Route navigation inspired by foraging insects: Following and finding a route again.

Doctoral Dissertation
by

Olivier J.N. Bertrand

Fakultät für Biologie
Universität Bielefeld

Route navigation inspired by foraging insects: Following and finding a route again.

Dissertation
zur Entlangung des akademischen Grades
Doktor der Naturwissenschaften
- Dr. rer. nat. -
der Fakultät für Biologie der Universität Bielefeld

vorgelegt von
Olivier J.N. Bertrand
Bielefeld, Oktober 2016

betreut von
Prof. Dr. Martin Egelhaaf
Dr. Jens-Peter Lindemann

Printed on non-ageing paper ISO 9706

Erklärung

Ich versichere, dass ich diese Arbeit selbständig und ohne unzulässige Hilfe verfasst habe, keine anderen als die angegebenen Quellen und Hilfsmittel benutzt und Zitate kenntlich gemacht habe.

____

Olivier J.N. Bertrand

Bielefeld, Oktober 2016

Contents

List of Figures

List of Tables

Acknowledgements

I would like to thank my supervisors, Martin Egelhaaf and Jens Lindemann. Without them I may not have discovered the field of neuroethology and asked myself how an insect can navigate. I have too many colleagues to thank. I am amazed by the cohesion we had and hopefully will still have among PhD students. I want to thank them not only as co-workers for their helpful comments and discussions, but also as friends for the many wonderful evenings we spent together. I want to thank especially Diana Rien, the former PhD student who helped me settle in Bielefeld. I had the chance to supervise two clever students Moritz Milde and Martin Müller.

Soon after my arrival in Bielefeld, I started to learn how to dance Forrò, thanks to Sarah Hessler. Dancing was on my list of dreams for many years, and this dream became a reality in Bielefeld. Sarah, a joyful and dynamic friend, taught me how to dance, and soon after I started, I became her teaching assistant. I lack the words to describe how amazing the Forrò group is. Thanks to this group (which grew to be large) I met many valuable people.

The distance from my home town triggered me to reconnect and personally exchange with two family members, my godfather Noël

Chatelus (who sadly passed away) and my uncle Daniel Bretnacher. Noël was for me a formidable researcher. He was, even after having retired, solving mathematical or mechanical problems, and was reading scientific news. At the same time, he kept his mind open to the world by singing in the choir, walking, etc. He reminded me many times to not lose contact to art, music and other non-scientific aspects of my life. It may be the most valuable advice I had. Daniel was a teacher, teaching mathematics to college students. I learned with him how to listen and accept that two people can be correct at the same time. He taught me a lot of advice on how to teach. Sadly, I had not enough time to discuss with his partner who sadly passed away. He was a philosopher and suggested reading two books to me which opened my mind.

I want to thank my close family deeply, my parents (Dominique and Michel), my sister (Claire), my brother (Jean), my adoptive grandmother (Thérèse Brun) and my godmother (Jocelyne Dorme). They have always been there whenever and wherever necessary. Jocelyne was often travelling when I was a child, and she probably gave me the strength to go to Bielefeld. Thérèse is someone with whom I loved to discuss. She knows how to listen, and senses when the discussion starts to be based on emotion and lacks rational argument. Jean is an amazing brother and I lack the words to describe all the valuable moments we had together. Claire, my older sister, showed me the way in life. I followed her steps until high school. Finally, I asked and still ask a lot from my parents. They supported my crazy pseudo-scientific experimentation turning my room in a workshop, my studies in Lyon, in Paris, in Santa Barbara (US), and helped me move to Bielefeld, etc. I know I can count on their support.

Abstract

Navigation is one of the most complex behaviours observed in the animal kingdom. A navigating animal needs to learn and recognize the characteristics at certain locations, to decide in which direction to move to reach its destination and to avoid collisions with objects during its journey. Many insects – bees, ants, and wasps – are fascinating navigators, and their behaviour has been scrutinized in great detail over the past century. With their brain weighing only a few milligrams, these insects have been an amazing source of inspiration for engineers to develop computationally parsimonious and energy-efficient algorithms, and puzzled scientists about how such a tiny animal can navigate efficiently in a complex world.

The thesis has been inspired by the stunning navigational skills of foraging insects. One of their skills is the ability to follow a habitual route between two locations. As it will be shown in the thesis, route navigation can arise from simple mechanisms; knowing its overall goal direction and employing a collision avoidance algorithm is sufficient to follow a route. However, the journey along a route of an agent, i.e. a biological or technical system, is not always smooth. The journey may be disrupted suddenly by external factors – such as wind or an impending danger – or by internal sources that lead to navigational errors. The agent will, thus, be at an unknown location away from its habitual route and have to find its route again to complete its journey. I will reveal in the thesis a variety of search strategies that an agent may use to find its route again in a cluttered environment, such as a city or a forest. Since a unique optimal search strategy does not exist, it will be shown that the agent can decide which strategy to follow, assuming it can estimate the distance it plans to travel and the distance it has been displaced from its route.

The thesis addresses fundamental questions of navigation by focussing on following and finding a habitual route again. To frame these dilemmas, (1) an overview of navigation will also be given to highlight common fundamental problems faced by any navigating agent, (2) the various degrees of complexity of different strategies to solve navigational tasks and (3) essential aspects of research on insect navigation. Although my modelling approach is inspired by the behaviour of foraging insects, it aims to provide general solutions for any moving agent on how to commute between two locations efficiently.

Zusammenfassung

Navigation ist eine der komplexesten beobachteten Verhaltensweisen im Tierreich: um sein Ziel zu erreichen, muss ein navigierendes Tier lernen, die Eigenschaften eines bestimmten Ortes zu erkennen, um eine Entscheidung über seine Bewegungsrichtung treffen zu können. Zudem muss es auch Kollisionen mit Objekten vermeiden. Viele Insekten, bspw. Bienen, Ameisen, und Wespen, sind hervorragende Navigatoren, deren Verhalten im vergangenen Jahrhundert von WissenschaftlerInnen eingehend untersucht wurde. Aufgrund ihrer Fähigkeit trotz ihres kleinen Gehirns, das nur wenige Milligramm wiegt, effizient zu navigieren, sind diese Insekten eine Quelle der Inspiration für WissenschaftlerInnen zur Entwicklung energieeffizienter Algorithmen.

Die vorliegende Dissertation wurde von der beeindruckenden Navigationsfähigkeit nahrungssuchender Insekten inspiriert. Sie haben die Fähigkeit einer bekannten Route zwischen zwei Standorten zu folgen. Wie in der Dissertation gezeigt, ist es bereits mit sehr einfachen Mechanismen möglich, einer Route zu folgen. Beispielsweise ist es ausreichend, die Richtung des Gesamtzieles zu kennen und einen Kollisionsvermeidungs-Algorithmus anzuwenden, um einer Route zu folgen. Dennoch verläuft die Bewegung des Agenten, das heißt eines technischen oder biologischen Systems, entlang einer Route nicht immer problemlos. Sie kann durch äußere Faktoren - wie z.B. Wind - oder interne Zustände, die zu Navigationsfehlern führen, gestört werden. Der Agent befindet sich dann an einem unbekannten Ort abseits des gewohnten Weges und muss zu seiner Route zurück finden, um seine Reise fortzusetzen. Durch die Verhaltenssimulation des Agenten wird in der Dissertation gezeigt, dass verschiedene Suchstrategien benutzt werden können, um in einer Umwelt mit vielen Objekten, z.B. einem Wald oder einer Stadt, zu seiner Route zurückzufinden. Um seine Navigationsfähigkeiten zu optimieren, kann ein Agent eine Strategie auswählen. Hierbei sind die Entfernung des Agenten von seiner Route und die zur Suche verfügbare Zeit wichtige Faktoren.

Diese Doktorarbeit befasst sich mit den grundsätzlichen Fragen der Navigation und insbesondere mit Navigation nach einem Abkommen von der Route. Um dieses Phänomen zu erklären, wird zunächst ein Überblick über Navigation gegeben, indem (I) das grundlegende Problem eines jede Navigationssystems, (II) die verschiedenen Grade der Komplexität des Navigationsverhaltens, und (III) wesentliche Aspekte der Forschung an Insekten-Navigation präsentiert werden. Obwohl mein Modellierungsansatz vom Verhalten nahrungssuchender Insekten inspiriert ist, zielt er darauf ab, allgemeine und effiziente Lösungen für - zwischen zwei Standorten pendelnde - Navigationssysteme zu entwerfen.

Introduction

The success of a species depends on the ability of the individual animals to protect, find resources, mate, and feed themselves and their offspring. Under real-world conditions, it is unlikely to find a location that provides the animal with resources and protection at the same time. Bees, for example, build their nest at an appropriate place to breed and protect their offspring, but have to travel to other locations to find food (i.e. nectar and pollen) [56, 162]. Bats living in caves and rats living underground face similar problems to bees, since they have to leave their secure home to gather food [52]. Migratory species face an even bigger challenge than bees, rats and bats: their breeding location and wintering grounds are separated by several thousands of kilometres. They have to undertake a long journey twice a year. Songbirds, for example, travel between the west and east coast of North America [154]. These examples illustrate that the animals are required to travel between different places and their survival depends on their ability to find, learn and return successfully to locations providing resources, security and a place to mate. The ability to reach a specific location in an environment is termed navigation [157].

Navigation, as illustrated by the previous examples, is wide-spread in nature. One of the most amazing examples is the honeybee, given its small size and organized society. Honeybees have fascinated humans for millenniums, from the Greek mythology to today [153], but they are not the only fascinating insects. Ants, wasps and bumblebees are also tiny animals travelling long distances to gather resources for their brood. Despite their tiny brains compared to human brains, they have many amazing skills. They explore the environment to discover food locations [129], learn to travel back and forth between those locations and their nest along habitual routes [94], and can learn and solve complex tasks [4, 3], such as solving and remembering to negotiate a maze [103]. The study of insect navigation is, therefore, a rich source of knowledge.

The interest for mankind in the study of insect navigation is not solely limited to gaining knowledge about complex behaviour and the underlying neuronal mechanisms. The study of navigation may also have a direct impact on the survival of honeybees and bumblebees, the two main pollinators of our crops. Indeed, beehives need to have enough resources to survive and, thus, efficient foragers finding flowers, collecting nectar and returning to their hive. An understanding of the methods they use to navigate may give hints to help bees to forage efficiently. Moreover, these insects, with their limited computational resources. are also energy-efficient systems and, nevertheless, manage to solve complex tasks. Their study has inspired and will inspire engineers to develop energy-efficient algorithms which contribute to improving autonomous robots both with respect to their performance and energy consumption.

Route navigation

The spatial behaviour of a navigating biological system, such as an animal, can be observed at various spatial scales, from the travel of birds across continents to the kinesin protein walking on microtubules. Most biological systems commuting between two locations, for example, a home and a food location, move along habitual routes. The navigation along such routes is a strategy which emerged in the animal kingdom at least 350 million years ago (Trilobites [13]) and is still seen nowadays in many species including humans. Hymenopterans, such as bees and ants, follow habitual routes between food locations and their nest. They have been studied in great detail over the past decades and, thus, are a great source of inspiration for understanding route navigation. The route-following algorithms used to explain the behaviour of ants rely mainly on a continuous memorization of the scenery along the route [6]. The continuous memorization is, however, demanding in terms of memory capacity. The further the agent travels, the more memory is required to remember the route. Bees travelling longer distances than ants may, thus, use other strategies to follow a route than a continuous memorization of the scenery. How can an agent, a biological or technical one, follow a route without memorizing all the places along the route (chapter 2)? The insect's journey may also be disrupted by the need to avoid an incoming danger or by navigational errors, which will displace the insect to an unknown location in the environment, i.e. away from its habitual route. Although ants have often been displaced away from their route to study their route-following algorithms [28, 175, 58], little is known about how an insect searches to find its habitual route again. How can an agent displaced away from its route find its route again efficiently (chapter 3)? Finally, how does the memorization of places along a route affect the risk of staying lost after having been displaced away from the route (chapter 4)?

Thesis outline

The first chapter of this thesis will provide an overview of navigation. The problem of navigation will be divided into four fundamental problems, and a formalism will be introduced for each problem. The strategies of navigation will be classified in a hierarchy of classes of increasing complexity. Each fundamental problem will be linked to each navigation strategy. Finally, an overview of insect navigation will be given in relation to the four fundamental problems of navigation.

In the second chapter, we will see that route-following may be achieved by taking advantage of the necessity to avoid collisions with objects in cluttered environments. By combining a bio-inspired collision avoidance system with a goal direction (e.g. given by the integration of the agent's past motion), the agent will follow idiosyncratic routes. This algorithm is one of the simplest routes following algorithms, because no location along the route needs to be memorized by the agent, as long as the overall direction of the goal is known by the agent.

In the third chapter, the agent will be assumed to know all locations along its route perfectly. The agent will be challenged to find its route again after having been displaced to an unknown location. It will be shown that an optimal search strategy does not exist, but that certain search strategies outperform others under specific conditions, such as the distance between the agent and its route and the distance the agent can travel before running out of energy.

In the fourth chapter, it will be assumed that the agent knows only certain locations along its route to safe memory space. It will be argued that the agent can still follow a sparsely memorized route by combining "naive" and "experienced" responses (later named predefined and calculated direction, respectively). Similar to the second chapter, the agent will be challenged to find its route again after having been displaced to an unknown location.

Finally, the limitations and advantages of three different classes of route following algorithms will be discussed with the help of the formalism introduced in the first chapter. An outlook for future research on route navigation in insects will then be given.

1 Overview of navigation

1.1 Navigation

Navigation is widespread not only in nature but also in our culture. Before the domestication of animals and the use of agriculture (more than 10,000 years ago [36]), humans were gathering and hunting for food, i.e. they were foraging. Nowadays, most humans do not forage, but still need to travel between certain different locations, such as their workplace, food stores and their home. Navigation in our world is not always easy. Most humans have experienced the situation of being lost. Humans have developed several tools to facilitate navigation in various environment, such as cities, forests, the sea or even space. We use maps or a navigation system (based on a global positioning system, GPS), for example, when placed in an unfamiliar environment or when required to reach an unknown location. Navigation without such a navigation system is challenging, even if the navigator has a map and a compass. The challenge to reach a series of unknown locations as quickly as possible with only the use of a compass and a map gave birth to the sport of “orienteering” (it first appeared in Sweden in 1886 as military training [168]). The recent use of GPS decreased the problem of navigating in an unfamiliar environment drastically, as the users always know where they are and where their destination is. Although we could use the navigation system provided by our smartphone constantly to commute between home and work, most people commute daily between those locations without using it. We, therefore, may ask how do we navigate, or more generally, what are the strategies concerned with navigating? The study of navigation may also provide a more intuitive and cheaper technical system to aid our navigation. Indeed, GPS is not available all the time, a costly system to maintain and uses a set of coordinates not intuitive for humans (an alternative project to mark places `http://what3words.com/`). Therefore, the aid of a navigation system currently based on GPS and maps may be substituted by a more natural system. Instead of hearing "turn right in 200 meters", for example, it will be more natural to hear "turn right after the bakery".

The topic of navigation will be addressed in the following section using four fundamental questions. Each question will be intuitively motivated and then formally described. The formal and mathematical descriptions of the four fundamental questions constitute an important conceptual basis of navigational behaviour observed in the animal kingdom. This basis will, at the end of the thesis, be used as a ground to discuss route navigation, and may serve in the future as a guide for the formulation of new research questions and the interpretation of behaviour observed in the animal kingdom. Different spatial behaviour has been observed in the animal kingdom and explained by different navigational strategies. The navigational strategies have been classified into seven classes to date. This classification is independent of the division of navigation in four fundamental questions. The links between the classification – adapted from classifications established previously [157, 50, 55] – and the four fundamental questions will be made in the second section of this chapter. The four fundamental questions, their formal descriptions and the classification will form the theoretical basis of the thesis. The latter will be highlighted by a section with examples observed in insects. It will be argued that insects are stunning navigators, as they face similar challenges to many other animals (including humans) and do so with a brain weighing only a few milligrammes.

1.2 A fourfold problem

Navigation is one of the most challenging tasks to be solved by animals and mobile artificial systems, i.e. an agent. The task of the agent while navigating is to change its current state "I am at the starting location" to the state "I am at the goal location". Navigation is, therefore, a special case of problem-solving; here, the location of the agent always pertains to its state. One divides any complex problem into smaller ones in order to find the solution. The individual problems will be introduced with an example: A biker [1] travelling

[1] The gender of the biker is not relevant in the present context: the neutral "they" will be used to refer to the biker.

from Lyon (co-ordinates: $45^\circ 46' N$ $4^\circ 50' E$) to Bielefeld (co-ordinates: $52^\circ 1' N$ $8^\circ 31' E$), and, subsequently, phrased as questions. The biker has already done the journey once and, therefore, remembers certain locations along the route. Those locations, such as a food store, hotel or route intersection, are characterised by certain features, such as the shape, colour or distance to objects surrounding the location. The biker, on the first trip, has detected and remembers some of those characteristics; the biker can detect characteristics at a place (*detectability*). On their second journey, looking for a familiar location, they have to compare the characteristics of places remembered and the characteristics detected at their current location; the biker can recognise a place (*recognizability*). However, recognising a peculiar location, such as a route intersection, will provide little help to our biker. They need to know in which direction they have to move; the biker has a sense of direction (*directionality*). Last but not least, the biker will have to avoid cars, buildings and trees along their journey. They may even have to venture along a novel route due to a broken bridge. The biker needs to estimate the feasibility of the motion planned; they have a sense of *feasibility*. Setting aside the obvious requirement of the motion of the agent, the problem of navigation can be divided into four questions:

Table 1.1: The four fundamental questions of navigation

What characteristics can be detected at this place?	*detectability*
How do I recognize the characteristics of one place?	*recognizability*
Where is (are) the other relevant place(s) relative to me?	*directionality*
How do I get to the other place(s) from here?	*feasibility*

I will give a formal definition for each of the problems in order to grasp a better understanding of the underlying mechanisms or strategies solving the four problems of navigation. The formalism will become useful when any model of navigation is discussed. The formalism will be used especially at the end of the thesis to discuss different strategies for route navigation. Each formal definition will be phrased around the journey of our biker.

1.2.1 What characteristics can be detected at this place?

Assuming that the biker does not leave their bike and does not use any other mode of locomotion, they can only reach places on the same landmass. The ensemble of all places reachable is called the environment. The environment contains a variety of features, such as the colour and the shape of objects, the light intensity of the sky or the texture of the ground. The features are distinct from each other (a spherical object cannot be cubic) and, therefore, the collection of all features forms a mathematical set X. A place is characterised by a subset A of the set of features X. A feature x is a characteristic of a place if $x \in A$. We have, thus, a characteristic function for all features:

$$1_A(x) := \begin{cases} 1 & \text{if } x \in A \\ 0 & \text{otherwise} \end{cases} \tag{1.1}$$

However, every feature at a place may not always be present or sensed by the biker. We assume, for instance, that our biker rode their first journey in summer, and their second in winter. On their second journey, most trees will not have leaves and the ground may be covered by snow. Certain characteristics observed during the first journey are no longer present. Our biker, moreover, cannot sense – without the use of additional devices – infrared or ultraviolet light. Due to the disappearance or appearance of certain features and the sensing limitations of the biker, they sense at a place a subset B of the set of features X.

An additional problem that our biker has to solve is how to trust their senses. Sensing a feature x may also be the result of noise in the environment (e.g. photon-noise in low light conditions) or in the processing stage (e.g. the nervous system). The biker, therefore, needs to discern between reliable and unreliable features. Each feature x in B has been sensed from either a reliable or an unreliable source of information. The biker emits two hypotheses: the feature x comes from a reliable source of information (first hypothesis H_1), or it comes from an unreliable source of information (second hypothesis H_2). The biker has to discard every unreliable feature x in B. The probability of having a reliable (respectively, unreliable) source of information knowing that the feature x has been sensed is the conditional probability $p(H_1|x)$ (resp. $p(H_2|x)$). The biker may trust their senses when the following inequality is satisfied: $p(H_1|x) > p(H_2|x)$. However, those two probabilities may not be known *a priori* by the biker. The biker may, on the other hand, have knowledge about the probability of sensing a feature x knowing they have a reliable (respectively, unreliable) source of information, i.e. the conditional probability $p(x|H_1)$ (resp. $p(x|H_2)$). Moreover, they may have an estimate of the probabilities of having a reliable or unreliable source of information, $p(H_1)$ and $p(H_2)$, respectively, through experience (or as a consequence of the evolution). Using the Bayes chain inequalities for conditional probabilities [167], one can show that:

$$p(H_1|x) \leq p(H_2|x) \Rightarrow \frac{p(x|H_1)}{p(x|H_2)} > \frac{p(H_2)}{p(H_1)} = L(x) \tag{1.2}$$

where $L(x)$ is the likelihood ratio of x

To sum up, a feature x in X is a detected characteristic of a place A for the biker when:

$$\begin{cases} 1_A(x) = 1 & \text{membership} \\ 1_B(x) = 1 & \text{sensible} \\ L(x) < \frac{p(x|H_1)}{p(x|H_2)} & \text{detectable} \end{cases} \tag{1.3}$$

1.2.2 How do I recognise the characteristics of one place?

I will define recognition in a broad context and, therefore, define recognition of an entity. Tree, mountain and river are, for example, entities. The word “entity” may be read as a tree, mountain or river, although an obvious loss in generality will, thus, be made. An entity can be dissimilar to another. The dissimilarity is a continuous function. Two identical entities do not differ and, therefore, have a dissimilarity equal to zero (*identity of indiscernible*). The dissimilarity between two entities can be infinite. It lies, thus, in the interval $[0, \inf[$ (*non-negativity*). Moreover, the measure of dissimilarity is symmetric, i.e. the dissimilarity between an entity x (a tree) and entity y (a mountain) is equal to the dissimilarity between the entity y (the mountain) and entity x (the tree) (*symmetry*). Finally, the dissimilarity between an entity x (a tree) and an entity z (a river) is smaller than, or equal to, the sum of the dissimilarities between the entity x (the tree) and an entity y (a mountain) and the dissimilarity between the entity y (the mountain) and z (the river) (*subadditivity*). The dissimilarity function can, therefore, be seen as a distance [2] function (or metric) between two entities. A place is, thus, recognised when the distance function between the current characteristic of one location and the memory of the characteristic at this location is zero, i.e. they are indiscernible. In order to recognise a place, the biker needs to compare – by using a distance function – the characteristic of the current place with the one stored in their memory.

1.2.3 Where is (are) the other relevant place(s) relative to me?

The concept of directionality is contained in the word “where”. A direction is the information contained in the relative position of one point – the place relative to me – with respect to another point – my position – without the information of distance. Mathematically, the

[2] Distance does not refer to the physical quantifier of how far one place is from another (i.e. usually measured in metres), but refers to the mathematical generalisation of the concept of the physical distance.

direction is the unit vector $\overrightarrow{AP}$, where A is the current state of the biker and P is the desired state to be reached. The number of dimensions of $\overrightarrow{AP}$ is equal to the number of dimensions characterising the state of the biker. When the state of the biker is only specified by their location (x, y, z), for example, the direction vector $\overrightarrow{AP}$ has three dimensions. The state of the biker may, however, contain many other parameters, such as the quantifier of energy the biker has currently.

The second part of the third fundamental question of navigation is the relevance of a place. Our biker wants to reach Bielefeld from Lyon. Although Bielefeld is the destination of the journey, it is not the place of immediate relevance to reach the start. The biker will have to regain energy, rest, may have to repair their bike, before reaching their final destination. Hotels, restaurants and grocery stores along the route will be relevant places to reach. When the biker is tired, reaching a hotel is more relevant than a restaurant. A nearby hotel – not one on the other side of a mountain – is then preferable. To generalise this example, the relevance of a place is one quantity assigned to a place. The quantity can be formalised as the output of a mathematical function. Every function transforms a set of inputs to an output (here, the output is the relevance). The set of inputs of the function comprises, but is not limited to, the state of the biker (e.g. level of energy), variables intrinsic to the place (e.g. the amount of food available), and the distance between the biker and the place. The function is defined by environmental constraints (the presence of a mountain), biological constraints (the need of the biker to sleep about eight hours per night) and temporal constraints (the hotel room is usually available only after 2 pm).[3] To conclude, the directionality of a place can be expressed as a unit vector, the direction from the current state of the biker A to the state of the biker once the relevant place is reached P, and the relevance of a place by a value $R = f(A, P)$.

1.2.4 How do I get to the other place(s) from here?

Once our biker knows where the current relevant place is relative to their current location, the place still has to be reached. The biker, therefore, needs to change their current state “I am at an intersection” to “I have reached the restaurant”. To do so, the biker will have to ride, negotiate turns, slow down, etc. The biker, therefore, needs to successively transform their state to arrive finally at the restaurant. The biker can change their state by applying a series of transformations or functions. They can reach their destination if a transformation f (or function) exists leading to the desired state P from the biker’s current state A. When the system of equation $P = f(A)$ is linear, it can be rewritten in matrix form $P = TA$. The existence of the transformation T is, in this case, trivial and can be

[3] The concept of the relevance of a place in navigation has a clear parallel in the theory of foraging. The optimality of a foraging model is assessed according to three assumptions: the decision (set of permissible inputs), constraints (the definition of a function) and the currency (the output of the function) [145]. The relevance of a place can, therefore, be seen as a currency in the context of optimal foraging.

solved by following the methods of linear algebra. However, realistic systems are often non-linear. The biker may simplify the problem by decomposing it into smaller or elementary transformations $P = f(A) = f_1(f_2(f_3(...f_i(\ldots f_n(A)\ldots)...)))$. Once the problem has been solved, the list of transformations should be remembered by the biker to avoid solving the problem again, because finding the function f can be highly demanding in time and computational power [83].

1.3 Hierarchy of navigational strategies

How the four fundamental questions may be answered depends on the context and complexity of the navigational task. An agent may not use the same strategy when reaching a nearby location compared to a distant one. Reaching Lyon from Bielefeld requires more planning than going back home from work. The complexity of navigational strategies can be grouped into 4 + 3 classes referring to local and non-local navigation, respectively. An agent using a local (respectively, non-local) navigation strategy needs to recognise a single location (respectively, multiple locations), such as the goal. Based on classifications established previously [157, 50, 55], I will introduce the seven different classes in the order of their increasing complexity: searching, direction following, aiming (named target approaching by Olivier Trullier et al. [157], and taxis by James L.Gould [55]), guidance (named piloting by James L.Gould [55]), state recognition-triggered response (named place by Olivier Trullier [157]), topological navigation and metric navigation (named survey navigation by Mathias O. Franz [50], true navigation by James L. Gould [55]). A definition of the respective navigational strategy will be given first in italics, followed by the limitations of the strategy and an explanation of when it is necessary or worth using. Human-centred examples will then be added to give an intuitive idea of the strategy, as well as information about its use in the animal kingdom.

1.3.1 Searching

The agent, away from its destination, does not orient actively towards it; it reaches its destination by chance.

Searching is a fundamental strategy and is used in a myriad of situations. When an agent does not have any information about the environment or has already used all the knowledge available to navigate, but has failed to reach its destination, needs to search. This simple navigation strategy is, however, inefficient. The agent may move for an extended period of time before finding its destination by chance. Students have to find their way in the university every year; they will have to locate the building (often indicated on a map) and then locate the specific room labelled by a number. New students may be unfamiliar with the logic – if any exists – underlying the room numbering and may just search for the room. For animals, the feeding locations may change over time (especially if the animal is hunting other animals). Moreover, a newborn “naive” animal may have no knowledge about the location of the feeding sites. In such situations, the only method to reach its destination is by searching (reindeer [95], spider monkeys [116], grey seals [2], fruit-flies [118], bees [120, 129, 87], ants [130], moths [119], marine predators [137, 70] albatrosses [71], insect review [11] and models review [74]). An agent displaced from a familiar location also needs to search to find familiar locations again (ants [79]).

1.3.2 Direction following

The agent, away from its destination, moves in a direction with neither the need to recognise a location visited previously nor the detection of information at the destination.

The risk of moving in a given direction is to move away from the destination when the direction has been wrongly indicated. When the direction points toward the destination, direction following is more efficient than searching, as the agent moves directly in the correct direction. The direction can be indicated by three modalities: communication, and idiothetic and allothetic information.

Communication is often used by humans. In order to reach a place, a naive human may seek help and ask a human familiar with the environment to indicate the direction to follow. The non-naive human, i.e. the expert, may, for example, only give the overall direction of the destination, and thus, the naive human only knows this direction. Passing directional information via communication has also been observed in bees [162]. Experienced bees indicate the direction and distance to a rewarding location, i.e. a food source, via a dance (the waggle dance).

An agent may also have acquired information of the goal direction from its own motion without using communication, i.e. idiothetic information (patent [86], models [98, 86, 61] or review [47]). The integration of its motion, i.e. the path travelled by the agent while searching for a previously unknown valuable location (such as a food source), gives the direction and distance of the starting location [61, 132]. However, if this method of path integration is subject to noise, the resulting information about the direction and distance of the goal may be biased. The error in the measurement of motion is likely to accumulate over time [161]. The agent can, however, compensate for this by searching once its path integrator indicates that it has reached the destination [100, 170].

Finally, the direction to follow may be provided directly by local information, i.e. directional cues. Certain species of ant mark the route between a food source and the nest by odours [41]. The individual on the route just has to follow the trail of odours until the destination, i.e. the end of the route, is recognised. This strategy is, however, not suitable for every environment. The markers along the route may vanish [59], for example, by evaporation (desert ants). Visual cues can also provide directional information. Bees and dung beetles, for example, use the bearing of the sun in the sky [44]. The sun is not the only allothetic directional cue detectable by vision. One may use the polarization pattern [81, 67, 44], the moon or the stars [34, 139].

1.3.3 Aiming

The agent, away from its destination, detects characteristics of the destination and aims towards it.

A conspicuous marker is required at the goal location for this strategy to be employed by an agent. Natural visual markers are often visible only within a limited range from the goal location, as they are hidden behind the clutter of the world, such as trees or buildings. An odour, as a conspicuous cue, is not occluded by objects, and sounds may even be transmitted or bent by the objects. The detectability of such cues, nonetheless, decreases with increasing distances from the goal (decrease of odour concentration or sound amplitude). As humans, for example, we may detect a bakery by the odour of bread even when the door or the bakery sign are hidden. Odour and sound are clearly used for aiming in the animal kingdom. Mosquitoes locate humans (or other animals) based on their odour [53], bears can locate a bee hive several metres away by the smell of the hive, and nocturnal barn owls detect and catch their prey thanks to the prey's noises [135]. Visual markers are also used in the animal kingdom, from door recognition (human) to nest hole entrance. The entry to a bee or wasp hive is a hole in the ground. The hole is dark and, therefore, can be distinguished from the different object laying on the ground around it (wasp: [180], bumblebees: personal communication Anne Loebecke).

1.3.4 Guidance

The agent, away from its destination, is guided toward its destination by establishing a relationship between information currently available and the characteristics of the destination location remembered.

Guidance is the most complex local navigation strategy. Even destinations hidden in the clutter of the world, without conspicuous markers, can be reached efficiently by the agent. The agent establishes a relationship between the characteristics of the goal location remembered and the one at its current location. The direction in which the agent should move to reach its goal is derived from the relationship. Therefore, the relationship is an ego-relationship. However, for an ego-relationship to be established, information surrounding the agent at its current location needs to be comparable to the information memorised previously at the goal location. Therefore, an ego-relationship can only be established within a limited area (the catchment area) around the destination [4]. We often indicate a destination in our daily life by providing a relationship between the location to reach and conspicuous landmarks. One may say, you will find a restaurant with delicious cheese at the intersection of Rolandstraße and Siegfriedstraße (Bielefeld, Germany). The navigator only needs to follow one of the two streets to find the intersection and, thus, the restaurant. Guidance was, before the spread of GPS, used by fishermen to locate their fishing spot, an area with

[4] A navigation problem may be described as “local” if a relationship between the information at the goal location and the information available to the agent at any other location in the environment exists.

a high density of fish, by using the bearings of lighthouses (or other high landmarks) on the coast. A seminal experiment inferring the use of guidance in the animal kingdom was performed by Tinbergen in 1938 [155]. The nest hole of a wasp in the ground was surrounded by pine trees. When the pine trees were displaced after the departure of the wasp, the returning wasp was searching for its nest at the centre of the circle formed by the pine trees. This type of experiment has been reproduced with many different animals (ants [109, 173], wasps [177] and bees [38]).

1.3.5 State recognition-triggered response

In a given state (e.g. at a given location), the agent recalls the direction to follow by recognizing the state (the location).

This strategy is a simple extension of any local navigation strategy, as it is based on a concatenation of sequentially applied local navigation strategies. It is the method typically employed by a GPS navigation system to indicate the route to follow between the current location and the final destination of the journey (e.g. take the second exit at the roundabout). Route following behaviour observed in ants has been modelled by a sequence of recognition-triggered responses (full memory [6, 1], infomax [6] and bioinspired network [1]). It has also been inferred in a recent study that wasps use such a strategy for homing [149]. A wasp leaving its nest performs complex manoeuvers around its nest, i.e. a learning flight. When the wasp returns home, it will encounter and recognise locations seen during its learning flight and, therefore, bear in the direction of the nest.

1.3.6 Topological navigation

The agent, at every familiar location, knows the procedure to reach at least one other familiar location; the agent may plan its route by moving virtually through a sequence of familiar locations.[5].

Topological navigation is the simplest form of navigation strategy allowing planning, i.e. to choose between different routes to reach the destination. This form of navigation is extremely powerful. Subway or bus plans in a city are based on the concept of topological navigation. They provide information about how to move from one station to another as long as they are connected by a subway or buses. When two stations are separated by at least one station, it is, therefore, impossible to move between those two stations (without additional information) without passing by at least one of the separating station. In order to assess that an animal uses a topological navigation strategy, one needs to show that the animal is able to choose different paths without relearning or exploring its environment again. The behaviour of rats is consistent with the use of topological information. In an experiment, rats were placed in an environment with three paths of different length between the start

[5] The intuitive and formal representation of topological navigation is a graph or network; the familiar locations are represented here by nodes, and the procedures to reach other locations from a given location are encoded by an edge, i.e. the connection between nodes.

and the goal location. During their exploration, they quickly learned to follow the shortest path. The blockade of the shortest path then triggers an exploration and the rats will, thus, follow the remaining shortest path, thus, still consistent with a state recognition-triggered response. Interestingly, when this path is also blocked, the rat select the remaining path directly, indicating an ability to select between paths, a behaviour coherent with topological navigation [156, 157].

1.3.7 Metric navigation

At any location, the agent knows the distance and direction to any other location; the agent can plan any route within the range of the map.[6].

An agent can, therefore, travel along novel paths by using the map, i.e. it does not need to use *a priori* established connections [7]. The navigation of many robots is based on this strategy. The class of algorithm self-localisation and mapping (SLAM) is commonly used in robots [25]. The presence of place cells in the brain of bats, rats and humans is an indicator of a potential map encoded in the brain [106]. Those cells encode the metric of the environment. To date, those cells have not been found in insects, thus, the use of metric

[6] With respect to the formal representation of topological navigation, a metric navigation can be conceived as a topological navigation with the nodes being anchored in space by its metric position

[7] Metric navigation is extremely difficult to perform. I invite the reader to try to navigate without a map in an unknown city. Take a taxi to a hotel without looking outside much, and then explore the surroundings of the hotel. (As a precaution, do not forget a backup system: the telephone number and address of your hotel)

navigation in insects is still under debate. To infer the use of a metric navigation by an animal by monitoring its behaviour, the animal needs to travel along a novel route. However, the novel route should not be followed by other simpler strategies, such as direction following or path integration. Insect behaviours have been explained so far by simpler strategies than the use of metric navigation strategy [26, 33, 29].

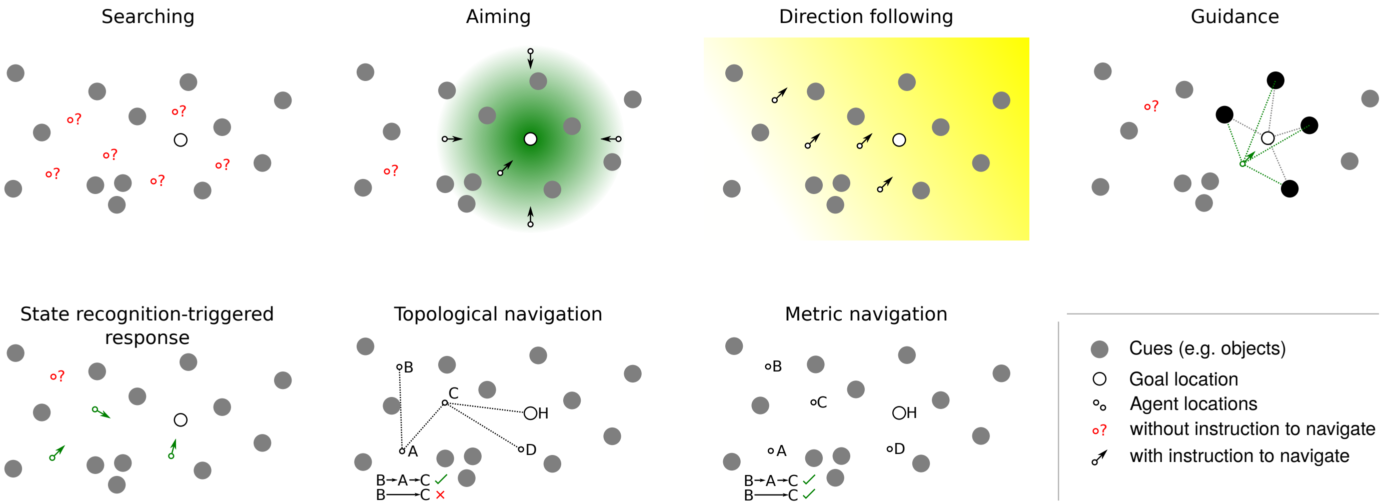


Figure 1.1: An illustration of the seven different navigational strategies

Table 1.2: Link between the hierarchy of spatial behaviour and the four fundamental problems of navigation

Navigational strategy	Problem	Capability-ies of the agent.
Searching	*detectability*	Characteristics of the goal
	recognizability	The goal location
	directionality	
	feasibility	
Direction following	*detectability*	Direction at certain location Characteristics of the goal
	recognizability	The goal location
	directionality	
	feasibility	
Aiming	*detectability*	Conspicuous cue(s) marking the goal
	recognizability	Conspicuous cue(s) marking the goal The goal location
	directionality	Bearing of the conspicuous cue(s)
	feasibility	
Guidance	*detectability*	Cues
	recognizability	Cues having a relationship with the one at the goal location The goal location
	directionality	Establish a relationship between current cues and memorized cues
	feasibility	
State recognition-triggered response	*detectability*	Cues
	recognizability	Cues at a known places The goal location
	directionality	Recall (sub)goal-direction from a known place
	feasibility	
Topological navigation	*detectability*	Cues
	recognizability	Cues at known places
	directionality	Select the procedure to reach a goal
	feasibility	Does a procedure or series of them exist?
Metric navigation	*detectability*	Cues Metric
	recognizability	Cues
	directionality	Follow a novel route based on the metric Select the procedure to reach a goal
	feasibility	Detect dead-end Does a procedure or series of them exist?

1.4 Insect navigation

Insects are relatively small animals compared to humans, and they use their tiny brains to solve complex tasks. The brood-caring insects have evolved to travel efficiently and repeatedly between their nest and locations rich in resources, such as food. Many hymenopteran species share common strategies to navigate, such as following routes, and, therefore, constitute an ideal ground for ethology. The computational limitations of a brain with a small number of computational units (i.e. neurones)[169] may restrict the complexity of algorithms used to mediate those behaviours and, therefore, constitute an ideal ground to reveal neuronal mechanisms of complex behaviour. The wide range of insect behaviour together with the limited underlying computational resources have inspired engineers to apply bio-inspired principles in robots and algorithms for several decades [143, 48]. Insects, therefore, have also been a great source of inspiration in the field of navigation [164, 82, 113]. Moreover, the navigational strategies employed by honeybees and bumblebees are linked to our food production. Indeed, honeybees and bumblebees pollinate our crops [160] and reach those crops thanks to their navigational skills.

We have seen that navigation is the successive transformation of the current state of the animal to its state at its destination. The state of the animal in navigation always contains the position of the animal in the environment. However, the state may contain other types of information. I will, therefore, first elaborate on the state of an insect: proprioception, motion estimation and integration, spatial-orientation, energy and hunger [8]. After the description of the state of an insect, I will provide answers to the four questions of navigation (*detectability*, *recognizability*, *directionality* and *feasibility*). Each section, focused on one of the four questions, will start with a description of the constraints imposed by the environment and the sensing ability of an insect. The sections will be highlighted by examples of the behaviour of insects and models inspired by those behaviours.

1.4.1 State of the insect

Proprioception

The sense of proprioception is important to answer the question: How does an insect get to the other place(s) from here? section 1.4.5 The ability of the animal to move actively is a precondition for navigation. The animal needs to produce forces to propel itself by moving limbs: legs in order to walk, wings in order to fly. Limbs are equipped with an extensive sensory system. Some elements of the sensory system are proprioceptors. They monitor the position of the leg segments or the wing orientation and amplitude. Insects equipped with proprioceptive sensors have information about their posture. These sensors are used to help the control and stability of the insect when

[8] The motivation of the agent is also an important component of the state. However, motivation cannot be measured or quantified without repetitive visits, i.e. an *a posteriori* measure of the animal's state. It has, therefore, been left out of this list

walking [27, Chapter 8.1] or flying [27, Chapter 9.7].

Motion estimation and integration

Navigation requires moving through the environment. Integrating the motion gives the insect an estimate of the distance between its initial and current location. This motion estimation is, however, subject to noise accumulation [161]. Nevertheless, motion integration is used for navigation. While a walking insect achieves motion integration by counting its steps [170], flying insects need to monitor the apparent displacement of objects (honeybees [142, 35, 8], bumblebees [9], flies [76] or review [171]). The motion integration can be achieved with a simple neural network [61].

