minimum sample size principle put forth by Lee *et al.* [160] and discussed in Ch. 3.3.1.

4.3.2 *Routing Comparison*

Having established that densities created under one routing type are asymptotically similar with a similar number of agents, I now turn to compare the two routing types to establish that C* routes differently than A* in the aggregate.

The binning at $5m \times 5m$, while visually pleasing, runs into analytical issues and produces no significant results, see Tab. 4.3. I. e. under the KDE test, all instances would be drawn from the same distribution. We attribute this to the complex spatial structure that is imposed by the $5m \times 5m$ bins as the blocks remain empty. The self-similarity of empty blocks within both routing patterns probably outweighs in terms of influence over the actual routes when comparing the distributions, see Fig. 4.1.

The issue does not appear for the larger $30m \times 30m$ bins. The blocks do not impose structure onto the paths. The two routing algorithms produce distinctive patterns that the KDE test attributes to different distributions. Keeping the limitations from the convergence section in mind, I discard the results for the average of 1 agent, 10 agents and 1000 agents.

I conclude that the generating procedure for the agents' movement data in a viable scenario, i. e. with 100 agents, is sufficiently distinct to justify the modifications of the algorithm.

4.3.3 *Summary*

I have shown that agents with cognitive routing and agents with classical routing differ in their behaviour visually as well as statistically. While a large number of agents is necessary to show the difference based on the statistical method, it holds asymptotically.

Agents under the C* routing algorithm behave more natural. Over-optimisation artefacts in A* routing such as *zigzagging* and the use of *bottlenecks* is substantially reduced or non-existent. This can be ascribed to the difference in memory as cognitive routing is more "fuzzy" and allows agents to "change their mind" in the process of navigating without changing strategy. This contrasts with previous models that only allowed for explicit re-routing based on a general re-evaluation of goals and perception.

A look into the mental representation of agents has shown that each agent exhibits the encoding errors that have been empirically gathered and analysed in Spatial Cognition over the last century, see Ch. 2. Lamentably, a comparison to real world data is lacking as finding or gathering appropriate data was out of scope for this thesis.

	$5m^2$ bins					
	C*			A*		
Compare # Agents	z	t	p	z	t	p
1 to 1	8.878	0.161	0*	23.837	0.151	0*
10 to 10	0.785	0.001	0.216	7.092	0.006	0*
100 to 100	-0.024	0	0.509	0.087	0	0.465
1000 to 1000	-0.052	0	0.520	-0.076	0	0.530
	$30m^2$ bins					
	C*			A*		
Compare # Agents	z	t	p	z	t	p
1 to 1	2.023	0.004	0.021*	6.6037	0.007	0*
10 to 10	0.108	0	0.456	3.6079	0	0*
100 to 100	0.355	0	0.361	0.1223	0	0.451
1000 to 1000	-0.295	0	0.616	-0.2957	0	0.616

Due to non-parametric t-values, p-values are computed differently, see Duong *et al.* [3].

TABLE 4.1: **Robustness results within level** – *Top half:* Statistics for small $5m \times 5m$ bins. *Bottom half:* Statistics for large $30m \times 30m$ bins. The columns provide z-values, t-values, and p-values for both types of algorithms. The minimum over 10 runs was taken to make this test conservative. A large z-value or a significant p-value at $\alpha = .05$ implies that the compared data sets are not drawn from the same distribution.

	$5m^2$ bins					
	C*			A*		
Compare # Agents	z	t	p	z	t	p
10×1 to 10	0.407	0	0.341	26.278	0.021	0*
10×10 to 100	-0.061	0	0.524	2.331	0.000	0*
10×100 to 1000	2.168	0	0.015*	25.972	0.001	0*
	$30m^2$ bins					
	C*			A*		
Compare # Agents	z	t	p	z	t	p
10×1 to 10	-0.0710	0	0.528	8.579	0.001	0*
10×10 to 100	0.2135	0	0.415	1.987	0	0.02*
10×100 to 1000	2.1924	0	0.014*	6.754	0	0*

Due to non-parametric t-values, p-values are computed differently, see Duong *et al.* [3].

Table 4.2: **Robustness results between levels** – *Top half:* Statistics for small $5m \times 5m$ bins. *Bottom half:* Statistics for large $30m \times 30m$ bins. The columns provide z-values, t-values, and p-values for both types of algorithms. The minimum over 10 runs was taken to make this test conservative. A large z-value or a significant p-value at $\alpha = .05$ implies that the compared data sets are not drawn from the same distribution.

	$5m^2$ bins			$30m^2$ bins		
Compare # Agents	z	t	p	z	t	p
1	0.915	0.009	0.180	4.335	0.008	0*
10	-0.567	-0.001	0.714	3.037	0.001	0.001*
100	0.539	0.000	0.294	4.155	0.000	0*
1000	-2.527	-0.000	0.994	1.347	0.000	0.088

Due to non-parametric t-values, p-values are computed differently, see Duong *et al.* [3].

Table 4.3: **Comparison results** – The columns provide z-values, t-values, and p-values for both bin sizes. The minimum over the comparison of 10 runs for each algorithm was taken to make this test conservative. A large z-value or a significant p-value at $\alpha = .05$ implies that the compared data sets are not drawn from the same distribution. A significant p-value at $\alpha = .05$ implies that the compared data sets are not drawn from the same distribution.

5

CONCLUSION

We can only see a short distance ahead, but we can see plenty there that needs to be done.

— Alan Turing

Agent-based modelling is a useful tool to explore models in general and in Cognitive Science in particular. Assumptions about the mind can be put to rigorous test by comparing the expected behaviour if the model was correct with some empirical data[13]. However, the lofty ideals often fail when confronted with reality. In this work, I analysed the degree of realism that the standard model of routing in ABM offers and I applied theories from Cognitive Science to suggest a more adequate model for simulating human navigation. The work focuses on the process of routing itself and does not include other confounders. I successfully demonstrated that only including an agent's uncertainty in memory into the model produces a more varied result and allows us to more effectively model real human navigation behaviour.

This contribution is especially important when we consider the most common application of pedestrian ABM: disaster prevention. The difference between the cognitive routing and the classical routing already alters the route choice sufficiently that even in a small scale scenarios simulations of crowds may be substantially off.

The big caveat of my research is that I could not gather empirical data to validate the model and thus had to fall back on matching my findings with the results of previous research in Cognitive Science. The estimation of the parameters of my model were thus not possible. However, the potential values gathered in the literature allowed for smoothly running models that exhibited the expected biases and errors. On the Computer Science side, I was able to show that the overhead of the computation is minor, but the increase in realism is substantial.

