Tactile SLAM : Simultaneous Reconstruction And Localization of Objects For Robotic Manipulation

Ghani Kissoum

Tactile SLAM: Simultaneous Reconstruction And Localization of Objects For Robotic Manipulation

---

Author:
Ghani KISSOUM

Supervisor:
Prof. Véronique PERDEREAU

A thesis submitted in fulfillment
of the requirements for the degree of
Doctor of Philosophy

École doctrorale (S.M.A.E.R) Sciences Mécaniques, Acoustique, Électronique et Robotique de Paris
Institut des Systèmes Intelligents et de Robotique

Paris, France

January 2023

Tactile SLAM: Simultaneous Reconstruction And Localization of Objects For Robotic Manipulation,

Author:
Ghani KISSOUM

Supervisor:
Prof. Véronique PERDEREAU

Institute: ISIR
Sorbonne Université, Paris, France

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ALGORITHMS

ABSTRACT

A manipulator robot intended for grasping and manipulating objects requires a perception system to know the characteristics of the object to be manipulated in order to correctly plan the assigned tasks. Robotic systems often use vision to get information about the shape and location of an object. However, vision is no longer operational in some situations, such as in dark and dusty environments. In these situations, tactile perception is more appropriate for seeking information on manipulated objects. It allows the acquisition of contact information through multiple contacts with the object and uses them to build a model of its shape.

This thesis work focuses on reconstructing object models from tactile information collected by contact sensors installed on the fingertips of a robotic hand through successive grasping and manipulating actions of the object. We are first interested in precisely modeling the surface, or a part of the object's surface, using the collected information. We have chosen a probabilistic method called "Gaussian Process Implicit Surface" (GPIS), which can efficiently describe different geometric forms from sparse tactile information.

Secondly, we are interested in the tracking of objects during exploration in order to ensure correct registration of the different parts of the object. The tracking is handled using a particle filter algorithm that re-estimates and corrects the object's pose at each new acquisition of tactile information. However, recursive pose tracking cannot be continuously guaranteed due to the sparsity of tactile information and the lack of a match between recent and previously collected data. We use a pose graph optimization method to overcome the limitations of the recursive estimation method.

Finally, the last part of this thesis focuses on developing methods for the autonomous acquisition of tactile information (autonomous exploration) with the robot. The developed methods are based on evaluating the uncertainty of the object's shape and the decomposition of its surface into areas to explore using a clustering algorithm. We use techniques from information theory to predict and quantify the information gained from each exploration action performed by the robot.

FRENCH ABSTRACT

Un robot manipulateur destiné à la saisie et à la manipulation d'objets nécessite un système de perception pour connaître les caractéristiques de l'objet à manipuler, afin de planifier correctement les tâches assignées. Les systèmes robotiques utilisent souvent la vision pour acquérir des informations sur la forme et la localisation d'un objet. Or, il existe des situations où la vision n'est plus opérationnelle, comme dans les environnements sombres et poussiéreux. Dans ces situations, la perception tactile est plus appropriée pour chercher des informations sur les objets manipulés. Elle permet, à travers de multiples contacts avec l'objet, d'acquérir des données sur sa surface et ainsi construire un modèle de sa forme.

Ce travail de thèse s'intéresse à la reconstruction de modèles d'objets, à partir d'informations tactiles collectées par des capteurs de contact installés sur les bouts des doigts d'une main robotique, à travers des actions de saisie et de manipulation successives de l'objet. Nous nous intéressons d'abord à la question de modélisation précise de la surface ou d'une partie de la surface de l'objet à partir des informations collectées. Pour cela, le choix a été porté sur une méthode probabiliste appelée "Gaussian Process Implicit Surface" (GPIS) qui offre la possibilité de décrire efficacement différentes formes géométriques, à partir de quelques informations tactiles.

Ensuite, nous nous intéressons au suivi de la pose d'objets pendant l'exploration, afin d'assurer un recalage correct des différentes parties de l'objet. Cela a été traité à l'aide d'un filtre à particules qui estime et corrige la pose de l'objet à chaque nouvelle acquisition d'informations tactiles. Cependant, le suivi récursif de la pose ne peut être continuellement assurée en raison de la rareté des informations et de l'absence d'une correspondance entre les nouvelles données et les données collectées précédemment. Nous présentons une méthode d'optimisation de graphe de poses pour dépasser les limites de la méthode d'estimation récursive.

Enfin, la dernière partie de cette thèse est dédiée au développement de méthodes d'acquisition autonome d'informations tactiles (exploration autonome) avec le robot. Les méthodes développées s'appuient sur une évaluation de l'incertitude sur la forme de l'objet exploré, et la décomposition de sa surface avec un algorithme de clustering en des parties à explorer. Nous faisons appel à des techniques issues de la théorie de l'information qui permettent de prédire et quantifier le gain d'information de chaque action d'exploration réalisée par le robot.

ACRONYMS

DOF	Degrees Of Freedom
GP	Gaussian Process
GPIS	Gaussian Process Implicit Surface
ICP	Iterative Closest Point
IGEF	Information Gain Estimation Function
IK	Inverse Kinematics
KLD	Kullback–Leibler Divergence
MPF	Manifold Particle Filter
NBT	Next Best Touch
PF	Particle Filter
PGO	Pose Graph Optimization
SE	Square Exponential
SLAM	Simultaneous Localization And Mapping
TP	Thin-Plate

SYMBOLS

We summarize here the principal mathematical functions and symbols used in this thesis:

Symbol	Description
$\mathbf{X}$	Set of random variable
$\mathbf{x}$	Random variable
$x^{[i]}$	i-th particle
$\hat{x}$	Estimated pose of an object
$k(,)$	kernel
K	Covariance matrix
$f(.)$	GPIS function
V	Variance of GPIS
m	Mean of GPIS
$c_i = \{p_i, n_i\}$	Contact data
p_i	Contact point
n_i	Contact normal
w_i	Weight of the i-th particle
σ_n	measurements noise
$P(.)$	Probability function
y	Contact observation

1 INTRODUCTION

Robotic manipulators are increasingly deployed in a variety of domains like industry, services, exploration, and medicine. As the tasks required of the robots become more complex and risky, robotic manipulators need to be more complex mechanical structures and equipped with advanced sensors and complex algorithms. These robotic manipulators are made to interact with their environments: recognize objects of the environment, assess humans, and handle objects. These tasks require efficient algorithms for perception, planning and control.

With robots acting in unstructured environments, accurate perception is essential to the success of a desired task. It provides important information for planning algorithms and object recognition. For example, we consider the task of a robotic manipulator, which has to grasp an object, re-orient it, and insert it in a desired place. This task requires the combination of several capabilities of the robot: localizing the object to grasp, planning the motion of the arm toward the object, grasping the object, and controlling the contact forces. Then the robot has to ensure that the object will be released correctly in the desired place. In all these steps, the correct perception of the object is a key factor for the success of the intended task.

In most existing robotic applications, vision systems are widely used to gather information about the environment. Various camera technologies provide rich information, and a large number of algorithms are available to process the vision data and extract the relevant information. However, vision systems are not efficient in many situations. Occlusions are the most known obstacle to vision that prevents obtaining the desired information. The second drawback of vision is the sensitivity to light conditions: a strong luminosity or a lack of luminosity can cause a loss of image quality. Thereby it does not make it possible to extract important information from the vision data. Also, transparent objects are not easily seen in vision data. Figure 1.1 depicts some situations where vision is ineffective for perceiving manipulated objects.

Another perception modality is needed to overcome this issue. Tactile sensing is an adequate alternative to vision. Equipping robotic manipulators with tactile sensors allows for obtaining information about an object. Even if this information is local, it offers more precision about the geometric and physical properties of the object. The technology for tactile sensing has grown and produced advanced tactile sensors that are very practical for robotic applications.

Figure 1.1: Examples of situations where visual perception is not effective in getting information about the manipulated object. From left to right: occluded object, transparent object, lack of luminosity. (The first picture on the left is adapted from ©https://uni-bielefeld.de/)

In this thesis, we are interested in the problem of exploring unknown objects in order to recover their shape using tactile sensors. We aim to develop a framework for building accurate models of objects for use in planning and manipulation tasks.

1.1 MOTIVATIONS

As advances in robotic applications allow them to perform a variety of tasks, robots are facing new and challenging situations. Robotic manipulators are still challenged by new tasks such as extraterrestrial or submarine exploration (Figure 1.2). In such missions, teleoperated robots can be used to collect desired objects from a shipwreck lying on the bottom of the sea or to gather rocks in an extraterrestrial operation. The visual conditions may not be advantageous in these missions due to the presence of unpredictable obstacles such as dust and lack of luminosity. The limitations of visual perception in those situations make tactile sensors more effective than cameras and lasers.

Advances in tactile sensing technology have produced efficient tactile sensors for robotics[1]. On the other hand, robotic manipulators are designed with greater dexterity. This opens up possibilities to manipulate objects with high dexterity and to obtain precise information about the objects being manipulated by means of tactile sensors. In this context, our goal is to improve the capabilities of robotic manipulators in exploring unknown objects to obtain accurate geometrical information about them.

We are interested in the development of a framework for the recovery of the geometry of unknown objects explored with a robotic hand (Shadow hand) equipped with tactile sensors on its fingertips. We aim to make this framework robust against noisy and uncertain data and appropriate for different geometric shapes.

1.2 RELATED WORKS

The use of tactile sensing in the robotic field has contributed to significant improvements in different applications: interaction with the environment, control of manipulation tasks,

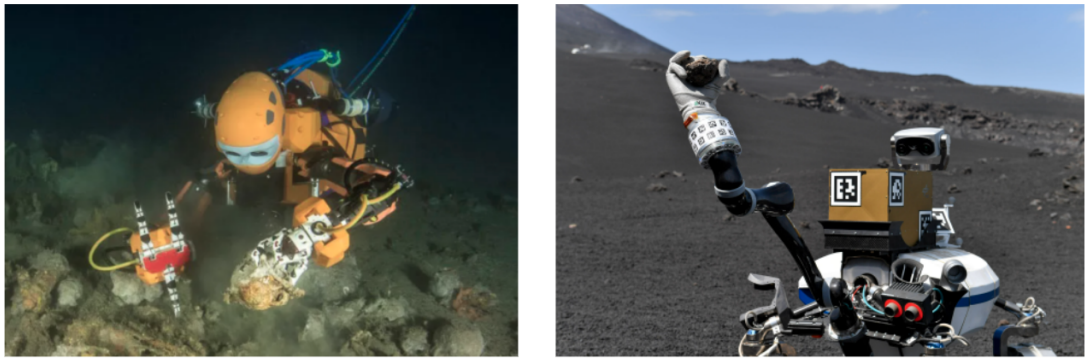


Figure 1.2: Examples of potential applications of tactile exploration of unknown objects. Left: Submarine humanoid handling archaeological object (©https://spectrum.ieee.org/). Right: Moon exploring robot taking a rock (©https://www.dlr.de/)

and object recognition. We review here the most important works related to the use of tactile sensors within robotic manipulators.

Devices for touch sensing

Sensing contact in robotic manipulation applications is technologically made with various principles. The most commonly used are the tactile sensors [2] based on capacitive or piezo-electric effects. They can be used to detect a single contact or managed in an array to construct tactile images when they are in contact with an object. These sensors are available in various dimensions and easily integrated into different robotic structures. Contact forces can be measured with these sensors in a specific range. Nevertheless, simple tactile sensors are fragile and can be easily damaged by high pressures.

Force-Torque sensors are more suitable for robotic manipulators when interacting with objects that require the application of high forces. They are commonly made upon strain-gauge technology. Force sensors used with robotic manipulators are designed with ellipsoidal or cylindrical shapes, which enables precise estimation of the contact point and the normal to the contact surface. This was shown with ATI nano17 sensors [3], and BioTac sensors [4].

In some recent applications, sensors with hair-like structures called "Whiskers sensors" are used with robots to sense contact with the environment. They are inspired by animals' and insects' abilities to sense their surrounding environment. These sensors consist of a set of probes that are attached together to the robot, capable of detecting contacts and sending binary information about the contact state. Whiskers sensors are used with mobile robotic platforms [5], and with robotic arms [6]. The advantages of using whiskers are to explore the environment without affecting the pose of the explored objects (non-intrusive sensors) and to get as much information as the available number of probes. However, they cannot be used to manipulate objects. An example of robots equipped with whiskers is depicted in Figure 1.3

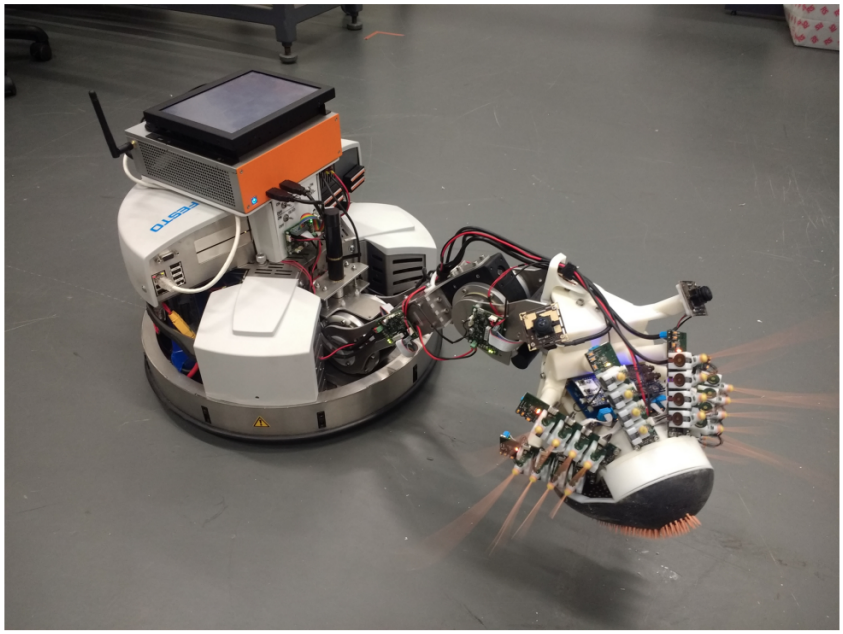


Figure 1.3: Mobile robotic plateform equipped with whiskers sensors [7].

Tactile servoing

Servoing robot manipulators to interact in a specific way with their environment requires precise movements and specific contact forces. The tactile sensing technology offers significant information for the controllers to take appropriate actions. A relevant framework for tactile servoing was developed by Zhanat KAPPASSOV, who implemented a variety of solutions for controlling the manipulation of an object, like point servoing, rolling, and contour following [8]. SOMMER et al. [9] developed a controller for multi-contact exploration of an unknown object by sliding motions using a multi-finger hand (Allegro) attached to a KUKA robotic arm. OKAMORA [10] proposed a framework for controlling the exploration by rolling and sliding motions. More recent work uses deep learning techniques to interpret the data provided by the tactile sensors and use them in a closed loop controller [11].

Tactile recognition

The ability to discriminate different objects is an important skill for robotic manipulators. Tactile sensing provides a huge amount of information for this purpose. Very significant features can be extracted from tactile data and used to manipulate objects. The first work on object identification was developed by ALLEN et al. [12] using parametric forms called superquadratics. Another parametric form to identify objects by their curvatures was presented in [13]. Sliding motions allow to identify objects by their textures, and this was proven in [14] using the SVM learning method. Some advanced tactile sensors offer highly precise information similar to images. These tactile images were used with matching techniques to recognize objects [15] and build a "bag-of-features" that serves to classify objects. For more efficient classification, a convolutional neural network was used with

tactile images in [16].

Tactile localization

Some precise handling tasks require a great deal of attention to the location of the object during the manipulation. In precise insertion tasks [17], inaccurate estimation of the object pose causes the failure of the desired operation. For these reasons, object pose estimation in robotic manipulation has become an important research interest. Technically, localizing an object using tactile data relies on finding the best geometric rigid transformation that relates the tactile data to the object model[18]. This implies a prior availability of a geometric model of the object. This problem is solved by using an optimization technique such as ICP (Iterative Closest Point) or more often using the Bayesian filtering technique. The problem of object pose estimation will be discussed in detail in Chapter 3.

Discussion

We can see from the different works reviewed that tactile sensing has been used for different purposes that in most cases were solved separately. And all the frameworks were developed under a set of assumptions. For instance, works on tactile exploration assume that the object is always fixed in the environment. And some simplifications were made regarding the shape of the object by considering only smooth curved shapes for sliding and rolling explorations. Works on tactile recognition and localization assume that the object is already in contact with the robot without considering the problem of how the robot can reach the object.
In real-world applications, such assumptions are not always guaranteed. For example, when exploring an unknown object, it may move during the exploration under the effect of the robot end-effector. If these motions are not well predicted, the reconstruction can not be accurate. Moreover, some objects need a high degree of dexterity to be explored and autonomous decision-making on the actions to be performed (active exploration). The problem of estimating the shape of an object and its pose simultaneously is very similar to a well-known problem in mobile robotics, which is called **Simultaneous Localization And Mapping (SLAM)**. It consists of simultaneously building a map of the environment and keeping track of the pose of the robot on it. The characteristics of the SLAM problem and its similarities with the problem studied in this thesis are discussed in the next section.

1.3 SIMULTANEOUS LOCALIZATION AND MAPPING (SLAM)

Mobile SLAM has been investigated for four decades. This problem was considered for the purpose of allowing the robot navigating in unknown environments to build a map of its environment in order to facilitate its localization and to plan trajectories. The solution to SLAM is challenging because of the uncertainties in the data provided to the robot. Commonly, the robot obtains data from internal sensors (such as encoders) that measure the movements of the robot (proprioceptive data) and exteroceptive sensors (such as cameras and lasers) that offer information to the robot about the environment. A good

SLAM solution is one that is able to reason efficiently with both sources of data in order to build an accurate estimation of the environment.

The SLAM problem is generally formulated as a probabilistic framework. Filtering techniques such as Kalman filters and particle filters are the most widely used techniques. In some other frameworks, the SLAM problem is solved as an optimization problem, formulated with an optimization equation that represents the distances between the measurements, and the solution is the optimal relative poses between the exteroceptive measurements. Recent solutions to SLAM combine sequential optimization techniques to improve the accuracy of the solution. Indeed, sequential filtering yields accumulated errors in the estimation. Global optimization techniques allow reducing these errors by considering all the data acquired at once. These two combinations are commonly called the front-end and back-end of the SLAM.

For almost forty years, SLAM has been a relevant field of research for mobile robots, drones, and undersea vehicles. The proposed frameworks range from 2D to 3D scene mapping. Different parts of SLAM were subjects of research: feature extraction from vision and laser data, scan matching improvements, graph optimization, and the loop closing problem. Recent SLAM frameworks integrate advanced deep learning techniques for 3D reconstruction. Active SLAM is also a recent and relevant research topic to make the SLAM process more autonomous.

In the field of robotic manipulation, SLAM would be of interest as in mobile robotics. Just as mobile robots need SLAM when navigating in unknown environments, a manipulator robot needs SLAM when manipulating unknown objects. There is a close similarity between the well-known SLAM problem in mobile robotics and object exploration in robotic manipulation. Both tasks aim to simultaneously build maps and models of the environment and objects in a progressive process.

The differences are in the perception type and the exploration procedures. For the perception modality, exteroceptive information is more challenging to obtain with tactile sensors than with vision and laser sensors. Tactile sensors have to make direct contact with the object to get the desired information. In mobile robotics, the exploration of an environment is commonly made by the teleoperation of a mobile robot rolling on wheels. In this case, the exteroceptive information collected in successive instants contains similarities that can be used by scan-matching algorithms to correct the pose of the robot in the environment. The exploration of an object with a robotic manipulator using tactile sensing is practically complicated. As for humans, blind exploration of objects can be made using different techniques: with a single hand or with two hands, by sliding motions on the surface of the object or by rolling the object between two hands. However, it is complicated to ensure a continuous and stable exploration and manipulation of an object when its geometrical and mechanical characteristics are unknown: the object's movements during this process are unpredictable. The main differences between the mobile SLAM and the tactile manipulation SLAM are summarized in Table 1.1.

	Mobile SLAM	**Tactile Manipulation SLAM**
Goal	Build a map of the environment	Build a geometric model of an object
Proprioception	Odometery	Joints encoders
Exteroception	Camera/LASER	Tactile sensors
Actions	linear displacements and rotations	Grasping / Sliding / Rolling

Table 1.1: Comparison between Mobile SLAM and Manipulation Tactile SLAM.

1.4 CHALLENGES

We consider the problem in which a robotic manipulator has to explore an unknown object. It collects contact data through its tactile sensors. This process raises many challenging questions that make the solution not easy to develop. In the following, main challenges are outlined:

- **Object stability:** The stability of the object is very critical during the exploration. Indeed, the exertion of forces on an object can lead to a considerable change in its pose. To avoid this situation, the applied forces must be adapted to guarantee the object's stability. However, the object characteristics must be known in advance to analyze the stability.
- **Manipulability:** The manipulation of objects requires a high degree of dexterity of the manipulator, and this is not always guaranteed, especially when the object to be explored is unknown.
- **Tactile data acquisition:** The robot must perform actions in order to collect contact data. These actions do not always result in positive contact due to the ignorance of the object shape and pose.
- **Sparsity of data:** Unlike vision data, tactile data is limited to a few contact points. This makes the exploration very time-consuming and tracking very difficult.
- **Continuity of exploration:** Depending on the geometric characteristics of the object, it is not always possible to explore its shape in a continuous way, like sliding fingers on its shape when the curvature is high or there are discontinuities in its shape. It must then be explored through discrete actions such as successive grasps.

1.5 PROPOSED FRAMEWORK

In this thesis, we are mainly interested in the problem of the perception of unknown objects through tactile sensing in order to build an accurate model of them. Recovering unknown objects' shapes from sparse tactile data implies finding a solution to several issues. First, we seek a solution for an adequate representation of shapes from tactile contacts. Secondly, we have to consider the problem of the uncertain poses of the object during exploration. The pose of the object must be estimated correctly to ensure an accurate reconstruction. Next, we focus on the problem of tactile data collection. To be more efficient, the robot must autonomously choose where to touch the object to improve

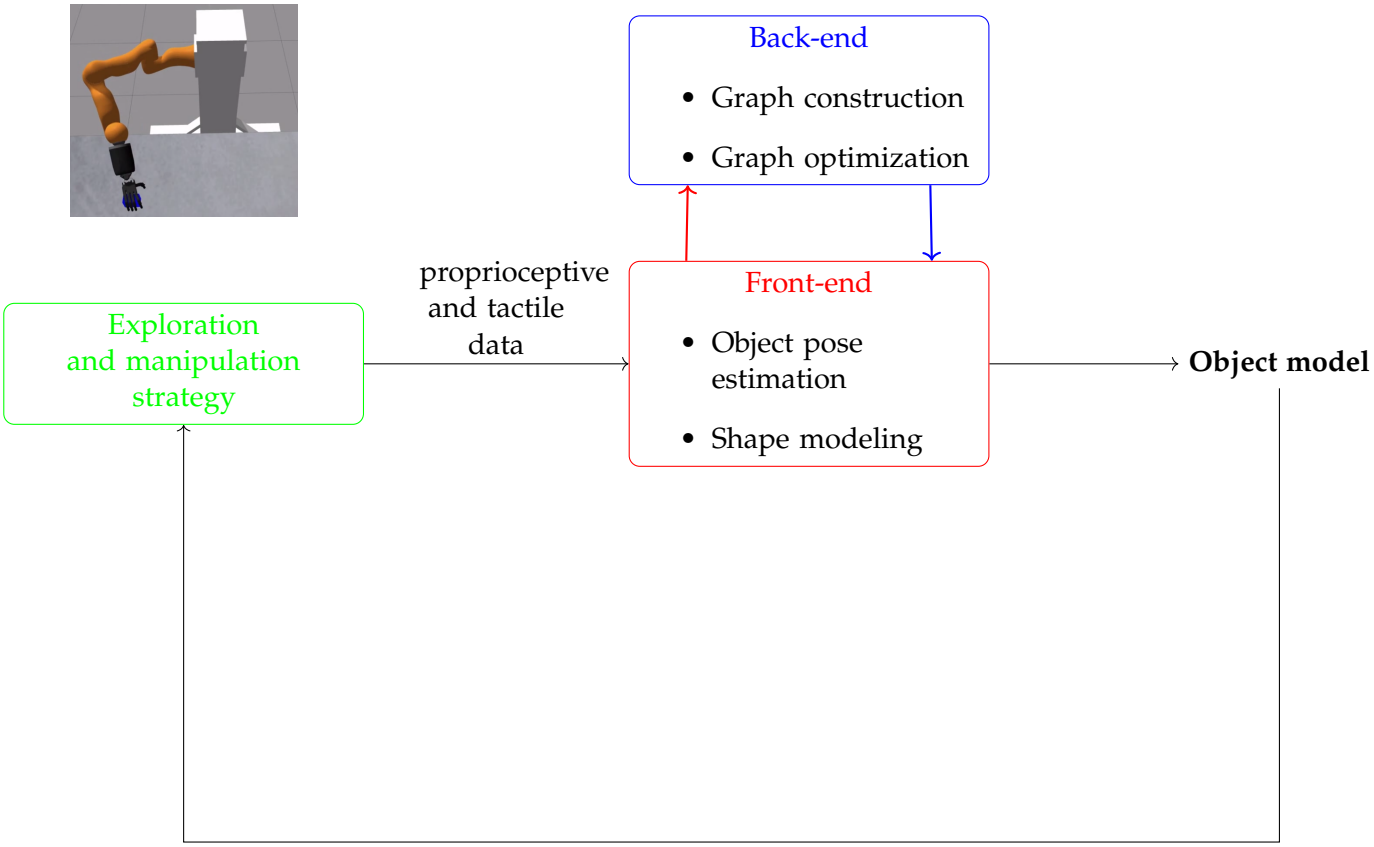


Figure 1.4: Proposed framework

its knowledge about its shape.

As specified so far, we consider our framework as the SLAM problem. This means considering the problem of mapping (reconstruction) and localization simultaneously. This is done in a sequential mode. As the manipulation and the acquisition of tactile data are performed sequentially, the reconstruction algorithm incrementally improves the representation of the object shape. Figure 1.4 illustrates the architecture of our proposed framework. The first proposed work is represented in the red block, it consists of an incremental reconstruction of the object by simultaneously focusing on the shape modeling based on tactile data and pose correction with a particle filter algorithm. Since sequential filtering for pose correction yields accumulated errors, an improvement to this method is implemented on the basis of pose graph optimization. It is shown in the blue block. It consists of a construction graph of all the tactile data collected during the exploration and the relative poses between them. The optimization of this graph allows the correction of the relative poses, thus improving the accuracy of the reconstruction. Finally, we propose an algorithm for active exploration allowing the robot to autonomously select the actions to be performed in order to improve the knowledge of the object. This work is illustrated by the green block, it consists in evaluating the uncertainty of the model and of the pose of the object and in selecting the adequate actions that are favourable to the reduction of this uncertainty.