Spatial-orientation

The orientation is defined relative to the environment. Although the insect may estimate its orientation by integrating its motion [132], it (and also any system) needs to detect the bearing of a distant object, such as the sun, the moon or another prominent landmark, to unambiguously estimate its orientation in space. The insect using the sun or the moon is faced with two challenges. The sun (or the moon) may be hidden behind clouds and moves during the course of the day (respectively, night). The scatter of the light through the atmosphere creates a polarisation pattern (polarisation is a property of any electromagnetic wave including light). Insects are equipped with sensors sensitive to the polarisation pattern of light and, therefore, can detect the bearing of the sun even when the sun is hidden behind clouds. Moreover, insects are able to compensate for the movement of the sun [101, 44, 34, 139]. Estimating the bearing of one distant object is one of the simplest strategies to measure its orientation. An alternative method to estimate its orientation is computing the dissimilarity between a memorised snapshot and the current view of the sky. The degree of dissimilarity between the two views provides the agent with an estimate of its current orientation. This strategy seems to be used by dung beetles [45].

Energy and hunger

Movements require the transformation of mechanical or potential energy into kinetic energy. Insects (similar to other animals) mainly use mechanical energy to propel themselves and, therefore, consume the energy provided by food. However, food in the environment is neither available at every instant in time nor at any location. The insect's energy, therefore, decreases during its journey [46]. The decrease in available energy results in an increase in the central excitatory state and an increase in the sensitivity of the insect's chemoreceptor (involved in food detection) [27, Chapter 2.3]. Food deprivation has long been used to motivate animals to learn a task. A male bumblebee, for example, deprived of food for 24 hours will perform a learning flight after feeding at a flower, indicating the relevance of finding the flower again. Male bumblebees, which are not deprived of food, do not learn the flower position [122].

1.4.2 What characteristics can be detected at a place?

Brightness

Insects are equipped with eyes. Although the resolution or the field of view of the eyes may differ between species (or between individuals in a species due to differences in body size, e.g. bumblebees [140]), they all share the property of sensing light intensity. No human needs to be convinced that brightness is a rich source of information in the world as it is the main sensory modality for humans in many tasks. The extraction of relevant information may, however, be extremely complex (see, for example, the broad field of object recognition). Driven by the argument that insects have limited computational power [77], a variety of simple methods have been characterised in the last few decades. The line between the objects (or the ground) and the sky – i.e. the skyline – in an environment containing objects (e.g. cities), for example, is a powerful characteristic of a place. Objects nearby have a bigger apparent size than far objects of a similar height. The skyline, therefore, changes between places and has been used for place recognition [146]. The skyline can also be used to determine in which direction to move. Two skylines taken at places nearby will match best when their orientations are aligned. An agent can, thus, scan the environment in order to find the best match, and then decide in which direction to move. This strategy has been used for route following [10]. One main advantage of the skyline in insect navigation is its ease to extract. Indeed, most insects are equipped with ultraviolet photoreceptors, and ultraviolet light is mostly found in the sky.

Distance

The distance to surrounding objects is another important characteristic of a place. It is not because an insect sees the flower that the insect is at the flower position. The flower may be further away. In order to reach and land on the flower, therefore, the insect needs to estimate how far the flower is. Insects, however, are not equipped with a direct method to estimate the distance to surrounding objects. The distance between their eyes is too small for distance estimation with binocular vision and they are not equipped with ultrasound or a laser rangefinder. They have to estimate the distance to surrounding objects with monocular vision [43] (or touch [69]). Two strategies are possible to estimate the distance to surrounding objects with vision: apparent size and apparent motion. Knowing the height of an object, the insect may extract the distance of the object by measuring its apparent size. However, the heights of objects are usually not known *a priori*. An alternative way to estimate distance is by determining the apparent motion of objects. When the animal moves through its environment, the surrounding objects move on its eyes in the opposite direction. Knowing its own velocity (speed and motion direction), the distance of an object may be extracted from its apparent motion [78]. This can be achieved only when the insect is mainly translating[9]. However, without knowledge about the

[9] A flying insect shows an active gaze strategy composed of two phases: saccades, containing most of the insect's rotation, and inter-saccades, containing most

velocity of self-motion, only the relative distance to the object can be measured. Complex algorithms are required in order to extract both the velocity of self-motion and distance [78, 147]. Nevertheless, the apparent motion of objects (or object-flow) has been shown to be used by insects for place characterization [38]. Moreover, it has been shown using a bio-inspired algorithm that the surrounding distance to objects provides sufficient information for local navigation (Martin Müller in prep).

Odours

Most places in the world have a characteristic odour. Flowers, for example, are known for their rich variety of odours, and dead animals emit a strong repelling smell for humans. Insects are equipped with batteries of sensors specific to certain odours, i.e. certain molecules or cocktails of molecules. The odour of a place can be detected by combining the different responses of odour receptors [99]. Foragers, such as bumblebees and honeybees, use odours to detect and recognise the flower species on which they are feeding [56, Chapter 9.6] and to detect whether the flower has been visited previously [56, Chapter 10.3]. The use of odours is not limited to foragers of nectar and pollen. It has been shown, for example, that some ant species use odours to reach their feeding location [22], follow a trail or reach their home [24, 77]. A forager returning erroneously to a home of another colony may quickly be attacked by the members of this colony,

of the insect's translation. The distance to surrounding objects during intersaccades can be estimated by using motion detection mechanisms [131].

evidently signalling it has not reached the correct location [60].

Other cues

Some insects are, moreover, able to detect sounds, vibrations, and electrostatic or electromagnetic fields. Although these cues may not naturally characterize relevant places for an insect, such as food and nest locations[10]. Nevertheless, ants can locate and reach a nest marked by vibratory or electromagnetic cues [23], and bumblebees can be trained to recognise places which are electrostatically marked [39].

1.4.3 How does an insect recognise a place?

We have seen in the last paragraph that places can be characterised by a large set of simple means. But how does an insect recognise a place? To answer this question, we are faced with a fundamental problem. The strategies employed by an insect to navigate are assessed experimentally by analysing its behaviour. This behaviour, however, only informs us about the direction followed by the insect at a given position in the environment, and, thus, not (or only indirectly) how an insect can recognise a place. Therefore, models constrained by the behavioural analysis are required to infer strategies used by insects to recognise places. Consequently, this paragraph

[10] Male crickets orient actively towards the song of a female in the context of mating behaviour. Although this behaviour involves orientation or aiming, it is more similar to predation than navigation, because the place to reach (here, the female) moves.

will only be highlighted by such models. One approach for place recognition is to compare the characteristics at the current location with those of every previously memorised location. For a comparator to be biologically plausible, it has to work at a low spatial resolution and uses mostly local computation (the visual sense of an insect is retinotopically mapped in the brain), as little memory as possible and involves little plasticity, because the early stages of the visual system of an insect are thought to be hardwired. The approach mentioned above requires all locations visited to be memorised and compared, i.e. a process requiring large memory space and computational power. The comparison between the characteristics detected at the current location and those memorised can be performed by a network (acting both as memory and comparator). The input of this network are the characteristics sensed at the current location, and the output is how similar those characteristics are compared with the ones stored in the memory [6]. Although this model solves the problem of memory, it involves computations across the entire visual field and numerous changes in the weights between neurones. Moving closer to an insect brain, the same concept can be achieved by a network of neurones having a structure similar to the mushroom body, an area of the brain necessary for associative learning of at least odours [1]. Although this model is not retinotopic, it is, to date, the likeliest solution to be implemented in the brain for place recognition.

1.4.4 Where is (are) the other relevant place(s) relative to the insect?

The most relevant places for insects are their home (where they can hide from predators and care for their brood) and food locations. Whilst home is a constant and unique place, food locations vary over time, so an insect may use more than one feeding location during their lifetime. The location of the relevant place is obtained via either communication or computation. There is little evidence to date for a metric navigation in insects. Indeed, evidence reported has been explained by simpler mechanisms than metric navigation [26, 33, 29]. Strong evidence supports the use of a state recognition-triggered response [4, 3]. Insects can be trained to associate the direction to move according to surrounding cues, as shown in a Y-maze experiment. A Y-maze can be seen as an intersection along a route. The insect has to choose to go to either the right or the left. One of the decisions will lead to the food, the other not. Therefore, the principle highlighted in the Y-maze experiment may be generalised to route following.

Pinpointing a specific location involves a local navigation strategy. Aiming is a possible strategy when the home location is unambiguously visible. However, the home entrance of solitary wasps and bumblebees (*bombus terrestris*) is accessible only via a tiny hole. A guidance strategy is, therefore, more probably employed by the insect. Guidance strategies can be classified into two categories: correspondence and holistic models [104]. The former models involve correspondence between characteristics at the home location and the ones at the insect location [26]. The correspondence between characteristics requires a spatial search across the entire visual field to determine the direction to follow. This search cannot be achieved locally and, thus, cannot be implemented retinotopically. However, the visual system of the insect brain is retinotopic, at least in the early stages. Therefore, correspondence between characteristics is unlikely to be calculated by insects [159, 93]. If the insect cannot find correspondences between the characteristics to determine the direction to follow, it can determine it by comparing the characteristics between the two locations as a whole (the principle behind holistic models). The insect acquires, for example, a panoramic image at its home location. When displaced away from the home location, it calculates the difference, pixel by pixel (i.e. locally in the visual field), between the panoramic image at its "home" and the panoramic image seen currently. The sum of the square of the difference gives a measure of how far the agent is from its location. The agent can then move along the gradient of this measure to minimise the difference [148, 178]. This measure is also lowest when the images have been acquired in the same orientation. Therefore, an agent can determine its orientation compared to a stored image by rotating [6]. The strategy of scanning its environment by rotation can be used to do route following when images along a route are acquired in the direction of the route [6]. Holistic models are the most plausible ones to explain insect navigation. The most popular ones are: the average landmark vector (ALV), based on a fixed number of detectable landmarks

[82]; Rotational image differences, based on the principle that two images match best when they are aligned [179]; and centre of mass ALV, based on the ALV without the need to detect a fixed number of landmarks [62], and are already very powerful for navigation.

1.4.5 How does an insect get to the other place(s) from here?

An insect knowing where to go also needs to determine how to go there. The insect could move slightly along the goal direction and determine again where to go. Since the determination of where to go is a demanding task, it may be more efficient to perform this calculation sporadically. Ants, for example, stop and scan the environment, probably to determine the direction in which to go [174]. Similarly dung beetles – pushing their ball of dung along a straight line – from time to time, walk on top of their ball and reorient themselves [45]. Those two examples are consistent with a sporadic update of the goal direction. Between two updates, the insect needs to follow the goal direction without complex computation (i.e. without involving guidance strategy), by using either an aiming or a direction following strategy. Walking insects can maintain their direction by using proprioceptive or allothetic information (such as the sun). However, flying insects can only use allothetic information, because they are moving in a medium itself moving compared to the ground (e.g. by the wind). Nothing is known to date about how often flying insects update their goal direction, but it is clear that honeybees can use allothetic cues, such as the sun, to determine a direction [81].

The environment may be cluttered with obstacles, or other natural barriers (e.g. water for walking insects). The insect needs to cope with obstacles by avoiding collision with them while maintaining its overall goal direction. Avoiding collisions is more important for flying insects than for walking ones, as wing damage may be fatal to the insect [49]. Therefore, I will only highlight examples concerned with flying insects. A simple method to avoid collisions is to integrate the apparent motion of surrounding objects into one collision avoidance direction. This method does not require the extraction of close objects from the background and distant objects (e.g. [90]). The collision avoidance direction may drive the insect away from obstacles and also centre the insect in corridors. Honeybees and bumblebees indeed centre while flying through a corridor [142, 35, 8, 88]. The collision avoidance direction can be seen as a repulsive force and the goal direction as an attractive force. The combination of those two forces leads the insect in a given direction [80].

2 Avoiding collision leads to common routes

This chapter is published as Olivier J.N. Bertrand, Jens P. Lindemann, and Martin Egelhaaf (2015). A Bio-inspired Collision Avoidance Model Based on Spatial Information Derived from Motion Detectors Leads to Common Routes. *PLoS Comput. Biol.*

2.1 Abstract

Avoiding collisions is one of the most basic needs of any mobile agent, both biological and technical. We propose a model of collision avoidance inspired by behavioral experiments on insects and by properties of optic flow on a spherical eye experienced during translation. Insects, such as flies and bees, actively separate the rotational and translational optic flow components via behavior, i.e. by employing a saccadic strategy of flight and gaze control. Optic flow experienced during translation, i.e. during intersaccadic phases, contains information on the depth-structure of the environment, but this information is entangled with that on self-motion. Here, we propose a simple model to extract the depth structure from trans-

lational optic flow by using local properties of a spherical eye. On this basis, a motion direction of the agent is computed that ensures collision avoidance. Flying insects are thought to measure optic flow by correlation-type elementary motion detectors. Their responses depend, in addition to velocity, on the texture and contrast of objects and, thus, do not measure the velocity of objects veridically. Therefore, we initially used geometrically determined optic flow as input to a collision avoidance algorithm to show that depth information inferred from optic flow is sufficient to account for collision avoidance under close-loop conditions. Then, the collision avoidance algorithm was tested with bio-inspired correlation-type elementary motion detectors in its input. Even then, the algorithm led successfully to collision avoidance and, in addition, replicated the characteristics of collision avoidance behavior of insects. Finally, the collision avoidance algorithm was combined with a goal direction and tested in cluttered environments. The simulated agent then showed goal-directed behavior reminiscent of components of the navigation behavior of insects.

2.2 Author Summary

The number of robots in our surroundings is increasing continually. They are used to rescue humans, inspect hazardous terrain or clean our homes. Over the past few decades, they have become more autonomous, safer and cheaper to build. Every autonomous

robot needs to navigate in sometimes complex environments without colliding with obstacles along its route. Nowadays, they mostly use active sensors, which induce relatively high energetic costs, to solve this task. Flying insects, however, are able to solve this task by mainly relying on vision and, although many robots carry a camera, a large majority has not used it for collision avoidance so far. Inspired by the abilities of insects, we developed a parsimonious algorithm to avoid collisions in challenging environments solely based on vision. We coupled our algorithm to a goal direction and then tested it in cluttered environments. The trajectories resulting from this algorithm show interesting goal-directed behavior, such as the formation of a small number of routes, also observed in navigating insects.

2.3 Introduction

Anyone who has tried to catch flying flies will be familiar with their amazing performance. Within a fraction of a second, flies perform high-speed turns to avoid a predator or a collision with an obstacle. The collision avoidance decisions are produced in a fly's brain with very limited neural resources [30, 169] and are transformed into an evasive turn within only a few milliseconds, a rather short time compared to human reaction times [163]. As such, flying insects have become an important model system for understanding the minimal computation requirements for spatial vision tasks, such as collision avoidance [43]. Engineers are also looking for fast and cheap collision avoidance algorithms, without the use of expensive devices, e.g. 3D laser rangefinders [150], or extensive computations, e.g. Lucas-Kanade optic-flow computation [91]. Any motion of an agent, such as an insect or a robot, induces apparent movement of the retinal image of the surroundings, i.e. optic flow. The optic flow experienced during translations in a static environment depends on the agent's speed, its nearness to objects and its motion direction. When the agent moves fast or close to objects, the optic flow amplitude will be high. By contrast, the rotational optic flow depends only on the ego-motion of the agent and, thus, is independent of the spatial layout of the environment. Information on the nearness of objects is relevant for determining a collision avoidance direction. Therefore, the translational optic flow can be exploited for collision avoidance. Flies, and also other insects and some birds, show an active gaze strategy, which separates the self-motion into saccades (i.e. mainly rotation) and intersaccades (i.e. mainly translation) [127, 126, 66, 14, 19, 18, 51, 76, 40]. The saccade amplitude of an insect or a bird is thought to be driven, at least in the vicinity of potential obstacles, by the optic flow gathered during the translation preceding the saccade.

Insects estimate the optic flow with correlation-type elementary motion detectors (EMDs), a concept first introduced by Reichardt and Hassenstein in the 1950s [117]. A characteristic property of the EMD is that its output does not exclusively depend on velocity, but also on the pattern properties of the stimulus, such as its contrast and spatial frequency content. Therefore, the nearness, extracted from optic flow estimated by insects, is expected to be entangled with properties of the textures of the environment. Visual-oriented tasks based on optic flow, such as collision avoidance, might, therefore, be a challenge. Several mechanisms of collision avoidance have been proposed based on behavioral experiments on various insect species [151, 90, 76, 114, 97, 108, 72, 133]. However, these models have not yet been shown to be functional under a wide range of conditions, or do not use optic flow measured by correlation-type elementary motion detectors.

In the present paper, we propose a model of collision avoidance based on EMDs which will be shown to be successful in various environments. The model of collision avoidance can be subdivided into three processing steps: (1) extraction of nearness from optic flow, (2) determination of a collision avoidance direction from the map of nearness estimations, i.e. where to go, and (3) determining a collision avoidance necessity, i.e. whether it is dangerous not to follow the collision avoidance direction. The nearness measurements will be shown to be proportional to a pseudo-norm of the optic flow, independent of the direction of motion, as long as the agent moves in a plane and has a spherical eye. The collision avoidance direction and necessity will be computed via spatial integration of the nearness. The collision avoidance algorithm will, firstly, be tested with geometrical optic flow, i.e. a measure of optic flow independent of object texture, to build a benchmark and show that optic flow information is sufficient to solve the problem. Then, EMDs will be used and the algorithm will be challenged in different environments. Finally, we will show that the collision avoidance algorithm based on EMDs can be coupled with a navigation direction in order to reach a given location without colliding with obstacles along the trajectory.

2.4 Results

2.4.1 Optic flow and relative nearness

When a distant object is approached at a high speed, the situation might be as dangerous as when a close object is approached at a slower speed. The relative nearness, i.e. the nearness times the agent's speed, can be seen as a measure of how soon the agent will collide with the object when the agent moves in the direction towards where the measurement was performed. This information is highly relevant for collision avoidance. Since the relative nearness is linked to the optic flow, the first step of the collision avoidance algorithm is to transform the optic flow into relative nearness. The translational optic flow, i.e. the optic flow experienced during the brief translatory phases of self-motion modeled after the intersaccadic intervals of insect flight, is determined jointly by the agent's self-motion and the three-dimensional structure of the environment. The independent extraction of these two parameters entangled in the optic flow field is challenging [78]. We will show that the three-dimensional structure of the environment can be extracted from translational optic flow if the translation is confined to a plane and the eye of the agent is spherical.

The optic flow field is a two-dimensional vector field, where each vector is the apparent velocity of the objects on the eyes of the agent. The optic-flow field experienced during translation results from the product of the relative nearness of objects in the environment and a factor depending on the angle between the direction of self-motion and the direction in which these objects are seen ("viewing angle"). A transformation removing the factor depending on the viewing angle is required to extract relative nearness from optic flow. The dependence of this factor on the viewing angle can be understood best when the relative nearness is constant for the entire visual field, i.e. when the agent is placed in the center of a sphere and moves in the equatorial plane. The optic-flow field for a spherical eye can be expressed for each point in the visual field in terms of the vertical flow component, i.e. the flow along the elevation, and the horizontal flow component, i.e. the flow along the azimuth. The horizontal flow component, experienced during a translation in the equatorial plane in the center of a sphere, increases from the front to the side (i.e. 90° away from the motion direction) and then decreases again towards the back (fig. 2.13B). The horizontal flow is independent of the elevation (see section 2.7.1,eq. (2.7)). Respectively, the vertical optic-flow component decreases from the front to the side, and then, increases again towards the back. By contrast, the vertical flow is not symmetric by rotation around the direction of motion and, therefore, depends on the elevation (see section 2.7.1,eq. (2.7)). It increases from the equator to the poles (fig. 2.13A). Therefore, the horizontal flow and the vertical flow have an antagonistic variation from the front to the back. Due to the assumption that the movement of the agent is constrained to the equatorial plane, the variation of the vertical flow with elevation does not depend on the direction of motion. This variation can, therefore, be corrected (see section 2.7.1, and fig. 2.13C). Interestingly, the sum of the horizontal flow squared and the corrected vertical flow squared can be shown to be independent of the viewing angle (see section 2.7.1,eq. (2.11)). The transformation will be called a retinotopically modified norm of the optic-flow field. When the agent does not move within a sphere, the result of this transformation will not be constant for every viewing angle, but equal to the product of speed (v) and nearness (μ), i.e. the relative nearness. The optic flow has two singular points, the focus of expansion (FOE) and the focus of contraction (FOC). At these two points, the result of the retinotopically modified norm of the optic flow will be null, independent of the nearness of objects.

The relative nearness can be extracted from the optic flow independent of the viewing angle, except for the FOE and the FOC. This problem can be solved by combining translational flow-fields arising from different directions of translational movement and, thus, with different FOCs and FOEs. However, an agent cannot easily obtain several translational flow fields centered at a given point in the world. On the other hand, the nearness to objects does not strongly differ in most realistic environments for two sufficiently close points in space. Therefore, let us consider an agent performing a translation composed of sub-translations in different motion directions, i.e. a combination of different forward and sideways motion components (fig. 2.1.1). Each sub-translation leads to an optic-flow field which has the retinotopically modified norm properties. The average optic-flow component obtained from the series of sub-translations also has the retinotopically modified norm properties, but does not have singular points (fig. 2.1.3). The agent can then compute the relative nearness to objects within its entire visual field by using the retinotopically modified norm of the averaged squared optic flow (fig. 2.1.4). When the object nearness for a viewing direction changes during the translation, the relative nearness map will be blurred. The longer the spatial lengths of the sub-translations are, the more the relative nearness map is blurred. This effect does not necessarily cause problems for collision avoidance, because the blurred relative nearness map still represents the overall depth-structure of the environment, though on a slightly coarser scale (fig. 2.2). Figure 2.2 shows the nearness map computed from the geometrical optic flow in an environment containing two objects. At a higher speed, the nearness map is blurred due to the integration of the geometrical optic flow over time.

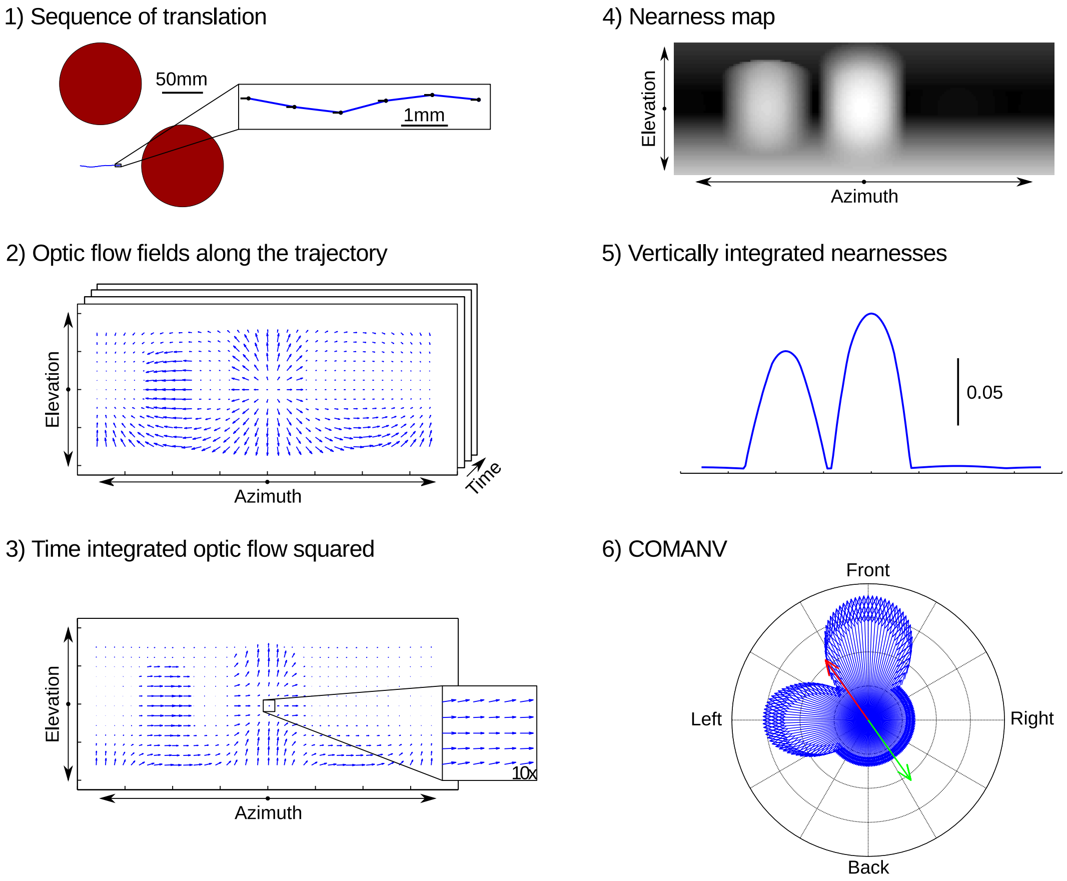

1) Sequence of translation
50mm
1mm
4) Nearness map
Elevation
Azimuth
2) Optic flow fields along the trajectory
Elevation
Azimuth
Time
5) Vertically integrated nearnesses
0.05
3) Time integrated optic flow squared
Elevation
Azimuth
10x
6) COMANV
Front
Left
Right
Back

Figure 2.1: Sketch of the algorithm from motion to CAD. 1) The motion of the agent consists of a series of translations in the null elevation plane. 2) Optic-flow fields along the trajectory contain FOEs and FOCs. 3) The time-integrated optic flow squared does not contain FOC and FOE. Inset is a 10× zoom at the mean motion direction of the agent. 4) Nearness map computed from time-integrated optic flow squared. 5) Nearness map averaged along the elevation. 6) Computation of the COMANV. Blue: representation of the vertically integrated nearness map in polar coordinates. Red: vectorial sum of the vertically integrated nearness vectors (COMANV). Green: vector directed opposite to the COMANV.

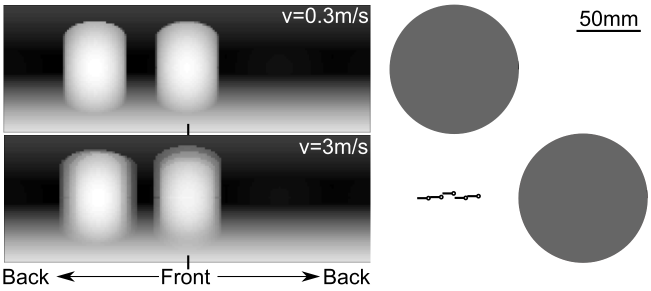


Figure 2.2: Blurred relative nearness of two cylindrical obstacles at high speed of the agent. Left panels: Nearness maps computed from optic flow experienced during translation at a speed of $0.3\mathrm{ms}^{-1}$ and $3\mathrm{ms}^{-1}$. Right panel: Trajectory at the speed of $3\mathrm{ms}^{-1}$ towards one obstacle. Black circle and black line represent the head and the body of the agent, respectively. Gray circles represent the objects seen from above.

2.4.2 Collision avoidance with geometrical optic flow

Once the relative nearness map is known, in which direction the agent should move to avoid a collision (collision avoidance direction, CAD) and how important it will be to follow this direction (collision avoidance necessity, CAN) need to be determined. Based on this information, the amplitude of the necessary saccade-like turn was determined. In order to establish a benchmark for the performance of this algorithm, it was developed, firstly, on the basis of the geometrical optic flow. Only the shape of the environment along the azimuth is required to perform collision avoidance for movements in a plane. Therefore, averaging the relative nearness map along the elevation does not lead to a loss in spatial resolution along the azimuth (fig. 2.1.5) and, thus, should not affect the collision avoidance performance. As will be shown below, this averaging was especially applicable when the relative nearness map was estimated on the basis of EMD responses (see CAD and CAN from EMD). The averaged relative nearness can be represented by vectors in polar coordinates with the argument of the vectors being the azimuth and their length the relative nearness averaged along the elevation (fig. 2.1.6). The vector sum of all averaged relative nearness vectors will be termed the Center-Of-Mass Average Nearness Vector (COMANV). It points towards the average direction of close objects in the environment (fig. 2.3). It may, therefore, be a plausible strategy to turn in the opposite direction to the COMANV, i.e. the CAD, to avoid obstacles. This zero-order approach is, to some extent, similar to the collision avoidance algorithm used by 3D range finder robots [105]. It may lead to suboptimal trajectories. A more optimal strategy would be to pick a direction without obstructions [80]. However, this strategy would require a reliable relative nearness map provided by local self-localization and mapping [15].

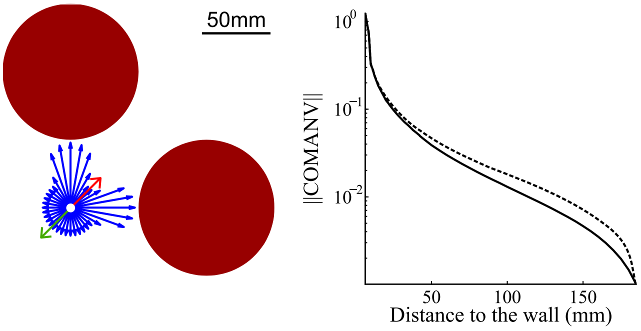


Figure 2.3: Direction and norm of COMANV. Left panel: Direction of the COMANV. Blue, red and green vectors are nearness vector, +COMANV, and -COMANV, respectively. Red disks represent the objects. (The norms of the vectors have been scaled.) Right panel: Norm of COMANV as a function of the distance to the box wall. Box height: 390mm (solid line) and 3900mm (dash line).

The argument of the COMANV provides the agent with a direction to avoid a collision by pushing it away from obstacles. Figure (fig. 2.4A) shows a closed-loop simulation of collision avoidance in a box. The agent trajectories converge at the center of the box. Indeed, if an agent is pushed away from obstacles at every location in an environment, it ends at a point in the environment equilibrating object distances. However, collision avoidance behavior is not necessary if the obstacles are sufficiently far from the agent. To allow the agent to assess when it has to avoid an object, a measurement of CAN is required. The argument of the COMANV, the CAD, has been used, so far, to compute the saccade amplitude and, thus, to determine the agent's new direction of motion. However, the norm of the COMANV also has interesting properties: It has the same unit as the relative nearness, i.e. the inverse of a time. Hence, the norm of the COMANV can be regarded as a measurement of the CAN: the larger the norm of COMANV, the larger is the necessity for the agent to make an evasive behavioral response. To assess the relationship between the distance to objects and the norm of the COMANV, the relative nearness map has been extracted at different distances to the wall. Since the apparent size of the objects might also affect the norm of the COMANV, different wall heights were used (fig. 2.3). The norm of COMANV increases with both the apparent size of an object and the nearness to it. The apparent size of the object has a smaller effect on the norm than the nearness. Thus, the norm can be used as a measurement of CAN.

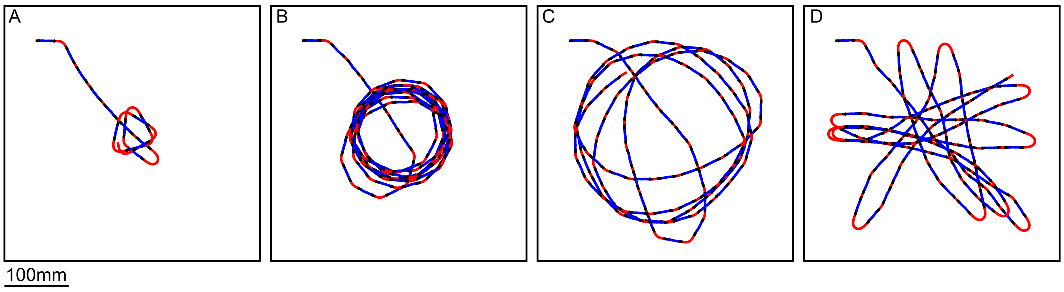


Figure 2.4: Close-loop simulations of trajectories of the agent equipped with the collision avoidance algorithm in a cubic box. Blue and red lines are intersaccades and saccades, respectively. A) The saccade amplitudes were computed such that the agent moves in the CAD after the saccade. B, C and D) The saccade amplitudes were computed such that the agent moves in a direction corresponding to only a fraction of the CAD after the saccade. The fraction of the CAD was computed with a sigmoid function, parameterized by a gain and a threshold, of the CAN. B) Gain= 2, Threshold= 1.6. C) Gain= 2, Threshold= 3.2 D) Gain=10^6, Threshold= 3.2.

Depending on the amplitude of the CAN, the agent may have one of two behavioral options. It should turn via a saccade toward the CAD calculated if the CAN is sufficiently high. Alternatively, it should continue moving straight if the CAN is smaller than a critical value.

However, the CAN does not necessarily affect the behavior of the agent in an all-or-nothing fashion, i.e. making a turn according to CAD or making no turn at all. Rather, a kind of compromise may also be possible. Since the CAN is a continuous variable, the agent may turn, via a saccade, towards a direction which is a compromise between CAD and the previous direction of motion. The compromise can be modeled as a weight given by a sigmoid function of the CAN (see Materials and Method). The saccade amplitude is then the product of this weight and CAD. The sigmoid function of CAN is parameterized by a threshold and a gain. The gain controls how much the saccade amplitude corresponds to CAD. A high gain will approximate a behavior with two distinct states: "turn, by a saccade, toward CAD" or "continue moving straight" (fig. 2.4D). A small gain will, however, generate a smooth transition between the two behaviors, modeling a decreasing saccade amplitude with decreasing CAN (fig. 2.4C). The threshold determines the border between the zone in the environment where saccade amplitudes are mainly driven by CAD, i.e. collision avoidance is necessary, and the zone where the saccade amplitudes are mainly driven by the previous direction of motion, i.e. collision avoidance is not necessary (fig. 2.15). The effect of the threshold can be seen by comparison figure (fig. 2.4B) and figure (fig. 2.4C).

In summary, the collision avoidance algorithm uses the COMANV to determine the collision avoidance direction CAD and to change the behavior of the agent. The algorithm can be subdivided into five steps.

1. Extract relative nearness map from optic flow during an intersaccade composed of a mixture of different forward and sideways motion components.

2. Compute the COMANV from the relative nearness map.

3. Extract the CAD (i.e. $arg(COMANV)$) and
   the CAN (i.e. $||COMANV||$).

4. Compute the saccade amplitude from CAD and CAN.

5. Generate the saccade

2.4.3 CAD and CAN from EMD

The collision avoidance algorithm has been designed on the basis of geometrical optic flow and operates successfully on this basis. However, the properties of the optic flow, as measured by EMDs, differ considerably from the geometrical optic flow. Several model variants of EMDs have been developed (e.g. [5, 20, 134]). We used a rather simple EMD model version (similar to [89, 68]) in this study, because we wanted to test whether collision avoidance can already be accomplished by the basic correlation-type motion detection mechanisms with as few model parameters as possible. The drawback with EMDs, at least from the perspective of velocity estimation, is that their responses do not only depend on the velocity of the retinal images, but also on their contrast and other textural properties [42, 17]. Therefore, it is not clear in advance whether the collision avoidance algorithm, as described above and being successful based on geometrical optic flow, will also work with optic-flow estimates based on EMDs. The collision avoidance algorithm based on EMDs will be tested in two steps. In this section, we will assess to what extent the COMAMV derived from EMD measurements matches the COMAMV based on geometrical optic flow. In the next section, we will test the collision avoidance algorithm equipped with EMDs under closed-loop conditions.

As the first essential step, the relative nearness map is extracted from EMD responses. The texture dependence of the EMD measurements is somewhat reduced by spatial averaging along the elevation of the visual field (see above). The consequences of this averaging are shown in figure 2.5 for an exemplary simulation (see also fig. 2.19). The agent performed a translation inside a box covered with natural images of grass. The relative nearness map obtained from EMDs does not only depend on the geometrical nearness, but also on the texture of the wall (fig. 2.5). Although integration along elevation reduces the pattern dependence to some extent, the integrated relative nearness map still contains "fake holes" (e.g. those that result from extended vertical contrast borders; fig. 2.5). These “fake holes” may mislead the agent when looking for relative nearness lower than a certain threshold. As the second step of the collision avoidance algorithm, the COMANV and the CAD have to be computed from the EMD-based relative nearness map. Ideally, the CAD based on EMDs should coincide with the one determined from the geometrical optic flow. The CADs determined in both ways are the same for the example shown in figure 2.5. To assess whether this finding also generalizes to other environments, the simulations were extended to cubic boxes with the agent translating parallel to one wall of the box from different starting positions. The angle between the CAD based on EMDs and the one based on geometrical optic flow were computed for every starting position. Figure 2.6 shows that the CADs based on geometrical optic flow are similar to the CADs based on EMDs if the agent is not too close to the wall and not too close to the center of the box (fig. 2.16 and fig. 2.16). Moreover, the higher the walls of the box are, the more CADs determined in the two ways coincide (fig. 2.6).

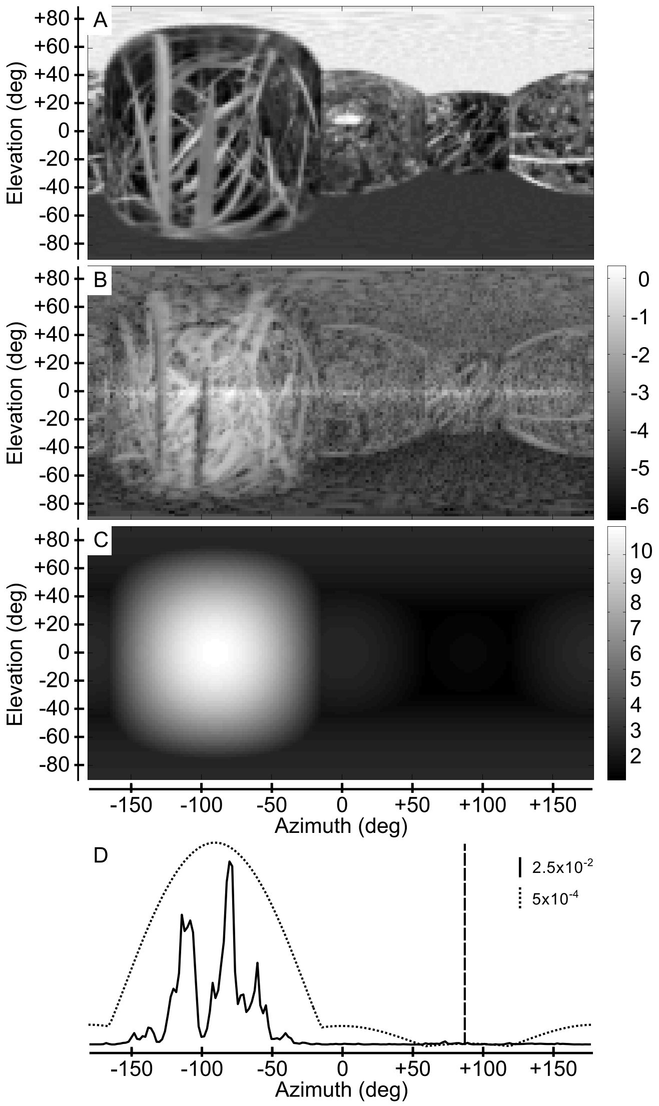


Figure 2.5: EMD responses and nearness map. A) Panoramic view of the environment, consisting of a cubic box covered with a natural grass texture, from the location where the nearness map was computed (front is azimuth 0°). B) log-scaled nearness map computed on the basis of EMD responses. C) Nearness map at the same location computed from the geometrical optic flow. D) Vertically integrated nearness map extracted respectively from EMD responses (solid line) and geometrical optic flow (dotted line). The vertical dashed line shows the CAD computed from the vertically integrated nearness map based on EMD responses. The direction matches the one computed with geometrical optic flow.

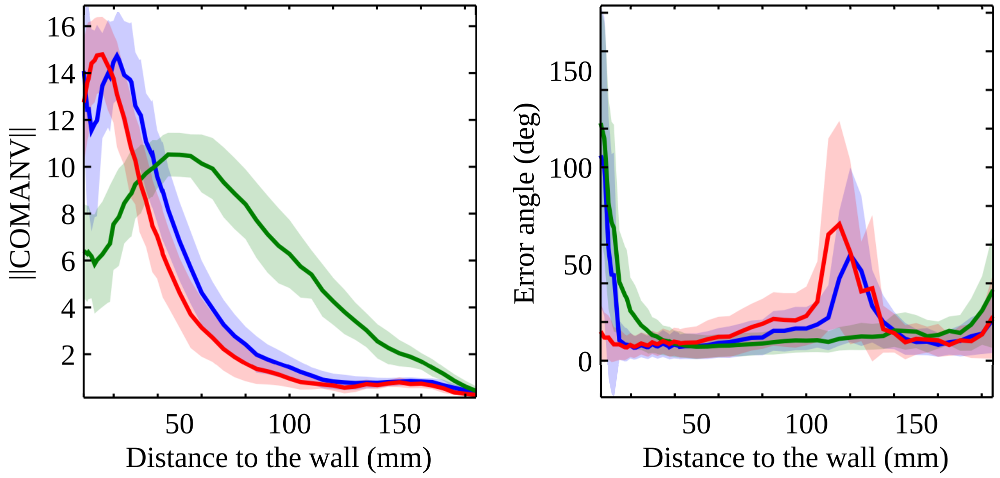


Figure 2.6: COMANV versus wall distance in a cubic box. The box (height: 390mm) was covered with random checkerboard patterns of either 1mm (blue) or 4mm (green). Red: the box had a height of 3900mm and was covered with a 1mm random checkerboard pattern. Left panel shows the norm of the COMANV computed on the basis of EMD responses. Right panel shows the angle between the COMANV computed from EMD responses and the control based on geometrical optic flow. Thick lines and shaded area represent the mean and the standard deviation, respectively, computed at a given distance from the wall.

The collision avoidance algorithm requires an increasing length of the COMANV with an increasing relative nearness to objects, in order to provide a good estimate of the CAN. The variation of the length of the COMANV with the distance to the wall has been studied with geometrical optic flow in a cubic box. The same environment has been tested with EMDs, but with several different patterns. Similar to the simulation of CAD, the length of the COMANV has been computed for several points in the corridor. Figure 2.6 shows the dependence of the CAN on the nearness to the wall. As expected, the wall texture changes the CAN, as does the nearness to the wall.

However, the CAN is still an increasing function of the nearness to the wall as long as the agent is not too close to the wall (fig. 2.16 and fig. 2.18). Therefore, the norm can be used as a reasonable estimate of CAN in this range.

