



HAL
open science

Human-Robot Motion: an Attention-Based Approach

Rémi Paulin

► **To cite this version:**

Rémi Paulin. Human-Robot Motion: an Attention-Based Approach. Robotics [cs.RO]. Université Grenoble Alpes, 2018. English. NNT: . tel-01864355

HAL Id: tel-01864355

<https://theses.hal.science/tel-01864355>

Submitted on 29 Aug 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ GRENOBLE ALPES

Spécialité : **Informatique**

Arrêté ministériel : 25 mai 2016

Présentée par

Rémi PAULIN

Thèse dirigée par

Patrick REIGNIER, Professeur des Universités, Grenoble INP
et **Thierry FRAICHARD**, Chargé de Recherche, Inria

préparée au sein du **Laboratoire d'Informatique de Grenoble**
dans l'**Ecole Doctorale Mathématiques, Sciences et Technologies de l'Information, Informatique**

Human-Robot Motion: an Attention-Based Approach

Thèse soutenue publiquement le 22 Mars 2018 ,
devant le jury composé de :

Mohamed Chetouani

Professeur des Universités, Université Pierre et Marie Curie, Président

Pierre De Loor

Professeur des Universités, Ecole Nationale d'Ingénieurs de Brest, Rapporteur

Yacine Amirat

Professeur des Universités, Université Paris Est Créteil, Rapporteur

Sylvie Pesty

Professeur des Universités, Université Grenoble Alpes, Examinatrice

Patrick Reignier

Professeur des Universités, Grenoble INP, Directeur de thèse

Thierry Fraichard

Chargé de Recherche, Inria, Directeur de thèse



Contents

Acknowledgments	ii
Abstract	iii
1 Introduction	1
2 Human-aware navigation: a literature review	3
3 Taking attention into account for human-aware navigation	10
3.1 Attention: a selection mechanism for information processing	10
3.2 Mechanisms and units of attention	11
3.2.1 Mechanisms of attention	11
3.2.2 Units of attention	12
3.3 Computational models of attention	13
3.3.1 Concepts	13
3.3.2 A literature review	15
3.3.3 Discussion	17
3.4 A new computational model of attention	18
3.4.1 Bottom-up saliency map	19
3.4.2 Top-down saliency map	19
3.4.3 Suppressive field	20
3.4.4 Attention map	21
3.4.5 Summary	22
3.5 Discussion	22
3.6 Attention field: a tool for human-aware navigation	28
3.7 Conclusion	31
4 Many-objective path planning: the next challenge of autonomous navigation	32
4.1 Introduction	32
4.2 Multi-objective evolutionary algorithms	33
4.2.1 Multi-objective concepts	33
4.2.2 Multi-objective evolutionary algorithms	35
4.2.3 Many-objective evolutionary algorithms	38
4.3 Multi-objective path planning: a literature review	39
4.3.1 Multi-objective algorithms for path planning	39
4.3.2 Path representation	40

Contents

4.3.3	Discussion	42
4.4	Approximation-guided evolutionary algorithm	42
4.5	Conclusion	43
5	An attention-based approach for human-aware navigation	45
5.1	An attention-based human-aware path planner	45
5.1.1	Objectives	46
5.1.2	Prior knowledge	50
5.1.3	Selecting an appropriate path	51
5.2	Application architecture for real-time many-objective path planning . . .	52
5.3	Simulation platform	57
5.4	Case studies	58
5.4.1	Human-aware motions: minimizing distraction	59
5.4.2	Human-aware motions for human-robot interaction	64
5.5	Conclusion	76
6	Conclusion	78

Acknowledgments

I would like to dedicate my PhD thesis to my family who have wholeheartedly supported me during my studies. I would like to thank my parents because without them I could not have progressed so far in my studies, my brother who is awesome and I would also like to thank Heather for brightening my everyday.

Without my supervisors, this research could not have been possible. And so I would like to say a big thank you to Prof. Patrick Reignier and Dr. Thierry Fraichard for all your time and support throughout my PhD.

Abstract

For autonomous mobile robots designed to share their environment with humans, path safety and efficiency are not the only aspects guiding their motion: they must follow social rules so as not to cause discomfort to surrounding people. Most socially-aware path planners rely heavily on the concept of *social spaces*; however, social spaces are hard to model and they are of limited use in the context of human-robot interaction where intrusion into social spaces is necessary. In this work, a new approach for socially-aware path planning is presented that performs well in complex environments as well as in the context of human-robot interaction. Specifically, the concept of attention is used to model how the influence of the environment as a whole affects how the robot's motion is perceived by people within close proximity. A new computational model of attention is presented that estimates how our attentional resources are shared amongst the salient elements in our environment. Based on this model, the novel concept of *attention field* is introduced and a path planner that relies on this field is developed in order to produce socially acceptable paths. To do so, a state-of-the-art many-objective optimization algorithm is successfully applied to the path planning problem. The capacities of the proposed approach are illustrated in several case studies where the robot is assigned different tasks. Firstly, when the task is to navigate in the environment without causing distraction our approach produces promising results even in complex situations. Secondly, when the task is to attract a person's attention in view of interacting with him or her, the motion planner is able to automatically choose a destination that best conveys its desire to interact whilst keeping the motion safe, efficient and socially acceptable.

1 Introduction

Robotics has been a growing research area for several decades now and we are starting to notice its appearance in our daily lives. Many of these systems need to be mobile - i.e. to navigate in their environment - in order to accomplish their task. In order to take full advantage of the usefulness of such systems they must do so with a high degree of autonomy. Autonomous navigation is a long studied topic and most approaches have traditionally been concerned with two main aspects: safety (which for the most part can be thought of as collision avoidance) and optimality in some sense (e.g. in terms of time of travel, traveled distance or energy consumed).

In the last decade, we have witnessed the growth of the service robotics sector (e.g. assistance for the elderly [1]) and a significant part of these mobile service robot technologies are moving towards everyday applications in households, offices and public places; these robots are designed to share our living space and to have varying degrees of interaction with humans; in the following, they will be referred to as *mobile robot companions*.

The coexistence in the same environment with humans adds new challenges for robot path planning: humans are social entities and motion in their vicinity must be planned accordingly. More specifically, by looking at our own behavior when navigating amongst other people we see that motion safety and efficiency are not the only aspects that guide our motion but that we additionally aim at respecting a large number of social rules so as not to cause discomfort to the people who share our environment. Likewise, mobile robot companions have to comply with these social rules if they want to successfully share their environment with humans. This has been confirmed by robot interaction experiments showing for instance that humans are sensitive to the spatial behavior of robots.

This leads to a redefinition of appropriate robot motion within a social context: while remaining safe and efficient, the motion must be deemed appropriate from a human point of view. The term *human-aware navigation* [2] or *human-robot motion* is commonly used to refer to the problem of obtaining appropriate motions for mobile robot companions. It is worth mentioning that in order to improve human-robot cohabitation, three main areas of interest can be distinguished: navigation, interaction and robot's appearance. This work will solely focus on the navigational aspects, i.e. designing socially appropriate trajectories.

Although mobile robots have actually shared the human living space as early as 1997 [3], it is only since around 2005 that this social dimension has started to be taken into account for robot navigation [4] by adding a social layer to the usual navigation architecture. A review of the literature shows that most of the approaches proposed so far rely upon the concept of *social spaces* [5], i.e. spatial regions of the environment to avoid in order to maintain surrounding people’s comfort by respecting their social expectations.

Such social spaces are primarily characterized using either the position of the person, e.g. personal space [6], or the activity he or she is currently engaged in, e.g. interaction space [7] and activity space [5]. Having defined the social spaces corresponding to the current situation, the robot’s motion is constrained to avoid as much as possible these spaces. The most common approach is to define costmaps on social spaces: the higher the cost, the less desirable it is for the robot to be at the corresponding position. The motion planner relies on the obtained costmaps in order to produce socially acceptable paths.

Such approaches are obviously highly relevant but social spaces suffer from several limitations: their size and shape is difficult to model for arbitrarily complex environments, they only depict the influence of the robot’s position on the comfort of surrounding persons, and they are of limited use in the context of human-robot interaction where intrusion into social spaces is often necessary.

In this work, we explore how the concept of attention can be taken into account to model how the influence of the environment as a whole affects how the robot’s motion is perceived by humans who share its environment. Following a preliminary investigation reported in [8], we introduce a new computational model of attention that estimates how the attentional resources of a given person are distributed among the persons and objects in his or her environment. Based on this model, we introduce the novel concept of *attention field* that can be viewed as an attention predictor and used to obtain socially appropriate motions in complex environments. We then build a path planner that relies on the concept of attention field in an attempt to optimize surrounding persons’ comfort during the motion whilst keeping the paths safe and short.

This thesis is divided into 6 chapters. Chapter 2 presents an in-depth review of the social robot navigation literature. In chapter 3, we introduce our new computational model of attention and the concept of attention field. A socially-aware path planner must simultaneously optimize the path safety, efficiency and comfort of surrounding people and we accordingly discuss the tools necessary for multi-objective path planning in chapter 4. In chapter 5, we build a socially-aware path planner that illustrates, using different case studies, how the attention field can be used to endow a mobile robot companion with socially appropriate motion capabilities in realistic and complex environments. Finally, in chapter 6, we summarize our work and suggest future research directions.

2 Human-aware navigation: a literature review

Mobile autonomous robots have shared the human living space as early as 1997 [3, 9–18]. Most of these prototype robots were deployed as part of large scale experiments in public environments. In such early experiments, the robot played a central role in the environment (e.g. museum tour guide) and expected surrounding persons to largely adapt to its behavior (e.g. follow); from a navigation point of view, humans were treated as objects.

Two challenges arise for robot navigation in general: *safety* and *efficiency*. Safety refers to the absence of collision with the obstacles in the environment whilst efficiency is usually measured in terms of time of travel, traveled distance or energy consumed to reach a goal. Thus all algorithms developed for robot motion planning aim to optimize a trade-off between safety and efficiency. Motion planning techniques can roughly be divided into global and local approaches: a global approach is primarily used for computing the entire path, then local low level navigation techniques are used to react to unforeseen obstacles and ensure the absence of collision. We will thereafter focus on global navigation techniques as they are the primary focus of human-aware navigation research and accordingly will be the focus of this work.

However, if mobile robots are to successfully share our living space, they will be expected to produce paths that are not only safe and efficient, but also deemed appropriate from a human point of view. These “social” robots must adapt their behavior to the humans sharing their environment, respecting social rules: their movement must be natural, predictable, not causing fear or surprise, not disturbing, etc. In other words, they must treat humans as social entities and not as objects. In order to produce such motions, new navigation techniques are required that take into account the effect of the robot’s presence upon nearby humans.

Around 2005, the social dimension of humans has started to be taken into account in robot navigation research and since then new motion planning techniques have been designed for human-aware navigation. Most of these socially-aware navigation strategies so far rely upon defining spatial regions of the environment that the robot should avoid in order to maintain surrounding people’s comfort by respecting their social expectations. Such regions are referred to in the literature as *social spaces*. As social spaces are not

strictly forbidden regions and may be entered depending on the situation [19, 20] (e.g. in the context of human-robot interaction), these spaces are defined as cost functions or potential fields [21–25] representing regions of the space to avoid (typically the higher the cost, the less desirable it is for the robot to be in the corresponding position): motion planners aim at finding trajectories that avoid these spaces as much as possible. Below we review the most popular types of social spaces encountered in the literature in the context of socially-aware robot path planning.

Personal space By far the most well-known and used social space in the context of socially-aware path planning is the *personal space*: it is an actively maintained region around each person into which others cannot intrude without causing discomfort (see figure 2.1).

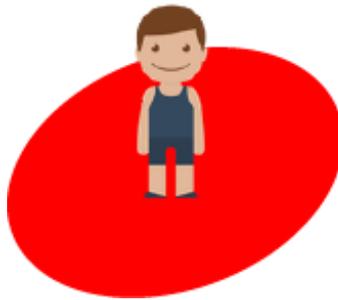


Figure 2.1: The personal space is a region around each person into which others cannot intrude without causing discomfort.

The concept of personal space is based on the studies of proxemics¹ first termed in 1966 by Hall [26] to describe the human management of space: it acknowledges that a person feels comfortable with another person up to a certain proximity within the space surrounding him or her. More specifically, four social spaces are defined: respectively the intimate, personal, social and public spaces; these spaces must be respected by other individuals and treated according to their relation with the person and the nature of the task they want to achieve. For example, two strangers usually don't come closer than the social distance, excepting circumstances such as in a crowded environment or handing over an object to one another; contrariwise, two close friends will regularly enter each others personal space. Robots are expected to use these rules of proxemics as a criteria to achieve socially acceptable behavior. Most studies on robot navigation in the proximity of humans now incorporate the concept of personal spaces, and although it has been the center of attention of the social robotic community for a while, modeling the personal space remains a challenge. Most studies introduce a slightly modified variant of the original model proposed by Hall [26] (represented using circles) in order to model additional social rules by taking into account not only the position of people

¹Proxemics is the study of spatial distances individuals maintain in various social and interpersonal situations.

but also other state variables such as their orientation or velocity. For instance, Kessler, Schroeter, and Gross [27] introduce an asymmetry in the personal space in order to bias the robot towards approaching the human from the side at an angle of about 45° (which is the preferred angle of approach according to Dautenhahn et al. [28]). Svenstrup, Bak, and Andersen [25], Svenstrup et al. [29], and Tranberg Hansen et al. [30] build a personal costmap that also favors approaching from the side and additionally prevents the robot from approaching the person too close from behind, considering that people prefer to be aware of moving objects in their environment (especially when close to them). Sisbot et al. [24] suggest modeling the personal space by a human-centered Gaussian whose width and height depends on the posture of the person (e.g. standing or sitting) in order to optimize the feeling of safety (e.g. when humans are standing, they tend to allow the robot to move closer), and also introduce a cost function with high values in regions situated behind the person. Scandolo and Fraichard [23] similarly augment the personal space at the back, and also enlarge the personal space of people along their direction of motion, in order to ensure that they have enough time to maneuver should something unexpected happen (the faster the human moves the larger the personal space at the front). Others studies such as Kirby, Simmons, and Forlizzi [22] adapt the personal space to include certain social behavioral preferences such as passing on the right, i.e. people tend to move to their right when making a move to avoid collisions; they do so by adding a region of increased cost to the right-hand side of people (they also enlarge the personal space at the front of the person). Finally, if the respect of proxemics seems to be essential when navigating around humans, the definition given by Hall [26] is not valid for humans in motion. When moving around other people, it is common to briefly enter other people’s personal spaces for efficiency reasons: we don’t want to make a large detour around each person encountered. To solve this issue, Luber et al. [31] suggest the use of a costmap (circular and centered on the individual) whose radius is dynamic and changes depending on the distance between robot and human during the course of the motion when they pass each other in order to generate more human-like behavior.

Interaction space When several persons are interacting (e.g. in a conversation), it is neither sufficient nor appropriate to consider the personal spaces of each person individually. Instead, people tend to maintain a larger social space around a group [32, 33]. Kendon [7] defines the o-space (orientation space) as the empty space surrounded by a group of people in interaction; he additionally defines the p-space (participants space) as a narrow band around the o-space where the people in interaction stand (see figure 2.2). Both of these spaces should be avoided by the robot, unless its task is to interact with the group or one of its members (it is common in this case to choose an end-pose for the robot inside the p-space). Rios-Martinez, Spalanzani, and Laugier [34] and Rios-Martinez et al. [35] present a simple way to estimate these spaces in the case of two agents interacting (based on their position and orientation), and successfully achieve motions that respect both the personal space and interaction space of surrounding people during navigation.

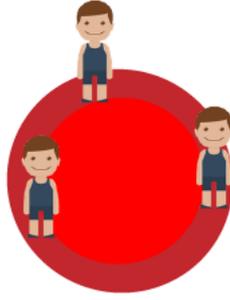


Figure 2.2: The interaction space for a group of people in interaction: the o-space (in red) and the p-space (in dark red).

Activity space Researchers very quickly started to take the issue of comfort beyond proximity criteria. When people are engaged in an activity, they define a corresponding space that should be avoided by others in order not to disrupt the activity and cause discomfort. In the context of socially-aware navigation, the corresponding social space is referred to as an *activity space* [5]. The size and shape of the activity space strongly depend on the corresponding activity. As an example Scandolo and Fraichard [23] define the activity space as the convex hull of the person and the objects he or she interacts with (see figure 2.3) and illustrate the use of such an activity space for socially-aware robot navigation on a scenario featuring a person watching television.

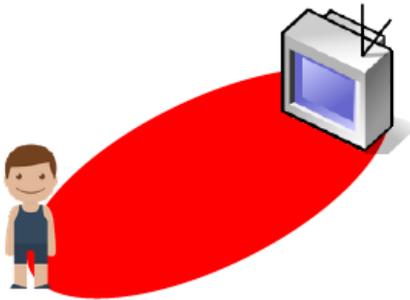


Figure 2.3: The activity space is typically defined as the convex hull of the person and the objects he or she interacts with.

Affordance space Closely related to the concept of activity space is the concept of *affordance space*: an affordance space is defined as a potential activity space [36], i.e. a social space that represents a potential activity provided by the environment [37]. The robot should avoid lingering in such regions because it could prevent an activity from taking place: for example the space in front of a bulletin board is an affordance space as it is likely at some point to become an activity space when looked at by people. An example of use and computation of affordance spaces in the context of socially-aware motion planning is found in Diego and Arras [38]: they define spatio-temporal affordance maps which contain probabilities that an activity takes place in the environment at a certain time; the robot then uses these affordance maps to plan its motion away from

regions which could cause potential disturbance. In a similar work, Sehestedt, Kodagoda, and Dissanayake [39] introduce a motion planner that uses a map of traffic density to minimize the probability of encounter between robot and humans.

Finally, people’s comfort is strongly influenced by the way they are aware of their surroundings. Some regions of their environment are inevitably more “perceptible” than others (e.g. a person will not see regions hidden behind obstacles, nor hear an acoustic signal originating near another more intense source of noise) which leads to the definition of *perceptive spaces* as regions that can be reached by an individual’s visual and auditory senses; if an event occurs outside the perceptive spaces, it will not be perceived by the individual. Similarly to social spaces, perceptive spaces can be used to help produce socially-aware motions: the invasion of perceptive spaces can cause distraction and people tend to place themselves at strategic positions in others’ perceptive spaces depending on what they want to achieve. For example, if wanting to interact, we position ourselves well inside the perceptive space of the other person. On the other hand, if we don’t want to disturb, a solution is to plan our movement outside the perceptive spaces. We expect robots to behave in a similar way. For instance a robot motion planner can use the fact that unperceived signals don’t influence the people’s comfort to navigate in the environment whilst causing the least amount of distraction to surrounding persons. Martinson [40] illustrates this by showing how to take advantage of the sources of noise located in the environment in order for a robot to cause the least amount of acoustic disturbance during its motion, i.e. using a noise map as a cost function. For humans in motion, Rios-Martinez et al. [35] similarly use the concept of *Information Process Space* as a region in front of the person to avoid. For each person, the information process space is the space within which all objects are considered as potential obstacles when a pedestrian is planning future trajectories and where psychological comfort is evaluated [41]. Finally in complex environments the robot sometimes has to move behind objects and therefore outside the field of view of the person; in order to ensure that the robot will not subsequently suddenly appear behind an object and therefore not cause surprise, Sisbot et al. [24] introduce an additional cost function (which they call “hidden zones”) which takes high values in regions of the space hidden visually from the person close behind obstacles, marking them as regions to avoid.

In conclusion, our review of the literature on human-aware navigation shows that during the last decade there has been a considerable amount of study done in this field and numerous techniques have been developed to produce paths that are not only safe and efficient, but also socially acceptable. In this review, only global motion planning techniques have been discussed: indeed local techniques for socially-aware motion planning deal mostly about human motion prediction or joint efforts in collision avoidance [2]. Most of the reviewed work focus on defining social spaces, i.e. regions to avoid in order to maintain surrounding people’s comfort by respecting their social expectations. These social spaces take into account the state of the persons in the environment (e.g. position, orientation, speed) as well as the interactions and activities they are engaged in.

The characterization of social spaces in arbitrarily complex environments however remains a challenge as the size and shape of these spaces strongly depend on the environment, the robot’s physical properties and many other context factors such as the people familiarity with the robot [42, 43]. For instance, to the best of our knowledge, the effect of the presence of obstacles on social spaces has not been much studied, e.g. are partly hidden regions behind an obstacle to be avoided?

Furthermore, social spaces only define spatial regions for the robot to avoid, and thus solely model how the robot’s position influences the comfort of surrounding people. It is clear however that not only the position of the robot but its entire state (e.g. orientation, speed, etc.) potentially carry social meaning and affect the people’s comfort in the course of its motion. As an example, a common approach is to consider separately the influence of the robot position (i.e. social spaces) and the robot’s velocity on the comfort of the people in the environment, and let a low level controller reconcile the global trajectory and the desired (maximum) speed. Whilst some studies have investigated how other characteristics of the robot such as its orientation or gaze direction [43, 44] affect the humans’ comfort, it is still unclear how to integrate this knowledge into the path planning process.

Social spaces appear to work best when the task of the robot doesn’t involve explicit interaction with humans, and thus the robot tries to minimize distraction. However, as social spaces define regions the robot should avoid, they are of limited use for end-pose selection in the context of human-robot interaction where intrusion in social spaces is necessary. Consequently, the motion planner must handle the constraints imposed by social spaces differently for end-pose selection. End-pose selection is therefore almost always decoupled from the path planning process and typically happens prior to path planning [2]. In complex environments this can lead to issues where the selected end-pose cannot be reached by a socially-acceptable path or even is not reachable: in these situations, a compromise between finding an appropriate end-pose for the task to achieve and optimizing people’s comfort during the motion should be sought.

Given these limitations of social spaces, we wish to investigate new tools for human-aware navigation which satisfy the following three requirements. Firstly, it must be able to seamlessly take into account relevant characteristics of the robot’s motion (e.g. orientation, speed, etc.) and deal with arbitrarily complex environment. Secondly our approach should be able to describe, depending on the task to achieve, both socially acceptable motions as well as appropriate end-pose selection that optimally conveys the robot’s intention to interact. Thirdly, our approach should reconcile end-pose selection and path planning in a single step, in order to yield an optimum compromise between appropriate end-pose and motion comfort.

Social spaces being mostly defined in an *ad hoc* manner so as to produce expected social behaviors, we try to model the underlying psychological mechanisms responsible for these social constraints. Such an approach is expected to solve the aforementioned limitations

of social spaces and to seamlessly handle arbitrary environments. In the next chapter, we accordingly introduce a model describing the full attentional state of the persons in the environment. Based on this model, we then introduce the novel concept of attention field and show how it can be used to yield socially-acceptable motions. In order to simultaneously take into account path safety, path efficiency, social constraints as well as appropriate end-pose selection, we discuss a multi-objective optimization algorithm in chapter 4 to be used by our path planner.

3 Taking attention into account for human-aware navigation

3.1 Attention: a selection mechanism for information processing

Given the limited ability of the brain to process the many inputs from our environment, it is not possible to perceive everything in real time. As an example, the human visual system confines high-resolution processing to the fovea and visual resolution rapidly decreases towards the periphery of the visual field in an attempt to restrict information input. Even so, the resulting flow of incoming information to our visual system is still far too extensive to be fully processed. In order to deal with this overwhelming excess of information, the visual system has to be endowed with a mechanism for selecting a small subset of input for more in depth processing [45], i.e. allowing only a small part of the information to reach awareness [46].

This mechanism, of selecting specific information to further process from all of the input, is called *attention*. Scholl [47] offers the following comprehensive description of attention: “intuitively, attention seems to be an extra processing capacity which can both intentionally and automatically select - and be effortfully sustained on - particular stimuli or activities”.

Attention describes a variety of complex selective processes in the nervous system. This makes attentional mechanisms difficult to model. In this chapter we will first review the fundamental mechanisms and units of attention and then look at some popular computational models of attention. We study the limitations of these models and develop a new computational model of attention. This new model predicts how attentional mechanisms select relevant elements (e.g. objects, features or locations) from the environment and distribute available attentional resources between them. We subsequently use this model to introduce the novel concept of *attention field* in the context of human-aware robot path planning.

It is noteworthy that although most literature focuses on attentional mechanisms in the context of visual attention, attentional mechanisms are not restricted to visual infor-

mation processing and are found in auditory signal processing for example. Therefore, although our model will be discussed in the context of visual information processing, it is expected to naturally extend to auditory information processing as well.

3.2 Mechanisms and units of attention

In this section we review the fundamental mechanisms and units of attention.

3.2.1 Mechanisms of attention

In most everyday life situations, visual attention control mechanisms can be split into two main categories [48]: *top-down* - also named endogenous - factors, such as current activity, prior knowledge of the surroundings and expectations, and *bottom-up* - also named exogenous - factors driven by sensory stimulation. In other words, visual attention selection is both influenced by decisions about allocation of attention derived by task demands and environmental cues. The interaction between these components controls where and to what we selectively devote visual attentional resources in our environment. Below we review bottom-up and top-down attentional mechanisms.

Bottom-up attentional mechanisms

We have all noticed that sometimes some objects or features stand out more so than others. There is indeed clear experimental evidence indicating that our attention can be captured by some stimulus in certain conditions [49]. In other words, a highly salient element often catches our attention (the *saliency* - or saliency - of an element is formally defined as a quantifying measure of its capacity to attract attention). This type of attentional capture is commonly referred to as *bottom-up* or exogenous attentional selection and is directly influenced by environmental cues, such as highly salient elements or sudden changes in the environment. Bottom-up attentional selection is a passive, reflexive, involuntary stimulus-driven mechanism. It is also independent of the nature of the particular task we are engaged in [46].

However it has been shown that stimuli do not always automatically capture attention owing to their saliency alone and our attentional state also critically depends on top-down mechanisms of attention.

Top-down attentional mechanisms

The task at hand greatly affects the way we allocate attentional resources. The more “focused” we are on a particular task or more generally a given region or direction of the environment, the less likely we are to notice elements or events irrelevant to the current task or located outside the region of interest. This endogenous mechanism of attentional selection is commonly referred to as *top-down*. It is considered to be active, voluntary and goal-directed, i.e. based on our current activity as well as prior knowledge of the surroundings and expectations [50]. In other words, top-down attentional selection reflects the intentional allocation of attentional resources to a predetermined element (e.g. objects [51, 52]) or region [53] of the environment. Endogenous attentional factors are still a much researched area as the exact mechanisms of top-down attentional selection have yet to be completely characterized.

3.2.2 Units of attention

So far we have talked about the mechanisms of visual attention whilst skipping over a fundamental question: what are the *units* of attention? In other words, what do we pay attention to? Objects, features, regions in space, directions, events? A fundamental task in the study of visual attention is to find the basic units with which allocation of attentional resources operates.

