This project has received funding from the European Unions Horizon 2020 research and innovation programme under grant agreement №820742

HR-Recycler: Hybrid Human-Robot RECYcling plant for electriCal and eLEctRonic equipment

D6.2 - Haptic Regression Report

WP number and title	WP6 - Haptic Regression Report
Lead Beneficiary	TUM
Contributor(s)	SDK
Deliverable type	Report
Planned delivery date	October 01, 2019
Last Update	March 25, 2020
Dissemination level	PU

Disclaimer

The HR-Recycler Consortium consists of the following partners:

No	Participant organisation name	Short	Type	Country
1	CENTRE FOR RESEARCH AND TECHNOLOGY HELLAS	CERTH	RTO	GR
2	FUNDACIO INSTITUT DE BIOENGINYERIA DE CATALUNYA	IBEC	RTO	ES
3	TECHNISCHE UNIVERSITAET MUENCHEN	TUM	RTO	DE
4	COMAU SPA	COMAU	IND	IT
5	FUNDACION TECNALIA RESEARCH & INNOVATION	TEC	RTO	ES
6	ROBOTNIK AUTOMATION SLL	ROB	SME	ES
7	FUNDACION GAIKER	GAIKER	RTO	ES
8	SADAKO TECHNOLOGIES SL	SDK	SME	ES
9	DIGINEXT	DXT	IND	BE
10	VRIJE UNIVERSITEIT BRUSSEL	VUB	RTO	BE
11	INDUMETAL RECYCLING, S.A.	IND	SME	ES
12	INTERECYCLING - SOCIEDADE DE RECICLAGEM	SA	INT	PT
13	BIANATT ANAKYKLOSI AIIE ANONIMI BIOMICHANIKI EMPORIKI ETAIRIA	BNTT	IND	GR

Document History

Version	Date	Status	Author / Reviewer	Description
0.1	13.09.2019	Draft	V. Gabler & G. Huber (TUM)	Initial draft
0.2	19.09.2019	Rev 1	M. Bosch, (ROB)	Internal Review
0.3	19.09.2019	Rev 1	D. Giakoumis, (CERTH)	Internal Review
0.4	24.09.2019	Rev 1	V. Gabler & G. Huber, (TUM)	Final Version
0.5	30.01.2020	Review	–	–
0.6	25.02.2020	Rev 2	V. Gabler (TUM)	Review Update

Acronyms

D	Deliverable.
DBSCAN	Density-Based-Spatial-Clustering for Applications with Noise.
DoF	Degrees of Freedom.
EKF	Extended Kalman Filters.
EV	Electric vehicle.
FEM	Finite Element Method.
FPD	Flat Panel Display.
FSR	Force-sensing resistors.
HRC	Human-Robot Collaboration.
HR-Recycler	Hybrid Human-Robot RECYcling plant for electriCal and eLEctRonic equipment.
KPI	Key Performance Indicator.
M	Month.
RANSAC	Random Sample Consenus.
SE	Sensor-equipped.
SLAM	Simultaneous Localisation and Mapping.
T	Task.
WEEE	Waste of Electrical and Electronic Equipment.
WP	Work Package.

Table of Contents

List of Figures

List of Tables

Executive Summary

The Hybrid Human-Robot RECYcling plant for electriCal and eLEctRonic equipment (HR-Recycler) project aims for a deployment of robotic systems in industrial disassembly sites. Unlike automated production lines where robot tasks can be configured in advance, disassembly in a recycling plant requires the robots to be robustly and dynamically handling a set of tasks from the observations about the objects and their components. Nevertheless, vision-based perceptual modules may face limitations, as visual inspection is often unsuitable for detecting material characteristics and inner fixtures such as screws, welding or adhesion. Thus, the HR-Recycler proposes new scientific concepts on robot-based haptic regression techniques for object disassembly. In order to fulfill the HR-Recycler use-case requirements defined under Deliverable (D)3.1: *User requirements and use cases*, this deliverable highlights how haptics can be exploited to supplement the weakness of the visual perception in terms of refinement of geometric shape estimations of unknown objects, as well as decomposition and regression of hidden materials and structures in the object. In order to relax the overall object identification problem, this delivery sketches the theoretic concept of a two-layered haptic identification framework. The first layer is outlining the shape refinement process that allows a robot to refine basic object shape estimations from haptic data. This module is motivated from the Bayesian Filter theory and thus allows a robot to actively decrease the uncertainty over perception of a given object while embedding knowledge from the vision-based perception system from Work Package (WP)5: *AI-enabled cell-level perception methods*. The output of this layer provides a set of shape candidates, which can then be used to regress further material parameters from the collected data in the second layer. The purpose of this deliverable is to outline the state-of-the-art of haptic data acquisition methods for subsequent implementation. As these active data acquisition methods are interdependent with the findings from Task (T)6.1: *Force-guided manipulation*. As a preliminary work towards implementation, this deliverable closes with description of the simulated and experimental results on finding fixture points in perfectly defined objects. Thus, this deliverable mainly serves as the foundation for D6.4: *Flexible probing* (due in Month (M)24) and D6.1: *Force-guided manipulation*(due in M20) which will conclude implementation and evaluation results of robot-based regression techniques for detecting fixture points within unknown objects, including flexible probing forming a special case of this identification process. The state of the art of haptic probing in the scope of case opening applications in industry and research is briefly summarized as

- Strong emphasis on screw detection and removal within robotic disassembly projects where humans and robots collaborate in close distance, e.g.
  - Visual fixture detection for electric motors

  - Visual servoing for screw removal on laptops
  - Robot assistance in HRC for disassembling Electric vehicle (EV) batteries
  - Case cutting of PC-monitor with adjustable energy bounds to overcome geometry uncertainties with destructive disassembly

- In the aspect of haptic object identification, a strong emphasis lies on shape refinements, e.g.
  - Refining object shape using grid based representation
  - Exploiting preknowledge of of underlying shape and refining shape uncertainty using haptic SLAM
  - Object differentiation using clustering methods or classification on tactile and kinesthetic features, e.g. texture or stiffness
  - Redefining object contour surfaces using sensor arrays
  - Combining proprioception and touch sensing for unknown shape recognition
- Research in Object Understanding via Flexible Probing mainly tackles the aspect of modeling, e.g.
  - Modeling needle tip deflections using Euler-Bernoulli beam theory
  - Modeling deflection estimation of a positioning machine comparing Euler-Bernoulli beam with Timoshenko Beam theory
  - Modeling large deflections for robotic manipulators and object shape estimations using the Kirchhoff elastic rod model

As outlined in detail in section 2.2, the contribution of the HR-Recycler in the context of the state of the art in industry and research is given as

- Extending methods from haptic rendering such as [9] that focus on object refinement by furthermore extracting construction information such as
  - material stiffness,
  - object elasticity based on current stiffness and shape estimation.
- Refining object knowledge such as the location of unknown fixtures via flexible probing:
  - iteratively apply forces onto the surface of the object,
  - compare the actual deflection with the estimated deflection by mapping the current shape and parameter estimation into a Finite Element Method (FEM) model.

1 Introduction

The HR-Recycler project aims to create a hybrid collaboration environment, where humans and robots will harmoniously share and undertake different processing and manipulation tasks at the same time, by targeting the industrial application case of Waste of Electrical and Electronic Equipment (WEEE) recycling. This deliverable is an intermediate theoretic report of T6.2: *Construction understanding through flexible probing* in WP6: *Robotic actions, planning and control* which overall implements the tools for navigation, manipulation, grasping and disassembly by the robots as well as safety control in robot planning. The conclusion of T6.2 including the implementation of the haptic regression methods and evaluation results will be reported in D6.4 in M24. This deliverable lays the foundation towards Key Performance Indicator (KPI) 4.3 that seeks to enable robots with the skill of "Understanding of structural properties through fusion of haptic probing with vision feedback". In particular, the methods outlined in this deliverable helps HR-Recycler "to be capable of opening 50% of all casings, where one fixture was visually not detected".

The disassembly of an unknown object must overcome various sources of uncertainty. On the one hand, the shape of objects can be approximated well from vision and depth sensors. On the other hand, the information may be incomplete as the visually occluded structures or the material composition that affects mechanical characteristics remains out of reach for such sensors. Thus, a robot shall exploit haptics to refine the knowledge over these objects. Haptic regression techniques entails depiction of components along a object; a clear identification as well as weather it may be further disassembled into smaller components.Thus, this process does not only include the identification of rigid body characteristics but also the interconnection of all individual components. The interconnections may be achieved by, for example, hidden fixtures such as screws, or non-reversible such as plastic snap-in fixtures. When disassembling devices without knowing their construction plans, fixtures are easily missed - either because they are not visible, such as snap-in fixtures, or because they are not properly recognized, such as recessed screws. In this context, we need to have a solution for understanding the object and identify the single components, such that the construction plan of unknown devices with shells and fixtures for disassembly can be obtained.

Thus, this deliverable tackles the challenge of exploring and identifying unknown objects from haptic data based on an uncertain prior from the vision data. The technique would allow a robot not only to iteratively refine the representation of the object but also to actively interact with the object to gain further knowledge that remains out of reach by a simple passive inspection. The regression procedure is intended to allow robots to dis-

assemble a project with a minimum amount of destruction, and thus increase the safety aspect that allows humans to work in the vicinity of the robot platform.

In this deliverable, we outline the current progress as well as shortcomings of the state-of-the-art in haptic regression in industry and academia. Having a closer look into these results, it becomes evident that none of these approaches is capable of closing the open gaps in the list of requirements of the HR-Recycler project which is reported in D3.1: *User requirements and use cases*. Given the fact that the general haptic regression problem is facing a variety of challenges when information beyond the geometric shape is required, we propose a framework that allows to decouple the haptic data acquisition into two relaxed sub-problems. While the first sub-problem tackles the geometric shape refinement process motivated from the Bayesian Filter theory that allows an incorporation of findings from the perception systems, the second forms an identification problem of the object parameterization. In other words, the former solves the active data acquisition and iterative belief update, whereas the latter tackles the haptic identification. The major assumption is that the shape refinement obtains a conditional probability over possible shape candidates given the current object parameterization. Analogously, the identification process is then run for each shape candidate under the assumption of each shape being correct. As a result, the two modules allow to update the current belief to eventually identify the object of interest.

This deliverable serves as a methodological basis for D6.4: *Flexible probing* (due in M24), which explicitly highlights the flexible probing as a scientific novelty within the HR-Recycler project, that in return forms a special case of the haptic regression module described above. Thus, the major emphasis is set on the basic tools and the embedding of this skill into the general HR-Recycler project. As the data acquisition is strongly emphasizing on the aspect of acquiring data samples that optimize the information gain, it is strongly interdependent with the manipulation techniques from T6.1: *Force-guided manipulation* (due in M26), the robotic hardware design from WP8: *Smart mechatronics hardware for efficient HRC*, as well as the results from WP5: *AI-enabled cell-level perception methods*. With this deliverable focusing on the methodological aspects of the haptic regression problem within the HR-Recycler, the evaluation of the data acquisition is restricted to a simulated environment, in which a realistic physics engine simulates the interaction of an articulated robot with an object. In contrast, the flexible probing is facing the haptic identification, which requires the post-evaluation of actual recordings. In this aspect, a major source of uncertainty is given by the unknown shape of an object, while the dependency on the actual robot platform can be relaxed. As a result, we decouple the analysis of the current state from the proposed framework and outline the basic content of the fixture point detection not only in simulation but also based on an initial experiment that assumes perfect geometric knowledge. This deliverable is closed with an outlook on how these findings will contribute to upcoming deliverables of the HR-Recycler project, especially within WP6.

1.1 Relation to Other Tasks and Deliverables

With the general goal in mind that robots should be capable of identifing shape, structure and composition of unknown objects, the haptic data acquisition forms one of core parts of the HR-Recycler project. While the vision-based perceptual analysis obtains initial beliefs over the disassembling objects is thoroughly evaluated in WP5, this deliverable gives an insight onto how the knowledge over these objects can be refined in an active (haptic) perception process. In combination with visual data, the sensor-motor object identification should iteratively decrease the uncertainty over perception of the actual object, and contribution towards decision over success of the overall disassembly process.

In contrast to the perception mechanisms from WP5, the active data acquisition directly incorporates the robot into this cognition process. Additionally, this deliverable does not only sketch how the findings from WP5 can be supplemented but also embeds the concept of flexible probing as a special case of WP6 into the context of the overall project. Thus, this deliverable outlines the methodological baseline and framework towards D6.4 which explicitly focuses on flexible probing as a scientific novelty in the haptic identification process within automated disassembly.

1.2 Structure of the Deliverable

The deliverable is structured as reported below:

Chapter 1 - Introduction -provides an overview over the deliverable.

Chapter 2 - Problem Statement - sketches the mathematical problem of haptic slam and the state of the art.

Chapter 3 - Background - sketches background information and theoretical foundations for the remainder of this deliverable.

Chapter 4 - Haptic Slam - outlines the schematics on how to explore unknown objects via haptic feedback and thus refine shape estimations of an unknown object

Chapter 5 - Flexible Probing - forms a special case of the haptic exploration process and outlines data acquisition via flexible probing.

Chapter 6 - Summary - summarizes the current state and gives an outlook onto future work.

2 Problem Statement

In the field of robotics, perception of unknown environments is mostly done based on visual data. Although computer vision techniques made impressive progress in the last decade, the abilities of artificial perception can not yet compete with human skills. Especially when looking at tasks where robots have to interact with unknown and changing surroundings for manipulating objects, one recognizes the inferiority of robots. Humans can instinctively acknowledge new situations and adopt their behavior accordingly. A major advantage is here given by the fact that humans do not solely rely on visual information but also take further sensual information into account. For instance, getting directly in touch through haptic feedback is one of those aspects. In recent years, this field gained more attention primarily on account of increasing computational power and improving sensor hardware. Moreover, other fields have proven that it is promising to analyze and reproduce human behavior and therefore, mimicking the human way of exploring unknown objects from haptic information bears a high potential towards improving robotic perception.

Using haptic feedback as an alternative source of information further allows to improve the knowledge about the robot's surroundings in terms of actually feeling the properties of objects. This approach known as tactile and haptic exploration enables to significantly increase and extend the results of visual object identification methods. For instance, visual data can only provide very little information about materials and their properties, inside structures and general mechanical design. However, such knowledge is especially crucial when it comes to tasks which do not only seek to identify but also manipulate objects, e.g. for disassembling unknown object casings for recycling.

Recent research in haptics has focused on the problem of minimizing geometric uncertainty through haptic feedback or distinguishing different sensor data. The goal of our work is to build upon this results and extend these methods to gather more information than the spatial dimensions of an object. Through taking other parameters into account, we want to establish a belief space over the underlying geometry as well as material types and their parameters. These parameters are for example given by the actual material properties like stiffness or elasticity. This results in an online inference algorithm which is able to provide information about the composition, structure and construction properties of an object. Last but not least, the evaluation of the elasticity forms a key aspect in the object identification, that allows to regress object data from force sensitive actuation. The remainder of this chapter, sketches the mathematical problem formulation tackled within this deliverable and sets the proposed methods in relation to recent findings from

literature.

2.1 Problem Statement

The objective of this work is to develop a method for identifying properties and parameters of unknown objects in terms of geometrical dimensions and material compositions. In order to provide such knowledge, we develop an approach to generate a model of the investigated object consisting of different shapes and parameters. Recent developments studied ways to improve the perception of geometric outlines through haptic and tactile feedback neglecting any further parameters, e.g. [9, 14, 38]. Starting from such methods, our resulting approach is not only able to classify the obtained data into clusters of different material types and shape models, but also to evaluate flexible probing strategy to infer object property, which in general is out of reach for pure visual systems. Having this goal in mind, the overall method can be divided into the following sub-tasks as depicted in Figure 2.1.

First, we have to design an haptic exploration scheme which gathers the necessary information from the object in the workspace corresponding to the left part of Figure 2.1. This allows to directly incorporate prior knowledge and / or beliefs as e.g. from visual perception. Given this initial belief, the current geometric shape is iteratively refined making use of Bayesian filter theory [44], defined as a defined cycle of control and measurement updates. This allows the proposed framework to update the current belief not only based on haptic feedback but also visual estimations. Given a set of possible object shapes, this algorithm intends to iteratively falsify and verify these shape hypotheses to ultimately converge to the actual object shape. This shape is expressed as a map $\boldsymbol{M}$ of the environment. The aim of the control update is to move the robot and its sensors represented through its state $\boldsymbol{x}_t$ at time t through the control input $\boldsymbol{u}_t$ which minimizes uncertainties with the next measurement update. More precisely, a measurement updates the belief $\mathbb{P}(\boldsymbol{M}|\boldsymbol{r}_t, \boldsymbol{x}_t)$ over the state $\boldsymbol{M}$ of an environment depending on the current measurement or sensor data $\boldsymbol{r}_t$ and the current pose of the robot $\boldsymbol{x}_t$. Eventually, each cycle improves the knowledge about the workspace iteratively, i.e. the uncertainty of the belief space is reduced with each iteration. This approach is inspired by the Simultaneous Localisation and Mapping (SLAM) [20] and the haptic SLAM method [9].