2.4.4 Collision avoidance with EMD

As shown above, information about the three-dimensional shape of the environment around the agent derived from EMD responses leads to appropriate CADs and a reliable estimate of CAN, as long as the agent is not too close to an obstacle, such as the wall of a flight arena. These results were obtained in open-loop simulations. Since EMDs use temporal filters, their responses also depend to some extent on the signal history. The time constant of the low-pass filter in one of the EMD branches is 35ms, i.e. in roughly the same range as the time between subsequent saccades of insects (20 to 100ms in flight arenas [126, 76] and 50 ms in our simulation). Thus, the EMD response during a given intersaccade also depends on the signals generated during the previous saccade, resulting in a somehow disturbed nearness map. Taking all this into account, open-loop simulations do not allow the collision avoidance performance under closed-loop conditions to be predicted.

A relatively simple and commonly used environment for experiments on collision avoidance behavior of insects are cubic or cylindrical flight arenas [126, 152, 96, 107]. In such an environment, the agent has to avoid only the wall. Thus, the task is easier to accomplish than if objects are also present. Figure 2.7 shows closed-loop simulations in boxes covered with six different wall patterns (see also fig. 2.20). The agent is able to avoid collisions for all wall pattern conditions except the random pattern with relatively large (35mm) pixels. However, the area covered by the flight trajectories varies tremendously with the pattern. The saccade amplitude depends on the gain and the threshold, which parameterize the sigmoid function of the CAN. These parameters have been kept constant for the different pattern conditions. By adjusting the threshold and the gain individually for each pattern condition, collision avoidance may be successfully performed by the agent, as long as the CAN increases with the nearness to objects and the CAD points away from obstacles (fig. 2.16 and fig. 2.17).

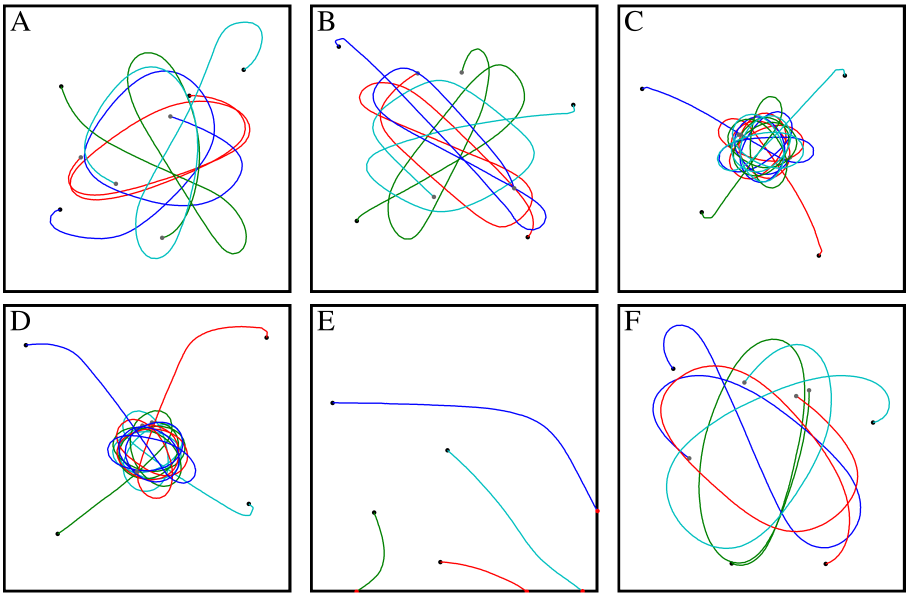


Figure 2.7: Trajectories of the agent with a collision avoidance system based on EMDs in a box $(40 \times 40 \times 40\text{cm})$ covered with different patterns (seen from above). Trajectories with four different starting positions are shown (see S12 for different starting position). The simulation time was 10sec or until the agent crashed. Walls of the box are covered with a natural pattern (A), a 1mm random checkerboard (B), a 4mm random checkerboard (C), an 8mm random checkerboard (D), a 35mm random checkerboard (E), and a random pattern with $1/f$ statistic (F). The gain and the threshold of the weighting function was 2 and 4, respectively, for all cases.

Until now, we have used a rather simple environment compared to those experienced by an agent under more natural conditions. Objects were added to the flight box to increase the complexity of the collision avoidance task. The objects had different sizes, shapes and textures. They were camouflaged, i.e. covered with similar patterns to the background, to increase the difficulty for the collision avoidance algorithm. Such situations occur frequently in nature, e.g. when a particular leaf is located in front of similar leaves. To discriminate such an object and to avoid a collision with it, relative motion on the eyes induced by self-motion of the animal and, thus, the relative nearness to the object as obtained from optic flow is the only cue available. Up to four objects were inserted into the box and covered with the same pattern as the walls. The agent was able to avoid collisions successfully, even in the box with four objects (fig. 2.8, fig. 2.21 and fig. 2.22). However, collisions were observed in boxes containing two and four objects each covered with a 4mm random checkerboard pattern (fig. 2.22).

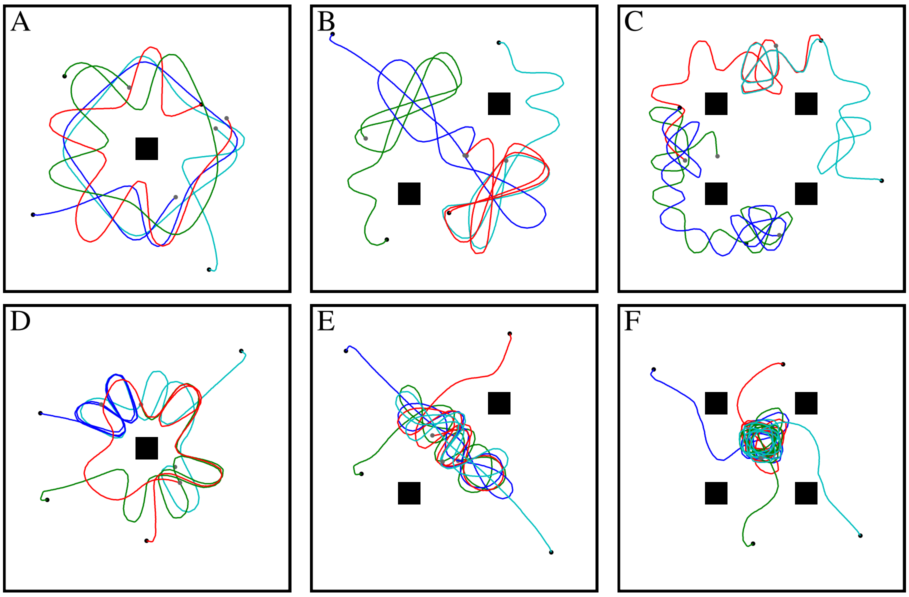


Figure 2.8: Trajectories of the agent with a collision avoidance system based on EMDs in a box $(40 \times 40 \times 40\text{cm})$ containing up to four objects and covered with different patterns (seen from above). The pattern on the objects and wall were 1mm and 4mm random checkerboards for the top and bottom panels, respectively. A, D) One object in the center of the box. B, E) Two objects on one diagonal. D, F) Four objects on the diagonals. The objects were vertical bars with a quadratic base with a side length of 3cm and a height of 40cm. The gain and the threshold of the weighting function was 2 and 4, respectively, for all cases.

2.4.5 Collision avoidance with a goal direction

Agents in natural environments may have to face even more complex situations than those tested so far, such as avoiding collisions in a cluttered environment with many objects. A forest is an example which contains many trees, i.e. many objects to avoid. Two different artificial environments with 35 randomly placed objects have been used to test the collision avoidance performance in cluttered environments. Again the objects were camouflaged with the same texture that covered the floor and the confinement of the environment. The agent tended to stay in a relatively small area of the environment where the walls were sufficiently distant (fig. 2.9). Hence, an agent equipped with only a collision avoidance algorithm did not travel through the artificial forest.

This task was only accomplished if the collision avoidance algorithm was slightly modified to support a goal direction. The saccade amplitude, so far, was the result of a compromise, based on the CAN, between the CAD and the tendency to keep the previous direction of motion. If this direction was replaced by the direction toward a goal, the saccade amplitude became a compromise between the CAD and the goal direction, depending on the CAN. When the CAN was below the threshold, as parameterized by the sigmoid function of the CAN, the saccade amplitude is mainly driven by the goal direction. By contrast, when the CAN is higher than this threshold, saccades would be mainly driven by the CAD. The significance of the CAN could clearly be seen for trajectories close to objects. Far from the object, the agent moved toward the goal, but when it came close to the object, saccade amplitudes tended to be driven by the collision avoidance algorithm, pushing the agent in the opposite direction (fig. 2.10). When the goal was located at the other end of the corridor, the agent was efficiently, i.e. without making many detours, and reliably, i.e. with a low rate of collisions, able to reach the goal (fig. 2.10,section 2.7.2,fig. 2.23).

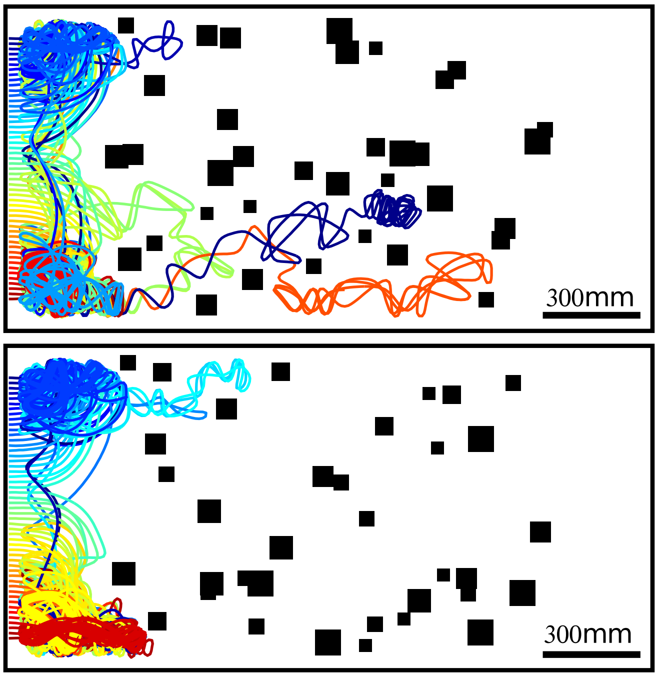


Figure 2.9: Trajectories of the agent equipped with an EMD-based collision avoidance system in two different cluttered environments with objects and the walls covered by 1mm random checkerboard patterns (seen from above). Fifty-one starting positions were tested, and simulations were run for 100sec or until the agent crashed. The trajectories are color-coded depending on their starting position. Objects are indicated by filled black squares. The gain and the threshold of the weighting function was 2 and 4, respectively, for all cases. (see also section 2.7.4)

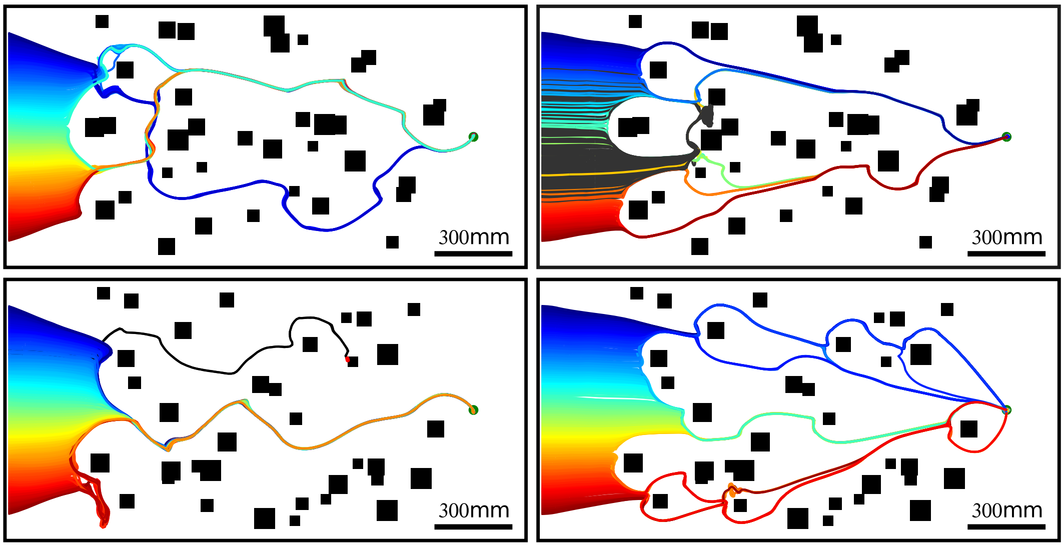


Figure 2.10: Trajectories of the agent equipped with an EMD-based collision avoidance system, but also relying on the goal direction, in two different cluttered environments with objects and walls covered by 1mm (right column) or 4mm (left column) random checkerboard patterns. The goal is indicated by the green dot. Two hundred one starting positions were tested and simulations were run either for 100sec (gray lines, i.e. dead-end), until the goal was reached (colored lines) or until a crash occurred (black lines). Note that the individual trajectories converge to only a small number of distinct routes. Apart from taking the goal direction into account, the simulations, parameters and environments are identical to those used for fig. 2.9. (see also section 2.7.5,section 2.7.6)

2.4.6 Route similarity in cluttered environment

The number of different trajectories close to the goal location in cluttered environments is much lower than the number of starting conditions. Therefore, agents, starting from different locations, but heading towards the same goal location, have trajectories converging to similar routes. This behavior is not only a consequence of the walls confining the cluttered environment. Indeed, a similar behavior is observed in a cluttered environment without confining outer walls (fig. 2.23). In order to classify the similarity of the different trajectories, each trajectory was first simplified into a sequence based on the position of the agent relative to the objects in the environment. Trajectories sharing the same sequence formed one class, i.e. a route [see material and methods]. In the first (resp. second) cluttered environment, 8 (resp. 11) and 4 (resp. 3) distinct routes were found for objects and walls covered with 1mm and 4mm random checkerboard patterns, respectively (fig. 2.24, fig. 2.25, fig. 2.26, fig. 2.27)

The routes followed by the agent may be determined by its starting position, i.e. neighboring starting positions may lead to the same route. Indeed, when an agent approaches an object from the right (resp. left), it tends to avoid it by a left (resp. right) turn. This "decision" will be taken for every obstacle along the trajectory taken by the agent, but each "decision" depends sensibly on the position of the agent relative to the object and the goal location. Therefore, the route followed by an agent may be sensitive to the starting position. Figure 2.11 shows that neighboring starting locations may lead to different routes.

The number of different routes close to the goal location is lower than the number of possible routes in a given environment. This indicates that, on the one hand, routes starting at different locations tend to converge into common routes and, on the other hand, different routes may share similar parts, i.e. sub-routes. As a measure of similarity between routes, the number of single sequence elements

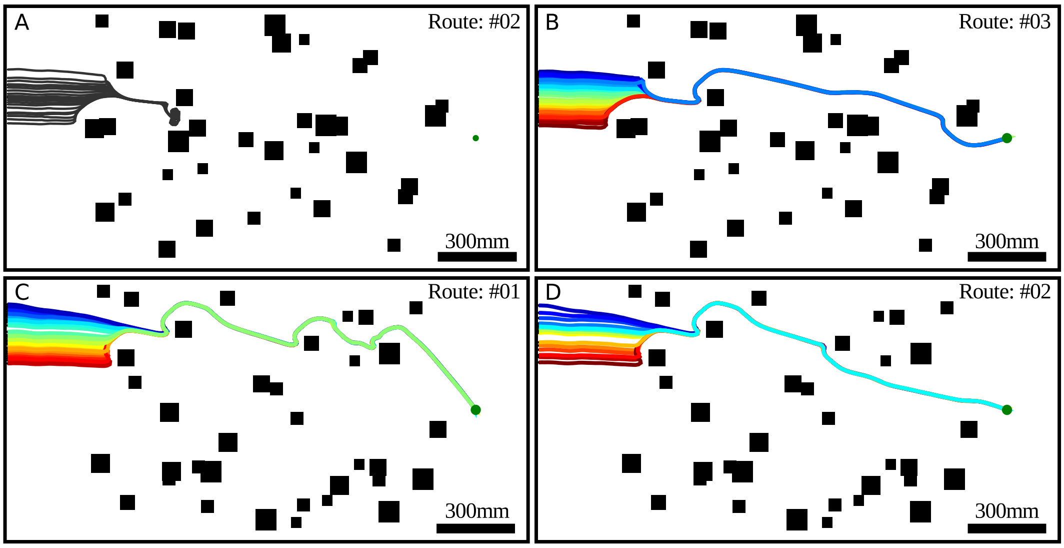


Figure 2.11: A selection of the routes shown in figure 2.10 for the two environments. Although the area of starting positions greatly overlap for a given environment, the trajectories converge on two different routes (compare A with B, and C with D). The simulations, parameters and environments are identical to those used for the right panel of fig. 2.9.

differing between two routes was used. Routes may be very similar to each other with less than five different single sequence elements (e.g. compare route #8 and route #10 in fig. 2.26). Different routes, therefore, share similar sub-routes. This indicates that a rather small number of locations exist where the agent "decides" to take a particular sub-route, e.g. avoid an object towards the left or the right, respectively.

The collision avoidance algorithm is affected by the pattern covering the walls and the objects in the environment. This dependency may lead to different routes. Therefore, routes obtained in an environment with given object locations have been compared after changing the texture of the environment to pinpoint texture-dependent effects. Interestingly, certain classes of routes are indeed the same for the different patterns, e.g. the second route for a 1mm random checkerboard texture matches the third route for a 4mm random checkerboard texture (fig. 2.12). Three routes (resp. one) out of the four (resp. three) routes for the 4mm random checkerboard textures are indeed found also for the 1mm random checkerboard textures covering the first (resp. second) environment. This finding indicates that, despite pronounced pattern effects resulting from the properties of EMDs, the performance of the collision avoidance algorithm is, on the whole, quite stable and, to a large extent, depends on the spatial structure of the environment.

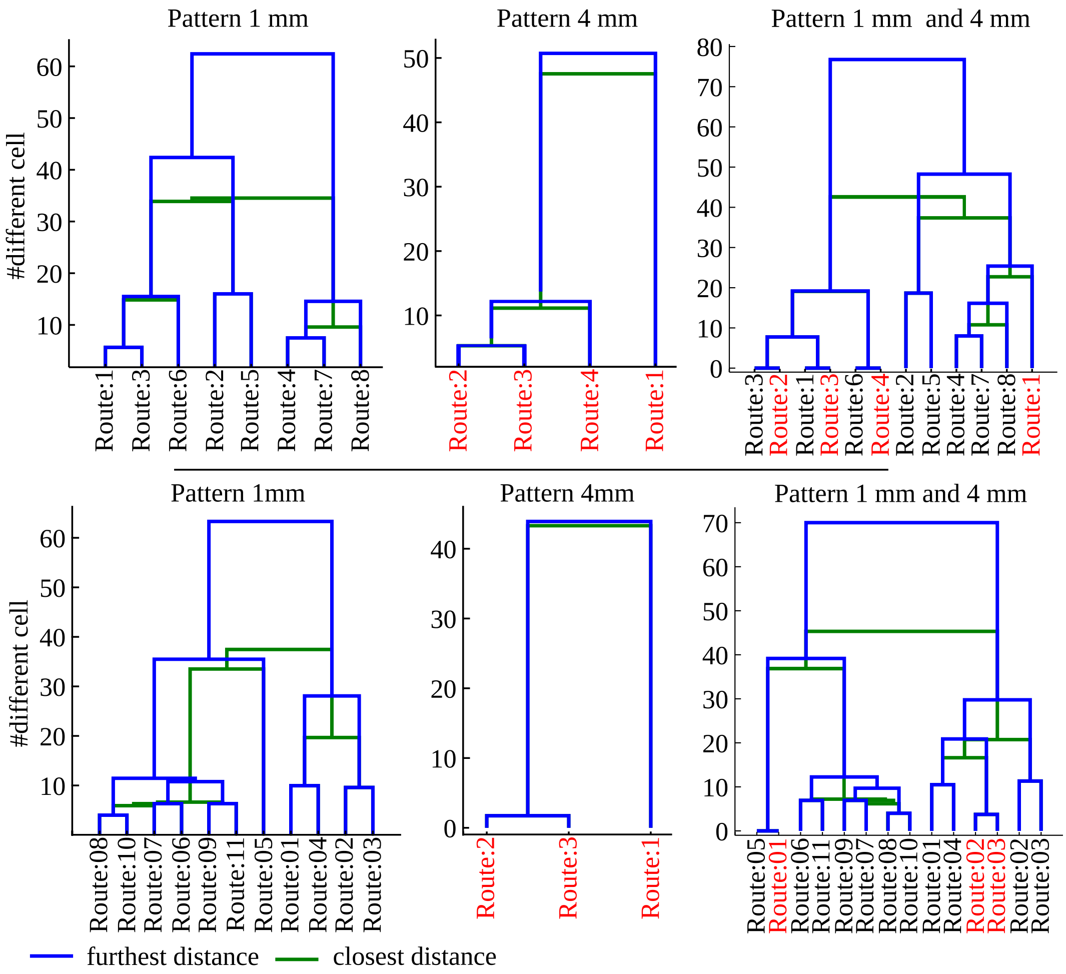


Figure 2.12: Dendrogramm of route similarity for the two different cluttered environments, top and bottom row, respectively. The routes followed by the agent (see fig. 2.9) are characterized by a cell sequence. Here, each cell is a triangle formed by neighboring objects. The route similarity is defined by the number of cells not shared by routes. First and second columns, path-similarity for 1mm and 4mm random checkerboard patterns, respectively. Third column, path-similarity across patterns. Note that identical routes are found for different patterns: for example, route #3 and route #2 in the first environment (top row) are identical to those for the environment covered by 1 mm and 4 mm random checkerboards, respectively. The routes are shown in fig. 2.24,fig. 2.25,fig. 2.26,fig. 2.27.

2.5 Discussion

We developed a model of collision avoidance based on correlation-type elementary motion detectors (EMDs), which accounts for the fundamental abilities of insects to avoid collisions with obstacles in simple and also complex cluttered environments. The model has been developed with geometrical optic flow, showing that apparent motion as the only source of information about the depth structure of the environment is sufficient to accomplish collision avoidance tasks. The transfer from geometrical optic flow to bio-inspired correlation-type EMDs has been shown to be successful in a range of environments. Moreover, when the collision avoidance algorithm is coupled to a goal direction, the coupling between goal-directed behavior and collision avoidance allowed the agent to move through cluttered environments, even if the objects, floor and the background were covered with the same texture. Interestingly, the trajectories traveled in the cluttered environment were very similar, irrespective of the starting condition.

In the following, we discuss three key aspects of this work: (1) the duration of the intersaccades and their shape and, thus, the conditions under which spatial information about the environment can be obtained; (2) the nearness measurements obtained from EMDs; and (3) how the convergence of individual trajectories into a small number of routes can arise from collision avoidance while the agent is heading towards a goal.

2.5.1 Changes in intersaccadic translation direction

The assumptions underlying our algorithm to extract a nearness map from optic flow are: (i) a spherical eye, (ii) a translation phase combining several directions of motion (i.e. a mixture of different forward and sideways motions), and (iii) all movements of the agent take place in the null elevation plane. The second assumption is required in order to average out the characteristic singularities in translational optic-flow fields, i.e. the FOE and the FOC, by integrating the optic-flow amplitudes obtained during translations in slightly different directions (i.e. a mixture of forward and sideways motions). Indeed, the direction of translational movements between saccades of flying insects is not always constant with respect to the orientation of the body long axis. This means that the relationship between the forward and sideways motion components may change systematically even between two consecutive saccades. Extreme examples in this regard are shown in figure 3 of [126]. However, more moderate continual changes in the ratio between forward and sideways translational components occur, as a consequence of inertia, after virtually all saccades, with the strength of these changes depending on saccade amplitude [126, 76]. Therefore, flying insects could average the optic flow generated on the eyes during these continual intersaccadic changes in flight direction and then extract the relative nearness to determine the direction and amplitude of the next saccade. However, as a consequence of the inevitable time constants of the motion detection system, the EMD responses following a saccade might also be affected by the rotational optic flow of the previous saccade. Although the rotational part of the optic flow could, in principle, be removed if the angular velocity of the agent was known, this transformation would not be straightforward on the basis of motion measurements based on EMDs [115], given their dependency on the texture of the environment.

The integration of the optic-flow amplitudes along intersaccades is also important to decrease the dependency on the texture in the environment characteristic of EMD responses. For simplicity, the intersaccade duration has been kept constant in our simulations, although in free-flying flies, it was found to vary from 20 to 100ms [126, 76]. By increasing the duration of an intersaccade, the dependency on the texture in the environment can be further decreased if the integration time is increased accordingly. An increase in integration time has similar effects to increasing the extent of spatial integration along the direction of motion [102]. However, the longer the duration of the intersaccadic translation is, the more blurred the relative points of nearness are, as shown in figure 2.2. On the other hand, the intersaccade duration might be linked to the collision avoidance necessity. Indeed, collision avoidance may be unnecessary when no obstacles are encountered, as shown by the closed-loop simulations of goal-oriented behavior (fig. 2.9 and fig. 2.10). If the collision avoidance necessity is low, long intersaccades are possible. However, if the collision avoidance necessity is high, short intersaccades followed by an evasive turn are required.

The third assumption of the algorithm of nearness calculation, i.e.

that the agent only moves in the null elevation plane, is certainly not exactly satisfied in free-flying insects. However, during most flight manoeuvres, changes in height occur to a much smaller extent than changes in the horizontal plane. Nonetheless, if an agent moves in another direction than in the null elevation plane and estimates the nearness with our algorithm, its estimation will have an error proportional to the upward component (section 2.7.1). As long as this component is small, the estimated nearness map will not be strongly affected. We are currently investigating how our algorithm for nearness estimation can be extended to arbitrary movements in three dimensions.

2.5.2 Nearness from measured Optic flow

We have shown that relative nearness can be extracted from geometrical optic flow. However, if the nearness algorithm receives its input from correlation-type EMDs, complications arise from the dependence of the EMD responses on (i) the contrast of the stimulus pattern, (ii) its spatial frequency content, and (iii) the fact that it is not related to velocity in a linear way, but first increases, reaches an optimum value and then decreases again [16]. Indeed, these dependencies are somehow reflected in the extracted relative nearness (see, for example, fig. 2.5D). The EMD has a quadratic response dependence on contrast if no additional nonlinearities are inserted in its input lines [117]. Indeed, it has been recently shown that the norm of the EMD response is correlated to the nearness times the local contrast or, in other words, EMDs have been shown to respond best to the contrast contours of nearby objects (see also, fig. 2.5) [131]. The dependency on contrast can, in principle, be reduced by applying a nonlinearity before the multiplication stage of an EMD [20, 5, 42]. Nonlinearities tend, however, to complicate the mathematical analysis of the EMD response. Therefore, we have chosen to use a simple EMD version. Moreover, we wanted to test how well the collision avoidance algorithm performs on the basis of the basic correlation-type motion detection mechanism with as few model parameters as possible.

The average output of an EMD depends on the temporal frequency of a motion stimulus (i.e. the ratio of angular velocity and wavelength of its spatial Fourier components) rather than its real velocity [42, 17]. To extract the real angular velocity from EMD responses, the dependency on temporal frequency needs to be reduced. Plett et al. suggested using the spatial power spectrum of a panorama to extract the angular velocity from EMD responses during rotation [115]. However this transformation is not suitable for translation, because the spatial frequency content of the panorama is not related unambiguously to the temporal frequency observed during translation. Therefore, it is not easily possible to compensate for these dependencies.

Nearness extracted from the nonlinear, but monotonic response range of the EMDs may lead to a distorted representation of the depth-structure of the environment. Nevertheless, larger EMD responses still correspond to greater nearness. Ambiguous nearness estimates will arise for the nonlinear and non-monotonic, i.e. the ambiguous, response range of the EMD. Therefore, close objects leading to large retinal velocities might be mistaken for far objects. In the context of nearness extraction during translation, this problem will arise for objects on the lateral side of the agent. To avoid this problem, an agent has two possibilities: reduce its translational speed or extract nearness information only in the more frontal parts of the visual field. The agent in our simulations was moving at a relatively slow speed. Therefore, this ambiguity was not observed. Moreover, flying insects have been concluded to use the frontal part of their visual field to compute saccade amplitudes (flies, [76]) and to reduce their flight speed when the clearance to objects in the environment gets small (e.g. bumblebees [9]; flies: [76]). It is, however, unclear how flying insects compute the CAD only taking into account motion measurements in the frontal part of the visual field.

A robot does not need to estimate optic flow with EMDs. The estimation can be carried out with image-based methods, e.g. the Lucas Kanade algorithm [91], or event-based methods [124]. Image-based methods are, however, time-consuming and, therefore, are not really suitable for applications in real-time. However, event-based flow-field detectors are fast and reliable. Our algorithm to extract nearness from optic flow is only local, i.e. it only uses the optic-flow vector in a given viewing direction to compute the relative nearness in this direction. Therefore, our collision avoidance algorithm could be easily coupled with EMDs to determine the nearness around the robot in real-time as a cheap alternative to a 3D laser rangefinder.

2.5.3 Goal-directed collision avoidance and route following

Both biological and technical agents often need to reach a goal, e.g. their nest in the case of many insects or a charging station in the case of a robot, without colliding with the objects along their trajectory. This goal direction in a real world could be provided by path integration and visual navigation [166] or, in the case of a technical agent, by GPS. Therefore, when the agent is in cluttered environments, it needs to somehow integrate the goal direction and the collision avoidance direction. Our collision avoidance algorithm has been shown to support a goal direction by using CAN, leading to a behavior that represents a kind of compromise between collision avoidance and reaching the goal. Interestingly, the trajectories of the agent, even in complex cluttered environments, tend to converge on a limited number of distinct routes largely independent of the starting position (see fig. 2.10) when coupled to a goal direction. The appearance of routes is not a unique property of our collision avoidance algorithm. Similar trajectories are also followed by an agent with different control strategies and different collision avoidance algorithms (e.g. see figure 6 in [114], and figure 4 in [133]). Ants perform in a similar way and also follow similar trajectories in cluttered environments when returning to their nest (e.g. [165, 79, 181, 94]). We could show that this type of behavior can be explained in a relatively simple way by combining our local collision avoidance algorithm only with an overall goal direction. By contrast, the similarities of trajectories of ants have often been interpreted within the conceptual framework of a route-following paradigm. According to this paradigm, the agent is assumed to store local information along the trajectory during an outbound run (e.g. leaving the nest), which will be used to determine the direction to follow during an inbound run (e.g. returning to the nest) [6, 10]. In our simulations, we observed that trajectories which may be interpreted as resulting from route-following could, alternatively, arise just from a collision avoidance algorithm coupled to a goal direction. Therefore, part of the route-following behavior observed in insects could be a consequence of a collision avoidance algorithm. Hence, the route-following direction does not need to be determined at every point along the trajectory, but its determination may be sparsely spaced.

The routes to the goal followed by our agent depend on the starting location, i.e. neighboring starting locations may lead to different routes (fig. 2.11). The same behavior has been observed in ants [94]. However, the different routes are not equivalent in term of efficiency. Indeed, in figure 2.11A and fig. 2.11B, the route #2 is dead-end, and route #3 reaches the goal. An agent may need to use the most efficient route, i.e. add waypoints in the environment indicating which route to follow. The routes #2 and #3 (resp. #1 and #2) shown in figure 2.11, for example, could be merged by adding only one waypoint just where the two routes emerge. Therefore, insects may place a small number of well chosen waypoints in the environment to prevent the dead-end problem observed in our simulations fig. 2.10 and possibly to select the most efficient routes (fig. 2.14) without requiring a large memory.

2.6 Materials and Methods

The simulations have been inspired by the flight and gaze strategies of flies and other insects. The trajectories have, thus, been separated into saccadic and intersaccadic phases. The intersaccade, i.e. a translation of 50ms at 0.1ms^{-}1, is used to gather information about the depth-structure of the environment from optic-flow measurements and to provide a CAD. The CAD controls the amplitude of the saccade following the translation phase. The yaw velocity of the saccade is computed according to a Gaussian velocity profile. This profile fits the experimentally determined template of saccades [126, 89]. Therefore, the saccade duration and the peak yaw velocity are given by the template and the saccade amplitude. Closed-loop simulations were performed in environments of different complexity, which were covered with different textures. The simulations were mainly written in MATLAB, with part of the code written in C for computational efficiency, and run at a sampling rate of 1kHz.

2.6.1 The simulation environment

The environments used in our simulations have been inspired by previous behavioral experiments on flies and bees. Schilstra and van Hateren used a cubic box with an edge length of 40cm [126, 66]. This box was covered with natural images on its side walls and with a black/gray and a white/gray irregular pattern on the floor and the ceiling, respectively. The side walls of the box were covered alternatively by 1mm, 8mm or 35mm black and white random checkerboard patterns to investigate the effect of the texture in another set of simulations. Simulations were also done in boxes with one, two or four obstacles covered with 1mm black and white random checkerboard patterns to make the collision avoidance task more difficult. The obstacles had the same height as the box and a square cross-section with an edge length of 30mm. The wall of the box had the same texture as the objects. The obstacles were placed at positions as shown in fig. 2.8; they were vertical bars of 40cm height and a quadratic base with a side length of 3cm.

Cluttered environments with 35 obstacles of different sizes were used in another set of simulations. Every wall in the environment (object sides and corridor walls) was covered by either a 1 mm or a 4 mm random checkerboard pattern. The obstacles had a square base and a height five times their side length, and were randomly positioned in the inner part of a $2000 \times 1000 \times 400$mm box. Two environments were selected based on the homogeneity of the obstacle positions. The first environment was composed of five objects with an edge length of 80mm, five with an edge length of 72mm, ten with an edge length of 64mm, five with an edge length of 56mm, five with an edge length of 48mm, and five with an edge length of 40mm. The second environment was composed of the same number of objects, except that ten objects with an edge length of 72mm and five objects with an edge length of 64mm were used.

2.6.2 Visual system and motion detection

Once the environment had been created, a panoramic view from any position within the environment could be generated and the distance to objects determined. The set of vertices X,Y,Z that define an object, as well as the floor and the ceiling, were translated to the current position of the agent. An environment map was rendered using OpenGL. The input image was sampled by Gaussian-shaped spatial low-pass filters ($\sigma = 2°$). The output of these filters formed the input to the photoreceptors that were equally spaced at $2°$ along the elevation and azimuth of the eye. The array of photoreceptors formed a rectangular grid in the cylindrical projection with 91rows and 181columns. The temporal properties of the peripheral visual system was modeled as a temporal filter with a kernel that was derived from an electrophysiological analysis of the responses of second-order visual interneurons in the fly visual system to white-noise brightness fluctuations [73, 75]. The filter kernel is a kind of temporal band-pass filter with a DC component (for a formal description, see [89]). The outputs were, furthermore, filtered with a first-order temporal high-pass filter (time constant 20ms) to remove the DC component. The filtered outputs of neighboring elements were fed into elementary motion detectors of the correlation type with a first-order temporal low-pass filter (time constant 35ms) in one of its branches. Each local movement detector consisted of two mirror-symmetrical subunits. In each subunit, the low-pass filtered signal of one input channel was multiplied with the high-pass filtered signal of the neighboring input channel. Elementary motion detector signals depend on the scenery (e.g. [42]). Therefore, a benchmark of optic-flow measurement independent of scenery was necessary. The optic flow can be computed when the self-motion of the agent and the nearness to objects is known. In a virtual environment, both pieces of information are accessible. The set of vertices X,Y,Z, that defines the object were translated to the current position of the agent. Then, the nearest point on the retina for each viewing direction was extracted. The geometrical optic flow was then computed from the nearness and the self-motion [78], giving a motion measurement independent of the texture of the scenery.

2.6.3 From EMD responses to nearness estimates

The optic flow experienced during translation is linked to the nearness of the agent to objects in the environment and its self-motion. Assuming that the agent moves in the null elevation plane and uses a spherical eye, it can be shown that the relative nearness ($v\mu$) is linked to a retinopically modified norm of the optic flow:

$$(v\mu(\epsilon,\phi))^2 = \sin^2(\epsilon)OF(\epsilon,\phi)^2_{\hat{\phi}} + OF(\epsilon,\phi)^2_{\hat{\epsilon}} \tag{2.1}$$

where v is the speed, μ is the nearness to the object in the viewing direction (ϵ,ϕ), ϵ the elevation, ϕ the azimuth, and OF the optic-flow vector. This function holds as long as the elevation is not zero, but a similar equation can be used for null elevation (section 2.7.1). The relative nearness cannot be computed at the FOE or the FOC due to the singularity, i.e. absence of apparent motion, in the flow field. To remove these singularities, the flow fields resulting in two different motion directions at the same location in the environment might be averaged. However, insects are unlikely to fly twice at the same position in the environment. However, they can fly subsequently in two different directions at two nearby points in space. The nearness at those points will be almost equal as long as the distance between these points is relatively small. Therefore, the agent performed a translation, composed of 50 segments with different motion directions. The motion direction of each segment followed a normal distribution centered at zero with a standard deviation of 18° (fig. 2.1.1). The translation is thought to correspond to an intersaccade of insect flight. A stack of optic-flow fields along the time, i.e. during the intersaccade, was gathered (fig. 2.1.2). Although it is possible to compute the nearness for each optic-flow field and then integrate the nearness over time, we used an alternative, but equivalent approach. The optic-flow fields were squared and integrated over time (fig. 2.1.3). Then, the integrated squared optic flow was used to compute the nearness map of the environment (fig. 2.1.4).

When the optic-flow field is estimated by EMDs, the estimations also depend on the motion history due to the temporal filters in the EMD. The saccade preceding the translation, therefore, interferes with the optic-flow measurements during the intersaccadic interval. This effect decreases over time. Therefore, the optic-flow field was not integrated during the entire intersaccadic phase, but only for the last five segments (i.e. last 5 ms).

2.6.4 Nearness and collision avoidance

Once the nearness map is known, it is an obvious strategy to avoid collisions by moving away from the maximum nearness value. However, the nearness map derived from the EMDs also depends on pattern properties. Thus, the nearness map was averaged along the elevation, giving the average nearness for a given azimuth and, thus, reducing the texture dependence (fig. 2.1.5). Each of these averaged nearness values could be represented by a vector in polar coordinates, where the norm of the vector is the averaged nearness and its argument the azimuth. The sum of these vectors points towards the average direction of close objects in the environment when the effect of the pattern on the EMD responses is sufficiently averaged out. Thus, the opposite of this vector will point away from the closest object and, thus, is selected as the motion direction of the agent in order to avoid a collision (fig. 2.1.6).

Moreover, the length of the vector increases with the nearness to objects and the apparent size of the object. Thus, its length can be used as a measure of the collision avoidance necessity. This measure drives the state of the animal between “collision avoidance” and “move in the previous direction” according to the following equation:

$$\begin{aligned} \gamma &= W\left(\|COMANV\|\right) \arg\left(COMANV\right) + (1-W)(\alpha+\sigma) \\ W\left(\|COMANV\|\right) &= \frac{1}{1+\left(\frac{\|COMANV\|}{n_0}\right)^{-g}} \end{aligned} \tag{2.2}$$

where, $COMANV$ is the vectorial sum of the vertically integrated nearness values, W is the weighting function based on the norm of the COMANV, α the goal direction and σ is a goal direction noise. The weighting function used in the simulation is a sigmoid function, which is driven by a gain g and a threshold n_0.

The goal direction has been fixed to zero for the simulation in boxes. The agent, thus, continues moving straight, i.e. a saccade with null amplitude, when the CAN is zero. In other simulations, the goal direction is different from zero. The agent then performs a saccade amplitude driven by the goal direction when the CAN is zero.

2.6.5 Cluster of trajectories and route similarities

Even when the starting position of different runs of the agent differ, the trajectories that are taken by the agent in a given cluttered environment tend to converge to similar routes. When approaching an obstacle, an agent may avoid it by a left or right turn, leading to either of two different routes. Therefore, the obstacles are the main factors affecting the overall structure of the trajectories and, thus, their similarity. In order to cluster trajectories into routes, a triangular meshing (Delaunay triangulation [54, p. 513-529]) of the environment was calculated with the nodes of the meshing corresponding to the center of mass of the obstacles. The meshing is, thus, composed of triangular cells formed by three neighboring obstacles. The cells in the meshing do not overlap. A trajectory of the agent crosses a succession of triangular cells and can, therefore, be associated to a sequence. Here, each element in the sequence represents a given cell in the meshing, i.e. a given region of the environment. The agent may visit a region more than once, by making detours or oscillating between two neighboring cells. The sequence was simplified in order to remove multiple visit by suppressing subsequences between identical sequence elements.

Once each trajectory has been associated to a sequence of cell occupancy, the trajectories sharing exactly the same sequence were attributed to a cluster, i.e. a route. Each route corresponds, therefore, to a unique sequence. Different routes may be similar. To quantify the similarity between routes, the number of single sequence elements, i.e. a cell, not shared by two routes was used. This measure of route similarity is similar to the Hamming distance [64] between route sequences.

2.7 Supporting Information

2.7.1 Optic flow and nearness

Advantage of a spherical eye

The optic-flow field is a two-dimensional vector field, where each vector is the apparent velocity of the images of objects on the eye of the agent. The optic-flow field experienced during translation results from the product of the relative nearness of objects in the environment and a factor depending on the angle between the direction of self-motion and the viewing direction of these objects ("viewing angle"):

$$\mathbf{OF} = -v\mu\left(\mathbf{u} - A\begin{pmatrix}\cos\epsilon\cos\phi\\ \cos\epsilon\sin\phi\\ \sin\epsilon\end{pmatrix}\right) \tag{2.3}$$

$$A = u_{\hat{x}}\cos\epsilon\cos\phi + u_{\hat{y}}\cos\epsilon\sin\phi + u_{\hat{z}}\sin\epsilon$$

where, $\epsilon \in [-\frac{\pi}{2}, +\frac{\pi}{2}]$, and $\phi \in [0, 2\pi]$ are the elevation and the azimuth of the agent's eye, respectively, μ is the nearness between the object and the agent in the direction (ϵ, ϕ), v is the speed of the agent and **u** is the direction of motion. The direction of motion is usually expressed as an earth coordinate system, i.e. Cartesian coordinates. However, an agent measures the optic flow in its eye referential system. We assume that the eye is spherical. The optic-flow field, therefore, needs to be expressed in spherical coordinates. A vector is conserved by a change of the referential system; however, its components change according to the elementary vectors' transformation. In the case of the transformation from Cartesian to spherical coordinates, the conservation of the vectors is expressed as:

$$\mathbf{OF} = OF_{\hat{x}}\hat{x} + OF_{\hat{y}}\hat{y} + OF_{\hat{z}}\hat{z} = OF_{\hat{\rho}}\hat{\rho} + OF_{\hat{\epsilon}}\hat{\epsilon} + OF_{\hat{\phi}}\hat{\phi} \tag{2.4}$$

where, **OF** is a vector, OF_x, OF_y, OF_z the component along $\hat{x}, \hat{y}, \hat{z}$, respectively, and $OF_\rho, OF_\epsilon, OF_\phi$ the component along $\hat{\rho}, \hat{\epsilon}, \hat{\phi}$, respectively.