Whilst early studies have characterized attention in spatial terms, i.e. attention can restrict processing to certain areas of the visual field [54], an increasing number of studies suggest that in numerous cases the units of attentional selection are objects [55], with consideration that an object can be any “well-formed perceptually distinguishable surface” [56]. Indeed an early empirical illustration of the view that objects can serve as units of attention comes from Duncan [52]. This feature of attention has important consequences on resource processing: if an object is attended to, then all of its features will receive attention. In other words, irrelevant attributes of an attended object will be selected along with the relevant attributes, therefore “wasting” attentional resources [57] (although this effect breaks down at high attentional loads [47]). Object-based descriptions of attentional selection are however difficult in practice due to the broad and imprecise definition of objects: in most cases we are free to consider almost anything as an object, group multi-object units or split intra-object parts into separate units. Importantly, attention is not limited to a single object and can be shared between several objects [58, p. 183].

In conclusion, objects are a strong candidate to serve as units of attention. However, it seems apparent that all units of attention discussed in the literature - e.g. objects, features and locations - are also relevant units of attention that all coexist and should

not be treated as mutually exclusive. For instance, Egly, Driver, and Rafal [59] have shown the co-existence of both location and object-based attentional processes. Even events might also serve as units of attention [47].

3.3 Computational models of attention

As most of the underlying physiological mechanisms of attention are still under active research and only partially understood, modeling attentional mechanisms is difficult. This section presents an overview of existing computational models of attention and starts by introducing high level concepts commonly used in this context. As before, these concepts are discussed within the context of visual attention but they can also be directly applied to other modalities, e.g auditory.

3.3.1 Concepts

We have seen that attentional selection is the result of a combined interaction between bottom-up and top-down mechanisms. A review of the literature on computational models of attention reveals that bottom-up and top-down mechanisms are typically described using *saliency maps* over the visual space which are then combined to produce an *attention map* highlighting regions where attentional resources are most likely to be allocated.

Bottom-up saliency map

Bottom-up attentional selection is a stimulus-driven mechanism. The cumulated effect of all the salient elements - i.e. objects or features - in the environment is well described by a two-dimensional scalar field over the visual space, named *bottom-up saliency map* that encodes saliency for every location in the visual space. The idea of a saliency map comes from the work of Koch and Ullman [60] who described a biologically-plausible framework for discussing bottom-up attentional selection. It is worth noting that the saliency of an element seems to depend mostly on feature contrast rather than absolute feature strength and thus critically depends on its surrounding context [46].

Figure 3.1 illustrates the concept of bottom-up saliency map on a simple scenario featuring grayscale objects on a white background (figure 3.1a). At each location the value of the bottom-up saliency map (figure 3.1b) is a measure of the color contrast between the color of the object seen at this location and the background color (white).

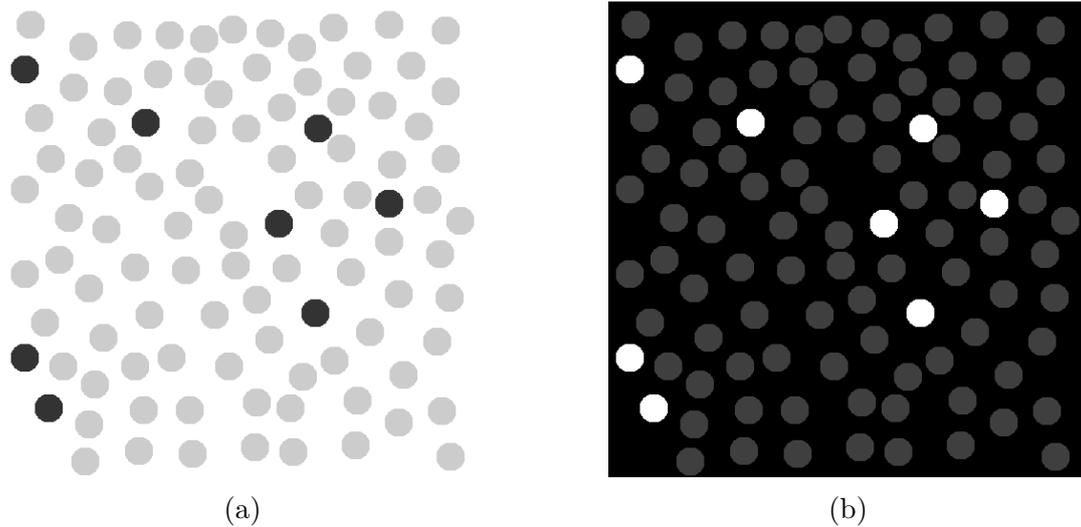


Figure 3.1: (a) Simple scenario featuring grayscale objects on a white background. (b) Bottom-up saliency map encoding for each location the color contrast with the background (the lighter the color the higher the value).

Top-down saliency map

Similarly to the bottom-up saliency map defined previously, a *top-down saliency map* - sometimes also referred to as *task-relevance map* - is a scalar field defined over the visual space in order to quantify the intentional allocation of attentional resources to predetermined objects, features or regions. It is based on cognitive factors such as current task, scene gist and context, short-term memory and expectation [50].

Figure 3.2 illustrates the concept of top-down saliency map. In this case, it simply highlights a central region in the image where the person expects relevant information to be found.

Attention map

Bottom-up and top-down mechanisms tightly interplay with regards to deciding which elements require further processing. Once both maps have been calculated, they are combined (the top-down saliency map acting as a mask or filter over the bottom-up saliency map) to produce an *attention map* or *priority map*. This attention map is a scalar field over the visual space that guides attention, i.e. features regions in the visual space that are most likely to be selected for in-depth processing. It is then commonly assumed that the focus of attention simply scans the attention map in order of decreasing saliency [46].

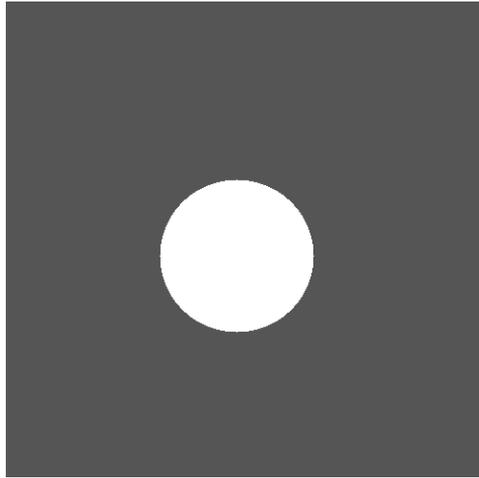


Figure 3.2: A simple example of a top-down saliency map highlighting a central region of the image expected to contain relevant information (the lighter the color the higher the value).

It is common to obtain the attention map by multiplying the bottom-up saliency map point-by-point by the top-down saliency map. As an example, combining the bottom-up saliency map shown in figure 3.1b and the top-down saliency map shown in figure 3.2 yields the attention map shown in figure 3.3. We see that the combined effects of the bottom-up and top-down maps enable to single out a promising object, both very salient (bottom-up) and located in the region of interest (top-down) for further analysis.

3.3.2 A literature review

In the previous section we discussed popular concepts used in attention modeling. Here we review some computational models of attention found in the literature.

Numerous studies do not take into account the influence of top-down mechanisms, and instead predict attention deployment based on bottom-up saliency maps only. Since top-down mechanisms are an essential component of attentional selection, in this review we focus on studies that model the competition between top-down and bottom-up attentional mechanisms. Several studies are reviewed below - the list is not exhaustive but to the best of our knowledge it is representative of most existing computational models of attention.

In 2006, Navalpakkam and Itti [61] presented a computational model of attention developed in the context of visual search that combines both bottom-up and top-down components. For a given image, bottom-up attentional selection is modeled by computing several low-level feature maps for a number of visual features (color, luminance and

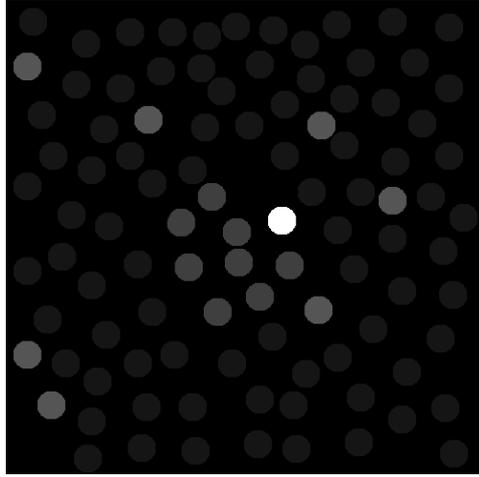


Figure 3.3: The resulting attention map is a combination of the bottom-up and top-down saliency maps. In this case a pointwise product between the two saliency maps shown in figure 3.1b and figure 3.2 is performed. As usual the lighter the color the higher the corresponding attention value.

orientation) at several spatial scales [62, 63]. These features maps are then combined (using a weighted sum) using top-down weights derived from learned statistical knowledge of the visual features of the target and distracting background. The combination of these feature maps yields an attention map that guides the focus of attention during the visual search task.

In 2007, Peters and Itti [64] introduced another computational model of attention in order to predict the eye movements of humans engaged in complex tasks (people playing video games). For a given image, this study similarly models bottom-up attentional selection using a bottom-up saliency map computed with the Itti-Kock saliency model [62, 63], i.e. obtained from low-level multi-scale visual features such as color contrast, luminance contrast, orientation, and motion. In contrast with the model by Navalpakkam and Itti [61], this study does not tune bottom-up saliency for any particular target but instead more generally uses learning techniques to compute top-down saliency maps that highlight task-relevant locations (irrespective of the content at these locations). It does this by associating particular types of scenes (identified by a low-level signature of the entire image) with corresponding top-down saliency maps (eye position prediction maps) using recorded gaze patterns. Bottom-up and top-down saliency maps are finally combined using a simple pointwise product to generate a predicted gaze density map.

The normalization model of attention from Reynolds and Heeger [65] attempts to model attentional selection at a lower level, i.e. how attention modulates neuronal activity in the visual cortex. In other words, for a given stimulus, the model outputs the corresponding neuronal responses. This model interestingly incorporates divisive normalization [66], a process that operates by dividing the response of each neuron by the “local

stimulus contrast” measured by the sum of responses across a population of surrounding neurons. As a result this model enhances responses to mild stimuli presented alone. In this model, each neuron is tuned for a different region in the visual field and feature set (e.g. orientation). In this sense, this model describes both spatial-based and feature-based attentional selection. The model also incorporates a top-down component which reshapes the distribution of activity across the population of neurons in favor of preferred stimuli. This model can therefore naturally handle concurrent arbitrary top-down goals, defined in spatial space or/and in feature space.

In the context of ambient applications and pervasive systems, Maisonnasse et al. [67] built a computational model of attention in order to detect interaction groups in intelligent environments. Compared to the previously described models, this model is purely object-based and aims to quantify how the attentional resources of each individual are distributed amongst the salient objects of their environment. Both top-down and bottom-up effects are taken into account. However this model suffers from computational instabilities and is centered on the idea that attentional selection is based on a single scalable spotlight¹ [53, 68] (or more accurately zoom-lens [69]) of attention which has been discounted by numerous studies [47], e.g. it does not model multimodal distributions of attention over the visual space.

3.3.3 Discussion

In this section we have presented an overview of existing computational models of attention. The normalization model of attention [65] stands out as it can simultaneously model spatial-based and feature-based attentional selection. It also has several important properties: for example it has the capacity to increase allocated resources to salient elements presented alone by incorporating the concept of divisive normalization. It is a low-level model that predicts the underlying neuronal responses but does not, however, explicitly predict the distribution of attentional resources. The computational model developed by Maisonnasse et al. [67] is the only model that explicitly quantifies how attentional resources are shared between the salient objects of the environment - but it suffers from instabilities and is unable to handle a multimodal distribution of attention. We therefore attempt to build a new high-level computational model of attention inspired from the model of Reynolds and Heeger [65] that relies on the well-proven concept on saliency maps [61, 64].

¹In the spotlight model of attention, attentional resources are focused on one single specific area of the visual field.

3.4 A new computational model of attention

In this section, we attempt to build a high-level model of selective attention that takes into account both top-down and bottom-up components in order to predict how attentional resources are shared between the salient elements (e.g. objects, features or locations) in our environment. We describe our model in the context of visual information processing. Based on the literature review, we aim to satisfy the following requirements for our model:

- Attention is a multimodal resource usually shared between several elements in our environment [47].
- The model must be able to seamlessly describe object-based, feature-based and spatial-based attention (see 3.2.2).
- The model must be able to handle concurrent arbitrary top-down goals, which cannot be described by a single spotlight of top-down attention [47].
- Inspired from the normalization model of neuronal responses of Reynolds and Heeger [65], the model must have the capacity to increase allocated resources to salient elements presented alone. We believe that this property of the model is necessary to explain many experimental results in the context of visual search tasks (next section details such results and illustrates the capacities of our model to predict them).

We describe the top-down and bottom-up components of attention using the widely adopted concepts of *saliency maps* (see 3.3.1). In order for our model to be able to describe both feature-based and spatial-based attention, these saliency maps are defined over the Cartesian product of the visual space and the feature space. A saliency map \mathcal{S} can be written as:

$$\mathcal{S} : \mathcal{VF} \rightarrow \mathfrak{R} \quad (3.1)$$

with

$$\mathcal{VF} = \mathcal{V} \times \mathcal{F} \quad (3.2)$$

\mathcal{V} is the two-dimensional visual space, where each vector $\mathbf{v} \in \mathcal{V}$ represents a possible gaze direction. \mathcal{F} is the feature space which can be of arbitrary dimension, where each vector $\boldsymbol{\psi} \in \mathcal{F}$ quantifies relevant visual features about the perceived elements such as

color, luminance, orientation, motion, etc. (these features are defined in a viewer-based reference frame).

An *attention map* is computed from the bottom-up and top-down saliency maps, from which we derive how attentional resources are shared between the salient elements of the environment.

3.4.1 Bottom-up saliency map

The bottom-up component of attentional selection is modeled by a unique bottom-up saliency map $BU : \mathcal{VF} \rightarrow \mathfrak{R}$ that highlights salient regions of the visual and feature space. This bottom-up saliency map is derived from the visual composition of the environment as well as the visual sensitivity variation over the visual field. Our model however does not specify how these salience values are obtained.

We illustrate this concept on the simple scenario introduced in section 3.3.1 and illustrated in figure 3.1a, featuring grayscale objects on a white background. For the sake of illustration, the feature space is reduced to a single property, using grayscale color as an example feature. A position in the visual field $\mathbf{v} = (x, y) \in \mathcal{V}$ is represented relative to the person’s gaze direction by its horizontal angular displacement $x \in [-90^\circ, 90^\circ]$ and vertical angular displacement $y \in [-90^\circ, 90^\circ]$. A grayscale color $\psi \in \mathcal{F}$ is represented by a corresponding real number ranging between 0 (white) and 1 (black).

We arbitrarily set the salience of an object to be equal to its color contrast value (relative to the background color) and we do not model the variation of visual sensitivity over the visual field. Thus for a given position in the visual field $\mathbf{v} \in \mathcal{V}$ and color $\psi \in \mathcal{F}$ the corresponding bottom-up saliency map value can be simply defined as (ψ_0 is the color of the background):

$$BU(\mathbf{v}, \psi) = \begin{cases} |\psi - \psi_0| & \text{if an object of color } \psi \text{ is seen at position } \mathbf{v} \\ 0 & \text{otherwise} \end{cases} \quad (3.3)$$

3.4.2 Top-down saliency map

Similarly to the bottom-up saliency map, a top-down saliency map $TD : \mathcal{VF} \rightarrow \mathfrak{R}$ is specified in terms of its spatial and featural extents. Typically, the top-down saliency map is derived from the current activity and is specified in terms of its gain for each location and feature combination, i.e. it is 1 everywhere except for small regions of interest in spatial and feature space where its value is greater than 1. Such a definition

of the top-down saliency map makes our model very flexible and able to describe a wide variety of endogenous attentional strategies. For instance, spatial top-down attention is characterized by a top-down saliency map narrow in the visual space but broad in the feature space. Conversely, non-spatial feature-based or object-based top-down attention is represented by a top-down saliency map narrow in the feature space and broad in the visual space. In both cases, the top-down saliency map can be multimodal if attending to multiple elements simultaneously [70, 71].

Once both the bottom-up and top-down saliency maps have been obtained, the bottom-up saliency map is multiplied point-by-point by the top-down saliency map, effectively modulating the bottom-up saliency map in a manner equivalent to an increase of bottom-up saliency in attended regions. We obtain an intermediary saliency map S with increased contrast between attended and non-attended regions:

$$S(\mathbf{v}, \psi) = BU(\mathbf{v}, \psi) \times TD(\mathbf{v}, \psi) \quad (3.4)$$

Once the bottom-up and top-down saliency maps have been merged, a normalization step is then applied.

3.4.3 Suppressive field

The presence of nearby distractors can suppress the attentional response to a salient element. In other words, local saliency depends on long range contextual influences. Inspired from Reynolds and Heeger [65], we introduce the *suppressive field* $s : \mathcal{VF} \rightarrow \mathfrak{R}$ in order to model this effect. The suppressive field describes the surrounding regions, both in visual space and feature space (i.e. the range of positions and features) that can contribute - if distractors are present - to the suppression of the salience of a given element.

As an example, the suppressive field can be modeled by a Gaussian field, i.e. for a given position in the visual field $\mathbf{v} = (x, y) \in \mathcal{V}$ and color $\psi \in \mathcal{F}$:

$$s(\mathbf{v}, \psi) = \frac{1}{(2\pi)^{\frac{3}{2}} \sigma_x \sigma_y \sigma_\psi} e^{-\left(\frac{x^2}{2\sigma_x^2} + \frac{y^2}{2\sigma_y^2} + \frac{\psi^2}{2\sigma_\psi^2}\right)} \quad (3.5)$$

Whereas the bottom-up saliency map is assumed to be selective for feature and location, the suppressive field is assumed to be largely non-specific [65], i.e. a salient element broadly affects other nearby elements (the values of σ are “large” compared to the typical variations of the corresponding quantities observed in the scenario).

The suppressive field is non-negative and it is normalized (its integral is arbitrarily set to 1) for computational simplicity.

3.4.4 Attention map

In order to yield the attention map, our model requires a bottom-up and a top-down saliency map as well as a suppression field as mentioned above. Using the suppressive field, each salience value of the previously obtained intermediary saliency map S is normalized with respect to its surrounding context - i.e. dividing each salience value by a weighed average of saliences pooled over a nearby range of positions and features - in order to yield the attention map A :

$$A(\mathbf{v}, \psi) = \frac{S(\mathbf{v}, \psi)}{\sigma + (s \otimes S)(\mathbf{v}, \psi)} = \frac{(TD \times BU)(\mathbf{v}, \psi)}{\sigma + (s \otimes (TD \times BU))(\mathbf{v}, \psi)} \quad (3.6)$$

\otimes is the convolution operator. The constant σ is a small strictly positive value and is only added for numerical stability reasons.

Instead of searching for a single focus of attention at a time - typically done through a winner-take-all process on the attention map (i.e. at a given time our attentional focus is directed to the peak value in the attention map), we use the attention map to calculate the distribution of attentional resources allocated to each salient element in the environment. Each salient element of interest e_i in the environment (e.g. object) is characterized by a subset of \mathcal{VF} (i.e. a range of positions and features) $E_i \subset \mathcal{VF}$. As an example, considering our simple scenario from section 3.3.1, for each object e_i :

$$E_i = \{(\mathbf{v}, \psi) \in \mathcal{VF} \mid \text{a part of object } e_i \text{ of color } \psi \text{ is seen at position } \mathbf{v}\}$$

The proportion of attentional resources r_{e_i} allocated to element e_i is taken to be the normalized sum of salience values in E_i :

$$r_{e_i} = \frac{\sum_{(\mathbf{v}, \psi) \in E_i} A(\mathbf{v}, \psi)}{\sum_{(\mathbf{v}, \psi) \in \mathcal{VF}} A(\mathbf{v}, \psi)} \quad (3.7)$$

3.4.5 Summary

In summary, we developed a new computational model of attention in order to predict how a person’s attentional resources are distributed between the salient elements of his or her environment. Our model uses the popular concepts of bottom-up and top-down saliency maps. It can simultaneously model object-based, feature-based and spatial-based attentional selection and it is naturally able to handle multimodal distributions of attention. Our model however does not explicitly precise how the bottom-up and top-down saliency maps are obtained as we believe that this process is highly situation dependent.

3.5 Discussion

In this chapter, we have developed a new computational model of attention that predicts how our attentional resources are shared between salient elements of our environment. An important property of our model is its capacity to increase allocated resources to salient elements presented alone. We illustrate this property on two simple experiments and show that our model can predict experimental results found in the literature in the context of visual search tasks.

In order to compare the output of our model - i.e. the amount of attentional resources allocated to each salient element in the environment - and the experimental results obtained in the context of visual search tasks - i.e. the reaction time² - we make the assumption that these two quantities are tightly related. In other words, we expect that the more attentional resources allocated by our model to a target element the faster it will be found in a visual search task. This assumption is coherent with the fact that the focus of attention is commonly assumed to scan the attention map in order of decreasing saliency [46].

We consider a visual search task where a person is tasked with looking for a single target that stands out from the others (the distractors). He or she has no expectations towards the properties of the target, i.e. there is no top-down influence. We do not model the variation of visual sensitivity over the visual field. The scenarios considered here are similar to the scenario illustrated in figure 3.1a. The two experiments feature circular grayscale objects, identical in size, on a uniform grayscale background. A fixed number of these objects is randomly scattered in the visual field. For the visual search task the target therefore differs from the distractors by its color only. The feature space is reduced to a single property using grayscale color as an example feature and we arbitrarily set

²In the context of visual search tasks, the reaction time is the time taken to find the target or the time required to determine if the target is present or absent.

the salience of an object to be equal to its color contrast value with the background. The bottom-up saliency map is defined in equation 3.3.

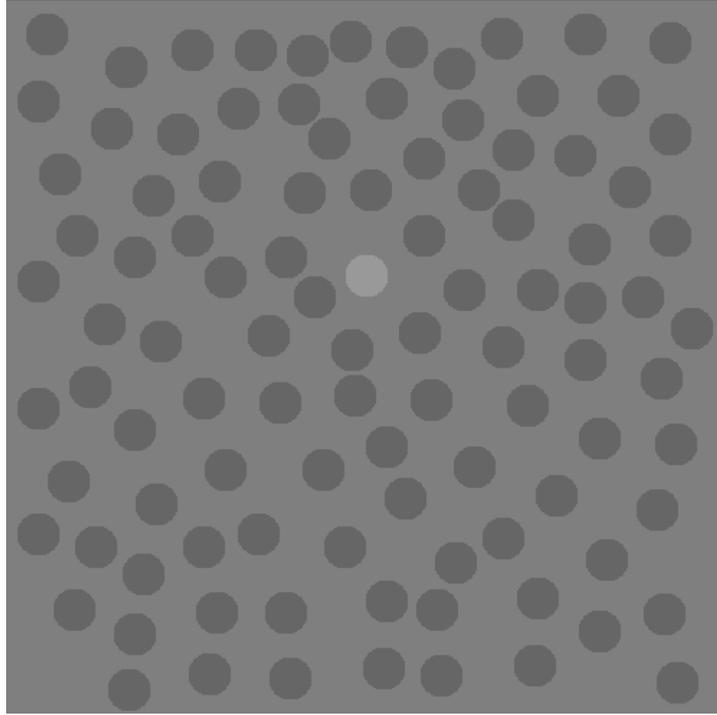
The suppressive field is modeled by a 3-dimensional Gaussian field as detailed in equation 3.5. We choose the values for σ_x and σ_y larger than the average angular distance between two nearest objects in order to reduce the influence of the objects distribution on the obtained results. Each object is seen under an angular distance of $\sim 10^\circ$, and 101 objects are spread uniformly in the visual field without touching each other. Accordingly we set $\sigma_x = \sigma_y = 20^\circ$. Each object has a grayscale color value ψ in the range $[0, 1]$, and we arbitrarily set $\sigma_\psi = 0.2$.

In our first experiment, we study the influence of the similarity between the target and the distractors on attentional resources allocation. Two scenarios are considered in which the background has a color $\psi_0 = 0.5$ and the target has a color $\psi = 0.4$ (respectively $\psi = 0.3$) and the distractors have a color $\psi = 0.6$ (respectively $\psi = 0.7$). The target and the distractors have the same color contrast relative to the background and thus have the same bottom-up salience. The scenarios are illustrated in figure 3.4. The attention maps obtained by our model for the two scenarios are represented in figure 3.5 and figure 3.6 displays the amount of attentional resources allocated to each objects. The obtained results clearly show that the presence of the distractors affect the proportion of resources allocated to the target object. More precisely, the target is allocated more attentional resources than the surrounding distractors, and the more similar the distractors are to the target object, the less attentional resources are allocated to the target. These results agree with the experimental results obtained in the context of visual search task, i.e. the search is fast when the target “pops out”³ [72] and slow when it is more similar to the distractors [73–77].

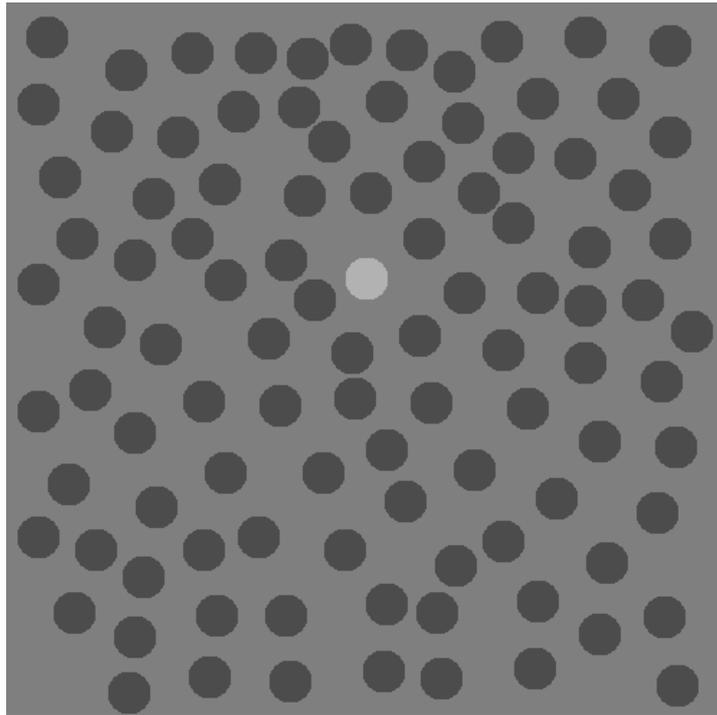
In our second experiment, we show that our model makes the target object stand out (i.e. it is allocated more attentional resources than the surrounding distractors) even if it has a lower bottom-up salience (i.e. color contrast) than the distractors. In this scenario, a target with a color $\psi = 0.5$ is present amongst distractors with a color $\psi = 1$ on a white background ($\psi_0 = 0$). The scenario is illustrated in figure 3.7 and the results are shown in figure 3.8 and figure 3.9.