The second part is to further process the collected data and extract information about the explored geometry and material decomposition based on sensor data such as contact forces and torques as shown in the right part of Figure 2.1. Denoting that the exact parameterization of an object beyond its geometric shape is defined by a parameter vector Θ, the estimation of these parameters is tedious to add into an iterative Bayesian Filter update. Instead we propose to use the belief space of the environment $\mathbb{P}(\boldsymbol{M}_t|\boldsymbol{r}_t, \boldsymbol{x}_t)$ from the first problem in combination with the collected data-set to form a regression problem that obtains an estimation over the inner parameters Θ. The collected data consists of measurements $\boldsymbol{r}_{1..t} = \{\boldsymbol{r}_1, \ldots, \boldsymbol{r}_t\}$, robot states $\boldsymbol{x}_{1..t} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_t\}$ in combination with the dedicated control inputs $\boldsymbol{u}_{1..t} = \{\boldsymbol{u}_1, \ldots, \boldsymbol{u}_t\}$, such that the nonlinear regression

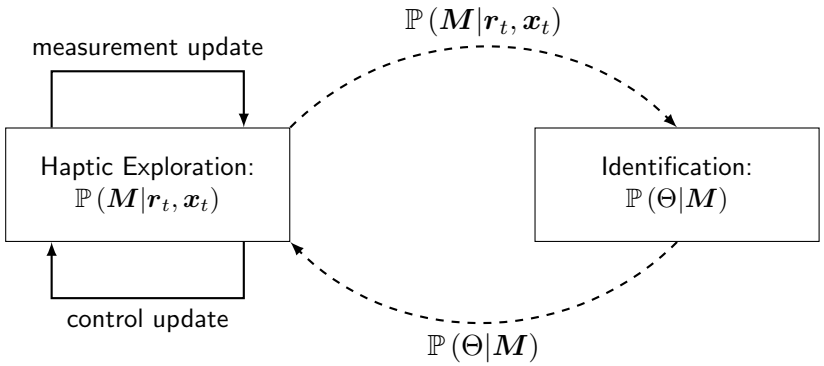


Figure 2.1: High-level Process Layout. The left box represents the haptic exploration and belief generation described as the first task. Extracting and identifying as the second part is depicted in the right box together with the corresponding transitions.

problem results in

$$\text{argmin}_{\Theta} = (f(\boldsymbol{x}_{1..t}, \boldsymbol{u}_{1..t}, \Theta|\boldsymbol{M}) - \boldsymbol{r}_{1..t},)^2, \forall \boldsymbol{M} \tag{2.1}$$

over unknown parameters Θ of the object. These parameters can be in regard of material types and their composition or about a detailed construction plan including connections and fixtures. As equation (2.1) shows, not only the current belief $\mathbb{P}(\boldsymbol{M}|\boldsymbol{r}_t, \boldsymbol{x}_t)$ from the exploration is used, but all previously accumulated data. In the remainder of this work, we outline two main identification processes. At first we show how unsupervised clustering algorithm allows to differentiate between different material types, and how this can used to update the current belief $\mathbb{P}(M)$ for the haptic exploration process. In addition to this, we outline how the evaluation of flexible probing as a special case for the identification of fixture points can be incorporated. In contrast to the first, the later focuses on evaluating small data sets with having exact knowledge about the object shape.

2.2 Related Work

In this section we sketch our current method in relation to the state-of-the art. Before outlining the aspect of haptic object identification, we give an insight into comparable applied research projects in automated disassembly.

2.2.1 Robotic Disassembly

Disassembly issue, which has been seen in earlier publications [5] and [40] and also in recent papers [6], [17], [18] and [15], with attracting social interest in recycling, is increasingly sophisticated and complicated, following and satisfying social requirements. Ways of disassembling the shell-containing-object have been partially but often studied with screwing or unscrewing problems. In [6], a method for detecting screws on a shell body of a electric motor is described. An image processing algorithm is presented for detecting screws, to automate the task of motor disassembly. It is not necessary to have a database

of templates for matching in the algorithm. The screws are detected based on their characteristics regarding its gray-scale, depth and HSV values. Fusion of features acquired from image and depth data is mainly introduced for detecting and locating of the bolts without a-priori knowledge or models.

DiFilippo et al. [17], [18] present a method for removing screws on laptop. In [17], the method integrates force and vision sensing to automatically locate and remove screws from various models of laptops. Their robot uses two different cameras in order to identify laptop features. The first camera is an overhead camera that can view the entire workspace and is used to find circles on the laptop that may contain screws. The second camera is mounted on the robot and used to find and center screw holes. They also uses Force-sensing resistors (FSR), which is triggered when the tip of their Sensor-equipped (SE) screwdriver makes contact with a surface. They regarded the method as the first step in disassembling laptops or other products with a plastic casing covering internal electronics.

In [18], Soar cognitive architecture has been presented with computer vision for automated robotic disassembly. Soar's semantic memory module is used for remembering location of circles that contained screws. In their approach, the memory is used to store different laptop models and their parameters. Soar has been trained with the dimensions of eight different laptop models and used to determine the model placed in the workspace and its orientation.

In [15]-[48], an implementation of a robot assistant for a task of unscrewing in disassembly with simple tools are described in terms of design of a bit holder. The unscrewing is for disassembling EV batteries in the context of human robot collaboration. On the assumption that EV batteries have a variety of screw and bolt fasteners with wiring as well as shells, the implementation with a KUKA LWR is shown.

In [50], as a cost effective solution, a screw-nail indentation approach for flexible disassembly is proposed. At the expense of partial destruction of contact surface, the indentation of the screw-nail can efficiently overcome the geometric uncertainties of products to be recycled. With the approach, dismantling of a plastic frame of a computer monitor has been introduced as a disassembly task. Methodology of disassembling shell-containing-object has not been well established and is, just as described above, still hot issue, however, a large part of them are focused on the problem of unscrewing or finding screws by using visual information. While our approach is also to contribute the problem of removing fixtures, proposed approaches in this report focuses on a problem of finding fixtures by using haptic information. Our approaches can be used in parallel with steps for finding fixtures by using vision information, as well as separately after the steps in order to find missed fixture.

2.2.2 Object Identification Based on Haptic Data

While visual information has been used in many cases to find targets (e.g. fixtures) in order to treat the targets after the searching steps in automatic disassembly processes, haptic information has been used to understand or recognize shape of object since early

times such as [1]. In [37], an approach for haptic object recognition based on extracting key features of tactile and kinesthetic data from multiple palpations using a clustering algorithm has been presented. In this approach, tactile sensor has been used for haptic sensation. The recognition framework has performed well on three different robotic hands; WRT102 Parallel Gripper, three fingered Schunk Dexterous Hand 2 and four fingered Anthropomorphic ARMAR-IIIb Hand.

Recent research here has studied the haptic exploration tasks introducing the name haptic SLAM (Simultaneous Localization and Mapping). The authors of [9, 8] take the results of visual SLAM techniques [20, 36] together with occupancy grid methods [22] and derive a concept of exploring the geometry of an unknown object while the pose of the sensor itself is also uncertain. The aim of these methods is to collect geometric and shape information without extracting direct geometrical features. Trough adaption of the FastSLAM [36] algorithm a novel method is proposed to iteratively learn the shape of the surface of objects. This is also used in a reverse-engineering approach of the human way of haptic exploration of an unknown object [7]. This haptic SLAM technique works well for extracting basic geometric shape information of objects as it is not providing more than a belief over an area being occupied by any kind of objects. However, for further identification tasks, other properties of the encountered objects may be of interest. Another method using haptic SLAM is presented in [39], although their algorithms require knowledge over the underlying object shape in advance.

On the other hand, there are methods to detect objects and especially edges of geometries through clever exploration strategies. The focus of this research lies on a lower level in terms of planning on how to maximize the knowledge about the environment. In [38], information of a tactile sensor array is used to explore objects and extract features with methods inspired by computer vision techniques. Similarly, there is work about contour extraction combined with shape detection [34]. They propose a method to actively find and follow contours based on tactile sensor information. The data source is again a sensor array measuring force data in 12 different points enabling the distinction between surfaces and edges. Anyway, their method seems to be limited to explore planar objects with right angles. In general, the detection and tracking of edges received a lot of attention [42, 14, 41]. Another approach is represented in [27] where range data and 2D images are combined to a generic object recognition algorithm.

For extracting more than shape and mapping information, computer vision reaches its limits. Visual sensors can only associate properties with priory known material types because obviously, they do not provide any direct data about the surface of the object. Therefore, few approaches exist which deal with finding geometric and material parameters at the same time based on haptic cues. In [32], a detailed review of tactile perception is provided. The authors present work about methods using surface and texture based information to find material properties and types. The same group recently proposed a method for shape recognition by combining proprioception and touch sensing in [33]. Another part deals with the estimation of stiffness and hardness of objects. For instance, [16] uses raw local tactile sensor data based on contact forces to classify those data points into material types using methods from computational intelligence. They compare different classifica-

tion algorithms and conclude that Support Vector Machines perform best in their setting. A similar approach is taken in [49]. Through data from a touch sensor, the algorithm classifies objects based on their texture. The authors also propose a Bayesian Exploration Algorithm which selects exploration actions to minimize the uncertainty in the underlying belief [26]. Nevertheless, most work is done on extracting either global shape information or local properties of sensed materials and surfaces.

However, perception of hardness and softness undergoes heavy investigation. For instance, the human way of haptic perception is studied in [28, 10] where the correlation between the actual physical properties and the perceived hardness is discussed. Additionally, the different cues used to determine the hardness are examined. In [10], the most important cues are the surface deformation and the ratio between force and displacement. The results of experiments in [28] show that stiffness and viscosity properties are correlated with the perceived hardness. Another conclusion suggests to take the frequency and damping of tapping the surface into account. One can conclude that stiffness and the deformation/displacement of an object are important measures or cues for discriminating different object and material types.

2.2.3 Object Understanding via Flexible Probing

There is also a part of attractive studies for understanding objects through bending. Lehmann et al. [31] present an pseudo-static-model-based approach for understanding an object: needle. In order to estimate the needle tip deflection and the needle's shape during insertion into soft tissue, their model is based on Euler-Bernoulli beam theory where the needle is modelled as a cantilever beam. Model component adjusts during insertion to the potentially changing load distribution due to tissue. Unknown load parameters are found by relating them to the force and torque measured by a force/torque sensor attached to the needle base. In their study, medical applications have been mainly assumed and a 20cm brachytherapy needle was used for their experimental validation. Wang et al. [47] also utilize the beam theory for online deflection estimation of X-Axis beam on positioning machine. They considered use of two beam theories, i.e. Timoshenko Beam theory and Euler-Bernoulli beam theory, and finally chose the latter beam theory as the basis in their study.

For more largely deforming situation, a Kirchhoff elastic rod has been considered for robotic manipulations [13]. In the study, an object, which is a planar elastic rod, is held at each end by a robotic gripper. Mochiyama [35] also utilize a Kirchhoff elastic rod for shape calculation problem for a metal strip in the context of robotic manipulation. Validity of a discretized spatial model of a closed elastica has been investigated. A method for real-time shape estimation based on force/torque information with utilizing a model of Kirchhoff elastic rod has been also proposed [43]. A shape estimation method for a soft robotic manipulator considering Cosserat rod approach is presented in [46]. Virtual "backbone curve" of the manipulator has been modelled as a Cosserat rod, which includes the effect of finite shear, curvature, and extension.

While these approaches described above are mainly based on direct use of the elastic

theory, extension or application to a plate like object is not strongly suggested. Our methods try to utilize haptic bending information to estimate location of hidden fixture for a plate like object.

2.3 Contribution of HR-Recycler in Haptic Data Acquisition

In contrast to research in haptics so far, the goal of WP6 within the HR-Recycler project is to actively explore the unknown objects and not only extract geometrical features but also identify material types and construction information like material borders and transitions. As outlined above, most of the recent work may conduct research on object recognition and identification but focuses on local data points and their classification. Methods like [16] perform online classification of sensor data with pre-trained models and methods from computational intelligence. We aim to extend such methods and work towards more generic application to obtain global object structures and construction information including object compositions. In order to achieve this, we exploit methods from the mapping and SLAM area [38, 34, 24] for the exploration part and methods from haptics for the identification tasks [49]. In contrast to recent work using haptic SLAM [9], we extend the existing framework to gather object information beyond geometrical features. The goal is to use the collected data and categorize the underlying object into geometric shape and estimate physical properties like the stiffness and elasticity.

We further extend the overall object identification process by outlining flexible probing as a special case of haptic data acquisition that allows robots to exploit and model the elasticity of objects, such that robots are able to locate fixture points, that are impossible to be detected solely from visual inspection. We outline this special case of the haptic data acquisition process on planar shaped objects, which have hidden fixture points a robot can identify via bending of said object.

The remainder of this work is structured as follows. The next chapter introduces the theoretical foundation, on which the methods for the proposed Haptic SLAM and flexible probing build upon. Given this, the haptic SLAM framework is outlined in detail in chapter 4, followed by the insights into the flexible probing in chapter 5. We summarize the current state of this task in chapter 6 and give an outlook onto future work.

3 Background

We incorporate different approaches from the fields of haptics and localization and mapping throughout this report. This chapter gives an overview of these methods and techniques and serves as the theoretical foundation for the remainder of this report. In particular, we provide an introduction to occupancy grids and the corresponding methods is given before we explain the concepts of SLAM and haptic SLAM. Proceeding from this, we provide fundamental information about physical parameters and properties of rigid bodies.

3.1 Occupancy Grids

Occupancy grid maps are a way to store time-evolving sensor data of 2D or 3D environments introduced by [22, 21]. It partitions space into cells which store an estimate of the state of the cell, i.e. the belief if a cell is occupied or not. Formally, the grid can be expressed as a map

$$\boldsymbol{M}_t = \{c_t(x_i, y_i, z_i)\} \in \mathbb{R}^{w \times d \times h}, \quad x_i \in [0, w],\ y_i \in [0, d],\ z_i \in [0, h]$$

where x_i, y_i and z_i represent the indices of a particular cell $c_t(\cdot)$ at time t in each dimension. The sizes in each dimension correspond to width $w \in \mathbb{N}$, depth $d \in \mathbb{N}$ and height $h \in \mathbb{N}$. In its simplest form, the state of a cell $c_t(x_i, y_i, z_i)$ is a discrete binary variable with the states *occupied* (occ) and *empty* (emp) [22]. In general, the states can be summarized in a set $\mathcal{C} = \{s_1, s_2\}$. The respective probabilities for these states are related through

$$\sum_{s_i \in \mathcal{C}} \mathbb{P}\left[c_t = s_i\right] = 1.$$

The states s_i and their probabilities can be interpreted as follows. If the probability $\mathbb{P}\left[c_t = s_i\right]$ is equal to one the cell c_t is most certainly in state s_i while a probability of zero corresponds to the opposite. Cells with no knowledge or an uncertain state are represented with a value of 0.5 meaning that both states are equally likely. This value of a cell is estimated based on measurements $\boldsymbol{r}_t$ via an update rule according to the update formalism of the Bayes' theorem [44]. From the prior belief $\mathbb{P}\left[c_t = s_i | \boldsymbol{x}_t, \boldsymbol{r}_t\right]$ with the knowledge of the robot position $\boldsymbol{x}_t$ and the last measurement $\boldsymbol{r}_t$ we can update the new belief via

$$\mathbb{P}\left[c_{t+1} = s_i | \boldsymbol{r}_{t+1}, \boldsymbol{x}_{t+1}\right] = \frac{\mathbb{P}\left[\boldsymbol{r}_{t+1} | c_t = s_i\right] \mathbb{P}\left[c_t = s_i | \boldsymbol{x}_t, \boldsymbol{r}_t\right]}{\sum_{s_k \in \mathcal{C}} \mathbb{P}\left[\boldsymbol{r}_{t+1} | c_t = s_k\right] \mathbb{P}\left[c_t = s_k | \boldsymbol{x}_t \boldsymbol{r}_t\right]} \tag{3.1}$$

where $\mathbb{P}[\boldsymbol{r}_{t+1}|c_t = s_i]$ has to be derived from the sensor model. The initial belief $\mathbb{P}[c_0|\boldsymbol{r}_0, \boldsymbol{x}_0]$ can be set arbitrarily given prior knowledge. Anyway, using the value 0.5 corresponding to an unknown state is common and plausible since it reflects the fact that there is no initial knowledge about the environment.

One drawback of this concept is that only two states $s_i \in \mathcal{C}$ of a cell can be encoded, e.g. either a cell is *occupied* or *empty*. Hence, Elfes et al. extended their initial proposal to the inference grid to enable the feature of storing more than one dimensional data [23]. As explained in [30], this inference grid consists of multiple stacked occupancy grids. Therefore, the resulting grid can be expressed as

$$\mathcal{M}_t^n = \{\boldsymbol{M}_t^j\} = \left\{\{c_t(x_i, y_i, z_i)\}^j\right\} \in \mathbb{R}^{w \times d \times h \times n}, \quad j \in [0, n] \tag{3.2}$$

where $n \in \mathbb{N}$ denotes the number of stacked occupancy grids. Instead of two different states, we can know express 2^n states at once. The overall update mechanism remains untouched as each layer can be seen as an independent occupancy grid. This means that the update formular in (3.1) can be applied independently for each layer $\boldsymbol{M}_t^j$of the inference grid.