5.1 OUTLOOK

This thesis has provide many departure points for further research, both for Cognitive Science as well as Computer Science. While the theoretical model used is widely established in Cognitive Science, trying to implement it in ABM has exposed several gaps in the literature. Some parameters were guesswork and others simply arbitrary. With regard to Computer Science, potential remains to expand ABM to be more human-like under an input from Cognitive Science.

5.1.1 *Follow-up Experiments*

To overcome these inaccuracies, it would be beneficial to devise a series of experiments, both in virtual reality and the real world, to get better estimates of the parameters. The distance estimation error has been widely studied, but the effect of the subdivision into different vistas points to the need for further research. Its possible experimental design would consist of multiple distance estimation tasks at a wide variety of distances, say in steps of $10m$ in the range from $10m$ to $2km$. This would allow to establish the turning point hypothesised by Daum & Hecht [130] and possibly to get a more accurate estimate of the relation of the range parameter and distance. see Eq. 3.1 and 3.2.

The more problematic parameter is the drift rate[142]. To underpin the theoretical considerations, it would be valuable to design a navigation experiment, that regularly queries the mental representation with sketch mapping and JRD tasks, to establish how the mental representation of distant location changes over time as participants move through the environment. Since the time scale of representational deterioration is unknown, the experiment would probably have to be repeated with different rates of querying.

5.1.2 *Expanding Mental Representation in ABM*

A part that was not covered throughout this thesis was the process of acquiring and forgetting knowledge about the environment which could potentially be used to reduce the computational load. This means to rely more on algorithms like D* and the like. The proposed algorithm C* is fully compatibly with such an approach as it merely requires to think of the heuristic function as something more complex then a simple distance to the destination. Incorporating D* into the MomenTUM framework[34,47] was lamentably beyond the time horizon of this thesis.

Another interesting approach would be to incorporate the Canadian Traveller Problem[169] which regards the issue when roads randomly pop in and out of existence (i. e. the snowfall temporarily closes a road). This is particularly interesting as the mechanism of probabilistically describing the existence of edges within the graph can be compared to the cognitive mechanisms of a mental representation that may lack some edges due to perception and memory issues.

Lastly, putting the cognitive routing model more rigorously to the test bench is a next step. Testing the cognitive routing in known scenarios of ABM and comparing it to real world data is thus another interesting follow-up to establish the usefulness of the approach.

Bringing together findings from Computer Science and Cognitive Science to improve ABM offers still many more venues beyond what is explored in this paper. However, it requires scientists that are well-versed in both areas to identify opportunities for cooperation and co-production.