In summary, we have illustrated some qualitative properties of our model and we have shown that it is able to describe well-known experimental results observed in the context of visual search. Naturally, these results do not establish a proper validation of our model, and further research work is needed in this direction.

³The “pop out” effect occurs when the target has a unique feature that makes it stand out from the distractors.

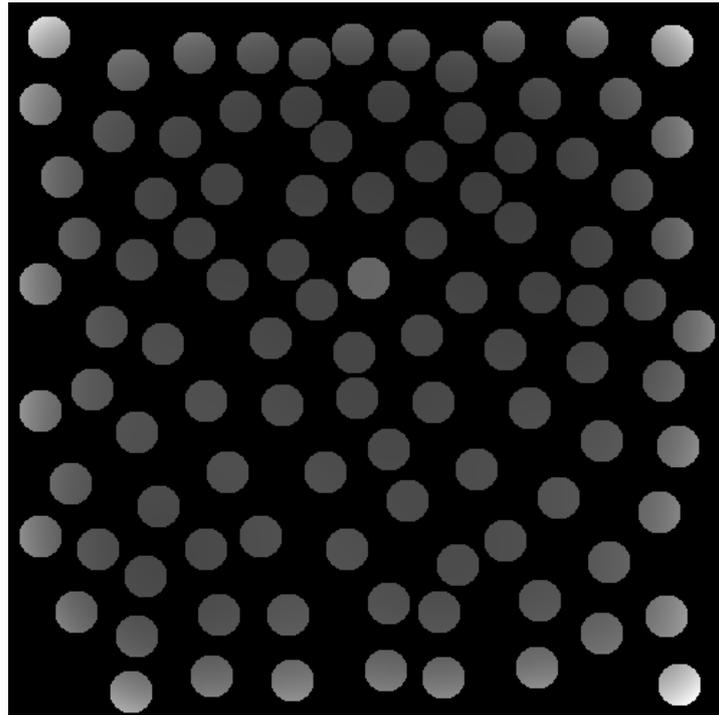


(a)

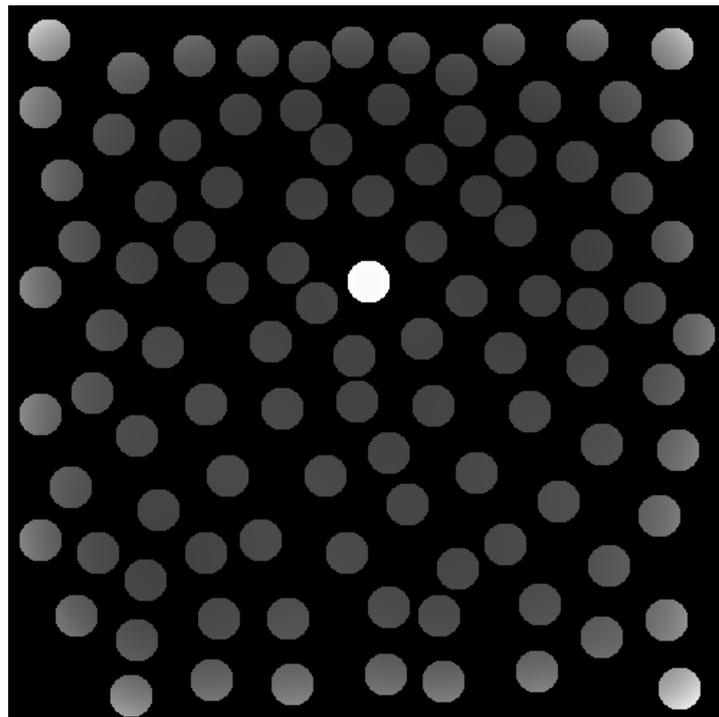


(b)

Figure 3.4: Experiment 1: Background color $\psi_0 = 0.5$. (a) Target color $\psi = 0.4$ and distractors color $\psi = 0.6$. (b) Target color $\psi = 0.3$ and distractors color $\psi = 0.7$.



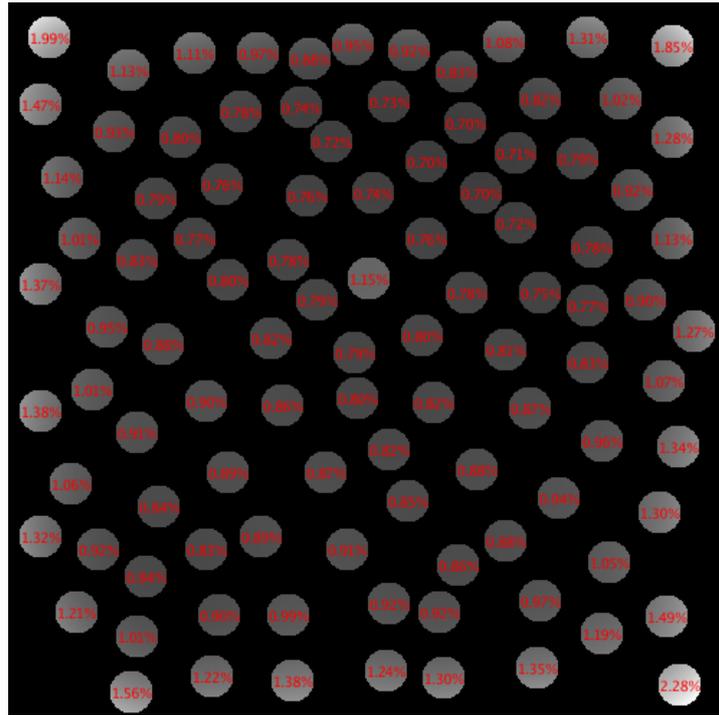
(a)



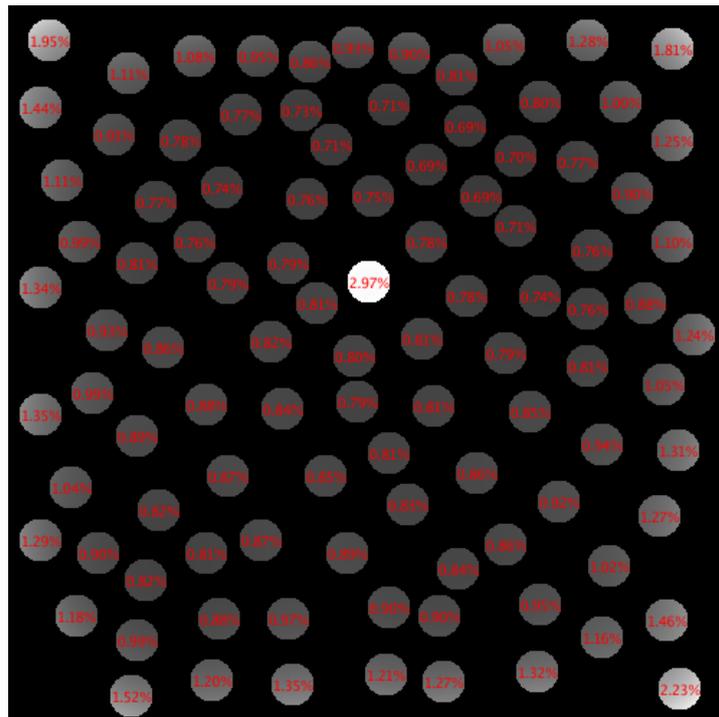
(b)

Figure 3.5: Experiment 1: Attention maps.

3 Taking attention into account for human-aware navigation



(a)



(b)

Figure 3.6: Experiment 1: Distribution of attentional resources.

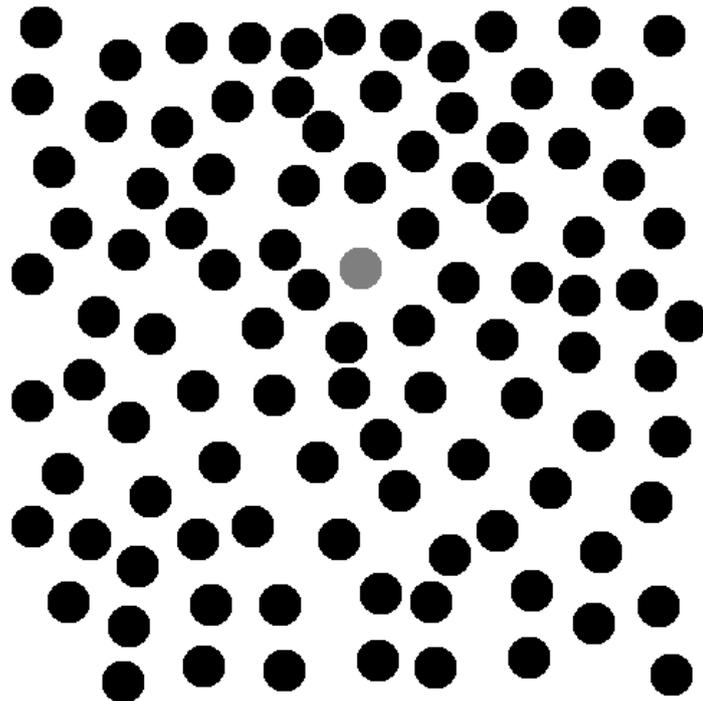


Figure 3.7: Experiment 2: Background color $\psi_0 = 0$, target color $\psi = 0.5$ and distractors color $\psi = 1$.

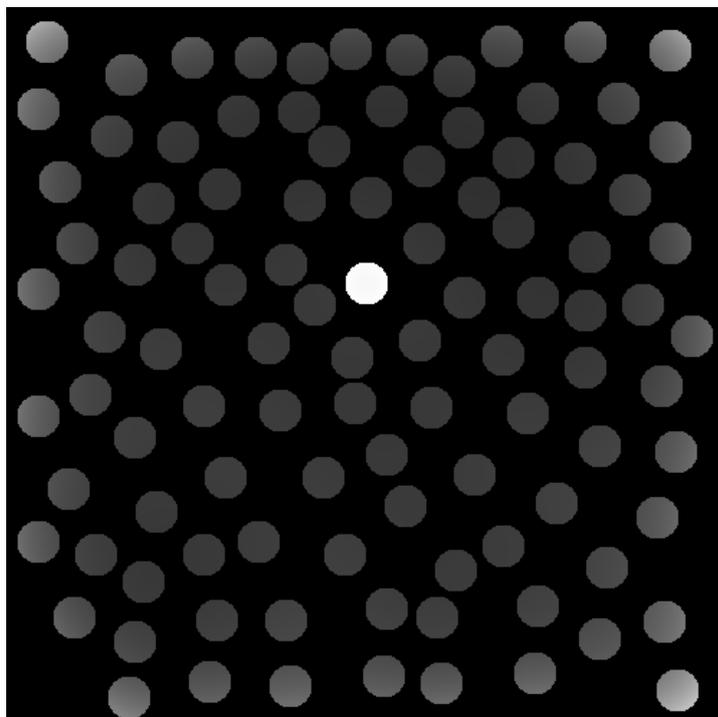


Figure 3.8: Experiment 2: Attention map.

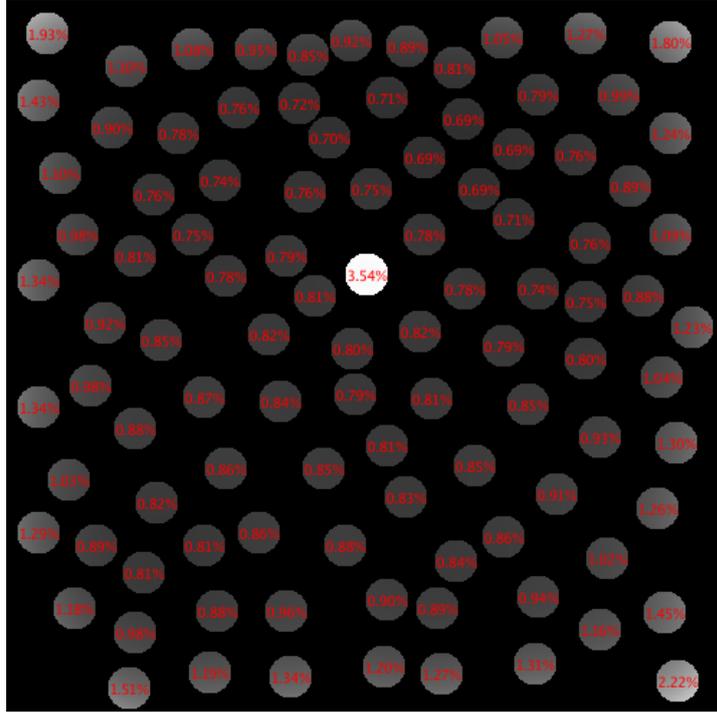


Figure 3.9: Experiment 2: Distribution of attentional resources.

3.6 Attention field: a tool for human-aware navigation

In order to properly quantify how the robot influences the comfort of the surrounding persons during the course of its motion, we must understand how this motion is perceived by these persons, i.e. how it affects their attentional state.

Our newly developed computational model of attention describes the full attentional state of the individuals, i.e. how the attentional resources of a person are shared between the surrounding salient elements of his or her environment. Using this model, in the context of human-aware navigation, we define for each person P_i a corresponding *attention field* \mathcal{A}_i on the robot's state space \mathcal{S} :

$$\mathcal{A}_i : \mathcal{S} \rightarrow \mathfrak{R}$$

For the sake of simplicity, in the case studies presented in chapter 5 we do not consider any features of the objects in the environment but only their position in the visual space, i.e. $\mathcal{VF} = \mathcal{V}$. Each object e_i is thus characterized by a subset of the visual space $E_i \subset \mathcal{V}$:

$$E_i = \{\mathbf{v} \in \mathcal{V} \mid \text{a part of object } e_i \text{ is seen at position } \mathbf{v}\}$$

We finally assume that each object e_i has a uniform bottom-up salience b_i , i.e. the bottom-up saliency map is simply defined as (we do not model variations of visual sensitivity over the visual field):

$$BU(\mathbf{v}) = \begin{cases} b_i & \text{if } \mathbf{v} \in E_i \\ 0 & \text{otherwise} \end{cases} \quad (3.8)$$

When the robot is in state $\mathbf{x} \in \mathcal{S}$, it is denoted $e_{robot}(\mathbf{x})$. For each possible value of the robot's state (e.g. position, orientation and velocity), the corresponding attention field value is equal to the amount of attentional resources attributed by the person to the robot when the robot is in that state:

$$\mathcal{A}_i(\mathbf{x}) = r_{e_{robot}(\mathbf{x})} \quad (3.9)$$

$r_{e_{robot}(\mathbf{x})}$ is computed using equation 3.7. In other words, the attention field aims to predict the amount of attentional resources the robot would receive when in a given state.

We can see that such a definition of the attention field naturally incorporates all the relevant characteristics of the robot's motion (e.g. position, orientation and velocity) as well as the current state of the person (e.g. position, orientation), the interactions and activities he or she is engaged in and more generally the influence of the environment as a whole. The attention field additionally accounts for the robot's physical properties such as its size and shape and other context factors such as the people familiarity with the robot (modeled by lowering the robot's salience over time).

For the sake of simplicity and visualization, only the 2-dimensional position of the robot is considered, i.e. $\mathcal{S} \subset \mathbb{R}^2$. Figure 3.10 depicts the attention field in a simple scenario featuring a person watching a painting. We can immediately see that the attention field discourages the robot to enter the space between the person and the painting, i.e. it extends in this case the activity space as defined by Scandolo and Fraichard [23]. The details on how the attention field is computed will be presented in chapter 5.

We distinguish two main types of uses of the attention field. Firstly when the task of the robot doesn't explicitly involve interacting with a person, it is best to minimize the distraction caused to the persons who share its environment. Distraction is defined as attracting the attention of the person away from its original focus, i.e. lowering the

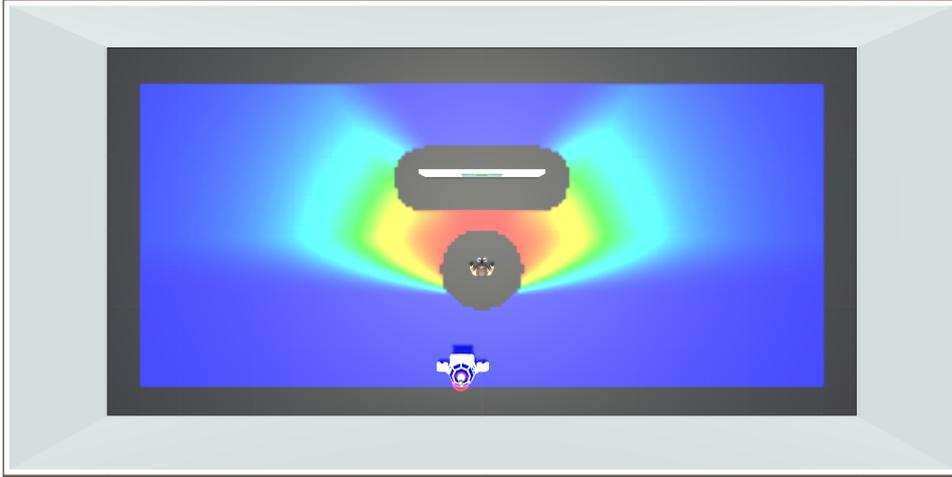


Figure 3.10: Illustration of an attention field in a scenario featuring a person watching a painting (the warmer the color the highest the value). The attention field is not defined in unfeasible regions for the robot (i.e. regions where a collision would occur).

attentional resources allocated to the initial object or region of focus in favor of a new (distracting) element; therefore the less attentional resources is attributed to the robot, the less the robot is distracting the person. The motion planner should thus aim to avoid as much as possible high value regions in the attention field to obtain non-distracting **appropriate paths**. Secondly, when the task of the robot involves interacting with a person, the robot's first aim is to acquire a certain amount of attentional resources from the person in order to convey its intention to interact. In this case, **appropriate end-pose selection** can be chosen based on the amount of attentional resources that the robot needs, i.e. the path should reach a high value point in the attention field.

Moreover, socially-aware motions should not cause surprise. Surprise is defined as the result of an unexpected significant event. From an attentional point of view, it can be described in terms of its effects on the person's attentional state, i.e. a sudden change in attentional resources distribution caused by the unexpected event. In the context of robot path planning, this generally corresponds to the sudden appearance of a robot (e.g. from behind an obstacle) leading to an abrupt change in the attentional resources allocated to the robot. The path planner should therefore aim to minimize local variations of the attention field along the path. This will discourage the robot from passing behind an obstacle too closely and constrain it to enter the person's field of view sufficiently far away. Additionally a smooth increase in the attentional resources allocated to the robot during the course of its motion is expected to yield better predictable motions.

In summary, in this section we formally defined the concept of attention field to be used for human-aware path planning. The attention field naturally incorporates all the relevant characteristics of the motion such as position, orientation and velocity of the

robot that may affect how the motion is perceived by surrounding persons. This novel concept of attention field can be used not only to produce non-distracting and non-surprising motions but also for end-pose selection in the context of human-robot interaction. This novel concept straightforwardly translates into motion planning constraints such as avoiding high values regions in the attention field in order not to distract the corresponding person or finding a path that ends on a high value point in the attention field so as to convey the robot's desire to interact.

3.7 Conclusion

Social spaces are mostly defined in an ad hoc manner. In this chapter, we tried to model the underlying psychological mechanisms responsible for these social constraints. More precisely, we built a new computational model of attention that quantifies how the attention of a given person is distributed among the persons and objects of his or her environment.

We would like to emphasize that, although our model is theoretically grounded and has been illustrated on two simple visual search scenarios, it has not been experimentally validated. Also it is a high level model that does not specify how the saliences of objects or persons are computed. Accordingly further research is needed in this direction.

Based on this computational model of attention, we introduced the novel concept of attention field that can be used for human-aware navigation. The attention field can be used to compute comfortable - i.e. not causing surprise and/or distraction - paths. The attention field can additionally be used for appropriate end-pose selection in the context of human-robot interaction. We expect this approach to overcome several limitations of social spaces discussed in chapter 2.

Socially-appropriate motions should not only optimize path surprise and path distraction (or end-pose selection) but also path safety and path efficiency. In next chapter we accordingly study the tools necessary for multi-objective path planning, i.e. optimization algorithms that can simultaneously handle multiple objectives. In chapter 5 we finally build a multi-objective path planner that relies on the concept of attention field in order to produce socially-acceptable motions. We also further illustrate the concept of attention field on simulated realistic case studies.

4 Many-objective path planning: the next challenge of autonomous navigation

4.1 Introduction

The robot global path planning problem aims to find a free or feasible path (i.e. most often a collision-free path) between two given configurations of the robot (or more generally between two subsets of its configuration space) and optimize this path with respect to specific criteria. In most cases, several objectives have to be optimized, e.g. path safety (e.g. avoid obstacles), path efficiency (e.g. path length), path smoothness, not cause discomfort to the surrounding persons, etc. A literature review reveals that most path planning studies use aggregation methods to scalarize the many objectives to be optimized into a single objective, and then aim at finding the path of minimum cost with respect to this objective. For instance in the context of human-aware navigation most studies use map-based approaches where they combine the relevant cost functions (typically path feasibility and social spaces) into a single costmap using a weighted sum [22–24] or maximum [23, 24] method. However, the obtained results are highly sensitive to the scalarization process (e.g. the weight vector used), and these methods are often unable to properly handle complex problems especially in the presence of many conflicting objectives. Since robot global path planning is intrinsically a multi-objective problem, multi-objective path planning has accordingly been a growing research topic in the last decade. Successful path planning algorithms capable of simultaneously handling multiple objectives have been developed, although to the best of our knowledge such studies limit the number of objectives that are handled simultaneously to three. Many-objective optimization, i.e. four objectives or more, is not much studied in the context of path planning. In addition, most studies consider point-to-point path planning where the goal configuration of the robot is known in advance. The aim of this chapter is to present a review of multi-objective path planning and more specifically many-objective optimization techniques. We then build on this review to single out a promising many-objective optimization algorithm that will be further investigated in the context of path planning. In this work, we only consider global path planning techniques, assuming full knowledge of the workspace. The robot will be considered holonomic and

more generally without kinematic constraints limiting its motion (a “free-flying” object approximation), a very good approximation for many new generation robots but which also stands for small sized robots compare to the size of the obstacles in their environment. Without loss of generality, two-dimensional motions will be considered. Finally, we make the assumption of environmental stationarity.

4.2 Multi-objective evolutionary algorithms

In this section we review the fundamental aspects of multi-objective optimization and more specifically multi-objective evolutionary algorithms. The knowledge gained in this review will enable us to subsequently select a promising optimization algorithm for many-objective path planning.

4.2.1 Multi-objective concepts

Since the many objectives to optimize are most often conflicting, unlike for single-objective optimization where there exists only a single optimum in objective space, there is no single best solution; instead, many “good” solutions exist which represent various compromises between the different objectives. Formally, we consider the problem of optimizing a vector of m objective functions $f = (f_1, \dots, f_m)$ with $\forall i \in \{1, \dots, m\} f_i : \mathcal{S} \rightarrow \mathfrak{R}$, where \mathcal{S} is known as the *search space* or *decision space*. Without loss of generality, we will consider that all the objectives are to be minimized by the optimization process. For each value $\mathbf{x} \in \mathcal{S}$ we define the corresponding objective vector \mathbf{y} :

$$\mathbf{y} = (y_1, \dots, y_m) = f(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_m(\mathbf{x}))$$

We naturally define the objective space S as all possible values of \mathbf{y} , i.e. the image of the search space by $f : S = f(\mathcal{S})$.

A partial order \prec named *Pareto dominance* is defined on the search space by the dominance relation:

$$\begin{aligned} \forall \mathbf{x}, \mathbf{x}' \in \mathcal{S}, \mathbf{x} \prec \mathbf{x}' \text{ (} x \text{ dominates } x') &\Leftrightarrow \\ \forall i \in \{1, \dots, m\} f_i(\mathbf{x}) \leq f_i(\mathbf{x}') &\text{ and } \exists j \in \{1, \dots, m\} \mid f_j(\mathbf{x}) < f_j(\mathbf{x}') \end{aligned}$$

A solution \mathbf{x} is simply called non-dominated within a subset of the search space if there is no solution in this subset dominating it. More specifically, if a solution is non-dominated within the entire search space, it is said to be *Pareto-optimal*, or simply non-dominated [78] and cannot be improved with respect to one objective without deteriorating another. The rest of the solutions are known as dominated solutions. Given the optimization problem at hand, Pareto-optimal solutions are optimal in the sense that no other solution in the search space are better when all objectives are considered. The set of all Pareto-optimal solutions is called the *Pareto-optimal set* or simply *Pareto set*:

$$\mathcal{P} = \{\mathbf{x} \in \mathcal{S} \mid \nexists \mathbf{x}' \in \mathcal{S} : \mathbf{x}' \prec \mathbf{x}\}$$

Similarly, the corresponding image under f in the objective space is called the *Pareto-optimal front* or simply *Pareto front* (the Pareto front is illustrated in figure 4.1):

$$P = f(\mathcal{P})$$

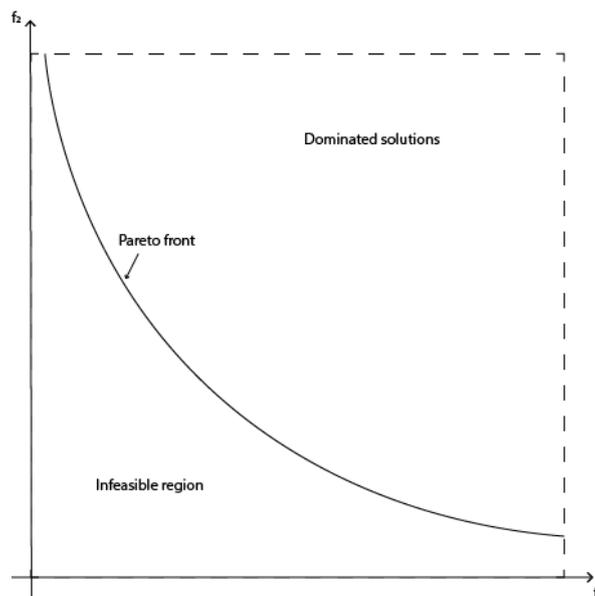


Figure 4.1: An illustration of the Pareto front for two objectives.

The aim of the optimization process is therefore to find a good approximation of the Pareto set, i.e. a set of good compromise solutions. It is important to note that, although we eventually seek a single compromise solution for our path planning problem, there is no satisfying methods that enable to find this solution in a single step; rather an approximation of the Pareto set is found, then one solution is selected from the set. The reasons for this are twofold: a) in most situations we have little or no knowledge about the structure of the Pareto front ahead of optimization time (in other terms we don't

know what we are looking for), thereby making it impossible to properly incorporate preferences into the optimization process (commonly done through a weight vector used for scalarizing the objective vector into a single objective) and b) multi-objective optimization algorithms perform better for complex problems with many objectives as their diversity preserving mechanisms prevent premature convergence towards non-optimal solutions.