In order to compare the knowledge available for particular cells distributed over the layers, different measures for the quality of the knowledge can be defined [24, 12]. Leaning on information theory, the entropy of a cell is defined as

$$H(c_t) = -\sum_{s_i \in \mathcal{C}} \mathbb{P}[c_t = s_i] \ln \mathbb{P}[c_t = s_i] \tag{3.3}$$

with s_i being the possible states of the cells. In [24], this is used to define the knowledge $\kappa(c_t) = 1 - H(c_t)$ of a cell. Following this approach, one can define the information gain of an control update $\boldsymbol{u}_t$ as the difference between cell entropies before and after an control update as follows [12].

$$I(\boldsymbol{u}_t) = H(c_{t-1}) - H(c_t)$$

Here, $\boldsymbol{u}_t$ is the control update between $t-1$ and t. Accordingly, $H(c_t)$ is the cell entropy at time t. This $I(\boldsymbol{u}_t)$ is also called mutual information [24]. One usage of the information gain and related measures is to sort cells by their amount of information available in order to decide which cell needs further investigation or exploration.

3.2 Visual and haptic SLAM

In general, the field of haptics in robotics deals with work based on direct interaction between all kinds of agents and their environment. In order to execute and plan tasks, the robot needs information about its environment and its own location in it. The SLAM method is a common approach to deal with a high correlation between the location and the

mapping tasks. Following the definition in [44], it basically aims to compute the probability distribution $\mathbb{P}[\boldsymbol{x}_t, \boldsymbol{M}|\boldsymbol{r}_{1..t}, \boldsymbol{u}_{1..t}]$ where $\boldsymbol{M}$ is the map of the environment, $\boldsymbol{r}_{1..t}$ all past and current measurements and $\boldsymbol{u}_{1..t}$ represent all past and current control inputs. This distribution is called the joint distribution of the robot state (location) and the map $\boldsymbol{M}$ [20]. In this setting, the map consists of k different landmarks defined by their position. Landmarks can be any kind of uniquely identifiable object. Naturally, other representations of the map are also possible, e.g. one could use grid-based approaches for storing information of the environment.

Having this probabilistic description in mind, the task can be split into two recurring parts. First, the estimation of measurements with the *observation model* $\mathbb{P}[\boldsymbol{r}_t|\boldsymbol{x}_t, \boldsymbol{M}]$. It describes the probability of observing the measurement $\boldsymbol{r}_t$ with known location $\boldsymbol{x}_t$ and a map of landmarks M. The second part is the *motion model* $\mathbb{P}[\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, \boldsymbol{u}_t]$ which is an expression of the state transition of the robot from state $\boldsymbol{x}_{t-1}$ to $\boldsymbol{x}_t$ through control input $\boldsymbol{u}_t$. This partition into observation and motion model makes it possible to treat the observations $\boldsymbol{r}_t$ and control actions $\boldsymbol{ut}$ separately.

For this generic SLAM problem, several solutions have been proposed in the last decades, e.g. the EKF-SLAM using Extended Kalman Filters (EKF) [44, Cha. 10]. A popular and more recent approach is called FastSLAM proposed in [36]. The authors use the concept of Rao-Blackwellized filters [19] to factorize the joint posterior into one factor depending on the robot pose and one depending on the map:

$$\mathbb{P}[\boldsymbol{x}_{1..t}, \boldsymbol{M}|\boldsymbol{r}_{1..t}, \boldsymbol{u}_{1..t}] = \mathbb{P}[\boldsymbol{x}_{1..t}|\boldsymbol{r}_{1..t}, \boldsymbol{u}_{1..t}] \prod_k \mathbb{P}[\boldsymbol{M}_k|\boldsymbol{x}_{1..t}, \boldsymbol{r}_{1..t}, \boldsymbol{u}_{1..t}]$$

This results in a robot pose or path estimator $\mathbb{P}[\boldsymbol{x}_{1..t}|\boldsymbol{r}_{1..t}, \boldsymbol{u}_{1..t}]$ and k landmark pose estimators $\mathbb{P}[\boldsymbol{M}_k|\boldsymbol{x}_{1..t}, \boldsymbol{r}_{1..t}, \boldsymbol{u}_{1..t}]$. The first one is solved by an particle filter and Kalman Filters are used for the landmark estimators. The map estimates are depending on the robot pose estimate and therefore, an individual local map estimate is associated to each particle in the particle filter [36]. For n particles, this results in nk Kalman Filters.

Based on this visual SLAM concept, a haptic approach is developed in [8]. Instead of the landmark representation of the map, the occupancy grid map is used. Hence, the algorithm does not estimate the position of landmarks but the state of the cells in the occupancy grid. The remaining algorithm is similar: each estimate of the exploration robot is represented with a particle filter. Each of those particles thereby contains an independent estimate of the contact map in the form of the occupancy grid. By adopting the FastSLAM approach [36], this paper provides a method to generate a contact map of the underlying object with the robot.

3.3 Stiffness of Rigid Bodies

When dealing with haptic sensor data, one needs to know which parameters of materials and objects in general are available for identifying and discriminating data and extract

object information. There are two measures available which can be calculated without complex measurement processes. First of all, the elasticity of a material should be considered. It is defined as the ratio between stress and strain and can be expressed as:

$$E = \frac{F/A}{\Delta l/l}$$

where F is the force applied to the area A resulting in relative deformation of length $\Delta l/l$. The physical name for this factor is Young's Modulus [3] having the meaning of how the material corresponds to applied bending and stretching forces. Subsequently, the stiffness is another parameter for the objects properties. In contrast to Young's Modulus, it is independent of the actual thickness and hence the volume of an object. It is expressed as:

$$k = \frac{F}{\Delta l}$$

representing the ratio between applied force F and deformation of the object Δl [2]. It provides information of how the surface of an object can be deformed by applying contact forces. With these parameters in mind, one can distinguish materials based on measured force and deformation or displacement.

4 Haptic Slam

To be able to process collected data in an efficient manner and cope with the large state space, we adopt the above mentioned framework of occupancy and inference grids. Based on an initial belief. e.g. from computer vision data, a discretized representation of the work-space is built from iteratively collected sensor data. Given this initial belief over the geometry of the object, we seek to refine the knowledge over the dedicated object. An imminent challenge lies in the refinement of the estimated geometric shape of the underlying object. Given initial sensor data, a set of shape candidates, represented via analytical parameterization of different types of shapes, can be inferred. Referring to the general problem definition from chapter 2, the depiction of these object shapes forms the current belief $\mathbb{P}(\boldsymbol{M})$. Given these shape candidates, we propose an iterative data acquisition method, that seeks to verify and falsify each of these candidates through haptic explorations. With these measurements adding further measurements to the overall data-set, the algorithm further allows not only to update the geometric decomposition but also the material decomposition Θ of the object. The remainder of this Section starts with an outline on the knowledge refinement via haptic exploration and is then closed by the estimation of simple material parameters, such as stiffness values. This serves as an initial proof of concept and its evaluation is thus restricted to a simulated environment. In contrast to this, the flexible probing in the upcoming chapter serves as an explicit special case of this parameter identification process, which is evaluated on actual hardware based on the assumption of knowing the exact shape of an object in advance. Before outlining how we infer and update the belief of the in general unknown object shape as well as the underlying material parameterization $\mathbb{P}(\boldsymbol{\Theta})$, we explain the applied inference grid insights in the following.

4.1 Grid-based Map for Accumulated Data

In order to keep track of the current belief of our environment, we use a grid-based framework as described in Section 3.1. The information about the state of a cell is using such an occupancy grid. However, the extension to the inference grid is used in order to process the actual sensor data.

4.1.1 Occupancy Information

In a first step to handle the occupancy information of each cell, the work-space is represented by the map:

$$\boldsymbol{M}_t = \{c_t\} \in \mathbb{N}^{w \times d \times h} \tag{4.1}$$

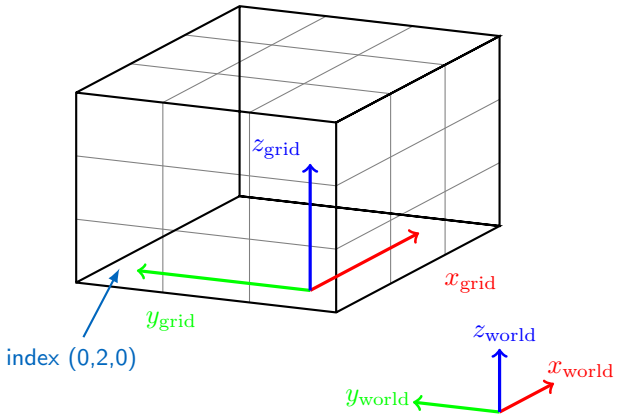


Figure 4.1: Illustrative Example of spatial Layout of the Occupancy Grid with $w = d = h = 3$ and a transformation as in equation (4.2)

where w, d, h denote the number of cells in each dimension. As this occupancy grid is a discretized representation of the work-space, a mapping from world coordinates to the grid is required. Figure 4.1 visualizes how the orientation of the occupancy grid can be derived from the world coordinate system through an Euclidean transformation $\boldsymbol{T}_{\text{W}}^{\text{M}}$. If the orientation is equal to the world coordinate frame, the transformation evaluates to

$$\boldsymbol{T}_{\text{W}}^{\text{M}} = \begin{bmatrix} 1 & 0 & 0 & x_{\text{grid}} \\ 0 & 1 & 0 & y_{\text{grid}} \\ 0 & 0 & 1 & z_{\text{grid}} \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{4.2}$$

representing a translation along the offset of the grid coordinates to the origin. In the general, the cells are identified by their indices where the cell at the origin of the grid frame has the index $(0, 0, 0)$ and all indices are non-negative. Furthermore, the transformation from grid indices to world coordinates is important to switch from solely grid-based notations into the world coordinates and vice-versa. Hence, we define those two mappings as follows.

$$f_{\text{w}\mapsto\text{g}} := \left\{ \xi_c \leftarrow \left\lfloor \frac{\xi_w}{\lambda_\xi} + 0.5 \right\rfloor, \quad \xi = x, y, z \right\} \quad \mathbb{R}^3_{\text{cart}} \mapsto \mathbb{N}^3, \tag{4.3}$$

$$f_{\text{g}\mapsto\text{w}} := \{ \xi_w \leftarrow 1.5 \xi_c \lambda_\xi, \quad \xi = x, y, z \} \quad \mathbb{N}^3 \mapsto \mathbb{R}^3_{\text{cart}}. \tag{4.4}$$

The first function $f_{\text{w}\to\text{g}}$ maps from world coordinates $x_{\text{w}}, y_{\text{w}}, z_{\text{w}}$ to cell indices x, y, z depending on the grid cell size $\lambda_{x,y,z}$. The latter (4.3) represents the other way. The result vectors have both four components due to the calculations being performed in homogeneous coordinates, i.e. the last entry of the result vector is ignored.

We use occupancy grids because it simplifies the measurement-to-cell matching. For incoming data, it is easy to determine if this point has already been explored. This reduces

the amount of data to store dense points with very close coordinates are associated to the same cell without further ado. As a consequence, it is not necessary to define a similarity metric to cluster points specifically or store the full information of all robot trajectories, which in general does not bring much benefit for the overall belief. This data association problem is a known pitfall for SLAM algorithms [4]. Another advantage is that with this method there is no need to make any assumptions about geometric features from the start on leading to more realistic results [9]. One disadvantage though is the cubic storage complexity of $\mathcal{O}(wdh)$ for grids with $w \cdot d \cdot h$ cells. This makes it unsuitable for large environments like maps of buildings or outside terrain. However, in the scope of this problem, we seek for refining visual perception output via haptic exploration. With computer vision baring initial beliefs over the structure and shape of the object, the refinement process can be limited to a small scale problem within a few meters at worst. Thus, the computational effort can be handled and the advantages outweigh the scaling drawback.

Having the occupancy grid from Section 3.1 in mind, each grid cell holds information about the state of this particular coordinate. It represents the belief if it is in one of the states $s_i \in \mathcal{C}$. In our setting, this means if it is occupied or not resulting in a set $\mathcal{C} = \{\text{occ}, \text{emp}\}$. However, in contrast to storing the actual probability $\mathbb{P}[c_t = s_i | \boldsymbol{r}_t, \boldsymbol{x}_t]$ in the grid as in [22], we use the log-odds notation of the current belief [9]. Given a new measurement $\boldsymbol{r}_t$ at time step t, the belief c_t of the cell in log-odds notation is updated as

$$c_t = c_{t-1} + \log \frac{\mathbb{P}[\boldsymbol{r}_t | \boldsymbol{f}_\text{s}]}{1 - \mathbb{P}[\boldsymbol{r}_t | \boldsymbol{f}_\text{s}]} \tag{4.5}$$

where c_{t-1} denotes the prior belief of that cell and $\boldsymbol{f}_\text{s}$ is the sensor reading. Given no initial data from a vision system, the initial prior results in equally likely cell occupancy probabilities $\mathbb{P}[s_i] = 0.5$, which means c_0 is equal to zero. If a prior belief is given we initialize free cells with a predefined negative value $-K$. With this notation, cells that are occupied evolve to having a very large belief moving to $+\infty$ and empty cells evolve towards $-\infty$ while unvisited cells or cells with an uncertain state remain close to zero. We use the log-odds notation since it provides a finer resolution to a greater value range and the updating requires less computation than the Bayesian update rule in equation (3.1). The measurement $\boldsymbol{r}_t$ based on the sensor reading $\boldsymbol{f}_\text{s}$ is incorporated through the probability $\mathbb{P}[(] \boldsymbol{r}_t | \boldsymbol{f}_\text{s})$ and therefore, a relation between those values has to be obtained. For this purpose, we establish a sensor model mapping sensor readings to this probability. To represent the occupancy of a cell, we use a simple sensor model based on a threshold decision rule assuming that a cell can either be completely occupied or empty, i.e. $c_t \in \mathcal{C}$. Based on sensor readings $\boldsymbol{f}_\text{s}$ representing information about contact between sensor and an object, the probability for a cell being occupied can be described by

$$\mathbb{P}[\boldsymbol{r}_t | \boldsymbol{f}_\text{s}] = \begin{cases} 0.9, \text{ if } \boldsymbol{f}_\text{s} > \boldsymbol{f}_\tau \\ 0.1, \text{ otherwise} \end{cases} \tag{4.6}$$

where $\boldsymbol{f}_\tau$ is a threshold in terms of a decision boundary. Note that the actual meaning of the sensor values are not of any matter here as long as it allows inference of the occupancy state at this position. Therefore, if a measurement $\boldsymbol{f}_\text{s} > \boldsymbol{f}_\tau$ is seen the corresponding cell is updated with $\mathbb{P}[\boldsymbol{r}_t | \boldsymbol{f}_\text{s}] = 0.9$ assuming that this cell is occupied, i.e. it has a high

probability of being in state $s_1 = \text{occ}$. In the opposite case with $\boldsymbol{f}_\text{s} \leq \boldsymbol{f}_\tau$, this cell is likely to be empty with $s_2 = \text{emp}$ and consequently, updated with $\mathbb{P}\left[\boldsymbol{r}_t | \boldsymbol{f}_\text{s}\right] = 0.1$.

For recovering the probability based on the log-odds values stored in the grid, one can use the following formula.

$$\mathbb{P}\left[c_t = s_i | \boldsymbol{r}_t, \boldsymbol{x}_t\right] = \frac{e^{c_t}}{1 + e^{c_t}} \tag{4.7}$$

4.1.2 Storage of Sensor Data

Since we want to be able to further process the sensor readings, storing the incurring measurement data is of interest in addition to the belief about the occupancy. Hence, not only a binary occupancy grid is maintained but it is extended to an inference grid. It follows that the overall map $\boldsymbol{M}_t^n$ consists of n grid layers resulting in $\boldsymbol{M}_t^n \in \mathbb{R}^{w \times d \times h \times n}$ with n being the dimension of the sensor data plus one for the occupancy grid. Accordingly, the first layer of this grid $\boldsymbol{M}_t^1$ holds the occupancy values from the previous Section while the higher order levels store the sensor values for further processing. This map can be expressed as

$$\mathcal{M}_t^n = \{\boldsymbol{M}_t^1, \ldots, \boldsymbol{M}_t^n\} = \{\{c_t\}_\text{occ}, \{c_t\}_{\text{s}_1}, \ldots, \{c_t\}_{\text{s}_\text{m}}\} \tag{4.8}$$

where the first grid layer $\{c_t\}_\text{occ}$ represents the occupancy state and the other layers hold the sensor values with $m = n - 1$ being the dimension of those. Each cell can be uniquely identified through its indices x_i, y_i and z_i and the number of the grid layer $j \in [1, m]$ resulting in the expression $c_t^j(x_i, y_i, z_i)$. For the sake of brevity, we omit the indices x_i, y_i and z_i when clarity allows it.