A

APPENDIX

Listing A.1: Simulation Configuration

```xml
<?xml version="1.0" encoding="UTF-8"?>
<simulator version="2.0.0" simulationName="ExampleSimulation">

    <!-- The simulation runs 1200 seconds = 20 Minutes, and has 1200/0.05 = 24000 steps of length 0.05 second. -->
    <timeState simulationEndTime="1000.0" timeStepDuration="0.05"/>

    <!-- The parallelization is done based on 3 threads. -->
    <threadingState threads="3"/>

    <!-- We log to the console on debug level. -->
    <logging>
        <loggingState type="Console" level="Debug"/>
    </logging>

    <!-- The layout was created via the AutoCad Plugin. -->
    <!-- We use the link option of the layout scenario to load it. -->
    <!-- The content of the scenario could have been copy-pasted to into the scenario tag. -->
    <layouts>
        <!-- TODO Warning: update path -->
        <scenario id="0" layoutLink="C:\Users\Jascha\Documents\ETH\STP_Master\Master Thesis\momentum_data\100_blocks_example_01.xml"/>
    </layouts>

    <!-- We need 2 lattices. -->
    <lattices>
        <!-- Used by the generated=0 -->
        <lattice id="0" scenarioId="0" latticeType="Quadratic" cellEdgeSize="0.46"/>
        <!-- Used by the xtDensity analysis=1 -->
```

```xml
        <lattice id="1" scenarioId="0" latticeType="Quadratic" cellEdgeSize="1.0"/>
        <!-- perception lattice -->
        <lattice id="2" scenarioId="0" latticeType="Quadratic" neighborhoodType="Touching" cellEdgeSize="0.1"/>
    </lattices>

    <!-- A simple routing graph is generated for the routing model=4 -->
    <graphs>
        <graphModel name="routing" id="0">
            <graphOperation id="0" name="raw" type="RawGraph" order="0">
                <property name="graphId" type="Integer" value="0"/>
            </graphOperation>
            <graphOperation id="1" name="seeds" type="VertexCreateSeedBased"/>
            <graphOperation id="2" name="corners" type="VertexCreateAtCorners">
                <property name="cornerDistance" type="Double" value="0.96"/>
            </graphOperation>
            <graphOperation id="3" name="portal" type="VertexCreatePortal">
                <property name="cellSize" type="Double" value="2.0"/>
            </graphOperation>
            <graphOperation id="4" name="minimalRegion" type="VertexCreateMinimalRegion">
                <property name="cellSize" type="Double" value="2.00"/>
            </graphOperation>
            <graphOperation id="5" name="remove" type="VertexRemoveSimple">
                <property name="mergeDistance" type="Double" value="0.23"/>
            </graphOperation>
            <graphOperation id="6" name="visibility" type="EdgeCreateVisibilityAngleBased">
                <property name="alpha" type="Double" value="12"/>
                <property name="visibilityTolerance" type="Double" value="0.23"/>
            </graphOperation>
            <graphOperation id="7" name="removeDispensible" type="EdgeRemoveUnreachable"/>
            <graphOperation id="8" name="removeLong" type="EdgeRemoveMeanDistance"/>
```

```
        <graphOperation id="9" name="toConfiguration" type="
            ToConfiguration">
            <property name="scenarioId" type="Integer" value="0"/>
        </graphOperation>
    </graphModel>
</graphs>

<!-- <exeuctionOrder> is not needed in this simulation, because we
    apply the classical approach. -->

<!-- The standard BlockingGeometries perception model is used.. there
    is no other. -->
<perceptualModels>
    <perceptual id="0" name="BlockingGeometries" type="
        BlockingGeometries">
        <property name="perceptionDistance" type="Double" value="500.0"/
            >
        <property name="latticeIdName" type="Integer" value="2"/>
    </perceptual>
</perceptualModels>

<!-- A single operational model, because we apply the "classical"
    simulation approach. -->
<operationalModels>
    <operational id="1" name="operational" perceptualModel="0">
        <walkingReference modelId="2"/>
        <standingReference modelId="3"/>
    </operational>
</operationalModels>

<!-- A single walking model. -->
<walkingModels>
      <walking id="2" name="socialForceModel" type="SocialForce">
        <property name="relaxation_time" type="Double" value="0.5"/>
        <property name="physical_interaction_kappa" type="Double" value=
            "1.4e5"/>
        <property name="physical_interaction_k" type="Double" value="0.2
            e5"/>
        <property name="panic_degree" type="Double" value="0.0"/>
```

```
            <property name="mass_behaviour_A" type="Double" value="29.0"/>
            <property name="mass_behaviour_B" type="Double" value="0.04"/>
        </walking>
    </walkingModels>

    <!-- A single standing model -->
    <standingModels>
        <standing id="3" name="JohannsonStanding" type="JohannsonStanding">
            <property name="relaxation_time" type="Double" value="0.5" />
            <property name="physical_interaction_kappa" type="Double" value="1.4e5" />
            <property name="physical_interaction_k" type="Double" value="0.2e5" />
            <property name="mass_behaviour_A" type="Double" value="29.0"/>
            <property name="mass_behaviour_B" type="Double" value="0.04"/>
            <property name="waiting_case" type="Integer" value="1" />
            <property name="massWaitingPoint" type="Double" value="1.0" />
        </standing>
    </standingModels>

    <!-- A single tactical model. -->
    <tacticalModels>
        <tactical id="4" name="tactical" perceptualModel="0">
            <routingReference modelId="5"/>
            <stayingReference modelId="6"/>
            <searchingReference modelId="8"/>
            <property name="goalDistanceRadius" type="Double" value="0.23" />
            <property name="routeMemory" type="Boolean" value="False" />
            <property name="tacticalControl" type="Boolean" value="True" />
            <property name="deepNodeSelection" type="Integer" value="3" />
        </tactical>
    </tacticalModels>

    <!-- A single routing model. Select one of the following for testing. -->
    <routingModels>
        <!--<routing id="5" name="unifiedRouting" type="UPRM">
            <property name="randomMode" type="Boolean" value="True"/>
```

```xml
            <complexProperty name="resultMode" type="CsvMatrix" valueType="Double">
                <entry file="C:\Users\Jascha\Documents\ETH\STP_Master\Master Thesis\momentum_data\basicExampleUPRM.csv" separator=";"/>
            </complexProperty>
        </routing> -->
        <routing id="5" name="cognitive" type="Cognitive"/>
        <!--<routing id="5" name="dijkstra" type="Dijkstra"/>-->

    </routingModels>

    <!-- A single staying model. -->
    <stayingModels>
        <staying id="6" name="shiftedRandomParticipating" type="ShiftedRandomParticipating">
            <property name="participateDistance" type="Double" value="2.0"/>
            <property name="numberOfGambles" type="Integer" value="60"/>
            <property name="safetyDistance" type="Double" value="0.1"/>
            <property name="groupPositionRadius" type="Double" value="2.0"/>
        </staying>
    </stayingModels>

    <!-- A single searching model. This is a dummy model, which provides no real searching behavior!. -->
    <searchingModels>
        <searching id="8" name="noSearching" type="NoSearching"/>
    </searchingModels>

    <!-- A single destination choice (strategic model) model. -->
    <strategicalModels>
        <strategical id="9" name="odMatrx" type="ODMatrix" perceptualModel="0">
            <complexProperty name="originDestination" type="CsvMatrix" valueType="Double">
                <!-- TODO Warning: update path -->
                <entry file="C:\Users\Jascha\Documents\ETH\STP_Master\Master Thesis\momentum_data\basicExampleODMatrix.csv" separator=";"/>
            </complexProperty>
```

```xml
        <complexProperty name="behaviorType" type="List" valueType="String">
            <entry index="1" value="Staying"/>
            <entry index="2" value="Staying"/>
            <entry index="3" value="Staying"/>
           <entry index="4" value="Staying"/>
           <entry index="5" value="Staying"/>
           <entry index="6" value="Staying"/>
           <entry index="7" value="Staying"/>
        </complexProperty>
        <property name="fulfilmentOverallDuration" type="Double" value="2.0"/>
    </strategical>
</strategicalModels>

<!-- A seed concept for the generator, provides basic pedestrian data -->
<pedestrianSeeds>
    <!-- The seed is used in the generator=0 -->
    <pedestrianSeed id="0" name="basic" type="NoDistribution">
        <property name="desiredVelocity" type="Double" value="1.34"/>
        <property name="maximalVelocity" type="Double" value="2.7"/>
        <property name="radiusMeter" type="Double" value="0.23"/>
        <property name="groupSize" type="Integer" value="1"/>
    </pedestrianSeed>
</pedestrianSeeds>

<!-- A single generator -->
<generators>
    <!-- The generator is used for the scenario=0 origin=0, which are in the exampleLayout.xml file. -->
    <!-- Lattice=0 is used and pedestrian are generated randomly on the lattice=0 and origin=0. -->
    <generator id="0" name="generator" scenario="0" origin="0" seed="0" type="Plan">
        <property name="startTime" type="Double" value="0"/>
        <property name="endTime" type="Double" value="Infinity"/>
        <property name="basicHeading" type="Double" value="-90"/>
        <property name="maximalPedestrians" type="Integer" value="1"/>
```

```
        <property name="safetyDistance" type="Double" value="0.5"/>
        <geometry geometryType="Point"/>
        <complexProperty name="interval" type="List" valueType="Double">
            <entry index="0" value="0"/>
            <entry index="1" value="200.0"/>
        </complexProperty>
        <complexProperty name="percentage" type="List" valueType="Double
            ">
            <entry index="0" value="1.0"/>
            <entry index="1" value="0.0"/>
        </complexProperty>
    </generator>
    <generator id="1" name="generator" scenario="0" origin="1" seed="0"
        type="Plan">
        <property name="startTime" type="Double" value="0"/>
        <property name="endTime" type="Double" value="Infinity"/>
        <property name="basicHeading" type="Double" value="0"/>
        <property name="maximalPedestrians" type="Integer" value="1"/>
        <property name="safetyDistance" type="Double" value="0.5"/>
        <geometry geometryType="Point"/>
        <complexProperty name="interval" type="List" valueType="Double">
            <entry index="0" value="0"/>
            <entry index="1" value="1.0"/>
        </complexProperty>
        <complexProperty name="percentage" type="List" valueType="Double
            ">
            <entry index="0" value="200.0"/>
            <entry index="1" value="0.0"/>
        </complexProperty>
    </generator>
</generators>

<!-- A single absorber -->
<absorbers>
    <!-- The generator is used in the scenario=0 destination=3, which
        are in the exampleLayout.xml file. -->
    <absorber id="0" name="absorber" scenario="0" destination="6" type=
        "DestinationSelected">
        <property name="vanishTime" type="Double" value="0.2"/>
```

```
    </absorber>
    <absorber id="1" name="absorber" scenario="0" destination="7" type=
        "DestinationSelected">
        <property name="vanishTime" type="Double" value="0.2"/>
    </absorber>
</absorbers>

<!-- Two analysis models are applied, occupancy and xt-Density.  -->
<analysisModels>

    <!-- The occupancy counts the number of pedestrians performing an
        activity in intermediate locations. -->
    <analysis id="0" name="occupancyAnalysis">

        <property name="call" type="Integer" value="20"/> <!-- Every 1
            seconds -->
        <property name="analysisStartStep" type="Integer" value="0"/>
        <property name="analysisEndStep" type="Integer" value="Integer.
            MAX_VALUE"/>

        <measure type="AreaOccupancy"/>

        <writerSource sourceType="Pedestrian">
            <property name="timeStep" type="Format" value="%d"/>
            <property name="id" type="Format" value="%d"/>
            <property name="targetID" type="Format" value="%d"/>
            <property name="behavior" type="Format" value="%d"/>
            <property name="x" type="Format" value="%.2f"/>
            <property name="y" type="Format" value="%.2f"/>
        </writerSource>
    </analysis>

    <!-- The xtDensity analysis computes the pedestrian density on
        lattice=1. -->
    <!--<analysis id="1" name="xtDensityAnalysis">

        <property name="call" type="Integer" value="20"/>
        <property name="analysisStartStep" type="Integer" value="0"/>
```

```
        <property name="analysisEndStep" type="Integer" value="Integer.
            MAX_VALUE"/>

        <measure type="XtDensity">
            <property name="latticeId" type="Integer" value="1"/>
            <property name="timeRange" type="Integer" value="3"/>
            <property name="maximalDensity" type="Double" value="5.0"/>
        </measure>

        <writerSource sourceType="Pedestrian">
            <property name="timeStep" type="Format" value="%d"/>
            <property name="id" type="Format" value="%d"/>
            <property name="x" type="Format" value="%.2f"/>
            <property name="y" type="Format" value="%.2f"/>
        </writerSource>
    </analysis>-->
</analysisModels>

    <!-- We print the pedestrian data, xt-density data, occupancy data,
        the number of leaving pedestrians and the configuration layout.
        -->
<outputWriters>
    <outputWriter id="0" name="pedestrianOutputToFile">
        <property name="call" type="Integer" value="10"/> <!-- Print 10
            * 0.05 = 0.5 seconds -->
        <property name="buffer" type="Integer" value="50"/>
        <writerTarget targetType="File">
            <property name="file" type="File" value="C:\Users\Jascha\
                Documents\ETH\STP_Master\Master Thesis\momentum_data\
                output\2017_BasicExample_Pedestrian.csv"/>
            <property name="index" type="Boolean" value="True"/>
        </writerTarget>
        <writerFormat formatType="Csv">
            <property name="index" type="Boolean" value="True"/>
            <property name="delimiter" type="String" value=";"/>
        </writerFormat>
        <writerSource sourceType="Pedestrian">
            <property name="timeStep" type="Format" value="%d"/>
            <property name="id" type="Format" value="%d"/>
```

```xml
                <property name="x" type="Format" value="%.2f"/>
                <property name="y" type="Format" value="%.2f"/>
                <property name="xHeading" type="Format" value="%.2f"/>
                <property name="yHeading" type="Format" value="%.2f"/>
                <property name="behavior" type="Format" value="%d"/>
                <property name="seedID" type="Format" value="%d"/>
                <property name="currentVertexID" type="Format" value="%d"/>
            </writerSource>
        </outputWriter>

        <outputWriter id="1" name="layoutWriter">
            <property name="call" type="Integer" value="0"/> <!-- In pre-processing -->
            <writerTarget targetType="File">
                <property name="file" type="File" value="C:\Users\Jascha\Documents\ETH\STP_Master\Master Thesis\momentum_data\output\2017_BasicExample_Layout.xml"/>
            </writerTarget>
            <writerFormat formatType="Single"/>
            <writerSource sourceType="Configuration">
                <property name="dataElement" type="String" value="layouts"/>
            </writerSource>
        </outputWriter>

    </outputWriters>
</simulator>
```

