Enhanced Learning Strategies for Tactile Shape Estimation and Grasp Planning of Unknown Objects

by

Shiyi Yang

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Mechanical and Mechatronics Engineering

Waterloo, Ontario, Canada, 2019

Examining Committee Membership

The following served on the Examining Committee for this thesis. The decision of the Examining Committee is by majority vote.

External Examiner:	Dr. Xiang Chen Professor Dept. of Electrical and Computer Engineering University of Windsor
Supervisor:	Dr. Soo Jeon Associate Professor Dept. of Mechanical & Mechatronics Engineering University of Waterloo
Internal Member:	Dr. William Melek Professor Dept. of Mechanical & Mechatronics Engineering University of Waterloo
Internal Member:	Dr. Hyock Ju Kwon Professor Dept. of Mechanical & Mechatronics Engineering University of Waterloo
Internal-External:	Dr. Nasser Lashgarian Azad Associate Professor Dept. of Systems Design Engineering University of Waterloo

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Grasping is one of the key capabilities for a robot operating and interacting with humans in a real environment. The conventional approaches require accurate information on both object shape and robotic system modeling. The performance, therefore, can be easily influenced by any noise sensor data or modeling errors. Moreover, identifying the shape of an unknown object under some vision-denied conditions is still a challenging problem in the robotics field. To address this issue, this thesis investigates the estimation of unknown object shape using tactile exploration and the task-oriented grasp planning for a novel object using enhanced learning techniques.

In order to rapidly estimate the shape of an unknown object, this thesis presents a novel multi-fidelity-based optimal sampling method which attempts to improve the existing shape estimation via tactile exploration. Gaussian process regression is used for implicit surface modeling with sequential sampling strategy. The main objective is to make the process of sample point selection more efficient and systematic such that the unknown shape can be estimated fast and accurately with highly limited sample points (e.g., less than 1% of number of data set for the true shape). Specifically, we propose to select the next best sample point based on two optimization criteria: 1) the mutual information (MI) for uncertainty reduction, and 2) the local curvature for fidelity enhancement. The combination of these two objectives leads to an optimal sampling process that balances between the exploration of the whole shape and the exploitation of the local area where the higher fidelity (or more sampling) is required. Simulation and experimental results successfully demonstrate the advantage of the proposed method in terms of estimation speed and accuracy over the conventional one, which allows us to reconstruct recognizable 3D shapes using only around optimally selected 0.4% of the original data set.

With the available object shape, this thesis also introduces a knowledge-based approach to quickly generate a task-oriented grasp for a novel object. A comprehensive training dataset which consists of specific tasks and geometrical and physical knowledge of grasping is built up from physical experiment. To analyze and efficiently utilize the training data, a multi-step clustering algorithm is developed based on a self-organizing map. A number of representative grasps are then selected from the entire training dataset and used to generate a suitable grasp for a novel object. The number of representative grasps is automatically determined using the proposed auto-growing method. In addition, to improve the accuracy and efficiency of the proposed clustering algorithm, we also develop a novel method to localize the initial centroids while capturing the outliers. The results of simulation illustrate that the proposed initialization method and the auto-growing method outperform some conventional approaches in terms of accuracy and efficiency. Furthermore, the proposed

knowledge-based grasp planning is also validated on a real robot. The results demonstrate the effectiveness of this approach to generate task-oriented grasps for novel objects.

Acknowledgements

I would like to thank my supervisor, Associate Professor Soo Jeon for giving me the unique opportunity to pursue PhD research in my dream field, robotics. His immense guidance and professional supervision was paramount in helping me to cultivate the knowledge over the last four years.

Throughout my PhD, I have benefited from collaborating with other researchers. In particular, my work in the area of object shape estimation was conducted in collaboration with Associate Professor Jongeun Choi from the Yonsei University. His technical insight allowed me to look at my research from a different perspective. I also would like to thank Professor William Melek who contributes ideas to Chapter 4 of this thesis.

Finally, I would like to immensely thank my loving wife, for her hard work, dedication and limitless support, and my parents, whose sacrifice, gave me the opportunity to pursue my goals.

Dedication

This is dedicated to my wife and my family,

Xin, Mom, Dad and my upcoming baby.

Table of Contents

List of Tables

List of Figures

Nomenclature

Υ	the region in which the target object locates
X	a finite subset of Υ contains all possible sample points, x
X_o	a finite subset of X contains all surface points, x_o
$\widehat{X}_o$	a finite subset contains all points on the estimated surface, $\hat{x}_o$
X_{s_n}	a finite subset of X_o contains s_n selected sample points
$\overline{X}$	$\widehat{X}_o \backslash (X_{s_n} \cup x_{s_{n+1}})$
$\hat{x}_o^*$	optimal next sample point on $\widehat{X}_o$
Y, Y_o, Y_{s_n}	noisy measurements of corresponding sample points
$\mathrm{m}(\bullet), \mathrm{k}(\bullet, \bullet)$	mean and covariance function
ε	i.i.d. Gaussian noise in measurements
$\sigma_0, \sigma_\varepsilon, \sigma_l$	hyperparameters of the covariance function
$\mu(\bullet), \mathbb{V}(\bullet)$	conditional mean and variance function
$\mathrm{MI}(\bullet), \mathrm{H}(\bullet)$	mutual information and entropy
$\phi(\bullet), \Phi(\bullet)$	increased amount of mutual information and the corresponding normalized value
$\mathcal{C}$	curvature covariance matrix
λ_ℓ, v_ℓ	eigenvalues and eigenvectors of $\mathcal{C}$
$\gamma(\bullet), \Gamma(\bullet)$	estimated curvature and the corresponding normalized value
T	objective function to tradeoff the conflict relation between mutual information and curvature
$\mathrm{tr}(\bullet)$	trace function
ν	mean of the predicted variance of X
α	weighting factor to balance the effect of two different criteria

$d_{\mathrm{H}}(\bullet)$	Hausdorff distance
ψ, Ψ	training data vectors and training dataset
ω, Ω	neurons and the codebook contains a number of combinations of neurons
β	index of the winning neuron
h, σ_h	learning rate of each neuron and variation rate between two neurons
$\mathbf{k}(\bullet, \bullet)$	Gaussian kernel
Ξ	similarity function
K	total number of neurons on each layer
$\mathbb{J}$	distortion function
$1\{\bullet\}$	indicator function

Chapter 1

Introduction

Due to the flexibility and high functionality of the robotic manipulators, they have been widely used in traditional manufacturing sectors such as the assembly lines [13]. In recent years, the robotic manipulators have been making their ways into other sectors where they get to interact with humans. Examples include hospitals [14], work cells for flexible manufacturing [15] and home automation [16, 17]. In order to carry out a large variety of everyday tasks(instead of repetitive tasks on factory floors), different types of multi-fingered robotic hands have also been developed. Controlling the robotic hand for every-day objects and tasks is still a major challenge even for the best artificial robotic manipulators available today.

One of fundamental issues in operating robots in such unstructured environments lies in the fact that robots need to be able to handle "novel" objects, i.e. objects that have not been seen or preprogrammed before. This thesis attempts to address this issue so as to advance the control of robotic hands for every-day objects and tasks. In particular, this research tackles two specific problems; 1) identifying, fast and accurately, 3D shape of an unknown object using tactile data, and 2) training robots to be able to grasp novel objects based on prior experience on similar objects.

In the area of robotic manipulation, the basic information of the object as well as the surroundings can be acquired from different types of sensors. The vision cameras and laser sensors provide by far the most informative data on the object shape and pose, if available. Both of them, however, are limited by occlusions, dim lighting, hazardous environment under smoke or fire and do not often exhaustively describe the object in terms of the shape or other specifications such as the mechanical property. Figure 1.1 illustrates an example that the target object is partially (B) or completely (C) occluded by the obstacles

or the environment. This limitation can be avoided through direct contact by utilizing tactile (or haptic) sensors which can often provide more accurate local information on the object properties. Manipulation tasks associated with heavy use of tactile sensors are often referenced as the *tactile exploration*. In recent years, the tactile exploration has been getting an increased attention for advanced hand control and widely used to manipulate the objects and interact with environments [18, 19]. A number of research studies have also shown that integrating the tactile data with other types of sensor data can improve the efficiency to manipulate the objects and interact with environments [20, 21].

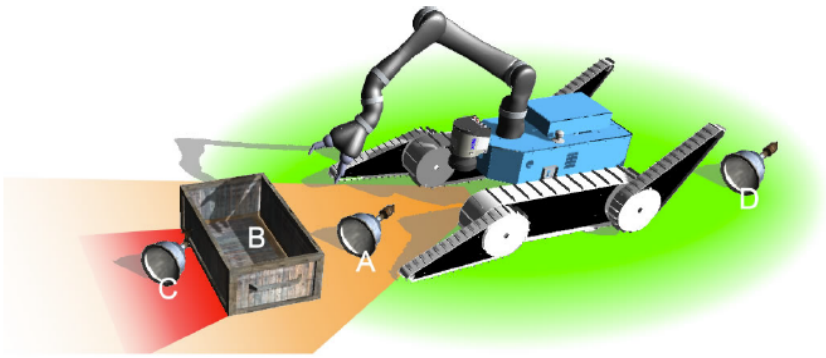


Figure 1.1: Visible and reachable objects. A and B are both visible and reachable, B is partially occluded. C and D are non-visible and reachable. [1]

An important observation obtained through an experiment, involving seven healthy human volunteers, is that the complexity of the manipulation process highly depends on target features [22]. Moreover, the results presented in [23, 24] also imply that there exist different types of relations between human grasping motion and object features. For instance, the approaching direction of the hand is determined by the object position and orientation, and the hand pose depends on the size of the object as well as the task that needs to be achieved. The results further demonstrate that the total human grasps can be categorized into a number of groups based on the target object shape and functionality. These results indicate two points which have important implications for grasp planning for robotic manipulators:

1. Grasp planning is highly relevant to the object features and the desired tasks. For instance, the hand pose of grasping a pencil is not suitable to grasp a basketball, but can be applied to grasp a small hammer. Similarly, to pour water from a mug, grasping the body from the side is apparently a good strategy instead of grasping from the top, since the hand will block the opening of the mug if grasp from the top.

2. The objects that share similarities in shape and functionality can be grasped and manipulated in a similar way. For instance, although two mugs have slightly different shapes, they can still be grasped using the same hand pose. In spite of different sizes, most of the screwdrivers can be grasped in the same way.

Given the information on the target object, the conventional approaches for finding an appropriate grasp try to determine the contact locations of each fingertip while satisfying some certain constraints, such as stability and equilibrium [25, 9]. In general, this type of approaches rely heavily on the accurate information on the object model as well as the robot kinematics and dynamics. However, it is not possible to acquire such accurate information in practice, due to the inherent errors from sensors and robotic systems. Moreover, the desired task associated with the target object is normally not considered in such approaches due to the difficulties of modeling a task. To address these technical issues, another type of approaches using data-driven techniques are developed in recent years [11]. Applying this type of approaches, a large number of grasp candidates are generated for some representative objects in simulation or on a real robot and recorded into a training dataset. When trying to grasp a novel object, the most suitable grasp candidate from the training dataset is selected based on the similarity between the target object and the training objects. In order to analyze such big dataset, some machine learning techniques have been applied to the grasp planning problem [26, 27].

1.1 Outline

The remainder of this thesis is organized as follows. Chapter 2 reviews the relevant research studies regarding the representation and estimation of the object shape as well as an overview of related research that apply machine learning techniques on the grasp planning problem. A novel multi-fidelity-based optimal sampling strategy to rapidly estimate the shape of an object via tactile exploration is proposed in Chapter 3. In Chapter 4, a task-oriented grasp planning method using a multi-step classification algorithm is developed. The concluding remarks of this thesis and several suggestions for the future work are provided in Chapter 5.

1.2 Contributions

This research aims to improve existing learning strategies applied to the problem of tactile exploration for shape estimation and that of task-oriented grasp planning. The contribu-

tions claimed in this thesis are as follows

1. Incorporating the mutual information and the geometric features of the target object into Bayesian statistics, we propose a new optimal sample points selection method to accurately estimate the shape of the target object with a limited number of sampling trials. Formulating it as a multi-objective problem, we develop a trade-off function that combines both criteria into a single objective function using a weighting parameter. By tuning the weighting factors, the objective function can adjust the priority between estimating the whole shape of the object and emphasizing on geometric features of the object. This idea has been validated through simulations as well as experiments using a robotic testbed.

2. As an enhancement to the batch learning SOM, we propose a knowledge-based approach to rapidly establish a suitable grasp for a novel object. A training set consisting of the geometric information on the training objects and a number of task-oriented grasps is first built up. Such training set is run through a multi-step clustering algorithm to generate a set of representative grasps. Applying Gaussian kernel to measure the similarity between training data, we also develop a new initialization method to locate the optimal initial neurons (centroids). In order to determine the optimal number of neurons, a auto-growing method is also developed. By considering the geometric information on a novel object, a good grasp is generated from the learned experience to accomplish a specific task for the novel object. As a result, the knowledge can automatically improve and update from grasping every novel objects. The proposed technique has been validated through experimental results using the robotic testbed.

Chapter 2

Literature Review

This chapter presents a summary of the literature and state-of-the-art methods in the area of:

- Representation and estimation for unknown 3D shape;
- Grasp planning using machine learning techniques.

Section 2.1 introduces some typical models to represent a target object, and some common algorithms to estimate unknown object shapes. In the following Section 2.2, one popular method, also known as Gaussian process implicit surface (GPIS), which has been widely used in recent years is reviewed. GPIS models the object shape as an implicit surface and estimates it by Gaussian Process Regression. GPIS as a tool is used to reconstruct the non-parametric probabilistic surfaces of the object from partial and noisy sensor data. It forms the basis of the estimation algorithm proposed in this thesis. Despite its versatility, computational complexity of GPIS makes it impractical when the training set is large. This has motivated a number of approximation methods that aim to increase the efficiency while maintaining the accuracy of GPR. Some representative approaches are discussed including strategies to select optimal sample points.

The accurate geometric information on the target object is a critical factor in the process of determining a good grasp using the conventional grasp analysis approaches. Even slight errors in the object dimension or pose may lead to irreparable failures. In the meantime, if the manipulation task is taken into account in the planning phase, then the challenge of finding a suitable grasp among an infinite number of candidates will significantly increase. A more efficient approach to address this problem is performed by applying some

machine learning techniques. Section 2.3 reviews some relevant research studies from two perspectives: learning affordance parts and classifying object categories. Then, some classic classification algorithms that are used for data analysis are reviewed in Section 2.4. One typical hierarchical clustering algorithm, known as the self-organizing map (SOM), is also reviewed. SOM is applied to develop the multi-step clustering algorithm proposed in this thesis. The performance of SOM or that of any other clustering algorithms usually suffer from two major issues: initialization and determination of the number of class. Some existing attempts to resolve these two issues are also introduced within this section.

Finally, the literature review is summaried in Section 2.5.