For most problems, it is impossible to find the exact Pareto set with any algorithmic method. Indeed the size of Pareto set is often exponential for discrete problems and infinite for continuous problems [79]. Moreover many objective functions are often not known analytically. Multi-objective optimization techniques therefore aim to find an approximation of the Pareto set and approximate techniques are required; amongst them, metaheuristic approaches [80] are widely used. Population-based metaheuristic algorithms are a natural choice particularly well suited for finding an approximation of the Pareto set as they can find a whole set of solutions concurrently in a single run, in parallel.

4.2.2 Multi-objective evolutionary algorithms

Although several types of population-based metaheuristic multi-objective optimization algorithms exist, *evolutionary algorithms* appear to be the most studied [81, 82] and are reported to be very effective for multi-objective optimization, being able to handle complex problems including discontinuities, multimodality or noisy objectives [83]. Multi-objective evolutionary algorithms (MOEAs) aim to gradually find promising patterns in solution vectors by evolving a population of solutions using mechanisms inspired from biological evolution (mutation, crossover, selection). A successful multi-objective evolutionary procedure should produce a set of compromise solutions whose objective vectors [81]:

1. are as close as possible to the true Pareto front (convergence)
2. cover the whole Pareto front
3. are uniformly distributed, i.e. sample well the Pareto front (diversity)

The general structure of a MOEA can be described as follows:

Step 1 (Initialization) Generate an initial population of solutions (usually chosen randomly within the search space) and evaluate their fitness (i.e. compute the corresponding objective vectors).

Step 2 (Variation) Generate new solutions from selected members of the population using usual genetic operators (crossover and mutation).

Step 3 (Evaluation) Evaluate the fitness of new solutions.

Step 4 (Archive) Add best solutions to an archive containing only the best solutions found so far (optional).

Step 5 (Selection) Retain a subset of the population for the next generation.

Step 6 If the termination criterion is not fulfilled, go back to step 2.

The first MOEA was proposed by Schaffer [84] in 1985 (vector evaluated genetic algorithm (VEGA)). Since then, a large amount of evolutionary approaches to multi-objective optimization have been developed in order to improve the quality of the obtained solutions. Before reviewing MOEAs in more detail, it is necessary to mention that they all clearly outperform a pure random search strategy [85]. Below we look into the steps of a MOEA and describe the most successful techniques used by MOEAs.

Selection An important step of evolutionary algorithms is the selection of the subset of the population that will remain for the next generation. The selection process requires a mapping of the objective vectors in the current population to a ranking criterion. In most MOEAs, this ranking criterion is computed in two steps. The first and primary step is a selection pressure towards the Pareto front that uses the concept of Pareto dominance to promote non-dominated solutions over dominated ones. This is generally carried out using Goldberg's notion of non-dominated sorting [86] that establishes a partial order within the solutions: the non-dominated solutions in the population are assigned a rank 1, then the non-dominated solutions in the rest of the population are assigned a rank 2, etc. until all solutions have been ranked. The secondary step rates solutions that are equal in respect to the previous criterion and aims at preserving diversity (and therefore tries to prevent premature convergence) by promoting solutions that are located in poorly explored regions of the Pareto front. The concepts of fitness

sharing¹ [87, 88], crowding distance² [89], k-th nearest distance³ [90] and hypergrid⁴ [91] are often used as such a secondary selection criterion. Together, the two criteria establish a complete order amongst solutions, aiming to ensure convergence whilst maintaining diversity. The best solutions in regards to this order are then selected to be part of the next generation.

Archive For computational efficiency reasons, as the number of objectives grows, the population often cannot contain enough solutions to adequately represent a good approximation of the Pareto set. Therefore, the MOEA has to store all the best non-dominated solutions found during the evolution in an external archive. In this case, the final archive is expected to contain a better representative sample of the Pareto set than the final population. In addition to providing a good approximation of the Pareto set, the archive is sometimes used during the evolution to guide the search towards previously discovered promising regions [92]. In cases where the archive grows too large, Laumanns et al. [93] suggest only storing an approximation of the non-dominated solutions seen during the evolution process using the concept of ϵ -dominance.

Termination criterion In most cases the MOEA will evolve its population for a fixed number of generations or else an allocated number of function evaluations or time budget. Recently, Deb and Abouhawwash [94] discussed the Karush-Kuhn-Tucker proximity measure (KKTPM) which estimates the proximity of a set of solutions from the (unknown) Pareto front, and can be used as a possible termination criterion for a MOEA.

Performance evaluation Although several quality metrics have been introduced to evaluate the quality of a population as an approximation of the Pareto set, the *hypervolume* (also called S-metric)⁵ [95] stands out as it has several qualities that makes it preferable to other performance metrics. For instance, a finite set of solutions that maximizes the hypervolume metric contains only Pareto-optimal solutions [96] and yields

¹Fitness sharing is a diversity preserving mechanism that penalizes solutions located in densely populated areas in objective space: it rates the solutions depending on the number of similar solutions in the population, i.e. the more crowded the area around the solution is in objective space, the worst the ranking.

²The crowding distance of a solution is defined as the sum of the distances between its neighboring solutions for each objective; it provides an estimate of the population density around a solution without the need of a fitness sharing parameter.

³The k-th nearest distance is a density estimation technique that approximates the population density around a solution using a decreasing function of the distance to the k-th nearest solution in objective space.

⁴The hypergrid ranking criterion divides the objective space into a m -dimensional grid to compute the population density: the more solutions in a given cell, the more crowded the region and the worst the ranking of these solutions.

⁵The hypervolume indicator was originally called “size of the dominated space”.

an approximation of the Pareto front that is well distributed [97]. The hypervolume measures the volume of the dominated portion of the objective space relative to a reference point (see figure 4.2) and therefore rewards both the convergence towards the Pareto front as well as the diversity of the population, i.e. a well-spread distribution of points along the Pareto front. As the computational complexity for calculating the hypervolume grows exponentially with the number of objectives [98], Monte Carlo approximation methods are typically used when the dimension of the objective space is larger than two.

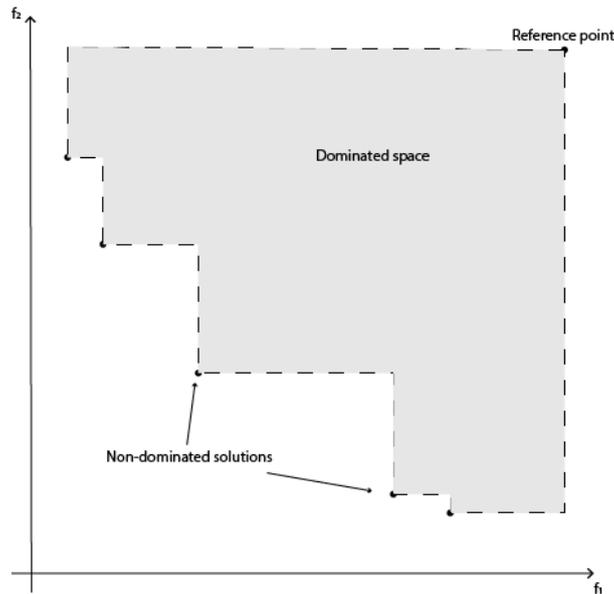


Figure 4.2: The hypervolume metric measures the volume of the dominated region of the objective space relative to a reference point.

4.2.3 Many-objective evolutionary algorithms

The difficulties encountered by multi-objective optimization techniques increase with the number of objectives to optimize. To emphasize this difficulty, problems with more than three objectives to be simultaneously optimized are commonly referred to as *many-objective* problems. In most multi-objective evolutionary algorithms, the selection process is biased towards the convergence of the population towards the Pareto front, whilst promoting the diversity of the population comes as a secondary goal. However, as the objective space grows in size, most solutions in the population become non-dominated with one another [99] thus severely lowering the efficiency of non-dominated sorting. Diversity preserving mechanisms then start to assume a larger and undesirable weight and affect the convergence of the population towards the Pareto front [100, 101]. Moreover, as the size of the objective space increases, the members of the population become

further away from each other; in this case, the reproduction step will tend to produce offspring that are far away from their parents, thus hindering the optimization process.

In summary, dealing with many (i.e. more than three) objectives is a significantly harder problem compared to the two or three objectives case. Consequently so far multi-objective path planning techniques do not consider more than three objectives to be simultaneously optimized.

4.3 Multi-objective path planning: a literature review

In this section, we present an overview of the current state of research on multi-objective path planning. It is worth noting beforehand that to the best of our knowledge, all studies on multi-objective path planning techniques are performed in the context of point-to-point path planning. As we will not discuss single-objective path planning here, the interested reader can refer to Hwang and Ahuja [102] and LaValle [103] for a comprehensive study.

4.3.1 Multi-objective algorithms for path planning

Here we present an overview of multi-objective optimization techniques used in the context of path planning.

Our review of the literature revealed that evolutionary algorithms are the most popular algorithms used for multi-objective path planning. Ahmed and Deb [104, 105] and Yuan et al. [106] use the popular non-dominated sorting genetic algorithm II (NSGA-II) [89] - a MOEA using non-dominated sorting combined with an explicit diversity preserving mechanism (crowding distance) and no archive. Wei and Liu [107, 108] and Chang [109] use an island parallel genetic algorithm (IPGA) [110], and Castilho and Trujillo [111] and Castillo, Trujillo, and Melin [112] use a simple custom genetic algorithm. Other population-based metaheuristic multi-objective optimization algorithms such as the multi-objective firefly algorithm [113] and the multi-objective shuffled frog-leaping algorithm [114] have also been shown to obtain good results in that context. Most studies suggest additional mechanisms to speed up convergence such as improving the initial population [113–115], deleting path nodes during the variation process in order to improve path length and path feasibility [116] or using a path repair mechanism (i.e. the unfeasible paths are modified by adjusting the sub-paths where collision occurs) [109, 111, 112, 116].

To the best of our knowledge, multi-objective path planning studies have so far consid-

ered two or three objectives. Ahmed and Deb [104, 105], Yuan et al. [106], Wei and Liu [107], Chang [109], Castilho and Trujillo [111], and Castillo, Trujillo, and Melin [112] simultaneously optimize path length and a measure of path safety. Wei and Liu [108] optimize path length and path smoothness (i.e. maximum curvature) whilst feasible paths are promoted by the selection process. Finally, Hidalgo-Paniagua et al. [113, 114], Hidalgo-Paniagua, Vega-Rodríguez, and Ferruz [115], and Jun and Qingbao [116] simultaneously optimize three objectives: path length, path smoothness and a measure of path safety.

In summary, multi-objective path planning is still a novel field of research and evolutionary algorithms seem to be the most popular choice as a multi-objective optimization algorithm. There appears to be no single “best” algorithm, although NSGA-II seems to be the most used. However, multi-objective path planning studies so far do not consider more than three objectives to be simultaneously optimized. Also, to the best of our knowledge, no methods have yet been proposed to pick a “good” solution from the approximation of the Pareto front in the context of path planning.

4.3.2 Path representation

Finding a good parametrization of the problem at hand is critical for the success of any evolutionary algorithms, and this effect gets more pronounced as the dimension of the objective space increases. In this section we review the most commonly used path parameterizations in the context of multi-objective path planning.

Choosing a robust path representation allowing for an unknown path length and shape is challenging, which is the reason why many studies make the assumption that the path is monotonous along one axis [105, 111, 112]. This assumption enables us to choose a fixed number of parameters to represent the path. For instance we can use a grid-based approach where the path moves one grid at a time towards its destination in a fixed number of steps. The drawbacks of this technique however is that it requires careful choosing of the grid size, i.e. small enough for describing complex paths and avoid obstacles and large enough so that the robot can be approximated as holonomic, i.e. perform any necessary rotation between two successive cells.

In order to lift the assumption of path monotonicity along one axis and handle more complex paths, a more flexible approach consists in defining the path using an ordered sequence of arbitrary intermediate points between which the path is then interpolated (with varying degrees of smoothness: from linearly piecewise paths [113–116] where segments join two consecutive points to smoother high-degree piecewise polynomial paths [107–109]). The variables to optimize are the coordinates of these points. This method enables the use of both a fixed [109] or variable [107, 108, 113–116] number of points.

To interpolate between points, in order to help the construction of smooth paths, spline functions of degree ≥ 2 are most often preferred since they are smooth at the knots. In the context of path planning, it is appropriate to consider two categories of splines: waypoints based splines where waypoints are points through which the path must pass and are interpolated using continuous polynomial functions, and control points based splines where the position of the control points influences the shape of the path but do not represent locations the path has to go through. The use of waypoint based splines has been shown to yield poor performances in comparison to control points based splines [104] as they have poor control over the bounds of the path and thus cannot constrain the path to a specified bounding polygon, therefore producing paths that may overshoot outside the environment boundaries (see figure 4.3).

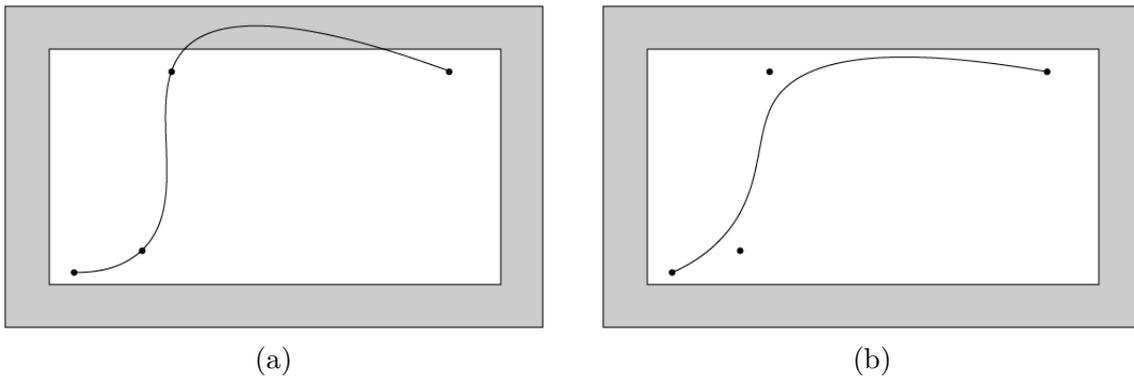


Figure 4.3: Waypoints based spline (a) and control points based spline (b).

The most popular type of control points based splines are basis splines or B-splines which are a generalization of Bézier curves that have C^2 continuity, therefore ensuring smooth paths. B-splines are particularly interesting for use in an evolutionary algorithm as they guaranty that the path always fits within the convex hull defined by the whole set of control points, a property inherited from Bézier curves [117]. In addition, B-splines have the local modification property, i.e. local changes in the list of control points is reflected by local changes in the resulting path. B-splines come in two flavors: uniform - i.e. uniform spaced knots - and non-uniform. Typically B-splines of order 3-5 are used. For example, Ahmed and Deb [104] use uniform B-splines of order 4.

Finally, the number of control points chosen to characterize the path greatly influences the performances of any (multi-objective) evolutionary algorithm, and the optimal number of points strongly depends on the problem considered. A simple approach used to deal with this problem is to run the algorithm for different values of the number of control points and observe its influence on the Pareto front. Chang [109] presents an extensive study of the influence of the number of points on path planning performances.

4.3.3 Discussion

Multi-objective path planning has grown into an active area of research and a review of the literature has shown that most studies rely on multi-objective evolutionary algorithms. However, multi-objective path planning studies have so far considered three or less objectives to be handled simultaneously by the optimization process, which is less than we need for our application. Although the chosen MOEAs (e.g. NSGA-II) perform well in this context, the literature on MOEAs suggests that these algorithms are largely outperformed by more recently developed evolutionary algorithms for many-objective (i.e. more than three objectives) optimization. In the next section, we accordingly select a promising recently developed MOEA to be applied to the many-objective path planning problem.

We also reviewed path parametrization in the context of evolutionary multi-objective path planning: B-splines appear to be a particularly adequate choice and will therefore be used by our path planner.

Finally, whilst many studies successfully use multi-objective optimization techniques to obtain a set of good compromise solutions to the path planning problem - most often in terms of path length, collision avoidance and sometimes path smoothness - we found no suggestions as to how to systematically pick a “good” solution on the Pareto front.

4.4 Approximation-guided evolutionary algorithm

In chapter 5, we show case studies that require to simultaneously optimize between 3 and 4 objectives; we therefore need a multi-objective algorithm that performs well with 3 objectives and more. Most existing MOEAs designed to deal with many-objective problems (i.e. 4+ objectives) use fairly different approaches compared to algorithms developed to handle a lower number of objectives; only a few algorithms offer good performances for both 3 and 4+ objectives. Also, a major problem of most many-objective optimization algorithms is that they tend to be computationally expensive, i.e. polynomial or even exponential in the number of objectives, and they are therefore not suited for real-time path planning.

Considering the aforementioned requirements, the recently developed approximation-guided evolutionary (AGE) algorithm [118] appears as an excellent MOEA for the many-objective path planning problem. Firstly, this algorithm has shown to perform very well handling both few and many objectives. Secondly, it has a worst-case runtime linear in the number of objectives and therefore is a promising candidate to achieve real-time path planning. Additionally, unlike most other many-objective evolutionary algorithms,

AGE has no meta-parameters influencing the evolution process besides the population and offspring sizes (although the desired approximation quality of the archive can be adjusted, it has been shown to mainly affect the computational cost of the algorithm with no major effect on the quality of the outcome). In order to achieve good performances when dealing with many-objective problems, AGE relies on an archive which stores an approximation of all non-dominated solutions seen during the evolution process. The archive is additionally used by the selection process to assess the quality of newly produced solutions. More precisely, assuming that the archive is a good approximation of the Pareto front, the quality of a population is measured by its approximation quality with respect to the archive: the selection process thus keeps solutions that contribute the most to a good approximation quality of the population. Parent solutions are selected for variation through a binary tournament that relies both on the concept of non-dominated sorting and crowding distance (inspired from NSGA-II [89]). The interested reader is referred to Wagner et al. [118] for a more detailed description and implementation details of AGE.

4.5 Conclusion

Choosing an algorithm carefully tailored to the multi-objective path planning problem as well as picking an adequate path parametrization is not trivial and greatly influences the performances of the optimization process. In this chapter, we have presented a review of the current research in multi-objective path planning and identified AGE [118] as a promising algorithm to be applied to the many-objective path planning problem.

The high computational needs of MOEAs remain however an obstacle for real-time path planning, although the use of AGE along with a deeper investigation into the performance problem is expected to help considerably reduce the running time of the path planner. Some suggestions for MOEA performance gains are: path repair mechanisms, knowledge integration techniques (e.g. approximate function evaluations), innovization⁶ [120], dimensionality reduction, GPU computing and massively parallel computing. Once the performance issue is solved, MOEAs are moreover ideal candidates for anytime path planning as the quality of the obtained solutions is refined during the evolution process.

Future work will also include exploring a particular type of B-spline, Pythagorean hodograph (PH) curves⁷ [121] for use in many-objective path planning as they have several advantages over other types of splines: a) the integral of the curvature is small, mean-

⁶Innovization is the idea of learning and deducing common patterns of good and near-optimal solutions from the population evolved by a MOEA [119].

⁷The hodograph of a curve is its first derivative. A polynomial parametric curve $M(t) = (x(t), y(t))$ is a Pythagorean hodograph curve if $\frac{dx}{dt}^2 + \frac{dy}{dt}^2$ is the square of another polynomial.

ing the obtained curves are smooth, b) lower curvature can be traded off for a longer curve naturally, c) most commonly used properties of the curve such as its length, curvature and “bending energy” are known in closed-form and can thus be calculated very efficiently [122].

Finally, several aspects of real world multi-objective path planning such as handling noisy objectives⁸ and dynamic objectives have not been discussed here and so far, to the best of our knowledge, have not been studied in the context of path planning.

In next chapter, we will use the tools discussed in this chapter to build a many-objective path planner that produces socially acceptable motions by taking the notion of attention into account.

⁸When dealing with noisy objectives, the performances of the chosen algorithm strongly degrade as a result of an erroneous estimation of the quality of solutions, leading to both the acceptance of bad solutions and the reject of good solutions.

5 An attention-based approach for human-aware navigation

In this chapter, we build a many-objective path planner that relies on the concept of *attention field* in order to obtain socially-aware paths. We illustrate the capacities of our path planner in several case studies.

5.1 An attention-based human-aware path planner

In order to find paths that are safe, efficient and comfortable from a human point of view, we build a many-objective path planner based on the many-objective evolutionary algorithm AGE [118] selected in the previous chapter.

We use uniform 2-dimensional B-splines for path parametrization. The coordinates of the control points of the B-spline are naturally taken as variables in the optimization process, i.e. a path is represented by a list of n 2-dimensional control points. Assuming that the robot's heading and the steering angle are respectively proportional to the first and second derivative of the path [123], we use B-splines of order 3 in order to ensure their continuity. The B-splines are clamped at the initial robot's position (and goal position in the case of point-to-point path planning) using multiplicity of knots.

The variation step of the evolution process uses polynomial mutation [124] (an operator widely used in multi-objective evolutionary algorithms) with an associated distribution parameter $\eta = 1.0$, and single point crossover. We have observed that the performances of the optimization process are sensitive to the mutation rate and that a mutation rate of 0.03 gives the best results (similar values have been reported by Ahmed and Deb [104]). The size of the population and the number of generated offspring are both set to 1000 individuals. The optimization process evolves 100 generations. For the archive we choose $\epsilon = 0.01$. As the performances of the optimization process depend on the number of control points n chosen to represent the paths, we do not fix the value of n and allow the number of intermediary control points to vary between 1 and 5. In order to achieve this, we initialize the population with paths having a variable number of control points taken randomly within this range; the chosen single-point crossover operator subsequently

accommodates variable length path representation. These parameters are kept constant throughout all this study, and have been selected through a series of prior evaluation of the algorithm’s performances (observing improvement in the approximation of the Pareto front for different parameters values) tested on the considered case studies.

Convergence of the evolution process is qualitatively evaluated by visualizing the evolution of the hypervolume metric of the obtained approximation of the Pareto front contained in the archive.

5.1.1 Objectives

We define five objectives to be relied upon by our path planner in order to produce socially acceptable paths. Whilst *path feasibility* and *path length* will always be optimized, three other objectives are introduced based on the concept of attention field to achieve better socially-aware motions: *path distraction*, *path surprise* and *path reach*. Although it is important to ensure that the path is as smooth as possible, path smoothness will not be taken as an independent objective; instead we rely on B-splines to naturally produce smooth paths [104]. Subsequently \mathcal{S} will denote the state space of the robot. In our case studies, we only consider the influence of the robot’s position and therefore $\mathcal{S} \subset \mathbb{R}^2$ is the set of all possible robot’s position in the environment. $p \subset \mathcal{S}$ will denote a path, represented by its control points.

Path feasibility

When using a standard population-based algorithm for multi-objective path planning, it is inefficient to handle obstacle avoidance as a hard constraint as this would reject many promising paths in the initial population and during the evolution [104]. Obstacle avoidance is thus handled as an objective through a cost equal to the normalized path integral of a *feasibility field* along the path:

$$Feasibility(p) = \frac{\int_p \mathcal{F}(\mathbf{s}) ds}{\int_p ds} \quad (5.1)$$

ds denotes an elementary arc length of the path. $\mathbf{s} : [a, b] \rightarrow p$ is an arbitrary bijective parametrization of the path p , $\mathbf{s}(a)$ and $\mathbf{s}(b)$ being the endpoints of the path and $a < b$.

The feasibility field \mathcal{F} is a scalar field on the robot’s state space simply defined as:

$$\mathcal{F} : \mathcal{S} \rightarrow \mathfrak{R}$$

$$\mathbf{x} \rightarrow \begin{cases} 1 & \text{if the robot collides with an obstacle when it is at position } \mathbf{x} \\ 0 & \text{otherwise} \end{cases} \quad (5.2)$$

For the sake of simplicity, the check for robot collision uses a spherical collider centered on the robot and slightly enlarged relative to its actual size in order to prevent too close proximity with obstacles. For more complex dynamic cases, we would expect the feasibility field to take into account the predicted motion of moving entities.

As we consider a static environment, the feasibility field is precomputed prior to the optimization process with a resolution of 10 cm x 10 cm. The integral is naively computed using a discrete sum, using 500 regularly spaced samples of the feasibility field along the path: for each point, the feasibility value is bilinearly interpolated based on the precomputed costmap.

Path length

Path efficiency is defined in terms of traveled distance - or path length:

$$Length(p) = \int_p ds \quad (5.3)$$

As there is not any closed-form solution in term of simple functions for the length of B-splines, we approximate the path length by the length of the polyline formed by 500 regularly spaced points along the path.

Path distraction

In situations where the task of the robot does not involve any explicit interaction with the surrounding persons, it is often best to keep distraction to a minimum. In chapter 3, we introduced the concept of attention field which provides a measure of the distraction caused by the robot as a function of its state. For each person P_i we thus define the corresponding path distraction as the maximum value of the corresponding attention field \mathcal{A}_i along the path:

$$Distraction_i(p) = \max_p \mathcal{A}_i(\mathbf{s}) \quad (5.4)$$

Intuitively, we believe that allocating a low amount of attentional resources to the robot for an long period of time is less likely to disrupt the person’s current activity than if the robot briefly obtains most of his or her attentional resources. In other words the robot should aim to always stay in a zone of low distraction (i.e. attention field value) rather than try to minimize the average of the attentional resources obtained along its path. In this sense, it is relevant to take the maximum of attentional field values along the path as a measure of the distraction caused by the robot’s motion.

As the attention field is not known analytically, an exact value cannot be computed for the path distraction. Similarly to the feasibility field, as we consider static scenarios, the attention field is precomputed prior to the optimization process with a resolution of 10 cm x 10 cm. We compute an approximation of the path distraction using straightforward discrete sampling, comparing 500 attention field values uniformly sampled along the path (for each point, the attention field value is bilinearly interpolated based on the precomputed costmap). It is worth noting here that such an approach does *not* scale with the dimension of the robot’s state space where it becomes too expensive to precompute an attention costmap prior to the optimization process. In this case, for instance, applying a differential optimization algorithm to the maximization problem is expected to yield good results.

The bottom-up saliency map BU and top-down saliency map TD are defined over the visual space \mathcal{V} only (see 3.6). A position in the visual field $\mathbf{v} = (x, y) \in \mathcal{V}$ is represented relative to the person’s gaze direction by its horizontal angular displacement $x \in [-90^\circ, 90^\circ]$ and vertical angular displacement $y \in [-90^\circ, 90^\circ]$. We assume that the saliences of the objects and persons in the environment are known and we compute the bottom-up saliency map using equation 3.8. In the considered case studies, we assume a very simple object-based top-down saliency map defined as:

$$TD(\mathbf{v}) = \begin{cases} 2 & \text{if a part of the object } e_i \text{ is seen at position } \mathbf{v} \text{ and the object } e_i \text{ is also attended to} \\ 1 & \text{otherwise} \end{cases} \quad (5.5)$$

In order to identify the object or person visible in a given direction, we perform a simple ray casting. We compute discrete bottom-up and top-down saliency maps with a resolution of $1^\circ \times 1^\circ$. Finally, the suppressive field is modeled by a two-dimensional Gaussian, i.e. for a given position in the visual field $\mathbf{v} = (x, y) \in \mathcal{V}$:

$$s(x, y) = \frac{1}{2\pi\sigma^2} e^{-\left(\frac{x^2+y^2}{2\sigma^2}\right)} \quad (5.6)$$

The value of σ is arbitrarily set to 60° . Qualitatively, the attentional resources allocated to an object or person e_i are strongly affected by the presence of a nearby object or person e_j (in visual space) if the absolute angular distance between e_i and e_j is less than 60° .