The values stored in the layers $j > 1$ do not represent a belief. In contrast, we want to store actual raw sensor readings and hence, the update rule based on the log-odds from equation (4.5) cannot be used appropriately. Instead, a new rule is defined formulated as

$$c_t^j = \begin{cases} f_\text{s}^j, & \text{if } c_{t-1}^j = 0 \\ \frac{1}{\eta}\left(\gamma c_{t-1}^j + f_\text{s}^j\right), & \text{otherwise} \end{cases} \tag{4.9}$$

where f_s^j denotes the jth dimension of the sensor reading $\boldsymbol{f}_\text{s}$ and c_{t-1}^j corresponds to the previous cell value, i.e. prior. Since the initial values are $c_0^j = 0$, the first case basically means that the first measurement of a cell is directly stored in the grid. In the second case, the parameters γ and η are normalizing constants, a natural way of choosing them is to use the mean of prior and new measurement, i.e. $\gamma = 1$ and $\eta = 2$. In case a prior from visual data is incorporated, as outlined above, the cell values will be initialized with a predefined prior. As a consequence the iterative update rule is capable of eventually overcoming false negative occupancy beliefs, but my require a tremendous amount of runs to achieve this. Nonetheless, as this algorithms seeks to refine the object specifications rather than free space misallocation, this risk is negligible.

Building upon this data grid representation, we eventually seek to estimate the exact shape of the object. Thus, we outline the proposed shape estimation and modeling in the following Section, which is then exploited to obtain a exploration strategy that tries to maximize the expected information gain in each step.

4.2 Analytical Shape Representations

Besides the discretized inference grid, a continuous representation of the dedicated object is of advantage. Hence, different geometric types of shapes are considered which are represented through a parameter vector Φ for a shape S. In detail, we provide models for spheres, cylinders, boxes and planes. This Section outlines a description of the parameters of each shape type and how to derive these descriptions from data.

One reason for using such analytical models is that the inference grid framework obviously performs a kind of discretization of the work-space leading to unavoidable errors and uncertainties between grid and the actual environment. Second, the parameter vectors have a much smaller dimensionality and hence a smaller spatial complexity than the inference grid reducing the overall computational effort significantly. Moreover, the encoded geometric information can be easily used later on for extracting object properties and parameters for the purpose of object identification.
It has to be noted that the actual shape estimation and representation does not form the core contribution of this task and is thus not limited to the expression sketched out in this Section. It rather serves as an initial baseline solution for the presented feasibility studies.

Sphere One of the simple description of a shape type is the sphere since it can be defined by only a center point $\boldsymbol{c}$ and the radius r. These two parameters define the sphere completely resulting in a parameter vector denoted by

$$\Phi_{\mathrm{s}} = \left[\boldsymbol{c}^{\top}, r\right] \in \mathbb{R}^4 \tag{4.10}$$

resulting in 4 Degrees of Freedom (DoF)s. Hence, if one wants to obtain a parameterization at least four points are needed. Algorithms such as Random Sample Consenus (RANSAC) then allow to fit models to a set of over-determined data points. Provided with a set of points $\boldsymbol{p}_i \in \mathbb{R}^3$, it estimates the center and the radius by minimizing the point-model distances and neglecting noisy data points. This model allows a straightforward representation of spherical objects or components.
The second geometric shape to mention here is the cylinder. For a well-defined geometry, one needs a starting point $\boldsymbol{s}$, the vector of the center axis $\boldsymbol{c}$ and the radius r. Additionally, for limiting the cylinder in the direction of the axis, the height h is specified as the distance from the starting point to the farthest point in the direction of the axis. This results in a 8 dimensional parameter vector formulated as

$$\Phi_{\mathrm{c}} = \left[\boldsymbol{s}^{\top}, \boldsymbol{c}^{\top}, r, h\right] \in \mathbb{R}^8 \tag{4.11}$$

Again, the RANSAC method can be used to estimate these parameters from many input points.

Planar Shape Another simple shape type is a planar surface with or without boundaries. In analytical geometry, there are several definitions describing a plane. For instance, it can be defined by a point $\boldsymbol{p} = [p_x, p_y, p_z]^\top$ on it and the normal vector $\boldsymbol{n} = [n_x, n_y, n_z]^\top$, which then results in

$$\begin{aligned} d &= -n_x p_x - n_y p_y - n_z p_z \\ 0 &= n_x q_x + n_y q_y + n_z q_z + d \\ 0 &= \frac{\boldsymbol{n} \cdot \boldsymbol{q} + d}{|\boldsymbol{n}|} \end{aligned}$$

for points $\boldsymbol{q} = [q_x, q_y, q_z]^\top$ on the plane. The latter form (4.12) is called the Hessian normal form. All these equations describe planes in $\mathbb{R}^3$ without any constraints, i.e. they have an infinite expansion. However, having the setup of a finite robot work-space in mind makes it clear that constraints in terms of borders are useful and required here. Hence, we add four points resembling the corners to the definition of the plane. Finally, each plane is defined by four corners $\boldsymbol{a_i}$ and the unit normal vector $\hat{\boldsymbol{n}}$ leading to a parameter vector

$$\Phi_\mathrm{p} = \left[\boldsymbol{a_1}^\top, \boldsymbol{a_2}^\top, \boldsymbol{a_3}^\top, \boldsymbol{a_4}^\top, \boldsymbol{n}^\top\right] \in \mathbb{R}^{15} \tag{4.12}$$

The corners $\boldsymbol{a_i}$ have to meet certain conditions such that they form a valid plane which can be expressed by the following relations.

$$\boldsymbol{a_i} \times \boldsymbol{a_j} \sim \hat{\boldsymbol{n}} \qquad i \neq j \tag{4.13}$$

which basically states that all mutations of the corners $\boldsymbol{a}_i$ have to be similar to the normal vector $\hat{\boldsymbol{n}}$.

Box Shape For box-like shapes, different parameterizations can be considered, e.g. through specifying all six surface planes. However, for the sake of simplicity we use a corner as starting point and then calculate the edge lengths in terms of width, depth and height. We even neglect orientations other than when the box is aligned with the coordinate axes. Hence, the width means the edge length in x-direction, depth the length in y-direction and height the length in z-direction. The parameter vector for this type is denoted as

$$\Phi_\mathrm{b} = \left[\boldsymbol{c}^\top, w, d, h\right] \in \mathbb{R}^6 \tag{4.14}$$

In order to estimate these parameters, a rather simple approach is taken here. First, the smallest x, y and z coordinate of all points p_i is calculated. These minimal coordinates are then used as the starting vector $\boldsymbol{c}$. In the same manner, the greatest x, y and z coordinate is extracted.

$$\boldsymbol{c} = \begin{bmatrix} \min\limits_{\text{All } x \text{ of } p_i} (x) \\ \min\limits_{\text{All } y \text{ of } p_i} (y) \\ \min\limits_{\text{All } z \text{ of } p_i} (z) \end{bmatrix} \quad , \qquad \boldsymbol{m} = \begin{bmatrix} \max\limits_{\text{All } x \text{ of } p_i} (x) \\ \max\limits_{\text{All } y \text{ of } p_i} (y) \\ \max\limits_{\text{All } z \text{ of } p_i} (z) \end{bmatrix} \tag{4.15}$$

The resulting edge lengths can then simply calculated via

$$\begin{bmatrix} w \\ d \\ h \end{bmatrix} = \boldsymbol{m} - \boldsymbol{c} \tag{4.16}$$

Note that this calculations do not consider any terms of errors induced by noisy and uncertain data points in contrast to the RANSAC method used above. The procedure here is not a parameter estimation in the proper sense. However, in a simulation environment the influence of noise can be controlled well enough to accept this inaccuracy.

Altogether, these shape models can be used to develop a belief space over the geometries in our work-space. In order to do so, the next Section defines a probabilistic filter based on Bayes and Particle Filters.

4.2.1 Particle-Like Filter for Shape Representations

In contrast to the grid-based approach, the second option provides a continuous view over the belief of the work-space. We define a probabilistic filter with concepts from Bayes and Particle Filters. As a first step, a set of shapes $\mathcal{S}_t$ representing different parametrizations Φ is defined as the fundamental object we operate on. At time t, it is defined as

$$\mathcal{S}_t = \left\{ S_t^i \right\}, \quad 0 < i \leq n, \tag{4.17}$$

where n is the number of parameterizations, i.e. the number of particles. The belief about the likelihood of a particle is denoted by $\mathbb{P}\left(S_t^i\right) = \mathbb{P}\left[S_i^t | \boldsymbol{x}_t, \boldsymbol{r}_t\right]$. Once again, $\boldsymbol{x}_t$ denotes the robot state and $\boldsymbol{r}_t$ the measurement both at time t. Naturally, the accumulated belief of all shapes has to fullfill the condition

$$\sum_{S_t^i \in \mathcal{S}_t} \mathbb{P}\left[S_t^i | \boldsymbol{x}_t, \boldsymbol{r}_t\right] = 1. \tag{4.18}$$

As known from Bayes Filters, the transition from time t to $t+1$ based on an incoming observation $\boldsymbol{r}_{t+1}$ is called the measurement update and can be formulated as follows.

$$\begin{aligned} \mathbb{P}(S_{t+1}^i | \boldsymbol{r}_{t+1}, \boldsymbol{x}_{t+1}) &= \mathbb{P}\left[S_t^i | \boldsymbol{x}_t, \boldsymbol{r}_t\right] \mathbb{P}\left[\boldsymbol{r}_{t+1} | S_t^i\right] \quad \forall S_t^i \in \mathcal{S}_t \\ \eta &= \frac{1}{\sum\limits_{S_t^i \in \mathcal{S}_t} \mathbb{P}'(S_{t+1}^i | \boldsymbol{r}_{t+1}, \boldsymbol{x}_{t+1})} \\ \mathbb{P}\left[S_{t+1}^i | \boldsymbol{r}_{t+1}, \boldsymbol{x}_{t+1}\right] &= \frac{1}{\eta} \mathbb{P}'(S_{t+1}^i | \boldsymbol{r}_{t+1}, \boldsymbol{x}_{t+1}) \quad \forall S_t^i \in \mathcal{S}_t. \end{aligned} \tag{4.19}$$

The first equation performs the actual update based on the prior belief $\mathbb{P}\left[S_t^i | \boldsymbol{x}_t, \boldsymbol{r}_t\right]$ and the sensor model $\mathbb{P}\left[\boldsymbol{r}_{t+1} | S_t^i\right]$, while the latter two equations normalize the new belief subject to the condition in (4.18) [44]. However, the sensor model maps the measured

values $\boldsymbol{r}$ to each shape S^i via the probability $\mathbb{P}\left[\boldsymbol{r}_{t+1}|S_t^i\right]$. In this particular case, a normal distribution is used denoted as:

$$\mathbb{P}\left[\boldsymbol{r}_{t+1}|S_t^i\right] = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{(\boldsymbol{r}_{t+1} - \boldsymbol{\mu}_i)^\top (\boldsymbol{r}_{t+1} - \boldsymbol{\mu}^i)}{2\sigma^2}\right) \sim \mathcal{N}(\boldsymbol{\mu}^i, \sigma) \tag{4.20}$$

with $\boldsymbol{\mu}^i$ being the expected sensor values given the shape S^i. In other words, if a measurement $\boldsymbol{r}$ has a large distance to the expected measurement at $\boldsymbol{\mu}^i$ the shape parametrization in particle S^i is unlikely to align with the actual object in the point $\boldsymbol{x}$ where the measurement has been taken.

Thinking of Bayesian Filters, one recognizes the absence of the control or motion update. In this setting, we assume the motions of the robot to be entirely deterministic and hence, it has no influence on the evolution of the belief. This filter enables to incorporate ongoing measurements into a current belief state without the need of taking a premature choice of the underlying parametrizations.

Having outlined the belief representations and update rule over the current shape candidates, the imminent question arises on how to exploit this knowledge and acquire new measurements in such a way that the uncertainty over the current object is decreased. Thus the next Section outlines the proposed strategies towards obtaining new data samples given the proposed framework.

4.3 Haptic Exploration

Before being able to extract information about the objects in our work-space, the robot has to collect data through exploration. In order to collect this sensor data in an efficient manner, we design a control flow based on a utility metric for exploring the environment. Starting with some initial knowledge, we iteratively update our map of the work-space. This initial belief of the work-space can have multiple sources. A natural way is to use data from visual sensors leaning on the way humans interact with their environment. Refining initial estimations of an object via feeling and touching, i.e. haptic feedback, is thus worth mimicking.

For this purpose, two different strategies are considered, one is based on the inference grid framework of Section 4.1 and the second one uses the methods of Section 4.2.1. The goal of both strategies is to process ongoing measurements and incorporate new data into the belief about the environment.

First, the method based on the inference grid framework is outlined. It results in a grid-based map of the environment storing information about the occupancy state and the corresponding measurements. The second approach uses the particle filters to compute and evolve analytical geometric representations based on the collected sensor data.

4.3.1 Grid-Based Exploration

As laid out in Section 4.1, the inference grid framework provides a suitable methods to map data of an environment and store it. We develop a strategy around it which enables accumulating incoming sensor data such that it can further processed afterwards. The basic map represents a belief about the state of the work-space in combination with the respective sensor data. Hence, the underlying data can be expressed by an inference grid $\mathcal{M}_t^n$. For the exploration, especially the first layer $\boldsymbol{M}_t^1$ regarding the occupancy information is of interest, compare (4.8). For selecting the next exploration step, the next action $\boldsymbol{a}_t$ has to be extracted. With the overall goal of increasing the available information and decreasing the uncertainty, this step is chosen such that it provides the highest possible gain of information. Hence, we need a method to rank different goals in the grid by their value of data. The resulting measure is derived in the next Section before it is used to outline the actual exploration strategy afterwards.

Ranking Function for the Inference Grid

The purpose of the ranking function is to find the cell which has the least information available, i.e. which belief is closest to the uncertain state. Hence, we calculate the gain of information a certain cell would yield if chosen for the next exploration. To build such a rank, we estimate this information gain for all cells in the map $\boldsymbol{M}_t^1$. As described in Section 3.1, there are several approaches more or less derived from the entropy. Based on those measures, an alternative proposed in [29] is defined as

$$U(c_t) = \sum_{r\in\mathcal{C}} \sum_{c\in\mathcal{C}} \mathbb{P}\left[r_t = r|c_t = c\right] \mathbb{P}\left[c_t = c\right] \ln\left(\frac{\mathbb{P}\left[c_t = c|r_t = r\right]}{\mathbb{P}\left[c_t = c\right]}\right) \tag{4.21}$$

called the utility of cell $c_t(x_i, y_i, z_i)$ at time t. The variables r and c stand for the possible results of the measurement and the possible states of the cell $\mathcal{C} = \{\text{occ}, \text{emp}\}$. The prior belief is represented through $\mathbb{P}\left[c_t = c\right]$ and can be calculated through equation (4.7) and the cell values. The probability $\mathbb{P}\left[r_t = r|c_t = c\right]$ can be derived from the sensor model and hence, one can deduce

$$\mathbb{P}\left[r_t|c_t\right] = \begin{cases} 0.9 & \text{if } r_t = c_t \\ 0.1 & \text{otherwise} \end{cases} \tag{4.22}$$

from equation (4.6), meaning the probability is high if measurement r_t is equal to the current belief c_t. Furthermore, with Bayes' law, the probability $\mathbb{P}\left[c_t = c|r_t = r\right]$ can be simply calculated based on the prior $\mathbb{P}\left[c_t\right]$ [29].

$$\mathbb{P}\left[c_t|r_t\right] = \frac{\mathbb{P}\left[c_t\right] \mathbb{P}\left[r_t|c_t\right]}{\sum_{c\in\mathcal{C}} \mathbb{P}\left[c_t = c\right] \mathbb{P}\left[r_t|c_t = c\right]}$$

With these definitions, the utility in (4.21) can then be directly calculated for all cells. Basically, it maps the prior belief of a cell to all possible measurements. That is, a prior

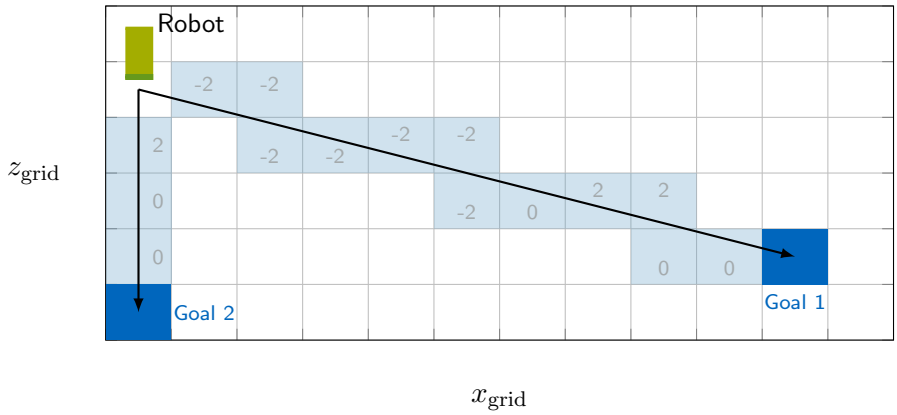


Figure 4.2: Basis for utility and accessibility calculations in a 2D grid for two different goal cells. The black arrow shows the direct connection of the position with the goal while the light blue cells indicate the result of the line discretization. The gray numbers in the cells show the respective belief stored in the cell.

expressing a high certainty about its state results in a low utility and vice versa. Arguing the other way, high values of the utility mean that the information added by exploring this cell is also high, i.e. the available knowledge about this cell is low and therefore, it would be a good choice for the next exploration move.