2.1 Object Representation and Estimation

Grasp planning for an object and the corresponding manipulation task can be improved from the geometric knowledge of the object. The estimation of such information highly depends on the chosen representation method. The choice of a suitable object representation is therefore critical for generating a successful grasping movement. This section introduces some models that are commonly used to represent objects and some typical estimation methods to the object shape.

2.1.1 Object Shape Representation

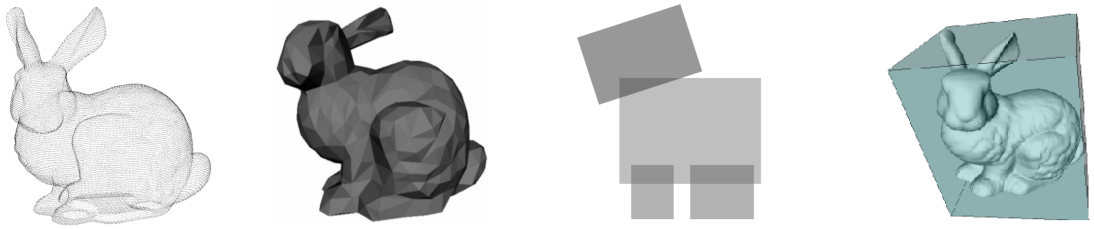
Given the first-hand sensory data, many approaches to represent the object of interest have been proposed. Figure 2.1 illustrates four different types of representation method for one object, the Stanford bunny.

Figure 2.1a shows a point cloud of the Stanford bunny obtained by a laser scan. In general, a point cloud is made by stationary laser scanners to hold a set of data points that are defined in a three-dimensional (3D) coordinate system and intended to represent the surface of the object of interest [28]. For instance, it is used to demonstrate a 3D reconstruction of an object surface that is obtained from tactile exploration with a multi-fingered robotic hand and tactile sensors in [29, 18].

Another use of point clouds is to generate a consistent polygonal mesh model via an unconstructed point cloud [30, 31]. Figure 2.1b illustrates the mesh model of the same bunny. A polygonal mesh contains a collection of vertices, edges and non-overlapping faces and consists of triangular contiguous (or quadrilateral, or some other simple convex)

polygons that are joined together along their edges. A mesh, therefore, is the easiest method to represent an object surface and is widely used in 3D computer graphics and visualization. An image-based modeling approach that renders an image on its faces at high resolution is proposed in [32]. In order to enable robotic grasp planning via shape completion, an architecture that accomplishes the shape completion from a single point cloud of a target object into a mesh model is provided in [33]. Besides the fact that meshes can provide the same accuracy as a point cloud does, they have some other advantages, including: 1) one can expand the faces and discard extra points that lie inside faces (this feature is especially useful for large flat surfaces) to show the details of the object; 2) one can easily detect intersections of polygons with other objects.

One challenging issue in robotic grasping is that there is a huge number of possible hand configurations to grasp an object for a robotic hand. A simplified method that models an object as a set of shape primitives, such as spheres, cylinders, cones and boxes, so as to choose an appropriate posture for the object is proposed in [34, 35]. Figure 2.1c shows the same bunny represented by four different sizes of rectangles. The estimation method associated with this approach will be introduced in Section 2.1.2. By implementing this representation method, the number of possible grasp can be limited.



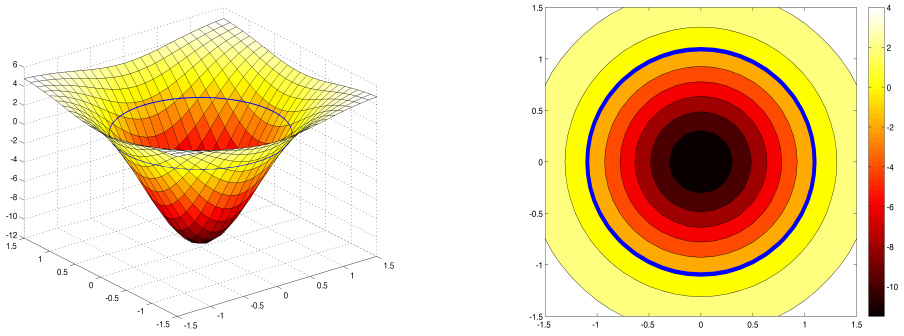
(a) Point cloud [36] (b) Mesh [36] (c) Composition [36] (d) Bounding Box [37]

Figure 2.1: Different representation of a Stanford bunny.

Modeling a 3D object by shape primitives is an efficient and valuable step for object representation [38]. However, in order to generate a good approximation by shape primitives, this approach has to suffer from the computational cost of processing a large number of data points. A promising solution that considers the tradeoff between good approximation and efficiency is proposed in [39, 37]. The basic idea of this approach, called the minimum volume bounding box computation, is originally proposed in [40]. Using this algorithm, the object shape can be simply described as a bounding box which can preserve the size of the object. Although this approach compromises accuracy of the object shape for a very compact representation, it can still largely reduce the cost of using shape primitives to approximate the object and is easy to implement.

Implicit Surface

In addition to the methods for the object representation introduced above, implicit surfaces have been extensively used in modeling environments [41], modeling objects [42] and scientific data visualization [43]. An implicit surface is a special level set which is defined as the zero level set of a real-valued function in 3D space. It can be used to determine whether a point is on the surface or not by a scalar function value. More precisely, only the points with zero function value exist on the surface. For those points which are not on the surface, their function value can be either simply described as 1 (outside) / -1 (inside) [44] or defined by the value of a signed distance function (SDF) [45]. Figure 2.2a presents an example of the zero level set of a 2D object that lives on a plane and is delimited by a contour. This contour is where the function crosses the zero-valued plane and indicated as a thick solid line. As to the points that are not on the surface, Fig. 2.2b illustrates the points with positive function values which are located outside the object (light color area) and the rest with negative function values (inside the object, dark color area).



(a) (b)

Figure 2.2: (a) The zero level set for a 2D object where $f(x) = 0$ is represented as a closed circle by an implicit function; (b) Example of different level sets for a 2D object.

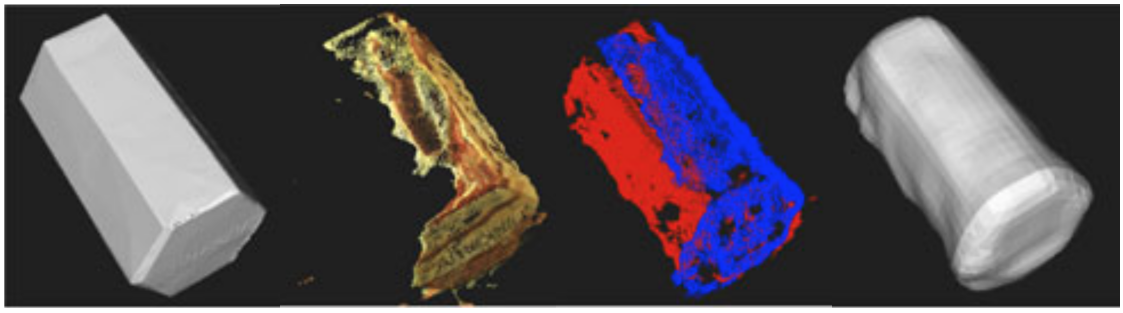
The first method using implicit surfaces to represent shapes is proposed in [46], where implicit surfaces are used to visualize 3D models of molecules in computer graphics research. In the machine learning field, later, implicit surface shape representations have been intensively investigated. A volumetric method for reconstructing surfaces by integrating multiple range images is proposed in [47]. The multi-level partition of unity implicit surface to construct surface models for local approximation is presented in [48]. An approach which is based on Support Vector Machines (SVMs) for approximating implicit surfaces

and computing deformation of 3D shapes is proposed in [49]. In this work, the author also proposed that shape deformations can be analytically described on functions rather than on the shape contours or on the shape primitives. The main reason is the difficulty or inefficiency to represent many smooth and deformable objects with such primitives, even using sphere. This idea inspired researchers to apply GP with implicit surfaces to model 2D curves or 3D surfaces and deal with shape uncertainty [4]. More specifically, given some known points on an object, a probabilistic prediction result for modeling the object can be obtained from GP posterior. The use of GP with implicit surfaces will be discussed in Section 3.2.2.

2.1.2 Object Shape Estimation

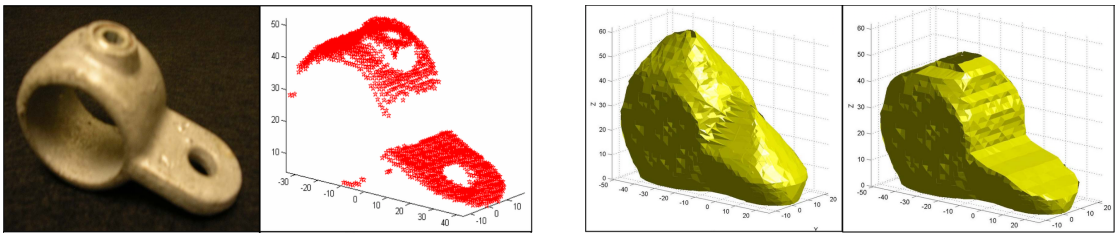
After choosing a proper representation method for the object, the next problem is how to estimate the shape of the object with available sensors for grasping the object. One promising approach is to approximate objects with shape primitives. The basic idea of this approach is to construct a shape database which contains a small number of parametric regular shapes, so that the target object can be approximated as a combination of these shape primitives. This method has been used for grasp planning in [34, 50, 51]. Assuming symmetry of the object, this method has been also applied to estimate the shape of an unknown target object from a single 3D snapshot for grasping in [52]. In this work, the raw sensory data is first processed by fitting a curve to a cross section of the point cloud of an object to exploit symmetry. Then, the back side of the object can be obtained according to the detected symmetry and the object can be reconstructed by fitting boxes and cylinders. As to the grasp planning, the grasp candidates are generated through randomization of grasp parameters (i.e. approach direction). A related approach that reconstructed a full shape of a man-made object by assuming planner reflection symmetry is presented in [2]. In this work, a complete point cloud that is merged from different view points is utilized. Figure 2.3 illustrates an example that the full shape of the target object is approximated using this approach.

Different from the techniques introduced above, an approach for grasping an object with no prior assumption about the object shape is proposed in [3]. A video camera which is mounted on the wrist of the robotic hand is used to capture the images of the object from multiple view points. After modeling the object shape, a pair of reasonably flat and parallel surfaces that are best suited for a parallel-jaw gripper is found by the proposed grasp planner. Applying this method, a process through which the shape of an object is reconstructed is illustrated in Fig. 2.4.



(a) (b) (c) (d)

Figure 2.3: Estimated full object shape by assuming symmetry. (a) Ground truth mesh; (b) Original point cloud from one side view; (c) Mirrored cloud with original points in blue and predicted points in red; (d) Reconstructed mesh based on mirroring [2].



(a) (b)

Figure 2.4: Unknown object shape estimated by shape carving. (a) (Left) Image of the target object, (Right) Point cloud of the object; (b) (Left) structured-light model, (Right) Model combined with point cloud data [3].

Another method applied to the approximation of the shape of an unknown target object for grasping is presented in [53]. In this work, the object is modeled by a quadric function, and the parameters of the quadric that fit the image data are estimated using multi-view measurements of the global object shape. The estimated object centroid determined the approach direction, and the estimated object size helped to generate the posture of the hand for grasping. Similar to this work, a related approach using a quadric approximation method for grasping an unknown object with a multi-fingered robotic hand is proposed in [54]. The experiment demonstrated in this work is also conducted with a camera which is carried by the robotic hand to record a number of images from around the target object. Instead of using shape primitives, an ellipsoid that bounded the object in each image is constructed to overestimate the object visual hull and is then approximated by a quadric.

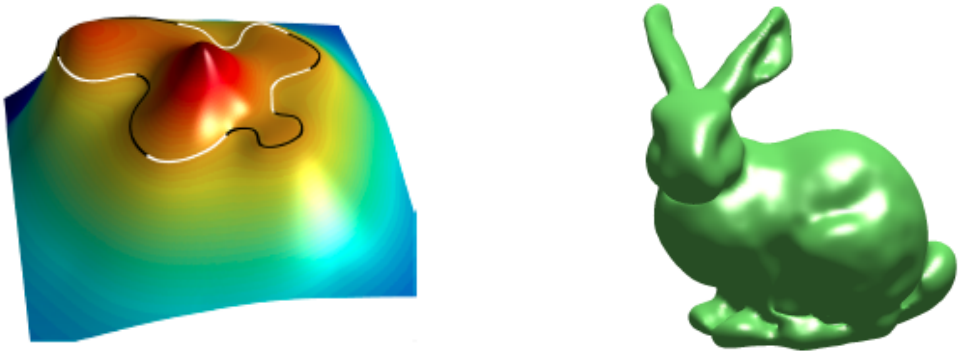
A pre-grasping shape of the hand is defined in which the initial grasp points on the quadric are in a plane parallel to the two minor axes of the ellipsoid. This grasp configuration is then refined through iterative reconstruction of the elastic surface of the object and a local optimal grasp planner.

2.2 Estimation Using Gaussian Process Regression

The research studies discussed above demonstrate that the basic information on the object shape can be acquired from different types of sensors, such as vision camera, laser scanner and tactile sensors. Such information is often corrupted by measurement noise which makes the derived knowledge uncertain. This problem cannot be efficiently solved by applying the estimation methods that have been introduced in the previous subsection, unless we can collect a large number of measurement data or just simply assume perfect measurements.

Given some noisy observations of the object shape at certain positions, what is the best estimate of the object shape anywhere else? One naturally may assume that the object shape can be described as a linear function $f(x)$. Then the function can be fitted by using a least-squares method. Moreover, even if $f(x)$ is quadratic or cubic other than linear, one can still use the principles of model selection to choose among the various possibilities. Most of the objects in the real world, however, cannot be simply described as a polynomial function due to the complexity of the shape.

A GP is a general approach that can put all noisy observed data together with some prior assumptions about $f(x)$ and predict the function value either near or far from the observed positions, rather than claiming $f(x)$ related to some specific models. Due to this unique feature of GP, it has been introduced to implicit surface modeling in [4]. This work provided a physical interpretation that the zero level set implied by a real-valued function $f(x) : \mathbb{R}^3 \rightarrow \mathbb{R}$ represented the surface of the object with non-zero volume. This type of method that estimates the function, $f(x)$, by GPs is then called Gaussian Process Implicit Surfaces (GPIS). The key idea of GPIS is to let $f(x)$ be distributed according to a GP, so that the estimation of GP can be conditioned on the corresponding function value. Furthermore, the surface contour is also random and characterized by the underlying process. This method, then, has been widely used for 3D shape reconstruction [55, 56], modeling deformable property of object [1], unknown object shape estimation and the corresponding grasp planning[44, 45], and tactile exploration [57]. Figure 2.5 demonstrates two examples for modeling a 2D curve and a 3D object. The main estimation algorithm proposed in this research proposal is based on GPIS model and will be introduced in the later chapter.