In most cases the more objectives that are considered by the many-objective path planning algorithm, the lower the robustness and performance. Thus for each case study considered, the first step is to try reducing the total number of objectives to be simultaneously optimized. In case studies featuring multiple persons P_i not to be disturbed, instead of considering each person's distraction independently - i.e. minimizing the path distraction for each person, we instead consider a single objective representing the path distraction along a *distraction field* D defined as a combination of each attention field \mathcal{A}_i :

$$\mathcal{D}(\mathbf{x}) = \max_i \mathcal{A}_i(\mathbf{x}) \quad (5.7)$$

The total path distraction is then defined as:

$$Distraction(p) = \max_p \mathcal{D}(\mathbf{s}) \quad (5.8)$$

Path surprise

Socially appropriate paths must try to minimize the effect of surprise caused to the surrounding persons. We define the surprise caused by the robot at a position \mathbf{s} when traveling along the path p as the instantaneous rate of change of the attentional field along the path at position \mathbf{s} . The path surprise caused to the person P_i can then be written as:

$$Surprise_i(p) = \max_p \frac{\partial \mathcal{A}_i}{\partial \mathbf{s}}(\mathbf{s}) \quad (5.9)$$

As the attention field is not known analytically, its directional derivative is approximated using a central difference:

$$\frac{\partial \mathcal{A}_i}{\partial \mathbf{s}}(\mathbf{s}) \approx \frac{\mathcal{A}_i(\mathbf{s} + d\mathbf{s}) - \mathcal{A}_i(\mathbf{s} - d\mathbf{s})}{2 \|d\mathbf{s}\|}$$

Also, similarly to the path distraction, we compute an approximate value for the maximum of the directional directive of the attention field along the path through straight-forward discrete sampling using 500 attention field values uniformly sampled along the path.

In order to reduce the number of objectives to be simultaneously optimized, in the presence of multiple persons P_i we consider a single objective representing the maximum value of the path surprise for all the persons:

$$Surprise(p) = \max_p \max_i \frac{\partial \mathcal{A}_i}{\partial \mathbf{s}}(\mathbf{s}) \quad (5.10)$$

Path reach

In situations where the robot’s task directly involves interacting with a person, the robot needs to express its desire to interact, i.e. it needs to attract the person’s attention. In this case we handle end-position selection as part of the path planning process in order to yield an optimum compromise between appropriate end-pose and motion comfort. For that purpose, we introduce a new objective - the path reach - to be optimized by our path planner. In the case where the robot wants to convey its desire to interact with a person P_i , the path reach is simply defined as the value of the attention field \mathcal{A}_i at the end of the path:

$$Reach_i(p) = \mathcal{A}_i(\mathbf{s}(b)) \quad (5.11)$$

$\mathbf{s}(b)$ denotes the end of the path.

5.1.2 Prior knowledge

To ensure best performances of the multi-objective optimization process, it is best to rescale the objectives into the same range in order to ensure that density computation mechanisms (e.g. crowding distance) don’t favor one objective over the others. Moreover all objectives to be optimized don’t have the same weight. For instance, although path length must be optimized so as not to produce unnecessary long paths, it should not

influence much the optimization process as long as the path length doesn't exceed a "reasonable" value. In order to rescale the objectives and incorporate prior knowledge into the optimization progress, we therefore scale the objectives through the following sigmoid function (x represents an objective value):

$$f_{\sigma_{obj}}(x) = 1 - e^{-\frac{x^2}{2\sigma_{obj}^2}} \quad (5.12)$$

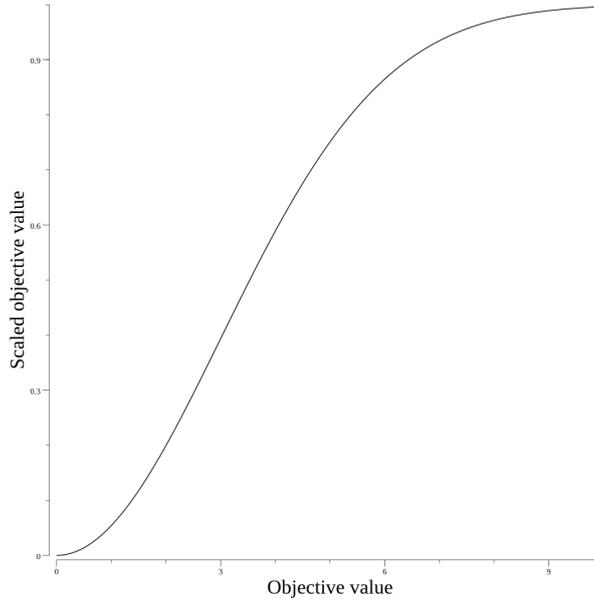


Figure 5.1: Example of a sigmoid function defined by equation 5.12 ($\sigma = 3$)

Figure 5.1 illustrates a sigmoid function defined by equation 5.12. This scaling step has been found to significantly improve the performances of the optimization process. The value of σ_{obj} encapsulates our preferences for the corresponding objective, i.e. as long as the objective value is less than σ_{obj} it should not interfere too much with the convergence of the other objectives. For instance, for the path length we found that a value for σ_{length} equal to the distance between the robot's initial position and the goal (or the person the robot wants to interact with in the case of human-robot interaction) yields good results. Such a value for σ_{length} will be used for all the considered case studies. No scaling will be applied to the path feasibility objective as by construction it already has values in the $[0, 1]$ range and there is no relevant prior knowledge to include.

5.1.3 Selecting an appropriate path

Once a good convergence of the multi-objective path planning optimization process has been reached and the obtained archive A represents an acceptable approximation of the

Pareto set, we need to single out a single solution that will be used as the result of our path planner. In the absence of preferences on the path itself (e.g. proximity to a previously computed path), we obtain good results by choosing a path p amongst the feasible paths closest to the smallest attainable values for all $m - 1$ objectives other than path feasibility using the Min-Max operator:

$$p = \min_{p \in A, Feasibility(p)=0} \max_{j \in [2, \dots, m]} (p[j] - o[j]) \quad (5.13)$$

For a path p and an index $j \in [2, \dots, m]$, $p[j]$ denotes the j^{th} objective value for the path p . o is the objective vector containing the smallest objective values for all the paths in the archive A :

$$\forall j \in [2, \dots, m] : \quad o[j] = \min_{p \in A} p[j] \quad (5.14)$$

5.2 Application architecture for real-time many-objective path planning

In this section, we describe an application architecture to implement and test our path planner.

Many frameworks exist to develop robotics applications (e.g. the Robot Operating System), but choosing such a framework often requires severe compromises on the communication patterns, performances, reliability, programming languages, development environment, available libraries - to name just a few. They moreover often require a high degree of encapsulation with the chosen solution. These limitations are mostly due to the strongly opinionated nature of existing frameworks, often not based on standards, and discouraged us from using such solutions. Instead we build a minimalist unopinionated architecture for our application based on standard solutions and ideas developed in the context of service-oriented architectures and web technologies.

Even our simple path planner features several components and we describe a highly distributed architecture in view of achieving reliable, real-time path planning. Each application component is run inside a Docker¹ container. In order to deploy and manage the components in our application, we use Kubernetes², the container orchestration

¹<https://www.docker.com>

²<https://kubernetes.io>

system that powers Google Kubernetes Engine³. Kubernetes provides containers and services management including monitoring and logging, and it runs on most platforms. In particular, the use of Kubernetes enables us to run our application both on a single development computer as well as on Google Kubernetes Engine where we can run our application on a cluster which can scale up to hundreds of instances. Kubernetes naturally enables us to dynamically insert and remove components from our running application, a feature that we make heavy use of. Lastly, Kubernetes provides the concept of pods as an abstraction above containers, i.e. a group of one or more co-located and co-scheduled containers (with shared network and storage). The use of pods enables us to handle dependencies between tightly coupled components and for components that must be run in a shared context for performance reasons (e.g. fast IPC socket based communication between containers). In a service-oriented architecture, typically one pod provides one service.

Finally, good communication between the components of our application is essential. In our application, we avoid encapsulation from a single communication middleware and we make use of several standard styles and protocols including REST, gRPC⁴ and ZeroMQ⁵ (the excellent performances of ZeroMQ are presented in Dworak et al. [125] through a comparison with other communication middlewares). Typically we use ZeroMQ with FlatBuffers⁶ over IPC sockets for performance critical communication within a pod, and gRPC (which enables easy tracing) for communication between pods. It is worth mentioning that the use of Kubernetes with Docker brings no or very little networking overhead for communication between pods.

We discuss below the components of our application. Each component is a minimalist Docker container, with the exception of the simulator UI. All the application code is written in Go⁷, with the exception of the simulation code which is written in C#.

Simulator In order to assess the benefits of our approach and in view of future validation experiments involving user studies and virtual reality, we want to carry out our experiments in a realistic environment and therefore we need a simulation tool. In the next section we will discuss using Unity⁸ as a simulation platform. The simulator features a robot listening for motion commands on a ZeroMQ TCP subscribe socket. The motion commands are serialized in JSON due to the lack of current support for Protocol Buffer in Unity 5.x (although the new 2017.x versions of Unity released in September

³<https://cloud.google.com/kubernetes-engine>

⁴gRPC (<https://grpc.io>) is a high performance, open-source universal RPC framework using Protocol Buffers over HTTP 2.0

⁵ZeroMQ (<http://zeromq.org>) is a message-oriented middleware library (which resembles the standard Berkeley sockets) supporting numerous communication patterns and various transports.

⁶<https://google.github.io/flatbuffers>

⁷<https://golang.org>

⁸<https://unity3d.com>

2017 introduce a new Mono runtime with .NET 4.6 support which will enable us to use the official C# Protocol Buffer library). The robot is considered holonomic and the motion commands are sent as a list of successive waypoints through which the robot should pass. The motion between these waypoints is handled by a local reciprocal velocity obstacles (RVO) planner which helps to deal with dynamic situations and/or uncertain environments better. Each of the robot's sensors (and other sensors in the environment) data are sent on ZeroMQ UDP publish sockets.

Simulator UI When experimenting in a simulated environment, it is useful to visualize this environment as well as to be able to make regular changes to the scene during a series of experiments. In order to achieve visualization and control over our environment, we connect a running Unity editor to our simulator and set up synchronization of often changed variables (e.g. position, orientation, salience) for the robot and a few selected objects in the environment. The simulator UI can listen to the published motion commands to display the corresponding path.

Perceived environment Collected sensor data need to be processed and the resulting information stored in a globally accessible database to be used by the different application components. Due to the highly dynamic and heavy nature of some data, each component must be able to retrieve just the data it needs and receive updates when available. Such a data sharing mechanism is typically implemented using the publish-subscribe pattern. We argue however that this pattern is not well-suited for this use for two reasons. Firstly, in order to preserve both network and application resources, only new data should be sent; this poses a problem on first connection when relying on the publish-subscribe pattern since the whole data set needs to be acquired before subscribing to updates. Secondly, this approach is not flexible as it requires a custom publish channel for each component with the data it needs, therefore it requires prior knowledge of each component's data needs. We therefore suggest the use of GraphQL⁹, a query language which enables a component to request just the data it needs and that can, for example, return it in JSON format (although the use of FlatBuffer or Protocol Buffer here would be more appropriate where latency and bandwidth are a bottleneck) along with new updates when available over a ZeroMQ TCP-based dealer-router channel.

Multi-objective evolutionary optimizer In order to allow to experiment with several different multi-objective optimization techniques, the optimizer must be decoupled from our path planner. The optimizer is configured by the path planner which provides it with the DNS names of the services representing the various objectives to be optimized. The path planner is responsible for starting and stopping the optimization process. During

⁹<http://graphql.org>

the evolution process, the population/archive is regularly sent on a ZeroMQ TCP publish socket and consumed by the path planner.

Path planner The path planner plays a central role in the application as it fulfills multiple roles. Firstly, it listens for user control commands regarding the path planning process (start, stop, etc.). Secondly, it makes sure that the optimizer and the objectives to be optimized are available (using the Kubernetes Go client), it configures the optimizer with the list of objectives and optionally some meta-parameters needed by the optimizer, and it subscribes to the output population/archive from the optimizer. Thirdly, it periodically selects a preferred path (which is close enough to the previous one) and publishes a motion command to be consumed by the simulator and visualization components.

Objectives In order to achieve high performance, the computation of objective values must take advantage of a parallel architecture. For each objective, multiple worker components are thus spawn and responsible for computing the corresponding objective values. Each objective worker often requires information on the environment that it uses to compute the objective values; accordingly, each worker locally stores within its in-process memory a partial copy of the “perceived environment” database containing the data it needs (frequently updated to reflect the state of the perceived environment). To minimize network traffic and decrease the load on the “perceived environment” component, a dealer is inserted between the workers and the “perceived environment” component. The dealer receives updates from the “perceived environment” and forwards these updates to each worker; the “perceived environment” thus only sends each update once per objective. A load balancer is responsible for distributing the computing load amongst the available workers. Finally, as the computation of objective values can be expensive, a memoizer¹⁰ is introduced in front of the load balancer. The memoizer listens for objective value requests from the optimizer on a ZeroMQ TCP router socket (the received path and returned objective value are serialized using FlatBuffers). All of these components are collocated in a pod and communicate over IPC sockets in order to obtain high performances.

The diagram shown in figure 5.2 illustrates the complete architecture.

¹⁰A memoizer helps speed up our optimization process by storing the results of expensive objective values computation and returning the cached result when asking for a previously computed value.

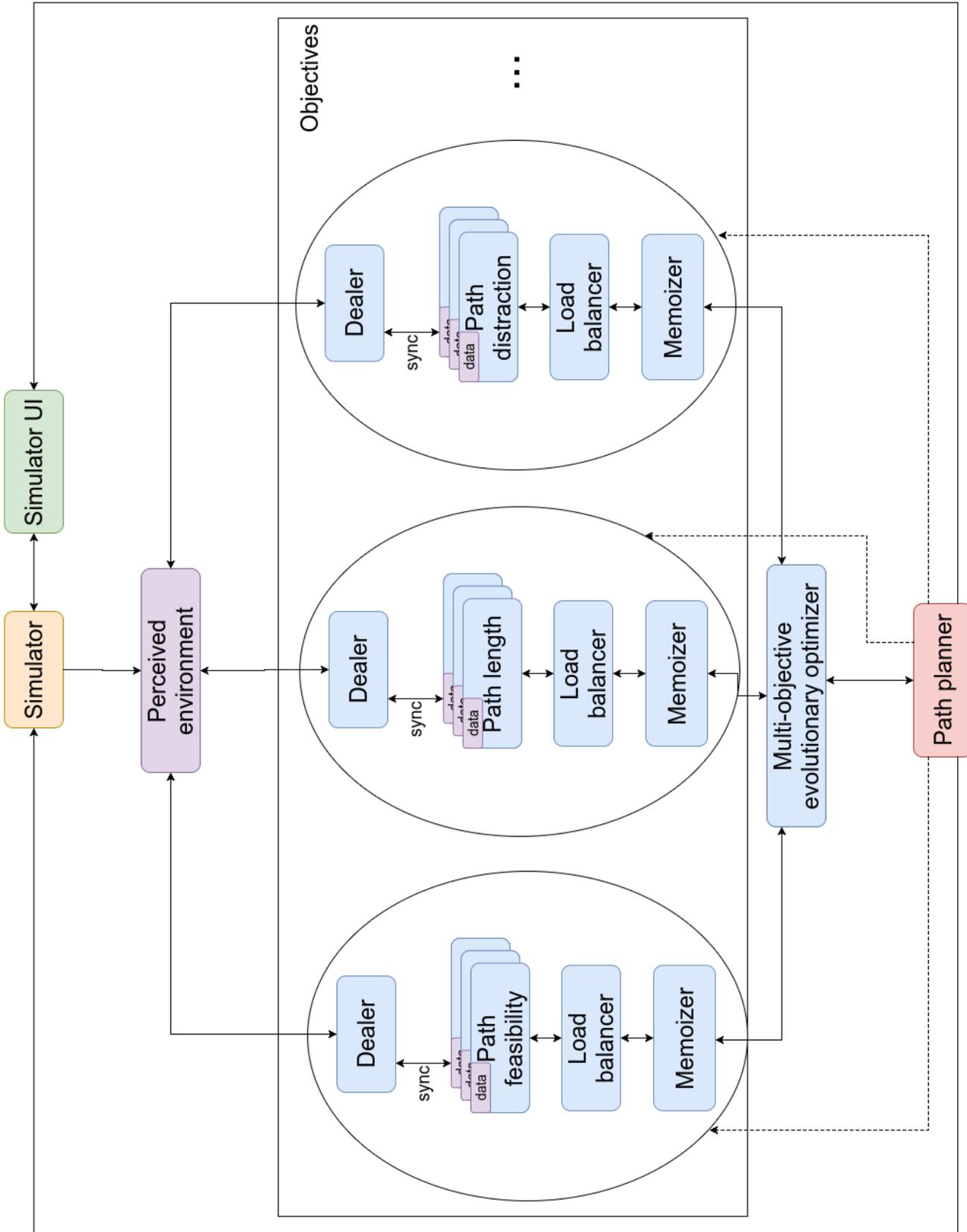


Figure 5.2: Application architecture. Rectangle shapes are containerized components. For each objective, multiple containers are grouped in a pod, indicated by a circular shape.

5.3 Simulation platform

Robotic simulators are used by academia and industries alike and have become an unavoidable tool in the workflow of robotics research. From industrial manipulators to social robots, the fields of research are varied and so are the requirements of the tools needed. For the last decade this has driven the development of many custom solutions, developed from the ground up and opinionated in favor of a particular branch of robotics research, such as Gazebo [126] and the Virtual Robot Experimentation Platform (V-REP) [127]. As these solutions appeared rather inflexible, hard to use and none of them were a good fit for our simulations needs, we decided to investigate the use of a game engine - Unity¹¹ - as an alternate platform for robotic simulation¹².

Unity appears to be an excellent choice as a robotic simulation software and stands out from existing solutions thanks to the following features:

Component oriented Similarly to V-REP, Unity’s component-based architecture is well suited for robotic simulation. Every scene object can have one or more embedded scripts. This enables a distributed control architecture (i.e. each object is individually controlled by its embedded scripts). Also this enables us to create portable content (e.g. sensors) from one scene to another.

2D and 3D Unity seamlessly simulates both 2D and 3D environments, and components can be reused between 2D and 3D modes.

UI As a game engine, Unity’s UI is both nice-looking, intuitive, and easily customized (e.g. for data visualization).

ZeroMQ Unity supports the use of ZeroMQ to communicate with external components of the robotic application.

Easy-to-use Unity is very easy to use, with good documentation and debugging tools.

Virtual reality (VR) Although we will not take advantage of the virtual reality capabilities of Unity in this work, we suggest that validation experiments make use of VR.

Performance As a game engine, Unity is highly optimized and can handle very complex environments.

¹¹<https://unity3d.com>

¹²Both robotic simulators and game engines indeed share most of the same functionalities: a physics engine, a rendering engine, and convenient programmatic and graphical interfaces.

Wide range of file format Unity supports a large range of 2D and 3D standard file formats.

Our experience so far with Unity shows that it can be used as a highly capable robotic simulation platform. We argue that its unopinionated nature towards robotics makes it one of the most robust, efficient and easy-to-use tools in this domain. Its component-based architecture is perfectly suited for robotic simulation and Unity can seamlessly handle visualization and physics of one or many robots, in realistic indoor or outdoor environments, in 2D or 3D. Simulated components can communicate with external application components using ZeroMQ. Unity is moreover expected to play a central role in conducting user studies using virtual reality.

In order to fully appreciate the use of Unity as a robotic simulation software, it is essential to carry out a performance comparison with existing dedicated solutions such as Gazebo and V-REP. We would also have liked to show more use cases to illustrate the benefits of this platform. Moreover, in order for Unity to be widely adopted as a robotic simulation platform, common sensors and plugins for commonly used robotic functionalities should be implemented.

However, Unity is not a perfect solution; amongst the limitations we encountered is the language restriction - currently C# being the only well supported language.

Finally, although this section has been solely concerned with Unity, we expect the use of other game engines such as Unreal¹³ to also be very successful as robotic simulation platforms.

5.4 Case studies

In this section, we present the results obtained by our path planner in several simple case studies in order to illustrate key points of our approach.

In order to properly test and appreciate the benefits of our path planner, experiments are performed in a simulated realistic environment. The environment features a service robot in a large room. Persons, obstacles and salient objects are added to the environment depending on the considered case study, but the environment will remain simple enough to unambiguously highlight the quality of the paths obtained by our path planner. The size and shape of the objects and persons in the environment are given realistic values and the size of the room is 20 m x 10 m. We assume perfect global knowledge of the

¹³<https://www.unrealengine.com>

environment including objects and persons saliences. Only static case studies will be considered.

The considered case studies are grouped according to the task given to the robot: minimizing distraction or attracting attention in view of initiating an interaction. Due to lack of time to implement and test the full architecture described earlier in this section, our path planner has been fully implemented but in a simplified version of the proposed software architecture. Although throughout this section arbitrary values are chosen for objects and persons saliences, it has been verified that the qualitative results presented in this chapter do not depend strongly on the chosen salience values.

5.4.1 Human-aware motions: minimizing distraction

When the task of the robot doesn't involve explicit interaction with the surrounding persons, it is in most cases best to try to minimize the distraction caused by the robot's motion. The two case studies considered here will show that the attention field enables us to describe interaction and activity spaces to avoid.

Towards attention-based interaction spaces

In this case study we illustrate the capacity of our path planner to avoid disturbing a group of people in interaction.

Description This example scenario features three persons in interaction, i.e. involved in a conversation. The saliences of the persons and the robot are given in table 5.1. Each person P_i attends to every other person $P_{j \neq i}$ (top-down focus of attention). The robot's task is to navigate to a destination point located across the room. The scenario is depicted in figure 5.3.

Table 5.1: Persons and robot bottom-up salience.

Persons	Robot
1	2

Method In this case study, the path planner aims at minimizing three objectives: path feasibility, path length and path distraction. The feasibility field and distraction field are respectively shown in figure 5.4 and figure 5.5. We set no prior knowledge on the path distraction objective.

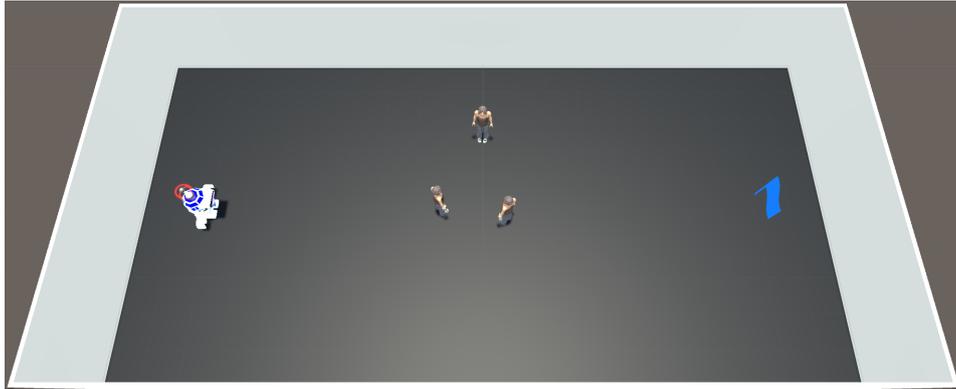


Figure 5.3: Three persons in conversation. The robot's task is to navigate across the room to a fixed destination point (represented by the flag).

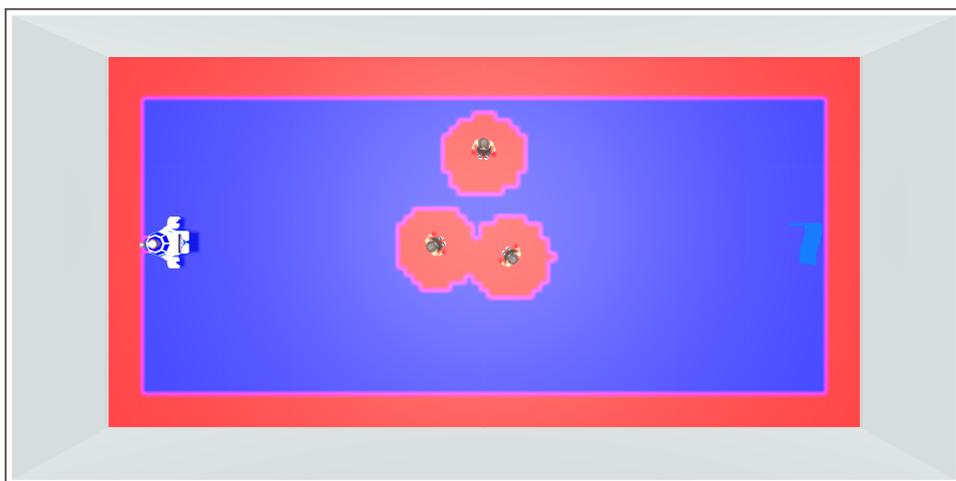


Figure 5.4: Feasibility field (the regions shown in red are unfeasible).

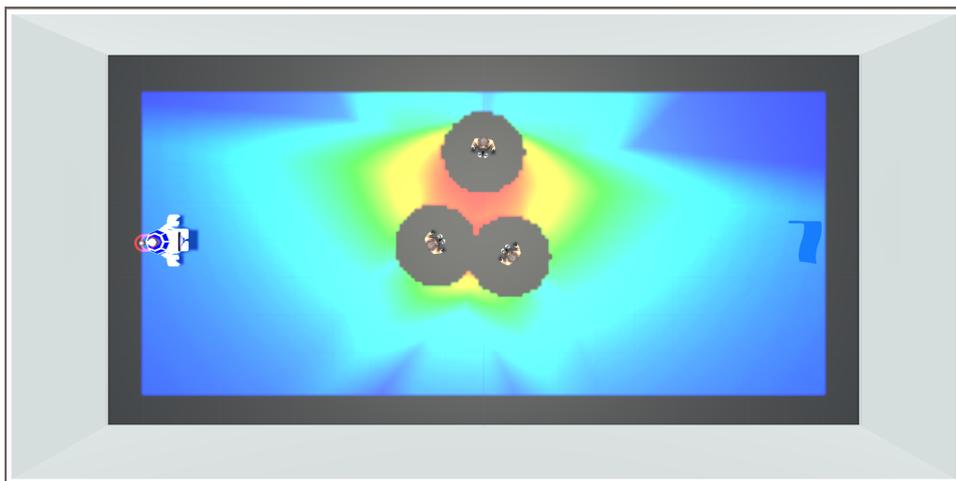


Figure 5.5: Distraction field (the warmer the color the highest the value). The distraction field is not defined in unfeasible regions.