BIBLIOGRAPHY

1. Castle, C. J. E. & Crooks, A. T. *Principles and concepts of agent-based modelling for developing geospatial simulations* London, UK, 2006.
2. Cioffi-Revilla, C. in *Introduction to Computational Social Science: Principles and Applications* (ed Cioffi-Revilla, C.) 2nd ed., 1–33 (Springer International Publishing, Cham, 2017). ISBN: 978-3-319-50131-4. doi:10.1007/978-3-319-50131-4_1.
3. Duong, T., Goud, B. & Schauer, K. Closed-form density-based framework for automatic detection of cellular morphology changes. *Proceedings of the National Academy of Sciences* **109,** 8382–8387 (2012).
4. Klatzky, R. L. *et al.* Acquisition of route and survey knowledge in the absence of vision. *Journal of motor behavior* **22,** 19–43 (1990).
5. Von Neumann, J. The general and logical theory of automata. *Cerebral mechanisms in behavior* **1,** 1–2 (1951).
6. Conway, J. The game of life. *Scientific American* **223,** 4 (1970).
7. Schelling, T. C. Models of segregation. *The American Economic Review* **59,** 488–493 (1969).
8. Schelling, T. C. Dynamic models of segregation. *Journal of mathematical sociology* **1,** 143–186 (1971).
9. An, L. Modeling human decisions in coupled human and natural systems: review of agent-based models. *Ecological Modelling* **229,** 25–36 (2012).
10. Epstein, J. M. & Axtell, R. *Growing artificial societies: social science from the bottom up* (Brookings Institution Press, 1996).
11. Cohen, M. D. & Axelrod, R. *Harnessing Complexity: Organizational Implications of a Scientific Frontier* (Simon and Schuster, 2000).
12. Cook, T. D., Campbell, D. T. & Day, A. *Quasi-experimentation: Design & analysis issues for field settings* (Houghton Mifflin Boston, 1979).