(a) (b)

Figure 2.5: Two estimation results using GPIS: estimation of a 2D contour in (a); estimation of the Stanford bunny using 800 surface points, one interior point, and a sphere of 80 exterior points [4].

2.2.1 Approximation Methods for GP Regression

One technical issue in the implementation of the GPIS is that, during inference, GP regression (GPR) gets often computationally demanding. Due to the inversion of covariance matrices, even a straightforward implementation of GPR requires $O(n^2)$ space and $O(n^3)$ time for a training dataset of n samples. Normally, there are three computational phases to implement GPR [58]. Each phase consumes different computational load.

- **Hyperparameter learning:** The hyperparameters are a set of free parameters whose values specify the covariance function (the details will be introduced in next chapter). They can be learned by maximizing the log marginal likelihood of the GPR model [59] and have a large influence on the posterior model.
- **Training:** Given a set of hyperparameters, all computations that do not involve the predicted data, are performed.
- **Prediction:** Only the computations involving the predicted data are performed. This phase may be significant if there is a very large test set.

Among these three phases, the most expensive operations are the hyperparameter learning and the inversion of the covariance matrix. Ideally, the training data for the hyperparameters is supposed to consist of all observations. However, due to the high computational

load or the availability of these measurements, some efficient computational methods are required. Notice that training the GP model on a given training set of data is the same as finding the optimal set of hyperparameters that maximizes the log marginal likelihood. Based on this idea, the Occam's razor principle that guaranteed the optimization of the hyperparameters and prevention of overfitting by the formulation of the learning mechanism is proposed in [60]. A method combining of stochastic search and gradient-based quasi-Newton optimization is presented in [61]. This gradient based method can efficiently achieve the convergence. In some cases, the log marginal likelihood may have multiple local maxima. An approximation method of the global maximum that utilizes a stochastic meta-heuristic optimizer to tune the values of hyperparameters is introduced in [62].

As to the inversion of the covariance matrix, several approximation methods have been proposed to address this problem. The simplest way is to use mathematical tools, such as Cholesky decomposition. The computational complexity is $n^3/6$ for the Cholesky decomposition in general. It is, however, not efficient when the training set is large. The most straightforward approach to dealing with a large amount of data is to select a subset of training data points and ignore the rest. This so called 'Subset of Data (SoD) approximation' simply applies GPR to the selected subset of size $m < n$. Therefore, the cost for the covariance matrix inversion using SoD would be $O(m^3)$. The selection of the subset can be performed either randomly or by an informed choice. Based on this idea, a fast selection scheme using entropy reduction techniques that took only $O(m^2 n)$ is proposed in [63]. A matching pursuit approach which had similar complexity to the previous scheme is presented in [64]. However, even though the computational costs can be efficiently reduced using SoD approximation method, there is a large difference between the predicted variance (which reflects the uncertainty) and the exact one. For instance, the predicted uncertainty may be high at certain locations because no training data points are close to them. The real uncertainty, however, should be much lower since the sample points which are close to those locations have been discarded.

An alternative approximation algorithm using self-defined covariance matrix based on induing points is proposed in [65]. The inducing points are selected as a small set of locations over the input training data. The complete dataset is then projected into the inducing points. Moreover, the Fully Independent Training Conditional (FITC) method is developed to approximate the new defined covariance matrix [66]. This idea has been applied to a related method that linearly projected all the training data points onto a lower-dimensional subspace in [67]. By combing the SoD and FITC methods, a novel approach is introduced in [58]. In this work, the author claimed that it is possible to approximate the marginal likelihood using the SoD and to apply the FITC algorithm to make predictions using the same data subset. As a result, the computational complexity

is expected to be reduced to $O(m^3)$ instead of $O(m^2n)$. Another method that divided the whole training dataset into several clusters is proposed in [68]. The basic idea is to equally divide the n training data points into $k = [\frac{n}{m}]$ clusters, and perform GPR in each cluster while discarding the training data points outside of the given cluster. The computational complexity can be reduced to $O(m^2n)$. However, the continuity of the predictions cannot be guaranteed, if the boundaries of the cluster are chosen improperly.

Alternatively, a fast nearest neighbor lookup with kd-trees has been used to efficiently compute the multiplication of a covariance matrix with a vector in [69]. Using this approximation method, the training data points are organized in kd-trees, and only the data which is closest to the query locations is processed by GPR. The size of the data storage can be reduced to $O(n)$ from $O(n^2)$.

In some particular applications, such as robot control, the training dataset normally consists of the previous knowledge about the system. In order to efficiently utilize the previous data and to combine the information from the new data, a sequential regression method is proposed in [70]. An alternative approach, named as Recursive GPR, that used the similar idea is presented in [71]. This method can guarantee the accuracy of the prediction while reducing the computation time. The sequential update method proposed in this thesis is also inspired by this idea.

2.2.2 Sample Points Selection

Based on all the approximation methods discussed above, it is obvious that the key idea of reducing the computational load of GPR is to minimize the size of training set. In our case, the training set represents the set of sample points on the object surface. Moreover, the process of estimating a 3D shape using GPIS usually involves updating the shape model every time we collect new measurement data. These technical issues suggest that we may improve the performance of the GPR for 3D shape estimation if the sample points can be selected in a more efficient (or optimal) way. One question addressed in this thesis is how we can determine the location of the next sample data such that the number of tactile data points to be sampled can be minimized without losing the accuracy in the estimated shape.

Intuitively, a good candidate for the next sample point should be selected at the place which has the highest variance, because it is normally either far away from any previous measurements or near the areas of large surface variation. In other words, this location has the lowest confidence in its prediction. This method has been applied in some recent studies to the problem of tactile exploration of the surface of an object [72, 73]. Instead of

using variance as criterion, an alternative approach to select the most informative location with respect to the entire region of interest is proposed in [74]. In this method, the amount of uncertainty, using Shannon's entropy [75], is defined as the conditional entropy of the unobserved locations after taking measurements from the observed locations. A good placement, therefore, should minimize this conditional entropy. More specifically, the selected sample points are most uncertain about each other. However, this optimization problem is shown to be **NP**-hard in [76]. In order to deal with this type of problem, the greedy algorithm that iteratively adds sample points has been commonly used [77]. Using this algorithm, a posterior entropy criterion that aimed at reducing the expected uncertainty about the posterior generalized variance of the function is applied in [78].

The entropy criterion, however, suffers from a problem that the sample points tend to be placed along the boundaries of the region. The cause of this phenomenon is that when the prediction is made based on the measurement from the region around the observed points, the estimated variance is clearly high along the boarders of the field of interest. An alternative improved criterion based on mutual information is proposed in [5]. In this work, the field of interest is considered as a discrete set of locations $V = S \cup U$ composed of two parts: a set S of positions which can be observed, and another set U of positions which no observations are possible. The goal is to find a set of positions for k sample points that would provide good predictions at all unobserved locations at $V \backslash X$. Specifically, this problem can be solved by

$$X^* = \underset{X \subseteq S}{\operatorname{argmax}} \, \mathrm{H}(V \backslash X) - \mathrm{H}(V \backslash X \mid X), \tag{2.1}$$

where for the sake of simple notation, $V \backslash X$ and X refered to the surface points at the locations $V \backslash X$ and X, respectively. Specifically, the set X^* maximally reduced the entropy over the rest of the field of interest $V \backslash X^*$. This criterion is equivalent to find the set that maximizes mutual information $I(X; V \backslash X)$ between the locations X and the rest of the space $V \backslash X$. Figure 2.6 demonstrates the difference in using entropy and mutual information to select the locations of sample point.

Based on this criterion, a smooth and well estimated surface in terms of the overall shape of the target object can be desired using GPIS [79]. Some key surface details, however, are lost in this process, due to the lack of information on some particular areas that contain distinctive geometric features. The areas with distinctive geometric features refer to local regions where high-fidelity samples are required to characterize their geometric shapes, e.g., sharp corners or high curvature areas. There are only a few attempts that take the local information (e.g., distinctive geometric features) into account for estimating the 3D shape.

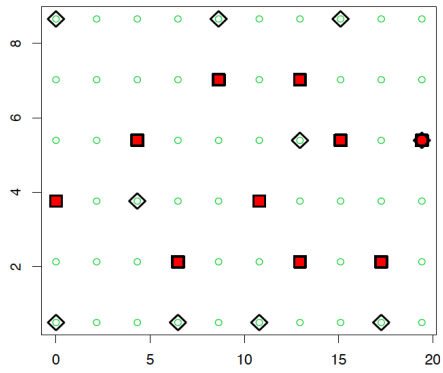
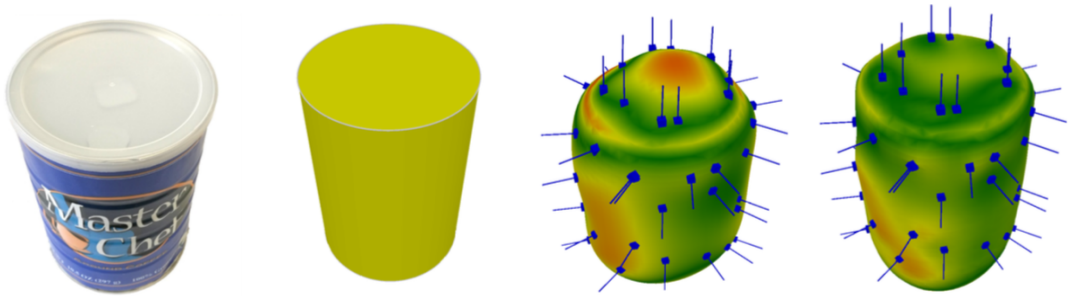


Figure 2.6: An example of sample points selection using entropy and mutual information criteria. The positions chosen using entropy are indicated as diamond sign, and the positions chosen using mutual information are indicated as square sign [5].

One recent work introduced a local implicit surface estimation algorithm that focused on the estimation of sharp corners and edges by considering the surface normals and local convexities or concavities at the contact positions [80]. Later, the same author presented a novel sensor which an Inertial Measurement Unit (IMU) and a pressure sensor to estimate the surface normal in [6]. Figure 2.7 shows two estimation results that are generated from the same sample points. The estimation of the object using only GPIS is shown in Fig. 2.7c. By considering the surface normal with GPIS, the estimation result can be improved and is shown in Fig. 2.7d. These experiment results indicate that the accuracy of the estimation of the object shape can be improved by considering the surface normal. This work, however, only focused on the estimation rather than the sampling strategy. As a result, a large number of sample points are required to render an accurate estimation for an object with standard geometric shape.

2.2.3 Estimation via Tactile Exploration Path

In tactile exploration for 3D shape estimation, the target object is reconstructed from sparse tactile data gained at the contact points where the robot finger touches down the object surface. In such a process, only one (or a few) contact point can be sampled at a time. In order to increase the efficiency of the estimation, some recent research work took into account the travel cost which is consumed during the exploration actions [81, 7]. The sample points are selected based on local information maximization and exploration cost minimization. The exploration cost is considered as the combination of the overall path



(a) (b) (c) (d)

Figure 2.7: Comparison of the estimation results of a coffee can with and without considering the surface normal information [6]. The target object (a), the ground true model (b), the estimation result using only GPIS (c) and the estimation result using GPIS and surface normal (d).

length of exploratory motions and the overall rotation of the finger during exploration. Figure 2.8 shows a simple example of exploration of a flat surface.

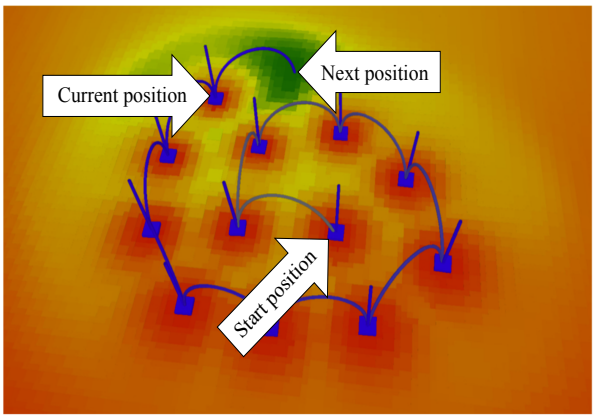


Figure 2.8: An example of exploration of a flat surface [7].

In recent years, another type of approach performed the tactile exploration in the context of paths on the surface of the object instead of a collection of discrete points [8, 19, 82]. Figure 2.9 shows an example of some paths and the estimated surface based on such paths. The variance of the estimation is visualized as the color on the surface. High variance is indicated as red, and low variance is indicated as blue. Intuitively, this type of approach is supposed to be more effective as trajectories on the surface of the object are

much more informative than a collection of points. This is because the number of points that are collected from trajectories is much larger than that are selected from active point sampling strategy. This can be seen from Fig. 2.9 that each trajectory is composed of a large number of discrete sample points, indicated as black dots. As mentioned earlier, the more sample points, the more computational load that GPR consumes. The improvement in terms of efficiency may not be as remarkable as what it has been claimed. Another issue of this type of approach is that each trajectory contains some redundant points. More specifically, there are some sample points on each trajectory that make only a little contribution to the estimation of the object shape. As discussed earlier, the good sample points should be far away from each other so that they can provide more information on the object. In other words, the closer two sample points get, the less information they can provide. In such approach, the number of sample points is determined by the frequency that is used to update GPIS model. With a high frequency, the sample points on the same trajectory are too close to each other. As a consequence, most of sample points on such trajectory do not only provide more information on the object, but increase the computational load by a significant amount. On the other hand, with a low frequency, the trajectories are nothing else but a series of discrete points. As a result, this type of approach becomes similar to the active point sampling strategy.

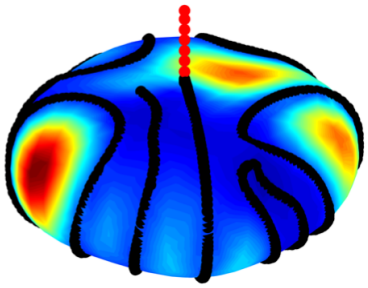


Figure 2.9: An example of surface exploration with query paths in simulation [8].

2.3 Grasp Planning

Given the geometric information on object and a set of criteria for the grasping task, how to generate a suitable grasp among an infinite number of candidates for the target object to fulfill all the requirements for manipulation is always considered as one major challenge in the development of fundamental functionalities of dexterous manipulation. In

this section, the basic idea of the conventional analytical approach for grasp planning and the limitations of such approach is first reviewed. Apart from the analytical approach, using machine learning techniques to tackle task-oriented grasp planning becomes popular in recent years. Some typical approaches are introduced within this section.

2.3.1 Grasp Analysis

Consider the case where a multi-fingered robotic hand grasps an object in 3D space, and each of the fingertips and the object make a point contact, as shown in Fig. 2.10. The contact between each finger and the object can be described as a mapping between forces exerted by the finger at the point of contact and the resultant wrenches at some reference points on the object [83]. In most of cases, the contacts are modeled based on some additional assumptions, such as Coulomb friction and rigid body modeling [83, 84].