1.6 CONTRIBUTIONS

The main contributions of this thesis compared to state of the art on tactile pose and shape estimation for robotic manipulation are:

- A method for simultaneous shape and pose estimation based on the Gaussian process implicit surface technique and the particle filter algorithm.

  Ghani KISSOUM , Véronique PERDEREAU "**Simultaneous Tactile Localization And Reconstruction of an Object During Robotic Manipulation**" accepted for publication in *20th International Conference on Advanced Robotics (ICAR)*, pages: 948–954, DOI: 10.1109/ICAR53236.2021.9659354

- An application of a pose graph optimization method to enhance the precision of the shape reconstruction.
  Ghani KISSOUM , Véronique PERDEREAU "**Pose Graph Optimization for Tactile Reconstruction And Localization of Objects**" to be submitted to *IROS 2023 : IEEE/RSJ International Conference on Intelligent Robots and Systems*,

- An active exploration strategy for shape and pose estimation based on information theory techniques.
  The last two contributions will be submitted to an international conference shortly.

1.7 THESIS OUTLINE

This thesis contains five chapters that describe the different parts of our proposed framework. These chapters are organized as follows:

- **Chapter 2:** is dedicated to the topic of modeling the shape of objects from tactile data. It reviews the most popular techniques used for modeling geometric shapes. Then, it analyses the shapes representation problem based on sparse tactile data.

- **Chapter 3:** addresses the problem of object pose estimation from tactile data. Two types of techniques are presented: recursive and global pose estimation. The proposed solution is applied to the problem of simultaneous reconstruction and localization in simulation and experimental frameworks.

- **Chapter 4:** deals with decision-making for tactile data collection. It proposes a framework for the active exploration of unknown objects for shape reconstruction and localization. The method is based on information theoretic aspects, and the framework is tested in a simulation environment.

- **Chapter 5:** discusses the limitations of the proposed framework and the potential improvements in the short and long terms.

2

OBJECTS SHAPE ESTIMATION FROM CONTACT INFORMATION

In robotic manipulation tasks, knowing the objects' geometric models is necessary for the robot to plan grasps and manipulations and to recognize or localize the object. This model is usually provided in the form of a mesh model or a point cloud of the object. In case this model is not provided in advance, it can be obtained by using different sensors such as vision, lasers, or tactile sensors.

In this thesis, we have made the choice of using the tactile modality only to perceive and reconstruct unknown objects. The limitation of this perception modality is that the amount of information that we get from the object is very sparse compared to visual information, and it requires making contact with the object on its entire surface to acquire a complete model of it. This process is generally time-consuming.

This chapter is dedicated to the techniques for modeling shapes of objects from space and noisy tactile information. We are interested in questions of the amount of tactile data necessary to build an accurate model of an object and the precision of the reconstruction that can be reached with little tactile information.

To answer these questions, we will first review state of the art on objects' shape modeling. Then, we will present an analysis of shape modeling based on simulated data. Finally, we will present the result of the implementation of our technique for modeling real objects on our robotic platform equipped with tactile sensors.

2.1 STATE OF THE ART ON SHAPE REPRESENTATION FOR ROBOTIC MANIPULATION

The representation of shapes from tactile information has been studied since the eighties with the development of advanced robotic manipulators integrating tactile sensors. There was a need to retrieve the shapes of objects to enable the robots to grasp and manipulate them. The first work in this direction that we find in the literature was proposed by [19], who used superquadratic forms to represent objects. This method is based on the identification of some parameters of superquadratic form that fit the acquired data.

Superquadratics are a family of shapes first introduced in [20], defined by a parametric equation that resembles to that of ellipsoids. A superquadratic can be described with the following equation:

$$\mathcal{P}(\eta,\omega) = \begin{pmatrix} a_1 cos^{\epsilon_1}(\eta)\ cos^{\epsilon_2}(\omega) \\ a_2 cos^{\epsilon_1}(\eta)\ sin^{\epsilon_2}(\omega) \\ a_3 sin^{\epsilon_1}(\eta) \end{pmatrix} \tag{2.1}$$

Where $\eta \in [-\pi/2, +\pi/2]$ and $\omega \in [-\pi, +\pi]$. To estimate a superquadratic from a point cloud, we need to find the size parameters a_1, a_2, and a_3 that describe the extent of the superquadratic along each axis. The parameters ϵ_1 and ϵ_2 describe the characteristics of the shape (Fig.2.1). To locate the superquadratic in space, we additionally need to introduce a rotational matrix R and a translation x_0.
Superquadratics were first used in an experimental setup to retrieve 3D shapes of objects from haptic data, with a multi-fingered dexterous hand[19]. In [21], superquadratics were used to fit a 3D point cloud obtained from haptic exploration of objects by a human operator wearing a glove equipped with sensors on its fingertips. The quality of the reconstruction depends on the distribution of the acquired points. It was improved by integrating normal information. [21] investigated the estimation of superquadratic representations from sparse and noisy contact data acquired by tactile sensors mounted on the fingertips of a robotic hand. They used a hybrid minimization method based on a genetic algorithm. This method is still used in recent works to get an approximate representation of objects for use in grasp planning problems [22][23].

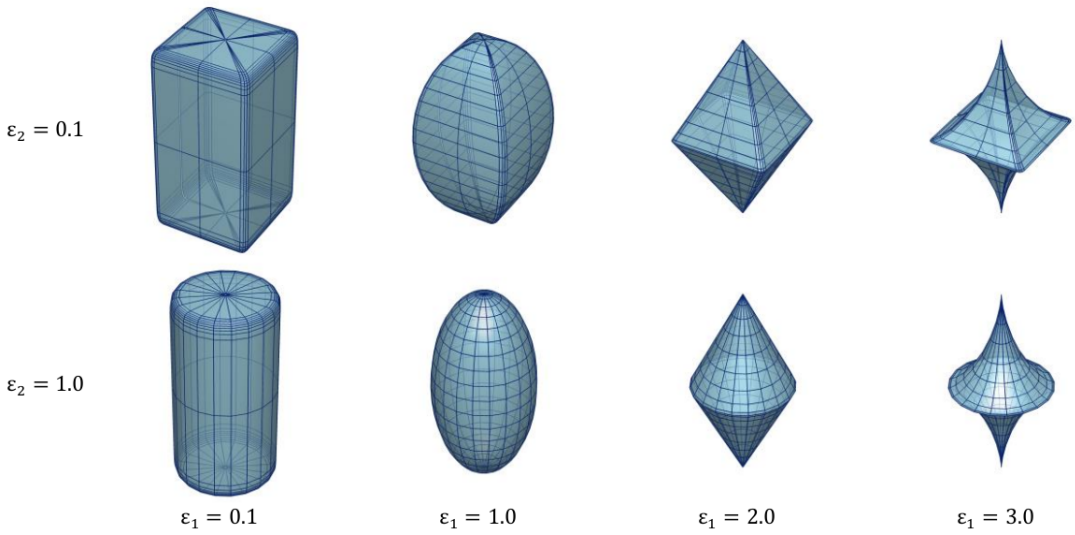


Figure 2.1: Superquadratics with different values of ϵ_1 and ϵ_2. Size parameters a_1, a_2 and a_3 are kept constant [24].

There is another parametric form that allows polynomial interpolations between the contact data to model the object surface. This is named Nagata patch [25]. It is based on defining triangular patches with the available data and then creating an interpolated surface for each patch. The interpolation is performed by the following polynomial equation:

$$p(\eta,\xi) = \mathbf{c}_{00} + \mathbf{c}_{10}\eta + c_{01}\xi + \mathbf{c}_{11}\eta\xi + \mathbf{c}_{20}\eta^2 + \mathbf{c}_{02}\xi^2 \tag{2.2}$$

with $0 < \eta < \xi < 1$ the parameters (or coordinates) of a point on the patch, and c_{ij} the coefficients of the polynomial equation calculated from the coordinates of the vertices points and corresponding normals. Nagata interpolation is illustrated in Figure 2.2

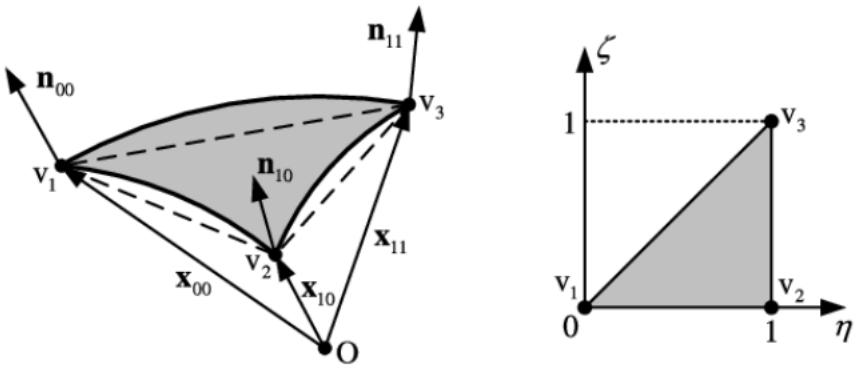


Figure 2.2: Figure adapted from [26]. Left: Illustration of the Nagata interpolation on a triangle defined by three vertices v_1, v_2 and v_3 with corresponding normals n_{00}, n_{10} and n_{11}. Right: representation of the triangular patch in its local coordinates.

In some approaches, objects are simply represented by the point cloud from the points acquired during the exploration[27][28]. In others, occupancy grid maps are used [29], which consists in dividing the space around the object into small cells with specific resolutions and defining whether each cell is occupied by the object or free.

In recent work on shape reconstruction, a nonparametric regression method based on the gaussian process was proposed. It is called the "Gaussian Process Implicit Surface" (GPIS) technique [30]. This technique allows to model shapes from some perceptual data in a probabilistic way, and different shapes can be considered with this technique.

In summary, the representation of the shape of an object can be simply achieved from the point cloud acquired by the exploration. However, this operation can be complex and time-consuming due to the number of actions that the robot has to perform to explore the entire surface of the object. Instead, an occupancy grid map can be made and define occupied and free cells on it. The problem with this representation is that: firstly, in the case of a grid map with high-resolution grid map, we have the same drawbacks as representing the object as a point cloud. Secondly, if the grid map is low resolution, the object representation can be obtained in less time but with low precision.
To obtain an accurate representation with a small amount of data, one can use superquadratic forms to approximate the shape of an object or generate an interpolated surface using polynomial forms with the Nagata patch, for instance. These techniques are efficient for recovering some classes of shapes but inaccurate with others.
For more accurate object representation, probabilistic and nonparametric methods seem the most appropriate for dealing with noisy and sparse data. For this reason, we opt for the Gaussian Process Implicit Surface (GPIS) technique to recover the objects' shapes with noisy tactile data. Another advantage of this technique is the ability to quantify the uncertainty of the reconstruction.

2.2 GAUSSIAN PROCESS IMPLICIT SURFACE

2.2.1 *Gaussian process regression theory*

We aim to describe the shape of the object as an implicit function f, which assigns to each point p in $\mathbb{R}^3$ a value that indicates whether the point is inside, outside or on the surface of the object. f is defined as follows:

$$f(p) \begin{cases} > 0, \text{ if } p \text{ is outside the object} \\ = 0, \text{ if } p \text{ is on the surface of the object} \\ < 0, \text{ if } p \text{ is inside the object} \end{cases} \tag{2.3}$$

The function f will be estimated as a Gaussian Process (GP), from already known data. In the following, we will detail the Gaussian process theory.
The function f is represented as a multivariable Gaussian:

$$P(f|\mathbf{X}) \sim \mathcal{N}(f|m, K) \tag{2.4}$$

m is the mean function, and K is a covariance matrix.
The Gaussian process is a collection of random variables, $\mathbf{X}$ is the known data,

$$\mathbf{X} \triangleq [\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N] \tag{2.5}$$

and

$$f \triangleq [f(\mathbf{x}_1), f(\mathbf{x}_2), ..., f(\mathbf{x}_N)] \tag{2.6}$$

The problem is to estimate the value of f at new points $\mathbf{X}_*$. Let's define $f_* = f(\mathbf{X}_*)$, we can express the joint distribution of f and f_* as:

$$\begin{bmatrix} f \\ f_* \end{bmatrix} = \mathcal{N}(\begin{bmatrix} m(\mathbf{X}) \\ m(\mathbf{X}_*) \end{bmatrix}, \begin{bmatrix} k & k_* \\ k_*^T & k_{**}) \end{bmatrix}) \tag{2.7}$$

with

$$m(\mathbf{X}) \triangleq [m(\mathbf{x}_1), m(\mathbf{x}_2), ..., m(\mathbf{x}_N)] \tag{2.8}$$

The covariance matrices are defined as follows:

$$k_{(ij)} = \mathbf{k}(\mathbf{x}_i, \mathbf{x}_j) \qquad k \in \mathbb{R}^{N \times N} \tag{2.9}$$

$$k_{*(i)} = \mathbf{k}(\mathbf{x}_i, \mathbf{x}_*) \qquad k^* \in \mathbb{R}^{N \times 1} \tag{2.10}$$

$$k_{**} = \mathbf{k}(\mathbf{x}_*, \mathbf{x}_*) \qquad k_{**} \in \mathbb{R} \tag{2.11}$$

$\mathbf{k}(\mathbf{x}_i, \mathbf{x}_j)$ is a kernel function that will be detailed later.

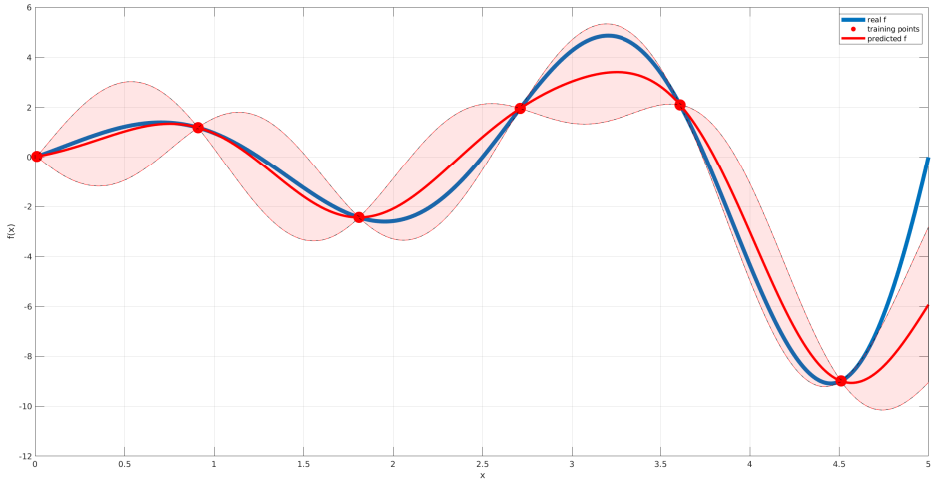


Figure 2.3: An example of the prediction of a function from a set of learned data. The real function is shown in blue. The known data (or training data) are shown as red circles. The predicted function is shown in red. The red area around the predicted function represents the standard deviation of the Gaussian process prediction.

Actually, in many applications, we always have noisy values of $f(\mathbf{x})$. Let's assume $y = f(\mathbf{x}) + \epsilon$, with ϵ a Gaussian noise with variance σ_n. Then, the covariance of the data is $K + \sigma_n I$ (I is the identity matrix), and the joint distribution becomes:

$$\begin{bmatrix} y \\ f_* \end{bmatrix} = \mathcal{N}\left(\begin{bmatrix} m(\mathbf{X}) \\ m(\mathbf{X}_*) \end{bmatrix}, \begin{bmatrix} K + \sigma_n I & k_* \\ k_*^T & k_{**} \end{bmatrix}\right) \tag{2.12}$$

We will now express the predictive expression of the Gaussian process.
The mean of the function f at a given point p_* is defined as:

$$f(p_*) = k_*(K + \sigma_n^2 I)^{-1} y \tag{2.13}$$

and the covariance as:

$$V = k_{**} - k_*^T (K + \sigma_n^2 I)^{-1} k_* \tag{2.14}$$

Figure 2.3 shows an example of the prediction of a function f according to a Gaussian process.

2.2.2 *Gaussian Process Implicit Surface for shape reconstruction*

After defining the theoretical aspects of the Gaussian process, we now detail its implementation and its uses with tactile information in the context of shape modeling.

For a clearer explanation, let us consider the reconstruction of the contour of an object, knowing some contact points. Assuming that we obtain from the tactile sensors the

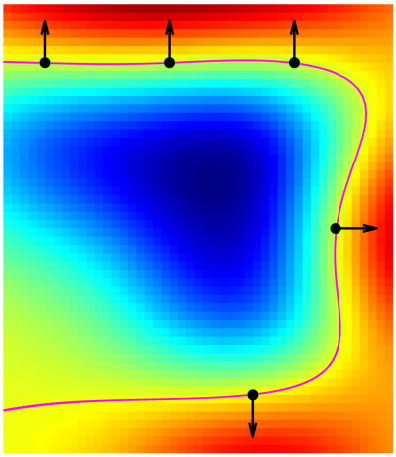


Figure 2.4: Partial reconstruction of a rectangular contour from sparse contact points.

coordinates of the contact points and the normal vectors to the surface. Our objective is to predict/estimate the object contour from the known data. As mentioned at the beginning of the section, the contour is described by the implicit function f that we will now estimate from the Gaussian Process (GP).

In the following, we explain how to design the GP, starting with the expression of the known information, then describing the GP algorithm, and finally evaluating its performance.

Starting from the integration of the training data, we first assign a 0 value $(y = 0)$ to the points on the surface (contact points) and then create other points according to the normal vectors, with $(y = -1)$ and $(y = 1)$ for the points inside and outside the surface, respectively.

As seen in the mean equation of the GP, there is always a matrix inversion, which is computationally expensive. Instead, we use the Cholesky matrix decomposition. We show below the GP algorithm:

Algorithm 2.1 Gaussian Process Regression Implementation

$L = cholesky(K + \sigma_n^2 I)$
$a = L^T \backslash (L \backslash y)$
$f_* = k_*^T a$
$Ls = L \backslash k_*$
$V = k_{**} - Ls^T Ls$

Algorithm 2.1 takes as input the contact information (contact point coordinates and normal vectors) and then generates a grid of test points in the plane around the coordinates of the points. We have chosen to create a mesh grid on the surface delimited by the

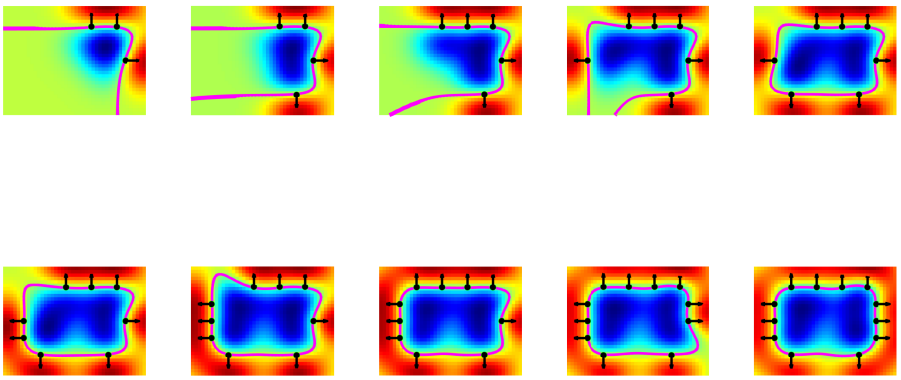


Figure 2.5: Progressive reconstruction of a rectangular contour with the square exponential (SE) kernel

contact data with a resolution of $5mm$.

We will use two different kernels in our implementation. The first one is called the "square exponential" (SE) kernel, and the second is a polynomial kernel called the "thin-plat" (TP) kernel.

The square exponential (SE) kernel is defined as follows:

$$\mathbf{k}(p_i, p_j) = \lambda \exp(-\frac{||p_i - p_j||^2}{\sigma^2}) \tag{2.15}$$

Parameters λ and σ are set to maximize the log marginal likelihood L [30]:

$$\begin{aligned} L &= \log(P(y|\lambda, \sigma)) \\ &= \frac{1}{2}\log(\det(K + \sigma_n)) + \frac{1}{2}y^T(K + \sigma_n)^{-1}y \end{aligned} \tag{2.16}$$

While the thin-plate (TP) kernel is defined as:

$$\mathbf{k}(p_i, p_j) = \frac{1}{12}(2||p_i - p_j||^3 - 3r||p_i - p_j||^2 + r^3) \tag{2.17}$$

Where r is the largest distance between two points p_i and p_j.

To illustrate the difference in performance between both kernels, we show the following example of the reconstruction of a simple rectangular shape in Figure 2.5 and Figure 2.6. As can be seen from the two figures, the reconstruction of the rectangular shape is different from the two kernels. More precisely, the curvatures of the reconstruction are different.

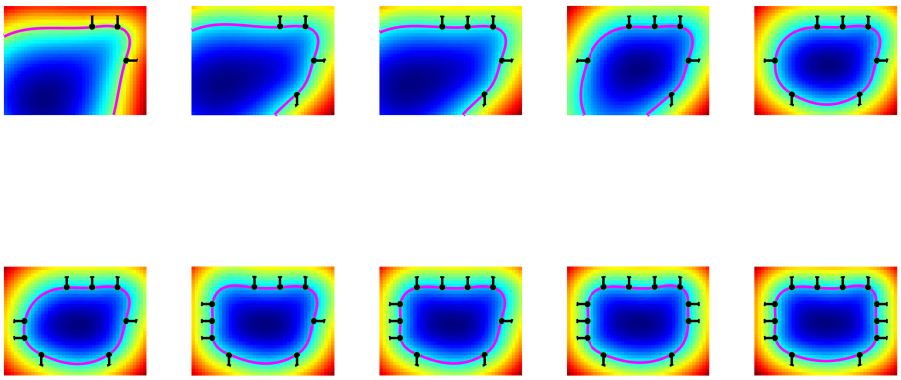


Figure 2.6: Progressive reconstruction of a rectangular contour with the (TP) kernel

For further analysis, we will evaluate the reconstruction with two different shapes. We consider a rectangular shape and an ellipsoid shape. We perform the reconstruction with both kernels and evaluate the performance of the reconstruction. The reconstruction is evaluated using the Hausdorff distance (Appendix C) between the object model and the reconstructed shape. This distance represents the maximum distance between two points of both models.

In this study, we randomly sample points from the real shape and progressively reconstruct the shape. At each new data acquisition, we recalculate the GPIS representation and compare the reconstruction with the real shape. The results are shown in Figure 2.7.

Figure 2.7 shows the reconstruction error of the two shapes, with the two kernels, depending on the number of contact points. The curves show that the reconstruction with the thin-plate kernel is relatively smoother than the reconstruction with the square exponential kernel (SE). And the reconstruction error is minimal with the thin-plate kernel (TP) compared to the square exponential kernel (SE) for a small amount of data. For a large amount of data, we see that the reconstruction has the same performance with both kernels. However, for shapes with corners, the SE kernel is more suitable for reproducing the corners, while for shapes with smooth curves, the TP kernel is more convenient. These characteristics help to choose an appropriate kernel if little knowledge about the shapes is available in advance.

2.2.3 *GPIS enhancement*

In some situations where knowledge about the objects to be modeled is available, many approaches are proposed to help improve the quality of the GPIS reconstruction.

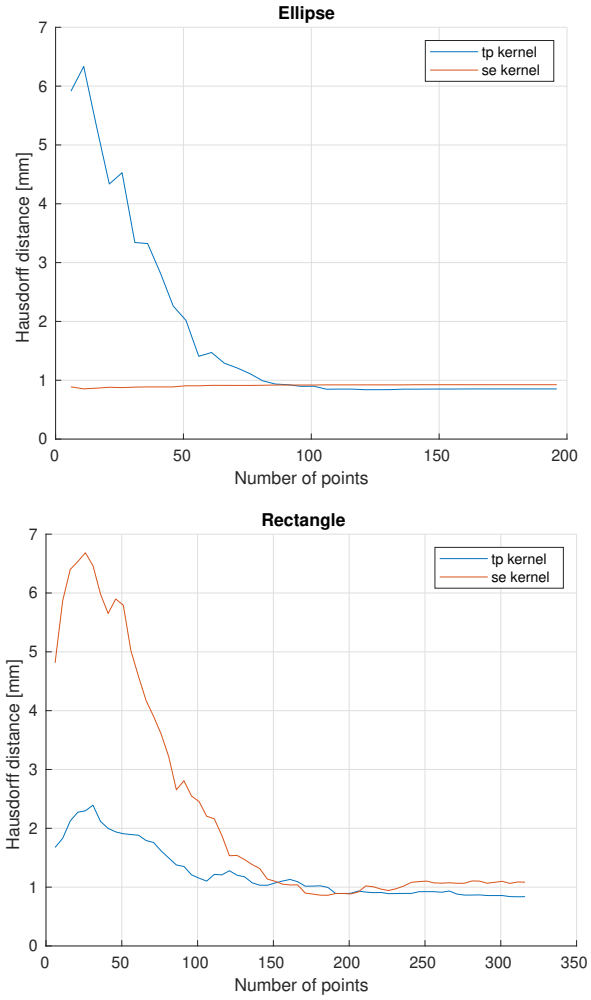


Figure 2.7: Reconstruction accuracy of two different shapes, using two different kernels. The reconstruction error is evaluated with Hausdorff distance detailed in Appendix C.

[31] proposed an improvement to the Gaussian process. The mean function of the GP was changed by introducing a non-zero prior mean. Equation (2.13) refers to a GP with zero mean. A non-zero prior mean of a GP is given by:

$$f(p) = m - k_*^T (K + \sigma_n^2 I)^{-1} (y - m) \tag{2.18}$$

The introduced prior mean value m represents in a GPIS the assumption on the point p whether it is on the surface, inside, or outside the object. The zero prior mean value means that all points are supposed to be on the object's surface.

The authors proposed to introduce primitive shapes as a prior mean value. For example, to choose a prior ellipsoidal shape for the reconstructed object, the prior mean equation is given by:

$$m_E^{\mu,\theta} = \frac{h}{2}((p-\mu)^T A_E^\theta (p-\mu) - 1) \tag{2.19}$$

where h is a scalar value, μ is the center of the object, and:

$$A_E^\theta = R_\theta^T A_E R_\theta \tag{2.20}$$

with R_θ the rotation matrix induced by the angle θ, and A_E a diagonal matrix with the ellipsoid parameters $diag(A_E) = [1/a^2, 1/b^2, 1/c^2]$

The result of introducing prior shapes in the GPIS is illustrated in Fig.2.8. This Figure shows the reconstruction of an apple (Figure 2.8(a)) from partial data obtained from one side of the apple. In Figure 2.8(b), a positive prior mean value was introduced, which means that all points are supposed to be outside the object, which explains the shape obtained with a reduced volume. In Figure 2.8(c), a zero prior mean was considered, which results in a large shape. Finally, in Figure 2.8(d), the introduction of a prior spherical shape resulted in a coherent reconstruction of the apple, which shows the benefit of introducing prior shapes.