13. Davis, J. P., Eisenhardt, K. M. & Bingham, C. B. Developing theory through simulation methods. *Academy of Management Review* **32,** 480–499 (2007).
14. Whetten, D. A. What constitutes a theoretical contribution? *Academy of management review* **14,** 490–495 (1989).
15. Sutton, R. I. & Staw, B. M. What theory is not. *Administrative science quarterly,* 371–384 (1995).
16. Batty, M. *Agent-based pedestrian modeling* (SAGE Publications, London, UK, 2001).
17. Batty, M. *The new science of cities* (Mit Press, 2013).
18. Aguilar, L., Lalith, M., Hori, M., Ichimura, T. & Tanaka, S. *A scalable workbench for large urban area simulations, comprised of resources for behavioural models, interactions and dynamic environments* in *International Conference on Principles and Practice of Multi-Agent Systems* (2014), 166–181.
19. Aguilar, L., Lalith, M. & Hori, M. in *Multi-agent Systems* (InTech, 2017).
20. Crooks, A. T. Exploring cities using agent-based models and GIS (2006).
21. Malleson, N., Heppenstall, A. & See, L. Crime reduction through simulation: An agent-based model of burglary. *Computers, environment and urban systems* **34,** 236–250 (2010).
22. Chen, X. & Zhan, F. B. Agent-based modelling and simulation of urban evacuation: relative effectiveness of simultaneous and staged evacuation strategies. *Journal of the Operational Research Society* **59,** 25–33 (2008).
23. Thrash, T., Hoffman, M., Kapadia, M., Hoelscher, C. & Schinazi, V. R. *Investigations of collective search behavior using virtual reality and agent-based simulations* in *Lecture series at the Future Cities Laboratory* (2017).
24. Kuipers, B. Modeling spatial knowledge. *Cognitive science* **2,** 129–153 (1978).
25. Smith, T. R., Pellegrino, J. W. & Golledge, R. G. Computational process modeling of spatial cognition and behavior. *Geographical Analysis* **14,** 305–325 (1982).
26. McDermott, D. & Davis, E. Planning routes through uncertain territory. *Artificial intelligence* **22,** 107–156 (1984).
27. Gopal, S., Klatzky, R. L. & Smith, T. R. Navigator: A psychologically based model of environmental learning through navigation. *Journal of environmental Psychology* **9,** 309–331 (1989).

28. Leiser, D. & Zilbershatz, A. The traveller: A computational model of spatial network learning. *Environment and behavior* **21,** 435–463 (1989).

29. Raubal, M. & Worboys, M. A formal model of the process of wayfinding in built environments. *Spatial information theory. Cognitive and computational foundations of geographic information science,* 748 (1999).

30. Seyfried, A., Steffen, B., Klingsch, W. & Boltes, M. The fundamental diagram of pedestrian movement revisited. *Journal of Statistical Mechanics: Theory and Experiment* **2005,** P10002 (2005).

31. Seyfried, A., Steffen, B., Klingsch, W., Lippert, T. & Boltes, M. *The fundamental diagram of pedestrian movement revisited empirical results and modelling* in *Traffic and Granular Flow* **5** (2007), 305–314.

32. Helbing, D. & Molnar, P. Social force model for pedestrian dynamics. *Physical review E* **51,** 4282 (1995).

33. Jonietz, D. & Kiefer, P. *Uncertainty in Wayfinding: A Conceptual Framework and Agent-Based Model* in *13th International Conference on Spatial Information Theory (COSIT 2017)* **86** (2017), 15.

34. Kielar, P. M., Biedermann, D. H. & Borrmann, A. MomenTUMv2: a modular, extensible, and generic agent-based pedestrian behavior simulation framework. *TUM-I1643* (2016).

35. Newell, A., Simon, H. A. *et al. Human problem solving* **9** (Prentice-Hall Englewood Cliffs, NJ, 1972).

36. Anderson, J. R. Problem solving and learning. *American Psychologist* **48,** 35 (1993).

37. Anderson, J. R. *Cognitive psychology and its implications* 7th ed. (Worth Publishing, New York, NY, USA, 2010).

38. Wooldridge, M. *An introduction to multiagent systems* (John Wiley & Sons, 2009).

39. Kielar, P. M. & Borrmann, A. *Spice: A Cognitive Agent Architecture for Computational Crowd Simulations in Complex Environments* 2017. `https://www.cms.bgu.tum.de/en/team/peter-kielar`.

40. Witmer, B. G., Bailey, J. H., Knerr, B. W. & Parsons, K. C. Virtual spaces and real world places: transfer of route knowledge. *International Journal of Human-Computer Studies* **45,** 413–428 (1996).

41. Allen, G. L., Siegel, A. W. & Rosinski, R. R. The role of perceptual context in structuring spatial knowledge. *Journal of Experimental Psychology: human learning and memory* **4,** 617 (1978).

42. Werner, S., Krieg-Brückner, B., Mallot, H. A., Schweizer, K. & Freksa, C. in *Informatik'97 Informatik als Innovationsmotor* 41–50 (Springer, 1997).

43. Montello, D. R. A new framework for understanding the acquisition of spatial knowledge in large-scale environments. *Spatial and temporal reasoning in geographic information systems,* 143–154 (1998).

44. Ishikawa, T. & Montello, D. R. Spatial knowledge acquisition from direct experience in the environment: Individual differences in the development of metric knowledge and the integration of separately learned places. *Cognitive psychology* **52,** 93–129 (2006).

45. Wang, L., Mou, W. & Sun, X. Development of landmark knowledge at decision points. *Spatial Cognition & Computation* **14,** 1–17 (2014).

46. Heliövaara, S., Korhonen, T., Hostikka, S. & Ehtamo, H. Counterflow model for agent-based simulation of crowd dynamics. *Building and Environment* **48,** 89–100 (2012).

47. Kielar, P. M. *Kognitive Modellierung und Computergestützte Simulation der Räumlich-Sequenziellen Zielauswahl von Fußgängern* Doctoral dissertation (Technische Universität München, 2017).

48. Shao, W. & Terzopoulos, D. *Autonomous pedestrians* in *Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation* (2005), 19–28.

49. Helbing, D. A Fluid-Dynamic Model for the Movement of Pedestrians. *Complex Systems* **6,** 391–415 (1992).

50. Helbing, D. A mathematical model for the behavior of individuals in a social field. *Journal of Mathematical Sociology* **19,** 189–219 (1994).

51. Henein, C. M. & White, T. *Agent-based modelling of forces in crowds* in *International Workshop on Multi-Agent Systems and Agent-Based Simulation* (2004), 173–184.

52. Werner, S., Krieg-Brückner, B. & Herrmann, T. in *Spatial cognition II* 295–316 (Springer, 2000).

53. Lynch, K. *The image of the city* (MIT press, 1960).

54. Schelhorn, T., O'Sullivan, D., Haklay, M. & Thurstain-Goodwin, M. STREETS: An agent-based pedestrian model (1999).

55. Klügl, F. & Rindsfüser, G. *Large-scale agent-based pedestrian simulation* in *German conference on multiagent system technologies* (2007), 145–156.

56. Wise, S. & Crooks, A. T. Agent-based modeling for community resource management: Acequia-based agriculture. *Computers, Environment and Urban Systems* **36,** 562–572 (2012).

57. Wise, S. *Using social media content to inform agent-based models for humanitarian crisis response* Doctoral dissertation (George Mason University, 2014).