However, there are cells which may be unreachable for the robot in realistic scenarios, e.g. the inside of rigid bodies or geometries with cavities. These cells result in a static utility of their initial value, since the information about the state of the cells cannot change without the cells being explored. Additionally, the insides of bodies are likely to have an initial belief $\mathbb{P}(c_t) = 0.5$ standing for uncertain state which will result in a relatively high utility values compared to explored cells. If parts of the rank are basically static the ordering of cells distorts the true purpose of the utility.

Hence, another measure is introduced which takes the accessibility of cells into account. This value helps to determine how likely it is that the robot can actually access this point in the work-space. Since the reachability of a point depends on the current pose of the robot, it is calculated based on the current robot pose $\boldsymbol{x}_t$ and the particular cell c_t. In order to assess how likely a cell can be accessed, we compare the state of all cells in between the position and cell itself. For this purpose, we need to extract all the cells on this way. For now, we focus on linear trajectories such that we can discretize the direct connection between these two points via the 3DDDA (3D Digital Differential Analyser) technique [**?**]. The method is depicted in a simplified 2D setting in Figure 4.2 where the goal cell and all light blue cells represent the result of the discretization. As a result, we obtain a set of cells $\mathcal{C}_t = \{c_1, c_2, \dots\}$ which is then used to calculate the accessibility.

As already mentioned, the accessibility measures the likelihood of a cell being reachable from the current position. It is obvious to use the state of occupancy of the intermediate

cells to determine if a cell is blocked by an object. So the derived function accumulates the sign of these cells and normalizes the result:

$$\alpha(\mathcal{C}_t) = \left| \frac{1}{|\mathcal{C}_t|} \sum_{\forall c_i \in \mathcal{C}_t} \text{sign}(c_i) \right| \tag{4.23}$$

where $|\mathcal{C}_t|$ denotes the cardinality of the set of cells $\mathcal{C}_t$. Table 4.1 shows the calculation of the accessibility for the setting depicted in Figure 4.2, separately for both goals 1 and 2. Each row in the table represents the accessibility for that particular cell where the cell index in the first column is constructed by simply counting through the members of the line. The second column shows the belief of the cell stored in the grid. Recall that values below 0 are associated with empty cells and greater 0 means occupied. The remaining columns show the calculations step-by-step, from sign over sum to the final value. As a result, cells which already have been explored have an accessibility value close to one depicted in the last column. The accessibility is higher if a cell has only empty cells above itself and with increasing distance and occupied cells in the way, the accessibility decreases. As one can see in Table 4.1 for the first goal, the accessibility is high as long as only empty cells are evaluated, but it decreases as soon as either uncertain or occupied cells appear. The values for the second goal further show that occupied cells on the way yield a decreasing accessibility for all following cells.
However, if the route only contains unvisited or uncertain cells meaning all cells are equal

Table 4.1: Example calculations of accessibility for grid depicted in Figure 4.2, each row represents the accessibility for the respective cell. The first column shows the index on the line, second is the corresponding belief stored for that cell. The remaining columns show the calculations of $\alpha(c_t)$ step by step.

goal	cell index	c_t	sign(c_t)	$\sum$	$\alpha(c_t)$
Goal 1	1	-2	-1	-1	1
	2	-2	-1	-2	1
	3	-2	-1	-3	1
	4	-2	-1	-4	1
	5	-2	-1	-5	1
	6	-2	-1	-6	1
	7	-2	-1	-7	1
	8	0	0	-7	0.875
	9	2	1	-6	0.667
	10	2	1	-5	0.5
	11	0	0	-5	0.455
	12	0	0	-5	0.417
Goal 2	1	2	1	1	1
	2	0	0	1	0.5
	3	0	0	1	0.333

to zero the accessibility here would be zero as well. This would result in the cell marked as "unreachable" even though the grid does not provide any knowledge about this trajectory. So if there is no knowledge about the occupancy of cells available yet, we only consider the distance between position and goal cell, i.e. the size of the set $\mathcal{C}_t$. Hence, the final accessibility measure can be formulated as:

$$\alpha(\mathcal{C}_t) = \begin{cases} \frac{1}{|\mathcal{C}_t|} & \text{if } \|\mathcal{C}_t\|_1 = 0 = 0 \\ \left| \frac{1}{|\mathcal{C}_t|} \sum_{\forall c_i \in \mathcal{C}_t} \text{sign}(c_i) \right| & \text{otherwise,} \end{cases} \tag{4.24}$$

where $\|\|_1$ denotes the L^1-norm. Finally having this setup in place, the utility from (4.21) and accessibility from (4.24) can be combined to derive a rank of all cells allowing to order them by their expected information gain if they were chosen to explore. As discussed with these functions above, it is desirable to explore cells with both a high utility and a high accessibility. Hence, we adjust the utility rank as the product:

$$\text{rank}(c_t, \mathcal{C}_t) = \alpha(\mathcal{C}_t) U(c_t), \tag{4.25}$$

with the requested cell c_t as input together with the set of cells $\mathcal{C}_t$ between c_t and the current position $\boldsymbol{x}_t$. This rank enables to sort cells by their degree of uncertainty, i.e. the degree of utility an action brings if this cell is explored. The higher a rank is, the lower the certainty is and therefore, the robot tries to explore this cell in the next step. Through including the accessibility, we decrease the tank of unreachable cells and thus avoid constantly reaching for cells that cannot be explored. This rank is now used to define the overall exploration strategy in the next Section.

Exploration Strategy

For the purpose of exploring the work-space and gather sensor data, an exploration flow is defined. It considers planning and carrying out the steps as wells as handling the collected data. A step-by-step description can be formulated as

1. Compute next action for exploration and perform this action
2. Approach the exploration goal
3. Start measurement by applying force once contact is established
4. Update inference grid when measurement is done

For the first step, the above ranking function (4.24) is used to filter for the best suitable action. Starting with the current position $\boldsymbol{x}_t$, the members of the inference grid are sorted such that the most useful goal for the next action $\boldsymbol{a}_t$ can be extracted. For this purpose, Algorithm 1 defines the procedure which calculates all necessary inputs for utility, accessibility and the ranking and finally returns the goal $\boldsymbol{g}_t$ of the next step. First, the corresponding grid cell $\boldsymbol{x}_t^c$ of the current position $\boldsymbol{x}_t$ has to be determined via (**??**). Then, the ranking $\boldsymbol{R}$ is calculated for each cell in the occupancy grid. As (4.25) states, the ranking function needs all cells on the axis between current position $\boldsymbol{x}_t^c$ and the respective

cell c_t as input. Therefore, the set $\mathcal{C}_t$ containing all these cells is obtained first before the actual ranking for cell c_t is calculated. Afterwards, the most useful goal is chosen as the cell with the highest overall rank. In the end, the next goal $\boldsymbol{g}_t$ and output of this algorithm is the position of the cell c_t^{goal} in world coordinates. This position is derived by transforming c_t^{goal} into the world by (4.3).

Algorithm 1: Obtain Next Exploration Goal Based on Grid Metric

Input: Position $\boldsymbol{x}_t$, Occupancy Grid $\boldsymbol{M}_t^1$
Output: Next goal $\boldsymbol{g}_t$
$\boldsymbol{x}_t^c \leftarrow f_{w\to g}(\boldsymbol{x}_t)$ ▷ transform $\boldsymbol{x}_t$ to grid cell
$\boldsymbol{R} \leftarrow \{\}$
foreach $c_t \in \boldsymbol{M}_t^1$ **do**
  $\mathcal{C}_t \leftarrow \texttt{discretize}(\Delta p = \boldsymbol{x}_t^c - c_t)$ ▷ discretize direct connection
  $\boldsymbol{R} \leftarrow \boldsymbol{R} \cup \text{rank}(c_t, \mathcal{C}_t)$ ▷ store expected rank from (4.25) in $\boldsymbol{R}$
$c_t^{\text{goal}} \leftarrow \underset{r\in\boldsymbol{R}}{\text{argmax}}\{r\}$ ▷ select cell with highest rank
$\boldsymbol{g}_t \leftarrow f_{g\to w}(c_t^{\text{goal}})$ ▷ transform cell to world coordinates

With this goal $\boldsymbol{g}_t$, a trajectory to approach the goal cell is generated before the measurement can be started in the third phase. In the setting of Figure 4.2, the black arrows depict the trajectories for the two goals. Furthermore, the sensor attached at the end-effector is aligned with the axis between position $\boldsymbol{x}_t$ and the goal $\boldsymbol{g}_t$ such that the sensor surface is normal to exploration axis. This alignment is useful to ensure that the sensor can actually detect contact in the case an object is in the path of it.

When the robot has reached the desired position the measurement procedure starts. As a consequence, the movement is slowed down at the near vicinity of the point of contact. During this contact phase an Cartesian Impedance controller is used with an attraction point around the chosen contact point. Once the sensor registers a contact force, an additional desired Cartesian Wrench is applied until the respective conditions are fulfilled depending on the downstream data processing, see Chapter 4.5. The obtained values are collected and expressed as the sensor reading $\boldsymbol{f}_{\text{s}}$ with the aim of using it to update the occupancy state with (4.5) and the remaining layers of the inference grid with (4.9).

After updating the inference grid, the next iteration of the exploration starts by repeating Algorithm 1. While this procedure allows to refine the shape estimation of an object, it also stores data which can then be used for the actual object identification from Section 4.5 and Chapter 5. To summarize, this subSection develops the methods for exploring an unknown environment and collect data of contained objects. Before proposing methods for processing this data, another procedure to explore the work-space is derived in the following subSection. It builds upon the continuous shape-based data representation from Section 4.2 and 4.2.1.

4.3.2 Shape-Based Exploration

A second approach for the exploration strategy is based on the analytical shape representation from Section 4.2 and 4.2.1. This representation eliminates uncertainties due to discretizations in contrast to the grid-based method. One disadvantage of this approach are the assumptions about the underlying shape types though. Since one can only provide a limited number of analytical forms, this method is based on the assumption that the object can be assembled through the basic shapes in Section 4.2.

In order to perform the next exploration step, a belief state over the shape types and parameters is generated. Further on, we develop an algorithm to find the best action $\boldsymbol{a}_t$ based on the current belief about the shapes. In this setting, *best* means the action which reduces the uncertainty through choosing the most action an action that allows rejecting shape candidates. For this purpose, a measure is defined which provides a rank of locations in the work-space according to their utility if they are explored.

Before turning to this algorithm, the basic data structure has to be defined. As in Section 4.2.1 described, a particle filter $\mathcal{S}_t$ is generated from initial data points. For each shape type in Section 4.2, a member is added to the filter and its parameters are constructed with the initial points. This results in a first guess of the parameters for each shape. Similarly, the initial belief of all shapes is uniformly distributed as

$$\mathbb{P}\left(S_0^i\right) = \frac{1}{n} \qquad 0 < i \leq n$$

where n denotes the number of shapes in the filter. Each of these particles S^i consists of a shape model based on the parameter vector Φ of the respective type. Based on this set $\mathcal{S}_t$, one can define a measure of utility for a group of points. This measure is then used to rank locations in order to choose the next action $\boldsymbol{a}_{t+1}$ of the exploration.

Ranking Function for Shapes

In order to sort the shapes in our particle filter, a function to rank different points is proposed. This rank consists of two parts, the so-called utility and a variance-like measure. The idea behind the utility is the same as the grid-based utility in (4.21). It helps to find locations with a high uncertainty meaning that the different shapes do not align in this particular point. Therefore, the utility of such points is high since it is useful to explore them in order to gain more knowledge at this point.

In general, the ranking operates on a set of points $\mathcal{P} = \{\boldsymbol{p}_i\}$ representing possible contact points on a surface of a shape as shown in Figure 4.3. It depicts two different sets $\mathcal{P}$ for two different and arbitrarily chosen exploration axes. In this case, the set of shapes $\mathcal{S}_t$ contains $n = 3$ shapes, the sphere, cylinder and box type. Since the number of possible contact points depends on the number of shapes in $\mathcal{S}_t$, the cardinality of those sets is equal $\#\mathcal{S}_t = \#\mathcal{P}$.
Now we can define the utility as the first part of the ranking. As (4.21), it is based on [29] and can also be described as the mutual information [24]. Accordingly, the utility for

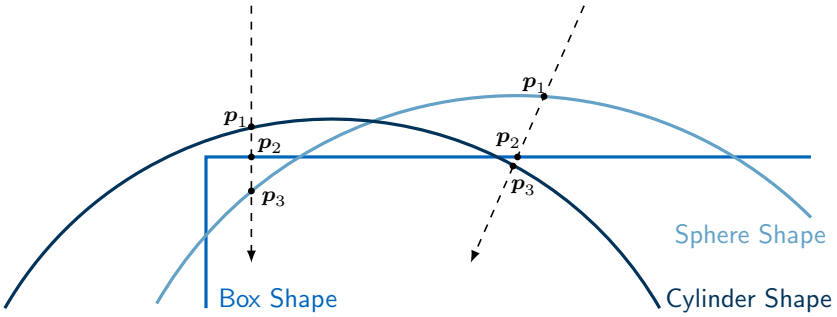


Figure 4.3: Basic Setup of Utility for Shapes. The blue lines represent the shape parametrizations in 2D, the dashed arrows show two different arbitrary exploration movements with their corresponding contact points $\boldsymbol{p}_i$. Each $\boldsymbol{p}_1$ through $\boldsymbol{p}_3$ form a set $\mathcal{P}$.

a set of points $\mathcal{P}$ and a set of shapes $\mathcal{S}_t$ can be calculated as follows

$$U(\mathcal{P}, \mathcal{S}_t) = \sum_{S_j \in \mathcal{S}_t} \sum_{\boldsymbol{p}_i \in \mathcal{P}} \mathbb{P}\left[\boldsymbol{r}_{t+1} = \boldsymbol{p}_i | S_j\right] \mathbb{P}\left[S_j^t | \boldsymbol{x}_t, \boldsymbol{r}_t\right] \ln \left(\frac{\mathbb{P}\left[\boldsymbol{r}_{t+1} = \boldsymbol{p}_i | S_j\right]}{\mathbb{P}\left[S_j^t | \boldsymbol{x}_t, \boldsymbol{r}_t\right]} \right) = \tag{4.26}$$

$$= \sum_{j \in \mathcal{S}_t} \sum_{i \in \mathcal{S}_t} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{(\boldsymbol{p}_i - \boldsymbol{\mu}_j)^2}{2\sigma^2} \right) \mathbb{P}\left[S_j^t | \boldsymbol{x}_t, \boldsymbol{r}_t\right] \ln \left(\frac{\exp\left(\frac{(\boldsymbol{p}_i - \boldsymbol{\mu}_j)^2}{2\sigma^2} \right)}{\sqrt{2\pi\sigma^2}\, \mathbb{P}\left[S_j^t | \boldsymbol{x}_t, \boldsymbol{r}_t\right]} \right)$$

where $\mathbb{P}\left[S_j^t | \boldsymbol{x}_t, \boldsymbol{r}_t\right]$ is the prior belief of shape S_j with the mean $\boldsymbol{\mu}_j$ from the particle filter. The probability $\mathbb{P}\left[\boldsymbol{r}_{t+1} = \boldsymbol{p}_i | S_j\right]$ represents the sensor model as in (4.20). This utility combines the knowledge of the prior belief with the influence of expected measurements and therefore, allows to estimate the expected impact of these measurements. The next step is to establish an overall exploration procedure in order to gather the desired knowledge about the work-space.

Exploration Strategy

In order to explore the work-space, a general step-by-step procedure for the phases of the exploration is provided. This policy is labeled as one exploration step which is repeated several times before the collected data is analyzed. Such a group of exploration steps is called an episode. In line with the process layout described in Figure 2.1, these episodes represent the first part of our overall approach of collecting sensor data. The second step of processing this data is described in the next section. However, the policy of one such exploration step can be outlined with the following phases.

1. Compute next action for the exploration
2. Approach exploration goal
3. Seek contact and apply force to start measurement

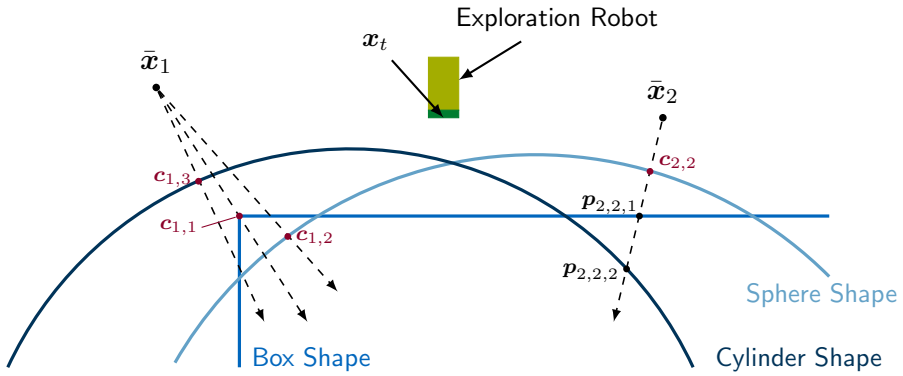


Figure 4.4: Choice of next action with 3 shapes shown in 2D. The points $\bar{\boldsymbol{x}}_1$ and $\bar{\boldsymbol{x}}_2$ are randomly sampled, each $\boldsymbol{c}_{j,k}$ depicts the possible contact points. On $\bar{\boldsymbol{x}}_1$, the closest contact points $\boldsymbol{c}_{1,k}$ with each shape are displayed through the dashed lines. For $\bar{\boldsymbol{x}}_2$ on the right, the contact $\boldsymbol{c}_{2,3}$ with the cylinder shape is shown exemplarily with the respective $\boldsymbol{p}_{2,3,l}$. The points $\boldsymbol{c}_{2,3}$, $\boldsymbol{p}_{2,3,1}$ and $\boldsymbol{p}_{2,3,2}$ are used to calculate the utility for the exploration axis $\bar{\boldsymbol{x}}_2 \rightarrow \boldsymbol{c}_{2,3}$.