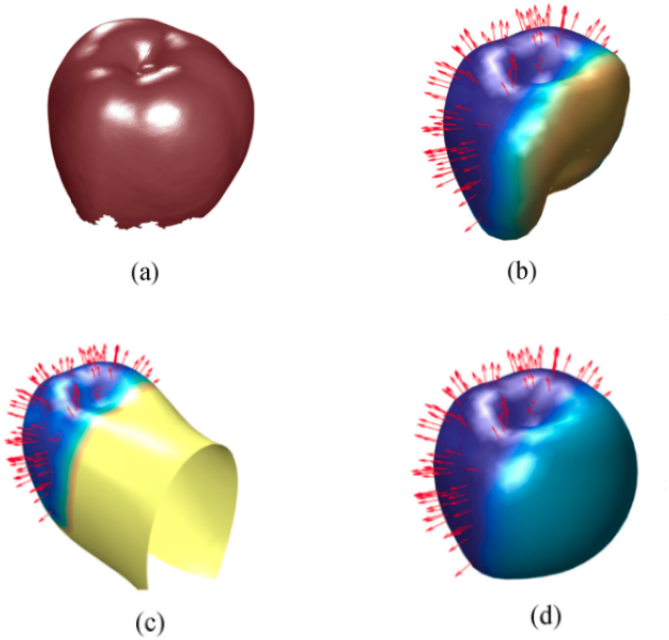


Figure 2.8: Reconstruction of the surface of an apple from partial observations, colors represent uncertainty (yellow represents high uncertainty).[31]

GPIS has limitations in modeling corners and sharp edges. To solve this problem, [32] proposed an enhancement of the GPIS by modifying the posterior mean of the GPIS to allow better reconstruction of edges and corners. The proposed modified mean function of the Gaussian Process Implicit Surface is given by:

$$f(p) = \sum_{i=1}^{N} \omega_i . y_i(p) . k(x, c_{i,p}) . \beta_i(p) \tag{2.21}$$

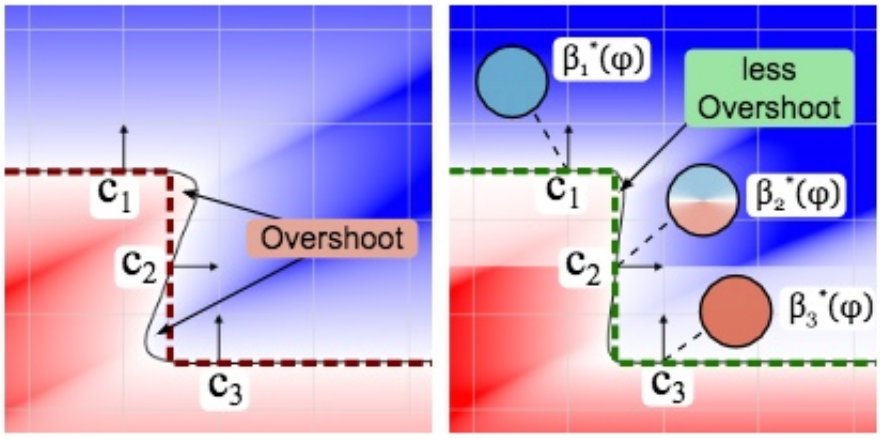
ω_i is the weight given to each contact point $c_{i,p}$, $k((p, c_{i,p}))$ is a kernel function. The contribution proposed by [32] resides in two functions y_i and β_i that describe the distance

between the tested point x and the contact c_i, and the local convexity/concavity, respectively. These two functions are given by:

$$y_i(p) = c_{i,n}.(p - c_{i,p}) \tag{2.22}$$

$$\beta_i(p) = 1 + sign(y_i(p))\beta_i^*(\phi_i(p)) \tag{2.23}$$

$c_{i,n}$ is the normal to the contact c_i, ϕ_i is an angular kernel. The results show a better local reconstruction of the corners compared to a basic Gaussian process (Fig. 2.9).



(a) ISP without local surface features: The resulting surface overshoots a the corners.

(b) Local surface features enabled: The resulting surface is more accurate with respect to the corners.

Figure 2.9: Illustration of the effect of the local surface features in reconstructing a corner from three contact points [32].

2.2.4 *GPIS in 3D*

We will use the same algorithm 2.1 for the three-dimensional reconstruction. However, we need to choose another implementation that can be more computationally efficient. Indeed, we need to generate more test points in 3D, which may be more time-consuming to test a large set of points.

For the 3D reconstruction, we first calculate the center of the obtained data, then normalize the distances of the data to the center so that the point with the maximum distance to the center is at a distance of $1cm$. To improve the reconstruction, we add some training points inside and outside the objects. To do this, we sample points from a spherical shape that encloses the object, with $y = 1$, and some points inside a sphere that lies inside the object. Figure 2.10 shows an example of recovering part of a cubic object from a few contact points.

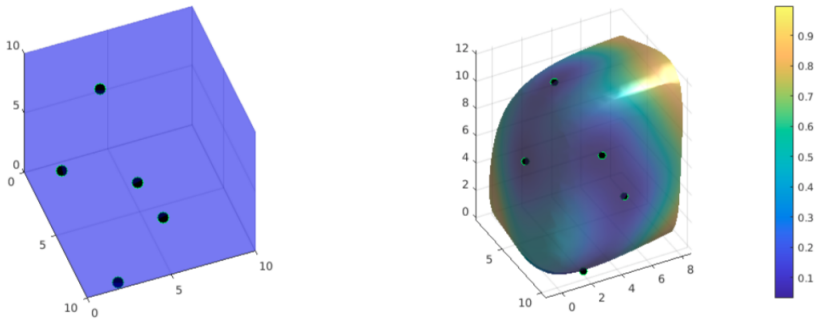


Figure 2.10: Recovering a part of a cubic shape from a small number of contact points.

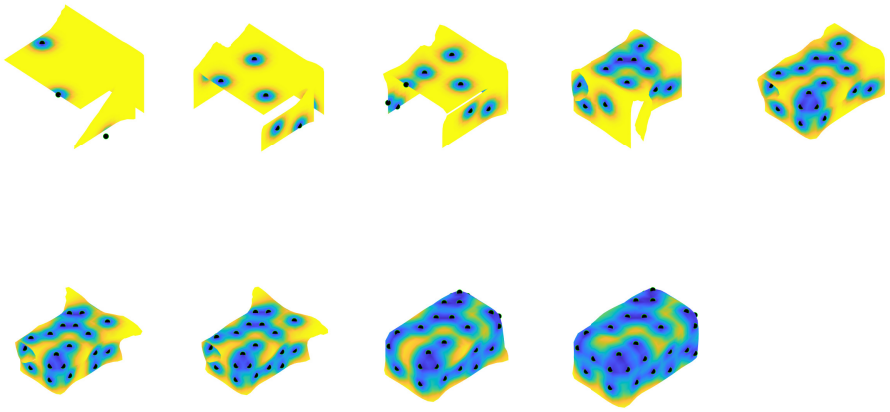


Figure 2.11: Progressive reconstruction of a rectangular object with dimensions 12cm x 6cm x 6cm.

To evaluate the precision and the amount of data needed to reconstruct a 3D object with GPIS, we performed a simulation of the progressive reconstruction of the shape of an object and calculated the error between the estimated shape and the real shape. We show in Figure 2.11 the progressive reconstruction of a rectangular object.

The reconstruction precision is evaluated at each new reconstruction by computing the Hausdorff distance between the estimated partial shape, and the real model of the object. This evaluation is shown in Figure 2.12 .

Figure 2.12 shows the Hausdorff distance between the estimated surface of the object and the real model in terms of the number of contact points. We can see from this Figure that we need approximately 80 contact points to obtain an accurate model of the object.

2.3 EXPERIMENTS WITH THE ROBOTIC HAND

This section is devoted to the experimentation of object reconstruction using the tactile data collected from sensors installed on the Shadow hand robot. We will mainly focus on

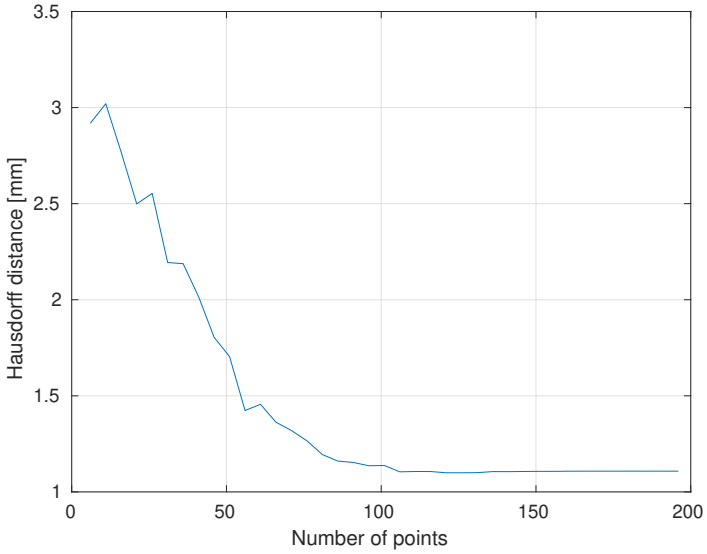


Figure 2.12: Reconstruction error in terms of the number of contact data.

partial shape reconstruction from a limited amount of contact data. We start by describing the characteristics of the tactile sensors used with the Shadow hand. Then we evaluate the results of the experiments.

2.3.1 *Sensors description*

There are two types of tactile sensors available with the Shadow hand robotic hand: ATI nano17 sensors and BioTac sensors. In fact, these sensors are force sensors. However, they can be used to measure the contact point and the normal with enough precision.

The ATI nano17 sensor is a force sensor that can measure the force vector in 3D. The position of the contact point and the normal to the surface can be inferred from the force vector, according to the ellipsoid shape of the skin covering the sensor [3].

The BioTac sensor is constituted of 19 electrodes organized on a semi-cylindrical surface (close to the shape of the human fingertip). These electrodes measure the contact pressure. The contact point and the normal vector can be deduced from the weighted average pressure over all the electrodes based on their Cartesian positions. The equation for estimating the contact location is as follows:

$$(X_c, Y_c, Z_c) = \frac{\sum_{i=1}^{19} \|e_i\| < X_i, Y_i, Z_i >}{\sum_{i=1}^{19} \|e_i\|} \tag{2.24}$$

Where (X_c, Y_c, Z_c) are the coordinates of the estimated contact point, e_i is the pressure measured on the electrode i, and (X_i, Y_i, Z_i) are the Cartesian coordinates of the electrode i. This equation is taken from [4], and more detail on the estimation can be found in this paper.

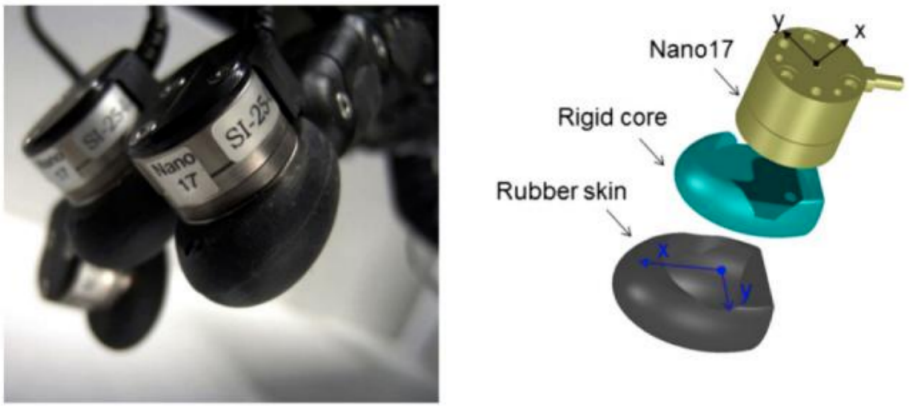


Figure 2.13: Left: ATI nano17 force sensors mounted on Shadow hand fingers. Right: Different layers composing the ATI nano17 sensor.

2.3.2 *Description of the experiments*

The goal of these experiments is to evaluate the performance of the Gaussian Process Implicit Surface modeling technique with real data obtained by grasping objects with the Shadow hand. The simulations presented above were done to analyze the performance of the GPIS technique with synthetic data. The following experiments will evaluate the shape reconstruction with data from real sensors, those being noisy and constrained by the Shadow hand kinematics.

These experiments will mainly evaluate the accuracy of reconstructing parts of an object with contact data collected during a single grasp. Reconstructing the global shape of an object by assembling different portions of its surface is a particularity of our framework. Most existing frameworks on GPIS shape modeling consider global shape modeling with a dense point cloud. For our work, it is necessary to accurately model the different parts of the object to ensure a consistent global reconstruction of its shape. The progressive reconstruction of an unknown object will be studied in Chapter 3 with the tracking of the object pose during successive grasps. However, ensuring accurate modeling of the portions of the object at every single grasp is essential to accurate global reconstruction.

In the following, we will perform some grasps of simple object shapes with the Shadow hand and evaluate the accuracy of the reconstruction of the grasped part of the object. The experiments will be carried out for the 2D (contour) reconstruction and 3D shape reconstruction.

2.3.2.1 *2D reconstruction*

The experiment uses simple shapes: a rectangle, a triangle, and an ellipse to evaluate the reconstruction. We perform a single grasp of each object and collect the contact data. The contact data (contact points and normals to the surface) are expressed in the palm frame. The GPIS reconstruction is performed with two kernels SE (Square exponential) and TP (Thin-Plate). The partial reconstructions of each shape are shown in Figure 2.14.

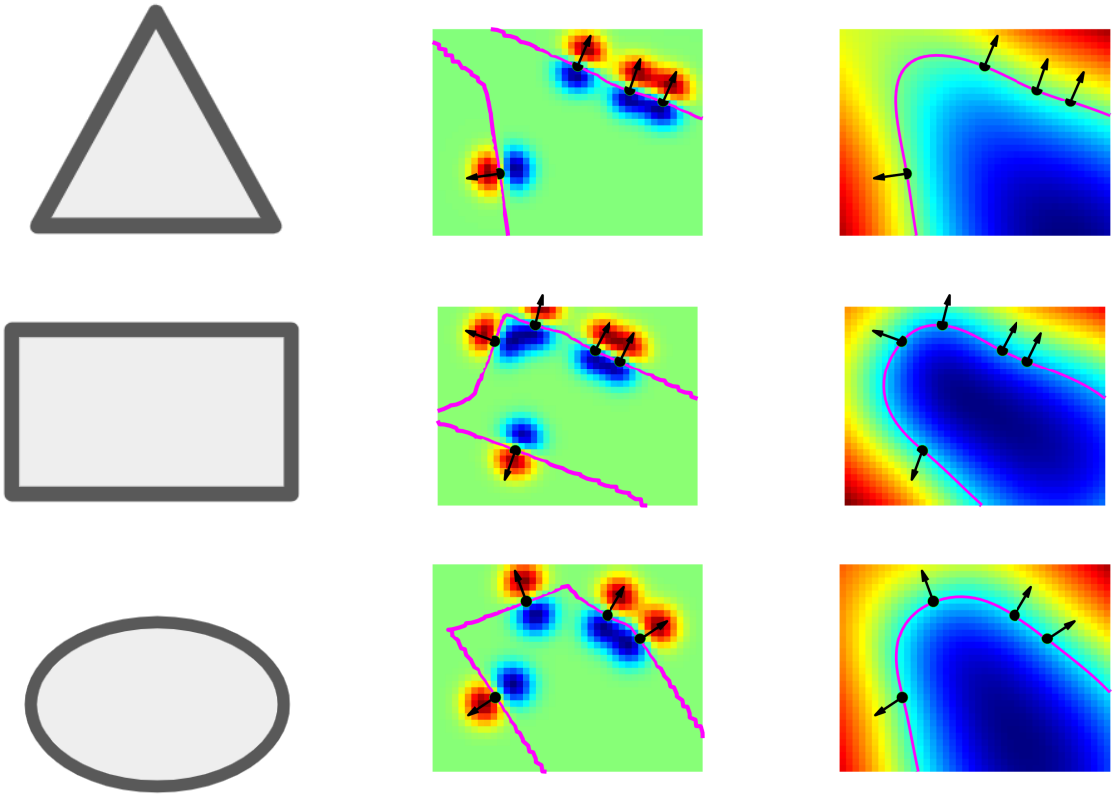


Figure 2.14: Illustration of partial recovery of a variety of shapes from a few contact points. The lines represent the different objects. The first column shows the real shape of the objects. The second column shows the partial reconstruction with the SE kernel. The third column shows the partial reconstruction with the TP kernel.

The partial representations of the shapes show different predictions with SE and TP kernels. SE kernel produces straight lines between contact points. This seems more appropriate for objects with faces. The corners of the object are better represented with the SE kernel than with the TP kernel. The reconstruction with the TP kernel produces smooth interpolations of curvatures between the contact points, which may be more appropriate for objects with curvatures, such as the ellipsoid object.

2.3.2.2 *3D reconstruction*

As for the contour reconstruction, in the 3D reconstruction, we will try to reproduce parts of the shape of the objects when they are grasped by a robotic hand. We will use two objects in this experiment: a sphere and a cube, on which we perform a single grasp with the Shadow hand and model their partial shape with the GPIS.

We can see from Figure 2.15 that the reconstruction is different with the two kernels. The SE kernel produces flat surfaces, while the TP kernel produces smooth curves between contacts. The edges are better retrieved with the SE kernel than with the TP kernel, which is more suitable for objects with curvatures.

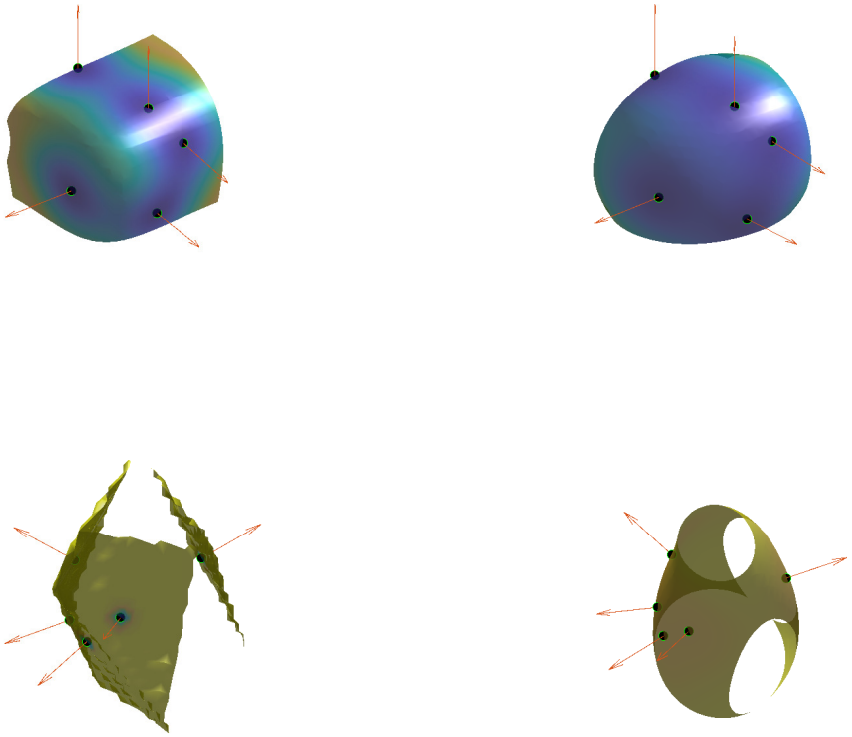


Figure 2.15: Illustration of the partial recovery in 3D of two different shapes from sparse contact points. The first row corresponds to the reconstruction of a part of a cubic object. The second row corresponds to the partial reconstruction of a spherical shape. The first and second columns refer to reconstruction with SE and TP kernels, respectively.

2.3.2.3 *Discussion*

Experiments focused on the analysis of the prediction of partial shapes using sparse contact data obtained with the Shadow hand. Ensuring the accuracy of partial predictions is essential to global shape reconstruction. We have shown through the experiments that it is difficult to guarantee accurate prediction for a variety of shape types with a small amount of data (a single grasp with the Shadow hand produces only five contacts). However, in the case of objects with faces, the SE kernel seems to offer a decent prediction accuracy. Therefore in the next chapter, we only use objects with faces for the problem of incremental shape reconstruction with pose tracking.

The GPIS technique has a limitation with objects with small faces and when two close contacts are obtained from two different faces. The GPIS prediction creates overshoots between two adjacent faces. In the example of the reconstruction of a rectangular object shape, we can observe the overshoots in the corners.

Some recent works on Gaussian Process Implicit Surface propose to use simple prior geometries with GPIS to enhance its performance (as in [33] and [34]). This can be advantageous in some applications, such as the 3D scene reconstruction from a point cloud.

However, the use of prior geometries in our work may distort the estimation of the relative pose between successive grasps.

2.4 CONCLUSION

This chapter has been devoted to the study of objects' shape reconstruction from tactile information. A comparison of different techniques for retrieving shapes from contact data shows that the GPIS technique is the most powerful for representing object shapes. The ability to recover different curvatures and to quantify the uncertainty of the reconstruction are the specific strengths of this technique.

The study presented in this chapter is essential for the following chapter since our framework is based on incremental shape reconstruction with successive grasps of unknown objects. The precision of the partial reconstruction performed with data collected at each grasp is important for the accuracy of the global reconstruction of an object. In the next chapter, we will combine the GPIS technique with pose estimation algorithms to deal with the uncertainties in the object pose during the reconstruction.

3

OBJECTS' POSE TRACKING AND CORRECTION

In this chapter, we address the problem of determining the pose of an object relative to a fixed frame by exploiting the contact information between the robotic hand and the object.

As mentioned in the introduction (Chapter 1), the object is not fixed during the reconstruction. The motions of the object during the exploration may result in an inaccurate reconstruction of the object. To avoid this problem, tracking is necessary to ensure accurate reconstruction.

The tracking of the object movement must exploit all the information provided by the robot during the exploration, starting with the prediction of the object motion from the motions of the robotic hand and the exteroceptive information gathered by the tactile sensors.

In this chapter, we will first express in detail our problem. Then we will explore the methods existing in the literature that deals with the problem of object tracking and localization in robotic manipulation. After that, we will explain our approach for tracking and correcting the pose of the object while reconstructing its shape.

3.1 PROBLEM DEFINITION

During the exploration of an unknown object to reconstruct its shape, the robotic hand performs successive grasps and manipulations of the object on a horizontal plane. This process is illustrated in Figure 3.1. When grasping, the hand is placed above the object, and the fingers are progressively closing until contact is made with the object. In this situation, when a finger touches the object, it may move it a little bit on the horizontal surface. After grasping, the hand performs a rotation of the object. During this rotation, some fingers may slip on the surface of the object, which means that the object is not moved as desired. These movements of the object must be taken into account during the reconstruction of the object. To do so, we have to estimate them precisely, exploiting the tactile data.

In order to retrieve the shape accurately, the data must be assembled correctly. This means that the contact points must be added correctly to the partial model of the object. We need to establish a good correspondence between the newly acquired data, and the partial model of the object. This is equivalent to finding the relative pose of the object with respect to the newly acquired data. To develop a solution to the problem of object pose tracking, we have to find answers to the following questions:

- How to predict the movements of the object during the exploration?

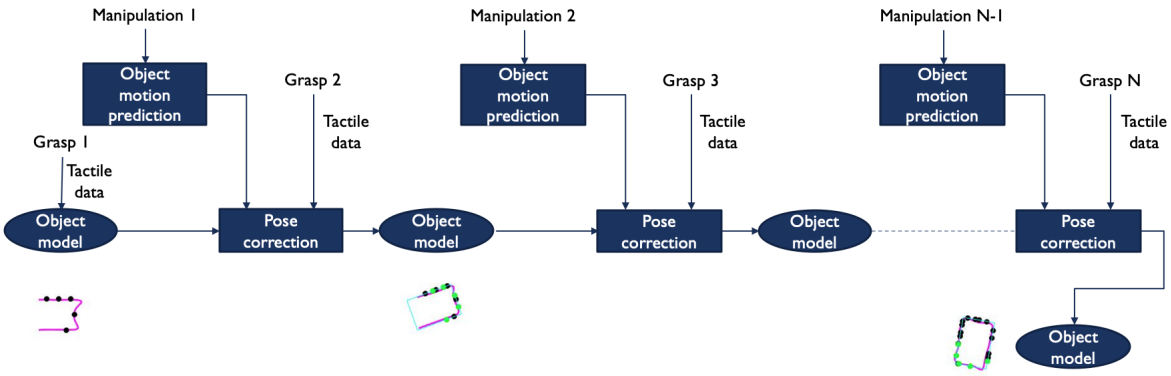


Figure 3.1: Overview of the simultaneous localization and reconstruction process. The bottom of the figure shows a progressive reconstruction of a rectangular object. The contact points are represented by black dots, and the model of the object is represented in magenta.

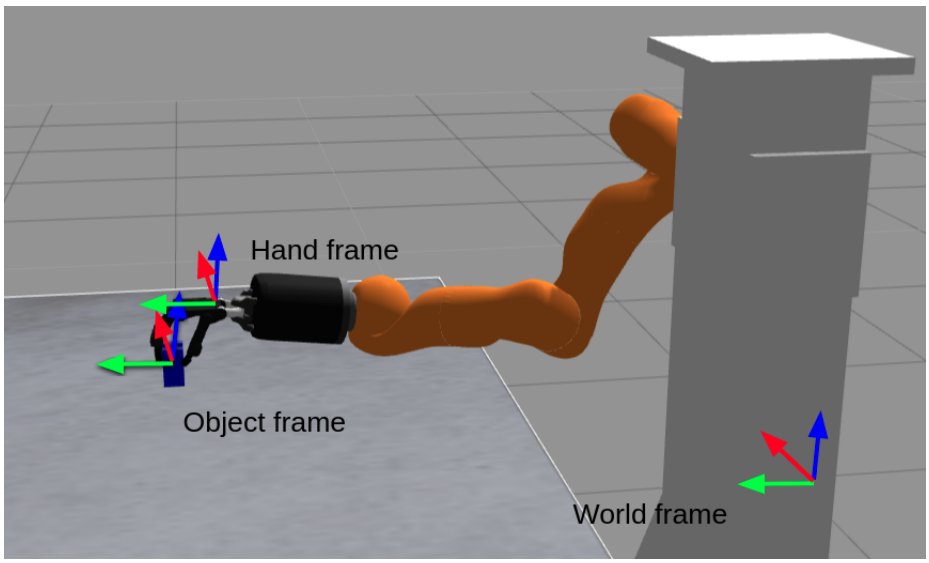


Figure 3.2: Representation of the different frames in which the problem is described.

- How to map the model of the object to new tactile data ?
- How to correct or smooth the errors of estimations?

In our framework, there are mainly three frames in which motions and poses are expressed (Figure 3.2):

- The world frame, related to the robot base R_0
- The robotic hand frame, fixed on the palm of the hand R_h
- The object frame R_{obj}

The movements of the object and the hand will be expressed in the world frame R_0, and the localization and reconstruction of the object will be made in this frame R_0.