58. Crooks, A. T. & Heppenstall, A. J. in *Agent-based models of geographical systems* 85–105 (Springer, 2012).

59. Johansson, A. & Kretz, T. in *Agent-based models of geographical systems* 451–462 (Springer, 2012).

60. Deo, N. & Pang, C.-Y. Shortest-path algorithms: Taxonomy and annotation. *Networks* **14,** 275–323 (1984).

61. Gibbons, A. *Algorithmic graph theory* (Cambridge university press, 1985).

62. Skiena, S. Dijkstra's algorithm. *Implementing Discrete Mathematics: Combinatorics and Graph Theory with Mathematica,* 225–227 (1990).

63. West, D. B. *Introduction to graph theory* (Prentice Hall, Upper Saddle River, 2001).

64. Kielar, P. M. & Borrmann, A. Modeling pedestrians' interest in locations: A concept to improve simulations of pedestrian destination choice. *Simulation Modelling Practice and Theory* **61,** 47–62. ISSN: 1569190X (2016).

65. Gramann, K., Müller, H. J., Eick, E.-M. & Schönebeck, B. Evidence of separable spatial representations in a virtual navigation task. *Journal of Experimental Psychology: Human Perception and Performance* **31,** 1199 (2005).

66. Thorndyke, P. W. & Hayes-Roth, B. Differences in spatial knowledge acquired from maps and navigation. *Cognitive psychology* **14,** 560–589 (1982).

67. Tomko, M. & Richter, K.-F. *Defensive wayfinding: Incongruent information in route following* in *International Workshop on Spatial Information Theory* (2015), 426–446.

68. Kielar, P. M., Biedermann, D. H., Kneidl, A. & Borrmann, A. A unified pedestrian routing model for graph-based wayfinding built on cognitive principles. *Transportmetrica A: Transport Science,* 1–27 (2017).

69. Golledge, R. G. *Path selection and route preference in human navigation: A progress report* in *International Conference on Spatial Information Theory* (1995), 207–222.

70. Dalton, R. C. The secret is to follow your nose: Route path selection and angularity. *Environment and Behavior* **35,** 107–131 (2003).

71. Hölscher, C., Tenbrink, T. & Wiener, J. M. Would you follow your own route description? Cognitive strategies in urban route planning. *Cognition* **121,** 228–247 (2011).

72. Conroy, R. A. *Spatial navigation in immersive virtual environments* Doctoral dissertation (Citeseer, 2001).

73. Schultheis, H. & Barkowsky, T. Casimir: an architecture for mental spatial knowledge processing. *Topics in cognitive science* **3,** 778–795 (2011).

74. Dijkstra, E. W. A note on two problems in connexion with graphs. *Numerische mathematik* **1,** 269–271 (1959).

75. Koenig, S., Likhachev, M., Liu, Y. & Furcy, D. Incremental heuristic search in AI. *AI Magazine* **25,** 99 (2004).

76. Korte, B., Vygen, J., Korte, B. & Vygen, J. *Combinatorial optimization* 4th ed. (Springer, 2012).

77. Berge, C. La theorie des graphes. *Paris, France* (1958).

78. Ford Jr, L. R. *Network flow theory* tech. rep. (Rand Corporation, Santa Monica, CA, USA, 1956).

79. Fredman, M. L. & Tarjan, R. E. Fibonacci heaps and their uses in improved network optimization algorithms. *Journal of the ACM (JACM)* **34,** 596–615 (1987).

80. Hart, P. E., Nilsson, N. J. & Raphael, B. A formal basis for the heuristic determination of minimum cost paths. *IEEE transactions on Systems Science and Cybernetics* **4,** 100–107 (1968).

81. Hart, P. E., Nilsson, N. J. & Raphael, B. Correction to a formal basis for the heuristic determination of minimum cost paths. *ACM SIGART Bulletin,* 28–29 (1972).

82. Pearl, J. Heuristics: intelligent search strategies for computer problem solving (1984).

83. Russell, S., Norvig, P. & Intelligence, A. *A modern approach to Artificial Intelligence* 25–27 (Prentice-Hall, Egnlewood Cliffs, 1995).

84. Korf, R. E. Depth-first iterative-deepening: An optimal admissible tree search. *Artificial intelligence* **27,** 97–109 (1985).

85. Björnsson, Y., Enzenberger, M., Holte, R. C. & Schaeffer, J. Fringe Search: Beating A* at Pathfinding on Game Maps. *CIG* **5,** 125–132 (2005).

86. Russell, S. J. *Efficient Memory-Bounded Search Methods.* in *ECAI* **92** (1992), 1–5.

87. Korf, R. E. Linear-space best-first search. *Artificial Intelligence* **62,** 41–78 (1993).

88. Koenig, S., Likhachev, M. & Furcy, D. Lifelong planning A*. *Artificial Intelligence* **155,** 93–146 (2004).

89. Stentz, A. *Optimal and efficient path planning for partially-known environments* in *Robotics and Automation, 1994. Proceedings., 1994 IEEE International Conference on* (1994), 3310–3317.

90. Stentz, A. *et al. The focussed D* algorithm for real-time replanning* in *IJCAI* **95** (1995), 1652–1659.

91. Koenig, S. & Likhachev, M. Fast replanning for navigation in unknown terrain. *IEEE Transactions on Robotics* **21,** 354–363 (2005).

92. Shimbel, A. *Structure in communication nets* in *Proceedings of the symposium on information networks* **4** (1954).

93. Bellman, R. On a routing problem. *Quarterly of applied mathematics* **16,** 87–90 (1958).

94. Moore, E. F. *The shortest path through a maze* in *Proc. Int. Symp. Switching Theory, 1959* (1959), 285–292.

95. Ramalingam, G. & Reps, T. An incremental algorithm for a generalization of the shortest-path problem. *Journal of Algorithms* **21,** 267–305 (1996).

96. Kleene, S. C. *Representation of events in nerve nets and finite automata* tech. rep. (Rand Corporation, Santa Monica, CA, USA, 1951).

97. Roy, B. Transitivité et connexité. *Comptes Rendus Hebdomadaires Des Seances De L Academie Des Sciences* **249,** 216–218 (1959).

98. Floyd, R. W. Algorithm 97: shortest path. *Communications of the ACM* **5,** 345 (1962).

99. Ingerman, P. Z. Path matrix, Algorithm 141. *Comm. ACM* **5,** 556 (1962).

100. Warshall, S. A theorem on boolean matrices. *Journal of the ACM (JACM)* **9,** 11–12 (1962).

101. Cahn, R. S. *Wide area network design: concepts and tools for optimization* (Morgan Kaufmann, 1998).

102. Thagard, P. in *The Standford Encyclopedia of Philosophy* (ed Zalta, E. N.) (Stanford University, Stanford, CA, 2008). `https://plato.stanford.edu/archives/fall2008/entries/cognitive-science/`.

103. Denis, M. & Loomis, J. M. Perspectives on human spatial cognition: memory, navigation, and environmental learning. *Psychological Research* **71,** 235–239 (2007).