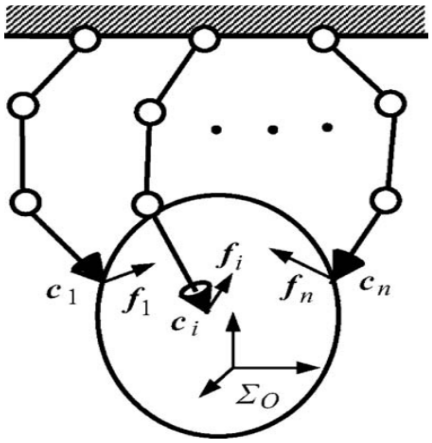


Figure 2.10: Grasp an object with a multi-fingered hand. [9]

Then, typical analytical approaches provide guarantees for the determination of the locations of the contact points for a given object in terms of the performance in resisting external wrench loads. One approach to solve this problem is to model the object as a combination of boxes so that the surface of the object can be treated as a finite number of flat faces [39]. Moreover, there is a constant normal for each face, and the position of a point on a face can be linearly parameterized by two variables. Based on these properties, assuming $n-1$ fingers had fixed positions on the object surface, a method that searched for the locations for the n-th finger to ensure *force-closure* (i.e. any external forces can be resisted by the resultant force) grasps is presented in [85]. Similarly, an algorithm that

computed the locations of the contact points to form a *force-closure* grasp from an initial random grasp is proposed in [86]. In this work, a grasp is first arbitrarily chosen on the face of a given object. Then, using the linear parametrization of a point on the object face, the algorithm moved each fingertip position until the *force-closure* property is achieved. These approaches are effective when the number of faces of the object is low. However, some commonly used objects in everyday life, such as mugs and bottles, may not easily to be modeled with a limited number of faces. Hence, these approaches suffer from a large computation load coming from the combinations of the large number of faces.

In order to deal with such an issue, objects are modeled with a cloud of 3-dimensional (3D) points or a triangular mesh. Based on geometrical analysis of the modeled object, an algorithm for computing three finger force-closure grasp for 2D and 3D objects is presented in [87]. Assuming the target object can be discretized as a large set of surface points, a complete and efficient algorithm for searching *force-closure* grasps of n hard fingers is proposed in [88]. In this work, every single point as well as their normal are available. Then, the algorithm starts with an initial set of contacts randomly chosen among the set of points and iteratively exchanges the contact points with other candidate locations until a *force-closure* grasp is obtained. Another type of approach explored the contact regions rather than the contact points without losing the *force-closure* property [89, 90].

Besides the *force-closure* property, the quality of grasp is another critical factor to determine the locations of the contact points or those of the contact regions. In order to measure the quality of each grasp, many optimization methods are proposed [91, 92]. Most of these methods computed optimal force-closure grasps by optimizing a certain objective function according to a pre-defined grasp quality criterion, such as minimum singular value of the grasp map matrix [93], area of the grasp polygon [94], distance between the centroid of the contact polygon and the object's centre of mass [95] and limitations on the finger forces [96].

In practice, most of the aforementioned approaches suffer from some problems during the implementation. One common assumption made in analytical approaches is that the precise information on structural properties (shape and weight) or surface properties (hardness and texture) of the object is available. However, such information is always uncertain, due to the noisy information coming from different types of sensors. In addition, systematic or unexpected errors that are inherited from the physical robotic system, or even the modeling errors of the robot's kinematics and dynamics can result in an inaccurate placement of fingertips. Furthermore, most of the analytical approaches do not consider task as a criterion to determine the choice of an optimal grasp. In order to address all these issues associated with the analytical approaches, some machine learning techniques have been employed. In the following subsections, some representative approaches are introduced

and discussed.

2.3.2 Task-Oriented Grasp Planning

After examining a variety of human grasps, an important conclusion is made in [97] that, besides the size and the shape of the object, the task to be performed is also a critical factor to determine the choice of an optimal grasp. It indicated that finding a good stable grasp is only a necessary but not a sufficient condition for an efficient manipulation. In other words, when some manipulation is expected after grasping, a good grasp should be made in accordance with the task at hand. For instance, although a grasp at the top of a mug satisfies the *force-closure* property, it is not a valid grasp for the task of pouring water from the mug since the cup opening would be blocked by the end-effector.

Analytical Approaches

If we carefully examine the geometric shapes of everyday objects, we can see that almost all objects are constituted by a particular component which is designed specifically to fulfill their functionalities and to make their grasps easier. For instance, the handle parts of mugs and teapots, the elongated parts of pencils and bottles and the handles of frying pans. These particular components can be extracted by the visual system of humans and are utilized to identify the novel objects [98]. Thus, most primates are able to determine a good grasp for an arbitrary object, even without the information on object shape or the knowledge of object type. Those components that are preferable to be grasped for certain tasks are normally referenced as the affordance parts. The grasps generated by applying this idea are called the semantic grasps. Inspired by these observations, a number of approaches are proposed to identify the affordance parts of the objects so that the generation of a task-oriented grasp can be made. The basic idea of this type of approach is illustrated as a flowchart in Fig. 2.12. In order to identify the affordance parts, a method imitating the human behaviors to choose a good grasp is proposed in [99, 10]. In this work, the contact points are computed to ensure the *force-closure* property. Figure 2.11 shows the basic procedure to detect the different types of natural graspable parts of some objects. The idea of segmentation is also applied with depth information in [100]. In this work, a supervised classifier is developed to detect only the graspable components of the object. The results of this work also demonstrated that the grasp planning can be significantly simplified by employing the segmentation algorithm. Another approach for estimating affordance parts only from 2D sources is developed in [101]. By modeling the everyday objects as 3D point cloud data, the similar method to detect the affordance

parts are applied in [102, 103]. In recent years, due to the efficient utilization of collected data and the promising results from the learning-based approaches, machine learning has attracted a lot of attention in robotics field. Some approaches that used convolutional neural networks (CNN) for object affordance detection for particular tasks are proposed recently in [104, 105].

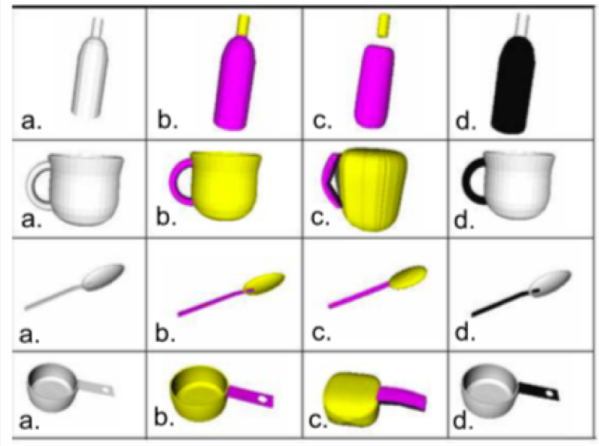


Figure 2.11: The procedure to detect the natural graspable part. The true shape of the objects (a), segmented into two parts (b), using shape primitives to approximate each constituting part (c) and the natural graspable part (d) [10].

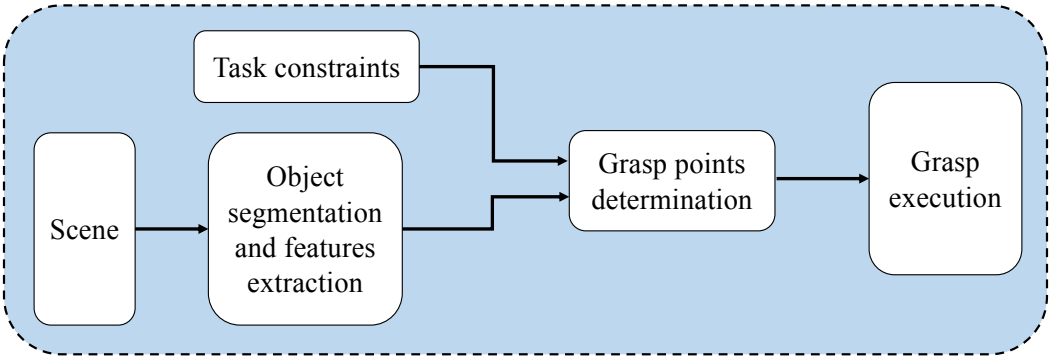


Figure 2.12: A typical functional flowchart of a system that generates task-oriented grasps by detecting the affordance parts for a particular task [11].

Affordance Components Detection

One key result demonstrated in these approaches shows that two different objects can be grasped in the same way as long as they shared similar affordance components. Based on this idea, another type of approach utilizes the labeled dataset of simulated grasps to simplify the grasp planning process. Figure 2.13 presents a functional flowchart that summarizes this type of approach. In the offline phase, a number of categories are created, and each category contains some training objects which share similarities in affordance parts. Each category normally contains a number of features, such as tasks, object features, actions and constraints. These features play a major role in the generation of a task-oriented grasp, and they can either be encoded via simulations or learned from human demonstrations. For each training object, a number of grasps that satisfy those features are generated. In the online phase, the novel object is first segmented so that the affordance part can be extracted. The detected affordance part is then approximated and categorized to determine to which category the novel object belongs. Instead of searching for the contact points and analyzing the corresponding grasp stability, a suitable grasp can be acquired from the dataset. This framework is used with a Bayesian network in [106, 107]. In this work, grasps that are encoded into the dataset are first instantiated in simulation and then labeled by human users. The trained network demonstrates the ability to detect the grasp points for a given task and is capable of determining if the target object is suitable for the given task. Later, this work is extended to include tactile information so that the stability of the generated task-oriented grasps can be evaluated in [108]. A semantic map that maps the local object geometry to a set of predefined constraints, such as graspable parts, relative orientation of the hand to the graspable part, hand posture, approaching direction and tactile data on the hand, is proposed in [109]. A set of semantic grasps is first generated with respect to the related local geometry. While the hand approaches the object, the hand is adjusted by maximizing the similarity between the planned grasp and the recorded grasp which corresponds to the current approaching direction. The results shown in this work demonstrate that the learned semantic grasps can also be applied to other similar objects. In order to improve the transferability from the training objects to the novel object within the same object category, a local optimization method is proposed in [110].

Category-Based Approaches

In the aforementioned approaches, objects that share similarities in affordance components are grouped in one category and one task-oriented grasp can be transferred within the

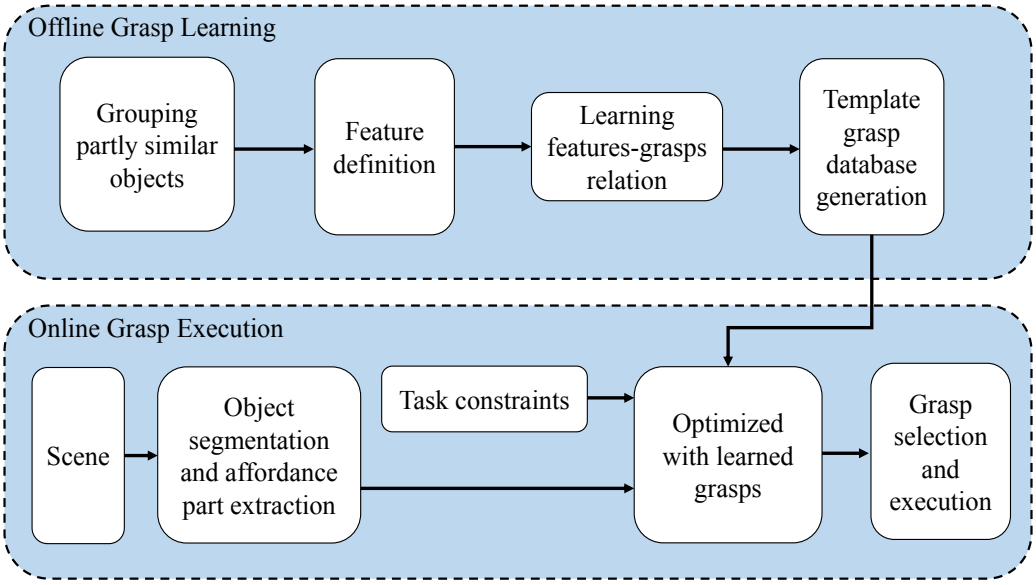
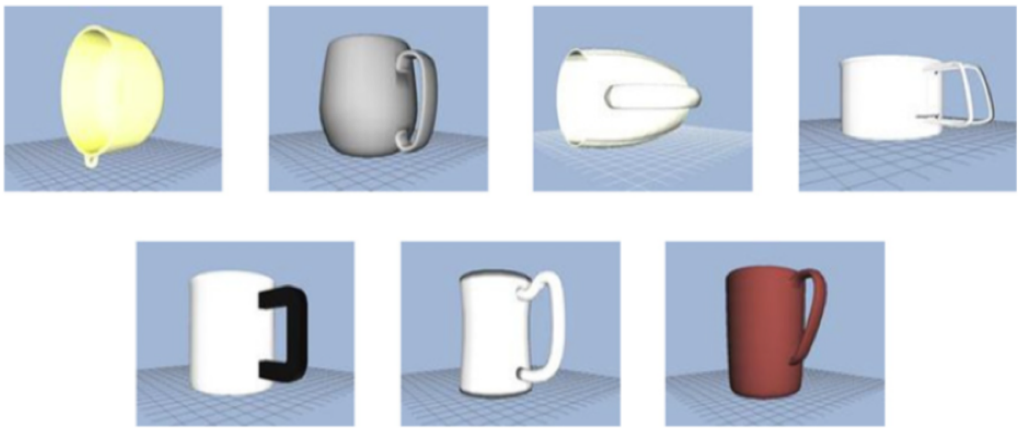


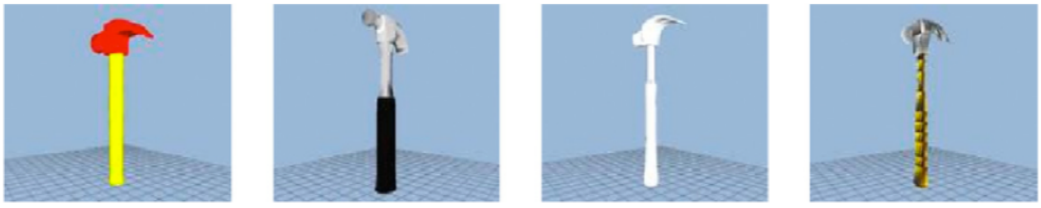
Figure 2.13: A typical functional flowchart of a system that generates task-oriented grasps by creating a semantic grasp dataset [11].

same category. Some research studies expand on this idea in a more general concept that everyday objects can be grouped together into categories based on shape and functionality [111]. More specifically, the objects within the same category can be used to fulfill the same task. Figure 2.14 shows two typical examples of category which contain similar objects in terms of shape and functionality. In addition, an observation from analyzing human grasping behavior indicates that similar objects can be grasped by using similar or even same grasp types [23]. Based on these ideas, a number of approaches have been proposed recently. Figure 2.15 illustrates a typical flowchart of such approaches. This type of approach is normally accomplished into two phases: offline phase and online phase. In the offline phase, some training objects that share similarities in shape and functionality are grouped together. Then, every training object is processed to generate grasp candidates for a certain task. All of such grasp candidates are then ranked based on some specific criteria, such as stability, probability for the desired task. Finally, the ranked grasp candidates and the corresponding objects are stored together in the dataset. In the online phase, the category to which the novel object belongs is first determined. Based on the shape of the novel object and the intended task, the object which is the most similar to the novel object is found in the dataset, and the grasp with the highest rank is retrieved. Normally, such grasp is optimized with the novel object and then executed. This kind of framework is

utilized in [26]. In this work, the shape, the size and the weight of the objects are assumed to follow Gaussian distributions. Then, a set of representative objects are constructed based on the mean of such features and used for each object type. The training data is collected in a simulation environment provided by GraspIt! [112]. This framework is also applied to physical experiments in [27]. In this work, the most similar object to the novel one is retrieved from the database, and the associated grasp that is the most suitable for the task is selected and applied to the novel object. Both of these two approaches, however, is unable to handle large variation in shape within the category, if the novel object is distinct from the training models. A more advanced approach that optimizes the grasp using all training models in the category is then proposed in [12, 113].