3.2 STATE OF THE ART

Object localization from tactile and visual data was first proposed by [35] using a matching technique between the tactile data and the object model. Since then, many works have expressed the localization problem as an optimization problem, where a cost function is defined as the distance between the data and the object model. Optimization techniques are an intuitive way to find the rigid transformation between the data and the model, which represents the object pose. However, this method suffers from several limitations: the problem of local minima, since the problem is not convex, and the uncertainty of the model and the data that can not be modeled.

Secondly, Bayesian filtering techniques have been largely used in the problem of pose estimation. [36] presented an Extended Kalman Filter (EKF) method to localize an object grasped by a robotic hand. However, the use of the Kalman filter and its variants is very limited due to the nature of the localization problem, which is nonlinear and multimodal. Thus, particle filtering techniques were more favoured for this problem. In this context, a large number of frameworks have been proposed in the last two decades. First, there are some works that propose a planar localization of objects with 3 D.O.F, using the particle filter techniques [37]. To extend the method to 6 D.O.F localization, the particle filter needs more particles to efficiently estimate the pose, and the number of particles needed grows exponentially, which will lead to an increase in the execution time of the algorithm. To overcome this limitation,[38] proposed the use of a scaling series particle filter to localize objects in 6 D.O.F, and [39] proposed the use of an unscented particle filter for the same problem. Table 3.1 summarizes these works.

Sequential filtering applied in the context of simultaneous localization and reconstruction problems produces accumulated errors. The SLAM community has been interested in global methods that take into account all measurements in the localization and mapping, the so-called "global pose graph optimization". It is based on the definition of a quadratic equation representing the distance between the model of the object, and the measured data defined in a certain pose. The objective is to find the transformation that minimizes the distance between the object model and the acquired data, which represents the pose of the object.

Many libraries propose to optimize the factor graph equation. The best-known libraries are: iSAM (Incrimental Smoothing And Mapping)[46], iSAM2[47], root(SAM)[48], g2o (Global Graph Otimization)[49], SPA (Sparse pose adjustment) [50]. These methods differ in the way that the optimization equation is processed. For our framework, the g2o library seems the most appropriate. It is more flexible for different types of sensor data and works for both 2D and 3D cases. Moreover, this library is already available in different programming languages.

Recent advances in tactile sensors technology have made it possible to design sensors that produce tactile images with high precision, such as "gelSight" sensors. This allows learning a neural network that maps the sensor observation to the pose of the object, as in [51] (Figure 3.3), and [52] which proposes to use Bayesian filtering with a learned obser-

Paper	Problem	Method	Number of D.O.F	Model of the object
[40]	Tracking an object in motion	Optimization	6	Point-cloud
[37]	Tracking an object in motion	Particle Filter	3	—
[41]	Global localization of a fixed object	RANSAC + Particle Filter	3	Mesh
[42]	Global localization of a fixed object	Particle Filter	6	—
[43] [38]	Global localization of a fixed object	Scaling Series Particle Filter	6	Mesh
[44]	Global localization of a fixed object	Unscented Particle Filter	6	Mesh
[36]	Global localization of a fixed object	EKF	6	Mesh
[45]	Global localization of a fixed object	Particle Filter	6	Mesh
[39]	Tracking an object in motion	Manifold Particle Filter	3	Points

Table 3.1: State of the art of existing methods for object pose estimation from tactile data

vation model. Learning approaches have shown high accuracy in tactile pose estimation. However, they require a large data set, and cannot be generalized to unknown objects.

We point out that there are two distinct problems in object pose estimation: object tracking while the object is moving and localization of a fixed object with sequentially acquired data. In our case, we need to track the object while it is being handled by the robotic hand. What makes the work more challenging is that we have no *a priori* model of the object, and the amount of data is very small.

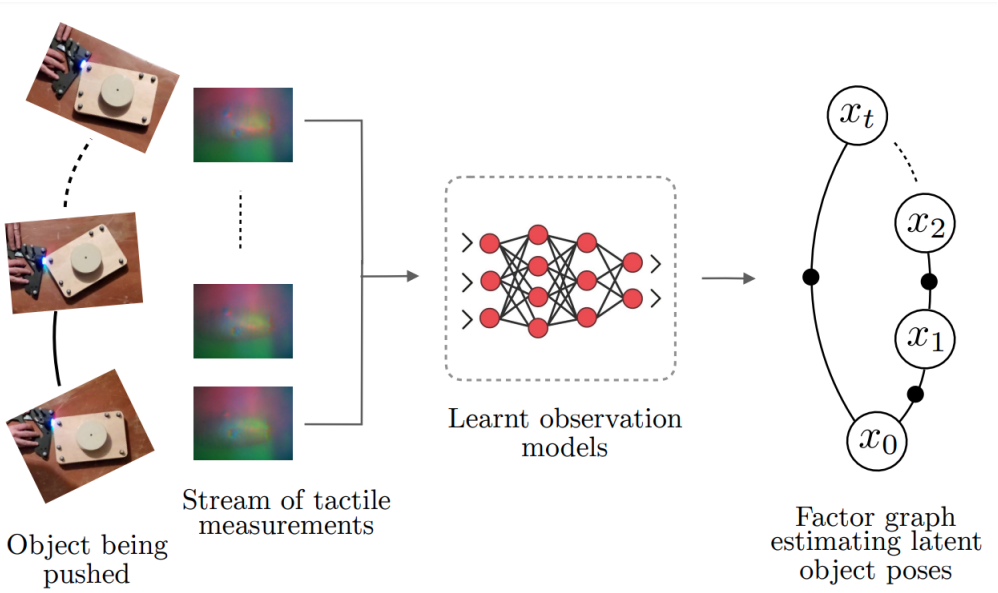


Figure 3.3: Framework for learning a measurement model for object pose estimation. Figure taken from [51]

3.3 OUR APPROACH

To estimate the pose of the object during tactile exploration, we decided to apply the particle filter approach. This choice is motivated by the fact that it is easier to design and more efficient for sparse data compared to optimization and learning approaches.
The optimization technique requires a convex problem to be able to find the global minimum. But the equation related to this problem is not always convex. Moreover, the amount of data is too small to get an efficient optimum (observability problem).
Learning techniques require a large amount of data to learn the mapping between the sensors' observations and the actual pose. In addition, this can be only done in the case where the object to localize is *a priori* known.

Our approach consists in applying a particle filter in order to track the pose of the object during exploration. This technique works sequentially: the pose of the object is updated when new tactile information is acquired, and the object model is updated accordingly (Figure 3.4). We will see later in this chapter that this sequential filtering yields accumulated errors in the pose of the object, which affects the reconstruction. To reduce this error, we developed a pose graph at the back end of the process, which consists in looking for relative poses between contact data and correcting them in a global manner. Figure 3.5 illustrates the architecture of this approach, and we explain hereafter the different parts of this approach.

3.3.1 *Particle filter*

Particle filters are a Bayesian filtering method that allows the sequential estimation of the posterior distribution of the state x_k at time k of a system, given the sensor observations

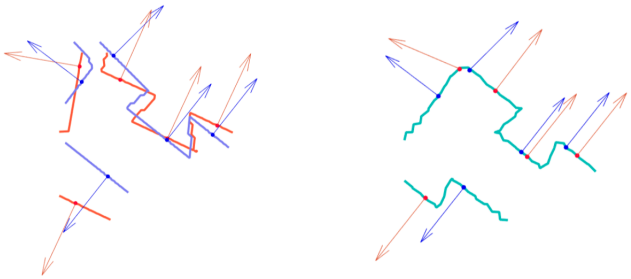


Figure 3.4: Scan matching process. Left: Two successive tactile measurements and their GPIS representation before alignment. Right: Updated shape representation after alignment of the measurements.

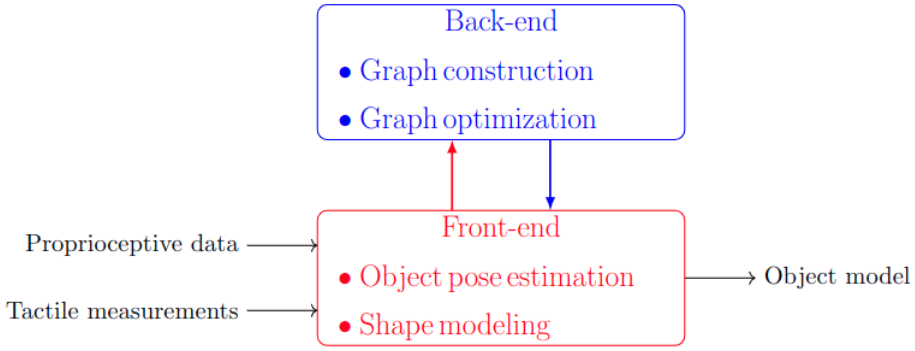


Figure 3.5: Pose tracking and correction architecture with front-end and back-end processing

$z_{1:k}$. According to the Bayes recursive inference, the state of the system is given by:

$$P(x_k|z_{1:k}) = \frac{P(z_k|x_k)P(x_k|z_{1:k-1})}{P(z_k|z_{1:k-1})} \tag{3.1}$$

$P(z_k|x_k)$ is the measurement likelihood, and $P(x_k|y_{1:k-1})$ is the prediction density.

Analytically, these probabilities are defined as follows:

$$P(x_k|z_{1:k-1}) = \int P(x_k|x_{k-1})P(x_{k-1}|z_{1:k-1})dx_{k-1} \tag{3.2}$$

and

$$P(z_k|z_{1:k-1}) = \int P(z_k|x_k)P(x_k|z_{1:k-1})dx_k \tag{3.3}$$

These probabilities can not be computed analytically unless strong assumptions are made. This is where the particle filter becomes interesting. The particle filter estimates the probability $P(x_k|z_k)$ by evaluating a set of samples (particles) $x_k^{(i)}$.

$$P(x_k|z_k) = \sum_{i=1}^{N_s} w_i \delta(x_k - x_k^{(i)}) \tag{3.4}$$

with $\delta(\dot{})$ is the Dirac function, and w_i the weight of each sample, and $\sum_{i=1}^{N_s} w_i = 1$

The particle filter is a numerical algorithm that is used to estimate the state by performing the Monte Carlo algorithm. It consists in generating a set of particles representing a hypothesis about the pose of the object from a defined probability distribution and then evaluating each particle using a measurement model. In our work, we are interested in estimating simultaneously the model f and the pose x of the object at each instant k, which can be expressed by the following expression:

$$P(f, x|z, u) = P(f|x, z) * P(x|u, z) \tag{3.5}$$

The index k is omitted in this equation for simplification.
$P(f|x, z)$ is estimated analytically with the Gaussian Process Implicit Surface technique described in the previous chapter.
$P(x|u, z)$ can be written as:

$$P(x_k|z_k, u_k) = \eta P(x_k|x_{k-1}, u_k) * P(z_k|x_k) \tag{3.6}$$

η is a normalization factor.

This equation comprises two parts: the motion model $P(x_k|x_{k-1}, u_k)$, and the observation model $P(z_k|x_k)$.

In our case, the motion is defined as follows:

$$x_k = F(x_{k-1}, u_k) + v_k \tag{3.7}$$

with v_k is a Gaussian noise that represents uncertain displacements of the object, and the function F computes the geometric transformation of the object pose when applying the action of the hand u_k.

$$F(x_{k-1}, u_k) = x_{k-1} \oplus u_k \tag{3.8}$$

$\oplus$ is the composition operator between two object poses. x_k is the prediction of the current object pose and x_{k-1} is the previous estimated pose. u_k represents the translation and the rotation of the hand when the object is grasped.

The second part of Equation 3.6 represents the measurement model, it measures the degree of matching between the newly acquired data and the model of the object in a given pose.

We define the measurement model by the following equation:

$$P(z_k|x_k) \propto \exp(-\sum_{i}^{N_c} |f(c_i)|) \tag{3.9}$$

N_c is the number of contact points acquired during a single grasp. This measurement model assigns a high score probability when the contacts c_i are on the surface of the object defined by the function f and a low probability otherwise.

The object pose and shape estimation are performed repeatedly in the following steps:

– Sampling: Generates N particles from the distribution $P(x_k|x_{k-1}, u_k)$.

– Weighting: Evaluate each particle using the measurement model, and assign a weight $w^{[i]}$ to each particle.

– Pose correction and shape update: Estimate the relative pose between the existing partial model and the newly acquired data.

– Resampling: Particles with low weight are removed and replaced with high weights particles.

These steps are organized in Algorithm 3.1 .

Algorithm 3.1 Simultaneous tactile reconstruction and localization algorithm

Input: u_k, x_{k-1}, z_k, f
Output: x_k, f
for i=1 to N **do**
  Sample $x_k^{[i]}$ from the motion model
  $w\_k^{[i]} = P(z\_k|x\_k^{[i]})$ ▷ Weight the particles $x\_k^{[i]}$
end for
%Normalize the weights
$W = \sum_{i=1}^{N} w_k^{[i]}$
for i=1 to N **do**
  $w_k^{[i]} = \frac{w_k^{[i]}}{W}$
end for
% Estimate the pose of the object
$x_k = \sum_{i=1}^{N} w_k^{[i]} x_k^{[i]}$
%Resampling
if $N_{\text{eff}} < N/2$ **then** Resample the particles
end if
Update the function f including the measurements z_k

3.3.2 *Measurements observability*

Checking the observability of the estimated error is important to avoid errors in the estimation. Observability checking means ensuring that the estimated relative pose exists and is unique. To illustrate this graphically, Figure 3.6 shows a simple example of a situation where a robotic hand grasps an object of rectangular shape with four fingers. We can see in this example that by translating the object along the X-axis, we will obtain a good match between the object model and the tactile contacts. Which means that we do not have observability around the X-axis. This can be verified analytically according to [53], considering the cost function that evaluates the matching between à set of points p_i' and p_i, with the corresponding normals n_i :

$$F_{pn}(R,T) = \sum_{i=1}^{N} [(Rp_i' + T - p_i) \cdot n_i]^2 \tag{3.10}$$

where R is a rotation matrix, and T a translation vector.

The linearization of this equation yields:

$$F_{pn}(\Delta R, \Delta T) = \sum_{i=1}^{N}[(\Delta R a_i + \Delta T - b_i) \cdot n_i]^2 \tag{3.11}$$

Considering small variations of R and T with:

$$\Delta R = I + S(x_R) \tag{3.12}$$

and

$$\Delta T = x_T \tag{3.13}$$

$S(\dot{})$ is the skew-symmetric matrix.

The equation 3.11 becomes:

$$f(x) = \sum_{i=1}^{N}[(x_r \times p'_i + p'_i + x_T - p_i) \cdot n_i]^2 \tag{3.14}$$

This can be written as:

$$f(x) = \sum_{i=1}^{N}\|\mathbf{y}_i - \mathbf{H}_i x\|^2 \tag{3.15}$$

with

$$\mathbf{y}_i = n_i^T \cdot (p'_i - p_i) \tag{3.16}$$

and

$$\mathbf{H}_i = [-(p_i \times n_i)^T, -n_i^T] \tag{3.17}$$

To check the observability, we need to calculate the eigenvalues of the following matrix:

$$A = \sum_{i=1}^{N}(H_i)^T H_i \tag{3.18}$$

The eigenvalues inform us about the observability along an axis. For example, in a 2D case, we have $\lambda = [\lambda_1, \lambda_2, \lambda_3]$, if $\lambda_1 = 0$. This means that we have no observability of the rotation of the object. λ_2 and λ_3 inform us about the observability along the X-axis and Y-axis, respectively.

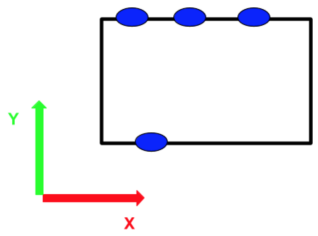


Figure 3.6: An example of a lack of observability around the X-axis. The figure shows four contact points (in blue) on a rectangular object. In this case, different values of X may correspond to the pose of the object.

3.3.3 *Graph Optimization*

Graph optimization allows to take into account all the data acquired during the exploration and to perform a global optimization of the state of the system. The graph optimization technique is based on the definition of a graph, which is constructed from nodes and edges. Nodes represent the states of the system and the acquired data at these poses, and edges represent the relative transformation between the states. Figure 3.7 show the principle of the graph.

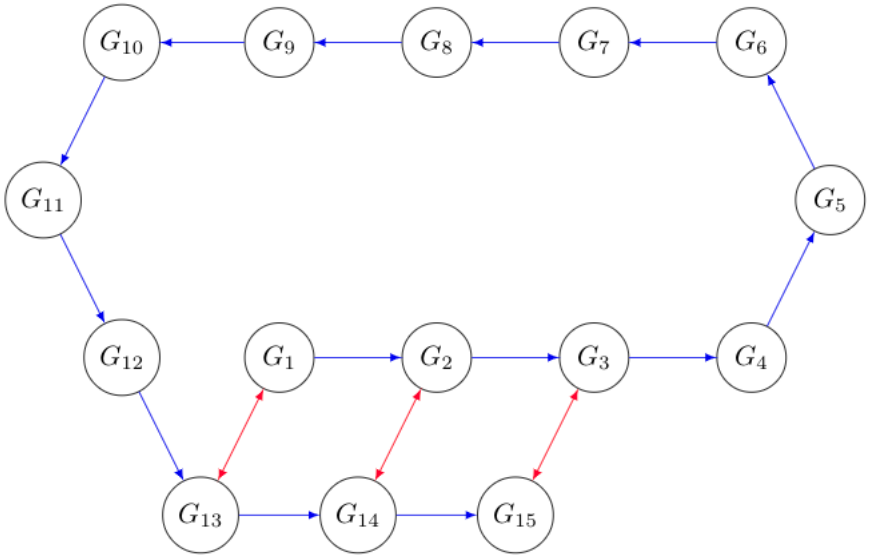


Figure 3.7: Pose graph representation. Each node is represented by a grasp G_i, which incorporates the contact information collected during the grasp. The links between the nodes represent the relative poses estimated between grasps.

The optimization problem is defined by the following equation.

$$x^* = \underset{x}{\operatorname{argmin}}\, E(x) \tag{3.19}$$

Where x^* is the vector of the optimized poses (of the object). $E(x)$ is given by the following equation:

$$E(x) = \sum_{ij} e_{ij} \Omega_{ij} e'_{ij} \tag{3.20}$$

$$e_{ij} = T_i^{-1} T_j \ominus T \tag{3.21}$$

Ω_{ij} is the inverse of the covariance matrix of relative measurements.

$$\Omega_{ij} = \Sigma_{x_i}^{-1} \tag{3.22}$$

The covariance of the relative measurements represents the uncertainty of the measurements and is calculated as described in [54]:

$$\begin{aligned} K &= \sum_j x_i^{[j]} x_i^{[j]T} P(x_i^{[j]} | x_{i-1}, u, f, z) \\ u &= \sum_j x_i^{[j]} P(x_i^{[j]} | x_{i-1}, u, f, z) \\ s &= \sum_j P(x_i^{[j]} | x_{i-1}, u, f, z) \\ \Sigma_{x_i} &= \frac{1}{s} K - \frac{1}{s^2} u u^T \end{aligned} \tag{3.23}$$

The minimization of the equation $E(x)$ (Equation 3.19) can be made using the Gauss-Newton algorithm. This needs first to approximate the error function e with its first-order Taylor expansion:

$$e_{ij}(x + \Delta x) \approx e_{ij}(x) + J \Delta x \tag{3.24}$$

with

$$J = \frac{\partial e_{ij}(x)}{\partial x} \tag{3.25}$$

3.3.3.1 *Graph construction*

The construction of the graph is based on the "similarities" between the acquired tactile contacts. "Similarities" means the contact information acquired from the same parts of the object. In order to detect these similarities, we need to compare some geometrical features obtained from the contact information. In our case, we assume that the object is composed of faces. Hence, it is easy to detect similarities as features.

To do so, in 2D reconstruction, we simplify the reconstructed shape into segments using the recursive Douglas-Peucker algorithm [55], and assign each contact to its segment. Then, we identify similar grasps, i.e. grasps whose contacts are obtained from the same faces of the object.

For the 3D reconstruction, we segment the shape into different faces in order to find similar measurements as shown in Figure 3.8. The segmentation is performed by a clustering algorithm detailed in Appendix A. It operates in two steps: the first one consists in clustering all contacts according to their normal direction. The second one is to create sub-clusters according to the distance between contacts. Then similar grasps are identified.

In order to identify similar grasps, we need to find grasps that are obtained from at least three faces, ensuring observability as described in the previous section. Next, we calculate the relative pose between the two similar grasps. The relative pose is computed by minimizing the following cost function inspired from [56] and [57]:

$$T^* = \underset{T}{\operatorname{argmin}} \sum_i (T(p_i) - p_i') \cdot R(n_i) + (T(p_i) - p_i') \cdot n_i' \tag{3.26}$$

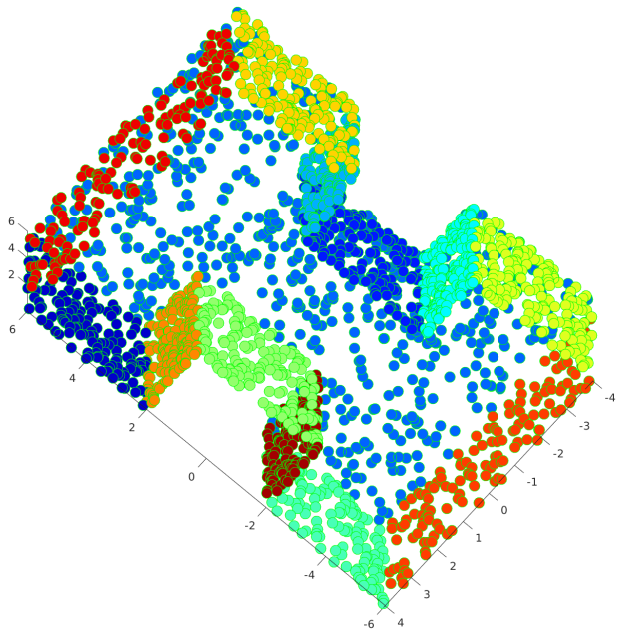


Figure 3.8: Segmentation of a 3D shape. Each face of the surface is represented with a specific color.

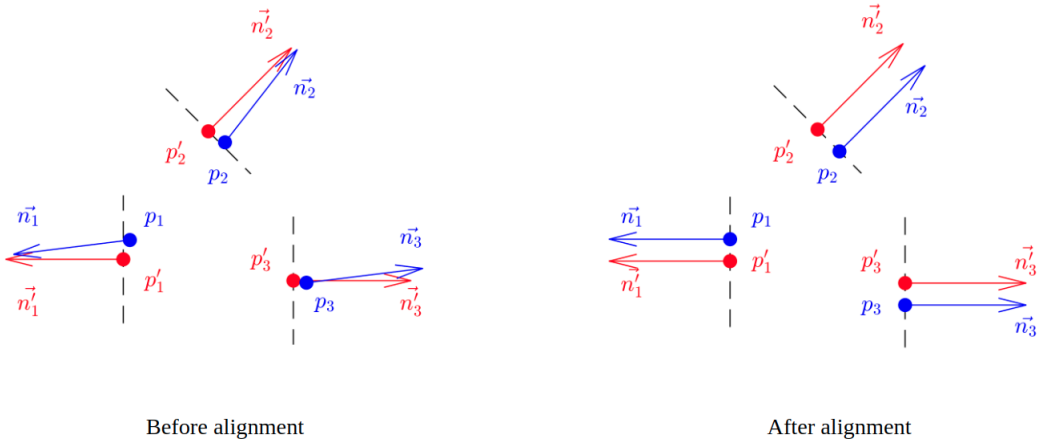


Figure 3.9: An illustration of contact points and normals alignment with corresponding faces.

With p_i and p'_i the associated points, and n_i and n'_i the associated normals. $T = [R|t]$ is the relative transformation between the two measurements. R is a rotation matrix, and t is a translation vector. An example of pose correction is shown in Figure 3.9.

3.4 SIMULATIONS

Simulations were performed on the Gazebo-ROS simulator. We used the Shadow robotic hand equipped with BioTac sensors on the fingertips and attached to the 7-D.O.F KUKA arm. Figure 3.10 shows the simulation environment.

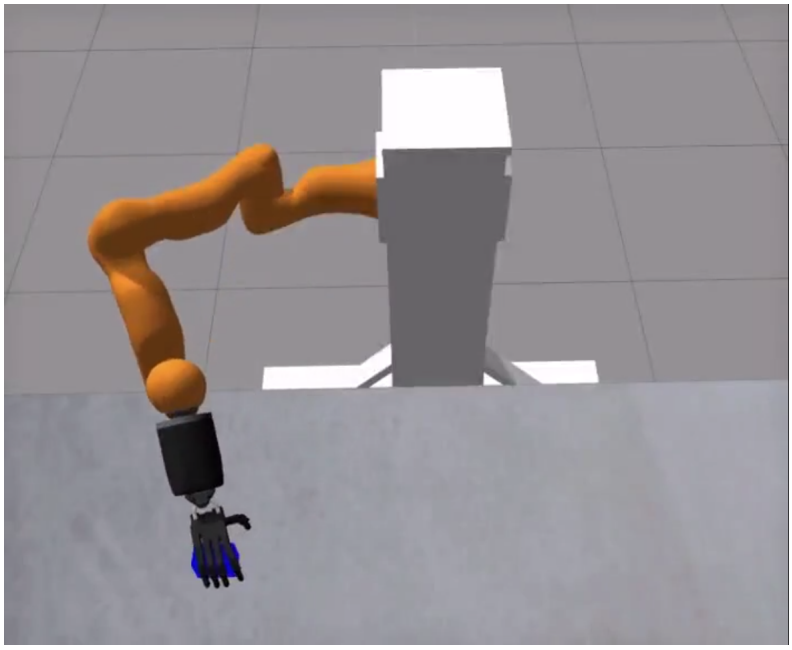


Figure 3.10: Gazebo simulation environment: the Kuka LWR arm and the Shadow hand is equipped with BioTac sensors.

The object to be reconstructed is at rest on a flat surface. The world frame is attached to the robot base frame. At the first grasp, an object frame is created. The origin of this frame is the center of the contact points, and its X-Y axes are aligned with the axes of the palm of the robotic hand frame. At each new grasp, the measured contact information is obtained in the robot frame and transformed into the object frame. The coordinates of the object frame are first predicted using the motion model after each manipulation and then updated after the acquisition of new measurements.

The noise on tactile measurements is assumed to be Gaussian with a standard deviation of $\sigma_{pos} = 1mm$ on the contact point location and $\sigma_{rot} = 5deg$ on the contact normal. We recall that a standard deviation of $1mm$ means that 99,7% of the noise values fall into a range of $\pm 3mm$.

Our algorithm was implemented in MATLAB, on an Intel-7 desktop PC. We used 20 000 particles. The time necessary to weigh such a number of particles is 6 seconds on average.