4. Update belief and store measured data when measurement is done

The first phase involves calculating the next action with the utility function of (4.26). Hereby, one exploration action $\boldsymbol{a}_t$ consists of a starting point $\boldsymbol{p}_t$ and a point $\boldsymbol{c}_t$ where the next contact is expected. This point $\boldsymbol{c}_t$ is also the location of the second step for applying forces and taking the desired measurements. Since the choice of this action $\boldsymbol{a}_t$ has a great influence on the available knowledge, it makes sense to search the best suitable actions here. Finding such actions leads to high decrease of uncertainty per step and furthermore, decreases the overall effort and number of steps necessary. For obtaining such an action $\boldsymbol{a}_t$, our proposed procedure can be formulated as in Algorithm 2. The inputs are the current position of the robot $\boldsymbol{x}_t$ and the set of shapes $\mathcal{S}_t$. Starting with the position of the robot $\boldsymbol{x}_t$, k new positions $\bar{\boldsymbol{X}} = \{\bar{\boldsymbol{x}}_j\}$ are sampled from a random distribution. For instance, a suitable choice for a distribution is a uniform distribution limited by workspace borders or a maximum distance to the current position $\boldsymbol{x}_t$, denoted as $\Delta\boldsymbol{x}$ relative to the current position in Algorithm 2. These new locations in $\bar{\boldsymbol{X}}$ are treated as candidates for the start of the next exploration action. Hence, we try to find the worthiest among them. For this purpose, the possible contact points with each shape are calculated for each of these candidates $\bar{\boldsymbol{x}}_j$ resulting in a set of contact points $\{\boldsymbol{c}_{j,k}\}$ as illustrated in Figure 4.4. In this case, there are three shape types, box, cylinder and sphere, and $k = 2$ sampled positions $\bar{\boldsymbol{x}}_j$. For $\bar{\boldsymbol{x}}_1$, all possible contact points $\boldsymbol{c}_{1,k}$ are shown which determine the possible exploration axis. On the other side at $\bar{\boldsymbol{x}}_2$, only one exemplary contact $\boldsymbol{c}_{2,2}$ is depicted together with the remaining contact points $\boldsymbol{p}_{2,2,l}$ on that axis. The dashed arrows represent the axis of motion if the corresponding pair $\bar{\boldsymbol{x}}_j$ and $\boldsymbol{c}_{j,k}$ was chosen as next action. As described with (4.26), all contact points on such a line are the input for the calculation of the utility.
This results in kn possible actions with k sampled positions and n shapes. After calculating the utility for all these actions, the action with the highest utility is chosen as the next

Algorithm 2: Obtain Next Exploration Goal Based on Analytical Shape Representation

Input: Position $\boldsymbol{x}_t$, Shapes $\mathcal{S}_t$
Output: Next contact point $\boldsymbol{x}_{t+1}$
$\bar{\boldsymbol{X}} \sim \text{unif}(\boldsymbol{x}_t - \Delta\boldsymbol{x}, \boldsymbol{x}_t + \Delta\boldsymbol{x})$ ▷ Sample k new positions in vicinity $\Delta\boldsymbol{x}$
$\mathcal{U} \leftarrow \{\}$
foreach $\bar{\boldsymbol{x}}_j \in \bar{\boldsymbol{X}}$ **do**
  foreach $S_t^k \in \mathcal{S}_t$ **do**
    $\boldsymbol{c}_{j,k} \leftarrow \texttt{closestPoint}(\bar{\boldsymbol{x}}_j, S_t^k)$ ▷ closest point on shape S_t^k from $\bar{\boldsymbol{x}}_j$
    $\mathcal{X} = \{\boldsymbol{c}_{j,k}\}$ ▷ candidate point set
    // add contact points with remaining shapes on line to candidates set
    foreach $S_t^l \in \mathcal{S}_t, l \neq k$ **do**
      $\mathcal{X} \leftarrow \mathcal{X} \cap \texttt{interSect}(\bar{\boldsymbol{x}}_j, S_t^l)$
    $\mathcal{U} \leftarrow \mathcal{U} \cup \texttt{calcUtil}(\mathcal{X}, \mathcal{S}_t)$ ▷ calculate utility according to (4.26)
$\boldsymbol{x}_{t+1} \leftarrow \underset{\bar{\boldsymbol{X}}}{\text{argmax}}\{\mathcal{U}\}$ ▷ maximize expected utility

exploration step. This means that the combination of $\bar{\boldsymbol{x}}_j$ and $\boldsymbol{c}_{j,k}$ is fixed. As one can see at $\boldsymbol{c}_{2,3}$ in Figure 4.4, the corresponding $\boldsymbol{c}_{j,k}$ may not be the first expected contact point. In this case, the contact point $\boldsymbol{p}_{2,3,1}$ of the sphere is estimated to be the hit first. Hence, the exploration goal of the robot is set to be the point in $\mathcal{P}_{j,k}$ which has the smallest distance to the chosen $\bar{\boldsymbol{x}}_t$. To summarize, the next action is determined by the approaching goal $\bar{\boldsymbol{x}}$ and the expected contact point $\boldsymbol{g}_t$.

Further on in the second phase of a exploration step, the robot moves to $\bar{\boldsymbol{x}}_t$ while changing the orientation of the sensor such that its surface is normal to the exploration axis $\bar{\boldsymbol{x}}_t \rightarrow \boldsymbol{g}_t$. In the third step when the robot has reached the near vicinity of the object $\bar{\boldsymbol{x}}_t$, the controller is again switched to an Cartesian Impedance controller with an attractor point at the desired point of contact. Figure 4.5 depicts the different movements in these two phases. The red arrows show the movements of the approaching phase and the blue arrow shows the procedure while seeking contact. The impedance controller slowly moves forward in the direction of the exploration axis until a contact is registered through the sensor. In this case, the robot applies force on the object until certain conditions are met, e.g. a threshold $\boldsymbol{f}_\tau$ is reached. These conditions are further discussed in section 4.5 as the specific measurement objectives are determined by the overall identification procedure. After the measurement, the last phase updates the belief of the shapes $\mathbb{P}\left(S_t^i\right)$ in $\mathcal{S}_t$ and stores the measured sensor data for further processing. The result is an array with one entry for each of the performed measurements. One entry thereby contains the location of the measurement and the sensor data itself. This marks the end of one exploration step and hence, the procedure starts over with the next step.

In summary, this section describes two different approaches for exploring a work-space by using different stochastic frameworks. The first one utilizes the inference grid method allowing effective data collection and accumulation while no presumptions about the un-

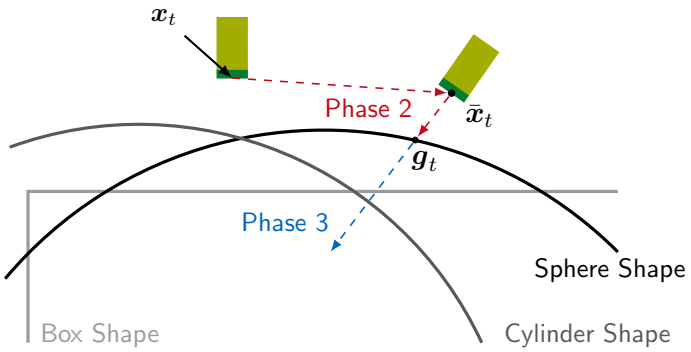


Figure 4.5: Paths of phases 2 and 3 of one Exploration Step. In phase 2, the robot moves to $\bar{x}_t$ while orienting its sensor, then it approaches the expected point of contact g_t. After reaching this point, the robots reduces its velocity waiting for contact in phase 3. If the sensor register a force, the controller switches to the impedance control and performs the measurement.

derlying objects have to be made. On the other hand is the exploration scheme based on analytical shape parametrizations which enables having a continuous representation of the collected data. The result of both methods is a bunch of measurement results containing information about the underlying objects in the environment. Hence, the next section presents different techniques of how to extract the desired information from the data.

4.4 Simulation Environment

In order to relax the inter-dependencies with T6.1 and D6.1, the evaluation of the current state of T6.2 is hold to a simulation environment. Due to its realistic and fast physics-engine, the software framework Mujoco is used. This software framework provides a full-featured simulation pipeline allowing the simulation of multi-joint dynamic systems subject to constraints like contacts and friction [45]. According to the authors, this engine can simulate systems with many degrees of freedom faster than real-time, e.g. the model of a 27 DoF humanoid robot. The provided API allows direct access to the applied forces and torques, joint positions, contact points and forces and so on. With its powerful modelling-language based on XML, it allows controlling several parameters of dynamic behavior including contacts between objects. For instance, the properties and behavior of contacts can be influenced through parameters based on stiffness. The contact constraints are then modeled as soft constraints through a spring-damper system. Furthermore, MuJoCo provides access to friction coefficients for sliding, torsional and rolling friction. Unfortunately, the information about contact forces cannot be partitioned into shear and normal forces but consist of just one accumulated force vector. Such partition would be useful in order to collect data about the surfaces of objects under steady sliding contact.

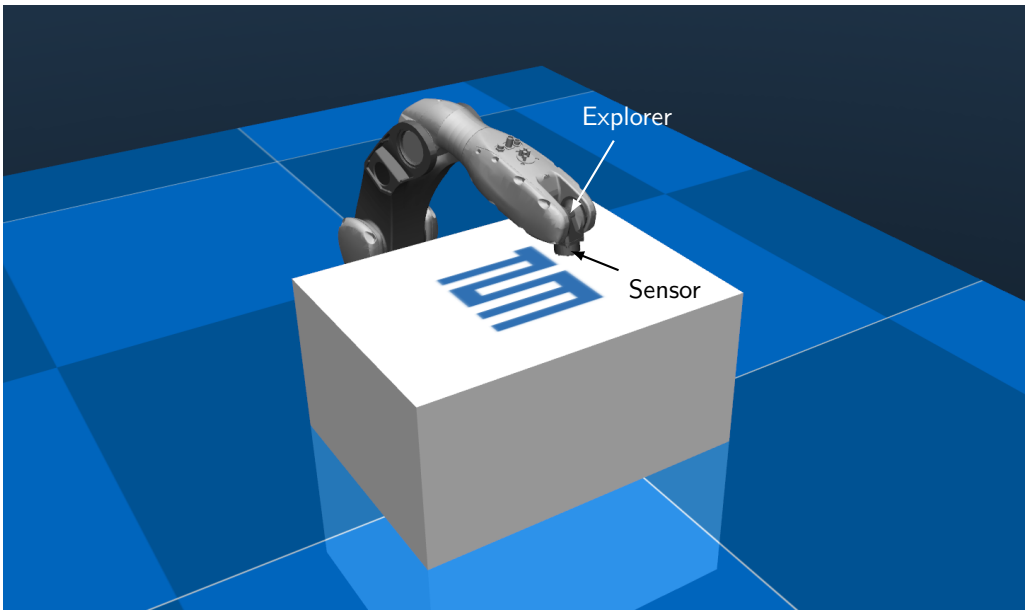


Figure 4.6: A Racer5 0.80 used in simulation with a force sensor attached to the end-effector simulated in Mujoco. Within this deliverable, we explicitly focus on the interaction between the end-effector, denoted as Explorer in the scope of this deliverable, and the actual object, which is given as a cubic box in here.

4.4.1 Robot and Environment

In the simulation environment, a 6-DoF COMAU Racer5 0.80 robot with a force-torque sensor attached on its end-effector is used to explore the work-space as shown in Figure 4.6. This sensor measures normal contact force with all surrounding objects as soon as contact is made. Due to simplifications within the physics engine in Mujoco, the contact force simulations are limited to normal force measurements. Hence, we base our approach only one these measured normal forces.
In order to move the robot across the simulated world, a simple Cartesian position and Impedance controller is implemented, that neglects joint limits. This is intended to decouple the actual control-flow from T6.1 and T6.2, while also allowing the current algorithm to set desired poses and superimposed wrench commands directly in Cartesian space. We focus on the evaluation of the haptic-SLAM algorithm, and forward the discussion of the actual control designs to deliverable D6.1, that tackles the actual application on force-guided manipulation.

4.4.2 Sensors and Collected Data

So far, sensors and the respective data has only been addressed theoretically. There are multiple different techniques of tactile sensors with focus on force, resistive or capacitive measurements. In our approach, a simple force sensor registering normal contact forces is used. This sensor type is used to collect information about the deformation of objects if a force is applied.

Based on the contact force, we track displacement or deformation l of the underlying surface. It is defined as the distance between the first contact point with $f > f_{t1}$ and the point where the force f exceeds a second threshold $f_{t2} > f_{t1}$. This behavior is shown in Figure 4.7 where the left green post shows the moment of initial contact with $f > f_{t1}$ and the right explorer is at the state where the measured force exceeds the threshold f_{t2}. Additionally, the deformation of the object is drawn by the white dashed line. The vector between these points represents the magnitude and direction of the displacement. In addition to the occupancy state in the first layer of the inference grid, we also store this displacement magnitude and direction in the higher layers for further processing. Finally, all layers of a particular cell of the grid can be represented in the vector $[c_t, l] \in \mathbb{R}^4$.

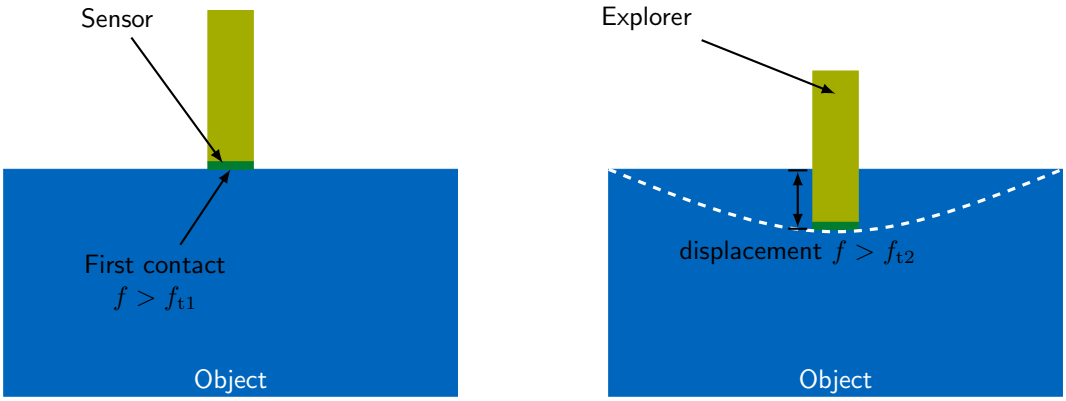


Figure 4.7: Measured displacement for clustering. The left green post shows the explorer at the first contact, the right one when the measured force exceeds the threshold. The white dashed line shows the object deformation through the explorer. The magnitude and direction of the displacement is stored.

The overall strategy for the haptic exploration can be summarized as follows. Starting at one surface of the inference grid, the next starting coordinates in this surface are chosen by either the local or global strategy. From those coordinates, the robot moves into the grid waiting for sensing a contact force. If the sensor reading f exceeds the threshold f_{t1} the robot keeps increasing the force until $f > f_{t2}$ and then the distance between those points is evaluated. The corresponding cells are updated with their occupancy state and the respective displacement values are stored. Afterwards or if the border of the grid was reached without contact, the robot moves back to the starting surface and the next coordinates are calculated based on the two strategies again. When the exploration has achieved a satisfying coverage of the work-space, the identification process is started. The next chapter explains the methods applied to the collected data in the inference grid.

4.5 Object Identification

As outlined in Chapter 2, the object identification part is run on a lower frequency but in return does not rely on an incremental update but rather solves a non-linear regres-

sion problem which allows to estimate object parameterizations. While the evaluation of flexible objects is handled as a special case within chapter 5, this Section outlines the basic decomposition of the unknown object into material types, denoted as Θ. In order to guess the most likely paramters, the collected data set is preprocessed in such a way that only measurements during contact phases are taking into account, which can be directly read from the outlined occupancy grid. Thus, all data from cells that are estimated to be empty, i.e. cells with a value $c_t, \text{occ} < 0$, is omitted for the regression problem.

In the following we give an insight how the simple estimation of material stiffness can contribute towards generating a refined object representation in combination with the iterative grid based approach from the previous Section.