(a) Mug category from Columbia Grasp Database [114].



(b) Hammer category from Columbia Grasp Database [114].

Figure 2.14: Two examples of category: mugs category (a) and hammers category (b) [12].

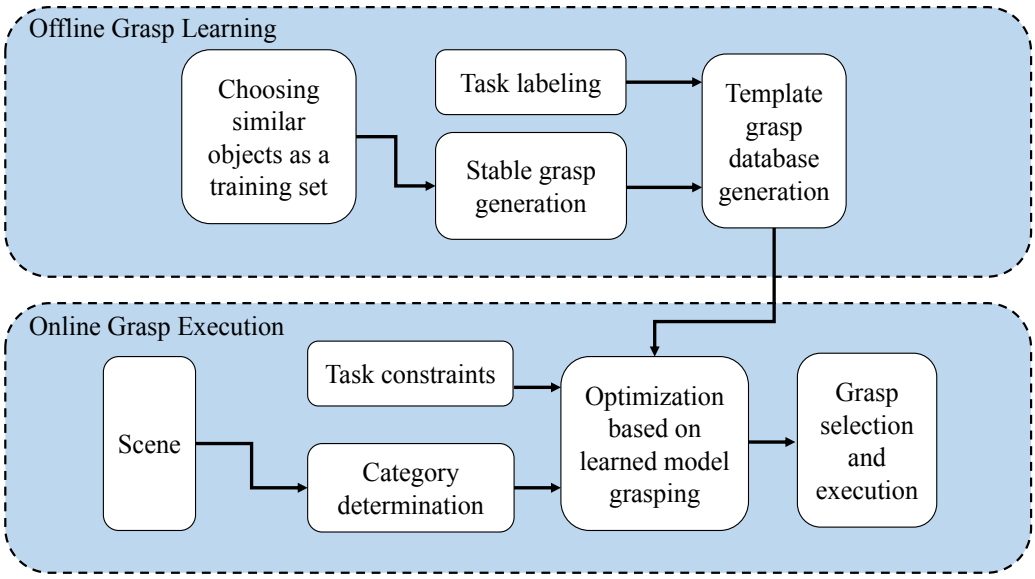


Figure 2.15: A typical functional flowchart of a system that generates grasps for the similar objects within the predefined categories [11].

2.4 Clustering for Grasp Candidates

All of the category-based approaches introduced previously can be categorized as data-driven approaches [11]. In the implementation of this type of approaches, more grasp data are generated, with more reliable grasp expected to be rendered for a target object, especially when the target object is not contained in the training set. On the other hand, the computational demand for analyzing the grasp data and that of planning a grasp would be significantly increased.

The research presented in [115] demonstrates that a globally optimal grasp is not necessarily required, as an average grasp (in the sense of *force-closure*) is acceptable for many common tasks. More specifically, a representative hand pose can be applied to grasp a group of objects that are very similar in shape. Based on this idea, the k-medoids algorithm [116] is applied to group the similar grasps together in [26]. As a result, a large number of grasp candidates generated for the training objects are condensed into a small set of representative grasps. The k-medoids algorithm is an unsupervised learning technique and related to the k-means algorithm which is the most common algorithm for the clustering problems. Both the k-medoids and k-means algorithm aim to group the data

into a few cohesive clusters. The k-means attempts to minimize the value of the distortion function which is defined as the sum of squared distances between each training data and the cluster center/centroid to which it has been assigned. Such cluster center may not belong to the training data. In contrast to the k-means algorithm, the k-medoids algorithm minimizes the sum of dissimilarities between each data and one of the training data which has been designated as the center of that cluster. As a consequence, the k-medoids is more robust to noise as compared to the k-means [117]. The k-medoids clustering algorithm is as follows

Input:	(1) n training data points (2) Number of clusters, k
Output:	k clusters and medoids (centers/centroids) of each cluster
1:	**Initialize**: randomly select k of the n data points as the initial medoids.
2:	**Assignment**: associate each data point to the closest medoid.
3:	**Update**: For each medoid, x_m, and each data point, x, associated with μ, swap x_m and x and compute the total cost of the configuration (that is the average dissimilarity of x to all the data points associated with x_m). Select the medoid, x, with the lowest cost of the configuration.
4:	Repeat steps 2 and 3 until there is no change in the assignments.

Table 2.1: K-medoids algorithm

There are many possible choices of metrics to measure the dissimilarity, such as Euclidean distance, Manhattan distance and Gaussian kernel. Euclidean distance is the most common choice, since it is easy to implement in practice. Manhattan distance is also known as the l_1 norm of the different between two vectors and given by

$$d_{man}(p, q) = \|p - q\|_1, \tag{2.2}$$

where p and q are two vectors in the same dimension. In general, Manhattan distance is a better option when the training data in different units. Gaussian kernel is a non-linear function of Euclidean distance and induces a geometric space which is well suited for clustering. Specifically, similar data can be pulled together closer and pushed away even further apart. Due to this unique feature of Gaussian kernel, it has been implemented to classify the grasp candidates in [118, 119].

Besides k-means and k-medoids, another popular choice of unsupervised learning is a Self-Organizing Map (SOM), also known as a Kohonen feature network [120]. It is a competitive feed-forward neural network and an effective algorithm for dimensionality reduction and feature extraction. It has been widely used for clustering problem in text clustering [121], image processing [122, 123], neuropsychological studies [124] and robotics [125]. A typical SOM normally consists of two layers: input layer and computational/competitive layer. Each neuron in the competitive layer is fully connected to all the original data in the input layer. Then, any high-dimensional data can be transformed into a one- or two-dimensional discrete map while preserving the topological structure of the original data. Specifically, the neurons in the competitive layer are normally arranged in a two-dimensional space, and each neuron contains a weighting vector that has the same dimension as the input data. For each input data, the value of a discrimination function (often defined by Euclidean distance) for each neuron is computed. The neuron which has the minimum value of such function is declared as the winning neuron (i.e. the best matching unit (BMU)). In other words, the winning neuron is the most similar to the input data. Then, the similarity between the winning neuron and the input data gets enhanced through the suitable adjustment of the weighting vector of the winning neuron and that of its neighboring neurons. Eventually, the locations of the neurons in the competitive layer are rearranged so that the SOM forms a topological map which preserves the same distribution as the input data.

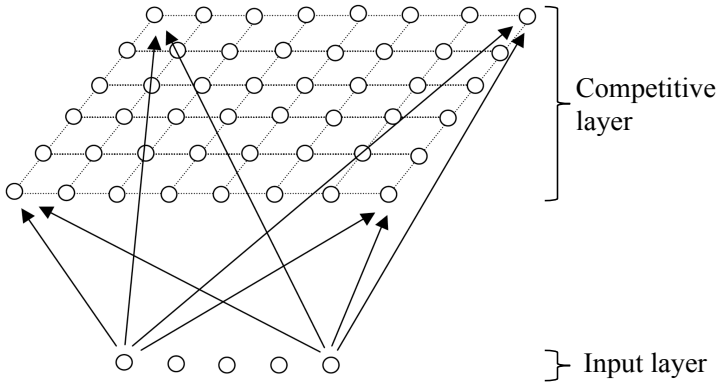


Figure 2.16: An example of the basic structure of SOM. The competitive layer is a two-dimensional space in this example.

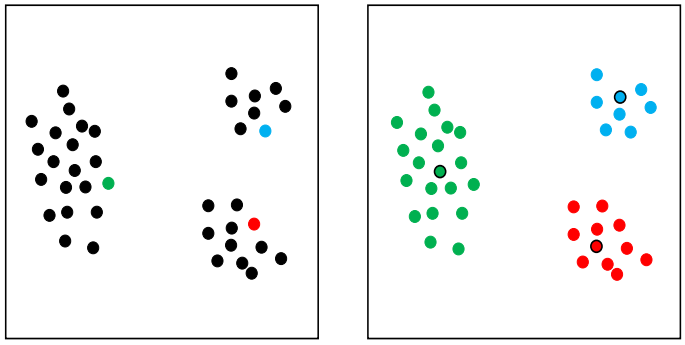
There are two different learning techniques for SOM: online/sequential learning and batch learning [120]. As mentioned previously, using online learning, the weighting vectors of the winning neuron and its neighboring neurons are updated immediately after the

determination of the winning neuron. This process is repeatedly performed for each input vector. This technique, however, suffers from large computational load as it constantly updates the weights and a high dependence on the order of the input data. As to the batch learning [126, 127], the weights remain the same while processing the input data and updated after examining every input data in the entire training set. Hence, batch learning technique is slightly more efficient in terms of computational cost and independent of the input order.

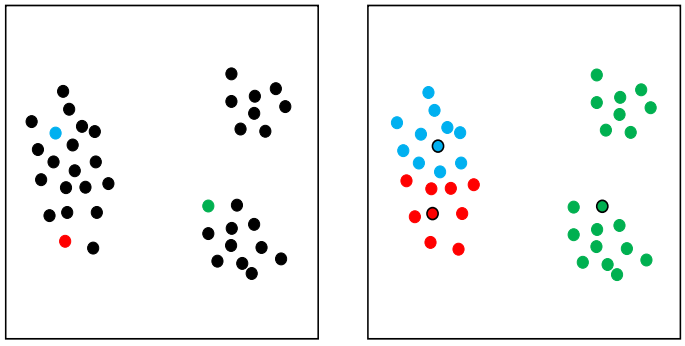
2.4.1 Initialization and Class Number Determination

Despite being used in a wide range of applications, SOM still suffers from two technical issues like other unsupervised learning algorithms: initialization of weight values for neurons and the determination of the number of clusters. As an iterative algorithm, SOM is sensitive to initial staring conditions of each neurons in the competitive layer. Figure. 2.17 illustrates an example of clustering using SOM. If the initial positions of the neurons are properly chosen, the input data can be well classified in a few of iterations, as shown in Fig. 2.17a. Otherwise, quality of the clustering is low, but the total of iteration is also increased, as shown in Fig. 2.17b. Furthermore, as many other clustering algorithms, SOM requires a fixed structure of the network in terms of the number of neurons and sequence of processing such neurons. Obviously, due to the unknown input data characteristics, all these conditions can not be satisfied in real-world applications.

There are a lot of initialization methods developed for clustering algorithms. Among these different methods, the most popular choices are random initialization, Forgy approach [128], Macqueen approach [129], the k-means++ method [130] and the Kaufman approach [116]. Suppose the desired number of clusters is k, using random initialization, the total input date is first randomly divided into k clusters. Then, the initial centroids of each cluster are computed. Random initialization is usually the first choice, since it is easy to implement and can also provide a good performance. Forgy approach works in a similar way to random initialization, as the initial centroids are randomly selected from the input data. Based on Forgy approach, Macqueen approach assigns each input data to the cluster with the closest centroid and recalculates the centroids after each assignment. Although these two approaches are easy to understand and implement, they can not provide reliable initialization and are normally ineffective [131]. Using the k-means++ method, the first centroid is selected at random from the input dataset. The i-th centroid is selected with the probability which is proportional to the distance to the previously selected centroids. As a result, the selected centroids are not close to each other. Unlike all these approaches, Kaufman approach is a greedy algorithm. The initial centroids are obtained by iteratively



(a) 3 clusters with 2 iterations.



(b) 3 clusters with 4 iterations.

Figure 2.17: An example of clustering with different initialization.

selecting k representative data from the input dataset. The details of this approach is as follows

The performance of all these approaches has been evaluated and compared in [132, 131]. The results suggest that the random initialization and Kaufman approach are the two best options in terms of performance. Moreover, Kaufman approach is normally more efficient compared to random initialization. Specifically, Kaufman approach requires a less number of iterations to converge to a local minima.

As mentioned at the beginning of this section, another drawback of SOM is that the

Input:	(1) n training data points (2) Number of clusters, k
Output:	k initial centroids
1:	**Select** the most centrally located data as the first centroid
2:	**for** every unselected data, x_i, **do**
3:	**Calculate** $C_{ji} = \max(D_j - d_{ji}, 0)$, where D_j is the Euclidean distance between x_i and its closest centroid, d_{ji} is the Euclidean distance between x_i and x_j
4:	**Calculate** $\sum_j C_{ji}$
5:	**end for**
6:	**Select** x_i which maximizes $\sum_j C_{ji}$
7:	**Stop**, if k centroids are selected. Otherwise, back to Step 2.

Table 2.2: Kaufman approach

structure of the network (i.e. the number of clusters) has to be predetermined. In most of the cases, there is no such information on details of the network in advance. Suppose the input data needs to be grouped into k clusters, one possible way is to choose some different values of k and apply the clustering algorithm to part of the data. Using cross-validation method to evaluate each clustering result, the optimal value of k can then be determined based on the predefined criteria [116, 133, 134]. Obviously, such approach consumes a significant amount of computational time. In order to address this technical issue, a number of networks which is capable of automatically adjusting the size of themselves to achieve optimal clustering have been developed. One classic approach that use statistic information on the input data to determine the size of the network is proposed in [135]. Specifically, a new centroid is inserted at the location where the data density of that cluster is much higher than others. On the other hand, a centroid is removed if the number of data around it is much lower than others. This approach has been recently improved by applying cluster operations (i.e. cluster creating and cluster merging) in [136]. Another type of approaches build a hierarchical structure using the self-organizing concept to classify dynamic data and preserve the topology property of such data [137, 138, 139]. Such hierarchical structures are normally referenced as tree structures. The number of output trees and the number of nodes contained in each tree can be estimated with no prior information on the input data.