Results Figure 5.6 shows the obtained path. This path has 5 intermediary control points.

Discussion The obtained path illustrates the capacity of our motion planner to yield a short path that is sufficiently far from the people in interaction so as not to cause distraction. In other words, our motion planner found a compromise between path length and path distraction. In the presence of interaction groups, we think that the distraction field distinctly represents the region surrounding the persons in conversation and therefore models well the corresponding interaction space.

Towards attention-based activity spaces

Although the concept of activity space as defined by Lindner and Eschenbach [5] cannot be trivially modeled by an attention-based approach, in practice, attention spaces are typically defined on a per-activity basis as the convex hull of the person and the object he or she interacts with [23]. This case study illustrates that the distraction field models well such activity spaces, and naturally handles multiple concurrent activities. We then show the capacity of our path planner to find a natural compromise between path length and caused discomfort.

Description This scenario features a person looking at two paintings and it is depicted in figure 5.7. The saliences of the paintings and the robot are given in table 5.2. The person attends to the two paintings (top-down focus of attention). The robot’s task is to navigate to a destination point located across the room. The environment is cluttered so that it is highly inefficient in terms of path length to find a path that does not disrupt the activity.

Table 5.2: Objects and robot bottom-up salience.

Frog painting (left)	Dog painting (right)	Robot
0.5	3	2

Method Similarly to the previous case study, the path planner aims at minimizing path feasibility, path length and path distraction. The feasibility field and distraction field are respectively shown in figure 5.8 and figure 5.9, and we set no prior knowledge on the path distraction objective.

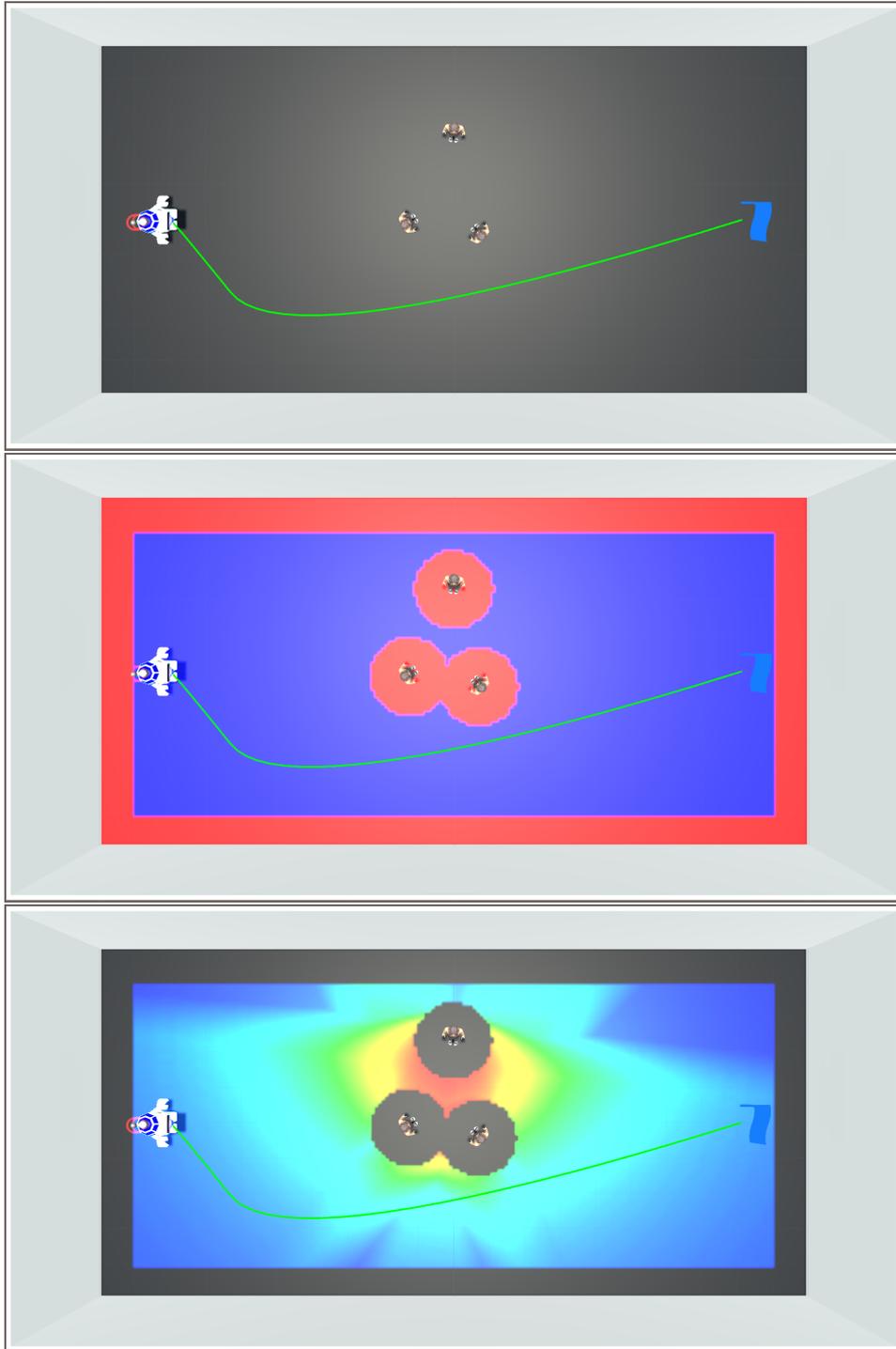


Figure 5.6: Non-distracting path obtained by our path planner.



Figure 5.7: A person involved in two concurrent activities (watching the two paintings). The robot's task is to navigate across the room to a fixed destination point (represented by the flag).

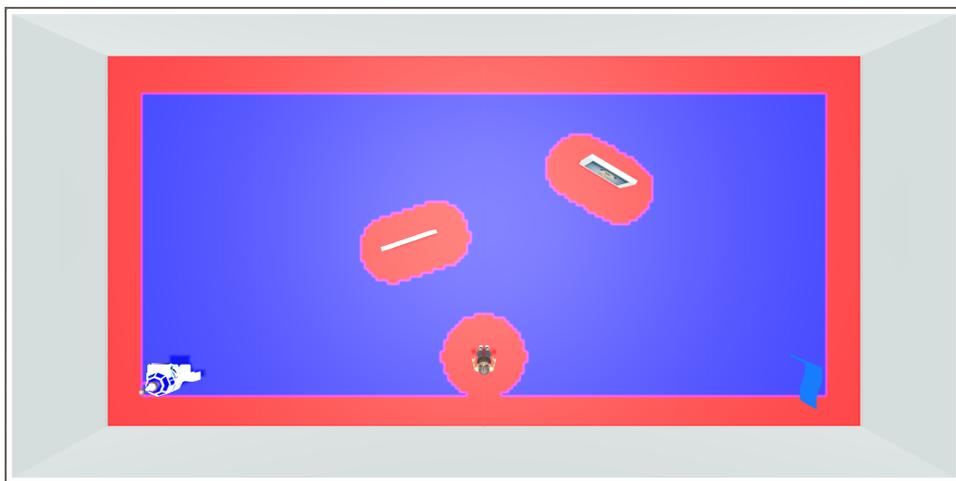


Figure 5.8: Feasibility field (the regions shown in red are unfeasible).

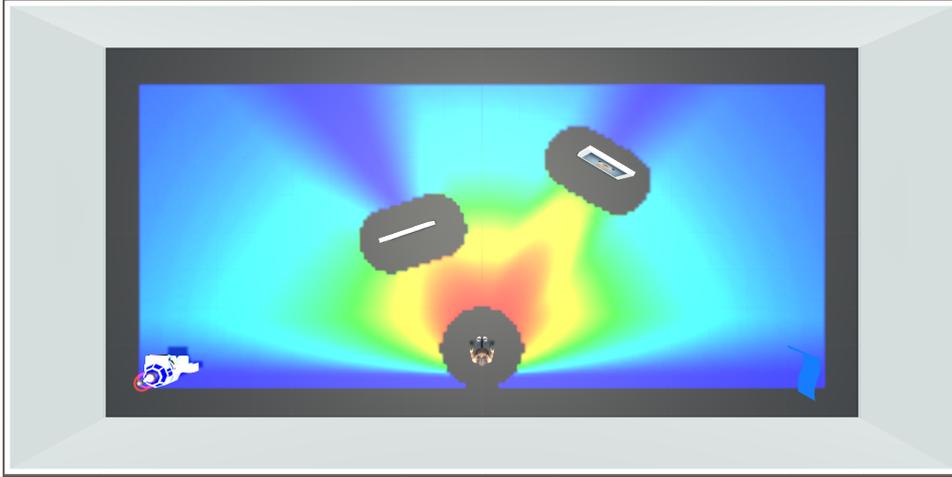


Figure 5.9: Distraction field (the warmer the color the highest the value). The distraction field is not defined in unfeasible regions.

Results In this scenario, our path planner favors a shorter path by not avoiding medium values in the distraction field, as shown in figure 5.10. The path has 3 intermediary control points.

Discussion In this simple scenario, we have shown that the distraction field manages to successfully characterize the social space corresponding to the current activities of the person. Furthermore, in this scenario an optimally non-disturbing path would lead to a very long and rather unnatural path. Instead, it seems more natural to find a compromise between path length and path distraction and an intrusion into medium values of the distraction field can lead to a much shorter path. Our path planner accordingly avoids taking “unnecessarily” long detours.

5.4.2 Human-aware motions for human-robot interaction

In the context of human-robot interaction, the robot needs to choose an end-pose that will optimally convey its intention to interact with a person by attracting his or her attention. The three considered case studies illustrate the capacity of our path planner to select an appropriate end-pose in view of interaction and simultaneously optimize the path comfort.

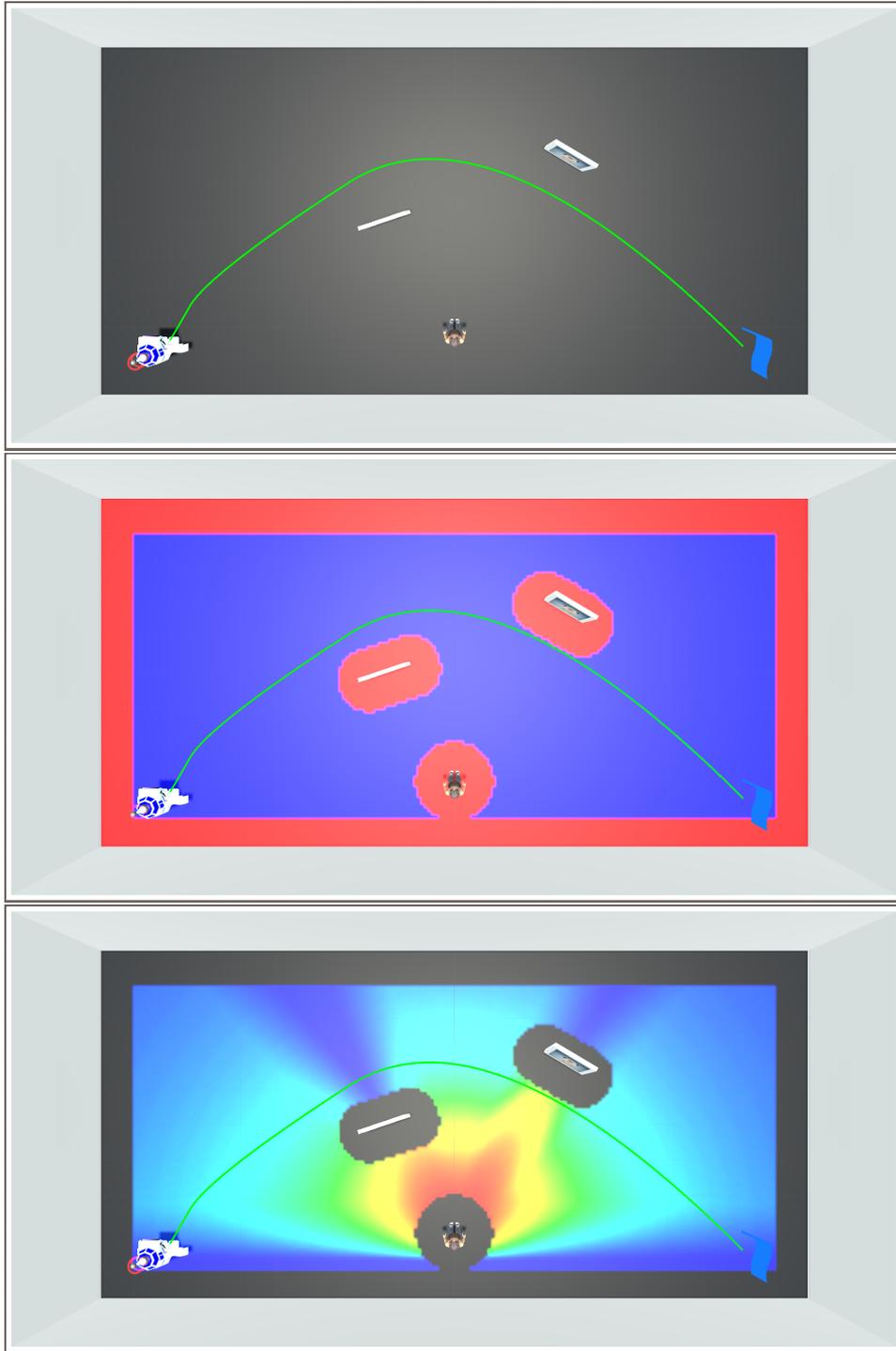


Figure 5.10: Path obtained by our path planner.

End-pose selection for human-robot interaction

We illustrate on two simple scenarios the capacities of our path planner to handle end-pose selection in the context of human-robot interaction.

Description The scenarios investigated here feature a person looking at a painting. The saliences of the painting and the robot are given in table 5.3. The person attends to the painting (top-down focus of attention). The two scenarios differ solely by the initial position of the robot, and they are depicted in figure 5.11a and figure 5.11b. The task of the robot is to convey its intention to interact: the path goal is thus unknown and it is a quantity to be optimized as part of the path planning process.

Table 5.3: Object and robot salience.

Frog painting	Robot
1	2

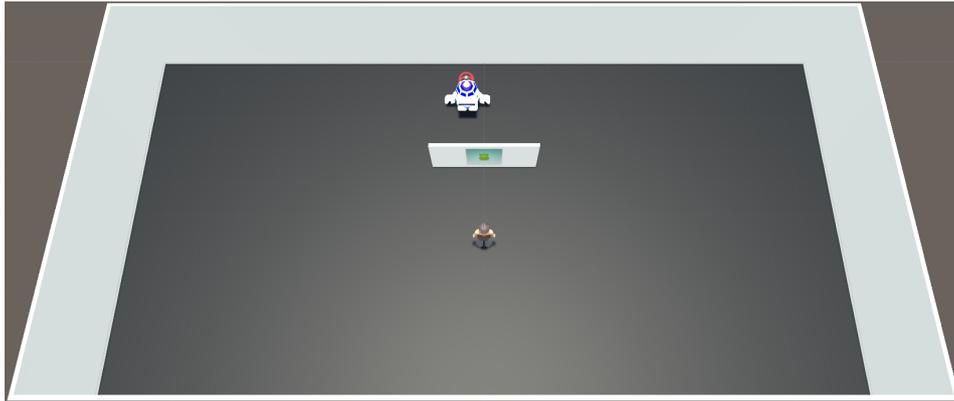
Method Here, the path planner simultaneously attempts to optimize the path feasibility, path length and path reach. The feasibility field and distraction field are respectively shown in figure 5.12 and figure 5.13. We set a prior knowledge for the path reach with $\sigma_{reach} = 0.01$.

Results The obtained paths are shown in figure 5.14 and figure 5.15. The paths both have 1 intermediary control point.

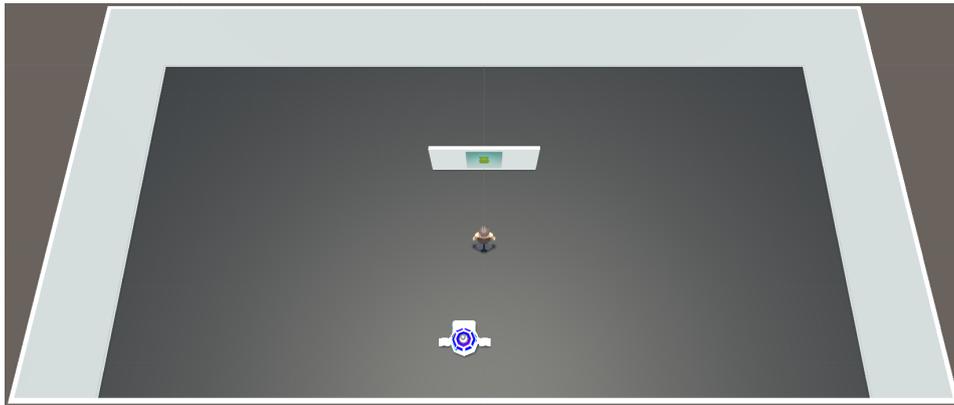
Discussion In these simple scenarios, our path planner successfully finds an optimal end-pose that conveys its intention to interact.

Non-surprising motion

Socially-aware motions must try to minimize the effect of surprise caused to the surrounding people during the robot motion. Here we illustrate how our motion planner can yield more socially acceptable paths by simultaneously optimizing the path surprise.



(a)



(b)

Figure 5.11: A person involved in an activity (watching the painting). The robot's task is to navigate to an optimal end-pose that conveys its intention to interact.

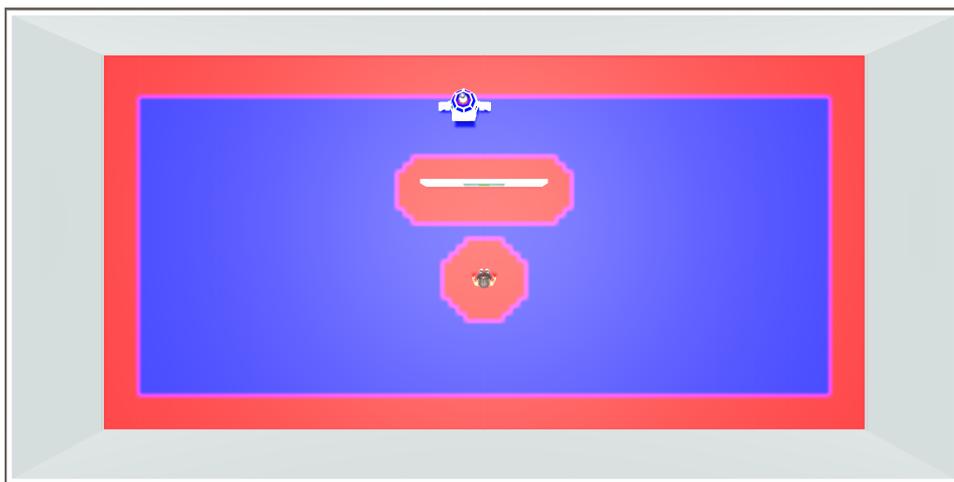


Figure 5.12: Feasibility field (the regions shown in red are unfeasible).

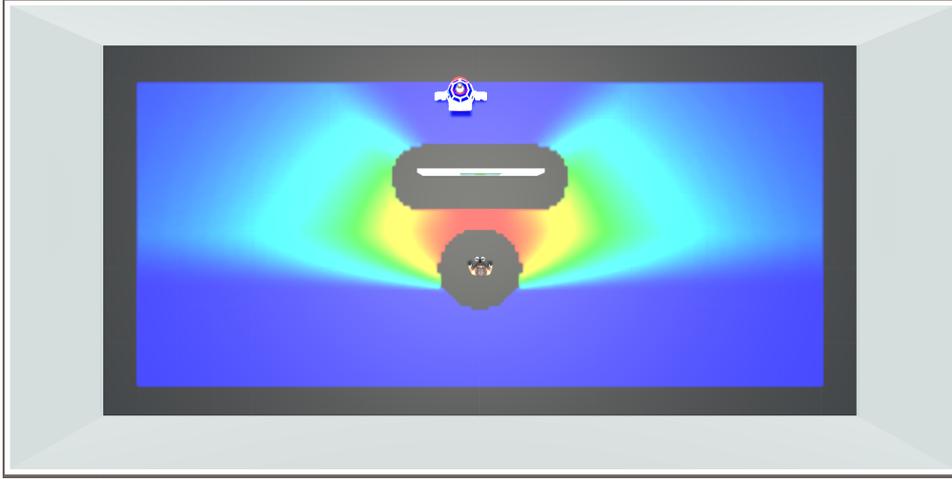


Figure 5.13: Distraction field (the warmer the color the highest the value). The distraction field is not defined in unfeasible regions.

Description We consider the scenarios introduced above and illustrated in figure 5.11a and figure 5.11b. The saliences of the painting and the robot are kept the same and they are given in table 5.3. The task of the robot is to convey its intention to interact.

Method Here, the path planner simultaneously attempts to optimize four objectives: path feasibility, path length, path reach and path surprise. We set a prior knowledge for both the path reach and the path surprise with $\sigma_{reach} = 0.01$ and $\sigma_{surprise} = 0.001$.

Results Figure 5.16 and figure 5.17 show a comparison between the path obtained with and without optimizing for the path surprise. The path for the scenario represented in figure 5.11a has 2 intermediary control points. The path for the scenario represented in figure 5.11b has 1 intermediary control point.

Discussion In these simple scenarios, our path planner successfully concurrently handles four objectives to be optimized. The resulting paths meet our expectations and seem to yield a good compromise between appropriate end-pose and motion comfort, whilst keeping the path short. In particular our path planner finds paths that enter the field of view of the person further away so as to avoid suddenly appearing close to the person and causing surprise. It is noteworthy to mention that the path planning process is currently quite slow: using all 8 cores of an Intel i7 Haswell processor, it takes approximately 2 seconds to evolve one generation of the population, thus the optimization process takes more than 3 minutes.

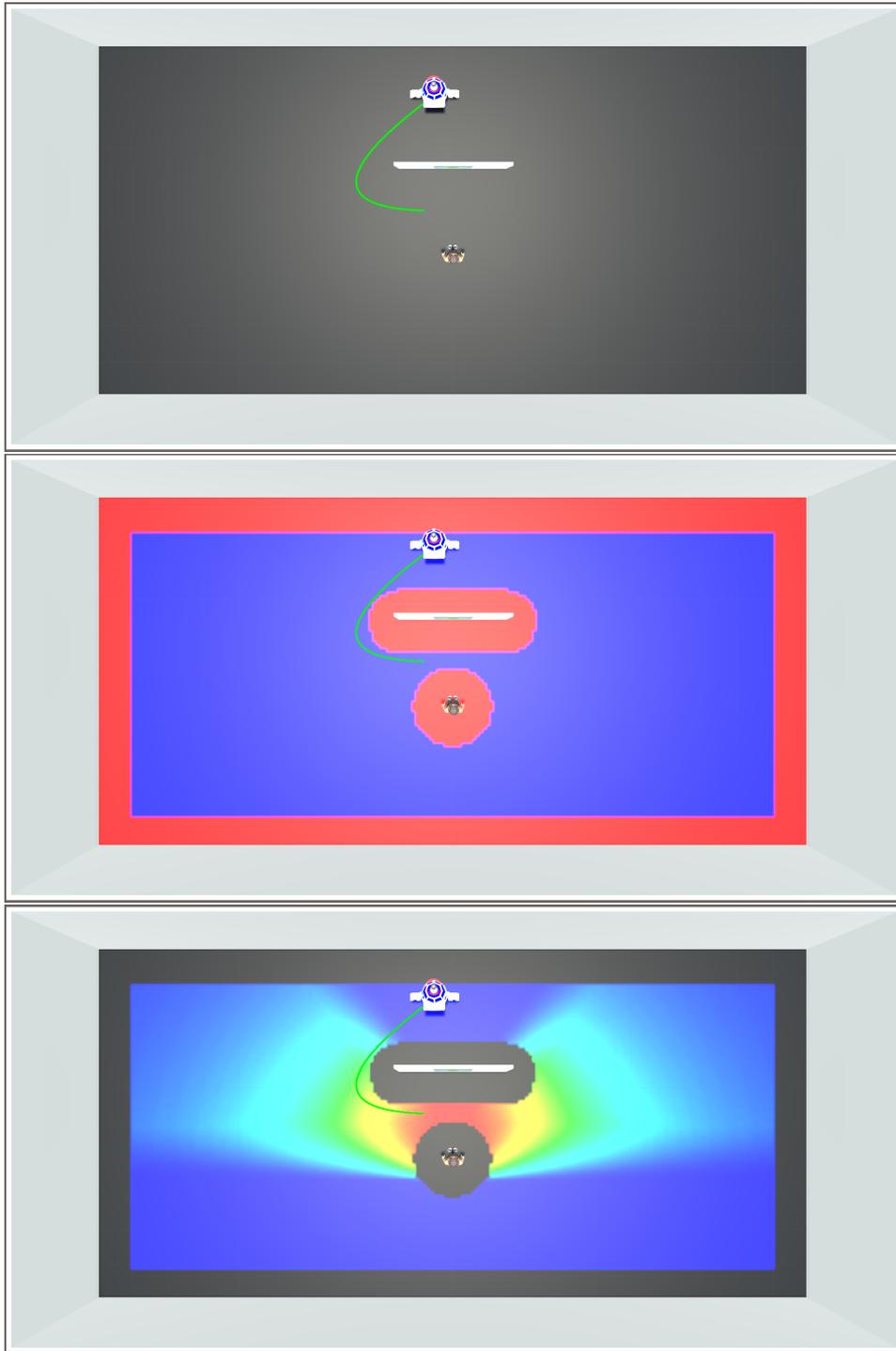


Figure 5.14: Path obtained by our path planner for the scenario represented in figure 5.11a