The qualitative results of the shape reconstruction are shown in Figure 3.11 with three objects. The object shape is modeled as a GPIS function. When enough data is available (dense point cloud, for instance), GPIS can provide a precise model of the object. However, the small amount of data available at each grasp produces only a small improvement in the object model. Thus, several grasps are needed to reconstruct the complete shape of the object.

3.4.1 *2D simulations*

In this part, we consider the problem of simultaneous reconstruction and tracking of the contour of an object, applying the approach described above. We perform 16 successive

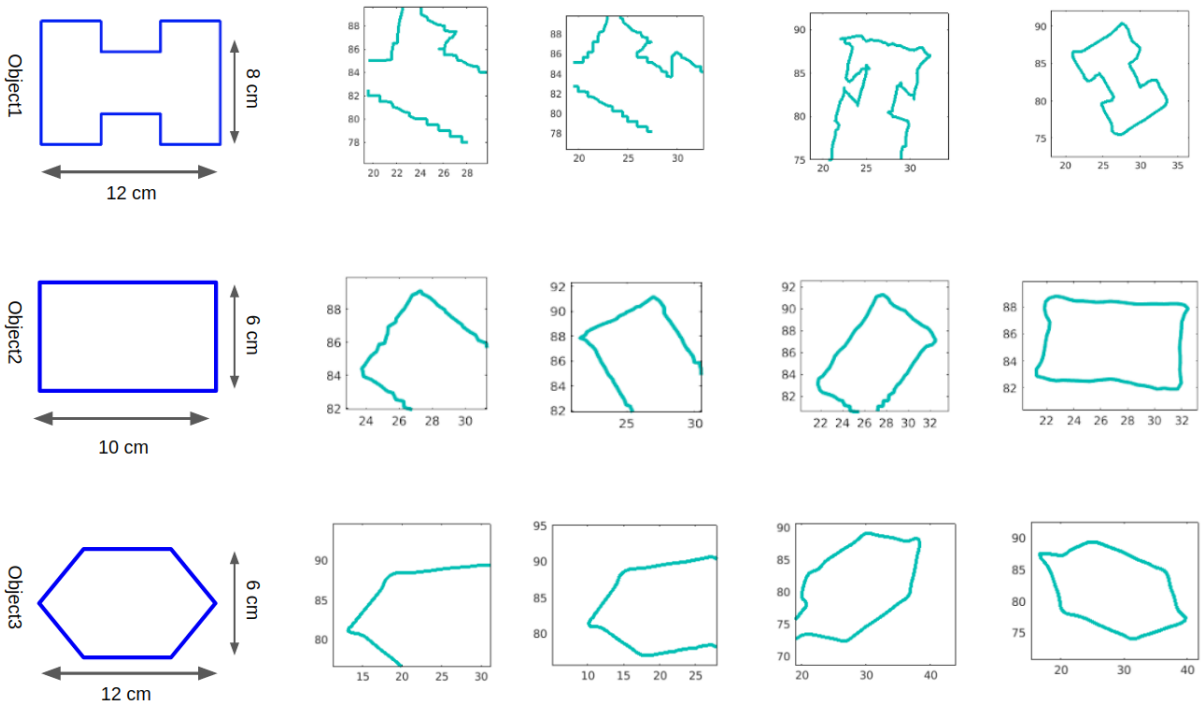


Figure 3.11: Progressive reconstruction of three objects in 2D. The real shapes of the objects are shown on the left. The progressive reconstruction of each object is shown from left to right.

grasps with each object. Figure 3.11 shows the three different objects used in the simulation and their progressive reconstruction.

We evaluated the reconstruction accuracy by comparing the reconstructed shape with the real shape of the object. We computed the reconstruction error, which represents the maximum distance between the estimated model and the real model. This error is calculated for different ranges of object motion uncertainty. Results are shown in Figure 3.12.

When the object pose is perfectly known, the mean reconstruction error is less than $3mm$. This corresponds to the uncertainties of tactile measurements. For high uncertainties in the object pose, the particle filter algorithm efficiently tracks the pose of the object, and the reconstruction error is always less than $9mm$.

3.4.2 *3D simulations*

This part describes the process of simultaneous reconstruction and localization of objects in three dimensions. We used three different objects in the simulations. Figure 3.13 illustrates the progressive reconstruction of the objects.

3.5 EVALUATION

The progressive reconstruction of some objects is shown in Figure 3.11. The qualitative results show that the object is correctly reconstructed even with high noise. These results were obtained in simulation after performing 16 grasps and planar reorientations with

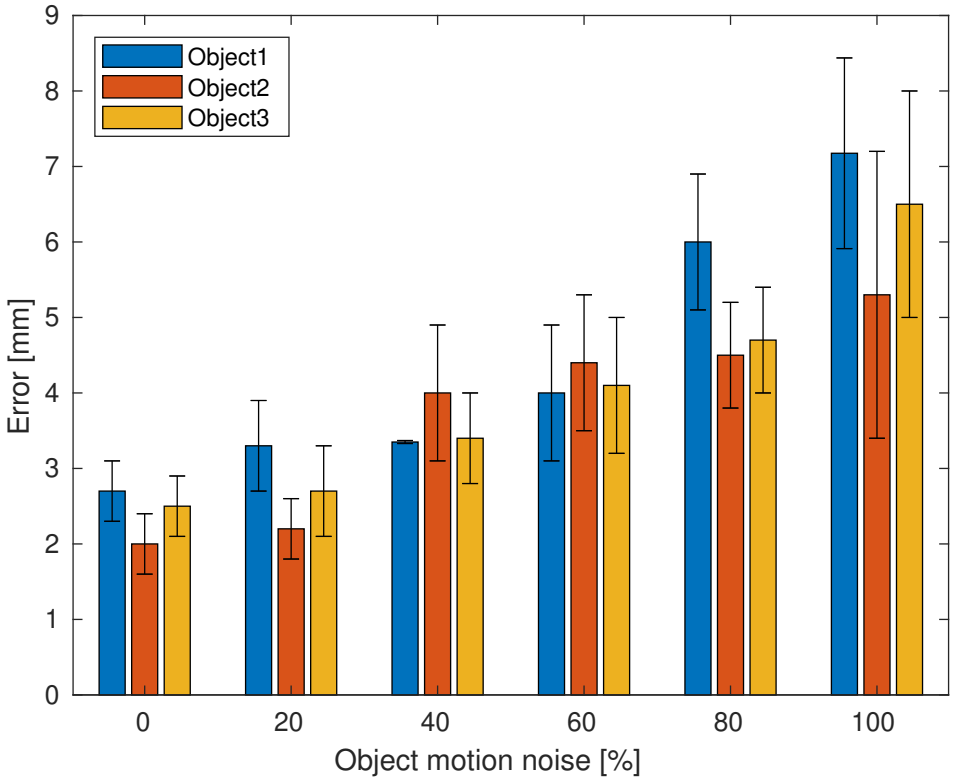
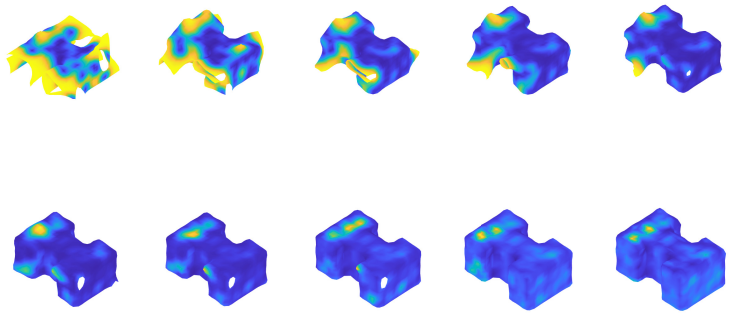


Figure 3.12: Reconstruction errors with different ranges in the object motion noise. 100% of the noise represents a Gaussian noise with $\sigma_t = 10mm$ on the object translation and $\sigma_r = 20deg$ on the object rotation.

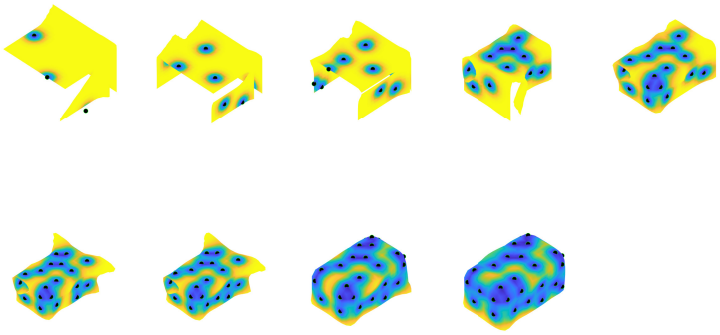
each object. In the following, we will evaluate the different parts composing our framework. We first evaluate the particle filter design. Then, we present the performance of the graph optimization.

3.5.1 *Particle filter evaluation*

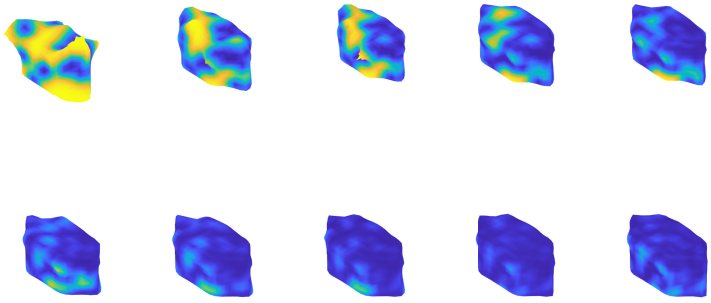
The particle filter performs the tracking of the object pose during successive grasps and manipulations. The success of the tracking depends on the motion and measurement models and the number of particles. Each particle is evaluated according to the measurement model defined in Equation 3.9, which measures the correspondence between two successive measurements. In our case, it is very challenging to measure the correspondence because of the sparsity of measurements. Many relative measurements may have a high weight with this measurement model, which induces large errors in the pose estimation and the reconstruction. To avoid such problems, we force our measurements to first find at least four contacts from the two successive measurements that are sufficiently close and check the observability with the method described in section 3.3.2. Otherwise, a zero value is assigned to the measurement likelihood.



(a) Object 1



(b) Object 2



(c) Object 3

Figure 3.13: Progressive 3D reconstruction of three objects

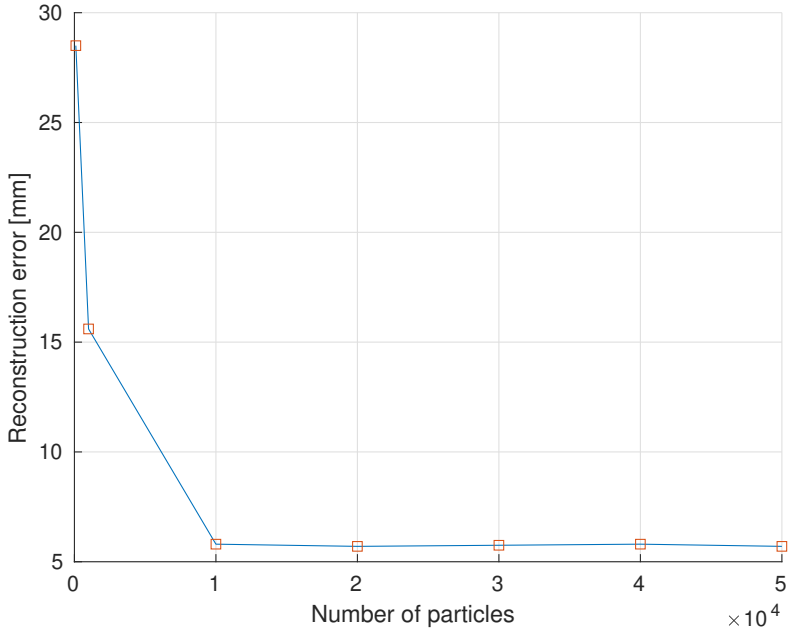


Figure 3.14: Reconstruction error as a function of particle number

In Figure 3.14, we represent the success rate of the particle filter as a function of the number of particles. Due to the conditions described in the paragraph above, it happens that all particles are weighted with a zero value, which causes the estimation to fail. This happens when the number of particles is insufficient. Simulations show that we need at least 10 000 particles to avoid the failure of the tracking. The algorithm was implemented on MATLAB, and it takes average 1.5 seconds to process this number of particles on a PC with an i7 processor.

3.5.2 *Pose graph evaluation*

The role of pose graph optimization is to enhance reconstruction accuracy by finding similarities between the measurements. We evaluate the performance of the graph according to the number of constraints in the graph. Figure 3.15 represents the error in shape reconstruction as a function of the number of constraints in the pose graph. The curves show that the more constraints in the graph, the better the accuracy of the reconstruction. The error of reconstruction is evaluated by the Hausdorff distance between the reconstructed model and the real model of the object.

Figure 3.16 shows the reconstruction accuracy of the three objects, using only the particle filter, and applying the graph optimization. The results show that the pose graph optimization significantly improves the accuracy of the reconstruction.

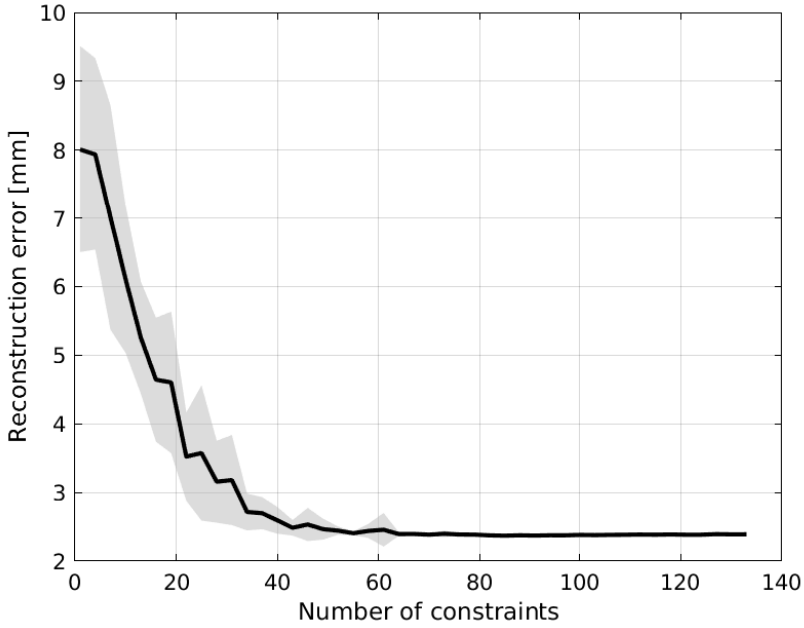


Figure 3.15: Reconstruction error as a function of the number of constraints

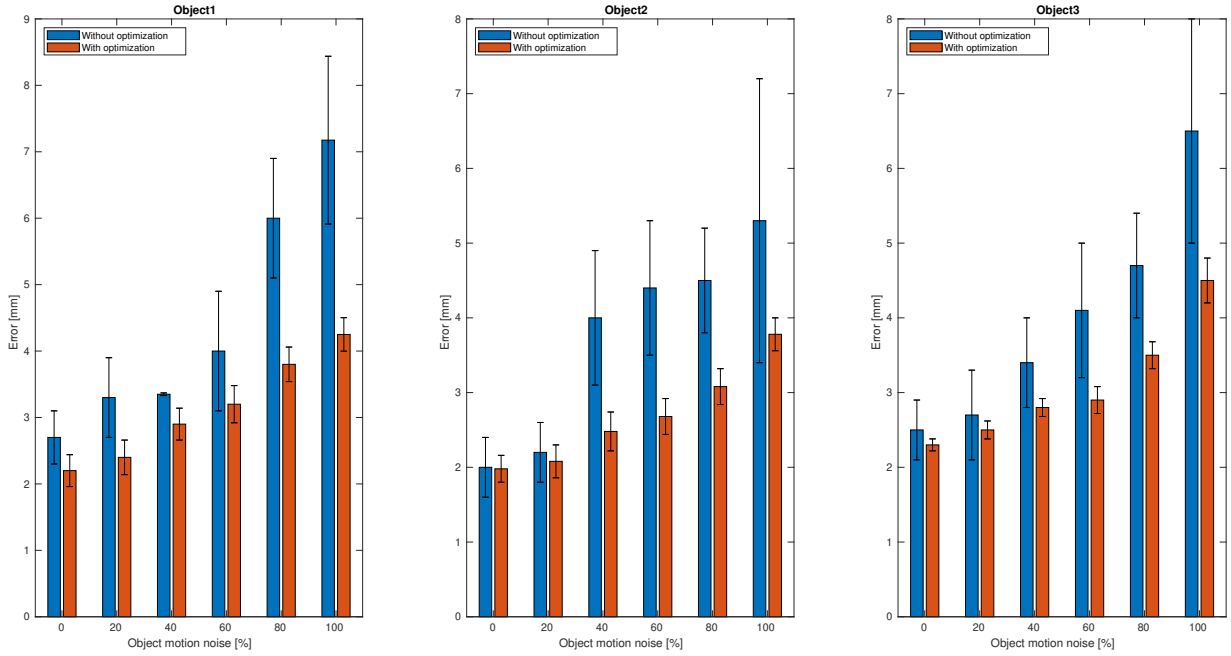


Figure 3.16: Evaluation of 2D reconstruction accuracy of three objects.

3.6 EXPERIMENTS

This section will analyze the performance of the incremental reconstruction and localization of objects from tactile data. We will first detail the experimental setup and then present our experiments for 2D and 3D shape reconstruction.

3.6.1 *Experimental setup*

Experiments were performed on the Shadow robotic hand equipped with ATI nano17 tactile sensors. The experiments consisted in collecting contact data from successive grasps and manipulations of an object. The object was manipulated on a horizontal surface. The robotic hand was manually guided to grasp the object to collect new contact data, then to reorient the object and execute a new grasp. The grasping and manipulation steps are shown in Figure 3.17.

The contact data is first obtained in the forearm frame and then transformed into the world frame. The world frame is defined on the workspace, at the surface of manipulation. We first had to manually identify the transformation between the forearm frame and the world frame.

In order to evaluate the accuracy of the reconstruction and localization, we installed a vision system to obtain the actual pose of the object in the world frame. The system uses two ArUco markers, one represents the world frame, and the second one is installed on the bottom of the object as the object frame. The camera was installed under the surface of manipulation to be able to detect both markers simultaneously. The surface of manipulation is transparent. Figure3.18 details the experiment setup.

We first evaluated the accuracy of the tactile data. The contact coordinates have an error of about $\pm 2.2mm$, and the normals contain errors which vary in the interval of [-8,+8] degrees.

3.6.2 *2D reconstruction*

Contour recovery is first made by combining the GPIS and particle filter algorithm, ensuring the recursive pose tracking. For this experiment, we first performed a series of measurements with the hypothesis that the relative pose always exists between the data obtained in successive grasps. The error in the motion of the object is experimentally evaluated at the magnitude of $10mm$ and $20deg$ on the rotation.

The pose graph was implemented later, and the evaluation shows the contribution of the pose graph optimization to the reduction of the reconstruction error. The error was evaluated as the maximum distance between the reconstructed contour and the actual contour of the object after alignment.

Figure 3.20 shows the accuracy of the reconstruction with the particle filter only (in blue), and after performing the pose graph optimization (in red). In this experiment, we assumed that the relative pose between two successive grasps always exists. The particle filter shows

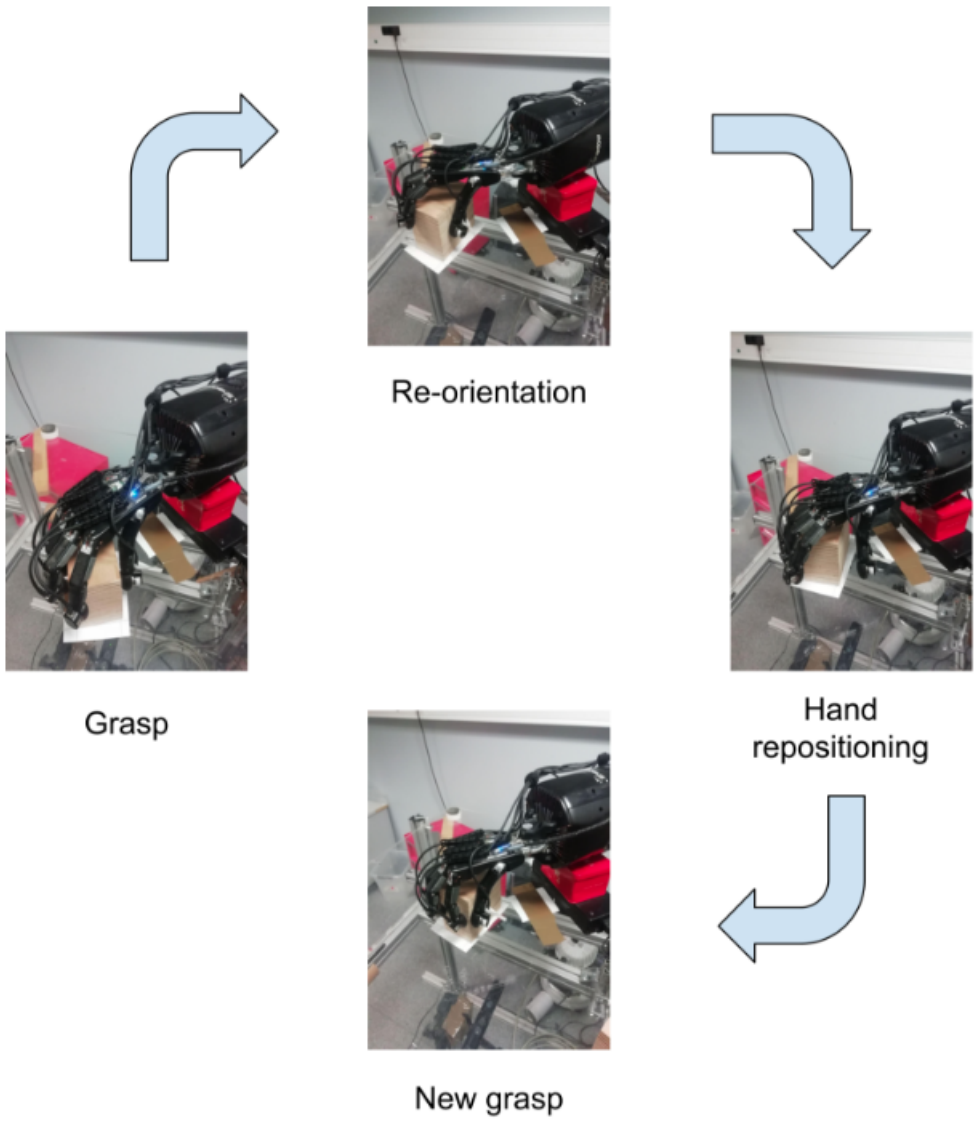


Figure 3.17: Illustration of the process of successive grasps and reorientations.

an acceptable performance in tracking the object pose. However, small errors exist and accumulate during successive grasps and manipulations. The pose graph optimization reduces these errors and improves the accuracy of the reconstruction.

3.6.3 *3D reconstruction*

For the 3D reconstruction, the data are collected in 3D coordinates with successive grasps and manipulations. In this experiment, it is very complex to implement a recursive particle filter to the tracking problem. The particle filter may fail at a given step to find the relative pose. In this case, we use pose graph optimization for the global estimation. The particle

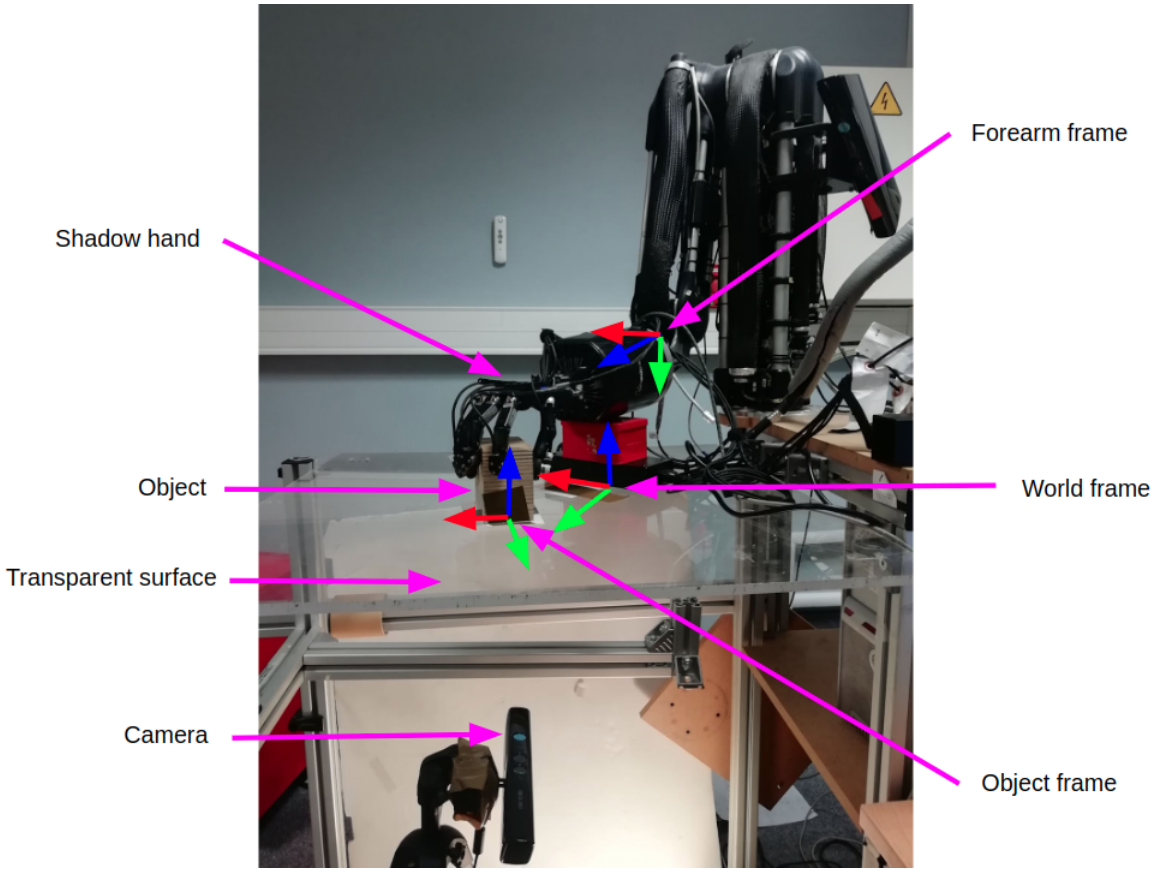


Figure 3.18: Experimental setup

Figure 3.19: Objects used in the experiments

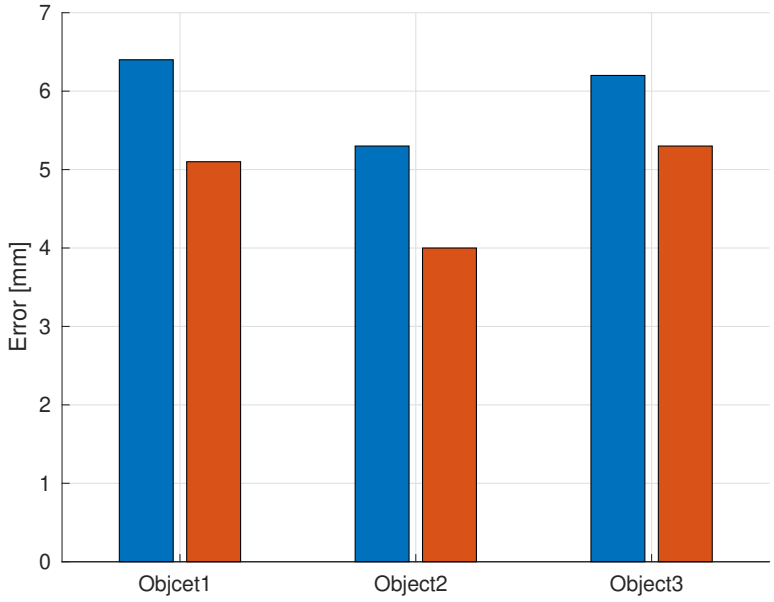


Figure 3.20: Error evaluation for contour reconstruction. In blue: reconstruction with particle filter tracking only. In red: reconstruction with particle filter and pose graph optimization

filter algorithm can only build some separate parts of the object. The accuracy of the reconstruction is shown in Figure 3.21.