The transformation from Cartesian coordinates to spherical coordinates is:

$$\begin{pmatrix} \hat{\rho} \\ \hat{\epsilon} \\ \hat{\phi} \end{pmatrix} = \begin{pmatrix} \cos\epsilon\cos\phi & -\sin\epsilon\cos\phi & -\sin\phi \\ \cos\epsilon\sin\phi & -\sin\epsilon\sin\phi & +\cos\phi \\ \sin\epsilon & +\cos\epsilon & 0 \end{pmatrix}^T \begin{pmatrix} \hat{x} \\ \hat{y} \\ \hat{z} \end{pmatrix} \tag{2.5}$$

Thus, the components in spherical coordinates are a function of the ones in the Cartesian coordinates:

$$\begin{cases} OF_{\hat{\rho}} & = +OF_{\hat{x}}\cos\epsilon\cos\phi \quad +OF_{\hat{y}}\cos\epsilon\sin\phi \quad +OF_{\hat{z}}\sin\epsilon \\ OF_{\hat{\epsilon}} & = -OF_{\hat{x}}\sin\epsilon\cos\phi \quad -OF_{\hat{y}}\sin\epsilon\sin\phi \quad +OF_{\hat{z}}\cos\epsilon \\ OF_{\hat{\phi}} & = -OF_{\hat{x}}\sin\phi \qquad\quad +OF_{\hat{y}}\cos\phi \end{cases} \tag{2.6}$$

Applying the transformation above to optic-flow field vectors, the component in the spherical coordinates are:

$$\begin{cases} OF_{\hat{\rho}} & = 0 \\ OF_{\hat{\epsilon}} & = \frac{v}{p}\left(u_{\hat{x}}\sin\epsilon\cos\phi + u_{\hat{y}}\sin\epsilon\sin\phi - u_{\hat{z}}\cos\epsilon\right) \\ OF_{\hat{\phi}} & = \frac{v}{p}(u_{\hat{x}}\sin\phi - u_{\hat{y}}\cos\phi) \end{cases} \tag{2.7}$$

The radial component is obviously null, because the optic flow is the apparent motion on the eye. The components along the elevation and azimuth contain information about the relative nearness, but this information is entangled with that on self-motion. We want to extract the relative nearness from this system of equations without knowing the direction of motion $(u_x, u_y, u_z)^T$. Let us assume, however, that the direction of motion is constrained to the equatorial plane, i.e. $u_z = 0$. By elevating the system of equations to the square and applying trigonometric identities, it can be shown for $\epsilon \neq 0$ that:

$$(v\mu)^2 = \frac{\sin(\epsilon)^2 OF_{\hat{\phi}}^2 + OF_{\hat{\epsilon}}^2}{\sin(\epsilon)^2} \tag{2.8}$$

We have used, until now, only the sum of both lines in the system of equations. By expressing $\mathbf{u}$ in polar coordinates, i.e. $\mathbf{u} = (\cos\theta, \sin\theta, 0)$, and using the ratio of $OF_\epsilon(u_z = 0)$ and $OF_\phi(u_z = 0)$, we get a factor telling how far away from the FOE and FOC we are looking, i.e. $\cos\left(2\left(\theta - \phi\right)\right)$

$$\begin{gathered}\frac{1-\cos\left(2(\theta-\phi)\right)}{OF_{\hat{\phi}}^2} = \sin(\epsilon)^2 \frac{1+\cos\left(2(\theta-\phi)\right)}{OF_{\hat{\epsilon}}^2} \\ \Rightarrow \begin{cases} \cos\left(2(\theta-\phi)\right) & = \dfrac{1-h(\epsilon)}{1+h(\epsilon)} \\ h(\epsilon) & = \sin(\epsilon)^2 \dfrac{OF_{\hat{\phi}}^2}{OF_{\hat{\epsilon}}^2} \end{cases}\end{gathered} \tag{2.9}$$

The term $\cos\left(2(\theta - \phi)\right)$ is independent of ϵ. It is, therefore, a constant for a given ϕ and θ. Thus, this factor can be computed for $\widetilde{\epsilon} \neq 0$ and used in the following equation to compute $v\mu$ for $\epsilon = 0$:

$$(v\mu)^2 = OF_{\hat{\phi}}^2 \frac{1 + h(\widetilde{\epsilon})}{h(\widetilde{\epsilon})} \tag{2.10}$$

This equation holds as long as the optic flow is not zero and the term $\cos\left(2(\theta - \phi\right)$ can be computed, i.e an $\epsilon \neq 0$ exists for a given ϕ.

Finally, the relative nearness can be computed from the optic-flow field experienced during a translation in the null elevation plane with the following equation, $\forall \epsilon, \widetilde{\epsilon} \neq 0$:

$$\boxed{(v\mu(\phi,\epsilon))^2 = OF(\phi,\epsilon)^2_{\hat{\phi}} \left(1 + \frac{OF(\phi,\widetilde{\epsilon})^2_{\hat{\epsilon}}}{OF(\phi,\widetilde{\epsilon})^2_{\hat{\phi}}} \frac{1}{\sin(\widetilde{\epsilon})^2}\right)} \tag{2.11}$$

Error on relative nearness

The motion of the agent has been assumed to be contained in the null elevation plane of the agent. The agent might not exactly move in this plane and, therefore, the upward component of the motion u_z will be different from zero. If the agent estimates the relative nearness with our set of equations, derived with $u_z = 0$, it will make an error proportional to u_z, but this error also depends on the viewing direction (ϵ, ϕ) and the forward and sideward component of the motion (u_x, u_y)

$$\begin{aligned}\frac{\sin(\epsilon)^2 OF^2_{\hat{\phi}} + OF^2_{\hat{\epsilon}}}{\sin(\epsilon)^2} &= (v\mu)^2(1 - 2u_{\hat{z}}\text{cotan}\epsilon(u_{\hat{x}}\cos\phi + u_{\hat{y}}\sin\phi) \\ &\quad + u^2_{\hat{z}}(\text{cotan}^2\epsilon - 1))\end{aligned} \tag{2.12}$$

Note: $u_z \in [-1, 1]$, because $\|\mathbf{u}\| = 1$, therefore, $u_z^2 < u_z$.

Limitation of a cylindrical eye

The relative nearness can be extracted by equation (#Eq: 12#), as long as the agent translates in the null elevation plane and has a spherical eye. However, for an agent with a cylindrical eye, the components of the optic flow in cylindrical coordinates are:

$$\begin{cases} OF_{\hat{\rho}} & = 0 \\ OF_{\hat{z}} & = -\frac{v}{p} u_{\hat{z}} \\ OF_{\hat{\phi}} & = +\frac{v}{p}(u_{\hat{x}} \sin\phi - u_{\hat{y}} \cos\phi) \end{cases} \tag{2.13}$$

where, ρ,z,ϕ are the cylindrical components, $\vec{u} = (u_x, u_y, u_z)^T$ and v the direction of motion and the speed of the agent, respectively, and p the distance to the closest object in the viewing direction defined by ρ,z,ϕ. The motion of the agent with a spherical eye has been constrained in the null elevation plane. This constraint for a cylindrical eye is a motion in a plane parallel to the base of the cylinder, i.e. $u_z = 0$. However, applying this constraint to the previous equation leads to $OF_z = 0$. The system has one equation and three unknown variables, therefore, it cannot be solved. Constraining the motion in a plane parallel to the base of the cylinder is, thus, not a suitable strategy to extract nearness from optic flow measured on a cylindrical eye.

2.7.2 Performance of the algorithm in cluttered environments

The performance of our collision avoidance algorithm combined with a goal direction was assessed and quantified in term of its reliability, i.e. the percentage of trajectories with a collision occurring before the goal is reached, and its efficiency, i.e. the distance traveled needed to reach the goal. The algorithm was tested in two cluttered environments, either with or without ceiling and outer walls, and either covered with a 1 mm or 4 mm random checkerboard pattern. For each cluttered environment, 201 starting positions were tested. For the overall 1,608 simulated trajectories, only three crashes were observed. Thus, our collision avoidance algorithm is reliable under the conditions tested.

The distance traveled is a poor measurement to quantify the efficiency of the algorithm, because it depends on the layout and dimension of the environment, the distance separating the starting and goal location, as well as the number of obstacles between those locations. Therefore, the efficiency of the algorithm was quantified as the ratio between the theoretically shortest trajectory between the starting and goal location and the distance traveled. The problem of finding the theoretically shortest trajectory is linked to graph theory, and is solved in this context by Dijkstra's algorithm [37]. Therefore, the environment was transferred to a graph-like representation, here the goal and starting location, and the corners of the obstacles were the nodes of the graph. Two nodes in the graph were connected if the line between those nodes was not crossing an object. The shortest path was computed using Dijkstra's algorithm [37]. This method is similar to allowing an agent to get as close as possible to an object. Our collision avoidance algorithm, however, prevented such behavior. Therefore, a second reference had to be calculated. A Voronoi diagram [54, p. 513-529] is composed of segments maximizing the distance between two neighboring nodes. When the nodes represent the center of mass of the objects, the segments represent the safest trajectory between two neighboring objects. The environment can, therefore, be represented by a graph with the segments and the intersection between segments representing the edges and the nodes of the graph, respectively. However this graph needs to be connected to the starting and goal locations. These starting and goal locations were connected to the closest node in the graph, outside the obstacles. The shortest trajectory in the graph, which differs from the theoretically shortest trajectory in the environment, is called the "Voronoi trajectory." This trajectory has been computed using Dijkstra's algorithm [37]. Figure S2 shows the efficiency of our collision avoidance algorithm. Our algorithm is more efficient than the "Voronoi trajectories" in most of the conditions tested.

2.7.3 Supporting figures

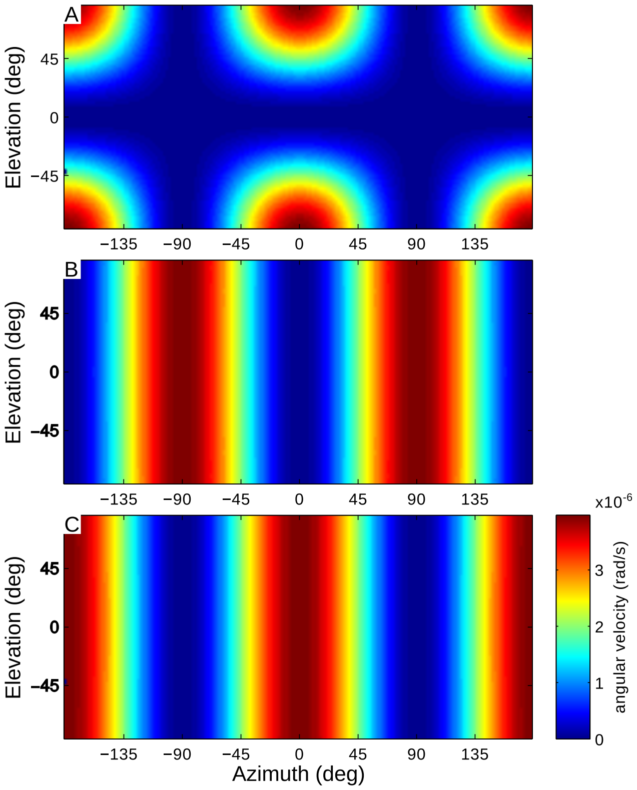


Figure 2.13: Variation of horizontal and vertical color-coded optic-flow component squared for a motion in the null elevation plane in the center of a sphere. A) Vertical optic-flow component squared. B) Horizontal optic-flow component squared. C) Vertical optic-flow component squared divided by the sine squared of the elevation of the viewing direction, i.e. the retinotopically corrected vertical optic-flow component. The null azimuth corresponds to the motion direction of the agent. Note that the horizontal component and the retinotopically corrected vertical optic-flow component have an antagonistic variation (compare B and C). The color code marks larger optic-flow values with warmer colors, and represents angular velocity in rad.s^{-1}.

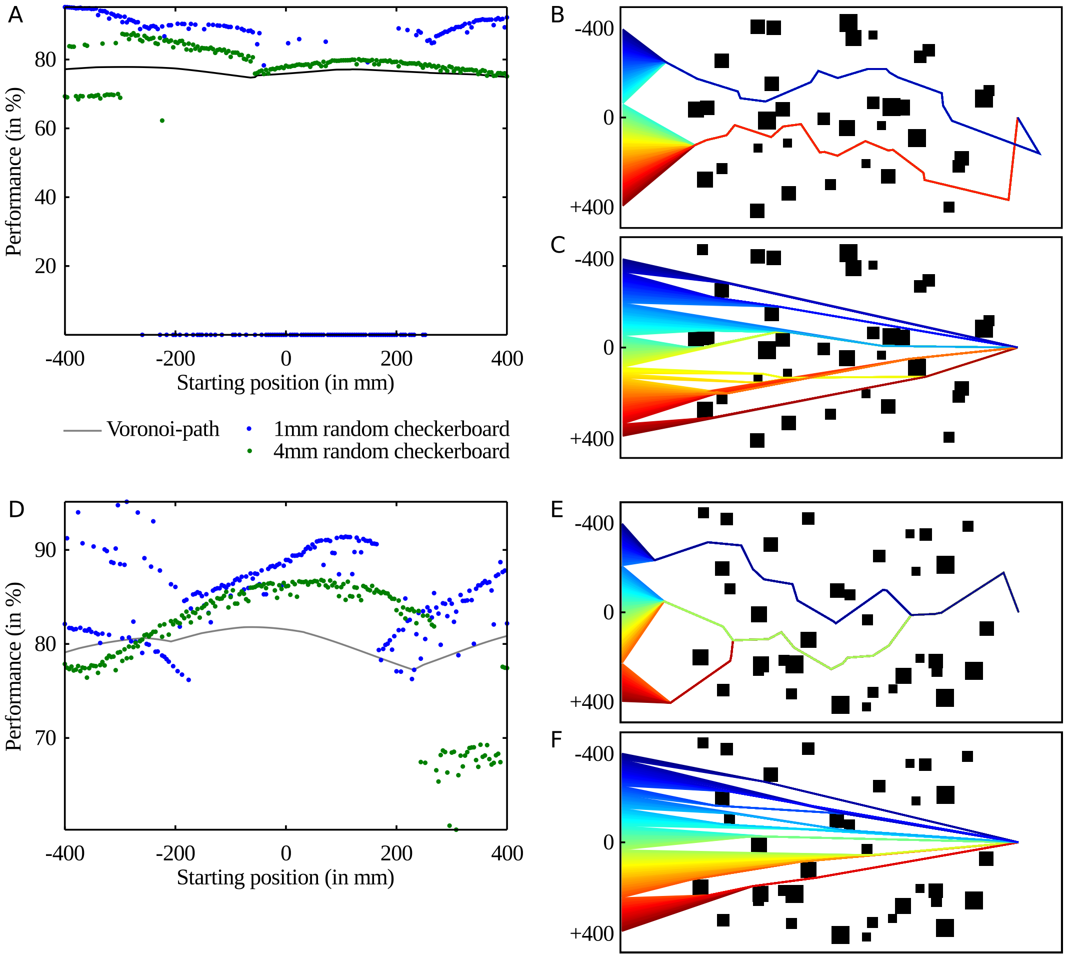


Figure 2.14: Performances of our algorithm based on the distances of travel to reach the goal in a cluttered environment relative to the distance of the shortest trajectories. The performance is the length of the shortest trajectory divided by the length of the trajectory followed by the agent. A, D) Relative distances of travel to reach the goal as a function of the starting position. Trajectories for the environments covered by a 1 mm and 4 mm random checkerboard pattern, respectively, are shown in Figure 10. B, E) Voronoi trajectories in the environments. C, F) Shortest trajectories in the environments. A, B and C (resp. D, E and F) are for the first (resp. second) environment. Note that the performance is above 60%, except for the trajectory with a dead-end.

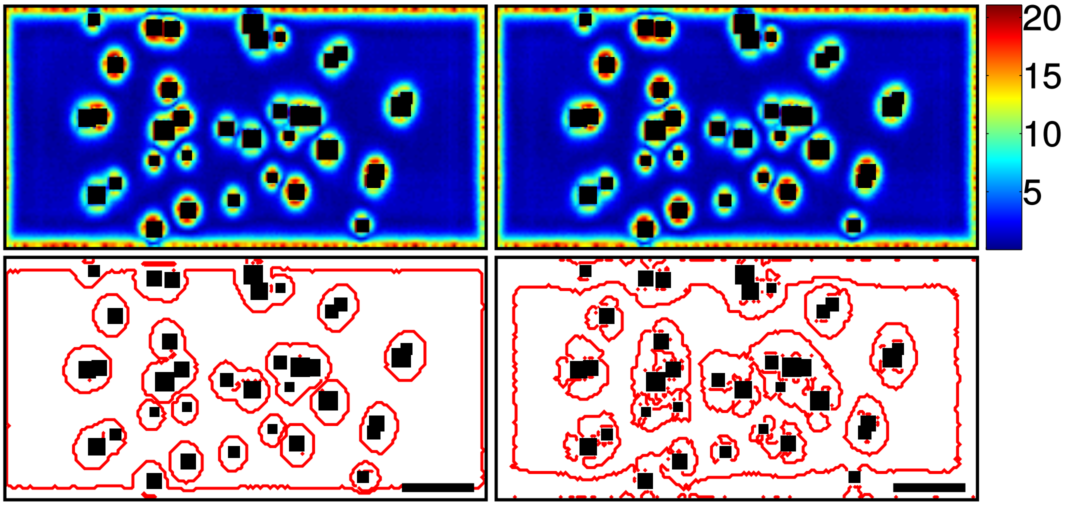


Figure 2.15: Norm of the COMANV in cluttered environment covered with 1 mm (left panels) and 4 mm (right panels) random checkerboard patterns. Top panels: Color-coded norm of the COMANV. Bottom panels: The red line delimits the zone where the weighting function is equal to 0:5. Here, the gain and the threshold of the weighting function are 2 and 4, respectively. Saccade amplitudes are mainly driven by the CAD when the agent is inside a contour (threshold line) containing an object. The contour represents the switch in behavior. Objects are indicated by black squares.

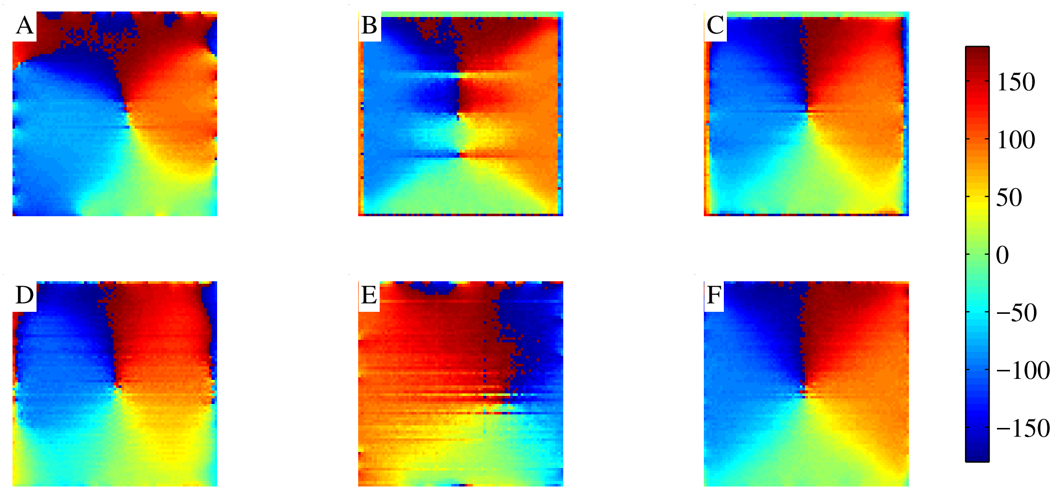


Figure 2.16: CAD in a cubic box covered with different patterns. The agent moved toward the north wall. The color represents the angle (in degrees) between the CAD and the direction of motion of the agent. Walls of the box are covered with a natural grass pattern (A), a 1 mm random checkerboard (B), a 4 mm random checkerboard (C), an 8 mm random checkerboard (D), a 35 mm random checkerboard (E), and a random pattern with 1 = f statistic (F). Note that the CAD for the 35 mm random checkerboard points in many places toward the wall of the box (E).

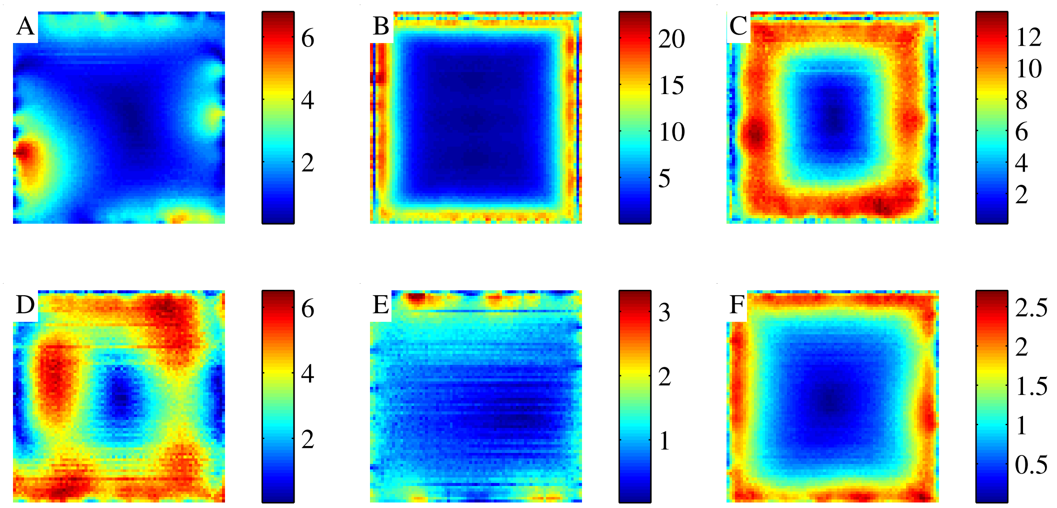


Figure 2.17: Norm of the COMANV in a cubic box covered with different patterns. The agent moved towards the north wall. The color represents the norm of the COMANV, i.e. the collision avoidance necessity. Walls of the box are covered with a natural grass pattern (A), a 1 mm random checkerboard (B), a 4 mm random checkerboard (C), an 8 mm random checkerboard (D), a 35 mm random checkerboard (E), and a random pattern with $1/f$ statistic (F). Note that the CAN is higher close to the wall than at the center of the box. The CAN remains low for the box covered with natural grass pattern in some places close to the wall, due to large leaves in the pattern.

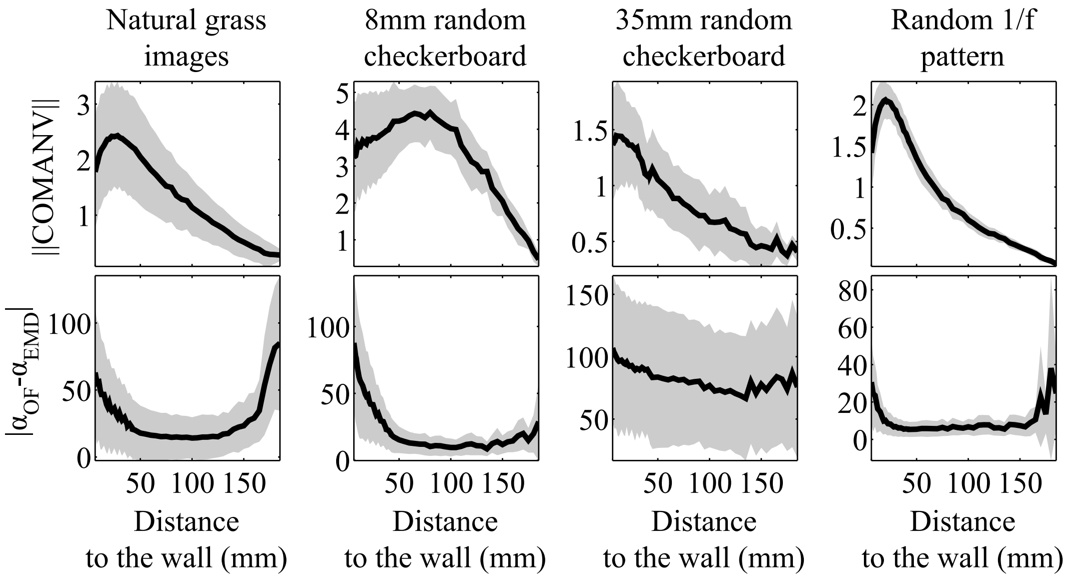
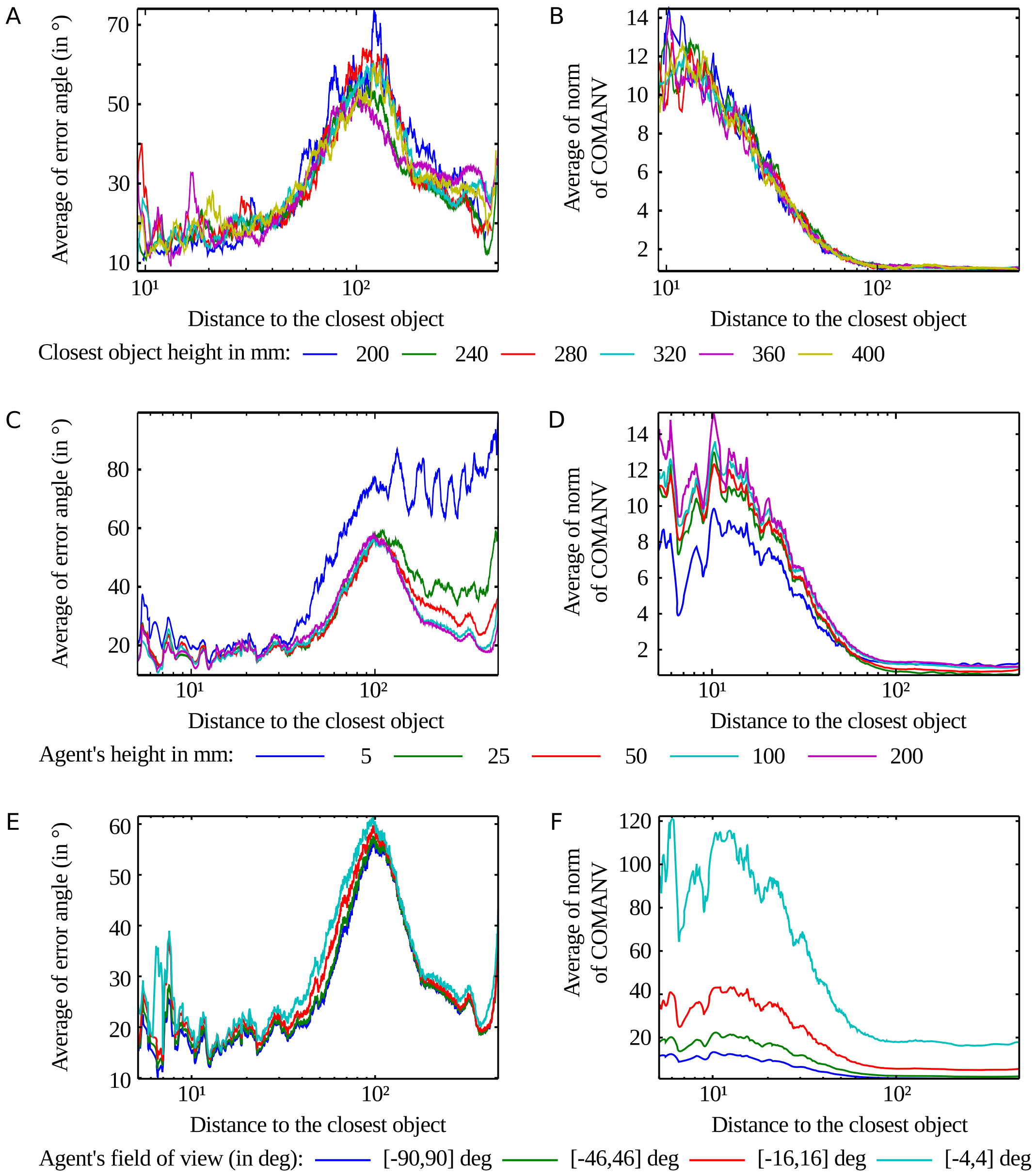


Figure 2.18: COMANV versus the distance to the closest wall in a cubic box covered with different patterns Top row shows the norm of the COMANV computed with EMD responses. Bottom row shows the angle between the COMANV computed with EMD responses and the control based on geometrical optic flow. Here α is the CAD. Thick lines and shaded area represent the mean and the standard deviation, respectively, computed at a given distance from the wall. Note that the norm of the COMANV is an increasing function of the distance as long as the agent is not too close to the wall.



(Legend on next page)

Figure 2.19: COMANV versus the distance to the closest object in cluttered environments. Left panels: Average of the error angle between the COMANV and the vector between the agent and the closest object. Right panels: Norm of the COMANV. The averages are computed with a sliding window with a window size corresponding to 10% of the distance to the closest object. A, B) Effect of the height of the closest object for an agent with a field of view (FOV) along the elevation of ±90°, and moving at an altitude of 100mm above the ground. Note that the height of the object does not strongly influence the COMANV. C, D) Effect of the moving altitude of an agent with a FOV along the elevation of ±90°. Note that the error angle far from the object is higher for motion close to the ground than far from the ground. E, F) Effect of the FOV of the agent. The agent moved at an altitude of 100mm above the ground. Note that the norm of the COMANV is strongly affected by the FOV along the elevation. The FOV along the elevation was ±180° for all cases.

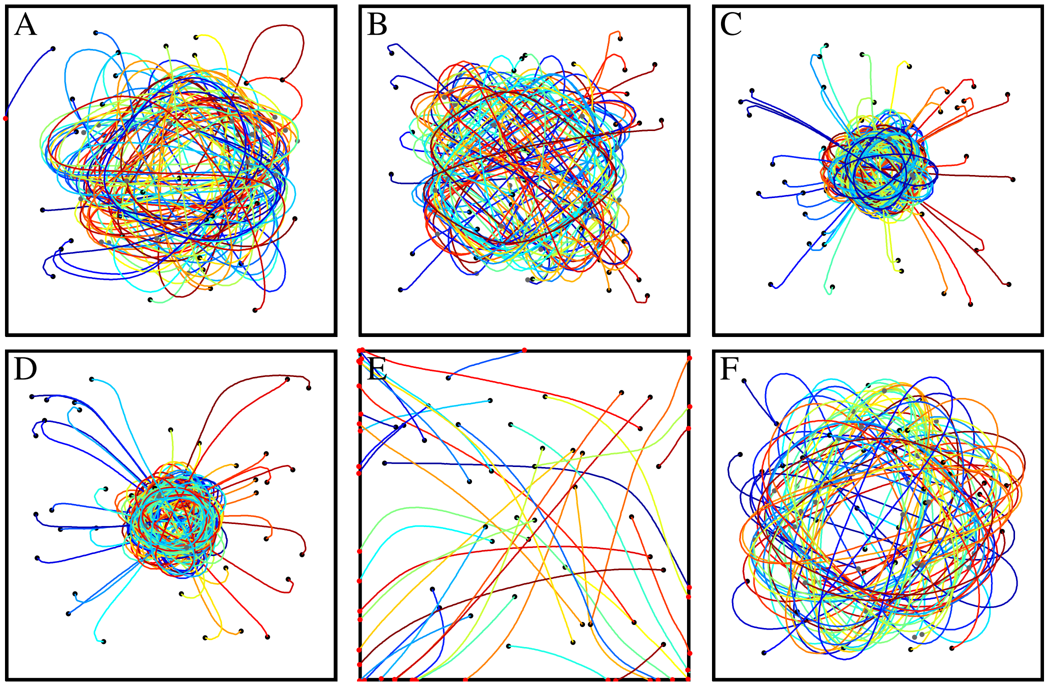


Figure 2.20: Trajectories of the agent with a collision avoidance system based on EMDs in a box $(40 \times 40 \times 40 \times \mathrm{cm})$ covered with different patterns (seen from above). Trajectories with 40 different starting positions are shown. The simulation time was 10sec or until the agent crashed. Walls of the box covered with a natural pattern (A), a 1mm random checkerboard (B), a 4mm random checkerboard (C), an 8mm random checkerboard (D), a 35mm random checkerboard (E), and a random pattern with $1/f$ statistic (F). The simulations, parameters and environments are identical to those used for fig. 2.7.

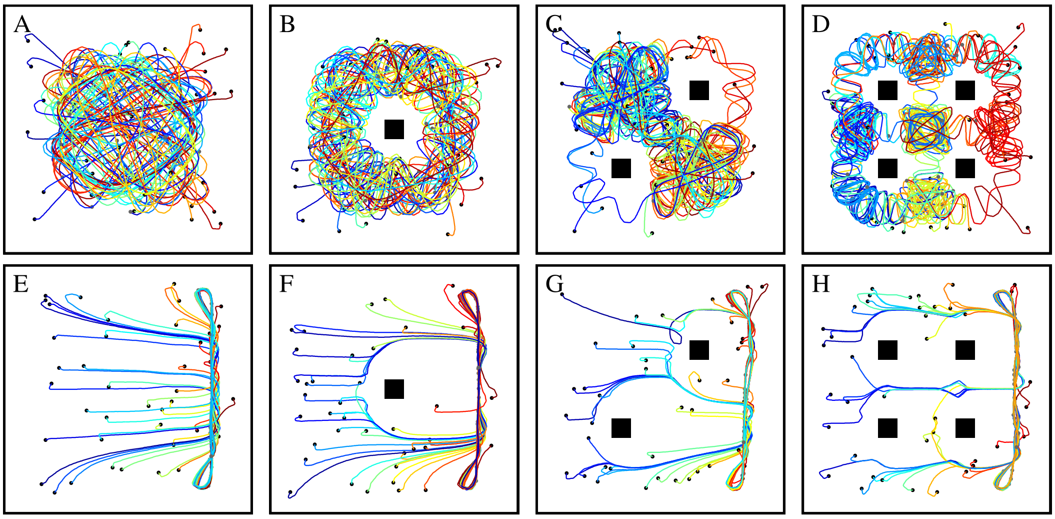


Figure 2.21: Trajectories of the agent with a collision avoidance system based on EMDs in a box $(40 \times 40 \times 40 \times \mathrm{cm})$ containing up to four objects and covered with $1\mathrm{mm}$ random checkerboard (seen from above). Top panels: Goal direction is moving forward. Bottom row: Goal direction is the right wall. A, E) No object in the box. B, F) One object in the center of the box. C, G) Two objects on one diagonal. D, H) Four objects on the diagonals. The objects were vertical bars with a quadratic base with a side length of $3\mathrm{cm}$ and a height of $40\mathrm{cm}$. Apart from taking the goal direction into account for the bottom panels, the simulations, parameters and environments are identical to those used for the top panels of fig. 2.8.

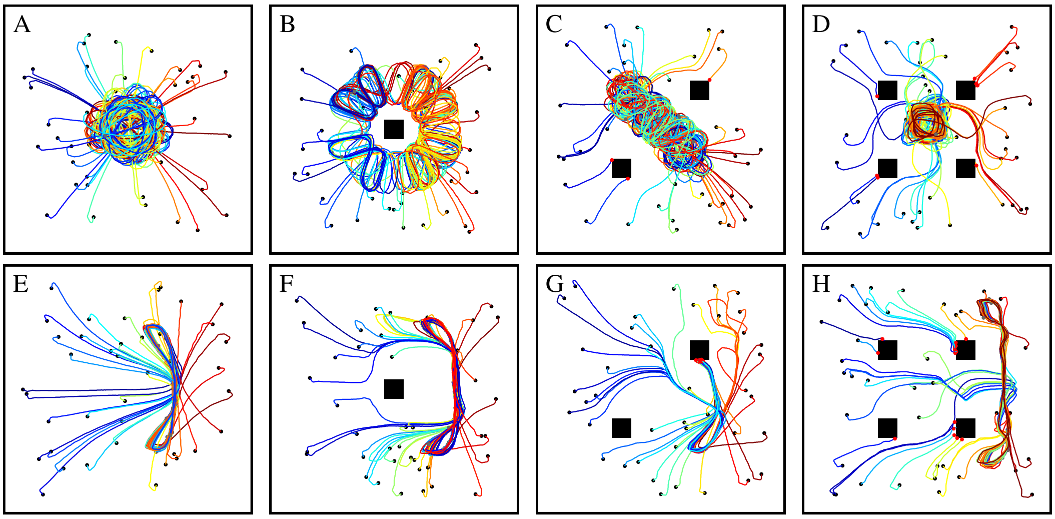


Figure 2.22: Trajectories of the agent with a collision avoidance system based on EMDs in a box $(40 \times 40 \times 40 \times \text{cm})$ containing up to four objects and covered with 4mm random checkerboard (seen from above). Top panels: Goal direction is moving forward. Bottom row: Goal direction is the right wall. A, E) No object in the box. B, F) One object in the center of the box. C, G) Two objects on one diagonal. D, H) Four objects on the diagonals. The objects were vertical bars with a quadratic base with a side length of 3cm and a height of 40cm. Apart from taking the goal direction into account for the bottom panels, the simulations, parameters and environments are identical to those used for the bottom panels of fig. 2.8.

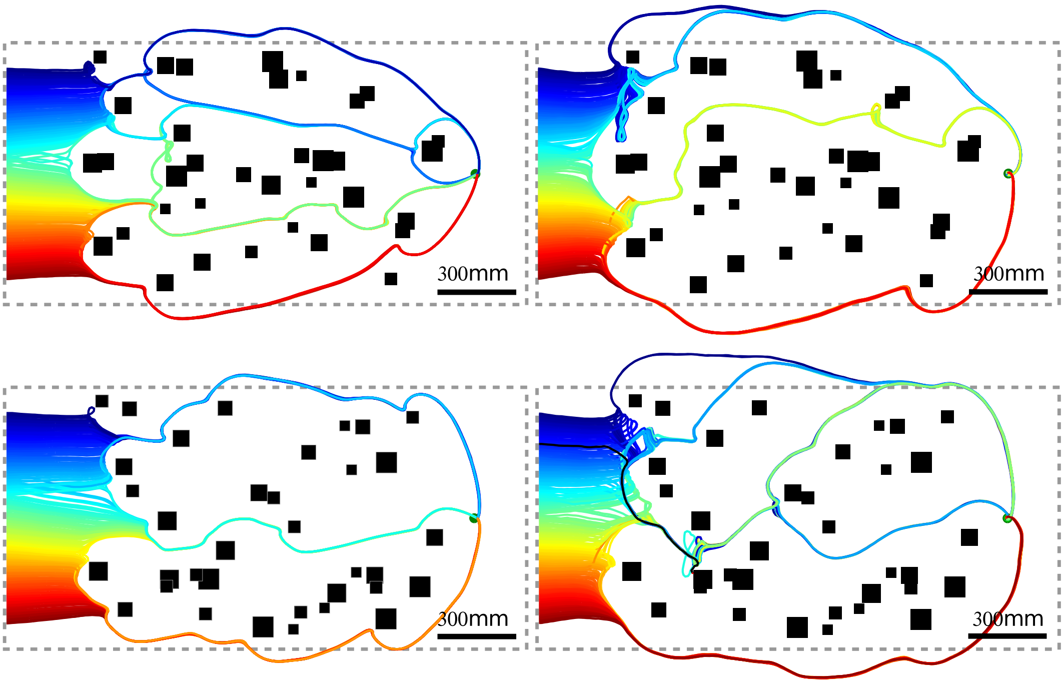


Figure 2.23: Trajectories of the agent equipped with an EMD-based collision avoidance system, but also relying on the goal direction in two different cluttered environments without walls. Objects were covered by a 1mm (left column) or 4mm (right column) random checkerboard pattern. The goal was pinpointed in the environments (Green dot). Two hundred one starting positions were tested, and simulations were run either for 100s (gray lines, i.e. dead-end), until the goal was reached (colored lines) or until a crash (black lines). Wall of the corridors of fig. 2.11 are represented by thick dotted gray lines and objects by filled black squares. The gain was 2 and the threshold 4 for all cases.