104. Siegel, A. W. & White, S. H. The development of spatial representations of large-scale environments. *Advances in child development and behavior* **10,** 9–55 (1975).

105. Thrash, T., Zhao, H., Duran, A., Frese, L. & Schinazi, V. R. *Evaluation of a computational framework for cognitive maps.* Tübingen, Germany, 2017.

106. Chrastil, E. R. & Warren, W. H. From cognitive maps to cognitive graphs. *PloS one* **9,** e112544 (2014).

107. Golledge, R. G. Human wayfinding and cognitive maps. *Wayfinding behavior: Cognitive mapping and other spatial processes,* 5–45 (1999).

108. Gärling, T., Lindberg, E., Carreiras, M. & Anders, B. Reference systems in cognitive maps. *Journal of Environmental Psychology* **6,** 1–18 (1986).

109. Golledge, R. G. Place recognition and wayfinding: Making sense of space. *Geoforum* **23,** 199–214 (1992).

110. Herman, J. F. & Siegel, A. W. The development of cognitive mapping of the large-scale environment. *Journal of Experimental Child Psychology* **26,** 389–406 (1978).

111. Loomis, J. M. & Knapp, J. M. Visual perception of egocentric distance in real and virtual environments. *Virtual and adaptive environments* **11,** 21–46 (2003).

112. Briggs, R. *Cognitive distance in urban space.* Doctoral dissertation (The Ohio State University, 1972).

113. Hayashi, T., Fujii, H. & Inui, T. *Modeling the cognitive map formation process based on psychological experiments using a computer graphics system* in *Systems, Man and Cybernetics, 1990. Conference Proceedings., IEEE International Conference on* (1990), 826–828.

114. Nguyen, T. D. *Estimating distances and traveled distances in virtual and real environments* Doctoral dissertation (University of Iowa, 2011).

115. Gilinsky, A. S. Perceived size and distance in visual space. *Psychological review* **58,** 460 (1951).

116. Gibson, E. J. & Bergman, R. The effect of training on absolute estimation of distance over the ground. *Journal of Experimental Psychology* **48,** 473 (1954).

117. Harway, N. I. Judgment of distance in children and adults. *Journal of experimental psychology* **65,** 385 (1963).

118. Lappin, J. S., Shelton, A. L. & Rieser, J. J. Environmental context influences visually perceived distance. *Attention, Perception, & Psychophysics* **68,** 571–581 (2006).

119. Witt, J. K., Stefanucci, J. K., Riener, C. R. & Proffitt, D. R. Seeing beyond the target: Environmental context affects distance perception. *Perception* **36,** 1752–1768 (2007).

120. Wohlwill, J. F. The development of "overconstancy" in space perception. *Advances in child development and behavior* **1,** 265–312 (1964).

121. Stevens, S. S. On the psychophysical law. *Psychological review* **64,** 153 (1957).

122. Baird, J. C. & Biersdorf, W. R. Quantitative functions for size and distance judgments. *Attention, Perception, & Psychophysics* **2,** 161–166 (1967).

123. Da Silva, J. A. Scales for perceived egocentric distance in a large open field: Comparison of three psychophysical methods. *The American Journal of Psychology,* 119–144 (1985).

124. Cook, M. The judgment of distance on a plane surface. *Attention, Perception, & Psychophysics* **23,** 85–90 (1978).

125. Teghtsoonian, R. & Teghtsoonian, M. Scaling apparent distance in a natural outdoor setting. *Psychonomic Science* **21,** 215–216 (1970).

126. Flückiger, M. La perception d'objects lointains. *La perception de l'environnement,* 221–238 (1991).

127. Montello, D. R. *Scale and multiple psychologies of space* in *European conference on spatial information theory* (1993), 312–321.

128. Grüsser, O.-J. in *Spatially oriented behavior* 327–352 (Springer, 1983).

129. Cutting, J. E. & Vishton, P. M. Potency, and contextual use of different information about depth. *Perception of space and motion* **69** (1995).

130. Daum, S. O. & Hecht, H. Distance estimation in vista space. *Attention, Perception, & Psychophysics* **71**, 1127–1137 (2009).

131. Loomis, J. M., Da Silva, J. A., Fujita, N. & Fukusima, S. S. Visual space perception and visually directed action. *Journal of Experimental Psychology: Human Perception and Performance* **18**, 906 (1992).

132. Worchel, P. Space perception and orientation in the blind. *Psychological monographs: general and applied* **65**, i (1951).

133. Schinazi, V. R., Nardi, D., Newcombe, N. S., Shipley, T. F. & Epstein, R. A. Hippocampal size predicts rapid learning of a cognitive map in humans. *Hippocampus* **23**, 515–528 (2013).

134. Kraemer, D. J. M. *et al.* Verbalizing, visualizing, and navigating: The effect of strategies on encoding a large-scale virtual environment. *Journal of Experimental Psychology: Learning, Memory, and Cognition* **43**, 611–621 (Apr. 2017).

135. Siegel, A. W. Externalization of cognitive maps by children and adults: In search of ways to ask better questions. *Spatial representation and behavior across the life span: theory and application/edited by LS Liben, AH Patterson, N. Newcombe* (1981).

136. Moar, I. & Carleton, L. R. Memory for routes. *The Quarterly Journal of Experimental Psychology* **34**, 381–394 (1982).

137. Mardia, K. V. & Jupp, P. E. *Directional statistics* (John Wiley & Sons, 2009).

138. Abramowitz, M., Stegun, I. A. *et al. Handbook of mathematical functions with formulas, graphs, and mathematical tables* (Dover, New York, 1972).

139. Watson, G. N. *A treatise on the theory of Bessel functions* (Cambridge university press, 1995).

140. Hochmair, H. & Frank, A. U. Influence of estimation errors on wayfinding-decisions in unknown street networks–analyzing the least-angle strategy. *Spatial Cognition and Computation* **2**, 283–313 (2000).

141. Davis, E. *Representations of commonsense knowledge* (Morgan Kaufmann, 2014).

142. Huttenlocher, J., Hedges, L. V. & Duncan, S. Categories and particulars: Prototype effects in estimating spatial location. *Psychological review* **98,** 352 (1991).

143. Crooks, A., Castle, C. & Batty, M. Key challenges in agent-based modelling for geo-spatial simulation. *Computers, Environment and Urban Systems* **32,** 417–430 (2008).

144. Anderson, J. R., Matessa, M. & Lebiere, C. ACT-R: A theory of higher level cognition and its relation to visual attention. *Human-Computer Interaction* **12,** 439–462 (1997).

145. Taatgen, N. A., Lebiere, C. & Anderson, J. R. Modeling paradigms in ACT-R. *Cognition and multi-agent interaction: From cognitive modeling to social simulation,* 29–52 (2006).