4.5.1 Material Composition and Parameter Estimation

As explained in chapter 3.3, one parameter describing material surfaces is the stiffness. Since it is defined through the ratio of applied force and deformation, the data from the exploration stored in the inference grid can be used to aggregate data points by this parameter. Hence, the stiffness of the underlying material can be estimated through

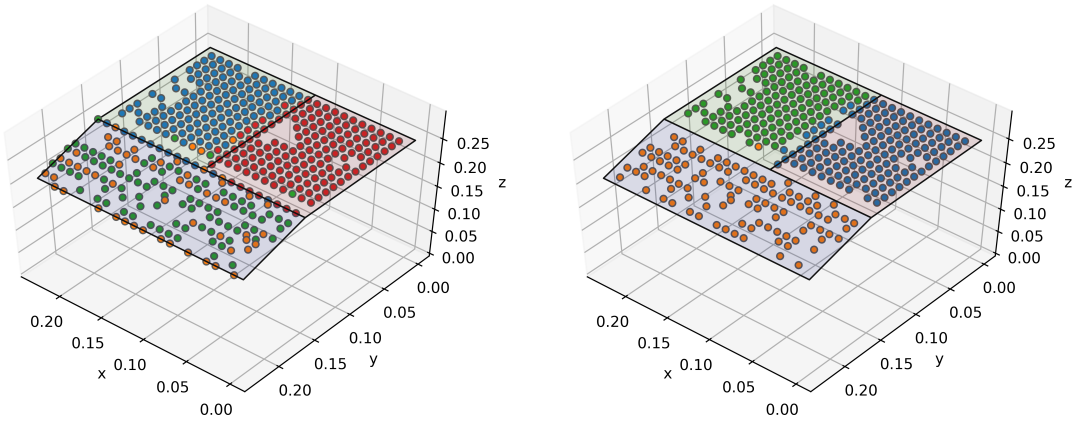
$$k = \frac{f_{\text{t2}}}{|l|} \tag{4.27}$$

where $|l|$ denotes the magnitude of displacement and f_{t2} represents the force threshold determining the displacement from the previous chapter. For classifying this data into different regions of equal parameters, unsupervised clustering algorithms as K-Means are used. K-Means uses a fixed and pre-defined number of classes [11]. It assigns data points to these classes according to the smallest distance to the class centers iteratively. After each iteration, the centers are updated by calculating the mean of all class members. If the members of classes do not change between two iterations the algorithm terminates. Obviously, the number of classes is a free parameter requiring knowledge about the data in advance.

An alternative bypassing this restriction is the Density-Based-Spatial-Clustering for Applications with Noise (DBSCAN) algorithm [25]. The number of clusters cannot be set in advance but can be influenced through the parameters *minPts* and ϵ. It partitions all data points into one of the following categories:

- Core point: has at least $minPts - 1$ other points within a distance of ϵ
- Border point: is not a core point but has at least one core point within a distance of ϵ
- Noise point: is neither a core nor a border point

Clusters are built by aggregating all core points within a distance of ϵ and all associated border points into one cluster. An important feature is that noise points are discarded and therefore not a member of any class. For our case, this is especially helpful since our sensor data may have outlying or noisy measurements. The number of classes and the



(a) K-Means with $n = 4$

(b) DBSCAN with $\epsilon = 0.27$, $minPts = 15$

Figure 4.8: Clustering with K-Means and DBSCAN. Object: three different materials with stiffness $k_3 = 2k_2 = 8k_1$.

threshold for border points are influenced through the parameters $minPts$ and ϵ. The first one has an influence over the number of classes while the latter one has a bigger impact on the number of noise points. This algorithm requires to try different parameters repeatedly to settle with a fitting choice.

Figure 4.8 shows an exemplary result of both clustering algorithms. The underlying object consists of three different materials where their stiffness is related through $k_3 = 2k_2 = 8k_1$. On the left in Figure 4.8(a), the result from K-Means shows that outlying data points prohibit that three accurate clusters can be built. The picture shows the result for $n = 4$ clusters because the algorithm cannot distinguish the material borders for $n = 3$ at all. In contrast, after some moderate parameter tuning with the DBSCAN classification, one can achieve the clusters shown in Figure 4.8(b). The borders between materials almost correspond to the ground truth borders while outlying points are filtered.
These clustering techniques provide a basic overview of the composition the the underlying object in the work-space. Based on this results, one can further estimate the properties of the structure like material borders, shapes and connections in between these materials. Given these clusters, the estimation of shapes can be updated by e.g. re-running the RANSAC algorithm for each cluster individually, thus retrieving new shape estimates, which can then be fed back to the exploration phase.
In contrast to the stiffness identification, the

5 Flexible Probing

This section shows a certain use case in a recyling plant we assume. A display which has plastic shells, such as Flat Panel Display (FPD), is one of mainly considered devices in the HR-Recyclerproject, due to the fact that FPD is one of the most commonly met WEEE material types and it is generated in extremely high quantities. When disassembling these devises, a robot is expected to run into a scenario where the robot has difficulty detaching parts of the casing due to unknown hidden fixtures. As a way of detecting these hidden fixtures, we propose a novel flexible probing technique. It contributes to the information extracting process of unknown geometrical features as well as material parameters as discussed in Section 4.5.1, e.g. hidden fixture locations, Young's modulus (elasticity).

5.1 Problem Description

"Displays/monitors" is one of main Use Cases the HR-Recyclerproject has selected as a specific type of WEEE to demonstrate capabilities of it's proposed system. Recycling plants we aim at have certain processes of detaching a shell from a body of this kind of E-devise.
Typically, the shell of the device is usually held together through multiple fixtures. Such fixtures can be reversible, such as screws, or non-reversible, such as plastic snap-in fixtures. Recycled devices, generally, will be randomly transferred to a work place in front of a robot without their construction plans in the recycling plant unless the robot has the plans beforehand. In this context, the recycled devices are unknown for the robot. When disassembling devices without knowing their construction plans, fixtures are easily missed - either because they are not visible, such as snap-in fixtures, or because they are not properly recognized, such as recessed screws.
In order to disassemble this kind of shell by robots, we take an approach, at first, to

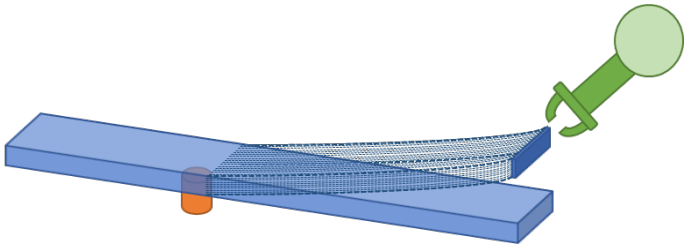


Figure 5.1: Illustration of *Flexible Probing* as one core technology for knowledge acquisition for Haptic SLAM.

open all recognized fixtures by unscrewing all screws and to cut open a perimeter of an enclosure. With this cutting approach, large part of the shell can be efficiently recycled. However, due to such missed fixtures, a cut-off case cannot be totally detached in a safe manner without breaking it. In this context, the robot should know where can be location of missed fixtures without the construction plans of the unknown devices.
After cutting open the perimeter, a robot gets an arbitrarily shaped plate like object, which can be shaped as a rectangular through controlled cutting by a manipulator robot having cutting tool on an end-effector of the manipulator. Then, if more than one missed fixture is exist with the cut-off rectangular object but invisible from vision sensors, the manipulator executes a human mimicking bending action. If there is no fixture, the cut-off object might be detached immediately from the body of the devise and carried to proper location for segregation.

5.1.1 Problem as a Practical Matter

A risk, that the cut-off case is broken in unpredictable manner and the fixtures are not segregate from the case, never disappears unless the missed fixtures is still remaining. If robot is forced to detach the case with breaking it in spite of existence of missed fixtures, debris can be fried and hit to delicate modules of the robot as well as human. Efficient segregation in a recycle context is also obstructed.
One of the reasons for missing fixtures is in a situation where the fixtures are physically hidden by the shell structure and never visible from vision sensors. Another reason would depend on performance of an image processing system in a situation where all fixtures are in view from vision sensors.
In the latter case, all fixtures might be able to be detected if a surface of the shell is completely examined by the vision sensor, e.g. controlling positions or orientations of a camera, zoom of the camera or etc., however, it takes long time accordingly. It also seems almost theoretically impossible that the system finds all fixtures perfectly only by existing vision sensors. Our bending approach contributes to solve this problem. The robot tries at first to find all fixtures by vision sensors perfectly but objectively imperfectly, wherein throughput should have priority. Then the robot bends the case, gets haptic information and identify location of the missed fixtures after the cutting-off process. If the robot can identify the location of the missed fixtures as a certain area, it can do an intensive search by the vision sensor only in the certain area with shorter span of time.
An allocation balance problem between vision and haptic approach can be also approached with human mimicking idea, wherein throughput and energy are optimized considering effects on human co-worker, however, methods for human mimicking behavior in this report is focused on the bending situation

5.1.2 Mimicking Human Behavior

We take a strategy to give a robot manipulator resemblance to human behavior in way of estimating location of hidden fixture attached with a plate like object. After cutting open the perimeter of the case, humans might pry the device case along the cut opening and flexing the cut-off case. Then, anyone who is used to dealing with elastic plate like objects by fingers would naturally think that, there is missed fixture located not around

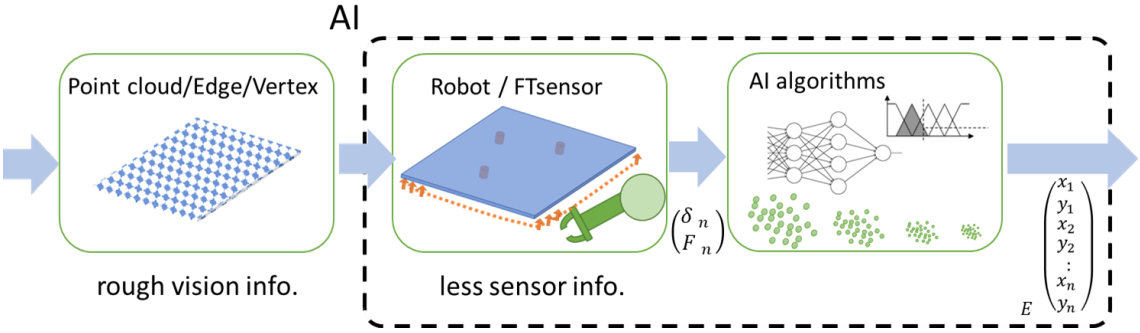


Figure 5.2: Overview of *Flexible Probing* as one core technology for knowledge acquisition for Haptic SLAM.

the finger but far from the finger if an edge of the cut-off case goes up when pushing up and bending the case by the finger at a location of the edge. Therefore we assume copying the human gesture in the situation. Inner operation process is, however, still impossible to copy humans. Hence, assuming AI reasoning is the most useful for representing human intelligence artificially, we investigate methods based on expert system, probabilistic approach and machine learning approach.
Humans accustomed to treating those materials also reason; if a sense of pushing or bending is harder, location of the missed fixture might be near from pushing location. Then humans would search missed fixtures in a limited area narrowed through estimation by information through the flexing so that the missed fixture can be found and removed easier and faster. The area, therefore, is a target of estimation in the following proposed method.

5.2 Pilot Study of Existing Techniques

Under the basic concept of mimicking the behavior of humans when manipulating different object, this section details methods for identifying location of missed fixtures which are invisible, with an overview of the methods shown in Figure 5.2. In this method, like human might do so, force information at the edge of the plate object is used for estimating the location of the hidden fixture. In robotics, F/T sensor or skin sensor is generally used to sense force at arbitrary position in space. We assume a robot can detect force information at the edge by manipulating end-effector equipped with this kind of force sensor. We investigate different methods for experimental identification of hidden fixtures, including Fuzzy Inference, particle filters and nonlinear regression. All methods are based on a comparison of measurements taken by the robot and simulation using the current object knowledge base. Figure 5.1 illustrates one of these experiments.
Given an unknown hidden fixture at location $\boldsymbol{x}_f$, the robot pulls the elastic casing of the object at a certain position $\boldsymbol{x}_m$ on the surface and takes a measurement of the force f_m and displacement δ_m. The measurement 3-tuple $(\boldsymbol{x}_m, f_m, \delta_m)$ with the current estimated location $\boldsymbol{x}_f^e$ is then used to improve the estimated location, by minimizing

$$\underset{\boldsymbol{x}_f^e}{\text{minimize}} \quad \delta_m - \delta_s(\boldsymbol{x}_f^e, \boldsymbol{x}_m, f_m)$$

the error between the simulated deflection δ_s and the measured deflection δ_m.

5.2.1 Finite Element Method

is a widely used and reliable technique for simulating elastic structures. In this section, effectiveness of the developed method is evaluated with finite element analysis. This method is based on a hypothesis that there is particular relationship between fixed and externally provoked displacement on the plate object. Simulations are carried out withFEM. This technique allows for simulation of elastic bending under external applied forces, of complex bodies.
Our initial simulations are conducted on a simple 2D plane. However, real 3D geometries of the objects at hand are expected to be delivered from Visual perception for real use at a later stage in this project.
Figure 5.3, shows simulation results for a 2D metal plate with a estimated fixture, under different extern load conditions. Simulations are carried out for external loads moving at 2 perpendicular edges of the plane. This extensive body of simulations is stored as FEM-Database for sampling based approaches.

5.2.2 Identification methods

For the identification stage that minimizes the discrepancy between simulation results and measurement, different methods are considered.

Neural Networks Approach Assuming that the object at hand is known, a object database as depicted in Figure 5.3 can be constructed. It consists of FEM results under a combination of different fixture locations and externally applied forces. Using this database, a neural network is trained to find the map

$$\text{FEM} : (\boldsymbol{x}_m, f_m, \delta_m)^{\text{sim}} \mapsto \boldsymbol{x}_f^{\text{sim}},$$

i.e. the simulated input measurements of the 3-tuple onto the simulated fixture location $\boldsymbol{x}_f$. Once the neural network is sufficiently trained, it can be used on-line, to feed it with a real measurements 3-tuple to estimate the fixture

$$\text{NN} : (\boldsymbol{x}_m, f_m, \delta_m) \mapsto \boldsymbol{x}_f^e.$$

Figure 5.4 illustrates this scheme.

Regression approach While the neural network approach need to learn a network for every object at hand, applying nonlinear regression can conduct FEM-simulations on-demand instead. Nonlinear optimization solvers can directly be applied to find the best candidate for a hidden fixture location, that minimizes the error

$$\underset{\boldsymbol{x}_f^e}{\text{minimize}} \quad \delta_m - \delta_s(\boldsymbol{x}_f^e, \boldsymbol{x}_m, f_m).$$

The objective function evaluation $\delta_s(\boldsymbol{x}_f^e, \boldsymbol{x}_m, f_m)$ applies FEM simulation once per iteration. Based on the displacement error, the solvers automatically moves the estimated fixture position $\boldsymbol{x}_f^e$ to improve the objective function.

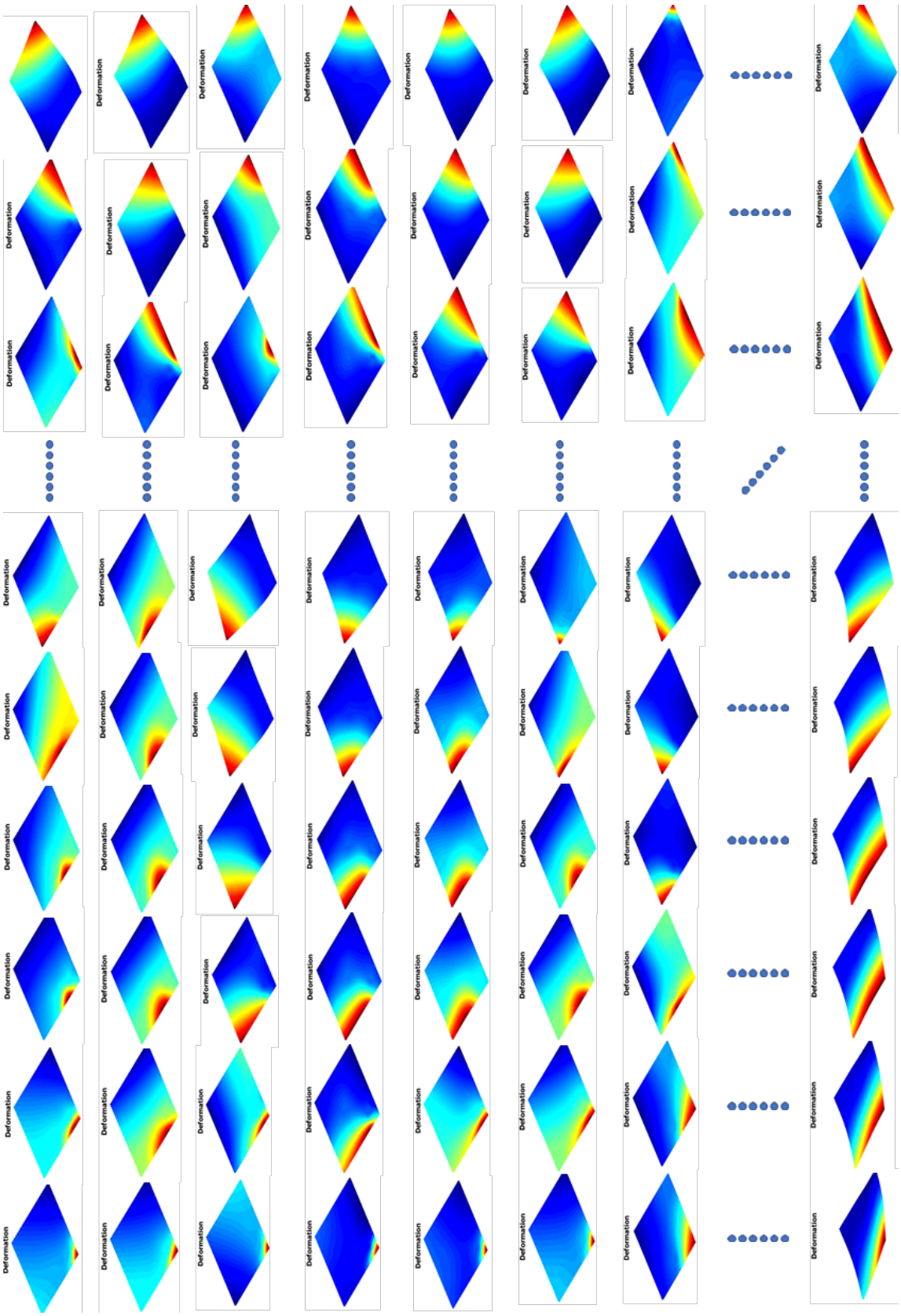


Figure 5.3: Results of 2D finite element analysis for varying probing (horizontal) and fixture locations (vertical).