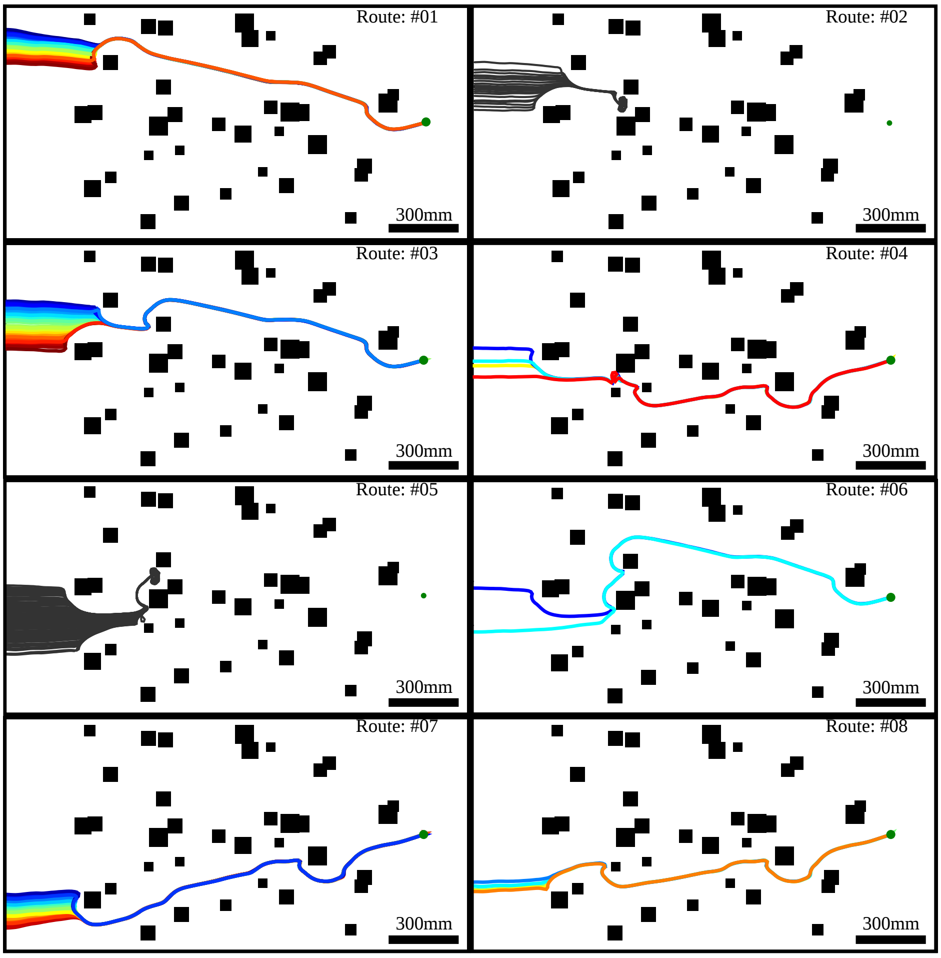


Figure 2.24: The routes, extracted from the trajectories shown in fig. 2.11, in the first cluttered environment which is covered by 1 mm random checkerboard. Several routes are found for neighboring starting locations, e.g. routes #4, #5 and #6. Certain routes share common subroutea, e.g. the end of route #7 and #8 are similar. The route similarities are shown in fig. 2.12 . Walls of the corridors are represented by thick black lines, and objects by filled black squares. Same colors as fig. 2.11.

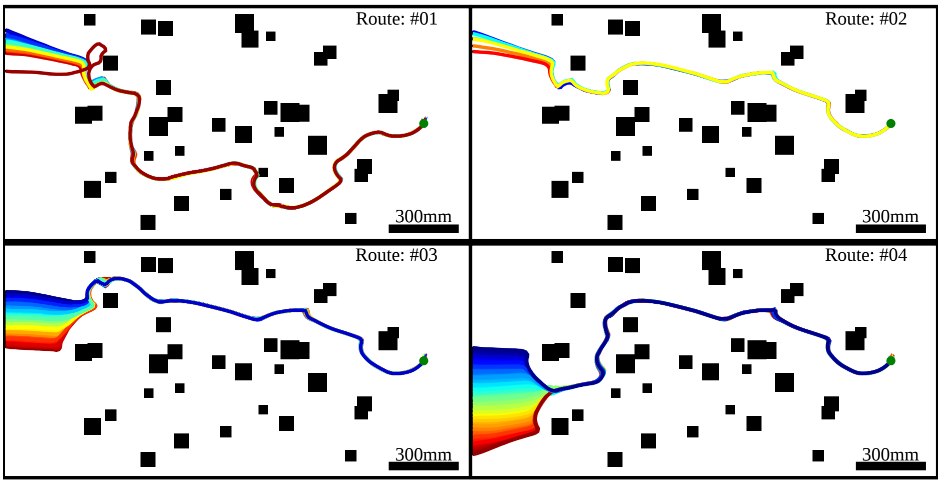
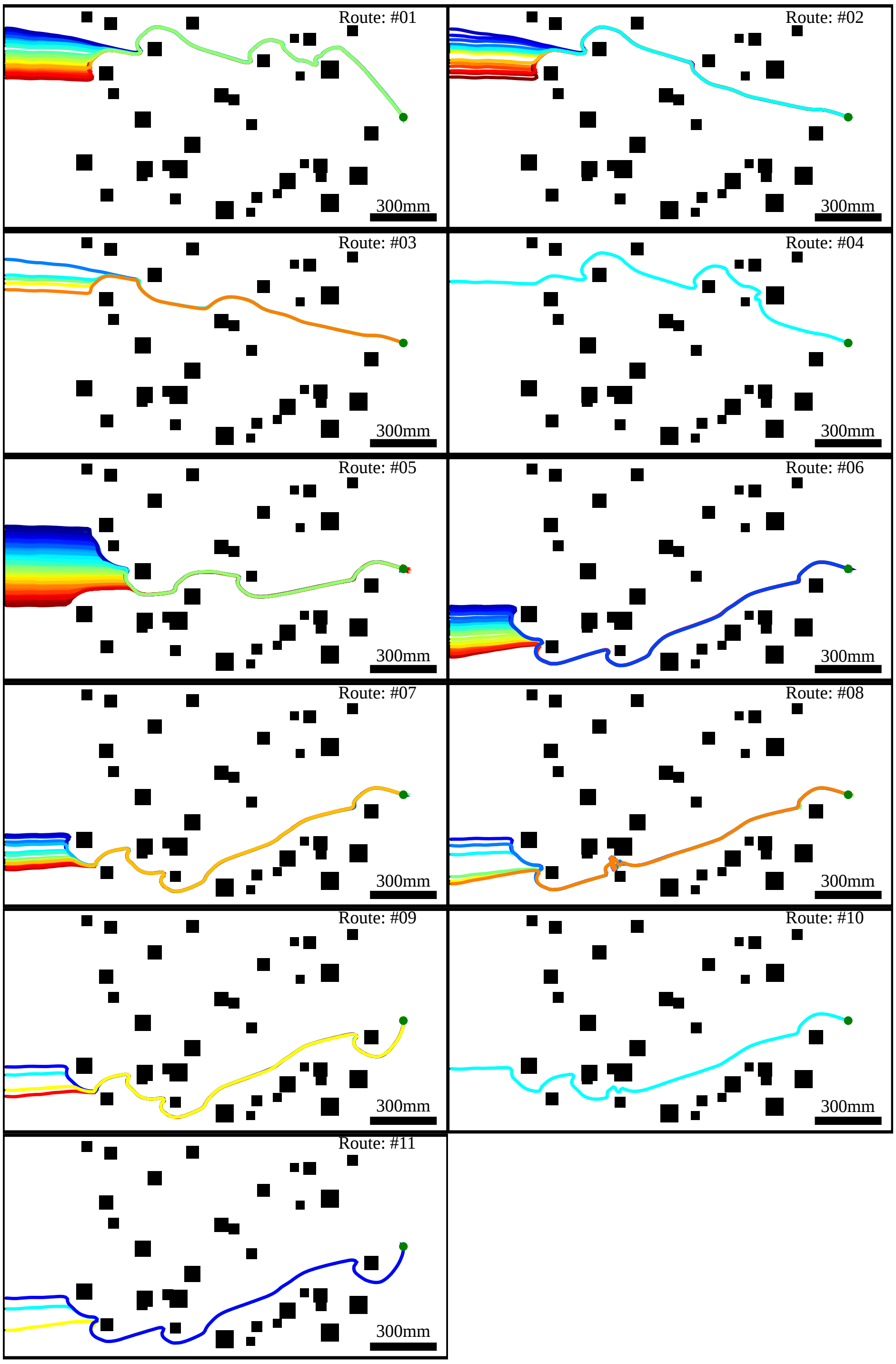


Figure 2.25: The routes, extracted from the trajectories shown in fig. 2.11, in the first cluttered environment which is covered by 4mm random checkerboard pattern. Routes #1 and #2 overlap in their starting locations. However, route #2 is shorter than route #1. Note that routes #2, #3 and #4 are identical to routes #3, #1 and #6, respectively, in the environment covered by a 1mm random checkerboard pattern fig. 2.12. The route similarities are shown in fig. 2.12. Walls of the corridors are represented by thick black lines, and objects by filled black squares. Same colors as fig. 2.12.



(Legend on next page)

Figure 2.26: The routes, extracted from the trajectories shown in fig. 2.11, in the second cluttered environment which is covered by 1mm random checkerboard pattern. Note that routes #5-11 and routes #1-4 form two distinct classes (see fig. 2.12). The route similarities are shown in fig. 2.12. Walls of the corridors are represented by thick black lines, and objects by filled black squares. Same colors as fig. 2.11.

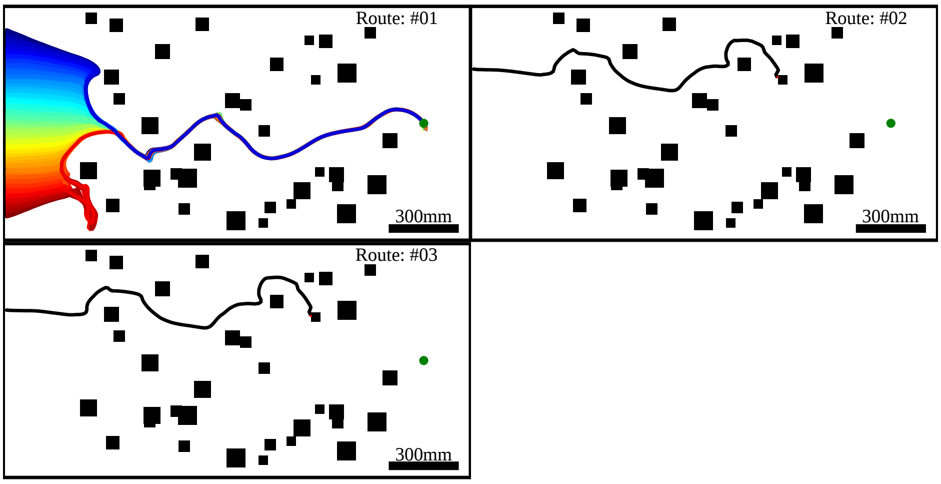


Figure 2.27: The routes, extracted from the trajectories shown in fig. 2.11, in the second cluttered environment which is covered by 4mm random checkerboard pattern. Routes #2 and #3 are very similar (see fig. 2.12) and lead to a crash of the agent. The remaining route (route #1) is identical to route #5 in the environment covered by 1mm random checkerboard (fig. 2.26). Walls of the corridors are represented by thick black lines, and objects by filled black squares. Same colors as fig. 2.11.

2.7.4 S1 Video

Simulation of agents without goal direction in the second cluttered environment. The trajectories are shown in figure 2.9, bottom. Objects and walls are covered by a 1mm random checkerboard pattern. Objects are represented by black squares, agent's current positions are represented by colored circles and agent's past positions are represented by a colored line.

2.7.5 S2 Video

Simulation of agents with goal direction in the second cluttered environment. The trajectories are shown in figure 2.10, bottom right. Apart from taking the goal direction into account, the simulations, parameters and environments are identical to those used for section 2.7.4.

2.7.6 S3 Video

The agent's view while moving through a cluttered environment. The trajectory is shown in figure 2.10, top-left. Objects and walls are covered with a 4mm random checkerboard pattern. Top: Agent's view. Overlay: The relative points of nearness along the azimuth extracted from EMD responses during the last intersaccade. The goal direction and CAD are indicated by the green and red line, respectively. The length of the red and green line indicate the necessity to follow the CAD or goal direction based on the weighting function. The dotted line is the saccade direction. Bottom: The agent's position and orientation in the environment. Objects are represented by black squares, agent's current position and orientation in blue and agent's past position in gray. The behavioral state of the agent, saccade or intersaccade is indicated by a white circle.

Acknowledgments

We thank Moritz Milde for helpful preliminary simulations.

3 How should I find my route again?

This chapter is planned to be published as Olivier J.N. Bertrand, Jens P. Lindemann, and Martin Egelhaaf (2016). Deterministic, systematic or random search: How should I find my route again?

3.1 Abstract

A fundamental navigation strategy is the ability to search efficiently for relevant places, the location of which being not exactly known. Although most of the research has been focused on optimal searching strategies to find locations dispersed sporadically in the environment, such as unvisited flower patches and prey, little is known about how an agent can find a route learned previously connecting relevant places. Indeed, most navigating animals (including humans) optimize their navigation by learning and following routes, but displacement from the route may occur frequently, for instance, when the animal needs to avoid an impending danger (such as a predator), is distracted and, thus, commits navigational errors, or,

in the case it is a light-weight animal, is displaced by wind or other currents. In this chapter, we assess systematically and thoroughly by simulations of a variety of bio-inspired search strategies the probabilities of an animal finding its route back within a given time after been displaced by a given distance away from the route. We found that a unique best search strategy does not exist. Rather, the best strategy depends on the displacement of the agent from its route and the distance travelled that the agent can invest searching.

3.2 Introduction

The daily routine of most humans includes the navigation between their home and their working place. Those humans, similar to many other animals, commute efficiently between two locations (for humans, their home and their working place; for honeybees, their hive and a patch of flowers) by following habitual routes. However, the journey of an animal or a technical system, i.e. an agent, may be disrupted by distractions causing navigational errors, such as the urge to avoid a danger or the encounter of external factors (e.g. the wind or some other current in the case of light-weight agents). The agent, after the disruption of its journey, will find itself displaced away from its known route to a potentially unknown location. The agent has to search to find its route again. Here we analyse the performance of different search strategies that allow an agent to find its known route again efficiently.

Although the navigational problem addressed here applies quite generally to autonomous agents and, thus, to navigating mammals, invertebrates and robots, we will illustrate the problem from the perspective of foraging brood-caring insects. The choice of foraging insects, such as bees, ants and many wasps, for illustration is not arbitrary, since the survival of their larvae depends on the navigational and search skills of the foragers (e.g. young bumblebee queens need to visit up to 6000 flowers per day to collect nectar and pollen and feed their larvae [136]). These insects have amazing skills despite their tiny brains, as they can travel several kilometres from their nest [57], move through cluttered environments and are able to find the nest again [79, 176, 31, 85, 7]. Foraging insects are, thus, a perfect example of great navigators with the additional advantage of consuming only a little energy and using computationally simple algorithms.

Foragers, such as ants and bees, need to gather resources (e.g. flowers) that are often scattered in the environment. To find good resources for the first time, the foragers need to search for them. A search incurs costs, e.g. energy and time. Thus, a forager is thought to organise its search to maximize its encounter rate with the targeted resource [11]. In order to search for the resources efficiently, foragers may employ a wide variety of search strategies [120, 129, 87, 130]. Once a rewarding resource has been found, a route between its location and the nest of the forager will be learned [79, 94].

Once a route between the home and the location of a good resource has been established, it is advantageous for the forager to use it to travel between both places. However, these routes may have been optimized according to different principles. They may minimize the time travelled, the distance travelled or the encounter with potential dangers, etc. (or for humans in a city, minimizing the time spend in traffic jams). The foragers may, for example, want to visit a rewarding food source frequently during the day and, therefore, may try to minimize the time travelled to fully exploit the resource before it is used by competing foragers. In open environments, fire ants, for instance, have been shown to follow the quickest route [110]. In cluttered environments, idiosyncratic routes have been observed where ants were travelling back and forth between a food location and the nest [79, 94]. As a cluttered environment contains many objects, it may also be relevant to avoid the danger of colliding with obstacles. Our study, therefore, takes three different types of routes between the food location and the nest location into account. Two of them, the shortest (mimicking the fire ants' strategy) and the safest (mimicking the motion of bees through cluttered environments) are based on the distribution of objects in the environment, i.e. the topology. The third route type is not based on the distribution of objects in the environment and, thus, is a non-topological route. The use of a non-topological route has two purposes: Firstly, ants living in a cluttered environment do not only follow the safest or the shortest route, but also idiosyncratic routes. Therefore, a third type of route is necessary. Secondly, our simulations are based mostly on the topology of the environment (due to a discretization of the environment, see section 3.5). Therefore, our results could be an artefact of the topology of the environment, and only apply to the shortest and safest route. The non-topological route is a kind of generalization of our findings.

However, an agent cannot take it for granted that it is able to follow its route established previously under all conditions. Rather, it may be displaced from it by errors of the route following mechanism, by escape manoeuvres if a danger unexpectedly appears (e.g. a predator) or by external forces (e.g. a gust of wind in the case of a light-weight animal). From an evolutionary point of view, the animal should find its route again as quickly as possible. The more time spent searching, the less food will be brought back home and the more probable an encounter with a predator will be. The animal should, thus, be guided by a search strategy which maximizes the probability of finding its route within a given time, that time being as short as possible. The limited locomotion speed of the animal then bounds the probability of finding the route again. Indeed, an animal displaced 10m away from its route and limited to travelling at 1m.s^{-1} cannot find its route again in less than 10s. The search strategy to employ, therefore, should depend on the distance between the current location of the animal and its route. In the previous example, the animal would find its route in exactly 10s only if it follows the shortest path between its current location and its route. In this case, the animal has no time to deviate from this path, i.e. it has to move along it deterministically.

To be sure that the animal finds its route, it has to be guided by a search strategy that will explore the environment over time. The most efficient strategy to explore the environment is a systematic search, such as a spiral search, because the agent guided by this strategy will visit every location only once. However, a systematic search is extremely difficult to perform in a cluttered environment. Due to the presence of objects, the agent cannot move strictly along a spiral, but has to make detours around the objects and, at the same time, not visit a location twice. It has, therefore, to remember which location has been visited previously. Its memory being limited, the agent can, thus, not follow a systematic search indefinitely. To visualize a systematic search, one can imagine an agent walking in a city with pavements. At an intersection, pavements are numbered, for example, from the left to the right of the agent. The agent, at an intersection of pavements, will take the first nth pavement which does not lead to an intersection visited already.

Because the systematic search strategy is difficult for an agent to perform, other strategies without extensive memorization of the objects visited previously need to be used by an agent. The agent may, for example, move randomly in the environment. To visualize a random search, one can imagine an agent walking randomly around a city. At every intersection of pavements, the agent chooses one of the pavements randomly. Although an agent moving randomly will visit the same locations in the environment multiple times, it will still, over time, explore the environment, and, therefore, be sure to find its route.

Random and systematic search are not the only strategies an animal can employ. The agent may move according to a deterministic local procedure. To visualize a deterministic search, we again used the agent walking in a city. At an intersection, the agent chooses the pavement based on certain characteristics at the intersection. The agent may, for example, choose the brightest pavement or the widest pavement. Inspired by the travelling of flying insects in a cluttered environment, our agent will, when guided by a deterministic search, avoid venturing along narrow paths (animals reduce their speed when travelling along narrow paths, e.g. [76]). However, because the deterministic strategy is not systematic or random, it may lead to visiting only a small portion of the environment and, therefore, never finding its route again. To compensate for this limitation, the animal may from time to time reorient itself randomly. The combination of deterministic and random search, i.e. a combined search, may be a powerful strategy to explore environments [74].

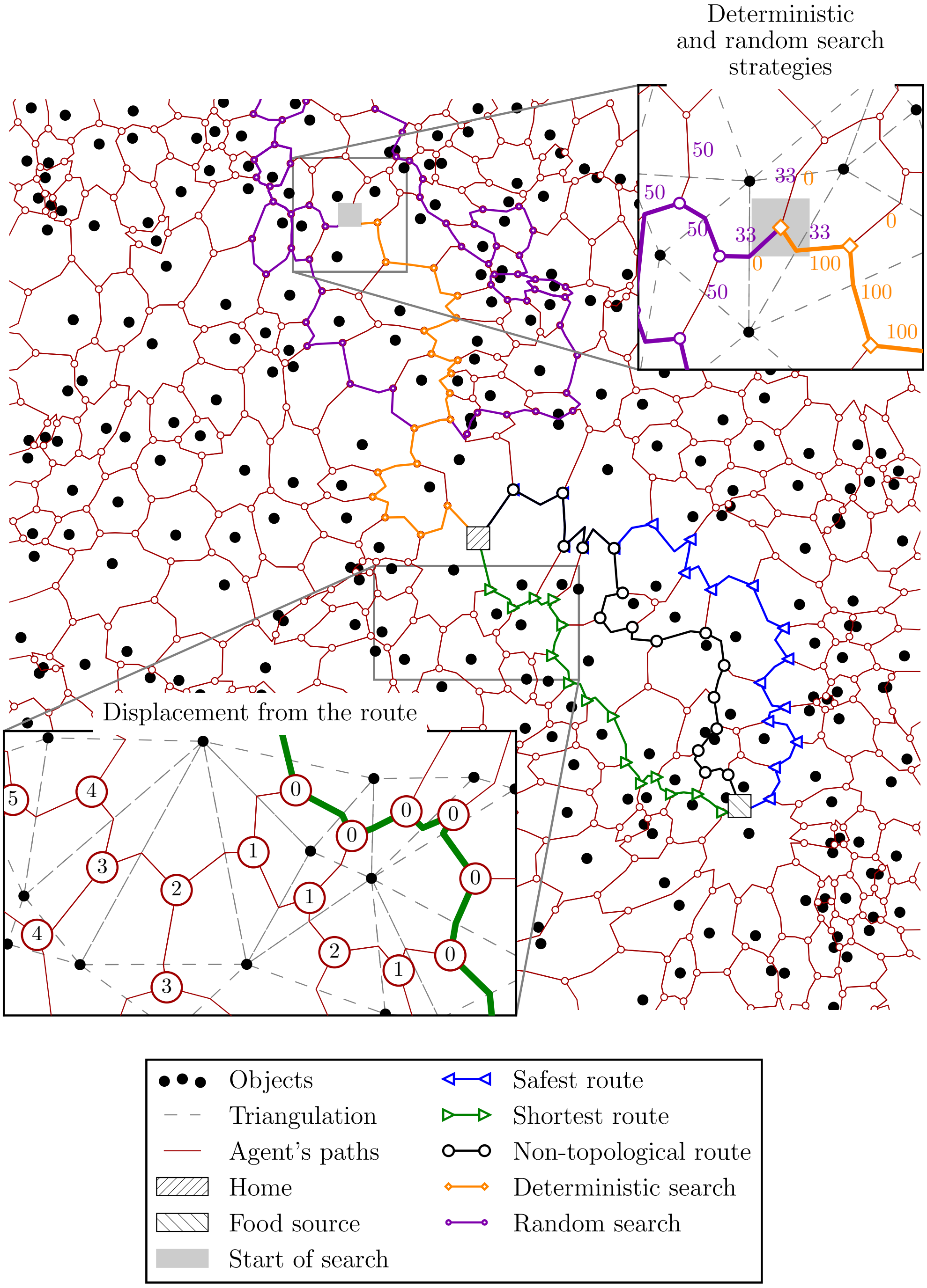
The variety of possible routes in cluttered environments is huge. In the present chapter, we focus on the "shortest" route, i.e. the route minimizing the distance travelled, the "safest" route, i.e. the route minimizing the venture along narrow paths (e.g. minimizing the risk of collisions with the numerous obstacles) and the "non-topological" route, i.e. not based on the distribution of objects (see section 3.5.2 for details). Inspired by the centring behaviour of animals flying along tunnels or in cluttered environments [142, 35, 76, 8, 88, 12, 123], the motion of the agent in our simulations has been restricted to paths centred between objects. This assumption allows us to represent the cluttered environments by a mathematical graph. On this graph, three elementary search strategies which guided an agent displaced away from its known route will be investigated: systematic, random and deterministic. Moreover, combinations of random and deterministic search will be studied, as it has been shown to be an optimal search strategy under certain conditions [74]. We address the following aspects systematically: (1) the probability of the agent finding a route with the different search strategies mentioned above; (2) determining the best search strategy to use; and (3) the link between the route type and the probability of finding the route again.

3.3 Results

3.3.1 Searching time and displacement

An agent in our simulation is assumed to know only a single route in an otherwise unknown environment (fig. 3.1, green line, for the shortest route). Here, the route known by the agent is the shortest route (for comparison with other route types, see section 3.3.3). Displaced away from its route, the agent is, therefore, at an unknown location (fig. 3.1 purple square). At the unknown location, the agent cannot follow a memorized direction associated with the location, since it has no representation in its “brain” of this location, i.e. this location has never been seen or never been learned by the agent. Nevertheless, the agent needs to move in order to find its route again.

Although the objects surrounding the agent do not provide information about the goal location, they still provide information about the topology of the immediate environment and, thus, may constrain the agent's next movements. In other words, the agent needs to make decisions based on the surrounding objects and, in this way, structures its search behaviour to find its route again. The environment being represented by a mathematical graph, i.e. a network of paths, the agent only makes a decision at an intersection of paths.



(Legend on next page)

Figure 3.1: Graph representation of a cluttered environment. A cluttered environment containing objects (black dots) was transformed into a network of the agent's potential paths (red lines), derived from triangular meshing of the objects (grey dashed lines). The agent makes a "decision" at an intersection of paths, i.e. a "decision" point (red dots). A series of paths connects the food source (backslashed square) to the home location (slashed square). Three route types: "shortest" (green), "safest" (blue) "and non-topological" (black), are shown. After a displacement from the route (e.g. grey square), the agent is led by a search strategy (e.g. deterministic search in orange, random search in purple). The probability of moving along one path by one or the other search strategy is shown in the upper right box. The displacement from the route is measured in the number of decision points (numbers in the lower left box).

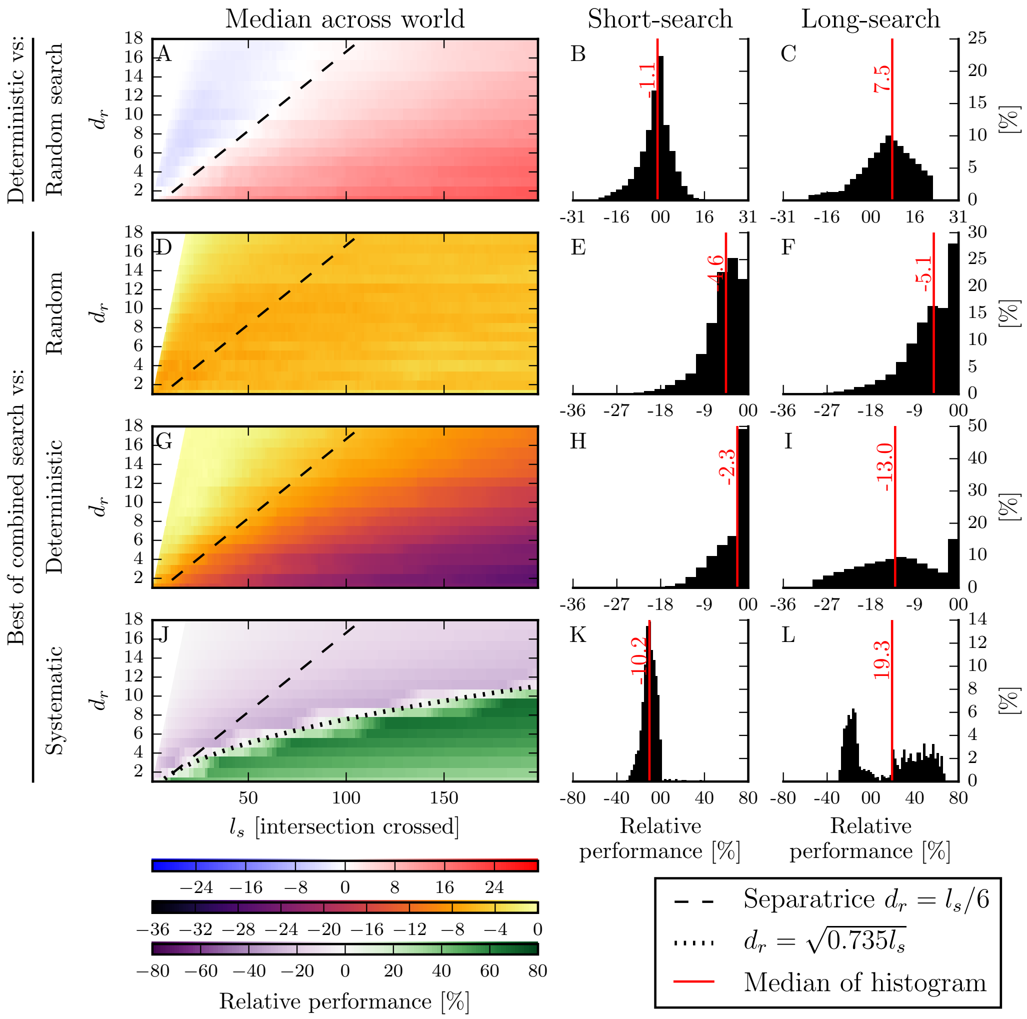
The agent displaced to an unknown location and then searching for its route cannot search forever. It may, for example, run out of energy [32]. To take this into account, the agent has been allowed to search only for a maximum distance travelled. The agent, thus, stopped searching either when the route has been reached or its "energy" has run out. Therefore, the route can be found by the agent only with a given probability. Moreover, the agent may be displaced by different distances from the route. The displacement, here, is measured in terms of intersections between paths (i.e. number of "decision" points) (fig. 3.1 box). The probability of finding the route depends, therefore, on the displacement from the route and the maximum allowed distance travelled during the search (also measured in terms of intersections crossed). Obviously, the agent has a null probability of finding the route if it has been displaced further away than the maximum allowed distance travelled for its search. Moreover, the probability of finding the route obviously depends on the search strategy. Therefore, the probability of finding the route needs to be compared between search strategies for different displacements from the route d_r and maximum allowed distance travelled during the search l_s.

Different search strategies are conceivable. The agent may decide randomly to pass to the right or the left of an object (random strategy). By contrast, the agent may determine its path according to local information, such as the distance between the objects surrounding the agent. This deterministic search strategy has been employed in our study and is based on the rule: “avoid venturing along narrow paths” (fig. 3.1, box). The agent may, moreover, combine the random and the deterministic strategy by switching between them from time to time. A combined search strategy is, therefore, a concatenation of a deterministic search strategy, i.e. the agent follows the rule for a given distance, moves along a random path, then follows the deterministic rule again for another given distance, etc. A large variety of combined search strategies exists, depending on the statistics of switching (see section 3.5 for details). One among them will be the best strategy, in the sense that it leads the agent to its known route with the highest probability. This strategy will be called the best combined search strategy, with the random and deterministic strategies conceived as extreme combinations and, thus, belonging to the combined search strategies. Since the deterministic and the random search strategy are the simplest search strategies among the combined search strategies, they will also be scrutinized with respect to the probability for the agent finding its route again.

By comparing the random search strategy with the deterministic search strategy, we observe two cases [fig. 3.2A]. When the agent can search for longer than six times the displacement from the route, i.e. it is displaced only relatively little compared to the maximum allowed distance travelled during the search ($l_s > 6d_r$, here l_s is the maximum allowed distance travelled searching, and d_r the displacement from the route), the random search has a higher probability of leading the agent to the route than the deterministic search strategy [fig. 3.2C]. By contrast, when the agent is allowed to search only for less than six times the displacement from the route, i.e. is displaced far from the route relative to the maximum allowed distance travelled during the search ($l_s < 6d_r$), the deterministic search strategy outperforms the random one [fig. 3.2B].

Among the deterministic, random and the variety of combined search strategies, one of them will lead the agent with the highest probability to the route, i.e. the best of the combined search strategies. In our simulation, the best of the combined search strategies has been determined independently for each environment, for a range of displacements from the route and for a variety of maximum allowed distance travelled during the search. If the best of the combined search strategies outperforms the deterministic or random search strategy by only a little, the agent should use the deterministic or the random search strategy, for the sake of simplicity. Indeed, when the agent is displaced far from the route ($l_s < 6d_r$), the best of combined search strategies has a similar probability of leading the agent to the route as the deterministic search strategy (the median gain in performance is only 1.3% [fig. 3.2H]). By contrast, when the agent is displaced by only a little ($l_s > 6d_r$), the best search strategy outperforms the deterministic (median gain in performance 12% [fig. 3.2I]) and the random search strategy (median gain in performance 6% [fig. 3.2I]) significantly.

Finally, the systematic (i.e. a spiral section 3.5) search strategy needs to be compared with the best of the combined search strategies. This strategy will certainly lead the agent to its route when the agent is allowed to travel a sufficiently long distance. We observe that the systematic search strategy leads the agent to the route with a higher probability than the best of the combined search strategies if the agent is allowed to move more than $1.36d_r^2$ (d_r being the displacement from the route). An agent, either knowing that it may travel a long distance to reach the route or having not been displaced far from the route, should search systematically. It will then most probably find its route. On the other hand, if the agent has no knowledge about how far it has been displaced from its route, it may use another search strategy better to maximize its chance of encountering its route.



(Legend on next page)

Figure 3.2: Comparison of performance of systematic, deterministic, random and best of combined search strategy to find a shortest route again. The agent, led by one of the search strategies, has a certain probability of reaching the shortest route. The performance of a search strategy is the probability of finding the route known by the agent, for different displacement from the route, d_r (y-axis of A,D,G,J) and with different maximum allowed distance travelled searching, l_s (x-axis of A,D,G,J). The first row shows the relative performance of the deterministic and the random search (A,B,C). The second to the fourth rows show the relative performance of the best of the combined search with the random search (D,E,F), the deterministic search (G, H, I), and the systematic search (J,K,L) respectively. The median of the relative probabilities of search strategies across environments is shown in the first column (A, D, G, J) for a given displacement from the route and maximum distance travelled searching. The median of the relative probabilities is colour-coded and quantified in percent. Towards the blue (respectively, red), the deterministic (respectively, random) search strategy outperforms the random (respectively, deterministic) search strategy. The distribution in relative performance across worlds, routes, displacements and allowed distance travelled searching is shown for short search (B,E,H,K) and long search (C,F,I,L), respectively. A short search (respectively, long search) is a search with a distance travelled shorter (respectively, longer) than six times the displacement from the route. The dashed black line separates short and long searches. The dotted black line is the equation $l_s = 1.36 d_s * *2$. The red lines overlaid on the histograms are the medians of the histograms.

3.3.2 When is a search strategy the best?

A unique and global best search strategy is unlikely to exist, because the optimal strategy depends on the context and constraints imposed on the system. An agent should opt for the strategy which might be the best within the current context. The context is in this study: the agent knows a single route in an otherwise unknown environment and it has been displaced by a given distance from its route. Furthermore, the agent needs to reach its route before it runs out of energy or is caught by a predator. The maximum allowed distance travelled searching, therefore, constrains the search of the agent.

Is it possible for the agent to guess the search strategy that leads to the route with the highest probability, only knowing the context defined above and the constraints the agent has to cope with? We found, by comparing the systematic search strategy with the best of the combined strategies, that the systematic search strategy is the best strategy as long as the agent can travel at least $1.36d_r^2$ searching (d_r being the displacement from the route)[fig. 3.2J][fig. 3.3B]. In the other cases, the agent should use the best of the combined search strategies. The best of the combined strategies has been determined independently for each environment, displacement from the route and maximum allowed distance travelled searching. The best combined search strategy differs between environment, displacement from the route and maximum allowed distance travelled searching.

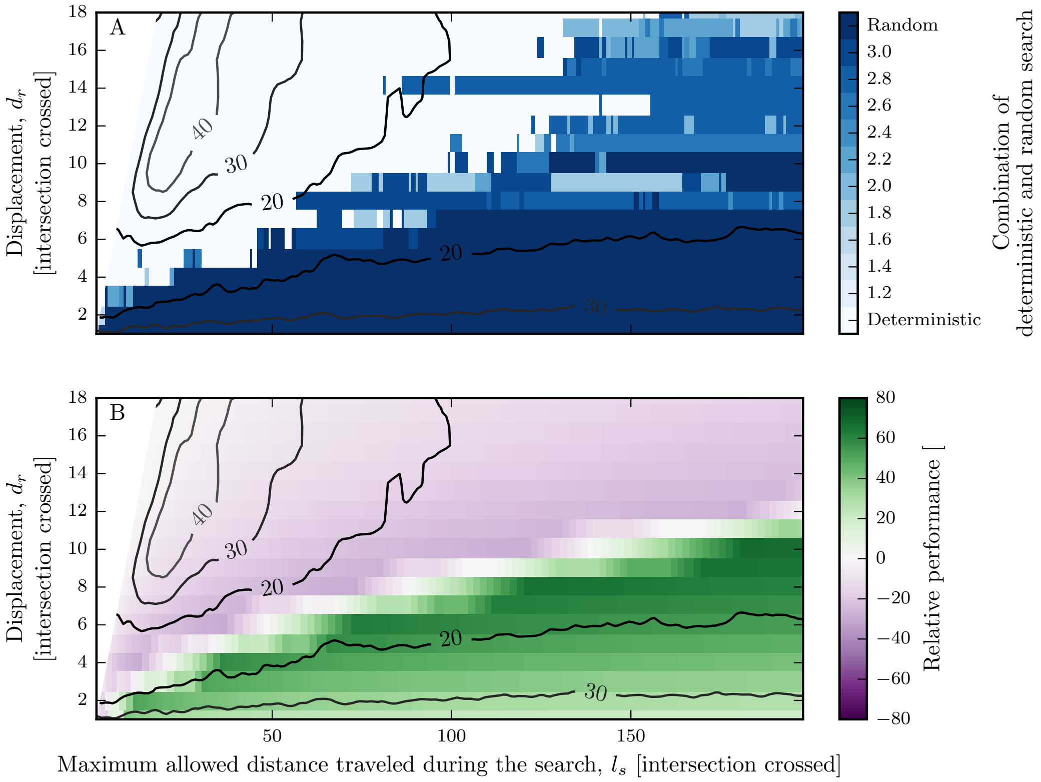


Figure 3.3: Which strategy is the best search strategy? A) Each combination of deterministic and random search was assigned to a colour. The combination, being the best of the combined search strategies across most environments, was shown for different displacement from the shortest route (y-axis) and the maximum allowed distance travelled searching (x-axis). The grey contour lines represent the probability of the combined search strategies to be the best. The chance level is $100/12 = 8 + 1/3\%$. B). The relative performance between the best of the combined searches and the systematic search is shown in Figure 2J. The grey contour lines are in A.

Is it possible for the agent to guess the best combined search strategy if it knows about its displacement from its route and the allowed distance it can travel, but does not know anything about the environment beside its route? A candidate for a good search strategy is, under these conditions, the search strategy that is the best on average across multiple environments. However, a combined search strategy may be the best by chance. Twelve combined search strategies have been used, therefore, the chance level for a combined search strategy to be the best is $100/12 = 8+1/3 \approx 8.33\%$. A candidate, therefore, needs to be the best in a significant number of environments, more than the chance level. We observe that the deterministic (respectively, random) search strategy is likely to be the best when the agent is not allowed (respectively, allowed) to travel a long time relative to its displacement from its route [fig. 3.3A]. In the mid-range, there is no candidate for the best search strategy, as every combined search strategy has the same probability of being the best (i.e. below chance level). The agent, in this case, has to gamble on which search strategy to use.

3.3.3 Probability of finding the route: which route?

Two locations in a cluttered environment can be connected by several routes. In our study, we consider three types of routes: the shortest route, the safest route and the non-topological route. How an agent can establish and follow such a route is beyond the scope of this study. A route is a series of paths; here, each path has a list of characteristics, e.g. its length. A route is generated by minimizing the sum of one of the path characteristics. The minimization of the distance travelled (respectively, the venture along narrow paths) gives a shortest (respectively, safest) route between two locations. However, the representation of the environment as a mathematical graph is also based on the topology of the environment. To avoid biases in our results, a third route type, independent of the topology, needs to be used. Therefore, every path had an additional characteristic, a random value assigned to the path. The random value is independent of the topology of the environment. The minimization of the random value assigned to the path gives a non-topological route between two locations. A route type having one of the route characteristics minimized (e.g. the distance travelled is minimized for a shortest route) has certain advantages over the others (e.g. the shortest route is shorter than any other routes).

Does the probability of finding a route depend on the route type? To investigate this problem, the agent was allowed to search for a distance proportional to its displacement from the route and used the best of the combined search strategies. An agent displaced two intersections from the route, for example, can search for the route at up to twelve intersections crossed (the proportionality factor (k) is, in this case, 6). We considered three proportionality factors: short search (k = 2), mid-range search (k = 6) and long search (k = 18). Each agent found its route with a given probability. The percentage of routes found with at least a probability of x percent was compared across three different probability factors and the three route types (shortest, safest and random) fig. 3.4A. We observed no differences between the route types, but, as expected, an increase in the probability of finding the route with increasing allowed search distance relative to the displacement from the route (k).

Does the best search strategy depend on the route type? The probability of finding its route back does not depend on the type of route the agent is looking for. However, the agent may have to employ a different strategy to reach the different types of route to reach optimal performance, since the best of the combined search strategies may differ across route types. The likelihood of each combination of deterministic and random search being the best of combined search was, therefore, assessed for the three route types and three proportionality factors k (see last paragraph). We observe that this likelihood does not differ significantly between route types, but changes dramatically with the changing proportionality factor k [fig. 3.4B]. In accordance with the previous section, the likelihood of random (deterministic) search being best increases with the increasing (respectively, decreasing) proportionality factor k fig. 3.4B.