146. Hoogendoorn, S. P., Bovy, P. H. L. & Daamen, W. Microscopic pedestrian wayfinding and dynamics modeling. *1st International Conference on Pedestrian and Evacuation Dynamics,* 124–154 (2001).

147. Press, W. H. *Numerical recipes 3rd edition: The art of scientific computing* (Cambridge university press, 2007).

148. Vogelaar, M. *Evaluate Bessel function J, Y, I, K of integer order* 1998. `https://www.astro.rug.nl/%7B~%7Dgipsy/sub/bessel.c` (2017).

149. Bresenham, J. E. Algorithm for computer control of a digital plotter. *IBM Systems journal* **4,** 25–30 (1965).

150. Helbing, D., Farkas, I. & Vicsek, T. Simulating dynamical features of escape panic. *Nature* **407,** 487 (2000).

151. Köster, G., Treml, F. & Gödel, M. Avoiding numerical pitfalls in social force models. *Physical Review E* **87,** 63305 (2013).

152. Johansson, F., Peterson, A. & Tapani, A. Waiting pedestrians in the social force model. *Physica A: Statistical Mechanics and its Applications* **419,** 95–107 (2015).

153. Kielar, P. M. & Borrmann, A. *Coupling Spatial Task Solving Models to Simulate Complex Pedestrian Behavior Patterns Kongress-/Buchtitel: Proc. of the 8th Conference on Pedestrian and Evacuation Dynamics* in *Proc. of the 8th Conference on Pedestrian and Evacuation Dynamics* (2016).

154. Kielar, P. *Github MomenTUM Issue 10 Response* 2017. https://github.com/tumcms/MomenTUM/issues/10%7B%5C#%7Dissuecomment-350216597 (2018).

155. Fischer, F. *Latex Overlay Generator* 2016. https://ff.cx/latex-overlay-generator/%7B%5C#%7D/v0.0.1 (2018).

156. Grübel, J. Random ODMatrix Generation. https://github.com/tumcms/MomenTUM/blob/master/momentum-documentation/UserGuide/Explanations/BehaviorModelExamples/StrategicBehaviorModelExamples/generate%7B%5C_%7Drandom%7B%5C_%7Dod%7B%5C_%7Dmatrix.Rmd (2017).

157. Simonoff, J. S. *Smoothing methods in statistics* (Springer Science & Business Media, 2012).

158. Schauer, K. *et al.* Probabilistic density maps to study global endomembrane organization. *Nature methods* **7**, 560–566 (2010).

159. Anderson, N. H., Hall, P. & Titterington, D. M. Two-sample test statistics for measuring discrepancies between two multivariate probability density functions using kernel-based density estimates. *Journal of Multivariate Analysis* **50**, 41–54 (1994).

160. Lee, J.-S. *et al.* The complexities of agent-based modeling output analysis. *Journal of Artificial Societies and Social Simulation* **18**, 4 (2015).

161. Duong, T. *ks: Kernel Smoothing* (2017). https://cran.r-project.org/package=ks.

162. Wand, M. P. & Jones, M. C. *Kernel smoothing* (Crc Press, 1994).

163. Chacón, J. E. & Duong, T. Multivariate plug-in bandwidth selection with unconstrained pilot bandwidth matrices. *Test* **19**, 375–398 (2010).

164. Aggarwal, C. C., Hinneburg, A. & Keim, D. A. *On the surprising behavior of distance metrics in high dimensional spaces* in *ICDT* **1** (2001), 420–434.

165. Gonzalez, E., Alarcon, M., Aristizabal, P. & Parra, C. *BSA: A coverage algorithm* in *Intelligent Robots and Systems, 2003.(IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on* **2** (2003), 1679–1684.

166. Kanehara, M., Kagami, S., Kuffner, J. J., Thompson, S. & Mizoguhi, H. *Path shortening and smoothing of grid-based path planning with consideration of obstacles* in *Systems, Man and Cybernetics, 2007. ISIC. IEEE International Conference on* (2007), 991–996.

167. Wiesmann, H. J. & Zeller, H. R. A fractal model of dielectric breakdown and prebreakdown in solid dielectrics. *Journal of Applied Physics* **60,** 1770–1773 (1986).

168. Rosenfeld, A. Digital topology. *American Mathematical Monthly,* 621–630 (1979).

169. Papadimitriou, C. H. & Yannakakis, M. Shortest paths without a map. *Theoretical Computer Science* **84,** 127–150 (1991).

CURRICULUM VITAE

PERSONAL DATA

Name	Jascha Grübel
Date of Birth	March 5, 1990
Place of Birth	Engen, Germany
Citizen of	Germany

TERTIARY EDUCATION

09/2016 – 02/2018	Eidgenössisches Technische Hochschule, Zürich, Switzerland *Final degree:* Master of Science in Science, Technology, and Policy
06/2017 – 08/2017	University of Michigan, Ann Arbor, Michigan, USA ICPSR Summer Program in Quantitative Methods in Social Research *Final degree:* EITM Certification
09/2014 – 04/2016	Eidgenössisches Technische Hochschule, Zürich, Switzerland *Final degree:* Master of Science in Computer Science with a Specialisation in Visual Computing
09/2010 – 09/2014	Eidgenössisches Technische Hochschule, Zürich, Switzerland *Final degree:* Bachelor of Science in Computer Science

EMPLOYMENT

07/2013 – present	Technical Research Assistant *Cognitive Science Research Group, Eidgenössische Technische Hochschule Zürich*, Zürich, Switzerland
07/2017 – 02/2018	Technical Research Assistant *Département de Science Politique et Relations Internationales, University of Geneva*, Geneva, Switzerland
04/2016 – 06/2016	Associate Staff at CASA *Center for Advanced Spatial Analysis, University College London*, London, United Kingdom
04/2015 – 09/2015	Computer Graphics Internship *Esri R&D Center-Zurich*, Zürich, Switzerland
02/2012 – 02/2015	Teaching Assistant *Eidgenössische Technische Hochschule Zürich*, Zürich, Switzerland
07/2014 – 09/2014	Internship via eth juniors *Swisslog*, Buchs AG, Switzerland

PUBLICATIONS

Articles in peer-reviewed journals:

1. Grübel, J., Thrash, T., Hölscher, C. & Schinazi, V. Evaluation of a conceptual framework for predicting navigation performance in virtual reality. *PLoS ONE* **12.** ISSN: 19326203. doi:10.1371/journal.pone.0184682. http://dx.plos.org/10.1371/journal.pone.0184682 (2017).

Conference contributions:

2. Grübel, J. *et al.* in *Spatial Cognition X* (eds Barkowsky, T., Burte, H., Hölscher, C. & Schultheis, H.) 1st ed., 159–176 (Springer International Publishing, 2017). ISBN: 9783319681887. doi:10.1007/978-3-319-68189-4_10.