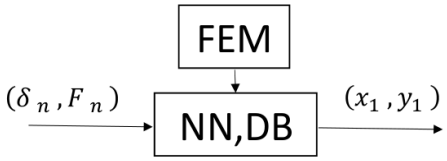


Figure 5.4: Scheme of a neural network to estimate fixture locations.

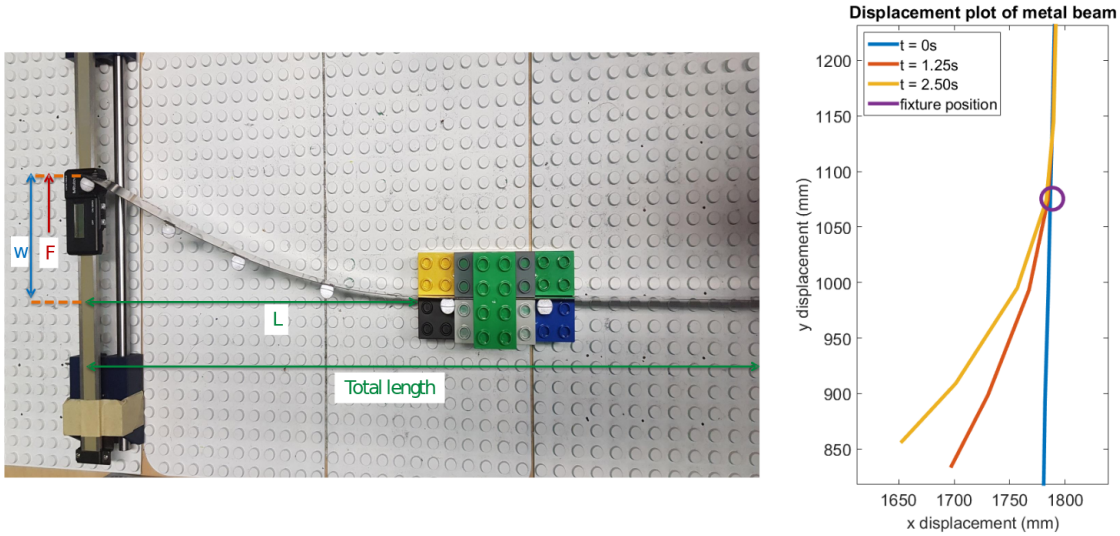


Figure 5.5: Experiments on detecting hidden fixtures in a 1D case.

5.3 Preliminary Experimental Results

Figure 5.5 shows the experimental setup of a 1D elastic displacement experiment. The position-controlled robot provokes a desired displacement δ_m, and measures the resulting force f_m due to the elastic potential of the metal sheet. A Qualisys Motion Tracking system is applied for tracking the full profile of the beam, for in-depth comparison with the simulated beam profile. Fixtures are then estimated.
Experimental results for 1D examples verify the strategy and hypothesis that there is particular relationship between fixed and externally provoked displacement on the plate object.
Experiments with 2D simple objects e.g. planes, as well as real case objects with comparison against results of FEM are yet to be conducted.

6 Summary

An application of automation technologies have shown great success within industry, wherein expert know-hows are explicitly available. However, the utility of these devices are only evident in cases where a perfect knowledge of the environment, the task procedure and thus last but not least the object characteristics are well known and defined in advance. The Hybrid Human-Robot RECYcling plant for electriCal and eLEctRonic equipment (HR-Recycler) instead seeks to allow robots to disassemble unknown objects within industrial environments. In such scenarios, a generic disassembly task faces tremendous uncertainties and knowledge gaps. In contrast to an assembly process, in which single components are iteratively combined to form a final product, the inner composition of an object is in general unknown. Despit the fact that computer vision and low-level sensor technologies have brought tremendous progress in identifying and monitoring objects, the abstracted geometry of objects are lacking precision and exact parameterization. Thus, this deliverable proposed a framework, which allows robots to incrementally gather haptic measurements from active interaction with the objects and update the current estimation of the object. In order to set this method in relation to recent advancements, this deliverable laid out the state-of-the-art in haptic data acquisition and estimation in the scope of haptic Simultaneous Localisation and Mapping (SLAM) and flexible probing. In contrast to these methods, the framework outlined in the remainder of this deliverable exceeds the haptic identification methods in terms of mere identification of object geometries. Rather, the proposed framework combines geometric data acquisition with the identification problem of the material parameterization. The deliverable further points out how this object identification problem can then update the geometry refinement process by e.g., rejecting false geometric estimations. The iterative update of geometry estimation with an additional stiffness classification has been proposed as a closed loop system and evaluated within a simulated environment. Further to this, the flexible probing, which has been laid out as a scientific challenge within Task (T)6 in the HR-Recycler project, forms a special case of the parameter identification problem. The haptic regression techniques were outlined and evaluated on a relaxed problem, where the exact shape is known while an inner fixture point is out of scope by passive data acquisition methods (e.g. vision). In contrast to the haptic SLAM framework evaluations, initial toy experiments served as a proof of concept on how a robot can improve the estimation over such hidden fixture points.

The presented results serve as the baseline towards delivery Deliverable (D)6.4 and lays out the fundamentals of T6.2. The delivery of these findings will further contribute towards the outcome of T6.1 and thus D6.1, as the according control designs need to meet the requirements for the data acquisition methods sketched in this delivery. In order to

allow robots to actually identify unknown objects and successfully disassembling them, the presented framework will be combined into a closed loop form, that is then to be applied onto the actual hardware platform.

Bibliography

[1] Peter K. Allen and Kenneth S. Roberts. Haptic object recognition using a multi-fingered dextrous hand. In *Proceedings of the 1989 IEEE International Conference on Robotics and Automation, Scottsdale, Arizona, USA, May 14-19, 1989*, pages 342–347. IEEE Computer Society, 1989.

[2] H. Altenbach. *Holzmann/Meyer/Schumpich Technische Mechanik Festigkeitslehre*. 2018.

[3] J. Awrejcewicz. *Classical Mechanics*, volume 28. 2012.

[4] T. Bailey and H. Durrant-Whyte. Simultaneous localization and mapping (SLAM): Part II. *IEEE Robotics and Automation Magazine*, 13(3):108–117, 2006.

[5] Ruzena Bajcsy and Constantinos Tsikos. Assembly via disassembly: A case in machine perceptual development. In *The Fifth International Symposium on Robotics Research*, pages 355–361, Cambridge, MA, USA, 1990. MIT Press.

[6] Mohamad Bdiwi, Aquib Rashid, and Matthias Putz. Autonomous disassembly of electric vehicle motors based on robot cognition. In *2016 IEEE International Conference on Robotics and Automation, ICRA 2016, Stockholm, Sweden, May 16-21, 2016*, pages 2500–2505, 2016.

[7] F. Behbahani. *Reverse-Engineering The Visual and Haptic Perceptual Algorithms in The Brain*. Doctor of philosophy, Imperial College London, 2016.

[8] F. Behbahani, G. Singla-Buxarrais, and A. Faisal. Haptic SLAM: An Ideal Observer Model for Bayesian Inference of Object Shape and Hand Pose from Contact Dynamics. In *Haptics: Perception, Devices, Control, and Applications*, pages 146–157, Cham, 2016. Springer International Publishing.

[9] F. Behbahani, R. Taunton, A. Thomik, and A. Faisal. Haptic SLAM for context-aware robotic hand prosthetics - Simultaneous inference of hand pose and object shape using particle filters. *International IEEE/EMBS Conference on Neural Engineering, NER*, 2015-July(1229297):719–722, 2015.

[10] W. Bergmann Tiest and A. Kappers. Cues for haptic perception of compliance. *IEEE Transactions on Haptics*, 2(4):189–199, 2009.

[11] C. Bishop. *Pattern recognition and machine learning*. Springer Verlag New York, New York, 1 edition, 2006.

[12] F. Bourgault, A. Makarenko, S. Williams, B. Grocholsky, and H. Durrant-Whyte. Information Based Adaptive Robotic Exploration. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, (October 2002):540–545, 2002.

[13] Timothy Bretl and Zoe McCarthy. Quasi-static manipulation of a kirchhoff elastic rod based on a geometric analysis of equilibrium configurations. *I. J. Robotics Res.*, 33(1):48–68, 2014.

[14] N. Chen, H. Zhang, and R Rink. Edge tracking using tactile servo. In *Proceedings 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems. Human Robot Interaction and Cooperative Robots*, volume 2, pages 84–89. IEEE Comput. Soc. Press, 1995.

[15] Wei Hua Chen, Kathrin Wegener, and Franz Dietrich. A robot assistant for unscrewing in hybrid human-robot disassembly. In *2014 IEEE International Conference on Robotics and Biomimetics, ROBIO 2014, Bali, Indonesia, December 5-10, 2014*, pages 536–541, 2014.

[16] S. Decherchi, P. Gastaldo, R. Dahiya, M. Valle, and R Zunino. Tactile-data classification of contact materials using computational intelligence. *IEEE Transactions on Robotics*, 27(3):635–639, 2011.

[17] Nicholas M. DiFilippo and Musa Jouaneh. A system combining force and vision sensing for automated screw removal on laptops. *IEEE Trans. Automation Science and Engineering*, 15(2):887–895, 2018.

[18] Nicholas M. DiFilippo and Musa Jouaneh. Using the soar cognitive architecture to remove screws from different laptop models. *IEEE Trans. Automation Science and Engineering*, 16(2):767–780, 2019.

[19] A. Doucett, N. Freitast, and S. Russent. Rao-Blackwellised Particle Filtering for Dynamic Bayesian Networks. *Proceedings of the Sixteenth conference on Uncertainty in artificial intelligence*, pages 176–183, 2000.

[20] H. Durrant-Whyte and T. Bailey. Simultaneous localization and mapping: Part I. *IEEE Robotics and Automation Magazine*, 13(2):99–108, 2006.

[21] A. Elfes. A tesselated probabilistic representation for spatial robot perception and navigation. *Proceedings of the NASA Conference on Space Telerobotics*, pages Volume 2 p 341–350, 1989.

[22] A. Elfes. Using Occupancy Grids for Mobile Robot Perception and Navigation. *Computer*, 22(6):46–57, 1989.

[23] A. Elfes. Dynamic control of robot perception using multi-property inference grids. In *Proceedings 1992 IEEE International Conference on Robotics and Automation*, pages 2561–2567. IEEE Comput. Soc. Press, 1992.

[24] A. Elfes. Robot navigation: Integrating perception, environmental constraints and task execution within a probabilistic framework. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 1093, pages 93–130, 1996.

[25] M. Ester, H. Kriegel, J. Sander, and X. Xu. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. Technical report, 1996.

[26] J. Fishel and G. Loeb. Bayesian exploration for intelligent identification of textures. *Frontiers in Neurorobotics*, 6(JUNE):1–20, 2012.

[27] D. Hegazy and J. Denzler. Combining Appearance and Range Based Information for Multi-class Generic Object Recognition. In Eduardo Bayro-Corrochano and Jan-Olof Eklundh, editors, *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, pages 741–748, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.

[28] K. Higashi, S. Okamoto, and Y. Yamada. What is the Hardness Perceived by Tapping? In Fernando Bello, Hiroyuki Kajimoto, and Yon Visell, editors, *Haptics: Perception, Devices, Control, and Applications*, pages 3–12, Cham, 2016. Springer International Publishing.

[29] B. Julian, M. Angermann, M. Schwager, and D. Rus. Distributed robotic sensor networks: An information-theoretic approach. *International Journal of Robotics Research*, 31(10):1134–1154, 2012.

[30] T. Korthals, J. Exner, T. Schopping, and M. Hesse. Semantical occupancy grid mapping framework. *2017 European Conference on Mobile Robots, ECMR 2017*, 2017.

[31] T. Lehmann, C. Rossa, N. Usmani, R. S. Sloboda, and M. Tavakoli. A real-time estimator for needle deflection during insertion into soft tissue based on adaptive modeling of needle-tissue interactions. *IEEE/ASME Transactions on Mechatronics*, 21(6):2601–2612, Dec 2016.

[32] S. Luo, J. Bimbo, R. Dahiya, and H. Liu. Robotic tactile perception of object properties: A review. *Mechatronics*, 48:54–67, 2017.

[33] Shan Luo, Wenxuan Mou, Kaspar Althoefer, and Hongbin Liu. iclap: shape recognition by combining proprioception and touch sensing. *Auton. Robots*, 43(4):993–1004, 2019.

[34] U. Martinez-Hernandez, G. Metta, T. Dodd, T. Prescott, L. Natale, and N. Lepora. Active contour following to explore object shape with robot touch. *2013 World Haptics Conference, WHC 2013*, pages 341–346, 2013.

[35] Hiromi Mochiyama. Model validation of discretized spatial closed elastica. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2016, Daejeon, South Korea, October 9-14, 2016*, pages 5216–5223. IEEE, 2016.

[36] M. Montemerlo, S. Thrun, D. Koller, and B. Webreit. FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem. *Aaai/iaai*, 593598:593–598, 2002.

[37] Stefan Escaida Navarro, Nicolas Gorges, Heinz Wörn, Julian Schill, Tamim Asfour, and Rüdiger Dillmann. Haptic object recognition for multi-fingered robot hands. In *2012 IEEE Haptics Symposium, HAPTICS 2012, Vancouver, BC, Canada, March 4-7, 2012*, pages 497–502. IEEE, 2012.

[38] Z. Pezzementi, E. Plaku, C. Reyda, and G. Hager. Tactile-object recognition from appearance information. *IEEE Transactions on Robotics*, 27(3):473–487, 2011.

[39] M. Schaeffer and A. Okamura. Methods for intelligent localization and mapping during haptic exploration. pages 3438–3445, 2004.

[40] T. Shibata and K. Tanie. Conceptual design of disassembly automation system for automated manufacturing with ecological recycling. In *Proceedings of 1993 IEEE/Tsukuba International Workshop on Advanced Robotics*, pages 25–28, Nov 1993.

[41] K. Suwanratchatamanee, M. Matsumoto, and S. Hashimoto. Robotic tactile sensor system and applications. *IEEE Transactions on Industrial Electronics*, 57(3):1074–1087, 2010.

[42] K. Suwanratchatamanee, R. Saegusa, M. Matsumoto, and S. Hashimoto. A simple tactile sensor system for robot manipulator and object edge shape recognition. *IECON Proceedings (Industrial Electronics Conference)*, pages 245–250, 2007.

[43] Ryo Takano, Hiromi Mochiyama, and Naoyuki Takesue. Real-time shape estimation of kirchhoff elastic rod based on force/torque sensor. In *2017 IEEE International Conference on Robotics and Automation, ICRA 2017, Singapore, Singapore, May 29 - June 3, 2017*, pages 2508–2515. IEEE, 2017.

[44] S. Thrun, W. Burgard, and D. Fox. *Probabilistic robotics*. MIT Press, 2005.

[45] Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2012, Vilamoura, Algarve, Portugal, October 7-12, 2012*, pages 5026–5033, 2012.

[46] Deepak Trivedi and Christopher D. Rahn. Model-Based Shape Estimation for Soft Robotic Manipulators: The Planar Case. *Journal of Mechanisms and Robotics*, 6(2), 03 2014. 021005.

[47] X. Wang, H. Gao, O. Kaynak, and W. Sun. Online deflection estimation of x-axis beam on positioning machine. *IEEE/ASME Transactions on Mechatronics*, 21(1):339–350, Feb 2016.

[48] Kathrin Wegener, Wei Hua Chen, Franz Dietrich, Klaus DrÃ¶der, and Sami Kara. Robot assisted disassembly for the recycling of electric vehicle batteries. *Procedia CIRP*, 29:716 – 721, 2015. The 22nd CIRP Conference on Life Cycle Engineering.

[49] D. Xu, G. Loeb, and J. Fishel. Tactile identification of objects using Bayesian exploration. *Proceedings - IEEE International Conference on Robotics and Automation*, pages 3056–3061, 2013.

[50] Bing-Ran Zuo, Alexander Stenzel, and Günther Seliger. Flexible handling in disassembly with screwnail indentation. In *Proceedings of the 2000 IEEE International Conference on Robotics and Automation, ICRA 2000, April 24-28, 2000, San Francisco, CA, USA*, pages 3681–3686. IEEE, 2000.