3.6.4 *Discussion*

In this experiment, we demonstrated the performance of the reconstruction of unknown objects with noisy and sparse data in an uncertain environment. The reconstruction was performed incrementally with successive grasps. We first used a particle filter algorithm to correct the pose between successive grasps. This works well in the 2D case and produces exploitable performance, although it results in some accumulated errors in the recursive process. These errors were corrected by the pose graph optimization, which considers all relative poses between grasps and leads to better performance. For the 3D case, the particle filter failed at a certain step of the reconstruction due to the sparsity of the data. Optimizing the pose graph avoids problems of recursive tracking, such as cumulative errors, and lack of observability. These experiments show the complementarity of recursive and global estimation in building accurate models of objects.

3.7 CONCLUSION

To summarize, this chapter has considered the problem of object tracking and pose correction during shape reconstruction. To track the pose of the object while reconstructing its shape, we used the probabilistic filtering technique, which is the most suitable for the nature of the problem and for the nature of the data. We added background processing in order to enhance the reconstruction by exploiting some geometric features of the object

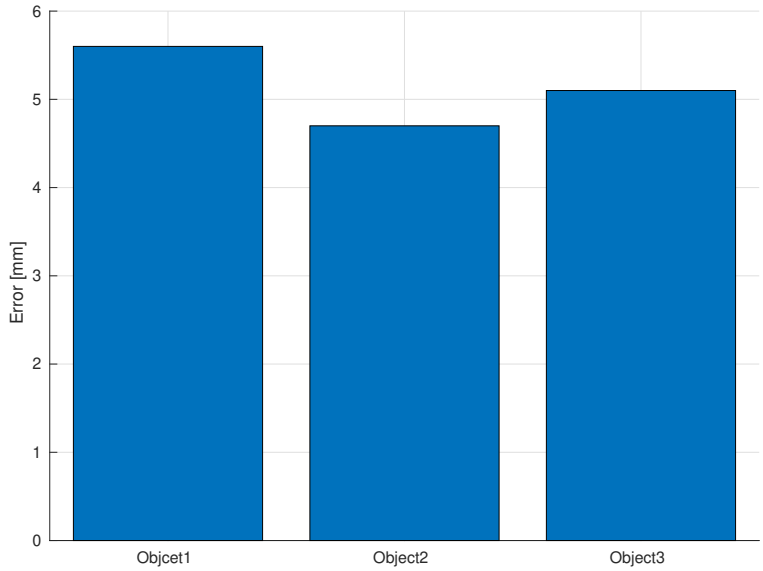


Figure 3.21: Evaluation of the 3D reconstruction.

shape in a so-known pose graph optimization.

The most challenging task in this framework is the search for matching between a small quantity of contact data and a partial model of the object. Resolving this problem was very important for precise shape reconstruction. For this purpose, we had to make some simplifications in the shape of objects by considering objects with faces only.

For the reconstruction of complex shapes, our framework can be adapted to more advanced tactile sensors that are able to get more complex geometric features in the shape of the object.

4

ACTIVE EXPLORATION

This chapter addresses the problem of autonomous tactile exploration of unknown objects by robotic manipulators. We present the different approaches used in this field and propose a framework for the active exploration and localization of unknown objects.

The work presented in this chapter is complementary to the one presented in the previous chapter. The previous chapter was devoted to the problem of simultaneous reconstruction and localization of unknown objects based on tactile information. This information was acquired by manually guiding the robot to feel the object. In this chapter, we aim to develop a framework in which the robotic manipulator acts autonomously to obtain tactile information in the task of recovering the shape of unknown objects.

4.1 PROBLEM FORMULATION

The goal of our framework is to have the robot perform autonomous actions that allow acquiring contact data in order to accurately reconstruct an unknown object. As a first naive solution, we can think of performing a set of actions (different grasps) that will be repeated sequentially. However, with this solution, some actions might be useless as they always acquire the same data, which can slow down the reconstruction process. An optimal solution to explore the object with a minimum number of actions and thus a minimum exploration time is to choose the actions in such a way that each action improves the knowledge about the object. For this purpose, we first need to quantify the uncertainty of the object's shape and its pose, then choose the appropriate action that reduces this uncertainty.

To develop a solution, we must first ask ourselves how to quantify the uncertainty about the shape of the object and identify which parts that are well-known and which need to be explored further. Since the pose of the object is considered in this problem, we must also quantify the uncertainty about it. At the second level, we need to specify the actions of the robotic manipulator that reduce these uncertainties and improve the knowledge of the object. Then, we need to organize the priorities for the most important information (information on the shape or on the localization). Finally, we aim to ensure that decision-making about action is more optimal than random or repetitive actions.

4.2 STATE OF THE ART

Tactile exploration is an important process for robots to obtain characteristics of unknown objects. This problem has been studied for more than two decades and remains a relevant research topic. Its goal is to make robots more autonomous in their interaction with unknown environments. Many approaches have been proposed over the last two decades to address the problem of autonomous tactile exploration in order to achieve various objectives: shape reconstruction, object recognition, and object pose estimation.

In the context of this thesis, we are mainly interested in the reconstruction and localization of objects. In the following, we will present existing works related to these two objectives: exploration for reconstruction and exploration for localization.

4.2.1 *Active exploration*

Active exploration consists of autonomously choosing where to touch the object in order to improve the information on its shape. As tactile perception only offers local information on the surface of the object, and the amount of information that is collected is very small compared to the information that we can get from vision systems, the tactile sensing modality requires a lot of time to gather a sufficient amount of data that can represent the shape of the object. Sometimes the robot may touch the object several times at the same location, which is not informative and causes a delay in the exploration. For these reasons, taking optimal actions in the exploration is necessary to reduce the time of exploration and avoid performing non-informative actions. Existing works reason with the uncertainty of the object shape to decide where to touch the object by minimizing the robot's trajectory while exploring the object. Several approaches are proposed in this context, with different exploration actions depending on the structure of the robotic platform and the characteristics of its tactile sensors.

The selection of actions for unknown object exploration is made by quantifying the knowledge of the object and identifying the parts that need to be explored. Then, the necessary actions are performed with the robot to explore these parts in order to obtain new information. An early framework proposed in this field is based on potential field exploration that defines the direction of the exploration [58]. The Gaussian Process Implicit Surface (GPIS) technique has become a popular method for object modeling in the last decades. This probabilistic technique contains information about the uncertainty of the estimated model. The variance of the GPIS model has therefore been used to guide the exploration of objects [59] [60]. The entropy of the GPIS is also a measure of the uncertainty of the model. It was proved in [61] that the differential entropy is better than the variance of the GPIS with multi-contact exploration. Besides these formal measures of uncertainty, [62] proposed empirical criteria to evaluate the uncertainty of the model and guide the exploration. In some proposed approaches that model objects with occupancy grid maps, the entropy of the map is used to plan the actions for improving the knowledge of the object[63]. Active exploration can also reason regarding the path travelled by the end effector as in [64].

Paper	**Type of exploration**	**Action selection criteria**	**Single/multi contact**
GPAtlasRRT: A Local Tactile Exploration Planner for Recovering the Shape of Novel Objects [65]	Sliding motions and discrete touches	variance of GPIS	multi-contact
A potential field approach to the dexterous tactile exploration of unknown objects [58]	Sliding motions	Potential field	multi-contact
Active Learning with Query Paths for Tactile Object Shape Exploration [59]	Sliding motions	Variance of GPIS	Single contact
Active multi-contact continuous tactile exploration with gaussian process differential entropy [61]	Sliding motions	Differential entropy of GPIS	multi-contact
Active tactile exploration based on cost-aware information gain maximization [62]	Discrete contacts	IGEF (Information Gain Estimation Function)	Single contact
Active Tactile Object Exploration with Gaussian Processes [60]	Discrete contacts	Variance of GPIS	Single contact
Active Visuo-Haptic Object Shape Completion [63]	Discrete contacts (to complete vision)	Variance of the grid map	Single contact

TANDEM: Learning Joint Exploration and Decision Making with Tactile Sensors [66]	Discrete contacts	entropy of a grid map	Single contact
Active tactile exploration with uncertainty and travel cost for the fast shape estimation of unknown objects [64]	Sliding motions	variance of the GPIS	Single contact
Tactile identification of objects using Bayesian exploration [67]	Discrete contacts	Bayesian exploration	Single contact

Table 4.1: State of the art on active tactile exploration.

4.2.2 *Active localization*

Object localization with tactile data was studied in the previous chapter. The most relevant approaches in this context were presented. However, in most existing work, this problem has been treated passively; that is, the data is obtained in a guided way (the robot is manually guided to acquire data). Some propositions have been made in the last few years on active object localization from touch. The problem is based on the definition of an information gain criterion that measures the amount of information that could be gained by performing a specific action. Thus, the algorithm chooses *the next best touch* [68] that reduces the uncertainty about the object pose.

Before making a decision on what action to perform, a quantification of the uncertainty about the pose is necessary. Since the problem of object pose estimation is a probabilistic problem (except with optimization techniques such as ICP) and is commonly solved with particle filtering techniques, the pose of the object is represented by a density function. The uncertainty of the density function is computed by the Shannon entropy formula, which was used in [69]. [70] proposed a different method to compute the entropy based on Gaussian kernels. The information gain of a specific action is quantified as the difference between the prior estimation and the posterior estimation [71]. Another criterion to measure the information gain in active exploration is the Kullback–Leibler (KL) divergence [72] [73]. In [74] different information gain criteria were evaluated in the problem of active localization: Kullback-Leibler divergence, Renyi divergence, Fisher information metric,

and Bhattacharya distance. The obtained results show that the efficiency of the different criteria is similar. However, Renyi divergence (which is a variety of the Shannon entropy) shows a little better performance than the other information gain criterion.

4.2.3 *Discussion*

As we have shown in this section, the problem of autonomous action selection is based on information theoretic criteria that are known to all and widely used in the field of active robotic perception. We observe through the reported approaches that active action selection is made for a single objective (shape recovery, object recognition, or object localization). To the best of our knowledge, there is no work that studies the problem of active action selection for simultaneous reconstruction and localization, which is the goal of our work.

4.3 EXPLORATION STRATEGIES

In order to explore the shapes of objects with tactile perception, humans use different strategies [75]. The most common is bi-manual exploration, where they can hold the object in one hand and explore it with the other hand. If the object is relatively heavy and can not move during the exploration, humans use sliding motions on its surface to discover its surface. And sometimes roll it between the fingers.

For robotic manipulators, the exploration of unknown objects requires many considerations related to kinematic planning, control of the manipulation, and the processing of perception data. Therefore, in many of the proposed frameworks, there are some fixed restrictions.

Manipulators with rich kinematics can perform bi-manual exploration like humans [28] and can build 3D models of objects. This has been made by equipping the hand with a large number of tactile sensors. The robotic manipulators equipped with a single end-effector can perform sequential probing on the surface of the object, but this requires a significant amount of time to complete the exploration. In some situations, sliding motions over the surface of the object can be performed to recover the shape. However, this is only possible with relatively smooth shapes and by using a control module to control the trajectory of the manipulator and maintain a certain force on the surface of the object[10]. Rolling the object between two tactile surfaces is also possible to recover its surface [76] [77]. However, this option is also limited to objects with smooth shapes. Sliding and rolling have many disadvantages for shape reconstruction: for sliding motions, the object must be fixed during the exploration.

Multi-fingered robotic hands equipped with fingertip sensors offer more abilities to explore shapes with different characteristics [9]. To prepare this task in a formal way (to avoid any unwanted behaviours), we have to make a set of assumptions to define the exploration. The first assumption is that, since the shape of the object is not known, we can not lift it because its stability would not be guaranteed. We assume that the object can stand on a supporting surface. Another choice is to do finger gaiting, which can be a

good solution for continuous exploration by exploiting the dexterity of the robotic hand, but planning the motions is kinematically complicated.

Instead, we opt for multiple grasping, where the robot closes its fingers on the object to obtain contact data, then releases it and grasps it with another posture. This is the most appropriate exploration strategy, which is less complex for motion planning and allows one to fully exploit the dexterity of the hand and arm.

4.4 INFORMATION GAIN CRITERIA

This section describes the information gain criteria for quantifying the uncertainty of the object's shape and pose mathematically.

4.4.1 *Entropy*

The entropy measures the disorder of the probability distribution. The value of the entropy is high for large uncertainties and low for large certainties. The equation of entropy used in most existing works is:

$$H(x, M|u, z) \tag{4.1}$$

With:

- x: the pose of the object
- M: the model of the object
- u: the exploration action
- z: the tactile measurement

The Shannon entropy is given by:

$$H(P) = \sum_i -P(x_i)\log(P(x_i)) \tag{4.2}$$

Where $P(x)$ is a probability distribution over the pose of the object. In the case of the particle filter, the entropy can be approximated as follows:

$$H(P) = \sum_i -w_i \log(w_i) \tag{4.3}$$

w_i is the normalized weight of the particle i. This entropy is high when all particles have the same probability, and the entropy takes its low value when one particle is more probable than the other particles.

Reney entropy is a variety of the Shannon entropy that can be conditioned by a parameter α and can be expressed as follows:

$$H_\alpha(P) = \frac{1}{1-\alpha}\sum_i P_i \log(P_i) \tag{4.4}$$

4.4.2 *Mutual information*

Some works consider the mutual information as the difference in the entropy between the current estimate $H(x)$ and the future entropy conditioned by the measurement z represented by $H(x|z)$ and expressed as follows:

$$M(x|z) = H(x) - H(x|z) \tag{4.5}$$

This equation gives the gain information after obtaining the measurement z.

4.4.3 *KL Divergence*

The Kullback–Leibler divergence measures the divergence between two probability distributions in the prior and posterior. It is computed as follows:

$$D_{KL}(P||Q) = \sum_i P_i \log(\frac{P_i}{Q_i}) \tag{4.6}$$

Where P and Q are the two probability distributions. In the context of our problem, P can represent the prior distribution over the object pose, and Q is the posterior probability on the object pose after performing the action $\hat{a}$ and obtaining the measurement $\hat{z}$.
The amount of KL divergence is small when the distribution Q is similar to the distribution P and high when the distribution Q is predominantly different from P.
Figure 4.7 shows an example of a probability distribution represented in 2D of the prior probability and three possible distributions for the posterior probability. The KL divergence shows that the second probability is the one that allows gaining the maximum information about the object pose.

4.4.4 *Uncertainty evaluation about the object shape*

This section introduces the equations for the variance of the GPIS model, its entropy, and the information gain function proposed in [78].

4.4.4.1 *Variance of GPIS*

The variance of the GPIS was already introduced in Chapter 2, and defined as follows:

$$V = k_{**} - k_*^T (K + \sigma_n^2 I)^{-1} k_* \tag{4.7}$$

More details on this equation are presented in section 2.2 in Chapter 2.

4.4.4.2 *Entropy of GPIS*

The differential entropy of GPIS was proposed by [61], and used as a measure to guide the exploration of the surface of an object.
Given data $D = \{p_i, y_i\}$ where p_i and y_i are the contact coordinates of contact points and

the state of contact ($y_i = 0$ if there is a contact), respectively.
The entropy of the Gaussian Process is given by:

$$h = \frac{1}{2}\log\left((2\pi e)^d \det(K_*)\right) \tag{4.8}$$

with d the dimension of the data, and K_* a matrix constructed from the covariance matrices as follows:

$$K_* = \begin{bmatrix} K & k_* \\ k_x^T & k_{**} \end{bmatrix} \tag{4.9}$$

Then, the differential entropy is evaluated with the following formula after removing impertinent constants:

$$h = \log(\det(K_*)) \tag{4.10}$$

4.4.4.3 *Information gain estimation function (IGEF)*

This method was proposed by [78] and consisted of a combination of four metrics. The first metric assigns high values for target points p_* that are distant from the current contact point p:

$$\Psi_1 = 1 - exp(-\frac{\|p - p_*\|^2}{\sigma_1^2}) \tag{4.11}$$

The second metric evaluates the path length from the current point p to the desired target p_*:

$$\Psi_2 = \frac{1}{Length(path(p, p_*))} \tag{4.12}$$

The third metric assigns high values for points that are closer to the already explored area, an offset mean μ_3 is introduced, and this metric is given by:

$$\Psi_3 = exp(-\frac{(\|p - p_*\| - \mu_3)^2}{\sigma_3^2}) \tag{4.13}$$

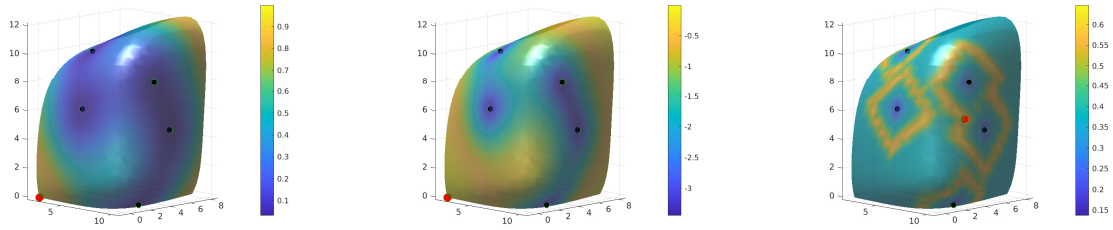
Finally, the last metric evaluates the fingertip rotation during the exploration. This is defined by an angular kernel:

$$\Psi_4 = exp(-\frac{2sin^2(\frac{1}{2}acos(n, n_*))}{\sigma_4^2}) \tag{4.14}$$

n is the normal on the current contact, and n_* is the estimated normal on the target point. σ_1, σ_3 and σ_4 are parameters tuned manually. The four metrics are combined to form the general exploration function Ψ:

$$\Psi = \prod_{i=1}^{4} \Psi_i \tag{4.15}$$

Figure 4.1 shows the quantification of the uncertainty for a cube after obtaining five contact points and estimating its surface by the GPIS method. A first view of the three GPIS



(a) Next best target to explore set by maximum variance

(b) Next best target to explore set by maximum entropy

(c) Next best target to explore set by maximum Information Gain Estimation Function (IGEF)

Figure 4.1: Representation of the uncertainty in shape modeling with three different techniques

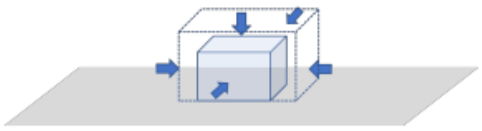
representations shows that there is a different quantification of the uncertainty according to the different methods. The variance of the GPIS gives high uncertainty to the areas far from the contact points, a medium uncertainty to the areas near the contact points, and a low uncertainty to the areas closer to the contact points. The uncertainty measured by the entropy function is similar to that one generated by the variance. For this reason, the variance function seems to be more adequate for estimating the uncertainty than the entropy, which is more complicated to compute. The IGEF method assigns a high uncertainty to the points near the contact points and a low uncertainty to the area away from the contact points because it quantifies the distance to the contact points.

4.5 SIMULATIONS

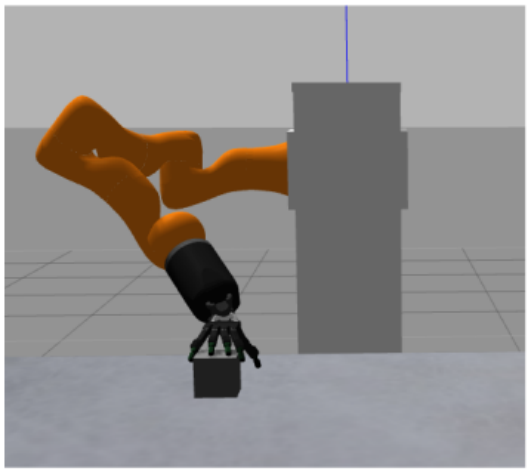
4.5.1 *Simulations on active shape reconstruction*

In this simulation, we are interested in the reconstruction of an unknown object fixed in the workspace of the robot. As in the previous chapter, the arm and hand must perform successive grasps to recover the shape of the object. In this simulation, the object will be modeled with the GPIS technique, and we choose the variance of the GPIS as a criterion for the uncertainty on the recovered surface. Firstly, the robot has to delimit the area of the object, which is done by defining a bounding box around the object. The bounding box is defined by performing a set of actions with the robotic hand to define the bounded box. Figure 4.4 illustrates this process of defining the bounding box.

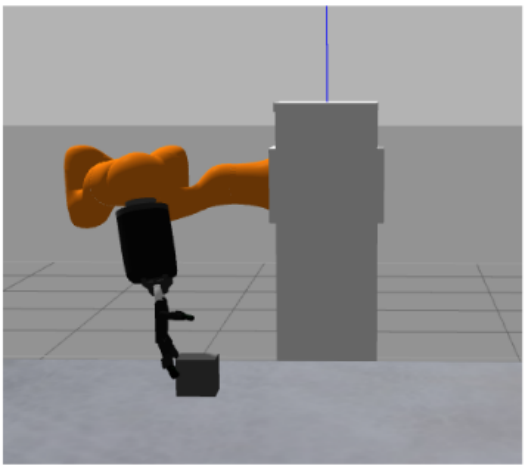
After defining the bounding box, we have the area where to create the GPIS representation. The hand performs successive grasps and updates the GPIS representation of the object consequently. To define the next grasp, we compute the variance of the GPIS representation, and the next grasp tries to reach the area of the object with maximum uncertainty. However, this approach is not suitable because the hand is always directed to a region distant from the previous grasp because the uncertainty of the GPIS model is larger at the extremities. To overcome this situation, we tried to make the next grasp closer to the previous one to avoid the hand travelling far from the current pose and to ensure some continuity in the reconstruction. The best way to achieve this is to implement



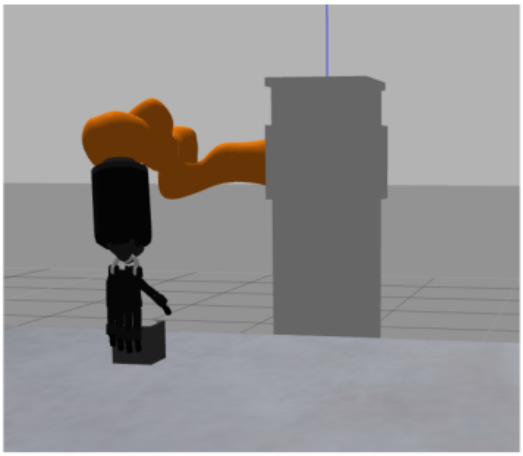
(a) The bounded box to be delimited by the robot



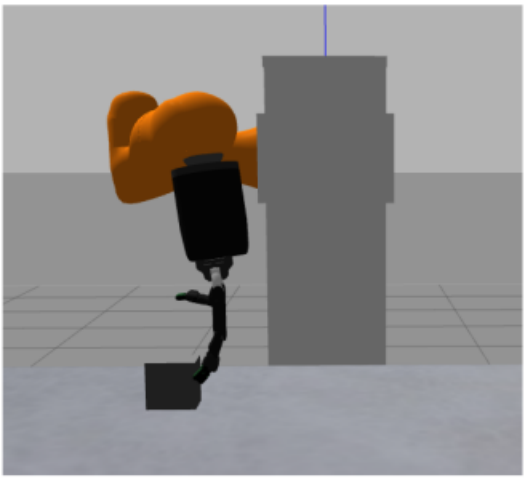
(b) Top side detection



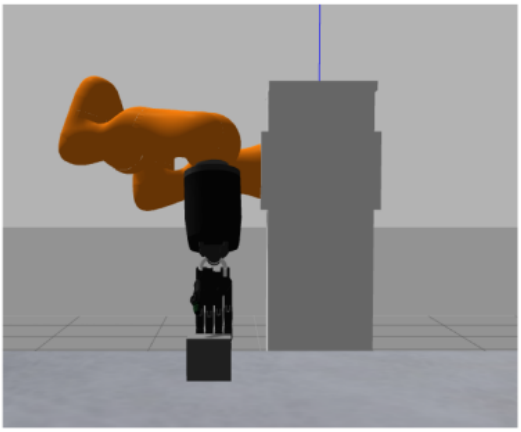
(c) Left side detection



(d) Front side detection



(e) Right side detection



(f) Back side detection

Figure 4.2: Delimitation of the area containing the object to be explored

a clustering algorithm that divides the shape of the object into parts according to the

variance and distance.

We have chosen an algorithm inspired by the one proposed in [79] in order to group the points of the GPIS representation whose variance is higher than a predefined threshold value in different clusters. This is called hierarchical clustering.