2.5 Conclusion

Combing GP and implicit surface presents an opportunity to estimate the shape of any unknown object. However, the implementation of GP consumes a large computational load, when there is a large test set or when the object shape is complex. The approximation methods discussed in the literature provide an idea that the most useful way to improve the efficiency of GP is to reduce the size of the training set. In order to estimate the unknown shape with minimum number of sample points, some specific criterion are required. The review in Section 2.2 has highlighted an opportunity to employ different criterion in the use of GPIS to determine the locations of the sample points. Combining the mutual information and the geometric features of the objects, this approach has potential to lead to an optimal sampling process that balances between the exploration of the whole shape and the exploitation of the local area where the higher fidelity (or more sampling) is required.

The recently proposed category-based grasp planning methods are based on the hypothesis that the objects share similarities in the functional component or even in the whole shape can be grasped using a similar type of hand pose. Apparently, this implies there exists a certain type of hierarchical relation between the sampling grasp candidates. In order to retain such relation of the collected data while clustering similar grasp candidates together, SOM is clearly a better choice compared to k-means and k-medoids.

Learning and practicing are the most two essential characteristics of all human beings. In particular, people normally attempt to accomplish some actions based on what they have learned and refine the skills through practice. However, almost all approaches reviewed in Section 2.3.2 construct a fixed size of training set and are not capable of updating such training set after each grasp execution. In order to address this issue, a dynamic framework that can automatically adjust the structure (i.e. inserting or removing a neuron and updating the weighting vectors) after grasping every novel objects is needed.

Chapter 3

Multi-Fidelity Sampling for Fast Bayesian Shape Estimation with Tactile Exploration

3.1 Introduction

As reviewed in Chapter 2, the object shape can be effectively modeled as an implicit surface and estimated by GPR. With limited prior information about the object size and location, the robot manipulator needs to iteratively choose a sequence of touch-down positions on the object surface. The more the robot touches on the surface, the better knowledge with respect to the object shape will be built up. Eventually, an accurate estimation of the object shape needs to be constructed with a limited number of sample points. However, due to the computational cost, sample points need to be selected in a rather smart way.

To address this issue, the optimal sample point selection problem is considered with a view to its application to tactile exploration tasks operating under the condition that the realistic 3D shape can be acquired with a smaller number of samples, thereby allowing us to estimate the object shape more rapidly. For this, two criteria are employed in the use of GPIS to determine the locations of the sample points: the maximal increase in mutual information and the concentration on high curvature area. Based on mutual information, the amount of information on the object can be maximized after selecting a sample point at each iteration. Based on curvature, the geometric features of the object can be provided from the locations of the sample points. Because the number of sample points is limited, there is a need to compromise these two criteria to determine the locations of the sample

points. One way to resolve such a multi-objective problem is to use a trade-off function that incorporates both criteria into a single objective function by the application of a weighting parameter. A trade-off function is then developed to induce a multi-fidelity shape estimation scheme as the weighting parameter gets successively refined based on the curvature estimation.

In this chapter, the proposed optimal sample point selection strategy and a sequential GPR are introduced. The former is related to the optimal sample point selection problem, and the latter is concerned with increasing the computational efficiency of GPR.

3.2 Surface Modeling and Estimation

This section reviews GPIS that forms the basis of the proposed optimal sample points selection strategy and the estimation algorithm used in this paper.

3.2.1 Implicit Surface

Let $\Upsilon \subset \mathbb{R}^3$ represent the region in which the object of interest locates, and let a subset $X \subset \Upsilon$ contain all possible (finite number of) sample points in Υ except the boundary points. A subset $X_o \subset X$ denotes all the points on the surface of the object, a finite subset $X_{s_n} = \{x_{s_1}, \cdots, x_{s_n}\} \subset X_o$ denotes the actual sample points on the surface and $X \backslash X_{s_n}$ indicates all the unobserved points within X.

Followed by the concept introduced in Section 2.1.1, the surface of the target object is represented as the zero level set of an implicit surface, and the implicit function used in this thesis is then defined as

$$f(x) \mapsto \begin{cases} < 0, & \text{if point } x \text{ inside the surface} \\ = 0, & \text{if point } x \text{ on the surface} \\ > 0, & \text{if point } x \text{ outside the surface} \end{cases} \tag{3.1}$$

where $x \in \mathbb{R}^3$ denotes a coordinate in Υ and $f : \mathbb{R}^3 \to \mathbb{R}$ is defined as a real-valued function. The advantage of using this method is that implicit surface can model any complex surface without specifying the exact shape.

3.2.2 Gaussian Process Implicit Surface

As mentioned previously, we employ GPR to estimate the shape of the target object within the region of interest. The estimate improves progressively as more sample points are collected.

In general, a GP (denoted by $\mathcal{GP}$) is used to describe a distribution over functions with a continuous domain (e.g., time or space). Formally

Definition 3.2.1. A Gaussian process is a collection of random variables, any finite number of which have a joint Gaussian distribution. [59]

Thus, a GP can be completely specified by its mean function $m(x)$ and covariance function $k(x, x')$. We first consider a real GP $f(x)$ written as

$$f(x) \sim \mathcal{GP}(\mathrm{m}(x), \mathrm{k}(x_i, x_j)), \tag{3.2}$$

where $x_i, x_j \in \mathbb{R}^3$ are the input data. In our case, the random variables from the above GP model represent the values of the implicit function $f(x)$ at position x. More specifically, for every point $x \in \Upsilon$ we have a corresponding noisy measurement y, described as

$$y = f(x) + \varepsilon, \tag{3.3}$$

where ε is defined as independent and identically distributed white Gaussian noise with variance σ_ε^2.

The covariance function, also known as the kernel function, can be defined by different models, such as squared exponential covariance function and thin plate covariance function [4]. In this thesis, it is assumed to be isotropic, which means that the covariance function is a function only of the distance between x_i and x_j. Furthermore, the classic squared exponential covariance function is used to specify the covariance between pairs of inputs, because it can produce very smooth functions and thus is suitable to represent the object surface. The classic squared exponential covariance function is defined as

$$\mathrm{k}(x_i, x_j) = \sigma_0^2 \exp\left(-\frac{\|x_i - x_j\|_2^2}{2\sigma_l^2}\right) + \delta_{ij}\sigma_\epsilon^2, \tag{3.4}$$

where σ_0, σ_ε and σ_l are hyperparameters of the covariance function. σ_0^2 is the signal variance and defines the overall vertical scale of variation of the function values. σ_ϵ^2 is the

noise variance of the GP model and δ_{ij} is the Kronecker delta where $\delta_{ij} = 1$ only when $i = j$ and $\delta_{ij} = 0$ otherwise. σ_l defines the characteristic length-scale which characterizes the variation rate of the correlation between two input positions. More specifically, a short length-scale mean the predictive variance can grow rapidly away from the observed data points. On the contrary, a large length-scale can make the function values spatially irrelevant.

In general, if we let $\theta = \{\sigma_0^2, \sigma_\varepsilon^2, \sigma_l\}$ represent all the hyperparameters, then the values of the hyperparameters can be learnt by maximizing the log marginal likelihood [59]:

$$\log p(Y \mid X, \theta) = -\frac{1}{2}\log|\Sigma_{XX}| - \frac{1}{2}Y^\intercal \Sigma_{XX}^{-1} Y^\intercal - \frac{n}{2}\log 2\pi, \tag{3.5}$$

where Y is the function value of X obtained by Eq. (3.3).

Due to the feature of GP, if we have a training set which consists of a set of input spatial locations X_{s_n}, and a set of corresponding measurements $Y_{s_n} = \{y_{s_1}, \cdots, y_{s_n} \mid y_{s_i} \in \mathbb{R}\}$, we can predict the function value, f_*, at any point $x_* \in X \backslash X_{s_n}$ conditioned on these measurements. The distribution of f_* given these measurements is a Gaussian

$$p\left(f_* \mid x_*, X_{s_n}, Y_{s_n}\right) \sim \mathcal{N}\left(\mu\left(x_* \mid X_{s_n}, Y_{s_n}\right), \mathbb{V}\left(x_* \mid X_{s_n}\right)\right), \tag{3.6}$$

whose conditional mean $\mu\left(x_* \mid X_{s_n}, Y_{s_n}\right)$ and the conditional variance $\mathbb{V}\left(x_* \mid X_{s_n}\right)$ are given by

$$\mu\left(x_* \mid X_{s_n}, Y_{s_n}\right) = \mathrm{m}(x_*) + \Sigma_{x_* X_{s_n}} \Sigma_{X_{s_n} X_{s_n}}^{-1} [Y_{s_n} - \mathrm{m}\left(X_{s_n}\right)], \tag{3.7}$$

$$\mathbb{V}\left(x_* \mid X_{s_n}\right) = \Sigma_{x_* x_*} - \Sigma_{x_* X_{s_n}} \Sigma_{X_{s_n} X_{s_n}}^{-1} \Sigma_{X_{s_n} x_*}, \tag{3.8}$$

where $\mu\left(x_* \mid X_{s_n}, Y_{s_n}\right)$ and $\mathbb{V}\left(x_* \mid X_{s_n}\right)$ denote the predicted (or estimated) mean and variance at location x_*, respectively. Σ is the covariance matrix, also known as kernel matrix, with entries generated by Eq. (3.4). Specifically, $\Sigma_{x_* x_*}$ is the auto-covariance for the function value of the location x_*, i.e., $\Sigma_{x_* x_*} = \mathrm{k}(x_*, x_*)$, and $\Sigma_{x_* X_{s_n}}$ denotes the cross-covariance between the function value of x_* and that at a point in X_{s_n}, i.e., $(\Sigma_{x_* X_{s_n}})_i = \mathrm{k}(x_*, x_{s_i})$. For the sake of simplifying the notation, x_* and X_{s_n}, when used as subscripts, will represent $f(x_*)$ and $f(X_{s_n})$ and will do so in the rest of this thesis. We then take the zero level set of the conditional mean as the estimated shape of the object. Denoting by $\widehat{X}_o$ the set of all points on the estimated surface after collecting n samples (X_{s_n}), we have the estimated shape described by

$$\widehat{X}_o = \{\hat{x}_o \mid \mu(\hat{x}_o \mid X_{s_n}) = 0,\ \hat{x}_o \in \Upsilon\}\,. \tag{3.9}$$

with the i-th element of the set $\widehat{X}_o$ denoted by $\hat{x}_{o_i}$.

3.3 Optimal Sample Points Selection

As mentioned in Chapter 2, the implementation of GPIS bears a large computational load as the training set increases. One efficient way to reduce the computational load of such process is to minimize the size of training set. In other words, sample points need to be optimally selected so that an accurate estimation can be obtained with a minimum number of sample points.

Such objective can be achieved by applying mutual information criterion. The details of such criterion will be introduced in the first part of this section. Some key surface details, however, are lost in this process, if mutual information is the only criterion to be considered. To address this issue, a novel multi-fidelity-based optimal sample points selection method is presented in the second part of this section. The proposed objective function can automatically balance the effect of two different criteria (i.e. mutual information and curvature) through a weighting factor.

3.3.1 Mutual Information

In order to let the algorithm gather information on the whole object shape, a mutual information criterion is applied. The mutual information, $\mathrm{MI}(X_{s_n})$, quantifies the amount of information obtained about the rest of the space of interest $X \backslash X_{s_n}$, through a set of sampled points, X_{s_n}, and is given by

$$\mathrm{MI}(X_{s_n}) = \mathrm{I}(X_{s_n}; X \backslash X_{s_n}) = \mathrm{H}(X \backslash X_{s_n}) - \mathrm{H}(X \backslash X_{s_n} \mid X_{s_n}), \tag{3.10}$$

where $\mathrm{H}(X \backslash X_{s_n})$[1] and $\mathrm{H}(X \backslash X_{s_n} \mid X_{s_n})$ denote the marginal entropy that represents the amount of information about the unobserved space $X \backslash X_{s_n}$ and the conditional entropy of $X \backslash X_{s_n}$ after obtaining measurement set X_{s_n}, respectively. By definition, the conditional entropy can be calculated as

[1] $\mathrm{H}(X) = \mathbb{E}[-\log \mathrm{P}(X)]$, where X is a discrete random variable with a probability mass function $\mathrm{P}(X)$.

$$\begin{aligned} \mathrm{H}(X \backslash X_{s_n} \mid X_{s_n}) &= \mathrm{H}(X \backslash X_{s_n}, X_{s_n}) - \mathrm{H}(X_{s_n}) \\ &= \mathrm{H}(X) - \mathrm{H}(X_{s_n}). \end{aligned} \tag{3.11}$$

Furthermore, the entropy of a Gaussian random variable $f(x_*)$ conditioned on the set of variables $f(X_{s_n})$ is a monotonic function of its variance, i.e.,

$$\begin{aligned} \mathrm{H}(x_* \mid X_{s_n}) &= \frac{1}{2} \log \left(2\pi e \mathbb{V}(x_* \mid X_{s_n})\right) \\ &= \frac{1}{2} \log \mathbb{V}(x_* \mid X_{s_n}) + \frac{1}{2} \left(\log(2\pi) + 1\right), \end{aligned} \tag{3.12}$$

where e is the Euler's number and the predicted value of $\mathbb{V}(x_* \mid X_{s_n})$ can be computed via Eq. (3.8), using GPR.

Optimizing the mutual information criterion via Eq. (3.10) is an **NP**-complete problem [5]. In other words, the time required to find the optimal set X_{s_n} increases very quickly as the size of X_{s_n} grows. Therefore, instead of searching for the entire set X_{s_n}, we can greedily and sequentially search for the next optimal sample point, $x^*_{s_{n+1}}$. If each desired next sample point is assumed to be selected from the estimated object surface, $\widehat{X}_o$, then each optimal sample point should minimize the total amount of uncertainty about the unmeasured space, $\widehat{X}_o \backslash X_{s_n}$, based on the previous measurements X_{s_n}. This can be expressed as

$$x^*_{s_{n+1}} = \underset{x_{s_{n+1}} \in \widehat{X}_o \backslash X_{s_n}}{\operatorname{argmax}} \phi(x_{s_{n+1}}), \tag{3.13}$$

where the increased amount of information, $\phi(x_{s_{n+1}})$, that the next sample point $x_{s_{n+1}}$ can provide with respect to the unobserved area can be expressed using Eqs. (3.10), (3.11) and (3.12) as

$$
\begin{aligned}
\phi(x_{s_{n+1}}) &= \mathrm{MI}(X_{s_n} \cup x_{s_{n+1}}) - \mathrm{MI}(X_{s_n}) \\
&= \{\mathrm{H}(\overline{X}) - \mathrm{H}(\overline{X} \mid X_{s_n} \cup x_{s_{n+1}})\} - \{\mathrm{H}(X \backslash X_{s_n}) - \mathrm{H}(X \backslash X_{s_n} \mid X_{s_n})\} \\
&= \{\mathrm{H}(\overline{X}) - \mathrm{H}(\widehat{X}_o) + \mathrm{H}(X_{s_n} \cup x_{s_{n+1}})\} - \{\mathrm{H}(\overline{X} \cup x_{s_{n+1}}) - \mathrm{H}(\widehat{X}_o) + \mathrm{H}(X_{s_n})\} \\
&= \{\mathrm{H}(X_{s_n} \cup x_{s_{n+1}}) - \mathrm{H}(X_{s_n})\} - \{\mathrm{H}(\overline{X} \cup x_{s_{n+1}}) - \mathrm{H}(\overline{X})\} \\
&= \mathrm{H}(x_{s_{n+1}} \mid X_{s_n}) - \mathrm{H}(x_{s_{n+1}} \mid \overline{X}) \\
&= \frac{1}{2} \log \left(\frac{\Sigma_{x_{s_{n+1}} x_{s_{n+1}}} - \Sigma_{x_{s_{n+1}} X_{s_n}} \Sigma^{-1}_{X_{s_n} X_{s_n}} \Sigma_{X_{s_n} x_{s_{n+1}}}}{\Sigma_{x_{s_{n+1}} x_{s_{n+1}}} - \Sigma_{x_{s_{n+1}} \overline{X}} \Sigma^{-1}_{\overline{X}\overline{X}} \Sigma_{\overline{X} x_{s_{n+1}}}} \right),
\end{aligned}
\tag{3.14}
$$

where $\overline{X}$ denotes the set $\widehat{X}_o \backslash (X_{s_n} \cup x_{s_{n+1}})$. The details of this algorithm are shown in Table 3.1.