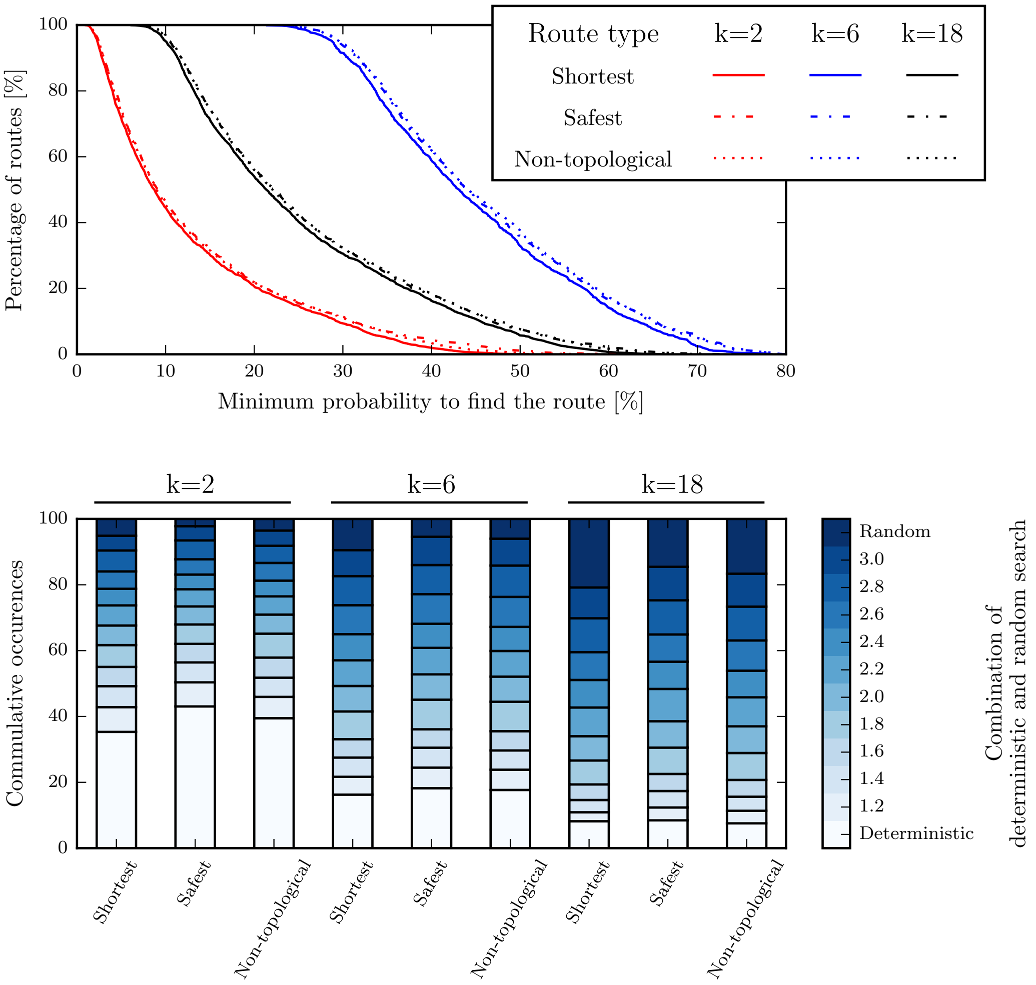


Figure 3.4: The effect of the type of routes. The agent was allowed to travel at most 2, 6 or 18 times its displacement from its known route, searching for its route by using the best of the combined search strategies. A) The percentage of route y found with at least a probability of x percent was plotted for the three proportionality factors $k = 2, 6, 18$, and the three route types, “shortest”, “safest” and “non-topological”. B) The percentage of each combination of deterministic and random search being the best of the combined search strategies (evaluated across environment) was plotted for the three proportionality factors and three route types as a stacked bar. The colour code is the same as in Figure 3.

3.4 Discussion

We have represented a cluttered environment by a mathematical graph to investigate the search strategy to employ to find a route again after having been displaced away from it. Three fundamental search strategies were investigated: systematic, deterministic and random, and ten combinations of deterministic and random search. We found that there is no unique optimal search strategy (among the variety of search strategies tested). An agent displaced away from its habitual route may opt for a search strategy based on the distance between its current location and its route, and the distance it plans to travel searching for its route. The agent, therefore, needs to estimate the distance between its current location and its route. We will discuss in the three following paragraphs how an agent may estimate this distance for three different situations: a displacement caused by wind, due to the need to avoid an incoming danger and due to navigational errors. A discussion on how long an agent plans to search will be followed by the implication of our finding for autonomous robots.

A light-weight agent may be displaced by wind (or other types of current) away from its route. The wind exercises a force on the agent so that the agent moves away from its route. Although the agent may compensate for wind drift [121, 158], strong gust of wind may displace the agent [172]. An agent may be able to sense this force directly by mechanosensors, or indirectly, by detecting an unexpected shift in the visual field. Insects can use both strategies to measure the displacement, as they are covered with hairs which can sense deflection and, therefore, the force exercised by the wind, and equipped with motion-sensitive neurons, which detect the motion of objects during the displacement. By integrating the force or the apparent motion, the agent may estimate how far it has been displaced from its route [171, 112]. The agent may also be able to estimate the direction of its displacement – ants, for example, can remember the wind direction – when displaced by a gust of wind [172]. In our simulation, the direction of the displacement was unknown by the agent. An agent with this knowledge may move precisely in the direction of the displacement [172]. However, as the estimate of the displacement may be noisy, the agent should not follow the direction of the displacement exactly. One may see this strategy as a combined search strategy; here, the deterministic rule is: follow the estimated direction of the displacement (in our simulation the rule was: avoid venturing along narrow paths).

An agent may be displaced by the need to avoid an incoming danger, such as a predator. The agent has to switch from a route-following behaviour to an escape behaviour. After the escape behaviour, the agent will be at an unknown location. The agent may have monitor its motion during its escape behaviour and, thus, similar to a displacement by wind or other current, the agent may know how far it has been displaced.

An agent may be away from its route due to navigational errors. The agent may follow a wrong direction at one location along the route. After some time, the agent may realise that it is no longer following the route and, therefore, starts searching for it. It may be difficult to estimate the displacement from the route, because the agent may not know when or where it made a mistake. Nevertheless, while an optimistic agent may assume that it is not displaced far from its route, a pessimistic agent may assume it made an error at the beginning of its journey and, therefore, both agents can have an estimate of the displacement from the route. The agent may also have paid attention to the scenery or other navigational cues along its erroneous journey and, thus, be able to backtrack until it finds its original route. Ants, for example, are able to backtrack when they have missed their nest entrance [175].

The agent displaced away from its route needs to estimate how long it plans to travel searching in order to opt for a good search strategy. The agent may want to find its route again as fast as it can for numerous reasons. An agent carrying resources from one location to another, for example, may be in competition with other agents. The quicker the agent can find its route, the more resources it will be able to gather. Wandering in the world, searching for its route, may also be dangerous. Indeed, the longer the agent is lost, the more likely it is that the agent will encounter a danger, such as a predator. Finally, the agent being limited in energy, cannot search for its route forever; the travel distance of the agent is bounded by the energy it can use.

Autonomous robots have gained many skills over the past few decades. One can foresee autonomous robots delivering post and packages, or cups of coffee. The task of the robot is then very similar to a foraging insect; it has to move along a route to deliver post to a series of houses. Although a robot delivering post may not need to avoid an impeding danger or be displaced strongly by wind, it can still quit its route due to navigational errors or be forced to make a detour due to a blocked street. The robot will, thus, find itself away from its route and will need to search for it. One of the greatest limitations of robots is their relatively short autonomy compared to animals. A robot, therefore, needs to find its route again quickly and, therefore, may opt for the best search strategy based on an estimate of the distance between its current location and the route, and the energy it can spend searching.

3.5 Materials and Methods

3.5.1 Graph representation of cluttered environments

A cluttered environment is an area or volume scattered or excessively filled with objects. Our cluttered environments contained 3750 objects placed randomly. A thousand environments have been generated. The environments have been simplified by considering only the position of the objects, forming a cloud of points. The cloud of points was then used to create a network of paths. Inspired by the centring behaviour of flying animals [142, 35, 76, 8, 88], the path of the agent always passes in the middle between two neighbouring objects. To determine neighbouring objects, we meshed the cloud of points by a Delaunay triangulation (i.e. the outer circle of each triangle does not contain any point). The centre of each triangle has been connected to each neighbouring triangle by a path composed of a line going from the centre of the first triangle to the middle of the common edge and a line going from the middle of the common edge to the centre of the second triangle. The centre of the triangle and the paths connecting two neighbouring triangles are referred to as a “safe location” and a “safe path”, respectively. The ensemble of safe locations and safe paths acts as a network of streets and intersections, and is a mathematical graph of connectivity 3.

3.5.2 Route and route type

A route from two distant locations in an environment is a succession of paths. In our graph representation of the environment, a route is a series of the agent’s paths between two safe locations. Every route in our environment ends at the location at the centre of the environment. It acts as the central place for the forager (e.g. its nest). Each route starts at another location in the environment. The start is not further away from the central location than a fifth of the total size of the environment.

Two locations can be connected by a large number of routes. Each route can be described by a series of variables, such as its length and its “safeness”. The agent’s paths pass in the middle between two objects. Therefore, a measure of the safeness of a path is the Euclidean distance between the path and those objects, i.e. half the length of the edge of the triangle crossed by the path. The shortest route has the minimal length, i.e. the sum of the lengths of the safe path is minimized. Similarly, minimizing the sum of the inverse of the safeness of each path along the route leads to the safest route. The safest and shortest route are two types of route based on the topology of the environment. A route type not based on the topology of the environment was also used. A random value was assigned to each path in the environment. The non-topological route was the route minimizing the sum of those values assigned randomly. The Dijkstra algorithm has been used [63] to find the shortest, safest and non-topological route between each safe location and the central safe location.

3.5.3 Search and searching strategies

When the agent is displaced from its route, it has to search. The search starts from an unknown location (a safe location not on the route), and lasts for an allowed number of intersections crossed, i.e. an allowed search distance. During a search, at each intersection, the agent moves along one of the two paths (the agent cannot backtrack directly). The agent could use one of the 13 search strategies: systematic, random, deterministic or ten combined search strategies.

An agent following the systematic search strategy would be led to move along a clockwise or counter-clockwise spiral. The agent turned clockwise (respectively, counter-clockwise) whenever the node reached by the resulting motion of the clockwise turn did not lead to a node visited previously.

An agent following the random search strategy selects the path randomly with equiprobability, i.e. it has 50% (50 %) chance to take the right (respectively, left) path.

An agent following the deterministic search strategy always moves along the path with the highest safeness.

An agent following one of the ten combined strategies switches between deterministic and random search strategies. A combined strategy consists of a deterministic search strategy interspaced by random path selection. The number of intersections crossed between two random path selections is determined by a heavy tailed distribution defined by:

$$p(l) = Cl^{-\alpha} \quad \text{for} \quad l \in \mathbb{N} \tag{3.1}$$

$\alpha = [1.2, 1.4, 1.6, 1.8, 2.0, 2.2, 2.4, 2.6, 2.8, 3.0]$ has been used.

The best of the combined search strategies in a given environment, displacement from the route and maximum search distance travelled, was the random, deterministic or one of the ten combined search strategies. The best of the combined search strategies is the strategy leading the agent to its route with the highest probability.

3.5.4 Procedure

A hundred environments containing 3750 objects placed randomly were generated. The environment has a limited size and the border may bias the simulation. To overcome this problem, only the locations within the square at the centre of the environment with a width of a fifth of the environment were used for the routes. The search starts for every location on the graph. The first intersection between every route with every search is then calculated. The probability of finding the route is the ratio of the number of searches intersecting the route and the total number of searches.

3.6 Supporting Information

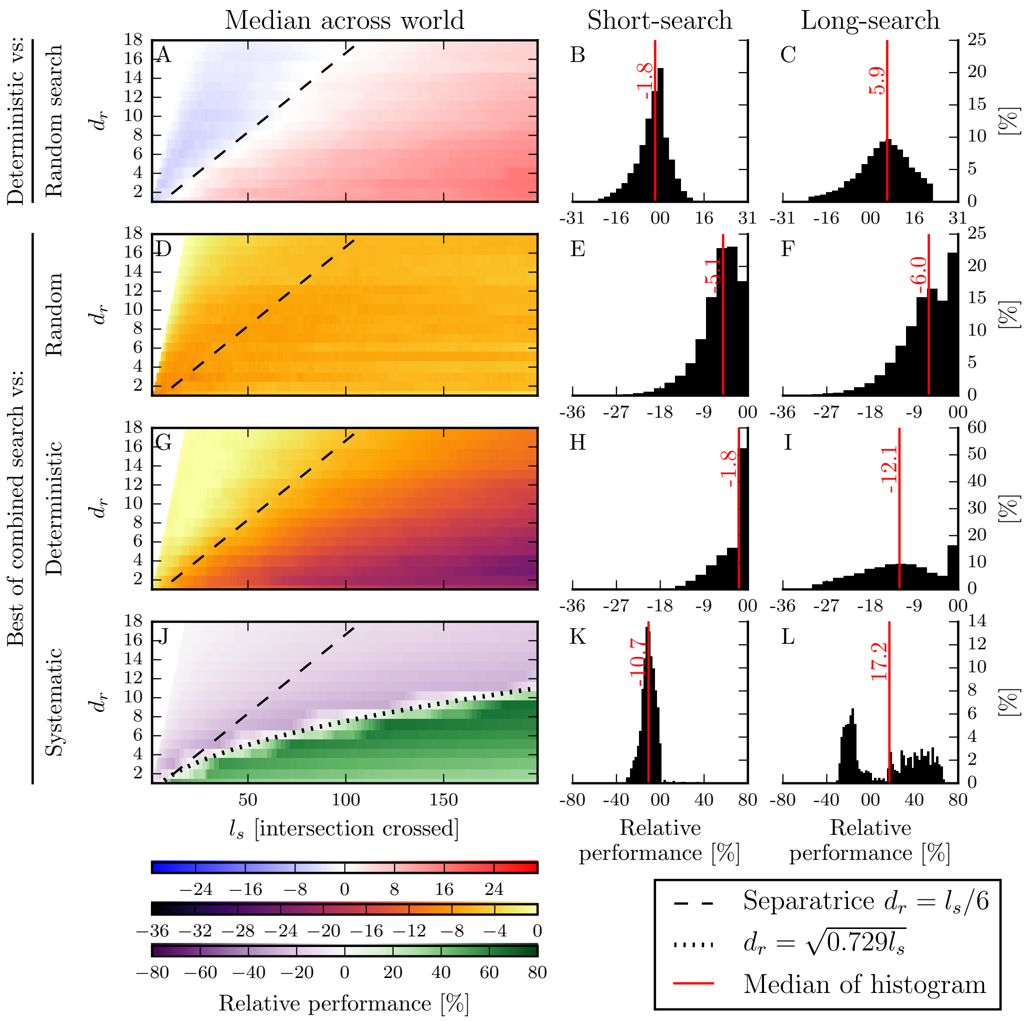


Figure 3.5: Comparison of performance of systematic, deterministic, random and best of combined search strategies to find a safest route again. Same as Figure 3.2, but for the safest route instead of the shortest route

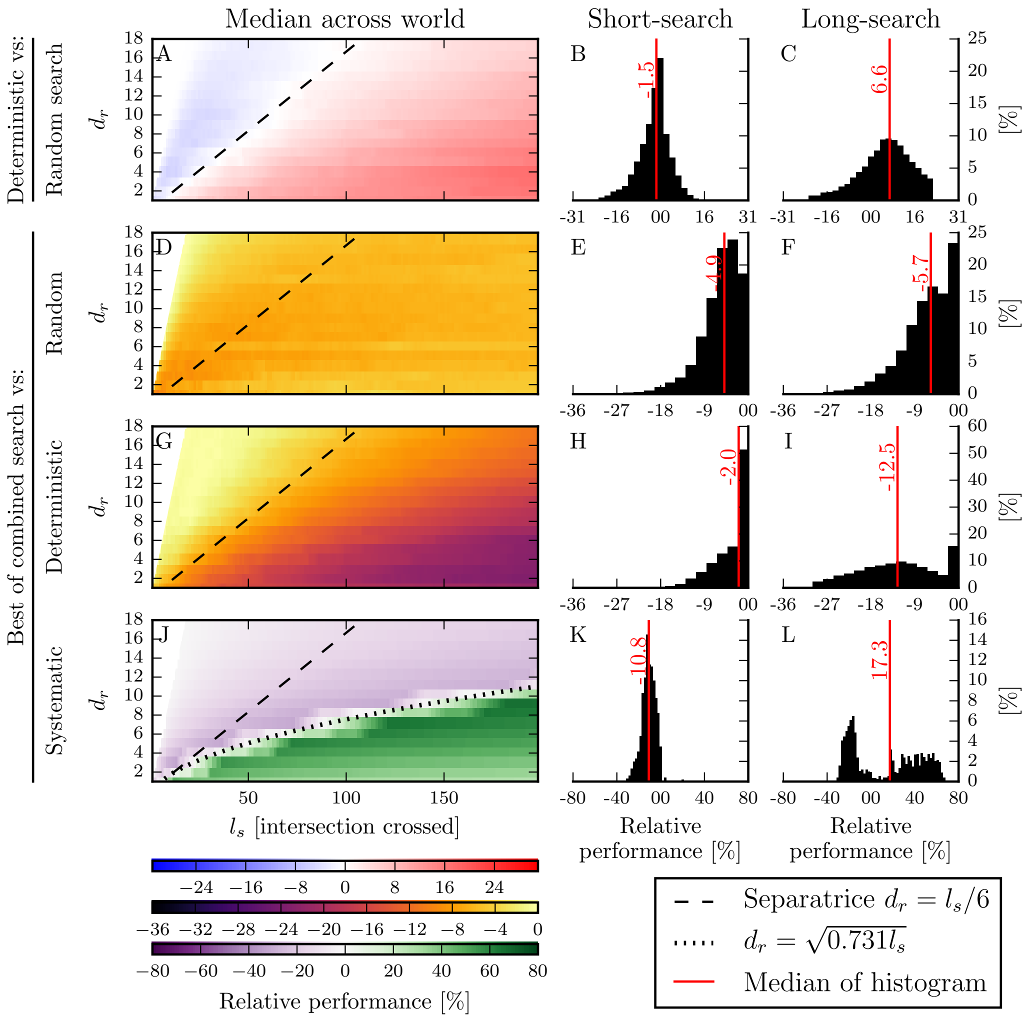


Figure 3.6: Comparison of performance of systematic, deterministic, random and best of combined search strategies to find a non-topological route again. Same as Figure 3.2, but for the non-topological route instead of the shortest route

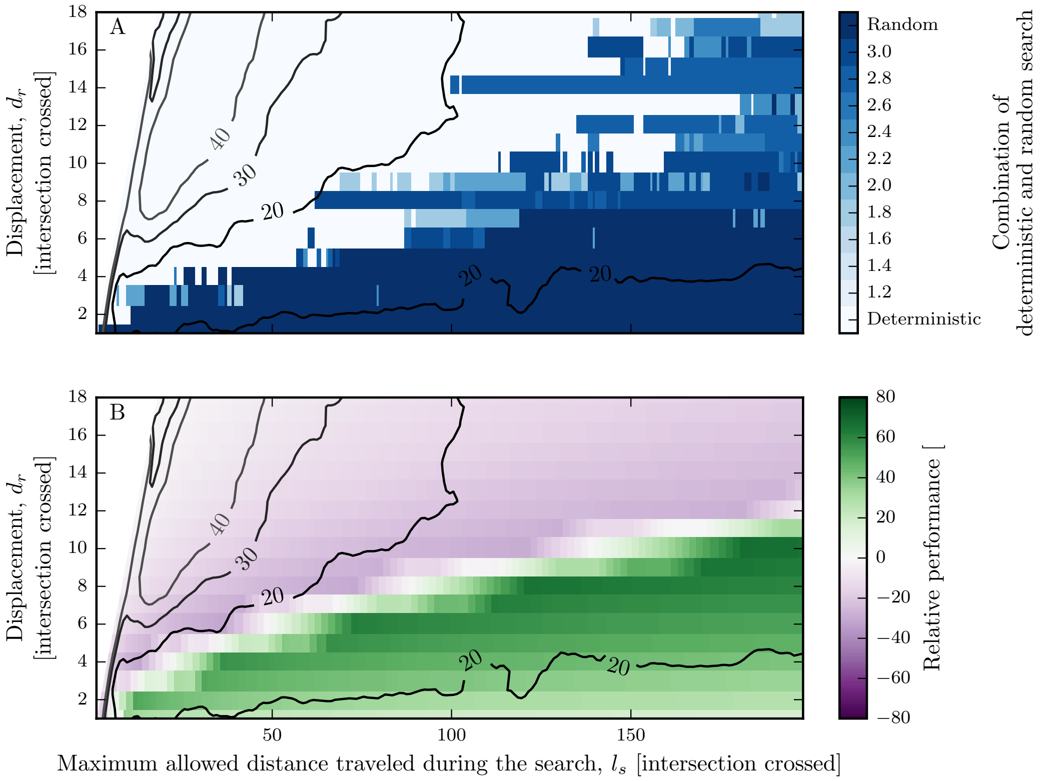


Figure 3.7: Which strategy is the best search strategy? Same as Figure 3.3, but for the safest route instead of the shortest route

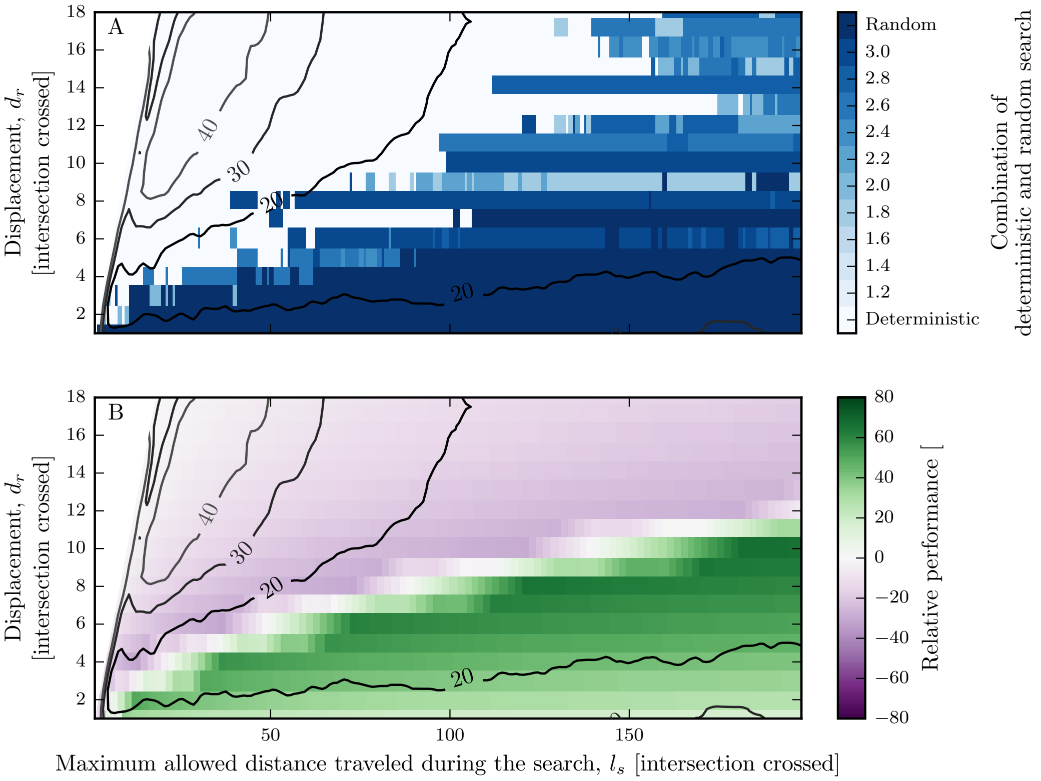


Figure 3.8: Which strategy is the best search strategy? Same as Figure 3.3, but for the non-topological route instead of the shortest route

4 The risk of being lost: how detailed a route should be memorized?

This chapter is planned to be published as Olivier J.N. Bertrand, Jens P. Lindemann, and Martin Egelhaaf (2016). The risk of being lost: how detailed a route should be memorized?

4.1 Abstract

When navigating in a cluttered environment, it is efficient to follow and memorize a route between two locations being visited repeatedly (e.g. home and work). However, little is known regarding how detailed a route should be memorized without increasing significantly the risk of being lost. The literature describes several strategies to follow and search for a route when an agent, a biological or technical system, has been displaced from it. Route-following algorithms often reduce the amount of memory required to follow a route by using sparse representations of the sensory input. However, the memory

required can also be reduced when all the places in the environment are not memorized. In this chapter, we use graph representations of virtual cluttered environments to investigate the amount of memory that can be saved by memorizing only locations necessary to follow a route, the risk for an agent staying lost away from its sparsely memorized route and whether or not the agent should memorize locations based on the local topology of the environment. By comparing the risk of an agent not finding a sparsely memorized route and the risk of not finding a fully memorized route, we found that the agent should memorize at least 60% of the locations along a route. The local topology does not provide any advantage when deciding which locations the agent should memorize.

4.2 Introduction

When new to a city, we struggle with planning, learning and optimizing the way between two locations, for example, between our home and a grocery store. We have to think and pay attention to our surroundings in order not to get lost. After a while, we do not need to even think about how to move and just follow “our” route between home and work; everything is already planned [65]. A route is a series of directions, i.e. a set of instructions associated to places along the route (e.g. turn right at the next intersection). One may follow a route without experience; for example, when an expert navigator provides a set of instructions to a naive navigator via communication before the start of the naive navigator's journey [92]. One may ask a pedestrian (i.e. expert navigator), for example, for the directions to reach a restaurant, a grocery store, a museum, etc. The amount of information passed by communication is, however, limited. To explain a route to a naive navigator, the expert omits many details and, therefore, gives the minimal set of instructions to follow the route. The set of instructions consists of directions associated to places, and those are memorized by the naive navigator. This leads to the question: How detailed do we need to memorize a route in order not to get lost?

Humans are not the only animals facing the problem of following a route. Central place foragers, such as ants and bees, are some of the masters of navigation in the animal kingdom. Despite their miniature brains, they forage several times a day [141, 144, 136], up to a few kilometres [111][56, Chapter 6] away from their nests to gather food and bring it back to their homes. To achieve this amazing performance, they probably remember a route efficiently [1]. Therefore, central place foragers are excellent systems to study the mechanisms underlying navigation. Ants follow idiosyncratic routes in cluttered environments, i.e. environments filled excessively with objects [79]. The idiosyncrasy indicates that the ant follows the route based only on its own experience, i.e. does not use information given by others, such as odours laid previously on the route by another ant. Route-following behaviour has been successfully modelled by a neuronal network associating route directions to familiar places [1]. Using visual information surrounding the ants, they know in which direction they have to move to follow the route at any location along the route. However, ants can follow a route with intermittent visual feedback [84]. The ability to follow a route with intermittent visual feedback indicates that the ants do not recall the route-direction continuously, but do it only from time to time or only at certain locations.

How can one follow the route without recalling the route-direction at every location? An agent in a maze, for example, only needs to remember where to turn right or left. By default, the agent moves forward, i.e. it maintains its direction. The idea used in the maze can be extended to cluttered environments. Exemplarily, the agent updates the direction to follow when a large change in the visual surroundings is observed (e.g. the number of objects) [138] In those two examples, the agent follows the route by combining calculated and predefined directions. While the calculated directions are obtained by the memorized association of route-direction and place, the predefined directions are “naive” responses, such as following the previous motion direction.

Calculated directions are stored in the “brain” of the agent and, therefore, require memory space. An agent, having only a limited amount of memory, can only follow routes with a limited number of calculated directions. However, the longer a route is, the higher the number of calculated directions. Therefore, the foraging range of the agent is limited. To increase this range, the agent can rely on predefined directions. By combining predefined directions and calculated directions, the agent can, thus, travel further than when using only calculated directions. The agent should always minimize the use of calculated directions for the purpose of increasing its foraging range. However, we will see that it may be risky for the agent to do so.

Locomotion of an agent does not always happen according to its plan. It may diverge from its normal route, for instance, when attempting to avoid an impending danger or as a consequence of navigational errors or distractions. Thus, the agent will end up at an unknown location. Once displaced from its route, it has to search for the route to find it again. It may, therefore, be necessary to memorize enough locations along the route to be able to recognize one of them during the search and not stay lost for a long time.

In the third chapter of this thesis, it was shown that there is generally no optimal search strategy to find a route. An agent has a certain probability of finding its route again within a given time. Depending on the distance between the agent's current position and its route, and the time the agent plans searching, the agent has a higher probability of finding its route again with a given search strategy than with others. However, the route was fully memorized by the agent in this model analysis; the agent did not used predefined direction along the route. Therefore, we may ask how the probability of finding a route depends on the degree of memorization of the route. The degree of memorization is defined as the number of memorized locations divided by the total number of locations along the route.

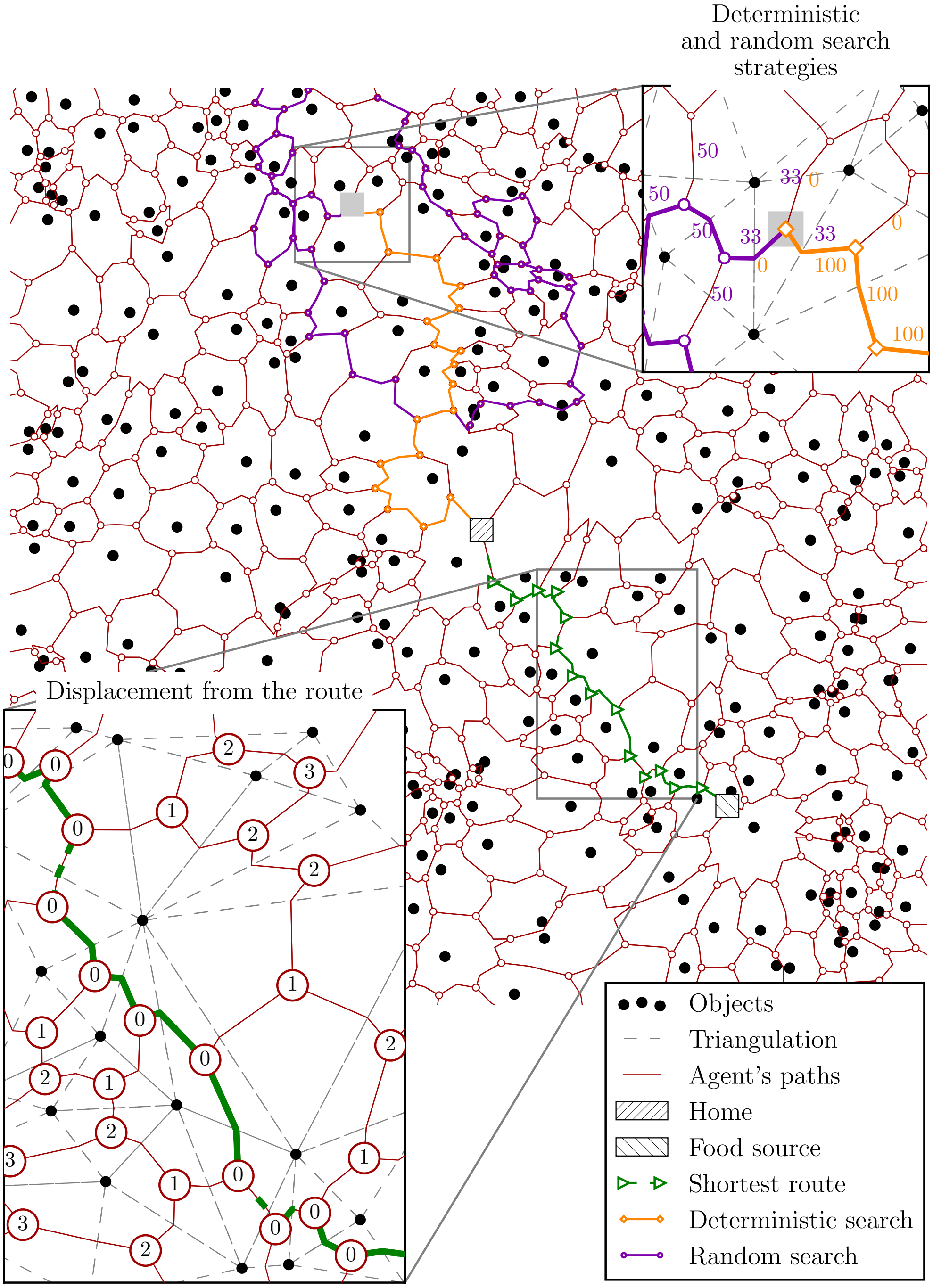
Inspired by the centring behaviour of flying animals between objects [142, 35, 76, 8, 88, 12, 123], we represented the environment by a series of paths centred between objects, forming a network of paths. Using this representation, we will first assess the probability of finding routes with different degrees of memorization. The predefined direction, decreasing the degree of memorization, will first be based on the "naive" response: avoid venturing into narrow paths. This predefined direction is based on the local topology of the environment. The term "topology" refers to the arrangement of the objects, paths and intersections of paths in the environment. To generalize our analysis to other predefined directions, we will compare routes memorized with topologically predefined directions to routes memorized with non-topologically predefined directions. The non-topologically predefined directions are based on a "naive" response which does not depend on the arrangement of the objects, paths and intersection of paths in the environment, but on a random value associated to a path. The "naive" response can be phrased as: moves along the path with the lowest random value. The agent still needs to memorize certain locations to follow a route. Therefore, we assess how much memory an agent can save by using topologically or non-topologically predefined directions. The search strategy leading the agent with the highest probability to its route may differ between a fully memorized route and a sparsely memorized route. This difference will be investigated.

4.3 Results

4.3.1 Finding sparsely and fully memorized routes

An agent, in our simulation, is assumed to know only a single route in a given cluttered environment (fig. 4.1, blue line). This route is the shortest one between the starting location (e.g. feeder) and the goal location (e.g. home), and is represented in the memory of the agent as a series of directions associated with known locations (fig. 4.1, blue triangle). How the agent extracts this information from its memory is beyond the scope of the present study and will be discussed later. We, thus, assume that the agent always recognizes the previously memorized location when the location is encountered during locomotion, and then moves in the memorized direction associated with this location. The route is defined as fully memorized if all locations along the route are memorized by the agent. However, in order to save memory, it may be sufficient for an agent to follow the route by memorizing only certain locations along the route. The locations to memorize are those with a route-direction different to the topologically predefined direction. The agent moves along the route as follows: when a memorized location is encountered, it moves in the direction associated with the memorizing; when a non-memorized location is encountered, it moves in the predefined direction (moves away from the narrow path). The route is said to be only sparsely represented in the memory of the agent if only a subset of locations along the route and the corresponding directions of locomotion is memorized (fig. 4.1, thick blue lines, dotted lines represent non-memorized locations).

Figure 4.1: Cluttered environment as a graph. The environment containing many objects (black dots) forms a mesh of triangles (dashed grey lines). The mesh is used to create a network of paths followed by the agent (red line). Three paths form an intersection or decision point (open red dot). The agent is assumed to know a single route (blue line) between its goal location (yellow square) and its home location (grey square). The route can be either fully memorized (dashed thick blue line) or sparsely memorized (thick blue line). The agent could follow the sparsely memorized route by using a predefined direction at those locations that were not memorized. In this example, the predefined direction was based on the “naive” response: avoid venturing along narrow paths. Only locations with their associated predefined directions aligned to the route-direction were not memorized. The agent was displaced from its route to an unknown location (e.g. purple square). The displacement was quantified in terms of the minimal number of intersections crossed between the agent’s current position and its route (see Box: displacement). The displaced agent was led by a combination of search strategies: deterministic (orange line) and random search (purple line). Both strategies are shown in the upper right box. The number at the side of the path indicates the probability of the agent following one or the other path when led by the search strategy.



(legend on previous page)

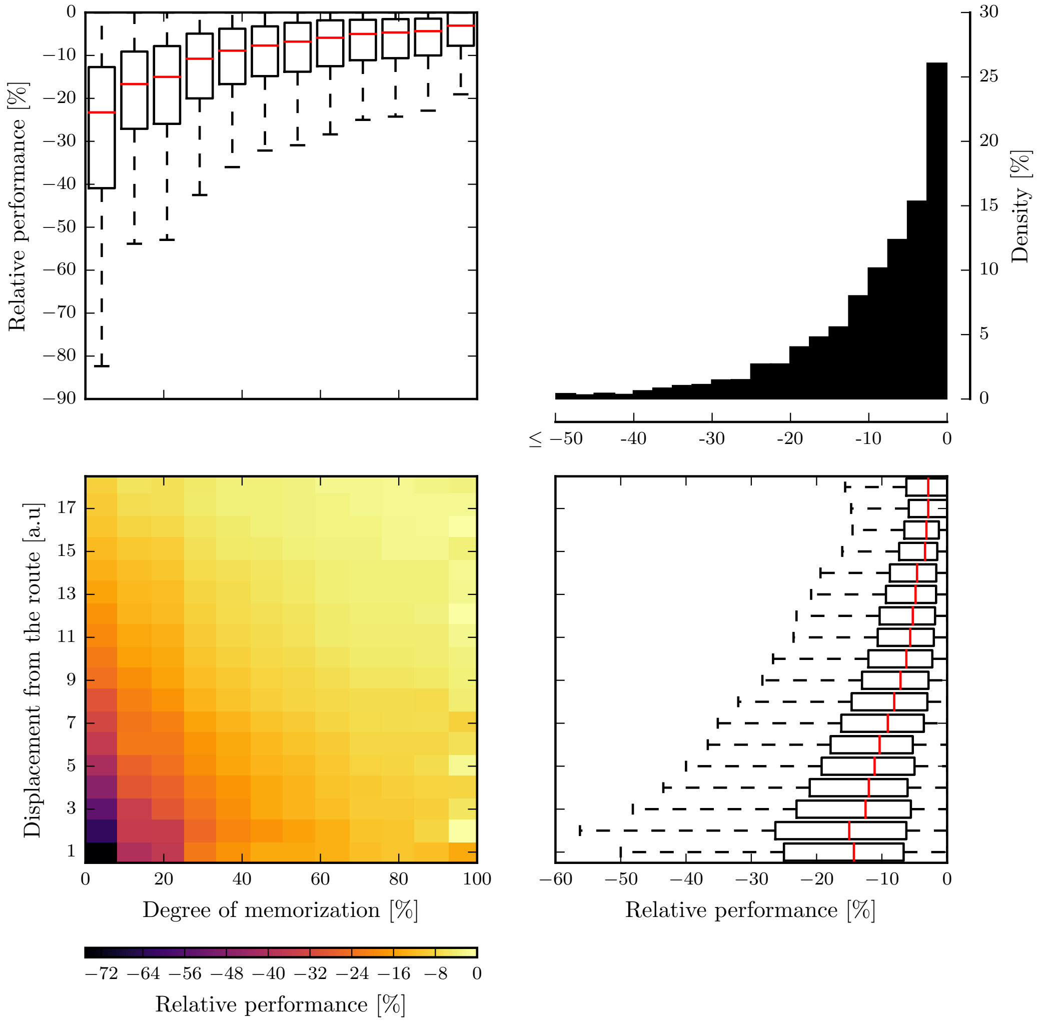
The journey of an agent following a route in a cluttered environment may be disrupted by some external influence that displaces the agent to an unknown location. After such an unintended displacement, the agent is assumed to try to find its memorized route again. To test how well the agent can find its route by some search strategies, we displaced it by different distances from its route. The displacement was quantified as the minimum number of intersection points between the fully memorized route and the start of the agent's search (fig. 4.1 Box-Displacement).

The displaced agent is assumed to search for the route it learned previously. This can be done by employing various strategies. In our simulations, an agent is not able to spend an infinite amount of time searching for its route, because it may, for example, run out of energy [46]. The agent has, thus, been allowed to search only for a given distance travelled. Displaced to an unknown location, the agent stopped searching when either one of the locations memorized along the route has been reached or it ran out of energy, i.e. after having crossed 100 intersection points. The agent will, therefore, reach the route only with a certain probability, depending on both the distance between the agent and its route and the search strategy employed by the agent (for full route, see chapter 3).

The agent employs a search strategy to find the route again [129]chapter 3. The agent could employ one of 12 search strategies: a random search, a deterministic search or one of ten combined search strategies. The searching agent – using a random search strategy – coming along one path may decide at an intersection point to move along one of the two remaining paths randomly (in our representation of the cluttered environment, each intersection connects three paths). An agent employing such a random search strategy, therefore, moves left or right at each intersection with a 50% probability. Alternatively, the agent – using a deterministic search strategy – moves along one of the two paths according to a deterministic rule. The rule, for example, can be based on local information available at the intersection. In our simulation, inspired by the behaviour of animals flying through cluttered environments, the rule was to avoid venturing into narrow paths, i.e. to follow the path with the widest distance between the neighbouring objects that had to be passed. Finally, the agent may combine the two search strategies by switching between the deterministic and the random search strategy from time to time. A large variety of combined search strategies exists, because the agent may spend more or less time following one or the other. One of the variety of combined search strategies (including the random and the deterministic search strategies) will lead the agent back to its route with the highest probability ("best of combined search strategies"). The best of combined search strategies has been determined for a large number of displacements from a number of previously learned routes, several environments, and depends on whether the route has been fully learned or is only sparsely memorized.

To answer the question: "How risky is it to memorize a route sparsely?", in order to find it after a displacement of the agent, we compared the probability of finding the sparsely memorized route to the probability of finding the route memorized fully again. We found that the sparser the route is memorized, the lower the probability of finding the route again [fig. 4.2A]. However, the relative probability flattens when more than 60% of the total number of locations along the route are memorized (i.e. degree of memorization of 60%). In other words, the risk of missing a sparsely memorized route increases significantly when the agent has memorized less than 60% of the route.

The probability of finding the route also depends on the displacement from the route. We found that the relative probability is higher for a small than a large displacement from the route. The risk of missing a sparsely memorized route, compared to a fully memorized one, increases with decreasing displacement from the route. This is surprising. Indeed, the agent searching for its route was, from time to time, following the deterministic search strategy, i.e. avoiding venturing into narrow paths. An agent using a deterministic search strategy and encountering a non-memorized location of the route will move along the route, because the predefined direction is based on the "naive response": avoiding venturing into narrow paths, i.e. the same rule used during a deterministic search strategy.



(Legend on next page)

Figure 4.2: Risk of memorizing a route only sparsely. An agent which is displaced from its route could travel up to 100 intersections to find its route again. Therefore, it could find a route with a given probability. The relative performance is the difference between the probabilities of finding the sparsely memorized routes divided by the probability of finding the fully memorized route again. The median relative performance is shown for different degrees of memorization of the route, and displacements from the route (lower left heat map). The distribution of the relative performance is shown for different degrees of memorization (upper left) and displacements from the route (lower right). The upper right histogram shows the distribution of relative performance for all routes in all environments. The relative performance decreases significantly for routes with a degree of memorization lower than 60%. It also decreases with decreasing displacement from the route.

4.3.2 What to memorize?

We have seen that it is important to memorize more than 60% of the total number of locations along the route in order to find it again after having been displaced from it. The agent was moving in a predefined direction when it encountered a non-memorized location on the route. The predefined direction was to avoid venturing into narrow paths, based on the local topology of the environment. However, other predefined directions can be used to follow a route. The colour, size and/or shape of the surrounding objects can be used. The predefined direction can, for example, be: “avoid the path between blue objects”. This predefined direction is not based on how objects, paths and intersection of paths are spatially separated in the environment and is, therefore, not linked to the local topology. A sparse route followed by the agent with a topologically (respectively, non-topologically) predefined direction is called a topologically (respectively, non-topologically) sparse route.