Hierarchical clustering works as follows: each element is considered as a singleton, and then the clusters are merged according to their similarity in a hierarchical way.
To develop our clustering algorithm, we first need to define the "features" on which segmentation will be based. To do so, firstly, the contact areas must be grouped according to their distance and must not be spread. Secondly, the curvature of each contact area must not vary much, so the surface must be smooth. Thirdly, the elevation of the contact area needs to be taken into account to allow the robot hand to be positioned in a way that it can reach the whole area. Our clustering algorithm is based on three principal features:

- **The Euclidean distance:** This defines the Euclidean distance between points $p_i(X_i, Y_i, Z_i)$ as the first criterion. This is expressed as follows:

$$d_{euc} = \sqrt{(X_1 - X_2)^2 + (Y_1 - Y_2)^2 + (Z_1 - Z_2)^2} \tag{4.16}$$

- **The Normal orientation:** This takes into account the angle between two normals of two points. The normals of the points reflect the curvature of the surface locally around these points. The angle between two normals is estimated as follows:

$$d_{normal} = \frac{\vec{n_1} \cdot \vec{n_2}}{\|\vec{n_1}\| * \|\vec{n_2}\|} \tag{4.17}$$

- **The height of the point:** This measures the elevation of the point according to the Z-axis

$$d_{height} = \sqrt{(Z_1 - Z_2)^2} \tag{4.18}$$

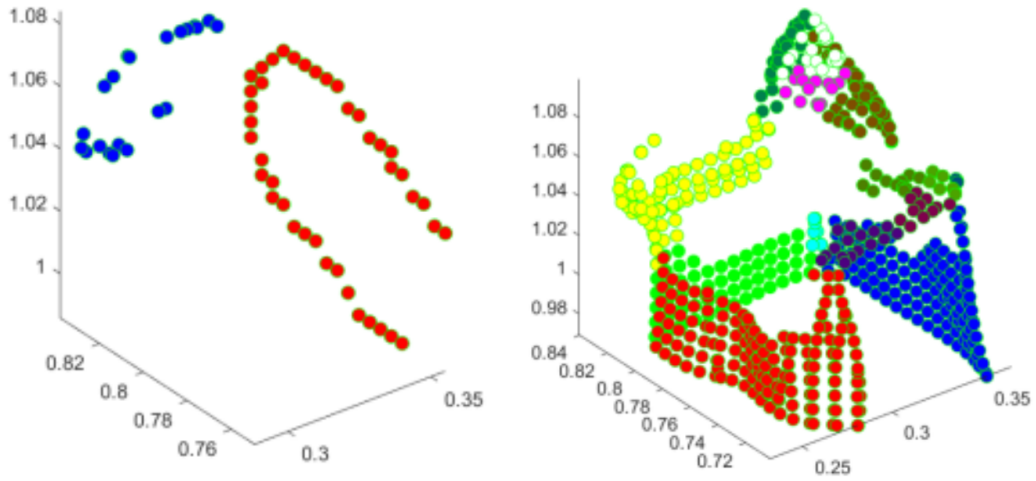
Thus, the dissimilarity equation is defined as follows:

$$D = 0.5 * d_{euc} + 0.4 * d_{normal} + 0.1 * d_{height} \tag{4.19}$$

The coefficients are fixed manually.
The implementation of the clustering algorithm is detailed in Appendix B, and the result of the clustering is shown in Figure 4.3.

definition of the grasp postures

When the contact areas are defined by the clustering algorithm, the hand chooses the next grasp in a way to gather more information about the shape of the object. In the first grasp, the surface predicted with the GPIS model may not reflect the real shape of the object, especially in the areas far from the current contacts. Thus, we choose the next grasp in a contact area that is closer to the current grasp. This choice has another important advantage which is to obtain similar features with successive grasps to facilitate the scan



(a) Clustering result for a small dataset (b) Clustering result for a small dataset

Figure 4.3: An illustration of the performance of the clustering algorithm

matching for pose correction.

To move to the next grasp, the hand must change its posture according to the characteristics of the selected contact area. For this purpose, the palm of the hand will be moved to approximately face the selected contact area, and the fingers will be closed to get the contacts.

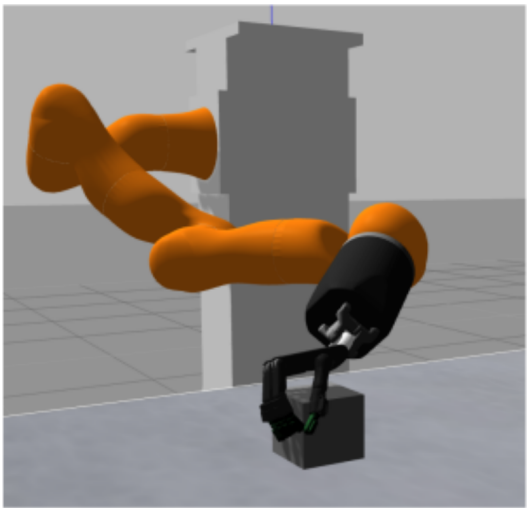
Examples of successive grasps are shown in Figures 4.4a, 4.4c, and 4.4e. And the final estimated shape is shown in Figure 4.5.

This figure shows the results of exploring a cubic object after performing more than 80 grasps. One of the limitations of the proposed solution is that sometimes the algorithm fails to reach some contact areas due to the unavailability of the inverse kinematic solution. This motivated us to integrate the manipulation of the object on the surface by translation and rotation motions, to change the pose of the object, and allow its exploration easily. However, this requires considering the tracking of the pose during the exploration.

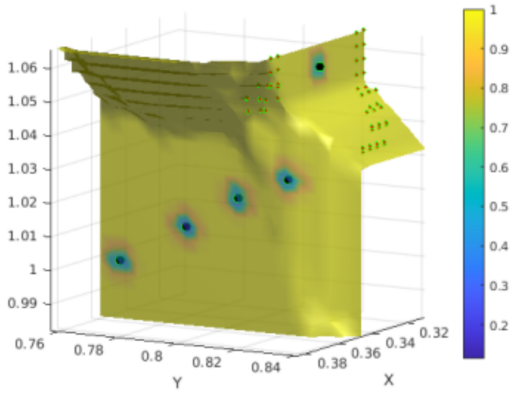
4.5.2 *Active reconstruction and localization*

In this section, we propose a simulation for active action selection for both shape reconstruction and localization. We present a method for active exploration of an unknown contour of an object by autonomous exploration with the robotic hand. So, the hand has to make a decision on which action to perform to improve the knowledge of the shape. Since the actions that can be performed require a long time to collect a sufficient amount of information, it would be necessary to choose at each step the action that offers the most valuable gain of information.

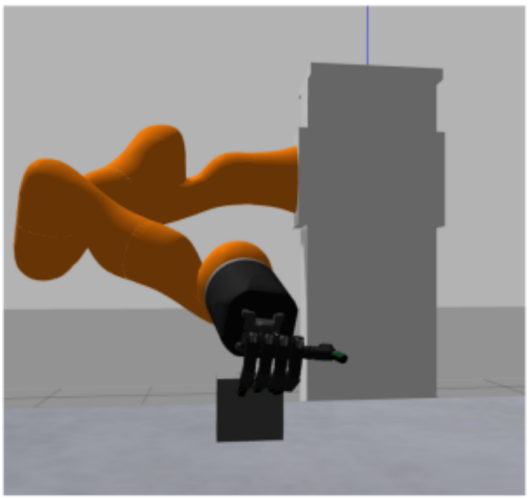
The term “action” refers to a grasp with the multi-fingered hand, characterized by the position and orientation of the palm with reference to the object frame. Some examples of grasps are shown in Figure 4.8. We have manually defined a list of ten candidate grasps



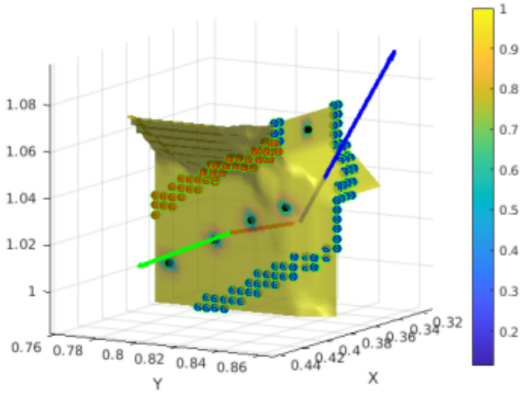
(a) First grasp



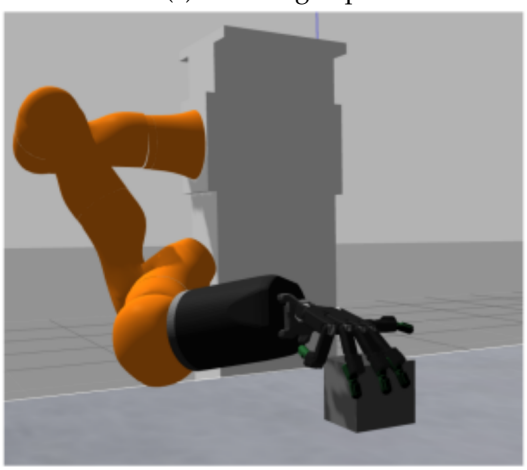
(b) First GPIS estimation



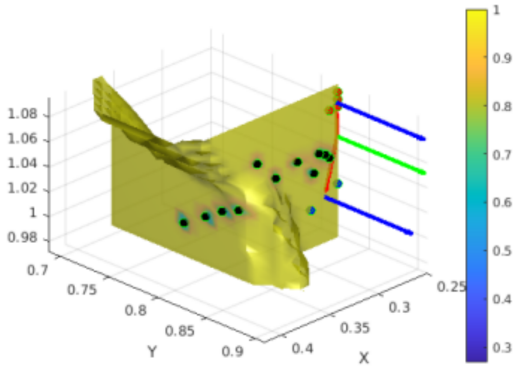
(c) Second grasp



(d) GPIS estimation update



(e) Third grasp



(f) GPIS estimation update

Figure 4.4: Progressive shape update with active exploration.

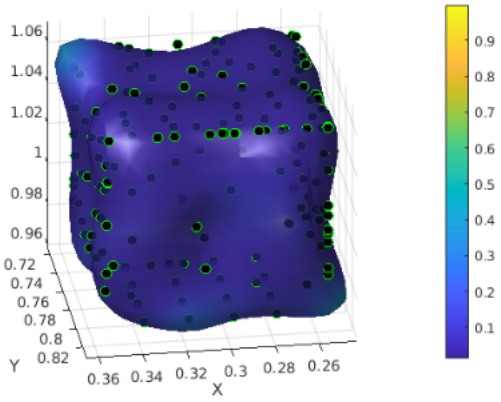


Figure 4.5: A reconstructed cube with 165 contacts.

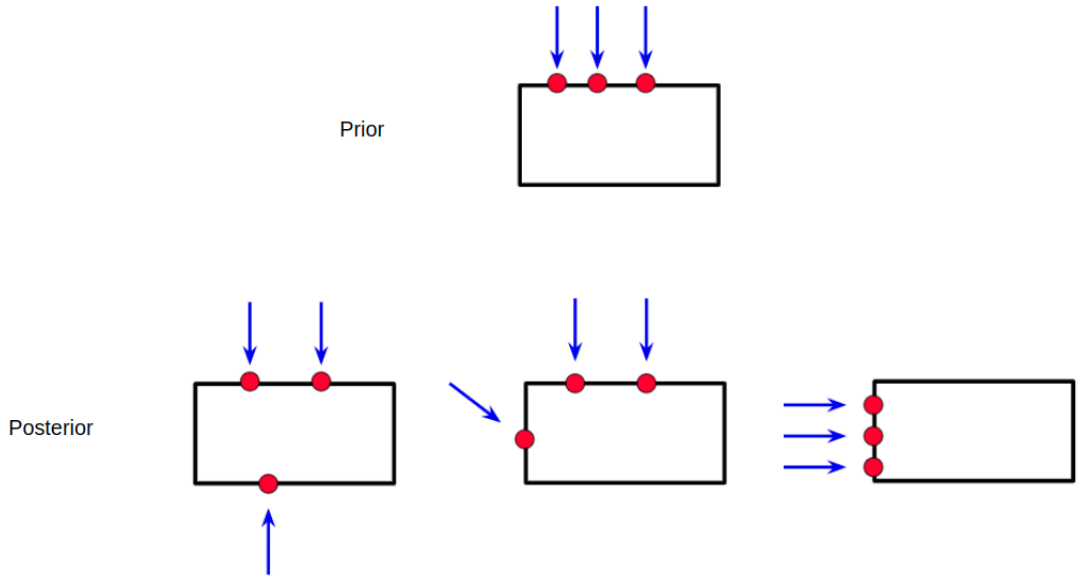


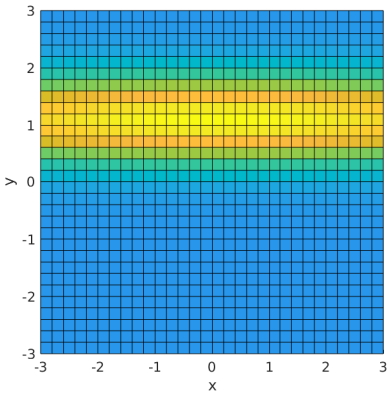
Figure 4.6: An example of possible actions to improve the pose estimation

with various postures (different poses of the palm and fingers with reference to the object) to be used in the following simulations.

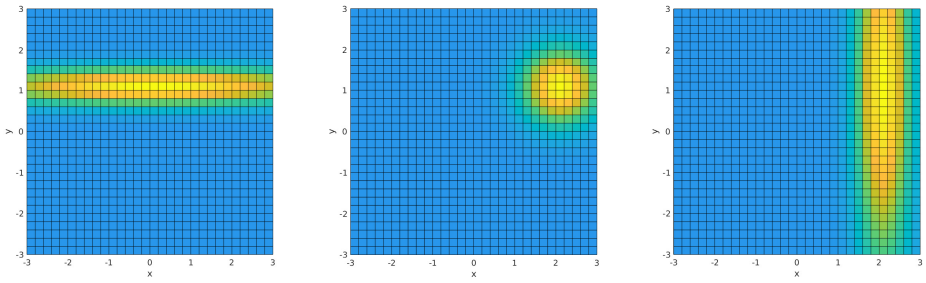
This simulation is divided into two parts. First, we consider only the reconstruction of the shape of an object in a fixed pose. Then, we consider the estimation of both the shape and the pose of the object in the second part. The shape to be reconstructed is of a rectangular object of dimension $10cmX6cm$.

4.5.2.1 *Part1: Reconstruction of the shape*

Shape reconstruction is done in an incremental process by successive grasps to collect data. In order to retrieve the shape with a minimum number of actions, each new action needs to offer more information about the unknown parts of the object. For this purpose, we use a metric to measure the gain of information of each next possible grasp. As the contour



Prior distribution



Posterior distributions

Figure 4.7: An example of the probability distribution of the pose of the object in 2D after simulating the actions shown in Figure 4.6

of the object is modeled using the GPIS technique, we use its variance as a measure of uncertainty.

The next best grasp is the one that reduces the uncertainty on the object surface in the areas with high variance. Thus, among the possible grasps, the optimal one is the one where the fingers are placed on the areas with high uncertainties. To do this, we perform a first grasp and build a partial model with the GPIS technique. We represent the points which are predicted to belong to the contour of the object and whose variance is below a fixed threshold. This allows us to choose a next grasp that is both on a high uncertainty area and closer to the current known areas. The optimal grasp is selected by evaluating

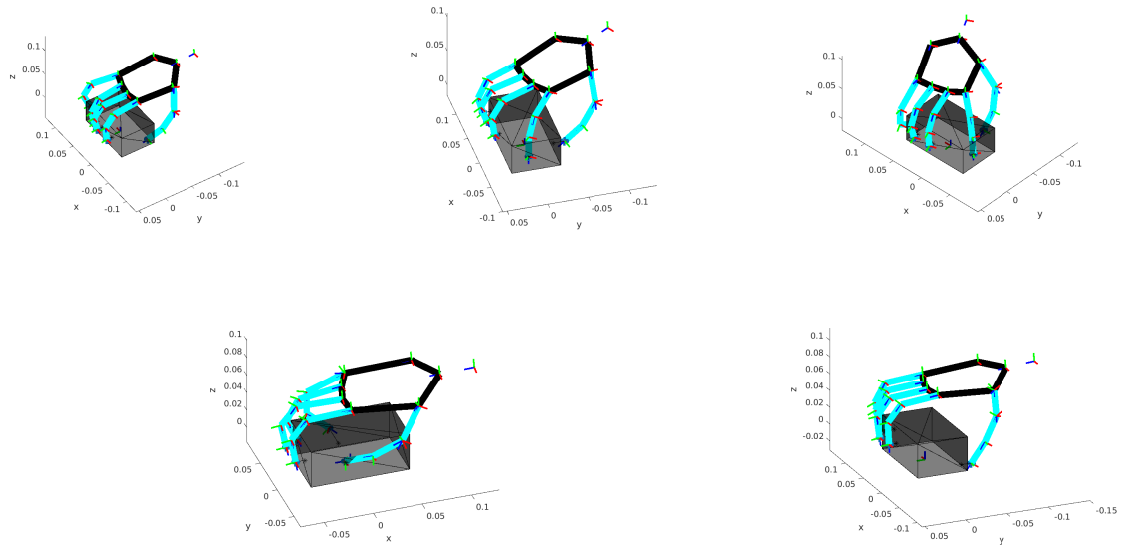


Figure 4.8: Examples of different grasp postures executed on a rectangular shape with the Shadow robotic hand model run on MATLAB.

Exploration strategy	Average Nb. of actions
Random exploration	22,1
Active exploration	16,3

Table 4.2: Results for shape exploration. The average number of actions needed to reconstruct 90 % of the contour of the object.

the reward of each possible grasp. This is computed by considering the variance of each contact point p_i, and defining the grasp reward G_r as follows:

$$G_r = \sum_{i=1}^{5} Var(p_i) \tag{4.20}$$

We evaluate this strategy by comparing it to a random grasp selection. The results are presented in Table 4.2. The table presents the average number of actions needed to explore 90% of the object contour. This average number is estimated over ten simulations. The results show that active exploration significantly reduces the number of actions needed to explore the object.

4.5.2.2 *Part2: Simultaneous active exploration and localization*

The following simulation focuses on the simultaneous reconstruction and localization of the object. We consider that the object is not fixed during the exploration and moves at each grasp under the effect of the fingers. So, the pose correction after each grasp is necessary to ensure an accurate shape reconstruction. The proposed algorithm focuses mainly on the incremental reconstruction of the shape but must ensure reliable pose correction

Exploration strategy	Average Nb. of actions
Random	56,7
$det(\Sigma)$	25,8
KL divergence	26,4

Table 4.3: Results for simultaneous exploration and localization of an object

of the object throughout the reconstruction process. For this purpose, it needs to manage some actions dedicated to the correction of the pose when it becomes highly uncertain.

The pose correction is performed with the recursive particle filter described in the previous chapter (chapter 3). At each step of the reconstruction, if the pose estimate is higher than a fixed variance threshold of the particle filter, the algorithm must perform an action to reduce the uncertainty of the pose. This action is selected according to a metric that measures the information gain. We specify that the information gain is evaluated on the observations that could be collected by all possible future actions, and that these observations are only hypothetical depending on the configuration of the grasp.

We evaluate our algorithm using different metrics for information gain: the variance of the PF, and the KL divergence introduced in Section 4.4.3. To compute the variance of the particles, we first assume that they follow a Gaussian distribution and use the following formula proposed in [70]:

$$\Sigma = \frac{\sum_{i=1}^{N} w^{[i]}(x^{[i]} - \bar{x})(x^{[i]} - \bar{x})^T}{1 - \sum_{i=1}^{N}(w^{[i]})^2} \tag{4.21}$$

Where $x^{[i]}$ and $w^{[i]}$ represent the particle i, and its weight, respectively. $\bar{x}$ is the mean value of the state x.

The matrix Σ can not be used directly to reason about the variance of the particles. Instead, a scalar value is preferred. We will use the determinant of the matrix Σ in our simulation.

The simulation results presented in Table 4.3 show that the performance is almost similar with the particle filter variance and with the KL divergence and that both allow the shape to be explored with fewer actions than with a random strategy.

To conclude about this simulation, information-theoretic tools contribute significantly to tactile perception for shape reconstruction and localization. The information gain criteria must be formulated in such a way to properly describe the objective of the active perception problem. In this simulation, we used the variance of the GPIS model to describe the uncertainty (and simultaneously the gain of information) about the object shape. And we evaluated two criteria for the object pose uncertainty, namely the variance of the particle filter and the KL divergence. The variance of the particle filter was simplified to the case of a Gaussian distribution. This simplification was made in order to consider a local uncertainty in the pose of the object, which can be represented by a single Gaussian distribution. However, for more challenging situations where the uncertainty in the pose becomes large,

the variance must be considered for multi-Gaussian distributions. Thus, the results shown in this simulation open new perspectives to extend the problem to the exploration of 3D shapes in case of high uncertainty on the object pose.

4.6 CONCLUSION

This chapter has been devoted to the study of the problem of action selection for object reconstruction and localization. Active exploration is mainly based on information-theoretic techniques that describe the uncertainty of the estimation and the information gain from a specific exploration action.

The main contribution of this chapter is the development of a framework for autonomous action selection for simultaneous reconstruction and localization of objects. Most existing works focus on a single objective of action selection (either shape recovery, object recognition, or object localization). To develop this framework, it was necessary to manage the priority between the desired information to obtain in order to ensure an accurate and optimal reconstruction. We presented a simulation framework for simultaneous shape and pose estimation in 2D, with autonomous and active action selection. This can be extended in future work to 3D active reconstruction in case of high pose uncertainty and adapted to other robotic platforms with different kinematic structures.

5

CONCLUSION AND FUTURE WORK

In this thesis, we were able to build a framework for building models of objects using limited contact information collected by tactile sensors installed on the fingertips of a robotic hand. We considered the challenging situation where the pose of the object is not fixed during its exploration. This requires keeping track of the object pose to ensure a correct reconstruction. A particle filter was designed for this purpose which ensured adequate recursive tracking. For more challenging situations in which recursive tracking was not appropriate, a pose graph optimization method was applied. Furthermore, we proposed algorithms for autonomously exploring unknown objects by simultaneously considering information gained on the shape and pose of the object.

Although the proposed framework presented a solution to the problem of simultaneous reconstruction and localization of objects using tactile sensing, improvements are needed to enable large-scale application in real-life situations and under various conditions. In this chapter, we discuss the limitations of the proposed frameworks and propose some improvements for future work.

5.1 COMPARISON WITH RELATED WORK

We introduce the most relevant frameworks that deal with the same problem studied in this thesis and make a qualitative comparison to our proposed framework. The concerned frameworks are outlined in Table 5.1. In all these works, the reconstruction of the objects is made incrementally without losing contact with the object. The proposed experimental designs ensure low uncertainty in the object pose during the exploration. So, the pose tracking during the exploration is relatively less challenging, and the final reconstruction is more accurate. However, in a real-life situation, the continuity of exploration is difficult to achieve due to several uncertainties. The successive grasps method proposed in our framework enables us to explore objects in a more efficient mode and best deal with unpredictable movements of the object during the reconstruction process. Regarding tactile information availability, the design proposed in [80] provides rich and relevant perceptive information, allowing the reconstruction of objects in less time than the other designs, and ensuring a precise tracking of the object's pose.

Paper	Shape model	Exploration method	pose estimation method
[81]	GPIS	Contour following	Pose graph optimization
[82]	Polygonal	Rolling on a flat surface	Parameterized Particle Filter
[80]	GPIS	Rolling between two fingers	ICP + Pose graph optimization

Table 5.1: Recent works on tactile SLAM

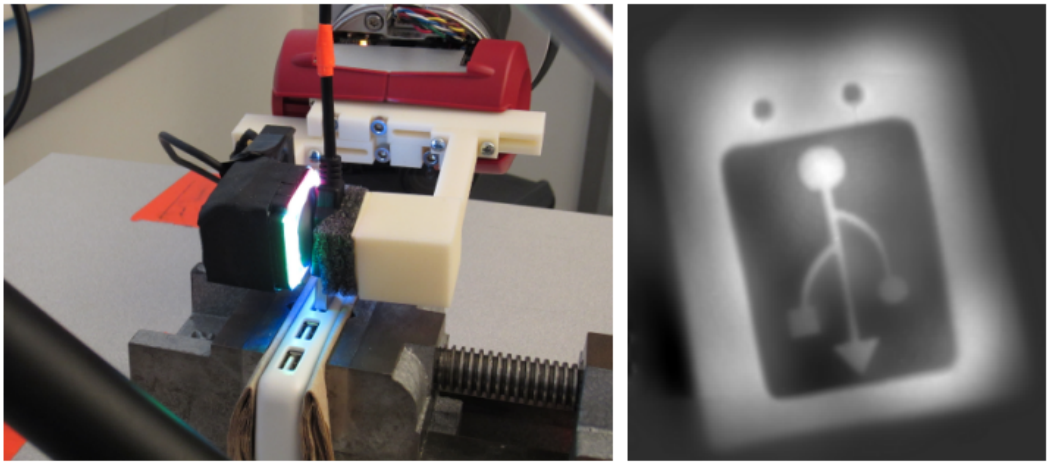


Figure 5.1: Image taken form [85], it illustrates the precision of the tactile information detected by the gelSight sensor. Left: gelSight sensor installed on a robotic gripper while grasping a USB connector. Right: Tactile image of the USB connector detected by the sensor.

5.2 LIMITATIONS AND POTENTIAL IMPROVEMENTS

The proposed framework is built from a succession of processing blocks concerning shape modeling, pose tracking, and global pose correction. Improvements may be brought to each of these blocks; they are presented in this section.

5.2.1 *Shape modeling accuracy*

The problem of tactile shape reconstruction depends primarily on the quantity and quality of the tactile data. A large quantity of tactile information can be obtained using multiple sensors on the surface of the robotic hand. In this case, the sensors must be adapted to the hand structure.

In our work, we have demonstrated the efficiency of our method using objects of simple shapes made of facets. Our experimental setup (with Shadow hand and ATI nano17 sensors) is not adapted to the reconstruction of other types of geometries (such as objects with curvatures) due to the sparsity of data that can be collected from one grasp, which produces an inaccurate partial reconstruction and then imprecise scan matching between two successive grasps.

For this purpose, advanced tactile sensors should be used to obtain more accurate geometric features for a variety of geometries, such as gelSight sensors [83], and DenseTac [84].

These types of sensors can capture different geometric features such as curvatures, normals [86], edges, and hardness [87] with high accuracy.

5.2.2 *Object tracking*

To track the pose of the object during the reconstruction, the particle filter seems to be the most appropriate method to track the motion of the objects. However, some improvements can be made to enhance its efficiency and ensure the pose estimation with high accuracy and low complexity. In our work, the implemented particle filter uses the motion model of the object to sample particles. This is the simplest method to sample particles, but it is not optimal, especially with the touch modality. The contact data are discriminative by nature [88] (either contact is present or not). This characteristic leads us to think about a more appropriate method to generate hypotheses for the particle filter. The optimal way to sample is from the measurement acquisition, which allows sampling hypotheses that correspond to the state of contact and the posture of the robotic manipulator. The manifold particle filter (MPF) seems to be the most appropriate in this case [39] [89].

MPF generates particles based on measurements obtained and the current state of the robotic hand and avoids inconsistent particles that may represent the object in a state of collision with the hand (or penetrating the hand), or particles that represent a pose of the object which is far from the measurements.

5.2.3 *Estimate other characteristics of the objects*

Our work was mainly focused on the shape and pose characteristics of objects. Other object characteristics, such as the mass and friction parameters, are also important in a manipulation task. These characteristics are essential to improve the pose prediction accuracy and to manipulate the object efficiently. The parameters related to the mass and the friction can be integrated into the particle filter like in [90] and [91], or additional exploration actions can be planned to estimate them as in [92] for the estimation of the center of mass, and [93] for the estimation of other features.