Input:	(1) Object field $\Upsilon \subset \mathbb{R}^3$ (2) Number of sample points n_s
Output:	Selected sample points $X_{s_n} \subset X_o$
1:	**Begin** with $X_{s_n} = \{x_{s_1}, \ldots, x_{s_n}\}$
2:	**for** $i = 1$ to n_s **do**
3:	**for all** $x_{s_{n+i}} \in \widehat{X}_o \backslash X_{s_n}$ **do**
4:	$x^*_{s_{n+i}} \leftarrow \mathrm{argmax}_{x_{s_{n+i}} \in \widehat{X}_o \backslash X_{s_n}} \phi(x_{s_{n+i}})$
5:	**end for**
6:	**Update** $X_{s_{n+i}} = X_{s_{n+i-1}} \cup x^*_{s_{n+i}}$
7:	**end for**

Table 3.1: Optimal Sample Point Selection Algorithm for Maximizing Mutual Information

3.3.2 Local Curvature Estimation

Equation (3.14) allows us to choose the best next sample point in the sense of better estimation of the whole object shape. The selected sample points, however, may not provide enough information on some particular areas that contain distinctive geometric features. The areas with distinctive geometric features are referred to the local regions

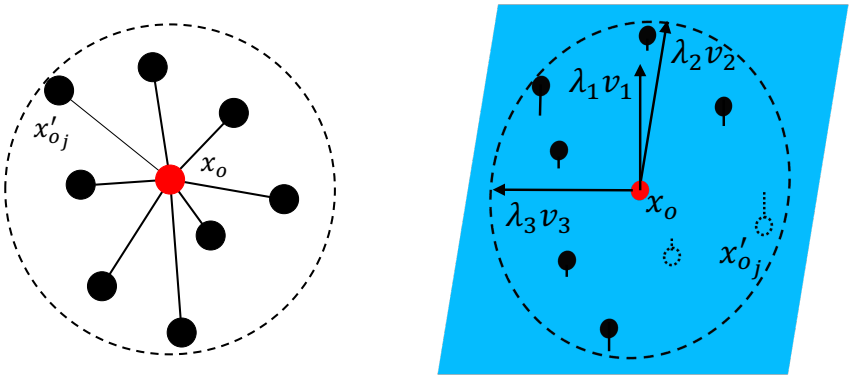
with relatively high curvature, such as corners and edges. In order to choose some high-fidelity samples to characterize such geometric shapes, the local curvature is considered as the additional criterion to determine the location of the sample points.

Although many different methods for curvature estimation have been proposed, the simplest method is based on the analysis of the eigenvalues and eigenvectors of the curvature covariance matrix of a local neighborhood [140, 141]. For each surface point, $x_o \in X_o$ (indicated as red dot in Fig. 3.1), we select k nearest neighboring points, $\{x'_{o_1}, \cdots, x'_{o_k}\}$ (indicated as black dot in Fig. 3.1). Then, the curvature covariance matrix, $\mathcal{C} \in \mathbb{R}^{3\times3}$ for each x_o can be expressed as

$$\mathcal{C} = \frac{1}{k}\sum_{j=1}^{k}(x'_{o_j} - x_o)(x'_{o_j} - x_o)^\intercal, \tag{3.15}$$

where the eigenvalues and the eigenvectors of $\mathcal{C}$ can be calculated by

$$\mathcal{C}v_\ell = \lambda_\ell v_\ell, \ell \in \{1, 2, 3\}\,. \tag{3.16}$$



(a) (b)

Figure 3.1: Selection of local neighborhood (a), and covariance analysis (b).

Since $\mathcal{C}$ is symmetric and positive semi-definite, its eigenvalues are all non-negative real numbers. Moreover, the eigenvectors, v_ℓ, form an orthogonal basis, corresponding to the principal components of the neighboring points. Assuming, $\lambda_1 < \lambda_2 < \lambda_3$, the surface

normal at point x_o can be approximated as the eigenvector v_1. In addition, as shown in Fig. 3.1b, the tangent plan (indicated as a blue plan Fig. 3.1b) at x_o is spanned by v_2 and v_3.

The variation along each of the eigenvector, v_ℓ, can be approximately described as the corresponding eigenvalue, λ_ℓ. Then, the total variation, i.e. the sum of the Euclidean distance from the centroid x_o to its neighbors, is given by

$$\sum_{j=1}^{k} \|x'_{o_j} - x_o\|_2^2 = \lambda_1 + \lambda_2 + \lambda_3. \tag{3.17}$$

In particular, the variation along the surface normal can be quantified as the minimum eigenvalue. We then define

$$\gamma(x_o) = \frac{3\lambda_{\min}}{\lambda_1 + \lambda_2 + \lambda_3}, \quad \left(\lambda_{\min} = \min_{i=1,2,3} \lambda_i\right). \tag{3.18}$$

as the surface variation at point x_o and use it to approximate the curvature at surface point x_o. Because the value of $\gamma(x_o)$ varies between 0 and 1, the case of $\gamma(x_o) = 0$ corresponds to a perfectly planar surface, and $\gamma(x_o) = 1$ is for an isotropic surface with 3 identical eigenvalues.

3.3.3 Objective Function

As introduced earlier, Eq. (3.14) emphasizes the exploration of the whole shape of the object, and Eq. (3.18) approximates the curvature of the object surface. In this subsection, we propose a method that automatically and adaptively selects optimal sample points in the sense of the estimation of the object shape while taking the geometric features into account.

Given a candidate for the next sample point, $\hat{x}_{o_i}$ ($\hat{x}_{o_i} \in \widehat{X}_o$), the value of the increase of the mutual information, $\phi(\hat{x}_{o_i})$, and the value of the curvature, $\gamma(\hat{x}_{o_i})$, are first normalized such that each value lies between 0 and 1. Specifically,

$$\Phi(\hat{x}_{o_i}) = \frac{\phi(\hat{x}_{o_i}) - \min\limits_{\hat{x}_o \in \widehat{X}_o} \left(\phi\left(\hat{x}_o\right)\right)}{\max\limits_{\hat{x}_o \in \widehat{X}_o} \left(\phi\left(\hat{x}_o\right)\right) - \min\limits_{\hat{x}_o \in \widehat{X}_o} \left(\phi\left(\hat{x}_o\right)\right)}, \tag{3.19}$$

$$\Gamma(\hat{x}_{o_i}) = \frac{\gamma(\hat{x}_{o_i}) - \min\limits_{\hat{x}_o \in \widehat{X}_o} \left(\gamma\left(\hat{x}_o\right)\right)}{\max\limits_{\hat{x}_o \in \widehat{X}_o} \left(\gamma\left(\hat{x}_o\right)\right) - \min\limits_{\hat{x}_o \in \widehat{X}_o} \left(\gamma\left(\hat{x}_o\right)\right)}. \tag{3.20}$$

The normalized quantities of $\Phi(\hat{x}_{o_i})$ and $\Gamma(\hat{x}_{o_i})$ can be used to construct an objective function

$$T_i\left(\hat{x}_{o_i}\right) = \alpha_{n+1}\Phi(\hat{x}_{o_i}) + (1 - \alpha_{n+1})\Gamma(\hat{x}_{o_i}), \tag{3.21}$$

where the weighting factor α_{n+1} provides the flexibility and adaptability in balancing the effect of two different criteria, and is formulated as

$$\alpha_{n+1} = \frac{\log\left(\sqrt{\nu_n} + 1\right)}{\log 2}, \tag{3.22}$$

where $\nu_n = \frac{1}{N}\operatorname{tr}\left(\mathbb{V}\left(X \mid X_{s_n}\right)\right)$ denotes the mean of the predicted variance of set X and N is the total number of points in X. The denominator, $\log 2$, scales the numerator value to make sure the value of α_{n+1} lies on the range $[0, 1]$. In effect, the weighting factor α_{n+1} controls the fidelity of the estimation of the local characteristic features of the 3D shape. Apparently, the more sample points are selected, the smaller the value of α_{n+1} becomes and the algorithm focuses more on selecting points from the areas with high curvature (which require high fidelity representation), rather than the unexplored areas. Eventually, the proposed algorithm can be written as greedily selecting a series of optimal sample points which can maximize the value of T_i. Specifically, after collecting n samples,

$$\hat{x}^*_{o_{n+1}} = \underset{\hat{x}_{o_i} \in \widehat{X}_o, \Phi(\hat{x}_{o_i}) \geq 0.5}{\operatorname{argmax}} T_i\left(\hat{x}_{o_i}\right). \tag{3.23}$$

As the number of sample points increases, the proposed algorithm will focus more on the high curvature regions and overlook some unexplored areas. In order to let the algorithm retain the capability of the exploration while exploiting some high curvature regions, we add an extra condition ($\Phi(\hat{x}_{o_i}) \geq 0.5$) to make sure that the sample points do not focus only on those particular areas.

Since the (desired) next optimal sample point, $\hat{x}^*_{o_{n+1}}$, may not be on the actual surface of the object, there are three different scenarios that may happen during the sampling process with tactile exploration: the expected sample point is inside, on, or outside the object surface. When we use a robotic manipulator to sample the surface points, the manipulator can be commanded to move along the direction of the surface normal at the desired sample point, $\hat{x}^*_{o_{n+1}}$, which can be estimated by Eqs. (3.15) and (3.16). If the expected sample point is on the actual surface, then an accurate measurement can be obtained. If the point is inside the actual surface, obviously the point cannot be reached. A measurement is consequently obtained at the point where the manipulator makes a touch while approaching the estimated object centre. If the point is outside of the actual surface, then the manipulator keeps moving along the approach direction until it touches the surface and that touch-down point is considered as an actual sample point. The details of the implementation of the proposed method to select a total of n_s sample points are shown in Table 3.2.

3.4 Sequential Gaussian Process Regression

In this section, we propose an algorithm which contains a sequential update method to efficiently update the conditional variance of GP and some other ways to reduce the required computational cost of implementing GP.

If we carefully examine Eq. (3.14), the numerator and the denominator inside the logarithm are nothing else but the conditional variance, $\mathbb{V}(x_{s_{n+1}} \mid X_{s_n})$ and $\mathbb{V}(x_{s_{n+1}} \mid \overline{X})$. In addition, both of them are calculated for every single points in $\widehat{X}_o/X_{s_n}$). This is shown on step 6 in Table 3.2. Furthermore, each entry in X_{s_n} has already been selected from previous iteration, and all entries in $\Sigma_{\overline{X}\,\overline{X}}$ have been computed at the first iteration while selecting the first sample point. Apparently, such process can be be made efficient in some way.

Suppose the sample set is updated to the nth step, i.e. $X_{s_n} = \{x_{s_1}, x_{s_2}, \ldots, x_{s_n}\}$, and the estimated shape $\widehat{X}_o$ consists of m possible sample points, i.e. $\widehat{X}_o = \{\hat{x}_{o_1}, \hat{x}_{o_2}, \ldots, \hat{x}_{o_m}\}$. The conditional mean and the conditional variance of $x_{s_{n+1}}$ given X_{s_n} can be obtained by Eq. (3.7) and Eq. (3.8) and expressed as

$$\mu(x_{s_{n+1}} \mid X_{s_n}) = \Sigma_{x_{s_{n+1}} X_{s_n}} \Sigma^{-1}_{X_{s_n} X_{s_n}} Y_{s_n}, \tag{3.24}$$

$$\mathbb{V}(x_{s_{n+1}} \mid X_{s_n}) = \Sigma_{x_{s_{n+1}} x_{s_{n+1}}} - \Sigma_{x_{s_{n+1}} X_{s_n}} \Sigma^{-1}_{X_{s_n} X_{s_n}} \Sigma_{X_{s_n} x_{s_{n+1}}}, \tag{3.25}$$

Input: (1) Object field of interest $\Upsilon \subset \mathbb{R}^3$
(2) Number of sample points n_s

Output: Sample points $X_{s_{n_s}}$

Begin with $X_{s_n} = \{x_{s_1}, \ldots, x_{s_n}\}$
for $i = 1$ to n_s **do**
  Estimate the object shape, $\widehat{X}_o$, by Eq. (3.7)
  Calculate the weighting factor α by Eqs. (3.8) and (3.22)
  for $\hat{x}_{o_i} \in \widehat{X}_o \backslash X_{s_{n+i-1}}$ **do**
    Calculate the mutual information of each point on the estimated surface, $\phi(\hat{x}_{o_i})$, by Eq. (3.14)
    Calculate the curvature of each point on the estimated surface, $\gamma(\hat{x}_{o_i})$, by Eq. (3.18)
    Normalize $\phi(\hat{x}_{o_i})$ and $\gamma(\hat{x}_{o_i})$, by Eqs. (3.19) and (3.20)
    Calculate T_i by Eq. (3.21)
  end for
  Determine the optimal sample point, $\hat{x}^*_{o_{n+i}}$, by Eq. (3.23)
  Move along the approach direction toward $\hat{x}^*_{o_{n+i}}$ until obtaining the actual touch down point, $x^*_{s_{n+i}}$
  Update $X_{s_{n+i}} = X_{s_{n+i-1}} \cup x^*_{s_{n+i}}$
end for

Table 3.2: The multi-fidelity optimal sample points selection algorithm

For notational simplicity, the value of the mean function is assumed to be zero. The computation complexity of both Eq. (3.24) and Eq. (3.25) is $O(s_n^3)$, because of the term $\Sigma^{-1}_{X_{s_n} X_{s_n}}$. In other words, they grow at the order of s_n^3.