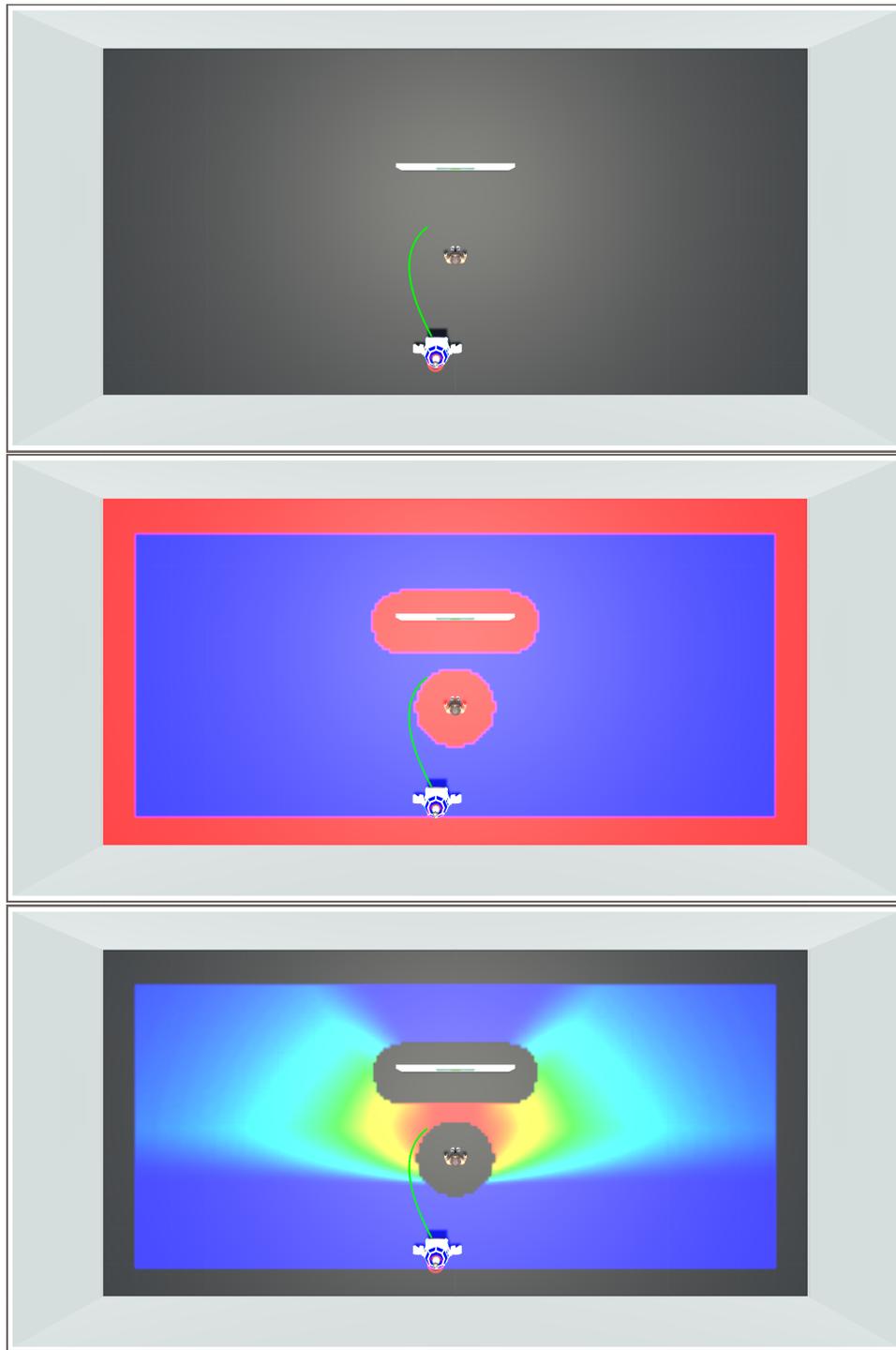


Figure 5.15: Path obtained by our path planner for the scenario represented in figure 5.11b

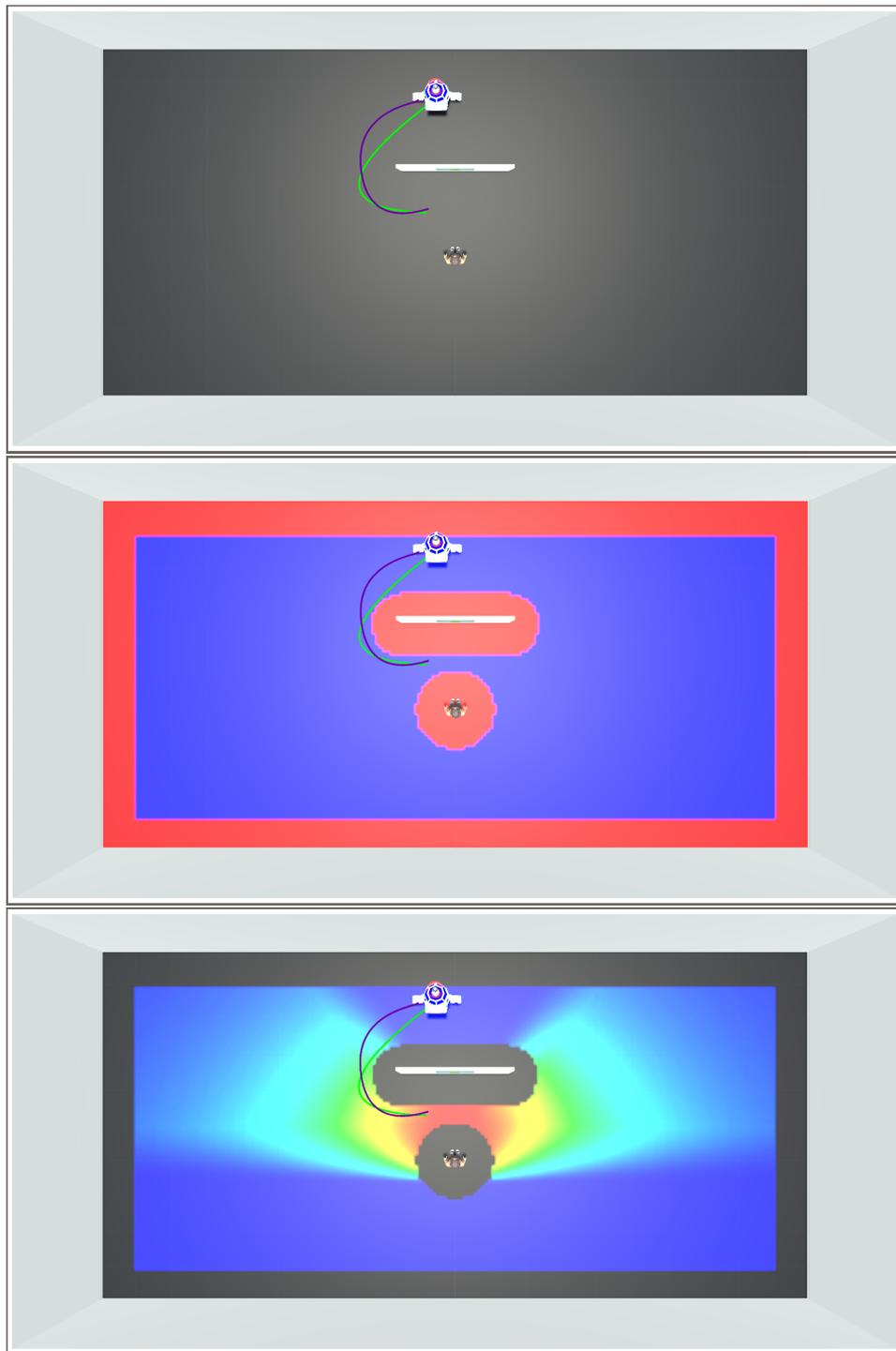


Figure 5.16: Path obtained by our path planner with (shown in purple) and without (shown in green) optimizing the path surprise for the scenario represented in figure 5.11a

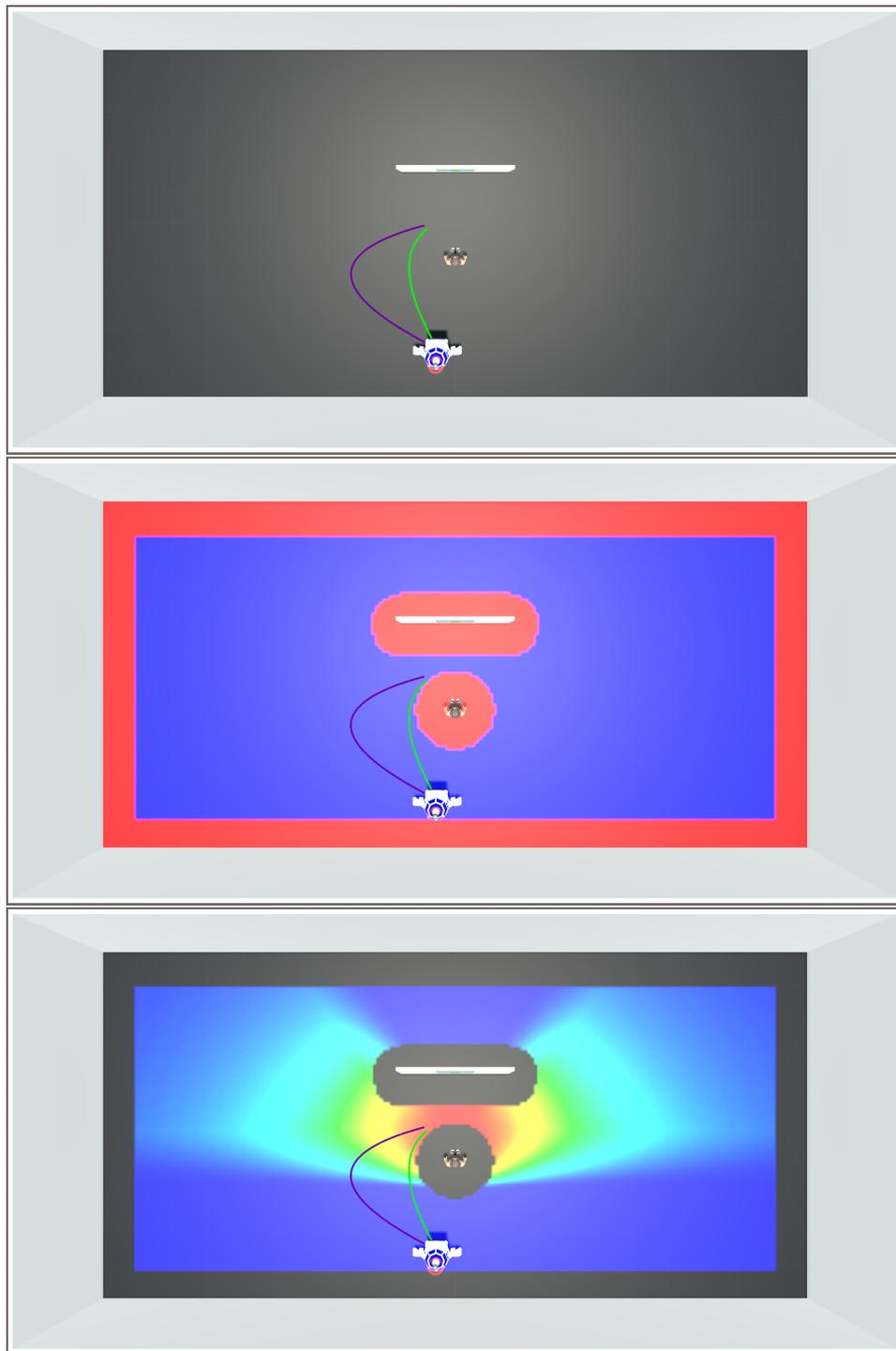


Figure 5.17: Path obtained by our path planner with (shown in purple) and without (shown in green) optimizing the path surprise for the scenario represented in figure 5.11b

Non-surprising motion: another example

Here, we illustrate the importance of optimizing the path surprise on another scenario.

Description We consider the scenario depicted in figure 5.18. The saliences of the painting and the robot are given in table 5.4. The person attends to the painting (top-down focus of attention). The task of the robot is to convey its intention to interact.

Table 5.4: Object and robot bottom-up salience.

Frog painting	Robot
3	2

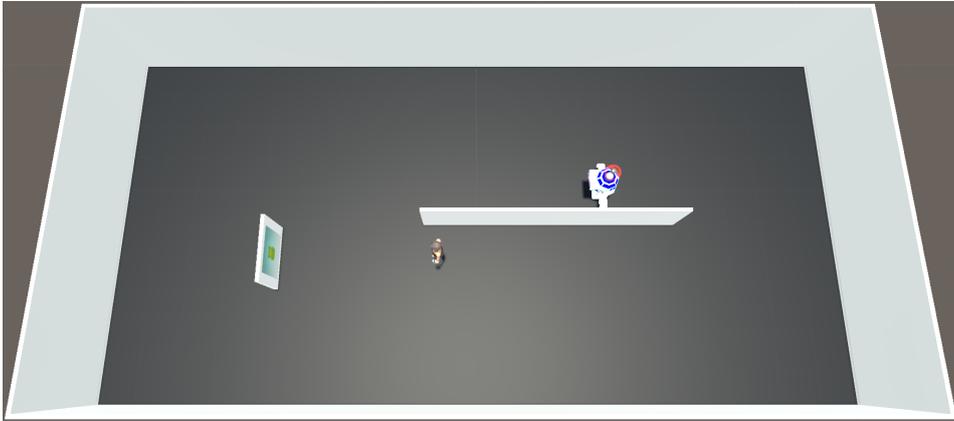


Figure 5.18: A person involved in an activity (watching the painting). The robot’s task is to navigate to an optimal end-pose that conveys its intention to interact.

Method Similarly to the previous case study, the path planner simultaneously attempts to optimize four objectives: path feasibility, path length, path reach and path surprise. We set a prior knowledge for both the path reach and the path surprise with $\sigma_{reach} = 0.01$ and $\sigma_{surprise} = 0.001$.

Results Figure 5.19 shows a comparison between the path obtained with and without optimizing for the path surprise. Both paths have 2 intermediary control points.

Discussion As in the previous case study, this scenario illustrates the importance of optimizing the path surprise. It also illustrates the capacity of our path planner to successfully concurrently handle four objectives to be optimized. The resulting path enters the field of view far away in order to minimize the effect of surprise. In this

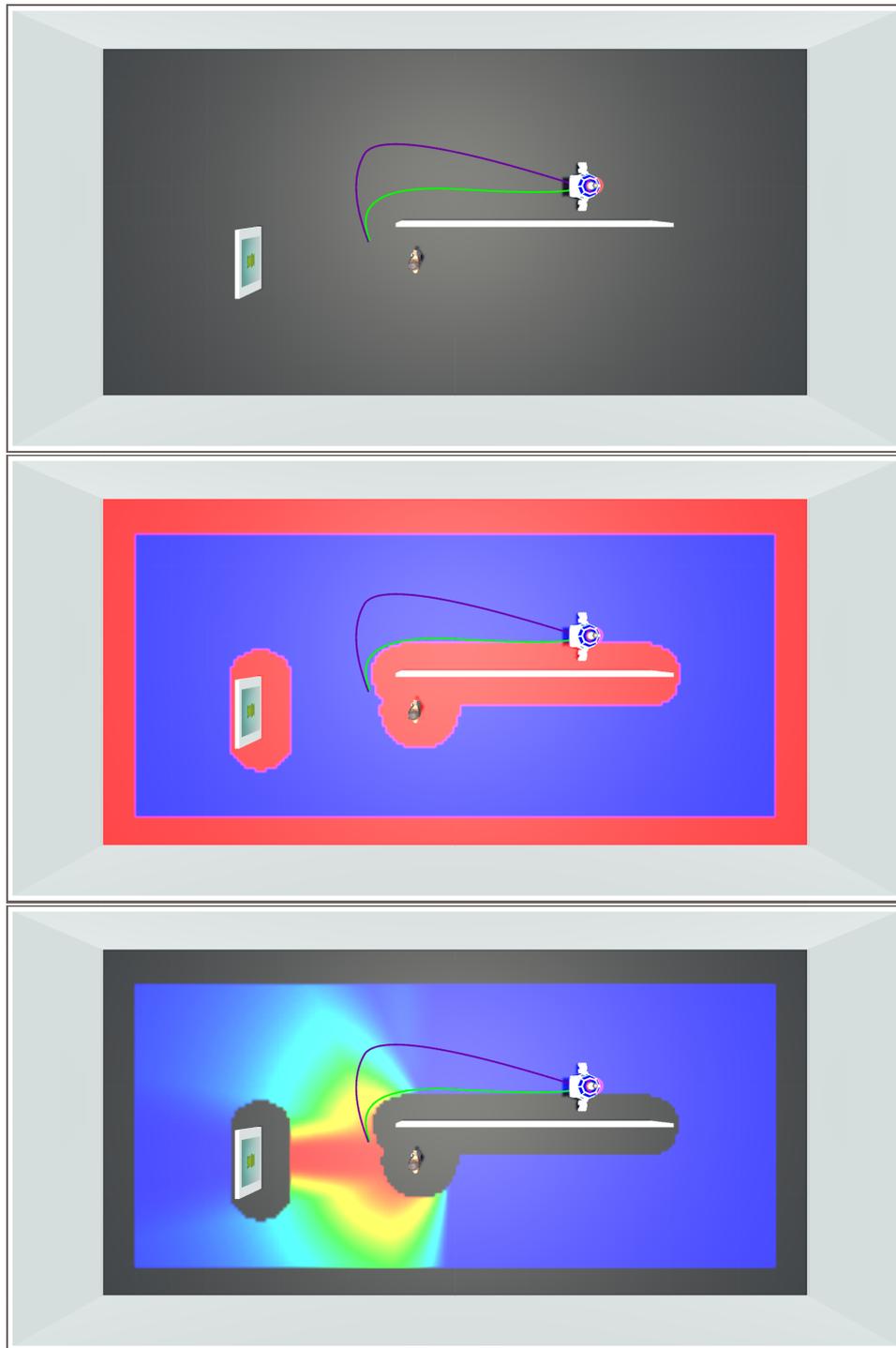


Figure 5.19: Path obtained by our path planner with (shown in purple) and without (shown in green) optimizing the path surprise.

scenario, we can additionally see that the distraction field extends the notion of activity space as defined by Scandolo and Fraichard [23].

5.5 Conclusion

In this chapter, we presented a novel path planner based on AGE, the many-objective evolutionary algorithm discussed in chapter 4. In order to yield socially acceptable motions, we defined five objectives to be relied upon by our path planner. Path safety and path efficiency are always considered by respectively minimizing a measure of the path feasibility and the path length. People’s comfort is taken into account using the concept of attention field introduced in chapter 3 to control the level of attentional resources allocated to the robot during its motion. Non-distracting motions are achieved by minimizing the attentional resources allocated to the robot along its path, i.e. avoiding high values of attention fields. In the context of human-robot interaction, the robot’s desire to interact is expressed in terms of reaching a pose that maximizes the amount of attentional resources allocated to it. Finally, the robot should always avoid causing surprise to the surrounding persons during its motion, for instance by avoiding to pass too closely behind obstacles and entering the persons’ fields of view sufficiently far away. For that purpose, we defined the surprise in terms of local variations of the attention field along the robot’s path. In this work, we only considered the influence of the robot’s position on the comfort of surrounding persons, i.e. for example the robot’s orientation and speed have not been taken into account in the attention field.

We illustrated the capacities of our path planner in several case studies. We have shown that our path planner is capable to simultaneously handle four objectives, e.g. path safety, path efficiency, path distraction and path surprise to achieve non-distracting motions. Our path planner finds paths that offer a good compromise between the aforementioned objectives. In the context of human-robot interaction, path distraction is no longer a quantity to minimize but instead our motion planner concurrently handles appropriate pose selection that maximizes the attentional resources allocated to the robot at its end-pose.

In summary, the results obtained by our path planner clearly qualitatively show the benefits of our approach. Although the obtained results are promising, this study deserves further attention. Firstly, in order to validate the usefulness of our approach, it is necessary to use salience computation techniques to estimate the salience of the objects and persons in the environment. As this is a difficult problem, it is likely that such experiments would first be carried out in the context of smart environments where saliences can be inferred from observing people’s gaze patterns. Secondly, although the chosen algorithm proved successful in obtaining satisfying paths, convergence has not been properly verified. Also, as suggested in section 4.5, we expect the use of Pythagorean hodograph curves to further improve the convergence of the multi-optimization process. Minimizing the path curvature is also important in order to avoid excitation of the robot dynamics and cause slippage, and further research will include the use of path smoothness as a secondary objective [105]. Thirdly, the superiority of the obtained paths in terms of human comfort still needs to be rigorously validated by performing user studies and comparing

our approach with current state-of-the-art socially-aware path planners. Fourthly, although we have presented an application architecture for many-objective path planning, we lacked the time to fully implement this architecture, and the current performances of the path planner are far too low (i.e. it takes about 3 minutes to compute each path) to be used for real-time path planning. Although the complete implementation of the proposed architecture is expected to greatly improve the performances of the path planning process, we expect further adjustments to the optimization algorithm (see 4.5) will be needed. Similarly, the current approach for computing attention field values clearly needs to be revisited as the current method is very slow. The development of a (near) real-time path planner will make it possible to investigate the potential of our approach in dynamic environments, where people constantly adjust their distribution of attentional resources in response to changes in their environment, i.e. the gaze direction and intention of the persons are expected to vary throughout the robot's motion. For instance, we expect that our attention-based approach might be able to successfully reproduce typical navigation behavior when joining a group of people. Fifthly the importance of considering avoidance of the personal space as an additional objective will be the topic of further research. Lastly, we plan to investigate how the audio modality should be taken into account in the computation of the attention field.

6 Conclusion

Mobile robot companions have to comply with social rules if they want to successfully share our environment; i.e. while remaining safe and efficient, the motion must be deemed appropriate from a human point of view. Socially-aware motion planners have so far heavily relied on the concept of *social spaces*. However, human-aware motion planning is still an active research field as designing socially-aware motions in arbitrarily complex environments remains a challenge, especially in the context of human-robot interaction where social spaces are of limited use.

In this work, we explored how the concept of attention can be taken into account to model how the influence of the environment as a whole affects how the robot’s motion is perceived by surrounding people. For this, we introduced a new computational model of attention that describes the full attentional state of the persons, i.e. estimates how the attention of a given person is distributed among the persons and objects of his or her environment. Based on this model, we introduced the novel concept of *attention field* that can be relied upon to design socially appropriate paths, even in complex environments. The attention field can naturally incorporate all the relevant characteristics of the robot’s motion such as position, orientation and velocity that may affect the comfort of surrounding persons. The attention field can be used to produce non-distracting and non-surprising paths, but also paths that convey the robot’s intention to interact in the context of human-robot interaction.

In order to show the benefits of the proposed approach, we built a many-objective path planner that naturally takes into account path safety, path efficiency as well as the surrounding people’s comfort through the use of the attention field, in order to produce socially acceptable paths. We described an application architecture for our path planner in view of performing real-time path planning. We subsequently illustrated the capacities of our path planner in several case studies.

The obtained results demonstrate that our path planner can naturally deal with complex situations and is able to find promising paths that are safe, efficient and comfortable, i.e. cause little distraction and surprise. More specifically, our path planner simultaneously optimizes path safety, efficiency (in terms of path length) and comfort (in terms of distraction and surprise), and yields good compromises between comfort and efficiency. In the context of human-robot interaction we additionally show how our path planner can simultaneously select an appropriate goal position that optimally conveys its intention

to interact with a person and optimize motion safety, efficiency and comfort.

We wish to emphasize that the results obtained in this study simply represent a proof of concept of the proposed approach. The full capacities of our approach are still to be explored and additional work should include: taking into account the robot's orientation and velocity in the attention field, studying the importance of additionally considering the personal space, modeling auditory saliences, and more generally studying the behavior of our approach in more diverse scenarios; for instance, we expect that such an attention-based approach might be able to successfully reproduce typical navigation behavior when joining a group of people in conversation. Beside the aforementioned suggestions, three main research directions emerge in order to further our study. Firstly, our path planner has been fully implemented but in a simplified version of the proposed software architecture; current performances need to be considerably increased in order to achieve real-time path planning. Although the complete implementation of the proposed architecture is expected to greatly improve the performances of the multi-objective optimization process, we expect further adjustment to the optimization algorithm will be needed in order to achieve real-time performances. This would then make it possible to investigate the potential of our approach when applied to dynamic case studies. Secondly, the computational model of attention presented in this study deserves further consideration: although theoretically grounded, it has not been experimentally validated. Also the saliences of the objects in the environment need to be properly estimated, most likely using learning techniques and in the context of a smart home environment. Thirdly, the superiority of such socially-aware paths, in terms of human comfort, still needs to be rigorously validated by performing user studies and comparing our approach with current state-of-the-art path planners.

In summary, in this work, we successfully took into account the notion of attention in order to produce socially acceptable paths in the presence of humans. We established the basis of such an approach, introducing a computational model of attention, suggesting how it can be used in the context of path planning, and built a many-objective socially-aware path planner based on the novel concept of attention field, before presenting promising results obtained with our approach. It is however still a work in progress, and throughout this study, we made suggestions on how to further this research and improve upon the different components of our path planner in order to achieve a robust, real-time social path planner. Exploring the potential of our approach in dynamic environments is expected to yield good results.

Bibliography

- [1] Hayley Robinson, Bruce MacDonald, and Elizabeth Broadbent. “The role of healthcare robots for older people at home: A review”. In: *International Journal of Social Robotics* 6.4 (2014), pp. 575–591 (cit. on p. 1).
- [2] Thibault Kruse et al. “Human-aware robot navigation: A survey”. In: *Robotics and Autonomous Systems* 61.12 (2013), pp. 1726–1743 (cit. on pp. 1, 7, 8).
- [3] Wolfram Burgard et al. “The interactive museum tour-guide robot”. In: *AAAI/I-AAI*. 1998, pp. 11–18 (cit. on pp. 2, 3).
- [4] Philipp Althaus et al. “Navigation for human-robot interaction tasks”. In: *Robotics and Automation, 2004. Proceedings. ICRA’04. 2004 IEEE International Conference on*. Vol. 2. IEEE. 2004, pp. 1894–1900 (cit. on p. 2).
- [5] Felix Lindner and Carola Eschenbach. “Towards a formalization of social spaces for socially aware robots”. In: *International Conference on Spatial Information Theory*. Springer. 2011, pp. 283–303 (cit. on pp. 2, 6, 61).
- [6] Edward T Hall. “A system for the notation of proxemic behavior”. In: *American anthropologist* 65.5 (1963), pp. 1003–1026 (cit. on p. 2).
- [7] Adam Kendon. “Spacing and orientation in co-present interaction”. In: *Development of Multimodal Interfaces: Active Listening and Synchrony*. Springer, 2010, pp. 1–15 (cit. on pp. 2, 5).
- [8] Thierry Fraichard, Remi Paulin, and Patrick Reignier. “Human-Robot Motion: An attention-based navigation approach”. In: *Robot and Human Interactive Communication, 2014 RO-MAN: The 23rd IEEE International Symposium on*. IEEE. 2014, pp. 684–691 (cit. on p. 2).
- [9] Sebastian Thrun et al. “MINERVA: A second-generation museum tour-guide robot”. In: *Robotics and automation, 1999. Proceedings. 1999 IEEE international conference on*. Vol. 3. IEEE. 1999 (cit. on p. 3).
- [10] Wolfram Burgard et al. “Experiences with an interactive museum tour-guide robot”. In: *Artificial intelligence* 114.1-2 (1999), pp. 3–55 (cit. on p. 3).
- [11] Erwin Prassler, Jens Scholz, and Paolo Fiorini. “Navigating a robotic wheelchair in a railway station during rush hour”. In: *The international journal of robotics research* 18.7 (1999), pp. 711–727 (cit. on p. 3).