5.2.4 *Pose graph*

Another potential improvement concerns the handling of confusing situations in pose estimation. These situations of lack of observability and ambiguities in the estimation of the pose can be detected but not managed in our framework. And the pose graph is built with simple similarities between the measurements. An advanced solution for pose graph optimization can be considered in order to manage situations where many solutions are possible for the relative pose between two grasps. The solutions proposed in [94] and [95] can be adapted to our framework to control the construction and optimization of the pose graph.

5.3 SUMMARY

Manipulating unknown objects remains a major challenge for robotic manipulators. While some recent research focuses on the problem of how to grasp and manipulate unknown objects using advanced machine learning techniques, learning the characteristics of objects is more valuable to allow the robots to perform more accurate tasks and transfer this knowledge about the objects to other robotic platforms.

In this thesis, we were mainly interested in the recovery of the geometry of objects and have considered the pose and the shape simultaneously. This was a relevant goal compared to the previous works that consider shape reconstruction and pose estimation problems separately.

Advances in tactile and force sensors technology have opened new areas of research in the field of robotic manipulation and allowed to solve more challenging manipulation tasks and to investigate new applications for robotic manipulators.

A

APPENDIX 1

We present here the algorithm for clustering the contacts on an object to distinguish between faces. The algorithm receives contact information (points and normals). It proceeds in two steps. The first step clusters the contacts according to the normals. This makes all the faces with the same normals in the same cluster. The second step creates subclusters from the already existing clusters in a way that each cluster represents a single face of the object. This is done using the Dijkstra algorithm, which computes a path from a start point to a goal point in the space of all contact points, and if this path contains a point which does not belong to the cluster, this means that there is another face of the objects standing between the two contacts, then it is necessary to separate them.

Algorithm A.1 clustering_contacts(points,normals)

1: Step 1: cluster contacts according to their normals
2: affect $contact_1$ to cluster C_1 ▷ Initialization
3: **for** each $contact_i$ **do**
4: compute the distance of $contact_i$ to each cluster C_j
5: **if** there is n_* such as $n_i \dot{n}_* > \epsilon_n$ **then**
6: affect n_i to C_i
7: **else**
8: Create new cluster and affect n_i to it
9: **end if**
10: **end for**
11: Step 2: cluster contacts according to their coordinates
12: **for** each cluster C_j **do**
13: **for** each pair of points p_i and p_k of the same cluster C_j **do**
14: path = `Dijkstra`(p_i,p_k)
15: **if** path contains a point p_* which is out of C_j **then**
16: create a new cluster from C_j
17: **end if**
18: **end for**
19: **end for**

B

APPENDIX 2

Here we illustrate the algorithm to divide the shape parts into clusters to explore. Algorithm B.1 contains the whole clustering instructions. Two sub-functions of the main algorithm are presented in Algorithm B.2 and Algorithm B.3.

In lines 4 and 5 of Algorithm B.1, we compute the variance of the sub-clusters, and the variance of the examined cluster, in order to decide if the subdivision is necessary. This is inspired by ANOVA (ANalysis Of VAriance) method [96].
The subdivision of the cluster is performed using the function $oneStepClustering()$ explained in Algorithm B.2.

Algorithm B.1 clustering(points,normals)

1: **for** $i = 1 : N_{clusters}$ **do**
2: $examinedCluster = clusters[i]$
3: $SubClusters = OneStepClustering(examinedCluster)$
4: $Var_{groups} = VarianceBetweenGroups(subClusters)$
5: $Var_{total} = Variance(examinedCluster)$
6: **if** $Var_{groups} > Var_{total}$ **then**
7: Subdivide clusters
8: **end if**
9: $subClusters = organizeClusters$
10: Add $subClusters$ to $clusters$
11: **end for**

Algorithm B.2 OneStepClustering(points,normals)

1: $d_1 = D_{euclidian}$
2: $d_2 = D_{normals}$
3: $d_3 = D_{height}$
4: $D = 0.55d_1 + 0.3d_2 + 0.15d_3$
5: $C = \texttt{linkage}(D)$
6: $clusters = \texttt{cluster}(C)$

Functions `linkage()` and `cluster()` are inherent function in MATLAB. `linkage()` constructs and agglomirative hierarchical tree. And the function `cluster()` constructs clusters from the obtained linkage.

Algorithm B.3 OrganizeClusters(points,normals)

```
1: Clusters_size = Compute the size of each cluster
2: min_cluster_size = quantile(clusters_size, 0.9)
3: for each cluster C_i do
4:     if size(C_i) < min_cluster_size then
5:         Add C_i to cluster_to_save
6:     else
7:         Add C_i to cluster_to_delete
8:     end if
9: end for
10: for each cluster_to_delete do
11:     compute distance to cluster_to_save
12:     affect cluster_to_delete to the nearestcluster_to_save
13: end for
```

C

APPENDIX 3

Hausdorff distance is used in this thesis to measure the error between the reconstructed shape of an object and its actual shape. It represents the maximum of all minimum distances between the points of the two shapes. Figure C.1 shows how the Hausdorff distance is computed between the two shapes A and B (in blue and red, respectively). Hausdorff distance is defined as follows:

$$d_H = \max(d_{AB}, d_{BA}) \tag{C.1}$$

with

$$d_{AB} = \max_{a \in A} \min_{b \in B} d(a, b) \tag{C.2}$$

and

$$d_{BA} = \max_{b \in B} \min_{a \in A} d(a, b) \tag{C.3}$$

$d(a, b)$ is the euclidean distance between points a and b.

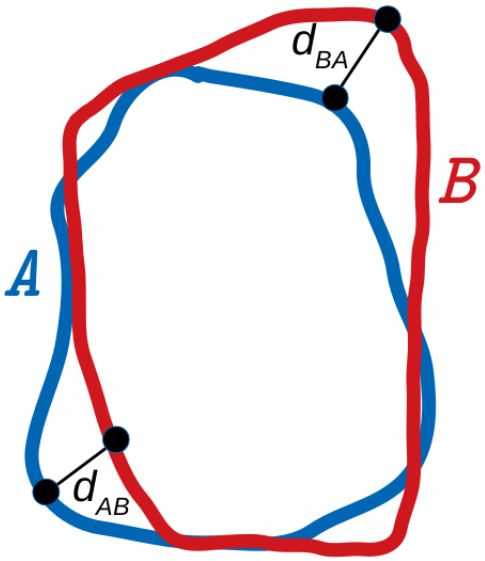


Figure C.1: Hausdorff distance computation between the shape A (in blue) and the shape B (in red).

BIBLIOGRAPHY

[1] Z. Kappassov, J.-A. Corrales and V. Perdereau, 'Tactile sensing in dexterous robot hands', *Robotics and Autonomous Systems*, vol. 74, pp. 195–220, 2015.

[2] C. Chi, X. Sun, N. Xue, T. Li and C. Liu, 'Recent progress in technologies for tactile sensors', *Sensors*, vol. 18, no. 4, p. 948, 2018.

[3] H. Liu, K. C. Nguyen, V. Perdereau, J. Bimbo Joao kelestemur2022tactileand Back, M. Godden, L. D. Seneviratne and K. Althoefer, 'Finger contact sensing and the application in dexterous hand manipulation', *Autonomous Robots*, vol. 39, no. 1, pp. 25–41, 2015.

[4] G. E. Chia-Hsien Lin Loeb, 'Estimating point of contact, force and torque in a biomimetic tactile sensor with deformable skin', 2013.

[5] C. Fox, M. Evans, M. Pearson and T. Prescott, 'Tactile slam with a biomimetic whiskered robot', in *2012 IEEE International Conference on Robotics and Automation*, IEEE, 2012, pp. 4925–4930.

[6] C. Xiao, S. Xu, W. Wu and J. Wachs, 'Active multiobject exploration and recognition via tactile whiskers', *IEEE Transactions on Robotics*, 2022.

[7] O. Struckmeier, K. Tiwari, M. Salman, M. J. Pearson and V. Kyrki, 'Vita-slam: A bio-inspired visuo-tactile slam for navigation while interacting with aliased environments', in *2019 IEEE International Conference on Cyborg and Bionic Systems (CBS)*, IEEE, 2019, pp. 97–103.

[8] Z. Kappassov, 'Touch driven dexterous robot arm control', PhD thesis, Paris 6, 2017.

[9] N. Sommer and A. Billard, 'Multi-contact haptic exploration and grasping with tactile sensors', *Robotics and autonomous systems*, vol. 85, pp. 48–61, 2016.

[10] A. M. Okamura, 'Haptic exploration of unknown objects', 2000.

[11] N. F. Lepora and J. Lloyd, 'Pose-based tactile servoing: Controlled soft touch using deep learning', *IEEE Robotics & Automation Magazine*, vol. 28, no. 4, pp. 43–55, 2021.

[12] P. K. Allen and P. Michelman, 'Acquisition and interpretation of 3-d sensor data from touch', 1990.

[13] R. Ibrayev and Y.-B. Jia, 'Surface recognition by registering data curves from touch', in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2006, pp. 55–60.

[14] M. Kaboli, R. Walker, G. Cheng *et al.*, 'In-hand object recognition via texture properties with robotic hands, artificial skin, and novel tactile descriptors', in *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*, IEEE, 2015, pp. 1155–1160.

[15] S. Luo, W. Mou, K. Althoefer and H. Liu, 'Iclap: Shape recognition by combining proprioception and touch sensing', *Autonomous Robots*, vol. 43, no. 4, pp. 993–1004, 2019.

[16] S. Pohtongkam and J. Srinonchat, 'Tactile object recognition for humanoid robots using new designed piezoresistive tactile sensor and dcnn', *Sensors*, vol. 21, no. 18, p. 6024, 2021.

[17] F. Von Drigalski, S. Taniguchi, R. Lee, T. Matsubara, M. Hamaya, K. Tanaka and Y. Ijiri, 'Contact-based in-hand pose estimation using bayesian state estimation and particle filtering', in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2020, pp. 7294–7299.

[18] J. M. Bimbo, 'Touch based object pose estimation for robotic grasping', PhD thesis, King's College London, 2016.

[19] P. K. Allen and K. S. Roberts, 'Haptic object recognition using a multi-fingered dextrous hand', in *Proceedings, 1989 International Conference on Robotics and Automation*, 1989, 342–347 vol.1. DOI: 10.1109/ROBOT.1989.100011.

[20] Barr, 'Superquadrics and angle-preserving transformations', *IEEE Computer Graphics and Applications*, vol. 1, no. 1, pp. 11–23, 1981, ISSN: 1558-1756. DOI: 10.1109/MCG.1981.1673799.

[21] A. Bierbaum, I. Gubarev and R. Dillmann, 'Robust shape recovery for sparse contact location and normal data from haptic exploration', in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2008, pp. 3200–3205. DOI: 10.1109/IROS.2008.4650982.

[22] G. Vezzani, U. Pattacini and L. Natale, 'A grasping approach based on superquadric models', in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2017, pp. 1579–1586.

[23] A. Makhal, F. Thomas and A. P. Gracia, 'Grasping unknown objects in clutter by superquadric representation', in *2018 Second IEEE International Conference on Robotic Computing (IRC)*, IEEE, 2018, pp. 292–299.

[24] R. Fontes Portal, J. Pereira Dias and L. Gonçalves de Sousa, 'Contact detection between convex superquadric surfaces', *Archive of Mechanical Engineering*, vol. 57, no. 2, pp. 165–186, 2010.

[25] T. Nagata, 'Simple local interpolation of surfaces using normal vectors', *Computer Aided Geometric Design*, vol. 22, no. 4, pp. 327 –347, 2005, ISSN: 0167-8396. DOI: https://doi.org/10.1016/j.cagd.2005.01.004. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S016783960500021X.

[26] D. Neto, M. C. Oliveira, L. F. Menezes and J. L. Alves, 'Improving nagata patch interpolation applied for tool surface description in sheet metal forming simulation', *Computer-Aided Design*, vol. 45, no. 3, pp. 639–656, 2013.

[27] A. Bierbaum, K. Welke, D. Burger, T. Asfour and R. Dillmann, 'Haptic exploration for 3d shape reconstruction using five-finger hands', in *2007 7th IEEE-RAS International Conference on Humanoid Robots*, 2007, pp. 616–621. DOI: 10.1109/ICHR.2007.4813935.

[28] N. Sommer, M. Li and A. Billard, 'Bimanual compliant tactile exploration for grasping unknown objects', in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2014, pp. 6400–6407.

[29] D. R. Faria, R. Martins, J. Lobo and J. Dias, 'Probabilistic representation of 3d object shape by in-hand exploration', in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2010, pp. 1560–1565.

[30] O. Williams and A. Fitzgibbon, 'Gaussian process implicit surfaces', in *Gaussian Processes in Practice*, 2006.

[31] W. Martens, Y. Poffet, P. R. Soria, R. Fitch and S. Sukkarieh, 'Geometric priors for gaussian process implicit surfaces', *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 373–380, 2017, ISSN: 2377-3774. DOI: 10.1109/LRA.2016.2631260.

[32] S. Ottenhaus, M. Miller, D. Schiebener, N. Vahrenkamp and T. Asfour, 'Local implicit surface estimation for haptic exploration', in *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, 2016, pp. 850–856. DOI: 10.1109/HUMANOIDS.2016.7803372.

[33] W. Martens, Y. Poffet, P. R. Soria, R. Fitch and S. Sukkarieh, 'Geometric priors for gaussian process implicit surfaces', *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 373–380, 2016.

[34] J.-P. A. Ivan, T. Stoyanov and J. A. Stork, 'Online distance field priors for gaussian process implicit surfaces', *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 8996–9003, 2022.

[35] P. C. Gaston and T. Lozano-Perez, 'Tactile recognition and localization using object models: The case of polyhedra on a plane', *IEEE transactions on pattern analysis and machine intelligence*, no. 3, pp. 257–266, 1984.

[36] M. Pfanne, M. Chalon, F. Stulp and A. Albu-Schäffer, 'Fusing joint measurements and visual features for in-hand object pose estimation', *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3497–3504, 2018.

[37] C. Corcoran and R. Platt, 'A measurement model for tracking hand-object state during dexterous manipulation', in *2010 IEEE International Conference on Robotics and Automation*, IEEE, 2010, pp. 4302–4308.

[38] A. Petrovskaya and O. Khatib, 'Global localization of objects via touch', *IEEE Transactions on Robotics*, vol. 27, no. 3, pp. 569–585, 2011.

[39] M. C. Koval, M. Klingensmith, S. S. Srinivasa, N. S. Pollard and M. Kaess, 'The manifold particle filter for state estimation on high-dimensional implicit manifolds', in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2017, pp. 4673–4680.

[40] J. Bimbo, P. Kormushev, K. Althoefer and H. Liu, 'Global estimation of an object's pose using tactile sensing', *Advanced Robotics*, vol. 29, no. 5, pp. 363–374, 2015.

[41] H. Nguyen and Q.-C. Pham, 'Touch-based object localization in cluttered environments', *arXiv preprint arXiv:1709.09317*, 2017.

[42] Y. Ding, J. Bonse, R. Andre and U. Thomas, 'In-hand grasping pose estimation using particle filters in combination with haptic rendering models', *International Journal of Humanoid Robotics*, vol. 15, no. 01, p. 1 850 002, 2018.

[43] A. Petrovskaya, O. Khatib, S. Thrun and A. Y. Ng, 'Bayesian estimation for autonomous object manipulation based on tactile sensors', in *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, IEEE, 2006, pp. 707–714.

[44] G. Vezzani, U. Pattacini, G. Battistelli, L. Chisci and L. Natale, 'Memory unscented particle filter for 6-dof tactile localization', *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1139–1155, 2017.

[45] M. Chalon, J. Reinecke and M. Pfanne, 'Online in-hand object localization', in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2013, pp. 2977–2984.

[46] M. Kaess, A. Ranganathan and F. Dellaert, 'Isam: Incremental smoothing and mapping', *IEEE Transactions on Robotics*, vol. 24, no. 6, pp. 1365–1378, 2008.

[47] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. Leonard and F. Dellaert, 'Isam2: Incremental smoothing and mapping with fluid relinearization and incremental variable reordering', in *2011 IEEE International Conference on Robotics and Automation*, IEEE, 2011, pp. 3281–3288.

[48] F. Dellaert and M. Kaess, 'Square root sam: Simultaneous localization and mapping via square root information smoothing', *The International Journal of Robotics Research*, vol. 25, no. 12, pp. 1181–1203, 2006.

[49] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige and W. Burgard, 'G 2 o: A general framework for graph optimization', in *2011 IEEE International Conference on Robotics and Automation*, IEEE, 2011, pp. 3607–3613.

[50] K. Konolige, G. Grisetti, R. Kümmerle, W. Burgard, B. Limketkai and R. Vincent, 'Efficient sparse pose adjustment for 2d mapping', in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2010, pp. 22–29.

[51] P. Sodhi, M. Kaess, M. Mukadam and S. Anderson, 'Learning tactile models for factor graph-based estimation', in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2021, pp. 13 686–13 692.

[52] T. Kelestemur, R. Platt and T. Padir, 'Tactile pose estimation and policy learning for unknown object manipulation', *arXiv preprint arXiv:2203.10685*, 2022.

[53] M. Barczyk, S. Bonnabel and F. Goulette, *Observability, covariance and uncertainty of icp scan matching*, 2014.

[54] E. B. Olson, 'Real-time correlative scan matching', in *2009 IEEE International Conference on Robotics and Automation*, IEEE, 2009, pp. 4387–4393.

[55] D. H. Douglas and T. K. Peucker, 'Algorithms for the reduction of the number of points required to represent a digitized line or its caricature', *Cartographica: the international journal for geographic information and geovisualization*, vol. 10, no. 2, pp. 112–122, 1973.

[56] H. Q. Luong, M. Vlaminck, W. Goeman and W. Philips, 'Consistent icp for the registration of sparse and inhomogeneous point clouds', in *2016 IEEE Sixth International Conference on Communications and Electronics (ICCE)*, IEEE, 2016, pp. 262–267.

[57] S. Rusinkiewicz, 'A symmetric objective function for icp', *ACM Transactions on Graphics (TOG)*, vol. 38, no. 4, pp. 1–7, 2019.

[58] A. Bierbaum, M. Rambow, T. Asfour and R. Dillmann, 'A potential field approach to dexterous tactile exploration of unknown objects', in *Humanoids 2008-8th IEEE-RAS International Conference on Humanoid Robots*, IEEE, 2008, pp. 360–366.

[59] D. Driess, P. Englert and M. Toussaint, 'Active learning with query paths for tactile object shape exploration', in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2017, pp. 65–72.

[60] Z. Yi, R. Calandra, F. Veiga, H. van Hoof, T. Hermans, Y. Zhang and J. Peters, 'Active tactile object exploration with gaussian processes', in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2016, pp. 4925–4930.

[61] D. Driess, D. Hennes and M. Toussaint, 'Active multi-contact continuous tactile exploration with gaussian process differential entropy', in *2019 International Conference on Robotics and Automation (ICRA)*, IEEE, 2019, pp. 7844–7850.

[62] S. Ottenhaus, L. Kaul, N. Vahrenkamp and T. Asfour, 'Active tactile exploration based on cost-aware information gain maximization', *International Journal of Humanoid Robotics*, vol. 15, no. 01, p. 1 850 015, 2018.

[63] L. Rustler, J. Lundell, J. K. Behrens, V. Kyrki and M. Hoffmann, 'Active visuo-haptic object shape completion', *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 5254–5261, 2022.

[64] T. Matsubara and K. Shibata, 'Active tactile exploration with uncertainty and travel cost for fast shape estimation of unknown objects', *Robotics and Autonomous Systems*, vol. 91, pp. 314–326, 2017.

[65] C. Rosales, F. Spinelli, M. Gabiccini, C. Zito and J. L. Wyatt, 'Gpatlasrrt: A local tactile exploration planner for recovering the shape of novel objects', *International Journal of Humanoid Robotics*, vol. 15, no. 01, p. 1 850 014, 2018.

[66] J. Xu, S. Song and M. Ciocarlie, 'Tandem: Learning joint exploration and decision making with tactile sensors', *arXiv preprint arXiv:2203.00798*, 2022.

[67] D. Xu, G. E. Loeb and J. A. Fishel, 'Tactile identification of objects using bayesian exploration', in *2013 IEEE International Conference on Robotics and Automation*, IEEE, 2013, pp. 3056–3061.

[68] P. Hebert, T. Howard, N. Hudson, J. Ma and J. W. Burdick, 'The next best touch for model-based localization', in *2013 IEEE International Conference on Robotics and Automation*, IEEE, 2013, pp. 99–106.

[69] B. Saund, S. Chen and R. Simmons, 'Touch based localization of parts for high precision manufacturing', in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2017, pp. 378–385.

[70] Y. Taguchi, T. K. Marks and J. R. Hershey, 'Entropy-based motion selection for touch-based registration using rao-blackwellized particle filtering', in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2011, pp. 4690–4697.

[71] S. Chen, B. Saund and R. Simmons, 'The datum particle filter: Localization for objects with coupled geometric datums', in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2017, pp. 3325–3332.

[72] N. Tosi, O. David and H. Bruyninckx, 'Action selection for touch-based localisation trading off information gain and execution time', in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2014, pp. 2270–2275.

[73] P. K. Murali, M. Gentner and M. Kaboli, 'Active visuo-tactile point cloud registration for accurate pose estimation of objects in an unknown workspace', in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2021, pp. 2838–2844.

[74] P. K. Murali, R. Dahiya and M. Kaboli, 'An empirical evaluation of various information gain criteria for active tactile action selection for pose estimation', in *2022 IEEE International Conference on Flexible and Printable Sensors and Systems (FLEPS)*, IEEE, 2022, pp. 1–4.

[75] R. L. Klatzky and S. Lederman, 'Intelligent exploration by the human hand', in *Dextrous robot hands*, Springer, 1990, pp. 66–81.

[76] C. Strub, F. Wörgötter, H. Ritter and Y. Sandamirskaya, 'Using haptics to extract object shape from rotational manipulations', in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2014, pp. 2179–2186.

[77] A. Bicchi, A. Marigo and D. Prattichizzo, 'Dexterity through rolling: Manipulation of unknown objects', in *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No. 99CH36288C)*, IEEE, vol. 2, 1999, pp. 1583–1588.

[78] S. Ottenhaus, L. Kaul, N. Vahrenkamp and T. Asfour, 'Active tactile exploration based on cost-aware information gain maximization', *I. J. Humanoid Robotics*, vol. 15, 1850015:1–1850015:21, 2018.

[79] K. Hang, M. Li, J. A. Stork, Y. Bekiroglu, F. T. Pokorny, A. Billard and D. Kragic, 'Hierarchical fingertip space: A unified framework for grasp planning and in-hand grasp adaptation', *IEEE Transactions on robotics*, vol. 32, no. 4, pp. 960–972, 2016.

[80] C. Pan, M. Lepert, S. Yuan, R. Antonova and J. Bohg, 'Task-driven in-hand manipulation of unknown objects with tactile sensing', *arXiv preprint arXiv:2210.13403*, 2022.

[81] S. Suresh, M. Bauza, K.-T. Yu, J. G. Mangelson, A. Rodriguez and M. Kaess, 'Tactile slam: Real-time inference of shape and pose from planar pushing', in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2021, pp. 11 322–11 328.

[82] B. Liang, W. Liang and Y. Wu, 'Parameterized particle filtering for tactile-based simultaneous pose and shape estimation', *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 1270–1277, 2021.

[83] W. Yuan, S. Dong and E. H. Adelson, 'Gelsight: High-resolution robot tactile sensors for estimating geometry and force', *Sensors*, vol. 17, no. 12, p. 2762, 2017.

[84] W. K. Do and M. Kennedy, 'Densetact: Optical tactile sensor for dense shape reconstruction', in *2022 International Conference on Robotics and Automation (ICRA)*, IEEE, 2022, pp. 6188–6194.

[85] R. Li, R. Platt, W. Yuan, A. ten Pas, N. Roscup, M. A. Srinivasan and E. Adelson, 'Localization and manipulation of small parts using gelsight tactile sensing', in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2014, pp. 3988–3993.

[86] J. Li, S. Dong and E. H. Adelson, 'End-to-end pixelwise surface normal estimation with convolutional neural networks and shape reconstruction using gelsight sensor', in *2018 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, IEEE, 2018, pp. 1292–1297.

[87] W. Yuan, C. Zhu, A. Owens, M. A. Srinivasan and E. H. Adelson, 'Shape-independent hardness estimation using deep learning and a gelsight tactile sensor', in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2017, pp. 951–958.

[88] M. Klingensmith, M. C. Koval, S. S. Srinivasa, N. S. Pollard and M. Kaess, 'The manifold particle filter for state estimation on high-dimensional implicit manifolds', *arXiv preprint arXiv:1604.07224*, 2016.

[89] A. Gonczarek and J. M. Tomczak, 'Articulated tracking with manifold regularized particle filter', *Machine Vision and Applications*, vol. 27, no. 2, pp. 275–286, 2016.

[90] L Zhang and J. Trinkle, 'Exploring the application of particle filters to grasping acquisition with visual and tactile occlusion', Technical Report 10-09, Department of Computer Science, Rensselaer ..., Tech. Rep., 2010.

[91] L. Zhang and J. C. Trinkle, 'The application of particle filtering to grasping acquisition with visual occlusion and tactile sensing', in *2012 IEEE International Conference on Robotics and Automation*, IEEE, 2012, pp. 3805–3812.

[92] K. Yao, M. Kaboli and G. Cheng, 'Tactile-based object center of mass exploration and discrimination', in *2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids)*, IEEE, 2017, pp. 876–881.

[93] M. Kaboli, K. Yao, D. Feng and G. Cheng, 'Tactile-based active object discrimination and target object search in an unknown workspace', *Autonomous Robots*, vol. 43, no. 1, pp. 123–152, 2019.

[94] S. Tully, G. Kantor, H. Choset and F. Werner, 'A multi-hypothesis topological slam approach for loop closing on edge-ordered graphs', in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2009, pp. 4943–4948.

[95] M. Hsiao and M. Kaess, 'Mh-isam2: Multi-hypothesis isam using bayes tree and hypo-tree', in *2019 International Conference on Robotics and Automation (ICRA)*, IEEE, 2019, pp. 1274–1280.

[96] H.-Y. Kim, 'Analysis of variance (anova) comparing means of more than two groups', *Restorative dentistry & endodontics*, vol. 39, no. 1, pp. 74–77, 2014.

COLOPHON

This document was authored using TeXstudio[1] and typeset based on the La Trobe PhD Thesis Template[2] (customization of the classicthesis[3] LaTeX template) using TeX Live[4] on a machine running LinuxMint[5]. Other notable software tools used during the production of this document: GNU Octave[6] and LibreOffice[7] for number crunching; GIMP[8] and Inkscape[9] for working with graphics; Mendeley Desktop[10] for reference management.

1 https://www.texstudio.org
2 https://github.com/bashimao/ltu-thesis
3 https://bitbucket.org/amiede/classicthesis
4 https://tug.org/texlive
5 https://linuxmint.com
6 https://www.gnu.org/software/octave
7 https://www.libreoffice.org
8 https://www.gimp.org
9 https://inkscape.org
10 https://www.mendeley.com