Is it more probable to find a topologically sparse route than a non-topologically sparse route? Let us assume that the agent had memorized either a topologically or a non-topologically sparse route. For a topologically sparse route, locations on the route with a route-direction aligned to the predefined direction “avoid venturing into narrow paths” was not memorized by the agent. Paths in the environment were associated with a random number to simulate the non-topological direction. The random number may, for example, be understood as the degree of ugliness of a path. At an intersection, the non-topological direction was pointing toward the path with the smallest number (e.g. towards the less ugly path). The random attribution of the weights represents a non-topologically predefined direction without loss of generality. For a non-topologically sparse route, locations on the route with a route-direction aligned to the predefined direction “move towards the path with the smallest number” was not memorized by the agent.

Similar to the previous section, the agent was displaced from its route, employed the best search strategy and could travel for up to 100 intersections. It had, therefore, a certain probability of finding its route again. The probability of finding the topologically sparse route was compared to the probability of finding the non-topologically sparse route. We have seen that the probability of finding a sparsely memorized route depends not only on the proportion of the route memorized, but also on the amount of displacement from the route [fig. 4.2A]. Therefore, only routes with a similar amount of memorization were considered.

We found that the probability of finding a topologically sparse route or a non-topologically sparse route are similar [fig. 4.2]. The distribution of the probabilities are broad [fig. 4.2A, fig. 4.2D], i.e. certain topologically sparse routes are easier to find than non-topologically sparse routes, but the opposite also occurs. The trend, however, is that the non-topologically sparse routes are easier to find than the topological ones [fig. 4.2B]. We conclude that how a route is memorized, i.e. which predefined direction is used to not remember certain locations, does not increase or decrease significantly the risk of staying lost for a long time.

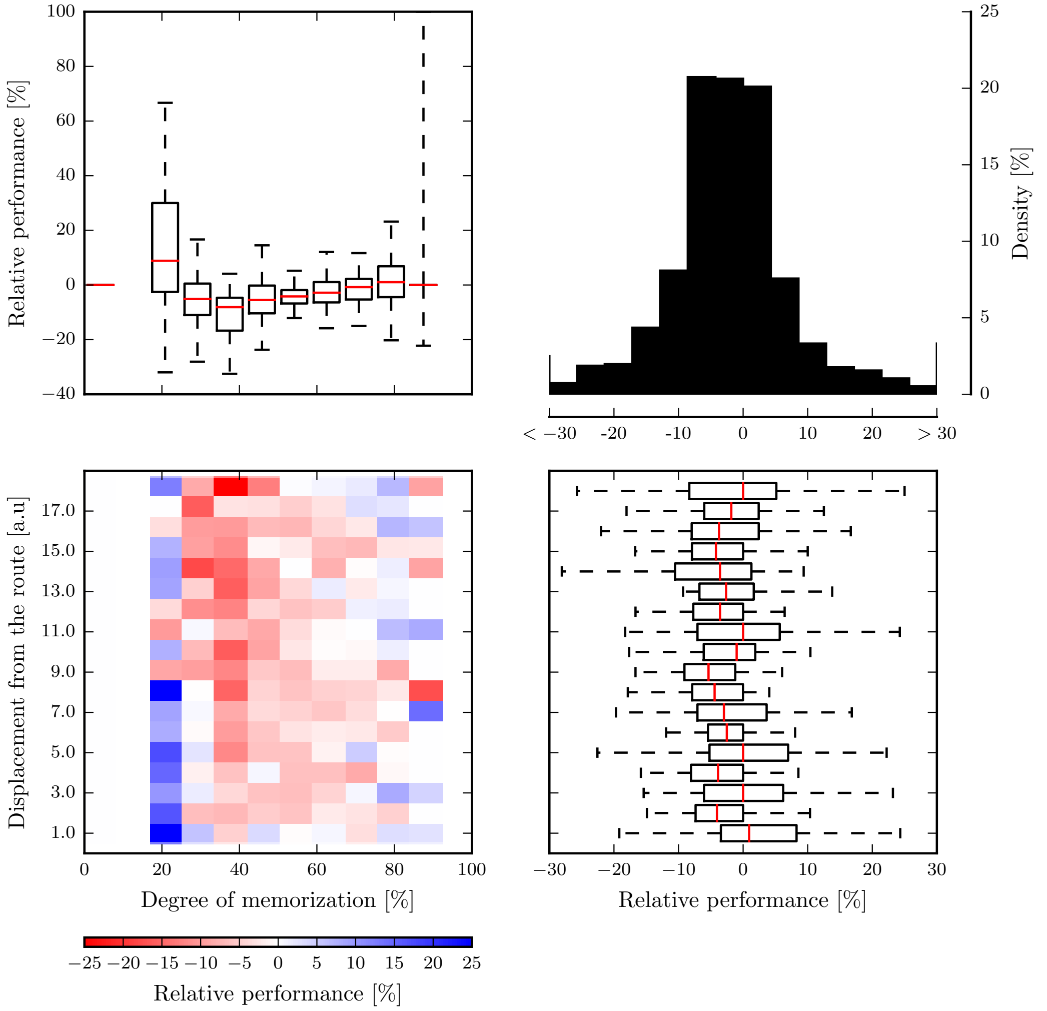


Figure 4.3: Topologically or non-topologically predefined directions. Similar to Figure 4.2, this figure compares the probabilities of finding the topologically sparse routes again with the probabilities of finding the non-topologically sparse routes again. The topologically (respectively, non-topologically) predefined direction was avoiding venturing along narrow paths (respectively, follow the path with the smallest width). See text for details.

4.3.3 The minimal number of locations memorized

Until now, only the shortest route has been considered, however, two locations can be connected by different routes. The route, for example, may have the distance travelled or the time travelled between two locations minimized. A given route type (e.g. shortest route) presents, therefore, an advantage compared to other route types (e.g. the shortest route is shorter than any other routes). We have seen that the agent may memorize a route sparsely without losing the ability to follow it. Indeed, an agent memorizing locations along the route with a route-direction not aligned to a predefined direction can still follow the route. However, the minimum number of memorized locations required to follow the route may depend on the route type. Therefore, certain routes may require a higher degree of memorization than others to follow them.

We considered three different route types: the “safest”, the shortest and the sparsest route. The safest route between two locations minimizes the risk of venturing along narrow paths and is inspired by the centring behaviour of flying animals. The route is the safest in terms of collision avoidance, because the agent does not need to have a precise collision avoidance algorithm to follow this type of route (the route is mainly composed of wide paths). The shortest route is the one with the shortest distance travelled, whereas the sparsest route is the one with the minimal number of memorized locations. How such a route can be established by an agent goes beyond the scope of this chapter.

The route type with the smallest number of locations that the agent has to memorize in order to go from a starting point to a distant goal is the sparsest route, by definition. This route type gives a lower limit on the number of locations that the agent needs to memorize to go from the starting location to the goal in the environment. Therefore, the sparsest route requires the lowest degree of memorization. We found that the agent may save around 70% of memory (i.e. a degree of memorization of 30%) by following such a route [fig. 4.4, dashed black line].

On a shortest route (respectively, safest), the agent may only need to memorize around 70% (respectively, 60%) of the total number of locations when the agent uses topologically predefined directions at the non-memorized locations. Finally, when the agent uses a non-topologically predefined direction, it has to memorize around 80% of the total number of locations. It does not depend on the route type, because the route types are all based on the topology of the environment.

4.4 Discussion

We have studied the risk of staying lost in cluttered environments. The agent, in our simulation, only knew a single route and was challenged to find its route again after having been displaced away from it. We argued that it is possible to memorize only certain locations on the route to follow it, but that it is risky to rely on such sparse

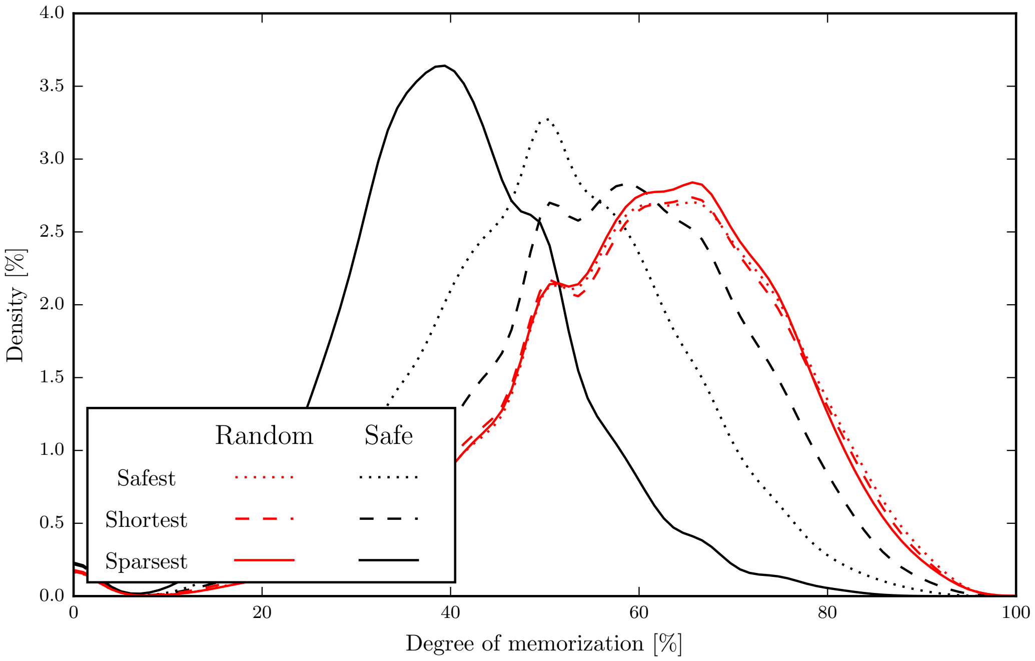


Figure 4.4: Distribution of degree of route memorization. The route was sparsely memorized with either a topologically (black) or non-topologically (red) predefined direction. The degree of memorization is the percentage of locations memorized. The distribution was obtained for 27,800 routes in 100 randomly generated environments. It was calculated for different route types: shortest (dashed line), safest (dotted line) or sparsest route (thick line).

representations of a route if the agent is displaced from it. The risk is to remain lost for a long time, because a sparsely memorized route is harder to find than a fully memorized route. We found that an agent should remember at least 60% of its route in order to not stay lost for a long time.

Many animals follow habitual previously learned routes to commute repeatedly between two locations (e.g. ants [79], bees [21], humans [65]). The greater the distance between the two locations is, the longer the route and, thus, the greater the number of locations to be remembered. An agent is, therefore, faced with a trade-off between remembering only certain locations along the route in order to forage long distances, and the risk of staying lost for a long time. In the two following paragraphs, we will discuss how often the agent may be displaced from its route and the risks caused by such a displacement.

What is the probability of being displaced from the route? As discussed in the third chapter (chapter 3), the agent may be displaced due to external factors, such as the passive displacement by wind or other currents for light-weight agents, the need to avoid an impending danger or internal factors, such as navigational errors. An agent may be able to compensate for wind-drift and, thus, maintain its course even in windy conditions [121, 158]. However, the agent, when avoiding a danger, will be displaced away from its route. An animal, for example, may cross areas with predators. A more technical example is a scouring robot. Assuming the robot located a victim of a natural catastrophe, the robot may have to guide a scouring team to the precise location of the victim. On the way to the victim, the robot will hopefully follow the shortest route, but may have to avoid danger, such as sliding terrain, fires or risk of explosion. In such situations, the robot should remember as many locations on the route as possible to minimize the risk of staying lost for a long time. Finally, the agent may be displaced by navigational errors. The probability of committing an error may be known by the agent; an agent with experience, for example, may know how many errors it has committed during its lifetime. Therefore, an agent can assess the risk of being lost by navigational errors directly and, thus, adjusts the number of locations it needs to learn in order to not stay lost for a long time.

What is the risk of being displaced away from the route? It is obvious that the further away the agent is displaced from its route, the lower the probability of finding its route again (see fig. 4.5). An agent should, therefore, limit its displacement from its route. As mentioned above, the agent may compensate for wind-drift and, thus, minimize a displacement caused by wind (or other currents). An agent avoiding an impending danger may try to stay close to its route while moving away from the danger. When an agent runs away from a predator, for example, the agent may perform complex manoeuvres in the vicinity of the route until the predator has lost track of the agent. Finally, a displacement caused by navigational errors may be difficult to limit, because the agent needs to “realise” it is no longer following the route, i.e. “realise” it made an error.

What is the gain of memorizing only certain locations along the route? The main gain is to use as little energy and memory as possible. Indeed, every time an agent compares its current sensory input to the one stored in its memory, it uses energy for the comparison. Therefore, an agent remembering, for example, 50% of the locations along the route, but still able to follow the route, will access its memory less often than an agent memorizing all the locations along the route and, thus, use less energy. Moreover, an agent not using its memory fully to memorize a route, may learn other routes to other locations. In a foraging context, learning more than one route can be tremendously beneficial. Indeed, if one location no longer provides resources, the agent can quickly reach the second location. The risk of staying lost, in this chapter, has been studied for an agent knowing only a single route. We believe that the risk of staying lost for an agent knowing multiples routes is lower than for an agent knowing a single route when the two agents know the same number of locations. The agent knowing multiple routes may find the first route when displaced away from the second one. However, the quantification of the risk of staying lost for an agent knowing multiples routes goes beyond the scope of this chapter.

What happens if the agent fails to recognize its route? Two locations may look fairly similar and, thus, hard to unambiguously recognize. When travelling along the route, this ambiguity can be resolved by using the context, for example, the past motion (the use of the past motion to improve place recognition is one of the advantage of self location and mapping [25]). In other words, the agent can "anticipate" which location needs to be recognized when travelling along a route. However, when the agent is searching for its route, such anticipation cannot be made, because the agent does not know which location along the route will be reached first. Therefore, the agent may fail to recognize a route location when searching and, thus, may cross its route without noticing it. In our simulation, the agent always recognized its route. However, the degree of memorization of a sparse route may be interpreted as the probability of recognizing a location along a fully memorized route (i.e. how well a location can be recognized; see for example [146]). Within this interpretation, an agent searching for its route needs to recognize it has reached a route location with a probably of at least 60% in order to not stay lost for a long time. In the case of a sparse route, the degree of memorization used in our simulation may be interpreted as the product of the probability of recognizing a location of the route and the ratio between the number of memorized locations and the total number of locations along the route.

In conclusion, we found that an agent should remember or be able to recognize at least 60% of the locations along a route in order not to stay lost for a long time. Although we explained our simulation with examples based on vision, the simulation can be used for other sensory modalities. The agent was moving along paths in cluttered environments. The network of paths can also be derived from the position of objects (as explained in this chapter), as well as from the position of olfactory or auditory cues. The agent may, for example, move along a route marked by conspicuous olfactory cues. When displaced away from its route, it moves along paths based on their odours. Thus, our findings are valid for any agent moving along a network of paths (intersections are always between three paths) and knowing a single route between two locations in the environment.

4.5 Materials and Methods

4.5.1 Graph representation of cluttered environments

A total of 100 environments containing 3750 objects placed randomly were generated. Inspired by the centring behaviour of flying animals [142, 35, 76, 8, 88], every environment was simplified by a network of paths centred between objects, i.e. a graph representation of the environment. Every graph had a connectivity of 3. For more details, see section 3.5.1.

4.5.2 Route and route type

A route connecting two distant locations in an environment is a series of paths. The central location of the environment was the end of every route in the environment. The route started from another distant location; here, this location was not further away from the central location than a fifth of the total size of the environment.

A route was either memorized fully or sparsely using either a topological or non-topological rule. An agent knowing a fully memorized route, knew every location along the route. An agent knowing a sparsely memorized route, knew only certain paths along the route, according to a rule. The topological rule was: memorize locations along the route with the route-following direction being follow the narrow path. The agent could, thus, still follow the route by avoiding venturing into narrow path when the non-memorized location along this route was encountered. The non-topological rule was based on a random value assigned to a path. The rule was: memorize locations along the route with the route-following direction being follow the highest value of the path. The agent could, thus, still follow the route by avoiding venturing into paths with the highest value when the non-memorized location along this route was encountered.

The start and the end locations were connected by three different route types: the shortest, safest and sparsest. The shortest route minimized the distance between the two locations, i.e. the sum of all path lengths along the route. The safest route minimized the inverse distance between objects and the route, i.e. the sum of the inverse between objects and paths along the route. The sparsest route minimized the number of memorized locations along the route; here, the rule to not memorize locations along the route was the topological rule. The Dijkstra algorithm has been used to find the shortest, safest and sparsest route between each location and the central location [63].

4.5.3 Search and searching strategies

When the agent is displaced from its route, it has to search. The search starts from an unknown location (a safe location not on the route) and lasts for an allowed number of crossed intersections, i.e. allowed search distance. During a search, the agent moves along one of the two paths at each intersection (the agent cannot backtrack directly). The agent could use one of the 12 search strategies: random, deterministic or ten combined search strategies (for details see section 3.5.3).

The best of the combined search strategies in a given environment, displacement from the route and maximum search distance travelled was the random, deterministic or one of the ten combined search strategies. The best of the combined search strategies is the strategy leading the agent to its route with the highest probability.

4.5.4 Procedure

A hundred environments containing 3750 randomly placed objects have been generated. The environment has a limited size and the border may bias the simulation. To overcome this problem, only the locations within the square at the centre of the environment with a width of a fifth of the environment are used for the routes. The search starts for every location on the graph. The first intersection between each route with every search is then calculated. The probability of finding the route is the ratio of the number of searches intersecting the route and the total number of searches.

4.6 Supporting Information

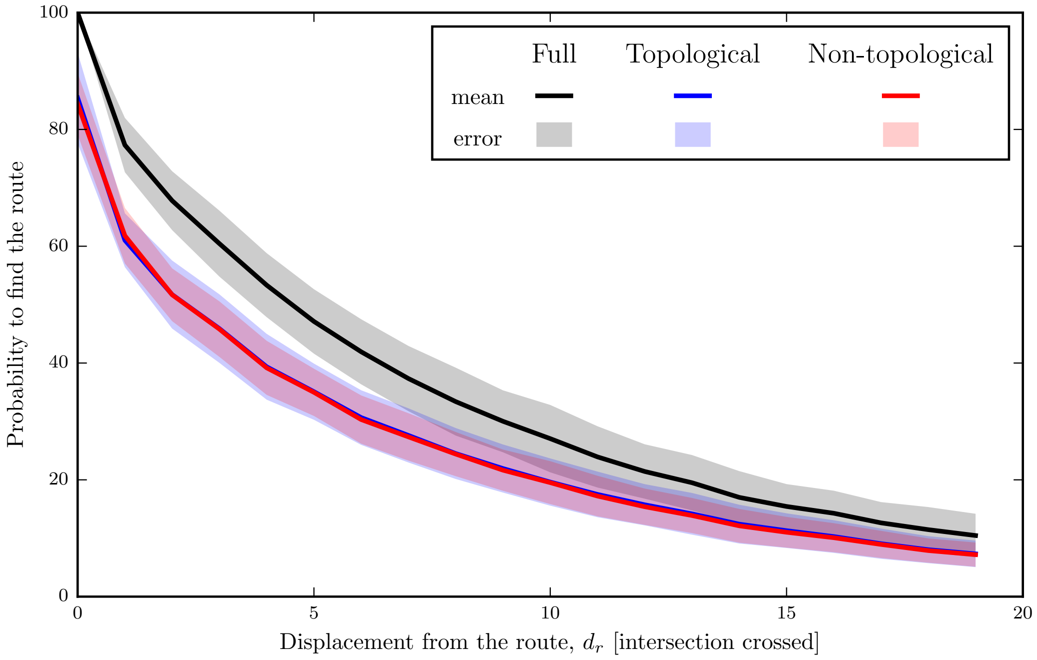


Figure 4.5: Probability to find the shortest again within a maximum allowed distance travelled of 100 intersections crossed and the best of the combined search strategies for different routes. The mean and standard deviation of the probability has been calculated accrossed 27800 routes for each type of memorization: full, topological, and non-topological

5 Concluding remarks

In the first chapter, we have seen that the complexity of navigation ranges from finding a destination by chance (searching) to moving along a novel route, i.e. a shortcut, between two locations in an environment (metric navigation). The focus of the thesis is route navigation, a behaviour often observed in the animal kingdom. In the three chapters following the overview of navigation, I analysed different aspects of route navigation: route-following in the second chapter, and finding a known route after having been displaced away from it in the third and the fourth chapters. Before starting a general discussion, the main conclusions of each chapter on route navigation will be summarized.

In the second chapter, I characterized a simple algorithm to follow routes. Guided by this algorithm, an agent followed a route without memorizing places. Instead, the agent was avoiding collisions with the surrounding objects, while trying to move toward its goal. While the collision avoidance algorithm was inspired by the visual system of flies, the goal direction was inspired by the ability of bees and ants to move directly toward their goal by following a direction derived from the integration of their motion. The route-following algorithm proposed in the second chapter is simple in terms of computation, because the algorithm is not based on a memorization of places and used a computationally parsimonious collision avoidance system.

In the third chapter, I addressed a completely different aspect of route navigation. The agent was assumed to know the route-direction at every location along a habitual route in a cluttered environment. The agent was then challenged to find its route after having been displaced away from it. The displacement of the agent may arise from navigational errors, wind or water current in the case of light-weight agents, or the need to avoid an impending danger. The displaced agent could use one of the following search strategies to find its route again: systematic, deterministic, random search or one of ten combinations of deterministic and random search. I found that the agent should adapt its search strategy based on the time it can invest in searching and the distance between its current location and the habitual route it needs to find. These findings were, moreover, robust for different types of routes: shortest, safest and non-topological routes.

In the fourth chapter, similar to the third chapter, the agent was challenged to find its habitual route again. However, in this chapter, the agent did not know every location along the route, but only a subset of these locations. Along the route, the agent was following calculated direction, i.e. direction associated to remembered locations along the route, and predefined direction, i.e. direction that a “naive” agent would follow. To guarantee the ability of the agent to follow its habitual route, the agent memorized the locations and their associated route-directions (calculated direction) only when the route-direction was not aligned with the direction that a “naive” agent would follow (predefined direction). The agent searching for its route could, therefore, cross the route without noticing it. I compared the risk of staying lost between fully and sparsely memorized routes. I found that the risk was significantly increased when the agent memorized less than 60% of the locations along the route.

Route navigation is one aspect of navigation (chapter 1). Therefore, each route navigation algorithm falls into one of the seven classes of spatial behaviour and is based on methods to solve the four fundamental problems of navigation. The following discussion will be focused on three different classes of route-following algorithm: place recognition-triggered response, familiarity response and global-local response. Each algorithm will be first defined and placed in relation to one of the seven classes of behaviour. Then the advantages and limitation will be phrased around three of the four fundamental problems of navigation: *detectability*, *recognizability* and *directionality*. The problem of *feasibility* – how do I get to other places - is assumed to be *a priori* solved by the agent, because it knows the route to follow and will, thus, not be discussed. The route-following algorithms will be highlighted, as in the first chapter, by the example of a biker travelling from Lyon to Bielefeld. The risk of an agent staying lost will also be considered. In a second section, the strategy potentially used by insects to follow a route will be inferred from their behaviour and an outlook on future research will be given.

5.1 Route navigation: strategies and formalism

5.1.1 Place recognition-triggered response

The agent has learned in which direction to move in order to follow the route by recognizing places along the route. This is a special case of state recognition-triggered response.

The place recognition-triggered response route-following strategy is one of the most intuitive route-following strategies, as humans use it to indicate directions to another person. However, this strategy is limited by the difficulty to recognize places. To recognize a place, an agent needs to compare the characteristics at the current place to the ones stored in its memory (section 1.2.2), and the characteristics of a place may be hard to detect or even change over time (section 1.2.1).

A route-following strategy based on place recognition can be formalised as follows. The route is a set of consecutive places, for the biker a series of intersections, hotel, restaurant, etc. Each place i has a set of detectable characteristics $\tilde{x}_i$ (e.g. a tree, the colour of a house or the topology of an intersection) and is associated with a direction to follow the route arg_i (e.g. take the first exit at the roundabout). To recognize the place, the agent uses a function $d(x_c, x_i)$ with x_c being the characteristics detected at the current location and x_i the memorized detectable characteristics at the place i, i.e. x_i is a subset of $\tilde{x}_i$. The route direction, i.e. the direction of the currently relevant place [1], is thus:

$$\overrightarrow{AP} = \begin{cases} arg_i & \text{if} \quad d(x_c, x_i) = 0 \\ \text{lost} & \forall i \quad d(x_c, x_i) \neq 0 \end{cases} \tag{5.1}$$

To identify a place unambiguously, an agent needs to have remembered enough characteristics of this place. Our biker, for example, should not only remember a given intersection as two orthogonally crossing streets, because most intersections will have this characteristic. An agent, thus, needs to remember characteristics that allow two different places to be unambiguously distinguished from one another. Therefore, to use a route-following strategy based on place recognition, the agent needs to encode a sufficiently large number of characteristics in its memory.

Any problem in detecting the characteristics of a place may fool the place-recognition algorithm and make the agent “believe” that it is placed at a location different from its actual location. The agent following its “belief” will, thus, move in an incorrect direction and, consequently, away from its route, i.e. the agent will commit a navigational error. On the other hand, any problem detecting the characteristics of a place may give a hint to the agent that something has gone wrong with route-following and, thus, trigger a search to find a known place. An agent may minimize the displacement from its route by reacting quickly to any missed recognition, thus, minimizing the risk of staying lost for a long time.

[1] The relevant place may be seen as an intermediate goal (see section 1.2.1 for more details)

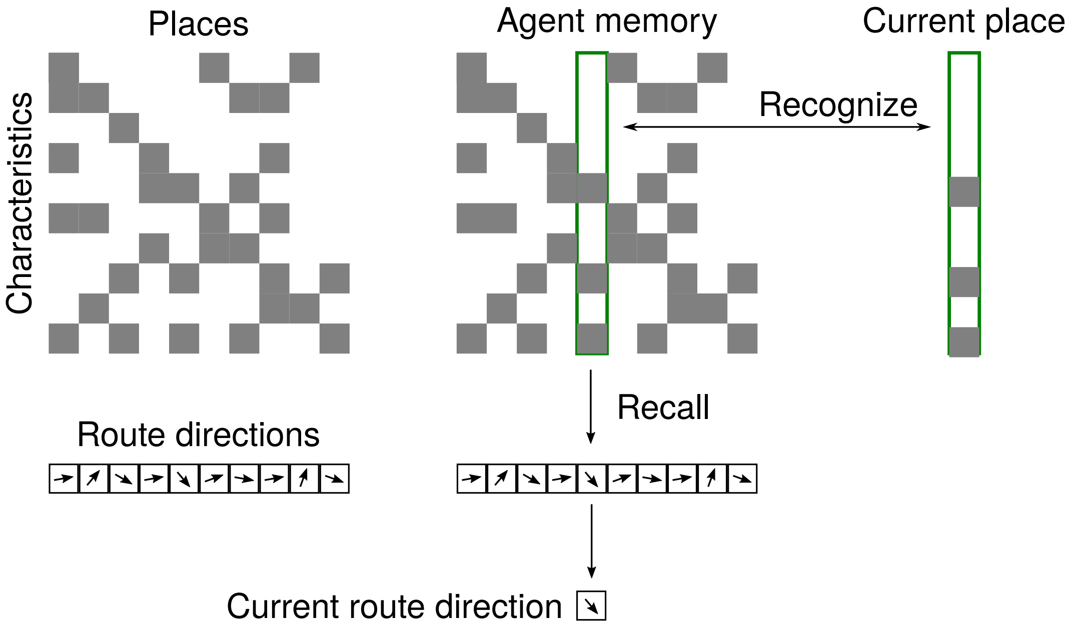


Figure 5.1: Placerecognition-triggered response schematic. The places along the route are associated with a route direction. Each place is described by a list of characteristics. The agent compares the characteristics of its current place with the ones stored in its memory. When a place is recognized, the route-direction associated with this place is recalled.

5.1.2 Familiarity response

The agent follows the route by moving along in the most familiar direction. The agent does not need to recognize the place to determine the route direction. This is a special case of a state recognition-triggered response.

The main advantage of a route-following strategy based on familiarity responses is that the agent can follow a route without recognizing places explicitly, thus, lifting some of the limitations of a route-following strategy based on place-recognition. An agent on the route moves along in the most familiar direction. The biker, for example, may have to take the third exit at a roundabout. Instead of remembering to take the third exit, it remembers to take the most familiar exit. To understand how a familiarity response could lead to route-following, we may think about this strategy as follows. The agent, instead of seeing the route as a sinuous path in the environment, sees the route as a straight line in a distorted environment. The distortion of the environment around the straight line is encoded in the agent's "brain" as a whole, i.e. the agent cannot identify specific places. This encoding is less demanding in terms of memory and computation, because redundant information can be merged together [6]. Our biker. for example, may take the third exit at different roundabouts along the journey. Instead of remembering them individually, the biker can learn the concept of taking the third exit at a roundabout.

A following strategy based on familiarity responses can be formalised as follows. The route is a set of consecutive places. Each place i has a set of detectable characteristics $\tilde{x}_i$ and is associated with a direction to which to move next arg_i. A function exists $x = f(\tilde{x}_0, ..., \tilde{x}_i, ..., \tilde{x}_n)$ which distorts the characteristics of all places along the route, such as $arg_i = 0$. To determine whether the current location of the agent is along the route, the agent uses a function $d(f(x_c), x)$, here x_c are the characteristics detected at the current location and x are the memorized characteristics of the route. The route direction, i.e. the direction of the current relevant place, is thus:

$$\overrightarrow{AP} = \begin{cases} 0 & \text{if} \quad d(f(x_c), x) = 0 \\ \text{lost} & \text{otherwise} \end{cases} \tag{5.2}$$

This strategy can only be used when the function f exists. The agent when lost, may start scanning the environment by taking different orientations to check whether other orientations are aligned with a familiar direction. When all orientations have been tested and none of them provides a familiar response, the agent should consider itself lost and start searching for its route. Similar to the place recognition-triggered route-following strategy, the agent may quickly realise that it has been displaced from its route. While an agent using a place recognition-triggered route-following strategy detects a displacement by checking whether it is at a known place, an agent following a route with familiarity responses detects a displacement by checking whether there is any familiar direction at a location.

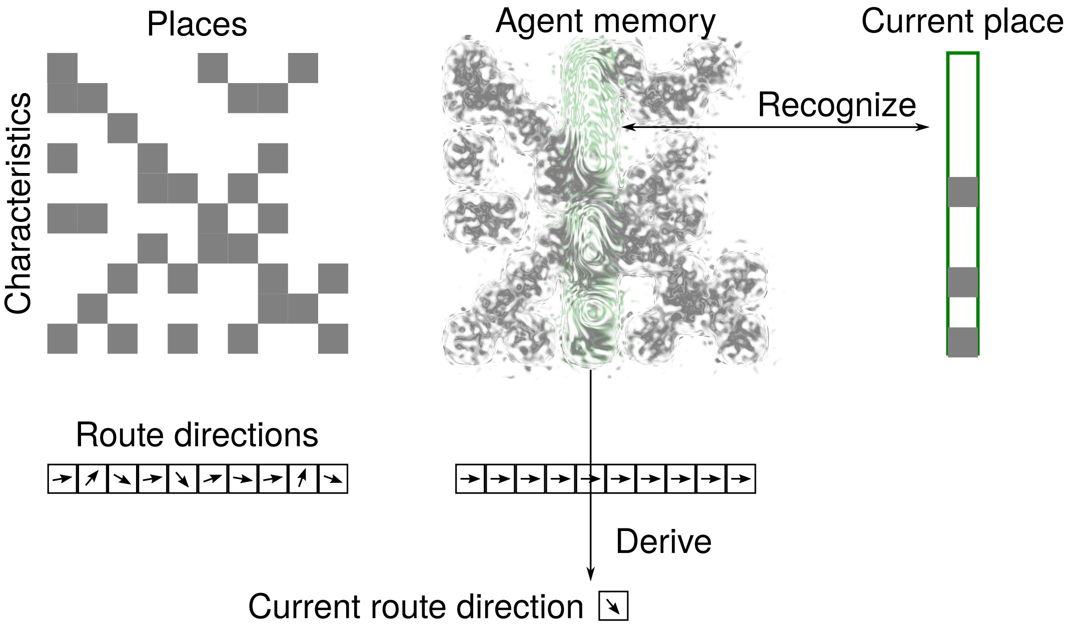


Figure 5.2: Familiarity response schematic. The places along the route are associated with a route direction. Each place is described by a list of characteristics. Instead of memorizing the places and the associated route directions, the agent memorizes a distortion of the places such that the route-direction is always the same along the route and, thus, independent of the current location. The agent compares the characteristics of its current place to the memory of the route. When the current place is familiar to the route, the agent follows the most familiar direction, i.e. the route direction.

5.1.3 Global-local response

The agent follows the route by combining a global direction and a local direction. While the global direction is the direction straight to the goal, the local direction guarantees that the agent avoids danger, i.e. object collision or predators. This algorithm derives from direction following. A route-following strategy based on global-local responses does not derive from state recognition-triggered response, but is a combination of two directions: the global direction and the local direction. The local direction steers the agent away from dangers, such as a collision with objects. The agent needs to detect certain characteristics of places, e.g. the distance to objects, so that it can avoid the danger, but is not required to recognize places along the route. The global direction is the direction straight to the goal. To compute this direction, the agent has to either see cues indicating the goal direction (i.e. aiming) or derive the global goal direction from its own motion. The biker, for example, has to travel along streets to avoid collisions with trees or buildings, but also wants to progress along their journey and, thus, wants to get close to the overall goal. The combination of the two directions leads the biker to follow the street without the need of recognizing any places along this street. Only when the global direction is derived from cues in the environment, does the agent obviously need to detect those cues reliably. The direction of the currently relevant place to which the agent moves is derived from the combination of the global direction and the local direction. Here, the importance to follow the local direction increases with the danger of, for example, an incoming collision.

The route-following strategy based on a combination of global-local responses can be formalised as follows. The route is a set of consecutive places. Each place i has a set of detectable characteristics $\tilde{x}_i$. The agent uses those characteristics to avoid danger, i.e. calculate a local direction $\overrightarrow{L}$. The agent also knows the global direction $\overrightarrow{G}$, either by detecting a conspicuous cue marking the goal location (i.e.

one of the detectable characteristics $\tilde{x}_i$ is the bearing of a conspicuous cue) or derived it from its own motion (i.e. one of the detectable characteristics $\tilde{x}_i$ depends on the motion of the agent). The direction of the next place to which to move is a function of the two directions:

$$\overrightarrow{AP} = f(\overrightarrow{L}, \overrightarrow{G}) \tag{5.3}$$

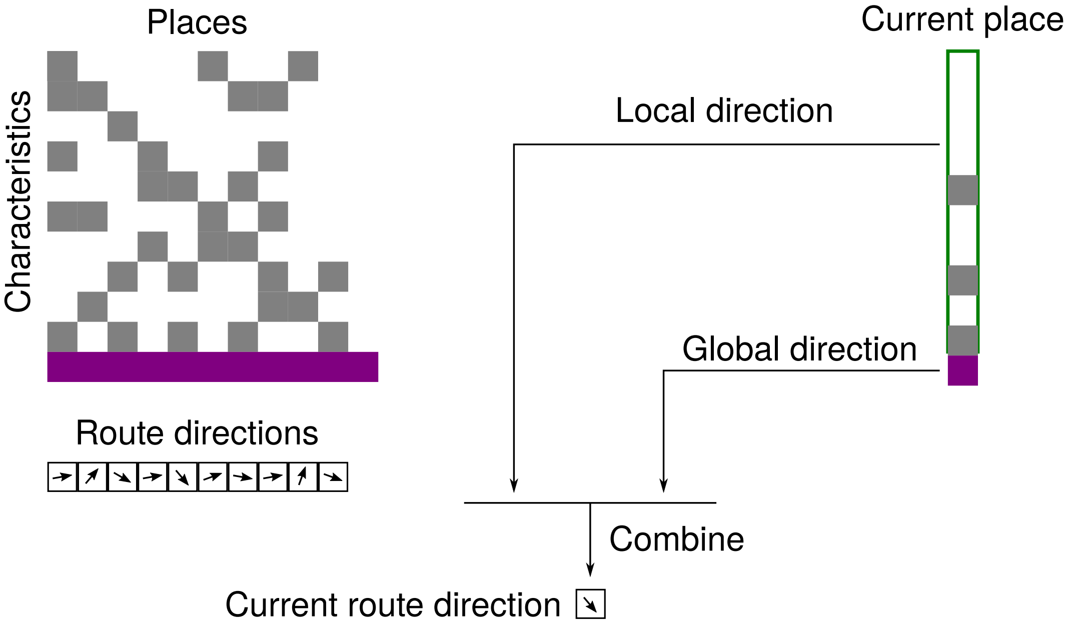


Figure 5.3: Global-local response schematic. The places along the route are associated with a route direction. Each place is described by a list of characteristics. The agent uses the characteristics of its current location to calculate a local direction and a global direction. The route-direction is a combination of the local and the global direction. The agent does not need to remember the characteristics of any place along the route, but only the one of the goal location.

One of the major problems of this strategy is that the agent cannot detect navigational errors quickly, because nothing was memorized along the route. The agent may have an estimate of the duration of its journey and, thus, realise that it has been travelling along the wrong path when it travelled longer than expected. The longer the expected duration of the journey is, the later the agent may realise it had made an error. Thus, this strategy should be used for relatively short routes to minimize the risk of staying lost for a long time.

5.2 Strategies used by insects, and outlook

The three classes of route-following algorithms can be sorted by complexity as follows: global-local response, familiarity response and place recognition-triggered response. To use the simplest route-following algorithm, only the goal direction needs to be known by the agent. Mechanisms that provide the global goal direction have been described for insects. When an insect travels from a feeding location to its nest, for instance, it can use the integration of its past motion (i.e. from the nest to the feeder) as the goal direction ("path integration"). This goal direction can then be combined with a local response, such as a collision-avoidance response with the objects surrounding the insect. Indeed, such responses may lead to the trajectory of many insects being centred between objects [142, 35, 76, 94, 8, 88], indicating that the insects prefer to keep some safety distance from objects. A global-local response strategy can, without doubt, be used by insects to follow routes [128]. As shown in the chapter entitled "Avoiding collision leads to common routes", this strategy can be used in various conditions to follow routes. In the following paragraphs, I will discuss how to, by experimental approaches, infer the strategy used by insects to follow a route.

The route-following strategy based on global-local responses does not involve the memorization of the scenery. Therefore, an insect displaced away from its route and searching for it may cross its route without noticing it. However, ants, when searching for their route, can recognize when they cross it. Thus, they either use a familiarity response or place recognition-triggered response to follow their route [79, 28].

The route-following strategy based on familiarity responses is possible only when the scenery around the route can be encoded as a whole in the insect's brain, i.e. the insect can recognize it is on the route, but cannot recognize specific places along the route. To reject the use of a familiarity response strategy, one would need to create an environment in which the route cannot be encoded as a whole in the insect's brain, i.e. the distortion function f does not exist. Exemplarily, the distortion function does not exist when the route contains two identical places associated with different route directions. Although the paths of ants have been reproduced in simulation by a route-following strategy based on familiarity response [6], the existence of the distortion functions has not been investigated systematically so far. Therefore, to date, it has not been shown that insects use a route-following strategy based on familiarity responses, but only that their behaviour in several contexts can be explained by this strategy [6].

The route-following strategy based on place-recognition is one of the most complex route-following algorithms, because every place along the route and the associated route-direction at the places need to be memorized by the agent. To show that the behaviour of an insect can be explained by this strategy, the two other – computationally much more parsimonious – route-following strategies first need to be ruled out. With the route-following strategy based on familiarity responses being, to date, a good candidate to explain the route-following behaviour of insects, it is not known if insects use place-recognition when following routes.

An insect may also combine different search strategies to follow a route. The global-local response strategy, for example, can be used between memorized places (chapter 4) and, thus, global-local and place-recognition responses are combined to follow the route. A hint toward the combination of strategies has been given by an experiment performed with ants [94]. The ant was placed on its habitual route, with its global direction, as determined from the integration of its past motion, pointing in the opposite direction to the route direction. The ant then travelled several centimetres along the global direction, but then reoriented itself to follow the route. The combination of route-following strategies in insects has, however, to date, never been assessed systematically.

5.3 Afterthought

Foraging insects have different connotations for most people. The ant invading our home is annoying and the wasp trying to catch a piece of our dinner is scary. On the other hand, the bumblebees buzzing in our garden are considered cute and fluffy flying insects; the honeybees, producers of our sweet honey, are regarded as useful. All these insects have a challenge to solve in common. They have to travel back and forth between their home and food locations. With a brain weighing only a few milligrams, they solve the problems of navigation in various situations where technical systems often fail.

Foraging insects are able to search efficiently for new locations rich in resources. During their search, they track their motion, and once a location has been found, they aim directly back home. Along their return trip, they remember the route. After having secured the resources in their home, they are ready to reach the same location again. On their way, moreover, they are able to avoid collisions with the numerous objects that are located on their paths. Efficient algorithms have been selected via the pressure of competition by evolution to solve all these problems. These algorithms are even working in man-made environments, such as cities. It is indeed not uncommon for a passionate observer to see bees flying in cities.

The navigation of foraging insects is one of the best-described complex behaviours in the animal kingdom. The abilities of bees to differentiate odours and to recognize patterns have been studied for decades. The seminal work of Santschi is over a century old [125].

Despite amazing works and discoveries, the strategy used by those insects is still not entirely understood.

This thesis contributes to the attempt to understand the strategies used by foraging insects to navigate. The thesis focused on how to follow routes and how to find a route again after being displaced from it. The abstract modelling of route-following and route-finding is interesting for biological research as a test bed for hypotheses on how an agent should move in the environment, on the one hand, and for engineers constructing autonomous robots as a source of inspiration, on the other hand. Additionally, the attempt in a general formalism of navigation (not only for insects) may be used as a basis for phrasing future relevant research questions.