BIBLIOGRAPHY

- [12] Illah R Nourbakhsh et al. “An affective mobile robot educator with a full-time job”. In: *Artificial Intelligence* 114.1 (1999), pp. 95–124 (cit. on p. 3).
- [13] Roland Siegwart et al. “Design, implementation and exploitation of a new fully autonomous tour guide robot”. In: *None*. LSA-CONF-2003-023. 2003 (cit. on p. 3).
- [14] Kai O Arras, Nicola Tomatis, and Roland Siegwart. “Robox, a remarkable mobile robot for the real world”. In: *Experimental Robotics VIII*. Springer, 2003, pp. 178–187 (cit. on p. 3).
- [15] Aurélie Clodic et al. “Rackham: An interactive robot-guide”. In: *Robot and Human Interactive Communication, 2006. ROMAN 2006. The 15th IEEE International Symposium on*. IEEE. 2006, pp. 502–509 (cit. on p. 3).
- [16] Sebastian Thrun et al. “Probabilistic algorithms and the interactive museum tour-guide robot minerva”. In: *The International Journal of Robotics Research* 19.11 (2000), pp. 972–999 (cit. on p. 3).
- [17] Illah R Nourbakhsh, Clayton Kunz, and Thomas Willeke. “The mobot museum robot installations: A five year experiment”. In: *Intelligent Robots and Systems, 2003.(IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*. Vol. 4. IEEE. 2003, pp. 3636–3641 (cit. on p. 3).
- [18] Irene Macaluso et al. “Experiences with CiceRobot, a museum guide cognitive robot”. In: *AI* IA 2005: Advances in Artificial Intelligence* (2005), pp. 474–482 (cit. on p. 3).
- [19] Kuo-Chen Huang, Jiun-Yi Li, and Li-Chen Fu. “Human-oriented navigation for service providing in home environment”. In: *SICE Annual Conference 2010, Proceedings of*. IEEE. 2010, pp. 1892–1897 (cit. on p. 4).
- [20] Chi-Pang Lam et al. “Human-centered robot navigation—towards a harmoniously human-robot coexisting environment”. In: *IEEE Transactions on Robotics* 27.1 (2011), pp. 99–112 (cit. on p. 4).
- [21] Soren Tranberg Hansen et al. “Adaptive human aware navigation based on motion pattern analysis”. In: *Robot and Human Interactive Communication, 2009. RO-MAN 2009. The 18th IEEE International Symposium on*. IEEE. 2009, pp. 927–932 (cit. on p. 4).
- [22] Rachel Kirby, Reid Simmons, and Jodi Forlizzi. “Companion: A constraint-optimizing method for person-acceptable navigation”. In: *Robot and Human Interactive Communication, 2009. RO-MAN 2009. The 18th IEEE International Symposium on*. IEEE. 2009, pp. 607–612 (cit. on pp. 4, 5, 32).
- [23] Leonardo Scandolo and Thierry Fraichard. “An anthropomorphic navigation scheme for dynamic scenarios”. In: *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. IEEE. 2011, pp. 809–814 (cit. on pp. 4–6, 29, 32, 61, 75).

BIBLIOGRAPHY

- [24] Emrah Akin Sisbot et al. “A human aware mobile robot motion planner”. In: *IEEE Transactions on Robotics* 23.5 (2007), pp. 874–883 (cit. on pp. 4, 5, 7, 32).
- [25] Mikael Svenstrup, Thomas Bak, and Hans Jørgen Andersen. “Trajectory planning for robots in dynamic human environments”. In: *Intelligent robots and systems (IROS), 2010 IEEE/RSJ international conference on*. IEEE. 2010, pp. 4293–4298 (cit. on pp. 4, 5).
- [26] Edward Twitchell Hall. “The hidden dimension”. In: (1966) (cit. on pp. 4, 5).
- [27] Jens Kessler, Christof Schroeter, and Horst-Michael Gross. “Approaching a person in a socially acceptable manner using a fast marching planner”. In: *International Conference on Intelligent Robotics and Applications*. Springer. 2011, pp. 368–377 (cit. on p. 5).
- [28] Kerstin Dautenhahn et al. “How may I serve you?: a robot companion approaching a seated person in a helping context”. In: *Proceedings of the 1st ACM SIGCHI/SI-GART conference on Human-robot interaction*. ACM. 2006, pp. 172–179 (cit. on p. 5).
- [29] Mikael Svenstrup et al. “Pose estimation and adaptive robot behaviour for human-robot interaction”. In: *Robotics and Automation, 2009. ICRA’09. IEEE International Conference on*. IEEE. 2009, pp. 3571–3576 (cit. on p. 5).
- [30] S Tranberg Hansen et al. “Adaptive human aware navigation based on motion pattern analysis”. In: *Robot and Human Interactive Communication, 2009. RO-MAN 2009. The 18th IEEE International Symposium on*. IEEE. 2009, pp. 927–932 (cit. on p. 5).
- [31] Matthias Luber et al. “Socially-aware robot navigation: A learning approach”. In: *Intelligent robots and systems (IROS), 2012 IEEE/RSJ international conference on*. IEEE. 2012, pp. 902–907 (cit. on p. 5).
- [32] Michael G Efran and James A Cheyne. “Shared space: The co-operative control of spatial areas by two interacting individuals.” In: *Canadian Journal of Behavioural Science/Revue canadienne des sciences du comportement* 5.3 (1973), p. 201 (cit. on p. 5).
- [33] Eric S Knowles et al. “Group size and the extension of social space boundaries.” In: *Journal of Personality and Social Psychology* 33.5 (1976), p. 647 (cit. on p. 5).
- [34] Jorge Rios-Martinez, Anne Spalanzani, and Christian Laugier. “Understanding human interaction for probabilistic autonomous navigation using Risk-RRT approach”. In: *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*. IEEE. 2011, pp. 2014–2019 (cit. on p. 5).
- [35] Jorge Rios-Martinez et al. “Navigating between people: a stochastic optimization approach”. In: *Robotics and automation (ICRA), 2012 IEEE international conference on*. IEEE. 2012, pp. 2880–2885 (cit. on pp. 5, 7).

BIBLIOGRAPHY

- [36] Jorge Rios-Martinez, Anne Spalanzani, and Christian Laugier. “From proxemics theory to socially-aware navigation: a survey”. In: *International Journal of Social Robotics* 7.2 (2015), pp. 137–153 (cit. on p. 6).
- [37] James J Gibson. “The theory of affordances”. In: *GIESEKING, Jen Jack; MANGOLD, William; KATZ, Cindi; LOW, Setha* (2014), pp. 56–60 (cit. on p. 6).
- [38] Gian Diego and Tipaldi Kai O Arras. “Please do not disturb! minimum interference coverage for social robots”. In: *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*. IEEE. 2011, pp. 1968–1973 (cit. on p. 6).
- [39] Stephan Sehestedt, Sarath Kodagoda, and Gamini Dissanayake. “Robot path planning in a social context”. In: *Robotics Automation and Mechatronics (RAM), 2010 IEEE Conference on*. IEEE. 2010, pp. 206–211 (cit. on p. 7).
- [40] Eric Martinson. “Hiding the acoustic signature of a mobile robot”. In: *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*. IEEE. 2007, pp. 985–990 (cit. on p. 7).
- [41] Kay Kitazawa and Taku Fujiyama. “Pedestrian vision and collision avoidance behavior: Investigation of the information process space of pedestrians using an eye tracker”. In: *Pedestrian and evacuation dynamics 2008*. Springer, 2010, pp. 95–108 (cit. on p. 7).
- [42] Michael L Walters et al. “A long-term human-robot proxemic study”. In: *RO-MAN, 2011 IEEE*. IEEE. 2011, pp. 137–142 (cit. on p. 8).
- [43] Leila Takayama and Caroline Pantofaru. “Influences on proxemic behaviors in human-robot interaction”. In: *Intelligent robots and systems, 2009. IROS 2009. IEEE/RSJ international conference on*. IEEE. 2009, pp. 5495–5502 (cit. on p. 8).
- [44] Jonathan Mumm and Bilge Mutlu. “Human-robot proxemics: physical and psychological distancing in human-robot interaction”. In: *Proceedings of the 6th international conference on Human-robot interaction*. ACM. 2011, pp. 331–338 (cit. on p. 8).
- [45] Jeremy M Wolfe and Todd S Horowitz. “What attributes guide the deployment of visual attention and how do they do it?” In: *Nature reviews neuroscience* 5.6 (2004), pp. 495–501 (cit. on p. 10).
- [46] Laurent Itti and Christof Koch. “Computational modelling of visual attention”. In: *Nature reviews. Neuroscience* 2.3 (2001), p. 194 (cit. on pp. 10, 11, 13, 14, 22).
- [47] Brian J Scholl. “Objects and attention: The state of the art”. In: *Cognition* 80.1 (2001), pp. 1–46 (cit. on pp. 10, 12, 13, 17, 18).
- [48] Robert Desimone and John Duncan. “Neural mechanisms of selective visual attention”. In: *Annual review of neuroscience* 18.1 (1995), pp. 193–222 (cit. on p. 11).

BIBLIOGRAPHY

- [49] Derrick Parkhurst, Kinton Law, and Ernst Niebur. “Modeling the role of salience in the allocation of overt visual attention”. In: *Vision research* 42.1 (2002), pp. 107–123 (cit. on p. 11).
- [50] Farhan Baluch and Laurent Itti. “Mechanisms of top-down attention”. In: *Trends in neurosciences* 34.4 (2011), pp. 210–224 (cit. on pp. 12, 14).
- [51] Irvin Rock and Daniel Gutman. “The effect of inattention on form perception.” In: *Journal of Experimental Psychology: Human Perception and Performance* 7.2 (1981), p. 275 (cit. on p. 12).
- [52] John Duncan. “Selective attention and the organization of visual information.” In: *Journal of Experimental Psychology: General* 113.4 (1984), p. 501 (cit. on p. 12).
- [53] Michael I Posner. “Orienting of attention”. In: *Quarterly journal of experimental psychology* 32.1 (1980), pp. 3–25 (cit. on pp. 12, 17).
- [54] Anne Treisman. “Features and objects: The fourteenth Bartlett memorial lecture”. In: *The quarterly journal of experimental psychology* 40.2 (1988), pp. 201–237 (cit. on p. 12).
- [55] Morris Goldsmith. “What’s in a location? Comparing object-based and space-based models of feature integration in visual search.” In: *Journal of Experimental Psychology: General* 127.2 (1998), p. 189 (cit. on p. 12).
- [56] Zijiang J He and Ken Nakayama. “Visual attention to surfaces in three-dimensional space”. In: *Proceedings of the National Academy of Sciences* 92.24 (1995), pp. 11155–11159 (cit. on p. 12).
- [57] Kathleen M O’craven, Paul E Downing, and Nancy Kanwisher. “fMRI evidence for objects as the units of attentional selection”. In: *Nature* 401.6753 (1999), pp. 584–587 (cit. on p. 12).
- [58] Daniel Kahneman, Avishai Henik, et al. “Perceptual organization and attention”. In: *Perceptual organization* 1 (1981), pp. 181–211 (cit. on p. 12).
- [59] Robert Egly, Jon Driver, and Robert D Rafal. “Shifting visual attention between objects and locations: evidence from normal and parietal lesion subjects.” In: *Journal of Experimental Psychology: General* 123.2 (1994), p. 161 (cit. on p. 13).
- [60] Christof Koch and Shimon Ullman. “Shifts in selective visual attention: towards the underlying neural circuitry”. In: *Matters of intelligence*. Springer, 1987, pp. 115–141 (cit. on p. 13).
- [61] Vidhya Navalpakkam and Laurent Itti. “An integrated model of top-down and bottom-up attention for optimizing detection speed”. In: *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’06)*. Vol. 2. IEEE. 2006, pp. 2049–2056 (cit. on pp. 15–17).
- [62] Laurent Itti, Christof Koch, and Ernst Niebur. “A model of saliency-based visual attention for rapid scene analysis”. In: *IEEE Transactions on pattern analysis and machine intelligence* 20.11 (1998), pp. 1254–1259 (cit. on p. 16).

BIBLIOGRAPHY

- [63] Laurent Itti and Christof Koch. “A saliency-based search mechanism for overt and covert shifts of visual attention”. In: *Vision research* 40.10 (2000), pp. 1489–1506 (cit. on p. 16).
- [64] Robert J Peters and Laurent Itti. “Beyond bottom-up: Incorporating task-dependent influences into a computational model of spatial attention”. In: *2007 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE. 2007, pp. 1–8 (cit. on pp. 16, 17).
- [65] John H Reynolds and David J Heeger. “The normalization model of attention”. In: *Neuron* 61.2 (2009), pp. 168–185 (cit. on pp. 16–18, 20).
- [66] David J Heeger. “Normalization of cell responses in cat striate cortex”. In: *Visual neuroscience* 9.02 (1992), pp. 181–197 (cit. on p. 16).
- [67] Jérôme Maisonasse et al. “Attentional model for perceiving social context in intelligent environments”. In: *IFIP International Conference on Artificial Intelligence Applications and Innovations*. Springer. 2006, pp. 171–178 (cit. on p. 17).
- [68] Charles W Eriksen and Yei-yu Yeh. “Allocation of attention in the visual field.” In: *Journal of Experimental Psychology: Human Perception and Performance* 11.5 (1985), p. 583 (cit. on p. 17).
- [69] Charles W Eriksen and James D St James. “Visual attention within and around the field of focal attention: A zoom lens model”. In: *Perception & psychophysics* 40.4 (1986), pp. 225–240 (cit. on p. 17).
- [70] Patrick Cavanagh and George A Alvarez. “Tracking multiple targets with multifocal attention”. In: *Trends in cognitive sciences* 9.7 (2005), pp. 349–354 (cit. on p. 20).
- [71] Stephanie A McMains and David C Somers. “Multiple spotlights of attentional selection in human visual cortex”. In: *Neuron* 42.4 (2004), pp. 677–686 (cit. on p. 20).
- [72] Anne M Treisman and Garry Gelade. “A feature-integration theory of attention”. In: *Cognitive psychology* 12.1 (1980), pp. 97–136 (cit. on p. 23).
- [73] Harold Pashler. “Target-distractor discriminability in visual search”. In: *Attention, Perception, & Psychophysics* 41.4 (1987), pp. 285–292 (cit. on p. 23).
- [74] John Duncan and Glyn W Humphreys. “Visual search and stimulus similarity.” In: *Psychological review* 96.3 (1989), p. 433 (cit. on p. 23).
- [75] Allen L Nagy and Robert R Sanchez. “Critical color differences determined with a visual search task”. In: *JOSA A* 7.7 (1990), pp. 1209–1217 (cit. on p. 23).
- [76] Anne Treisman. “Search, similarity, and integration of features between and within dimensions.” In: *Journal of Experimental Psychology: Human Perception and Performance* 17.3 (1991), p. 652 (cit. on p. 23).
- [77] Jeremy M Wolfe. “Guided search 2.0 a revised model of visual search”. In: *Psychonomic bulletin & review* 1.2 (1994), pp. 202–238 (cit. on p. 23).

BIBLIOGRAPHY

- [78] Chankong Vira and Yacov Y Haimes. *Multiobjective decision making: theory and methodology*. 8. North-Holland, 1983 (cit. on p. 34).
- [79] Karl Bringmann et al. “Approximation-guided evolutionary multi-objective optimization”. In: *IJCAI Proceedings-International Joint Conference on Artificial Intelligence*. Vol. 22. 1. 2011, p. 1198 (cit. on p. 35).
- [80] Michel Gendreau and Jean-Yves Potvin. *Handbook of metaheuristics*. Vol. 2. Springer, 2010 (cit. on p. 35).
- [81] Kalyanmoy Deb. *Multi-objective optimization using evolutionary algorithms*. Vol. 16. John Wiley & Sons, 2001 (cit. on p. 35).
- [82] Carlos A Coello Coello, Gary B Lamont, David A Van Veldhuizen, et al. *Evolutionary algorithms for solving multi-objective problems*. Vol. 5. Springer, 2007 (cit. on p. 35).
- [83] Carlos M Fonseca and Peter J Fleming. “An overview of evolutionary algorithms in multiobjective optimization”. In: *Evolutionary computation* 3.1 (1995), pp. 1–16 (cit. on p. 35).
- [84] J David Schaffer. “Multiple objective optimization with vector evaluated genetic algorithms”. In: *Proceedings of the 1st international Conference on Genetic Algorithms*. L. Erlbaum Associates Inc. 1985, pp. 93–100 (cit. on p. 36).
- [85] Eckart Zitzler and Lothar Thiele. “Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach”. In: *IEEE transactions on Evolutionary Computation* 3.4 (1999), pp. 257–271 (cit. on p. 36).
- [86] David E Goldberg. “Genetic algorithms in search, optimization, and machine learning”. In: *Addion wesley* 1989 (1989), p. 102 (cit. on p. 36).
- [87] David E Goldberg, Jon Richardson, et al. “Genetic algorithms with sharing for multimodal function optimization”. In: *Genetic algorithms and their applications: Proceedings of the Second International Conference on Genetic Algorithms*. Hillsdale, NJ: Lawrence Erlbaum. 1987, pp. 41–49 (cit. on p. 37).
- [88] Bruno Sareni and Laurent Krahenbuhl. “Fitness sharing and niching methods revisited”. In: *IEEE Transactions on Evolutionary computation* 2.3 (1998), pp. 97–106 (cit. on p. 37).
- [89] Kalyanmoy Deb et al. “A fast and elitist multiobjective genetic algorithm: NSGA-II”. In: *IEEE transactions on evolutionary computation* 6.2 (2002), pp. 182–197 (cit. on pp. 37, 39, 43).
- [90] Eckart Zitzler, M Laumanns, and Lothar Thiele. “Spea2: Improving the strength pareto evolutionary algorithm for multiobjective optimization”. In: *Evolutionary Methods for Design, Optimization, and Control* (2002), pp. 95–100 (cit. on p. 37).
- [91] Joshua D Knowles and David W Corne. “Approximating the nondominated front using the Pareto archived evolution strategy”. In: *Evolutionary computation* 8.2 (2000), pp. 149–172 (cit. on p. 37).

BIBLIOGRAPHY

- [92] Marco Laumanns, Eckart Zitzler, and Lothar Thiele. “A unified model for multi-objective evolutionary algorithms with elitism”. In: *Evolutionary Computation, 2000. Proceedings of the 2000 Congress on*. Vol. 1. IEEE. 2000, pp. 46–53 (cit. on p. 37).
- [93] Marco Laumanns et al. “Combining convergence and diversity in evolutionary multiobjective optimization”. In: *Evolutionary computation* 10.3 (2002), pp. 263–282 (cit. on p. 37).
- [94] Kalyanmoy Deb and Mohamed Abouhawwash. “An optimality theory based proximity measure for set based multi-objective optimization”. In: (2012) (cit. on p. 37).
- [95] Eckart Zitzler. “Evolutionary algorithms for multiobjective optimization: Methods and applications”. In: (1999) (cit. on p. 37).
- [96] Mark Fleischer. “The measure of Pareto optima applications to multi-objective metaheuristics”. In: *International Conference on Evolutionary Multi-Criterion Optimization*. Springer. 2003, pp. 519–533 (cit. on p. 37).
- [97] Joshua Knowles and David Corne. “Properties of an adaptive archiving algorithm for storing nondominated vectors”. In: *IEEE Transactions on Evolutionary Computation* 7.2 (2003), pp. 100–116 (cit. on p. 38).
- [98] Lyndon While et al. “A faster algorithm for calculating hypervolume”. In: *IEEE transactions on evolutionary computation* 10.1 (2006), pp. 29–38 (cit. on p. 38).
- [99] Hisao Ishibuchi, Noritaka Tsukamoto, and Yusuke Nojima. “Evolutionary many-objective optimization: A short review.” In: *IEEE congress on evolutionary computation*. 2008, pp. 2419–2426 (cit. on p. 38).
- [100] Robin C Purshouse and Peter J Fleming. “On the evolutionary optimization of many conflicting objectives”. In: *IEEE Transactions on Evolutionary Computation* 11.6 (2007), pp. 770–784 (cit. on p. 38).
- [101] Salem F Adra and Peter J Fleming. “Diversity management in evolutionary many-objective optimization”. In: *IEEE Transactions on Evolutionary Computation* 15.2 (2011), pp. 183–195 (cit. on p. 38).
- [102] Yong K Hwang and Narendra Ahuja. “Gross motion planning—a survey”. In: *ACM Computing Surveys (CSUR)* 24.3 (1992), pp. 219–291 (cit. on p. 39).
- [103] Steven M LaValle. *Planning algorithms*. Cambridge university press, 2006 (cit. on p. 39).
- [104] Faez Ahmed and Kalyanmoy Deb. “Multi-objective path planning using spline representation”. In: *Robotics and Biomimetics (ROBIO), 2011 IEEE International Conference on*. IEEE. 2011, pp. 1047–1052 (cit. on pp. 39–41, 45, 46).
- [105] Faez Ahmed and Kalyanmoy Deb. “Multi-objective optimal path planning using elitist non-dominated sorting genetic algorithms”. In: *Soft Computing* 17.7 (2013), pp. 1283–1299 (cit. on pp. 39, 40, 76).

BIBLIOGRAPHY

- [106] Jin Yuan et al. “Step-spreading map knowledge based multi-objective genetic algorithm for robot-path planning”. In: *2007 IEEE International Conference on Systems, Man and Cybernetics*. IEEE. 2007, pp. 3402–3407 (cit. on pp. 39, 40).
- [107] Jiun-Hau Wei and Jing-Sin Liu. “Collision-free composite? 3-splines generation for nonholonomic mobile robots by parallel variable-length genetic algorithm”. In: *Computational Intelligence for Modelling Control & Automation, 2008 International Conference on*. IEEE. 2008, pp. 545–550 (cit. on pp. 39, 40).
- [108] Jiun-Hau Wei and Jing-Sin Liu. “Generating minimax-curvature and shorter η 3-spline path using multi-objective variable-length genetic algorithm”. In: *Networking, Sensing and Control (ICNSC), 2010 International Conference on*. IEEE. 2010, pp. 319–324 (cit. on pp. 39, 40).
- [109] Han-Chih Chang. “High-quality path planning for autonomous mobile robots with η 3-splines and parallel genetic algorithms”. In: *Robotics and Biomimetics, 2008. ROBIO 2008. IEEE International Conference on*. IEEE. 2009, pp. 1671–1677 (cit. on pp. 39–41).
- [110] Kenneth A De Jong. *Evolutionary computation: a unified approach*. MIT press, 2006 (cit. on p. 39).
- [111] O Castilho and L Trujilo. “Multiple Objective Optimization Genetic Algorithms For Path Planning In Autonomous Mobile Robots.” In: *Int. J. Comput. Syst. Signal* 6.1 (2005), pp. 48–63 (cit. on pp. 39, 40).
- [112] Oscar Castillo, Leonardo Trujillo, and Patricia Melin. “Multiple objective genetic algorithms for path-planning optimization in autonomous mobile robots”. In: *Soft Computing* 11.3 (2007), pp. 269–279 (cit. on pp. 39, 40).
- [113] Alejandro Hidalgo-Paniagua et al. “Solving the multi-objective path planning problem in mobile robotics with a firefly-based approach”. In: *Soft Computing* (2015), pp. 1–16 (cit. on pp. 39, 40).
- [114] Alejandro Hidalgo-Paniagua et al. “MOSFLA-MRPP: multi-objective shuffled frog-leaping algorithm applied to mobile robot path planning”. In: *Engineering Applications of Artificial Intelligence* 44 (2015), pp. 123–136 (cit. on pp. 39, 40).
- [115] Alejandro Hidalgo-Paniagua, Miguel A Vega-Rodríguez, and Joaquín Ferruz. “Applying the MOVNS (multi-objective variable neighborhood search) algorithm to solve the path planning problem in mobile robotics”. In: *Expert Systems with Applications* 58 (2016), pp. 20–35 (cit. on pp. 39, 40).
- [116] Hu Jun and Zhu Qingbao. “Multi-objective mobile robot path planning based on improved genetic algorithm”. In: *Intelligent Computation Technology and Automation (ICICTA), 2010 International Conference on*. Vol. 2. IEEE. 2010, pp. 752–756 (cit. on pp. 39, 40).
- [117] Eric W. Weisstein. “Bézier Curve.” *From MathWorld—A Wolfram Web Resource*. URL: <http://mathworld.wolfram.com/BezierCurve.html> (cit. on p. 41).

BIBLIOGRAPHY

- [118] Markus Wagner et al. “Efficient optimization of many objectives by approximation-guided evolution”. In: *European Journal of Operational Research* 243.2 (2015), pp. 465–479 (cit. on pp. 42, 43, 45).
- [119] Kalyanmoy Deb and Aravind Srinivasan. “Innovization: Innovating design principles through optimization”. In: *Proceedings of the 8th annual conference on Genetic and evolutionary computation*. ACM. 2006, pp. 1629–1636 (cit. on p. 43).
- [120] Abhinav Gaur and Kalyanmoy Deb. “Adaptive Use of Innovization Principles for a Faster Convergence of Evolutionary Multi-Objective Optimization Algorithms”. In: (2016) (cit. on p. 43).
- [121] Rida T Farouki and Takis Sakkalis. “Pythagorean hodographs”. In: *IBM Journal of Research and Development* 34.5 (1990), pp. 736–752 (cit. on p. 43).
- [122] Herman Bruyninckx and Dominiek Reynaerts. “Path planning for mobile and hyper-redundant robots using Pythagorean hodograph curves”. In: *Advanced Robotics, 1997. ICAR’97. Proceedings., 8th International Conference on*. IEEE. 1997, pp. 595–600 (cit. on p. 44).
- [123] John Connors and Gabriel Elkaim. “Manipulating B-Spline based paths for obstacle avoidance in autonomous ground vehicles”. In: *ION National Technical Meeting, ION NTM*. Citeseer. 2007, pp. 22–24 (cit. on p. 45).
- [124] Kalyanmoy Deb and Santosh Tiwari. “Omni-optimizer: A generic evolutionary algorithm for single and multi-objective optimization”. In: *European Journal of Operational Research* 185.3 (2008), pp. 1062–1087 (cit. on p. 45).
- [125] Andrzej Dworak et al. “Middleware trends and market leaders 2011”. In: *Conf. Proc.* Vol. 111010. CERN-ATS-2011-196. 2011, FRBHMULT05 (cit. on p. 53).
- [126] Nathan Koenig and Andrew Howard. “Design and use paradigms for gazebo, an open-source multi-robot simulator”. In: *Intelligent Robots and Systems, 2004. (IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*. Vol. 3. IEEE. 2004, pp. 2149–2154 (cit. on p. 57).
- [127] Eric Rohmer, Surya PN Singh, and Marc Freese. “V-REP: A versatile and scalable robot simulation framework”. In: *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*. IEEE. 2013, pp. 1321–1326 (cit. on p. 57).