Glossary

agent is a generic name for a biological or technical system which can move through a real or virtual world and make decisions. v–vii, 3–5, 9, 10, 17–25, 28, 30, 32, 33, 39, 43, 44, 46–53, 55–57, 59, 60, 63–72, 74–78, 80–82, 85–91, 94–96, 98–103, 105–109, 113, 114, 117–119, 121–125, 127–130, 132–145, 151, 152, 154–158, 160–162, 164, 165, 168, 169, 171, 175–177, 179–190, 192, 194

aiming is a spatial behaviour. The agent, away from its destination, detects characteristics of the destination and aims towards it. 17, 21, 28, 36, 38, 40, 188

allothetic means being centred in places. Allothetic cues are for example landmarks, the sun, or the moon. 19, 20, 40, 41

collision avoidance is a guidance strategy to prevent agent to collide with objects or other agents. v–vii, 4, 41, 43–48, 51, 53–59, 62–65, 67–70, 72, 73, 75, 77, 79–82, 84, 88, 95–97, 102, 106–109, 168, 179, 180

COMANV stands for Center of mass of average nearness vector. v, vii, 52, 54–59, 62, 88, 89, 100, 102, 103, 105

direction following is a spatial behaviour. The agent, away from its destination, moves in a direction with neither the need to recognise a location visited previously nor the detection of information at the destination. 17, 19, 26, 28, 40, 187

EMD stands for elementary motion detector. ii, iii, v–vii, 46–48, 54, 58–63, 65, 67, 69, 70, 73, 75–80, 86–88, 103, 106–109, 114

familiarity response is a strategy to follow a route. The agent follows the route by moving along in the most familiar direction. The agent does not need to recognise the place to determine the route direction. This is a special case of a state recognition-triggered response. ix, 181, 184–187, 190–192

global-local response is a strategy to follow a route. The agent follows the route by combining a global direction and a local direction. While the global direction is the direction straight to the goal, the local direction guarantees that the agent avoids danger, i.e. object collision or predators. This algorithm derives from direction following. ix, 181, 187–192

guidance is a spatial behaviour. The agent, away from its destination, is guided toward its destination by establishing a relationship between information currently available and the characteristics of the destination location remembered. 17, 22, 23, 28, 38, 40

idiothetic means being centred on oneself. Idiothetic cues include the proprioception, or the vestibular system. 19, 20

local navigation An agent using a local navigation strategy needs to recognise a single location, i.e. the goal. A navigation problem may be described as local if a relationship between the information at the goal location and the information available to the agent at any other location in the environment exists. 17, 22, 23, 35, 38

metric navigation is a spatial behaviour. At any location, the agent knows the distance and direction to any other location; the agent can plan any route within the range of the map. 18, 25, 26, 28, 38, 179

optic flow or optical flow is the pattern of apparent motion of objects, surfaces, and edges in a visual scene caused by the relative motion between an observer (an eye or a camera) and the scene. 43, 44, 46–53, 58–62, 66, 75–78, 80, 86, 87, 91–93, 95, 103

place recognition-triggered response is a strategy to follow a route. The agent has learned in which direction to move in order to follow the route by recognising places along the route. This is a special case of state recognition-triggered response. 181, 182, 190, 191

searching is a spatial behaviour. The agent, away from its destination, does not orient actively towards it; it reaches its destination by chance. iii, iv, 17–20, 23, 28, 117, 118, 121, 124, 127, 129, 132–134, 138, 139, 141, 142, 144, 155, 160, 179–181, 186, 191

state is the description of the situation an agent is currently placed in. The state of an agent in navigation include the agent's position, but also, for example, its velocity, and its level of energy. ii, 9, 15, 16, 23, 29, 30, 32, 57, 88, 114

state recognition-triggered response is a spatial behaviour. In a given state (e.g. at a given location), the agent recalls the direction to follow by recognising the state (the location). 17, 25, 28, 38, 182, 184, 188

topological navigation is a spatial behaviour. The agent, at every familiar location, knows the procedure to reach at least one other familiar location; the agent may plan its route by moving virtually through a sequence of familiar locations. 18, 24, 25, 28

Bibliography

1. Ardin, P., Peng, F., Mangan, M., Lagogiannis, K. & Webb, B. Using an Insect Mushroom Body Circuit to Encode Route Memory in Complex Natural Environments. *PLoS computational biology.* ISSN: 1553-7358. doi:10.1371/journal.pcbi.1004683 (2016).

2. Austin, D., Bowen, W. D. & Mcmillan, J. I. Intraspecific variation in movement patterns : modeling individual behaviour in a large marine predator. *Oikos.* doi:10.1111/j.0030-1299.1999.12730.x (2004).

3. Avarguès-Weber, A., Dyer, A. G., Ferrah, N. & Giurfa, M. The forest or the trees: preference for global over local image processing is reversed by prior experience in honeybees. *Proceedings of the Royal Society B: Biological Sciences.* ISSN: 1471-2954. doi:10.1098/rspb.2014.2384 (2015).

4. Avarguès-Weber, A. & Giurfa, M. Conceptual learning by miniature brains. *Proceedings of the Royal Society B: Biological Sciences.* doi:10.1098/rspb.2013.1907 (2013).

5. Babies, B., Lindemann, J. P., Egelhaaf, M. & Möller, R. Contrast-Independent Biologically Inspired Motion Detection. *Sensors.* ISSN: 1424-8220. doi:10.3390/s110303303 (2011).

6. Baddeley, B., Graham, P., Husbands, P. & Philippides, A. A Model of Ant Route Navigation Driven by Scene Familiarity. *PLoS Computational biology.* doi:10.1371/journal.pcbi.1002336 (2012).

7. Baird, E. & Dacke, M. Finding the gap: a brightness-based strategy for guidance in cluttered environments. *Proceedings of the Royal Society B: Biological Sciences.* ISSN: 1471-2954. doi:10.1098/rspb.2015.2988 (2016).

8. Baird, E. & Dacke, M. Visual flight control in naturalistic and artificial environments. *Journal of comparative physiology. A, Neuroethology, sensory, neural, and behavioral physiology.* ISSN: 1432-1351. doi:10.1007/s00359-012-0757-7 (2012).

9. Baird, E., Kornfeldt, T. & Dacke, M. Minimum viewing angle for visually guided ground speed control in bumblebees. *The Journal of experimental biology.* doi:10.1242/jeb.038802 (2010).

10. Basten, K. & Mallot, H. A. Simulated visual homing in desert ant natural environments: efficiency of skyline cues. *Biological cybernetics.* doi:10.1007/s00422-010-0375-9 (2010).

11. Bell, W. J. Searching Behavior Patterns in Insects. *Annual Review of Entomology.* ISSN: 0066-4170. doi:10.1146/annurev.en.35.010190.002311 (1990).

12. Bhagavatula, P. S., Claudianos, C., Ibbotson, M. R. & Srinivasan, M. V. Optic Flow Cues Guide Flight in Birds. *Current Biology.* ISSN: 09609822. doi:10.1016/j.cub.2011.09.009 (2011).

13. Błażejowski, B., Brett, C. E., Kin, A., Radwański, A. & Gruszczyński, M. Ancient animal migration: a case study of eyeless, dimorphic Devonian trilobites from Poland. *Palaeontology.* ISSN: 00310239. doi:10.1111/pala.12252 (2016).

14. Boeddeker, N., Dittmar, L., Stürzl, W. & Egelhaaf, M. The fine structure of honeybee head and body yaw movements in a homing task. *Proceedings of the Royal Society B: Biological Sciences.* ISSN: 1471-2954. doi:10.1098/rspb.2009.2326 (2010).

15. Borenstein, J. & Koren, Y. The vector field histogram-fast obstacle avoidance for mobile robots. *IEEE Transactions on robotics and Automation* (1991).

16. Borst, A. & Egelhaaf, M. Detecting visual motion: theory and models. *Visual motion and its role in the stabilization of gaze* (ed Miles, F. A.) (1993).

17. Borst, A., Reisenman, C. & Haag, J. Adaptation of response transients in fly motion vision. II: Model studies. *Vision research.* ISSN: 0042-6989 (2003).

18. Braun, E., Dittmar, L., Boeddeker, N. & Egelhaaf, M. Prototypical components of honeybee homing flight behavior depend on the visual appearance of objects surrounding the goal. *Frontiers in behavioral neuroscience.* ISSN: 1662-5153. doi:10.3389/fnbeh.2012.00001 (2012).

19. Braun, E., Geurten, B. & Egelhaaf, M. Identifying prototypical components in behaviour using clustering algorithms. *PLoS one.* doi:10.1371/journal.pone.0009361 (2010).

20. Brinkworth, R. S. A. & O'Carroll, D. C. Robust Models for Optic Flow Coding in Natural Scenes Inspired by Insect Biology. *PLoS Computational biology.* doi:10.1371/journal.pcbi.1000555 (2009).

21. Buatois, A. & Lihoreau, M. Evidence of trapline foraging in honeybees. *The Journal of experimental biology.* ISSN: 1477-9145. doi:10.1242/jeb.143214 (2016).

22. Buehlmann, C., Graham, P., Hansson, B. S. & Knaden, M. Desert Ants Locate Food by Combining High Sensitivity to Food Odors with Extensive Crosswind Runs. *Current Biology.* ISSN: 09609822. doi:10.1016/j.cub.2014.02.056 (2014).

23. Buehlmann, C., Hansson, B. S. & Knaden, M. Desert Ants Learn Vibration and Magnetic Landmarks. *PLoS one.* ISSN: 1932-6203. doi:10.1371/journal.pone.0033117 (2012).

24. Buehlmann, C., Hansson, B. S. & Knaden, M. Path integration controls nest-plume following in desert ants. *Current Biology.* ISSN: 1879-0445. doi:10.1016/j.cub.2012.02.029 (2012).

25. Cadena, C. *et al.* Past, Present, and Future of Simultaneous Localization And Mapping: Towards the Robust-Perception Age. arXiv: 1606.05830 (2016).

26. Cartwright, B. & Collett, T. Landmark maps for honeybees. *Biological Cybernetics.* ISSN: 0340-1200. doi:10.1007/BF00318718 (1987).

27. Chapman, R. F. (F., Simpson, S. J. & Douglas, A. E. (E. *The insects : structure and function* 5th. ISBN: 052111389X (Cambridge University Press, 2013).

28. Cheng, K., Narendra, A., Sommer, S. & Wehner, R. Traveling in clutter: Navigation in the Central Australian desert ant Melophorus bagoti. *Behavioural processes.* doi:10.1016/j.beproc.2008.10.015 (2009).

29. Cheung, A. *et al.* Still no convincing evidence for cognitive map use by honeybees. *Proceedings of the National Academy of Sciences of the United States of America.* ISSN: 1091-6490. doi:10.1073/pnas.1413581111 (2014).

30. Chiang, A.-S. *et al.* Three-Dimensional Reconstruction of Brain-wide Wiring Networks in Drosophila at Single-Cell Resolution. *Current Biology.* doi:10.1016/j.cub.2010.11.056 (2011).

31. Crall, J. D., Ravi, S., Mountcastle, A. M. & Combes, S. A. Bumblebee flight performance in cluttered environments: effects of obstacle orientation, body size and acceleration. *Journal of Experimental Biology.* ISSN: 0022-0949. doi:10.1242/jeb.121293 (2015).

32. Cresswell, J. E., Osborne, J. L. & Goulson, D. An economic model of the limits to foraging range in central place foragers with numerical solutions for bumblebees. *Ecological Entomology.* ISSN: 0307-6946. doi:10.1046/j.1365-2311.2000.00264.x (2000).

33. Cruse, H. & Wehner, R. No Need for a Cognitive Map: Decentralized Memory for Insect Navigation. *PLoS Computational Biology* (ed Friston, K.) ISSN: 1553-7358. doi:10.1371/journal.pcbi.1002009 (2011).

34. Dacke, M., Baird, E., Byrne, M., Scholtz, C. H. & Warrant, E. J. Dung Beetles Use the Milky Way for Orientation. *Current Biology.* ISSN: 09609822. doi:10.1016/j.cub.2012.12.034 (2013).

35. Dacke, M. & Srinivasan, M. V. Honeybee navigation: distance estimation in the third dimension. *Journal of Experimental Biology.* doi:10.1242/jeb.002089 (2007).

36. Diamond, J. Evolution, consequences and future of plant and animal domestication. *Nature.* doi:10.1038/NATURE01019 (2002).

37. Dijkstra, E. W. A note on two problems in connexion with graphs. *Numerische mathematik* (1959).

38. Dittmar, L., Stürzl, W., Baird, E., Boeddeker, N. & Egelhaaf, M. Goal seeking in honeybees: matching of optic flow snapshots? *The Journal of Experimental Biology.* ISSN: 0022-0949, 1477-9145. doi:10.1242/jeb.043737 (2010).

39. Dominic, C., Whitney, H., Sutton, G. & Robert, D. Detection and Learning of Floral Electric Fields by Bumblebees. *Science.* doi:10.1126/science.1230883 (2013).

40. Eckmeier, D. *et al.* Gaze Strategy in the Free Flying Zebra Finch (Taeniopygia guttata). *PLoS one.* doi:10.1371/journal.pone.0003956 (2008).

41. Edelstein-Keshet, L., Watmough, J. & Ermentrout, G. B. Trail following in ants: individual properties determine population behaviour. *Behavioral Ecology and Sociobiology.* doi:10.1007/BF00170717 (1995).

42. Egelhaaf, M. & Borst, A. Transient and steady-state response properties of movement detectors. *Journal of the Optical Society of America. A, Optics and image science.* ISSN: 0740-3232 (1989).

43. Egelhaaf, M., Boeddeker, N., Kern, R., Kurtz, R. & Lindemann, J. P. Spatial vision in insects is facilitated by shaping the dynamics of visual input through behavioral action. *Frontiers in Neural Circuits.* ISSN: 1662-5110. doi:10.3389/fncir.2012.00108 (2012).

44. El Jundi, B., Smolka, J., Baird, E., Byrne, M. J. & Dacke, M. Diurnal dung beetles use the intensity gradient and the polarization pattern of the sky for orientation. *The Journal of experimental biology.* ISSN: 1477-9145. doi:10.1242/jeb.101154 (2014).

45. El Jundi, B. *et al.* A Snapshot-Based Mechanism for Celestial Orientation. *Current Biology.* ISSN: 09609822. doi:10.1016/j.cub.2016.03.030 (2016).

46. Ellington, C. P., Machin, K. E. & Casey, T. M. Oxygen consumption of bumblebees in forward flight. *Nature.* ISSN: 0028-0836. doi:10.1038/347472a0 (1990).

47. Etienne, A. S. & Jeffery, K. J. Path integration in mammals. *Hippocampus.* ISSN: 1050-9631. doi:10.1002/hipo.10173 (2004).

48. Floreano, D., Ijspeert, A. J. & Schaal, S. Robotics and Neuroscience. *Current Biology.* ISSN: 09609822. doi:10.1016/j.cub.2014.07.058 (2014).

49. Foster, D. J. & Cartar, R. V. What causes wing wear in foraging bumble bees? *Journal of Experimental Biology.* doi:10.1242/jeb.051730 (2011).

50. Franz, M. O. & Mallot, H. A. Biomimetic robot navigation. *Robotics and Autonomous Systems.* ISSN: 0921-8890. doi:10.1016/S0921-8890(99)00069-X (2000).

51. Geurten, B. R., Kern, R., Braun, E. & Egelhaaf, M. A syntax of hoverfly flight prototypes. *The Journal of experimental biology.* doi:10.1242/jeb.036079 (2010).

52. Geva-Sagiv, M., Las, L., Yovel, Y. & Ulanovsky, N. Spatial cognition in bats and rats: from sensory acquisition to multiscale maps and navigation. *Nature Reviews Neuroscience.* ISSN: 1471-003X. doi:10.1038/nrn3888 (2015).

53. Gillies, M. T. & Wilkes, T. J. A comparison of the range of attraction of animal baits and of carbon dioxide for some West African mosquitoes. *Bulletin of Entomological Research.* ISSN: 0007-4853. doi:10.1017/S0007485300003412 (1969).

54. Goodman, J. E. & O'Rourke, J. *Handbook of discrete and computational geometry* ISBN: 9781584883012 (CRC press, 2010).

55. Gould, J. L. & Gould, C. G. *Nature's compass the mystery of animal navigation* ISBN: 9781400841660 (Princeton University Press, 2012).

56. Goulson, D. *Bumblebees: behaviour, ecology, and conservation* ISBN: 978-0199553075 (Oxford University Press, 2010).

57. Goulson, D. & Stout, J. C. Homing ability of the bumblebee Bombus terrestris (Hymenoptera: Apidae). *Apidologie.* ISSN: 0044-8435, 1297-9678. doi:10.1051/apido:2001115 (2001).

58. Graham, P. & Mangan, M. Insect navigation: do ants live in the now? *The Journal of experimental biology.* ISSN: 1477-9145. doi:10.1242/jeb.065409 (2015).

59. Grimm, J. *Hansel and Gretel* (ed Grimm, W.) (1812).

60. Guerrieri, F. J. & D'Ettorre, P. The mandible opening response: quantifying aggression elicited by chemical cues in ants. *The Journal of experimental biology.* ISSN: 0022-0949. doi:10.1242/jeb.008508 (2008).

61. Haferlach, T., Wessnitzer, J., Mangan, M. & Webb, B. Evolving a Neural Model of Insect Path Integration. *Adaptive Behavior.* ISSN: 1059-7123. doi:10.1177/1059712307082080 (2007).

62. Hafner, V. V. Adaptive Homing–Robotic Exploration Tours. *Adaptive Behavior.* ISSN: 1059-7123. doi:10.1177/10597123010093002 (2001).

63. Hagberg, A. A., Schult, D. A. & Swart, P. J. *Proceedings of the Python in Science Conference (SciPy): Exploring Network Structure, Dynamics, and Function using NetworkX* Pasadena, CA USA, 2008.

64. Hamming, R. W. Error detecting and error correcting codes. *Bell System technical journal* (1950).

65. Hartley, T., Maguire, E. A., Spiers, H. J. & Burgess, N. The Well-Worn Route and the Path Less Traveled. *Neuron.* ISSN: 08966273. doi:10.1016/S0896-6273(03)00095-3 (2003).

66. Hateren, J. v. & Schilstra, C. Blowfly flight and optic flow. II. Head movements during flight. *Journal of Experimental Biology* (1999).

67. Heinze, S. in (Springer Berlin Heidelberg, Berlin, Heidelberg, 2014). doi:10.1007/978-3-642-54718-8_4.

68. Hennig, P., Kern, R. & Egelhaaf, M. Binocular Integration of Visual Information: A Model Study on Naturalistic Optic Flow Processing. *Frontiers in Neural Circuits.* ISSN: 1662-5110. doi:10.3389/fncir.2011.00004 (2011).

69. Hoinville, T., Harischandra, N., Krause, A. F. & Dürr, V. in (Springer International Publishing, 2014). doi:10.1007/978-3-319-09435-9_11.

70. Humphries, N., Queiroz, N., Dyer, J. & Pade, N. Environmental context explains Lévy and Brownian movement patterns of marine predators. *Nature.* doi:10.1038/nature09116 (2010).

71. Humphries, N. E., Weimerskirch, H. & Sims, D. W. A new approach for objective identification of turns and steps in organism movement data relevant to random walk modelling. *Methods in Ecology and Evolution.* ISSN: 2041210X. doi:10.1111/2041-210X.12096 (2013).

72. Hyslop, A., Krapp, H. G. & Humbert, J. S. Control theoretic interpretation of directional motion preferences in optic flow processing interneurons. *Biological cybernetics.* doi:10.1007/s00422-010-0404-8 (2010).

73. James, A. C. Nonlinear operator network models of processing in the fly lamina. *Nonlinear Vision: Determination of Neural Receptive Fields, Function, and Networks. RB Pinter and B. Nabet, editors. CRC Press, Boca Raton* (1992).

74. James, A., Plank, M. J. & Edwards, A. m. Assessing levy walks as models of animal foraging. *J.R. Soc Interface.* doi:10.1098/rsif.2011.0200 (2011).

75. Juusola, M., Uusitalo, R. O. & Weckström, M. Transfer of graded potentials at the photoreceptor-interneuron synapse. *The Journal of General Physiology.* ISSN: 0022-1295, 1540-7748. doi:10.1085/jgp.105.1.117 (1995).

76. Kern, R., Boeddeker, N., Dittmar, L. & Egelhaaf, M. Blowfly flight characteristics are shaped by environmental features and controlled by optic flow information. *The Journal of experimental biology.* ISSN: 1477-9145. doi:10.1242/jeb.061713 (2012).

77. Knaden, M. & Graham, P. The Sensory Ecology of Ant Navigation: From Natural Environments to Neural Mechanisms. *Annual Review of Entomology.* ISSN: 0066-4170. doi:10.1146/annurev-ento-010715-023703 (2016).

78. Koenderink, J. J. & Doorn, A. J. Facts on optic flow. *Biological Cybernetics.* ISSN: 0340-1200. doi:10.1007/BF00365219 (1987).

79. Kohler, M. & Wehner, R. Idiosyncratic route-based memories in desert ants, Melophorus bagoti: how do they interact with path-integration vectors? *Neurobiology of learning and memory.* doi:10.1016/j.nlm.2004.05.011 (2005).

80. Koren, Y. & Borenstein, J. *Potential field methods and their inherent limitations for mobile robot navigation* in (IEEE, 1991). doi:10.1109/ROBOT.1991.131810.

81. Kraft, P., Evangelista, C., Dacke, M., Labhart, T. & Srinivasan, M. V. Honeybee navigation: following routes using polarized-light cues. *Philosophical Transactions of the Royal Society B: Biological Sciences.* doi:10.1098/rstb.2010.0203 (2011).

82. Lambrinos, D., Möller, R., Labhart, T., Pfeifer, R. & Wehner, R. A mobile robot employing insect strategies for navigation. *Robotics and Autonomous Systems.* ISSN: 0921-8890. doi:10.1016/S0921-8890(99)00064-0 (2000).

83. Lavalle, S. M. *Planning algorithms* ISBN: 0-521-86205-1 (2006).

84. Lent, D. D., Graham, P. & Collett, T. S. A Motor Component to the Memories of Habitual Foraging Routes in Wood Ants? *Current Biology.* ISSN: 0960-9822. doi:10.1016/j.cub.2008.11.060 (2009).

85. Leonhardt, S. D., Kaluza, B. F., Wallace, H. & Heard, T. A. Resources or landmarks: which factors drive homing success in Tetragonula carbonaria foraging in natural and disturbed landscapes? *Journal of Comparative Physiology A.* ISSN: 0340-7594. doi:10.1007/s00359-016-1100-5 (2016).

86. Levei, R. W. & Judd, T. Dead reckoning navigational system using accelerometer to measure foot impacts. *US Patent 5583776* (1996).

87. Lihoreau, M., Ings, T. C., Chittka, L. & Reynolds, A. M. Signatures of a globally optimal searching strategy in the three-dimensional foraging flights of bumblebees. ISSN: 2045-2322. doi:10.1038/srep30401 (2016).

88. Linander, N., Baird, E. & Dacke, M. Bumblebee flight performance in environments of different proximity. *Journal of comparative physiology. A, Neuroethology, sensory, neural, and behavioral physiology.* ISSN: 1432-1351. doi:10.1007/s00359-015-1055-y (2015).

89. Lindemann, J. P., Kern, R., van Hateren, J. H., Ritter, H. & Egelhaaf, M. On the computations analyzing natural optic flow: quantitative model analysis of the blowfly motion vision pathway. *The Journal of neuroscience.* doi:10.1523/JNEUROSCI.1132-05.2005 (2005).

90. Lindemann, J., Weiss, H., Möller, R. & Egelhaaf, M. Saccadic flight strategy facilitates collision avoidance: closed-loop per-

formance of a cyberfly. *Biological Cybernetics.* ISSN: 0340-1200. doi:10.1007/s00422-007-0205-x (2008).

91. Lucas, B. D. & Kanade, T. *An iterative image registration technique with an application to stereo vision* in (Morgan Kaufmann Publishers Inc., 1981).

92. Macmahon, M. & Stankiewicz, B. *Human and automated indoor route instruction following* in (28th Annual Conference of the Cognitive Science Society, 2006).

93. Mangan, M. *Visual homing in field crickets and desert ants: A comparative behavioural and modelling study* PhD thesis (University of Edinborgh, 2011).

94. Mangan, M. & Webb, B. Spontaneous formation of multiple routes in individual desert ants (Cataglyphis velox). *Behavioral Ecology* (2012).

95. Mårell, A., Ball, J. & Hofgaard, A. Foraging and movement paths of female reindeer: insights from fractal analysis, correlated random walks, and Lévy flights. *Canadian Journal of Zoology.* doi:10.1139/z02-061 (2002).

96. Martin, E. & Jens, P. L. Texture dependance of motion sensing and free flight behavior in blowflies. ISSN: 1662-5153. doi:10.3389/fnbeh.2012.00092 (2012).

97. Martin, N. & Franceschini, N. *Obstacle avoidance and speed control in a mobile vehicle equipped with a compound eye* in (IEEE, 1994). doi:10.1109/ROBOT.1990.126044.

98. Maurer, R. & Séguinot, V. What is modelling for? a critical review of the models of path integration. *Journal of Theoretical Biology.* ISSN: 00225193. doi:10.1006/jtbi.1995.0154 (1995).

99. Menini, A. *The Neurobiology of Olfaction* ISBN: 9781420071979 (CRC Press/Taylor & Francis, 2010).

100. Merkle, T., Knaden, M. & Wehner, R. Uncertainty about nest position influences systematic search strategies in desert ants. *The Journal of experimental biology.* ISSN: 0022-0949. doi:10.1242/jeb.02395 (2006).

101. Merlin, C., Heinze, S. & Reppert, S. M. Unraveling navigational strategies in migratory insects. *Current opinion in neurobiology.* ISSN: 1873-6882. doi:10.1016/j.conb.2011.11.009 (2012).

102. Meyer, H. G., Lindemann, J. P. & Egelhaaf, M. Pattern-Dependent Response Modulations in Motion-Sensitive Visual Interneurons—A Model Study. *PLoS one.* doi:10.1371/journal.pone.0021488 (2011).

103. Mirwan, H. B. & Kevan, P. G. Maze Navigation and Route Memorization by Worker Bumblebees (Bombus impatiens (Cresson) (Hymenoptera: Apidae). *Journal of Insect Behavior.* ISSN: 0892-7553. doi:10.1007/s10905-015-9507-3 (2015).

104. Möller, R. *A Biorobotics Approach to the Study of Insect Visual Homing Strategies* Habilitationsschrift, Wirtschaftswissenschaftliche Fakultät der Universität Zürich. 2002.

105. Montano, L. & Asensio, J. R. *Real-time robot navigation in unstructured environments using a 3D laser rangefinder* in (IEEE, 1997).

106. Moser, E. I., Kropff, E. & Moser, M.-B. Place cells, grid cells, and the brain's spatial representation system. *Annual review of neuroscience.* ISSN: 0147-006X. doi:10.1146/annurev.neuro.31.061307.090723 (2008).

107. Mronz, M. & Lehmann, F.-O. The free-flight response of Drosophila to motion of the visual environment. *Journal of Experimental Biology.* doi:10.1242/jeb.008268 (2008).

108. Muratet, L., Doncieux, S. & Meyer, J.-A. A biomimetic reactive navigation system using the optical flow for a rotary-wing uav in urban environment. *Proceedings of the International Session on Robotics* (2004).

109. Narendra, A., Gourmaud, S. & Zeil, J. Mapping the navigational knowledge of individually foraging ants, Myrmecia croslandi. *Proceedings of the Royal Society B: Biological Sciences.* ISSN: 1471-2954. doi:10.1098/rspb.2013.0683 (2013).

110. Oettler, J. *et al.* Fermat's principle of least time predicts refraction of ant trails at substrate borders. *PloS one.* ISSN: 1932-6203. doi:10.1371/journal.pone.0059739 (2013).

111. Osborne, J. L. *et al.* Bumblebee flight distances in relation to the forage landscape. *The Journal of animal ecology.* ISSN: 1365-2656. doi:10.1111/j.1365-2656.2007.01333.x (2008).

112. Pfeffer, S. E. & Wittlinger, M. Optic flow odometry operates independently of stride integration in carried ants. *Science (New York, N.Y.)* ISSN: 1095-9203. doi:10.1126/science.aaf9754 (2016).

113. Philippides, A., Steadman, N., Dewar, A., Walker, C. & Graham, P. in (Springer International Publishing, 2016). doi:10.1007/978-3-319-42417-0_24.

114. Pichon, J.-M., Blanes, C. & Franceschini, N. *Visual guidance of a mobile robot equipped with a network of selfmotion sensors* in (International Society for Optics and Photonics, 1990). doi:10.1117/12.969868.

115. Plett, J., Bahl, A., Buss, M., Kühnlenz, K. & Borst, A. Bio-inspired visual ego-rotation sensor for MAVs. *Biological cybernetics.* ISSN: 1432-0770. doi:10.1007/s00422-012-0478-6 (2012).

116. Ramos-Fernández, G., Mateos, J. & Miramontes, O. Lévy walk patterns in the foraging movements of spider monkeys (Ateles

geoffroyi). *Behavioral Ecology and.* doi:10.1007/s00265-003-0700-6 (2004).

117. Reichardt, W. Autokorrelationsauswertung als Funktionsprinzip des Zentralnervensystems. *Z. Naturforsch* (1957).

118. Reynolds, A. & Frye, M. Free-flight odor tracking in Drosophila is consistent with an optimal intermittent scale-free search. *PloS one.* doi:10.1371/journal.pone.0000354 (2007).

119. Reynolds, A., Reynolds, D. & Smith, A. Appetitive flight patterns of male Agrotis segetum moths over landscape scales. *Journal of Theoretical.* doi:10.1016/j.jtbi.2006.10.007 (2007).

120. Reynolds, A., Smith, A., Menzel, R. & Greggers, U. Displaced honey bees perform optimal scalefree search flights. *Ecology.* doi:10.1890/06-1916.1 (2007).

121. Riley, J. R. *et al.* Compensation for wind drift by bumble-bees. *Nature.* ISSN: 00280836. doi:10.1038/22029 (1999).

122. Robert, T., Frasnelli, E., Collett, T. S. & Hempel de Ibarra, N. *Male bumblebees learn where to drink but not how to get back home* in (Montevideo, Uruguay, 2016).

123. Ros, I. G. & Biewener, A. A. Optic flow stabilizes flight in ruby-throated hummingbirds. *The Journal of experimental biology.* ISSN: 1477-9145. doi:10.1242/jeb.128488 (2016).

124. Ryad, B., Charles, C., Xavier, L., Sio-Hoi, I. & Chiara, B. Event-Based Visual Flow. *IEEE Transactions on neural networks and learning systems.* doi:10.1109/TNNLS.2013.2273537 (2014).

125. Santschi, F. Comment s' orientent les fourmis. *Rev. Suisse Zool* (1913).

126. Schilstra, C. & Hateren, J. H. Blowfly flight and optic flow. I. Thorax kinematics and flight dynamics. *Journal of Experimental Biology.* ISSN: 0022-0949, 1477-9145 (1999).

127. Schilstra, C. & Van Hateren, J. H. Stabilizing gaze in flying blowflies. *Nature.* doi:10.1038/27114 (1998).

128. Schmidt, I., Collett, T. S., Dillier, F.-X. & Wehner, R. How desert ants cope with enforced detours on their way home. *Journal of Comparative Physiology A* (1992).

129. Schultheiss, P., Cheng, K. & Reynolds, A. M. Searching behavior in social Hymenoptera. *Learning and Motivation.* ISSN: 00239690. doi:10.1016/j.lmot.2014.11.002 (2015).

130. Schultheiss, P. *et al.* Similarities and differences in path integration and search in two species of desert ants inhabiting a visually rich and a visually barren habitat. *Behavioral Ecology and Sociobiology.* ISSN: 0340-5443. doi:10.1007/s00265-016-2140-0 (2016).

131. Schwegmann, A., Lindemann, J. P. & Egelhaaf, M. Depth information in natural environments derived from optic flow by insect motion detection system: a model analysis. *Frontiers in computational neuroscience.* doi:10.3389/fncom.2014.00083 (2014).

132. Seelig, J. D. & Jayaraman, V. Neural dynamics for landmark orientation and angular path integration. *Nature.* doi:10.1038/nature14446 (2015).

133. Serres, J. R. & Ruffier, F. Biomimetic Autopilot Based on Minimalistic Motion Vision for Navigating along Corridors Comprising U-shaped and S-shaped Turns. *Journal of Bionic Engineering* (2015).

134. Shoemaker, P. A., O'Carroll, D. C. & Straw, A. D. Velocity constancy and models for wide-field visual motion detection in insects. *Biological cybernetics.* doi:10.1007/s00422-005-0007-y (2005).

135. Sillar, K. T. (T., Picton, L. & Heitler, W. J. *The neuroethology of predation and escape* ISBN: 9780470972243 (2016).

136. Silvola, J. Respiration and energetics of the bumblebee Bombus terrestris queen. *Ecography.* ISSN: 0906-7590. doi:10.1111/j.1600-0587.1984.tb01119.x (1984).

137. Sims, D., Southall, E., Humphries, N. & Hays, G. Scaling laws of marine predator search behaviour. *Nature.* doi:10.1038/nature06518 (2008).

138. Smith, L., Philippides, A., Graham, P., Baddeley, B. & Husbands, P. Linked local navigation for visual route guidance. *Adaptive Behavior* (2007).

139. Smolka, J. *et al.* Night sky orientation with diurnal and nocturnal eyes: dim-light adaptations are critical when the moon is out of sight. *Animal Behaviour.* ISSN: 00033472. doi:10.1016/j.anbehav.2015.10.005 (2016).

140. Spaethe, J. Interindividual variation of eye optics and single object resolution in bumblebees. *Journal of Experimental Biology.* ISSN: 0022-0949. doi:10.1242/jeb.00570 (2003).

141. Spaethe, J. & Weidenmüller, A. Size variation and foraging rate in bumblebees (Bombus terrestris). *Insectes Sociaux.* ISSN: 0020-1812. doi:10.1007/s00040-002-8293-z (2002).

142. Srinivasan, M. V., Lehrer, M., Kirchner, W. H. & Zhang, S. W. Range perception through apparent image speed in freely flying honeybees. *Visual neuroscience.* doi:10.1017/S095252380000136X (1991).

143. Srinivasan, M. V., Zhang, S. W., Chahl, J. S., Stange, G. & Garratt, M. An overview of insect-inspired guidance for application in ground and airborne platforms. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering.* ISSN: 0954-4100, 2041-3025. doi:10.1243/0954410042794966 (2004).

144. Stelzer, R. J. & Chittka, L. Bumblebee foraging rhythms under the midnight sun measured with radiofrequency identification. *BMC Biology.* ISSN: 1741-7007. doi:10.1186/1741-7007-8-93 (2010).

145. Stephens, D. & Krebs, J. *Foraging theory* ISBN: 978-0691084428 (1986).

146. Stone, T., Differt, D., Milford, M. & Webb, B. *Skyline-based localisation for aggressively manoeuvring robots using UV sensors and spherical harmonics* in (IEEE, 2016). ISBN: 978-1-4673-8026-3. doi:10.1109/ICRA.2016.7487780.

147. Strübbe, S., Stürzl, W. & Egelhaaf, M. Insect-Inspired Self-Motion Estimation with Dense Flow Fields—An Adaptive Matched Filter Approach. *PLOS one.* ISSN: 1932-6203. doi:10.1371/journal.pone.0128413 (2015).

148. Stürzl, W. & Zeil, J. Depth, contrast and view-based homing in outdoor scenes. *Biological Cybernetics.* ISSN: 0340-1200, 1432-0770. doi:10.1007/s00422-007-0147-3 (2007).

149. Stürzl, W., Zeil, J., Boeddeker, N. & Hemmi, J. M. How Wasps Acquire and Use Views for Homing. *Current biology.* ISSN: 1879-0445. doi:10.1016/j.cub.2015.12.052 (2016).

150. Surmann, H., Lingemann, K., Nüchter, A. & Hertzberg, J. *A 3D laser range finder for autonomous mobile robots* in (2001).

151. Tammero, L. F. & Dickinson, M. H. Collision-avoidance and landing responses are mediated by separate pathways in the fruit fly, Drosophila melanogaster. *Journal of Experimental Biology.* ISSN: 0022-0949, 1477-9145 (2002).

152. Tammero, L. F. & Dickinson, M. H. The influence of visual landscape on the free flight behavior of the fruit fly Drosophila melanogaster. *Journal of Experimental Biology.* ISSN: 0022-0949, 1477-9145 (2002).

153. Tavoillot, P.-H. & Tavoillot, F. *L'abeille (et le) philosophe étonnant voyage dans la ruche des sages* (ed Jacob, O.) ISBN: 978-2-7381-3251-2 (2015).

154. Thorup, K. *et al.* Evidence for a navigational map stretching across the continental U.S. in a migratory songbird. *Proceedings of the National Academy of Sciences.* ISSN: 0027-8424. doi:10.1073/PNAS.0704734104 (2007).

155. Tinbergen, N. Über die Orientierung des Bienenwolfes. *Z. Vergl. Physiol.* ISSN: 0028-0836. doi:10.1038/18678 (1938).

156. Tolman, E. C. Cognitive maps in rats and men. *Psychological review.* ISSN: 0033-295X (1948).

157. Trullier, O., Wiener, S. I., Berthoz, A. & Mezer, J.-a. Biologically based artificial navigation systems: review and prospects. ISSN: 0301-0082. doi:10.1016/S0301-0082(96)00060-3 (1997).

158. Vance, J. T., Faruque, I. & Humbert, J. S. Kinematic strategies for mitigating gust perturbations in insects. *Bioinspiration & Biomimetics.* ISSN: 1748-3182. doi:10.1088/1748-3182/8/1/016004 (2013).

159. Vardy, A. & Moller, R. Biologically plausible visual homing methods based on optical flow techniques. *Connection Science.* ISSN: 0954-0091. doi:10.1080/09540090500140958 (2005).

160. Velthuis, H. H. & van Doorn, A. A century of advances in bumblebee domestication and the economic and environmental aspects of its commercialization for pollination. *Apidologie.* ISSN: 0044-8435. doi:10.1051/apido:2006019 (2006).

161. Vickerstaff, R. J. & Cheung, A. Which coordinate system for modelling path integration? *Journal of Theoretical Biology.* ISSN: 0022-5193. doi:10.1016/j.jtbi.2009.11.021 (2010).

162. Von. Frisch, K. *Dance Language and Orientation of Bees* ISBN: 9780674418776 (HUP, 1966).

163. Warzecha, A.-K. & Egelhaaf, M. Response latency of a motion-sensitive neuron in the fly visual system: dependence on stimulus parameters and physiological conditions. *Vision research* (2000).

164. Weber, K., Venkatesh, S. & Srinivasan, M. Insect-inspired robotic homing. *Adaptive Behavior* (1999).

165. Wehner, R. Desert ant navigation: how miniature brains solve complex tasks. *Journal of Comparative Physiology A.* doi:10.1007/s00359-003-0431-1 (2003).

166. Wehner, R. The architecture of the desert ant's navigational toolkit (Hymenoptera: Formicidae). *Myrmecol News* (2009).

167. Wikipedia. *Chain rule (probability) — Wikipedia, The Free Encyclopedia* [Online; accessed 31-August-2016]. 2015.

168. Wikipedia. *Orienteering — Wikipedia, The Free Encyclopedia* [Online; accessed 31-August-2016]. 2016.

169. Witthöft, W. Absolute Anzahl und Verteilung der Zellen im Hirn der Honigbiene. *Zeitschrift für Morphologie der Tiere* (1967).

170. Wittlinger, M., Wehner, R. & Wolf, H. The ant odometer: stepping on stilts and stumps. *Science.* doi:10.1126/science.1126912 (2006).

171. Wolf, H. Odometry and insect navigation. *The Journal of Experimental Biology.* ISSN: 0022-0949, 1477-9145. doi:10.1242/jeb.038570 (2011).

172. Wystrach, A. & Schwarz, S. Ants use a predictive mechanism to compensate for passive displacements by wind. *Current Biology.* ISSN: 1879-0445. doi:10.1016/j.cub.2013.10.072 (2013).

173. Wystrach, A., Mangan, M., Philippides, A. & Graham, P. Snapshots in ants? New interpretations of paradigmatic experi-

ments. *The Journal of Experimental Biology.* ISSN: 0022-0949, 1477-9145. doi:10.1242/jeb.082941 (2013).

174. Wystrach, A., Philippides, A., Aurejac, A., Cheng, K. & Graham, P. Visual scanning behaviours and their role in the navigation of the Australian desert ant Melophorus bagoti. *Journal of comparative physiology. A, Neuroethology, sensory, neural, and behavioral physiology.* ISSN: 1432-1351. doi:10.1007/s00359-014-0900-8 (2014).

175. Wystrach, A., Schwarz, S., Baniel, A. & Cheng, K. Backtracking behaviour in lost ants: an additional strategy in their navigational toolkit. *Proceedings of the Royal Society B: Biological Sciences.* doi:10.1098/rspb.2013.1677 (2013).

176. Wystrach, A., Schwarz, S., Schultheiss, P., Beugnon, G. & Cheng, K. Views, landmarks, and routes: how do desert ants negotiate an obstacle course? *Journal of comparative physiology. A.* ISSN: 1432-1351. doi:10.1007/s00359-010-0597-2 (2011).

177. Zeil, J. Orientation flights of solitary wasps (Cerceris; Sphecidae; Hymenoptera). *Journal of Comparative Physiology A: Neuroethology, Sensory, Neural, and Behavioral Physiology.* doi:10.1007/BF00189396 (1993).

178. Zeil, J., Hofmann, M. & Chahl, J. Catchment Areas of Panoramic Snapshots in Outdoor Scenes. *Journal Optical Society of America A* (2003).

179. Zeil, J. Visual homing: an insect perspective. *Current opinion in neurobiology.* ISSN: 1873-6882. doi:10.1016/j.conb.2011.12.008 (2012).

180. Zeil, J. *Visual homing in insects* in (Sapporo, Japan, 2014).

181. Zeil, J., Narendra, A. & Stürzl, W. Looking and homing: how displaced ants decide where to go. *Philosophical Transactions of the Royal Society B: Biological Sciences.* ISSN: 0962-8436, 1471-2970. doi:10.1098/rstb.2013.0034 (2014).