Consider the set X_{s_n} as a union of $X_{s_{n-1}}$ and x_{s_n}. Because the values of the squared exponential covariance function are always positive, $\mathrm{k}(\bullet, \bullet') > 0$, all the sample points in X are correlated. Hence, Eq. (3.24) can be rewritten as

$$
\begin{aligned}
\mu(x_{s_{n+1}} \mid X_{s_n}) &= \mathbb{E}[x_{s_{n+1}} \mid X_{s_{n-1}}, x_{s_n}] \\
&= \mathbb{E}[x_{s_{n+1}} \mid X_{s_{n-1}}] + \mathbb{E}[\tilde{x}_{s_{n+1}|s_{n-1}} \mid \tilde{x}_{s_n|s_{n-1}}], \\
&= \Sigma_{x_{s_{n+1}} X_{s_{n-1}}} \Sigma^{-1}_{X_{s_{n-1}} X_{s_{n-1}}} Y_{s_{n-1}} \\
&\quad + \frac{\Sigma_{x_{s_{n+1}} x_{s_n}} - \Sigma_{x_{s_{n+1}} X_{s_{n-1}}} \Sigma^{-1}_{X_{s_{n-1}} X_{s_{n-1}}} \Sigma_{X_{s_{n-1}} x_{s_n}}}{\Sigma_{x_{s_n} x_{s_n}} - \Sigma_{x_{s_n} X_{s_{n-1}}} \Sigma^{-1}_{X_{s_{n-1}} X_{s_{n-1}}} \Sigma_{X_{s_{n-1}} x_{s_n}}} \\
&\quad + \frac{Y_{s_n} - \Sigma_{x_{s_n} X_{s_{n-1}}} \Sigma^{-1}_{X_{s_{n-1}} X_{s_{n-1}}} Y_{s_{n-1}}}{\Sigma_{x_{s_n} x_{s_n}} - \Sigma_{x_{s_n} X_{s_{n-1}}} \Sigma^{-1}_{X_{s_{n-1}} X_{s_{n-1}}} \Sigma_{X_{s_{n-1}} x_{s_n}}}.
\end{aligned} \tag{3.26}
$$

where

$$
\tilde{x}_{s_{n+1}|s_{n-1}} = x_{s_{n+1}} - \mathbb{E}[x_{s_{n+1}} \mid X_{s_{n-1}}], \tag{3.27}
$$

$$
\tilde{x}_{s_n|s_{n-1}} = x_{s_n} - \mathbb{E}[x_{s_n} \mid X_{s_{n-1}}]. \tag{3.28}
$$

Applying the similar idea, Eq. (3.25) can be expressed as

$$
\begin{aligned}
\mathbb{V}(x_{s_{n+1}} \mid X_{s_{n-1}}, x_{s_n}) &= \Sigma_{\tilde{x}_{s_{n+1}|s_{n-1}} \tilde{x}_{s_{n+1}|s_{n-1}}} \\
&\quad - \Sigma_{\tilde{x}_{s_{n+1}|s_{n-1}} \tilde{x}_{s_n|s_{n-1}}} \Sigma^{-1}_{\tilde{x}_{s_n|s_{n-1}} \tilde{x}_{s_n|s_{n-1}}} \Sigma_{\tilde{x}_{s_n|s_{n-1}} \tilde{x}_{s_{n+1}|s_{n-1}}} \\
&= \Sigma_{x_{s_{n+1}} x_{s_{n+1}}} - \Sigma_{x_{s_{n+1}} X_{s_{n-1}}} \Sigma^{-1}_{X_{s_{n-1}} X_{s_{n-1}}} \Sigma_{X_{s_{n-1}} x_{s_{n+1}}} \\
&\quad - \frac{(\Sigma_{x_{s_{n+1}} x_{s_n}} - \Sigma_{x_{s_{n+1}} X_{s_{n-1}}} \Sigma^{-1}_{X_{s_{n-1}} X_{s_{n-1}}} \Sigma_{X_{s_{n-1}} x_{s_n}})^2}{\Sigma_{x_{s_n} x_{s_n}} - \Sigma_{x_{s_n} X_{s_{n-1}}} \Sigma^{-1}_{X_{s_{n-1}} X_{s_{n-1}}} \Sigma_{X_{s_{n-1}} x_{s_n}}}.
\end{aligned} \tag{3.29}
$$

The details of the procedure to derive Eqs. (3.26) and (3.29) are presented in Appendix A.

Apparently, the computational complexity of the conditional variance has been reduced to $O((s_{n-1})^3)$ by using Eq. (3.29). In addition, because each entry of $\Sigma_{X_{s_{n-1}} X_{s_{n-1}}}$ can be retrieved from the previous iterations, $\Sigma_{X_{s_{n-1}} X_{s_{n-1}}}$ can be expressed as

$$\Sigma_{X_{s_{n-1}}X_{s_{n-1}}} = \begin{bmatrix} \Sigma_{x_{s_1}x_{s_1}} & \Sigma_{x_{s_1}x_{s_2}} & \cdots & \Sigma_{X_{s_{n-2}}x_{s_{n-1}}} \\ \Sigma_{x_{s_2}x_{s_1}} & \Sigma_{x_{s_2}x_{s_2}} & & \\ \vdots & & \ddots & \\ \Sigma_{x_{s_{n-1}}X_{s_{n-2}}} & & & \Sigma_{x_{s_{n-1}}x_{s_{n-1}}} \end{bmatrix}, \tag{3.30}$$

Similarly, $\Sigma_{x_{s_n}X_{s_{n-1}}}$ and $\Sigma_{X_{s_{n-1}}x_{s_n}}$ can also be expressed as follows:

$$\Sigma_{x_{s_n}X_{s_{n-1}}} = \Sigma^{\intercal}_{X_{s_{n-1}}x_{s_n}} = \begin{bmatrix} \Sigma_{x_{s_n}x_{s_1}} & \Sigma_{x_{s_n}x_{s_2}} & \cdots & \Sigma_{x_{s_n}x_{s_{n-1}}} \end{bmatrix}. \tag{3.31}$$

As to other metrics, $\Sigma_{x_{s_{n+1}}x_{s_{n+1}}}$, $\Sigma_{x_{s_{n+1}}x_{s_n}}$, $\Sigma_{x_{s_{n+1}}X_{s_{n-1}}}$ and $\Sigma_{X_{s_{n-1}}x_{s_{n+1}}}$, they have to be computed at each iteration since they are associated with the new measurement $x_{s_{n+1}}$. Note that the denominator in Eq. (3.29) is exactly the conditional variance, $\mathbb{V}(x_{s_n} \mid X_{s_{n-1}})$, which has been computed on the last step.

Because the numerator of Eq. (3.14) is exactly the formula of the conditional variance, the sequential update formulation presented above can improve its computation efficiency. As to the denominator of Eq. (3.14), some computational techniques can also be applied to reduce the computational time.

Every time when we are on step 3 and before we proceed to the second loop, $\overline{X} = \widehat{X}_o$ (due to the precision of the data points, we may assume $X_{s_n} \notin \overline{X}$). Hence, $\Sigma_{\overline{X}\overline{X}}$ consists of the correlation between each possible sample points on the estimated surface, $\widehat{X}_o$, and is given by

$$\Sigma_{\overline{X}\overline{X}} = \begin{bmatrix} \mathrm{k}(\hat{x}_{o_1}, \hat{x}_{o_1}) & \mathrm{k}(\hat{x}_{o_1}, \hat{x}_{o_2}) & \cdots & \mathrm{k}(\hat{x}_{o_1}, \hat{x}_{o_m}) \\ \mathrm{k}(\hat{x}_{o_2}, \hat{x}_{o_1}) & \mathrm{k}(\hat{x}_{o_2}, \hat{x}_{o_2}) & \cdots & \mathrm{k}(\hat{x}_{o_2}, \hat{x}_{o_m}) \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{k}(\hat{x}_{o_m}, \hat{x}_{o_1}) & \mathrm{k}(\hat{x}_{o_m}, \hat{x}_{o_2}) & \cdots & \mathrm{k}(\hat{x}_{o_m}, \hat{x}_{o_m}) \end{bmatrix}. \tag{3.32}$$

Instead of calculating $\Sigma_{\overline{X}\overline{X}}$ for each individual $\hat{x}_{o_i}$ at each iteration, one can compute $\Sigma_{\overline{X}\overline{X}}$ only once before the second loop starts and simply eliminate the row and column which are associated with $\hat{x}_{o_i}$ in the following iterations. For instance, when calculating the value of Eq. (3.14) for $\hat{x}_{o_i}$ on the step 6, $\Sigma_{\overline{X}\overline{X}}$ can be expressed as

$$\Sigma_{\overline{X}\overline{X}} = \begin{bmatrix} \mathrm{k}(\hat{x}_{o_1}, \hat{x}_{o_1}) & \dots & \mathrm{k}(\hat{x}_{o_1}, \hat{x}_{o_{i-1}}) & \mathrm{k}(\hat{x}_{o_1}, \hat{x}_{o_{i+1}}) & \dots & \mathrm{k}(\hat{x}_{o_1}, \hat{x}_{o_m}) \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \mathrm{k}(\hat{x}_{o_{i-1}}, \hat{x}_{o_1}) & \dots & \mathrm{k}(\hat{x}_{o_{i-1}}, \hat{x}_{o_{i-1}}) & \mathrm{k}(\hat{x}_{o_{i-1}}, \hat{x}_{o_{i+1}}) & \dots & \mathrm{k}(\hat{x}_{o_{i-1}}, \hat{x}_{o_m}) \\ \mathrm{k}(\hat{x}_{o_{i+1}}, \hat{x}_{o_1}) & \dots & \mathrm{k}(\hat{x}_{o_{i+1}}, \hat{x}_{o_{i-1}}) & \mathrm{k}(\hat{x}_{o_{i+1}}, \hat{x}_{o_{i+1}}) & \dots & \mathrm{k}(\hat{x}_{o_{i+1}}, \hat{x}_{o_m}) \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \mathrm{k}(\hat{x}_{o_m}, \hat{x}_{o_1}) & \dots & \mathrm{k}(\hat{x}_{o_m}, \hat{x}_{o_{i-1}}) & \mathrm{k}(\hat{x}_{o_m}, \hat{x}_{o_{i+1}}) & \dots & \mathrm{k}(\hat{x}_{o_m}, \hat{x}_{o_m}) \end{bmatrix}. \quad (3.33)$$

Moreover, $\Sigma_{\hat{x}_{o_i}\overline{X}}$ and $\Sigma_{\overline{X}\hat{x}_{o_i}}$ can also be acquired from Eq. (3.32) and be given by:

$$\Sigma_{\hat{x}_{o_i}\overline{X}} = \Sigma_{\overline{X}\hat{x}_{o_i}}^{\mathsf{T}} = \begin{bmatrix} \mathrm{k}(\hat{x}_{o_i}, \hat{x}_{o_1}) & \dots & \mathrm{k}(\hat{x}_{o_i}, \hat{x}_{o_{i-1}}) & \mathrm{k}(\hat{x}_{o_i}, \hat{x}_{o_{i+1}}) & \dots & \mathrm{k}(\hat{x}_{o_i}, \hat{x}_{o_m}) \end{bmatrix}. \quad (3.34)$$

3.5 Simulation and Experimental Results

To validate the performance of the proposed sampling method, we conducted a number of simulations and experiments to estimate different 3D shapes. The performance of using mutual information as a criterion to select sample points is first compared with that of using variance. Then, the proposed algorithm that incorporates mutual information and local curvature is simulated to estimate two objects with complex shape, a rubber duck and the Stanford bunny. Finally, the algorithm is validated by executing a tactile exploration operation to the shapes of a tea pot on a real robotic manipulator (Barrett WAM and BarrettHand).

3.5.1 Comparison of Different Criteria

As mentioned in Chapter 2, in most of the existing studies on the tactile exploration, the optimal sample points are selected at the locations with the highest uncertainty. Specifically, these studies searched for the places in which the shape is poorly estimated due to the missing information or high noise. However, the performance of this type of strategy is limited, because the locations with the highest uncertainty did not necessarily provide the largest information with respect to the entire region of interest. Moreover, this type of strategy suffered from a problem that the sample points are intended to be placed along the boarders of the region, because the estimated variance is high near the boundaries. On

the contrary, the mutual information criterion considers the increase amount of the total information on the whole shape, rather than the prediction quality at the particular place. As a result, the sample points that are selected using mutual information as criterion are most uncertain about each other and cover the space of interest well.

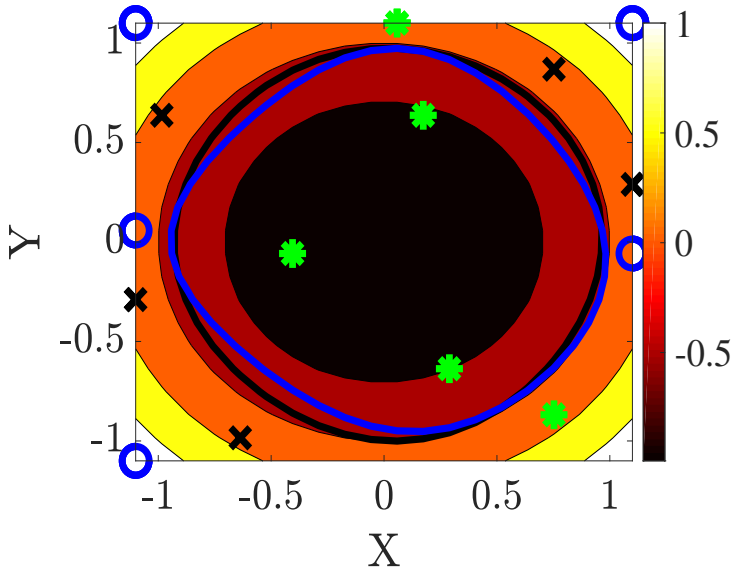


Figure 3.2: The estimation of the implicit surface of $x^2 + y^2 = 1$.

Figure 3.2 illustrates a simple example that the implicit surface (zero level set) of a real-valued function $x^2 + y^2 = 1$ is estimated using two different types of criteria: variance and mutual information. The values of the hyperparameters are fixed as $[1, 1, 1]$ in this example. The implicit surface of such function is a circle which is represented as the boundary between red and orange regions. Five sample points which are represented as the green asterisk ('$*$') marks are randomly selected at the beginning. Then, based on the two criteria, only five more sample points are iteratively selected from the region of interest ($-1.1 \leq x \leq 1.1$, $-1.1 \leq y \leq 1.1$). The blue circle ('$\circ$') marks indicate the locations of the sample points with highest variance at each iteration. The black cross ('$\times$') marks indicate locations of the sample points that provide the most increase of information in regard to the entire region of interest at each iteration. Using variance as selection criterion, it can been seen that all of the sample points are selected from the boundary of the space. Such selections from the boundary are likely to waste information and result in a poor estimation, as shown as a blue contour in Fig. 3.2. On the same simple example, the mutual information criterion leads to search for the sample points at the locations where

lack the information with respect to the whole space. As a result, the selected sample points are spread out over the entire region and a good estimation can be obtained, as shown as a black contour in Fig. 3.2.

3.5.2 Validation of Proposed Sequential GP Regression

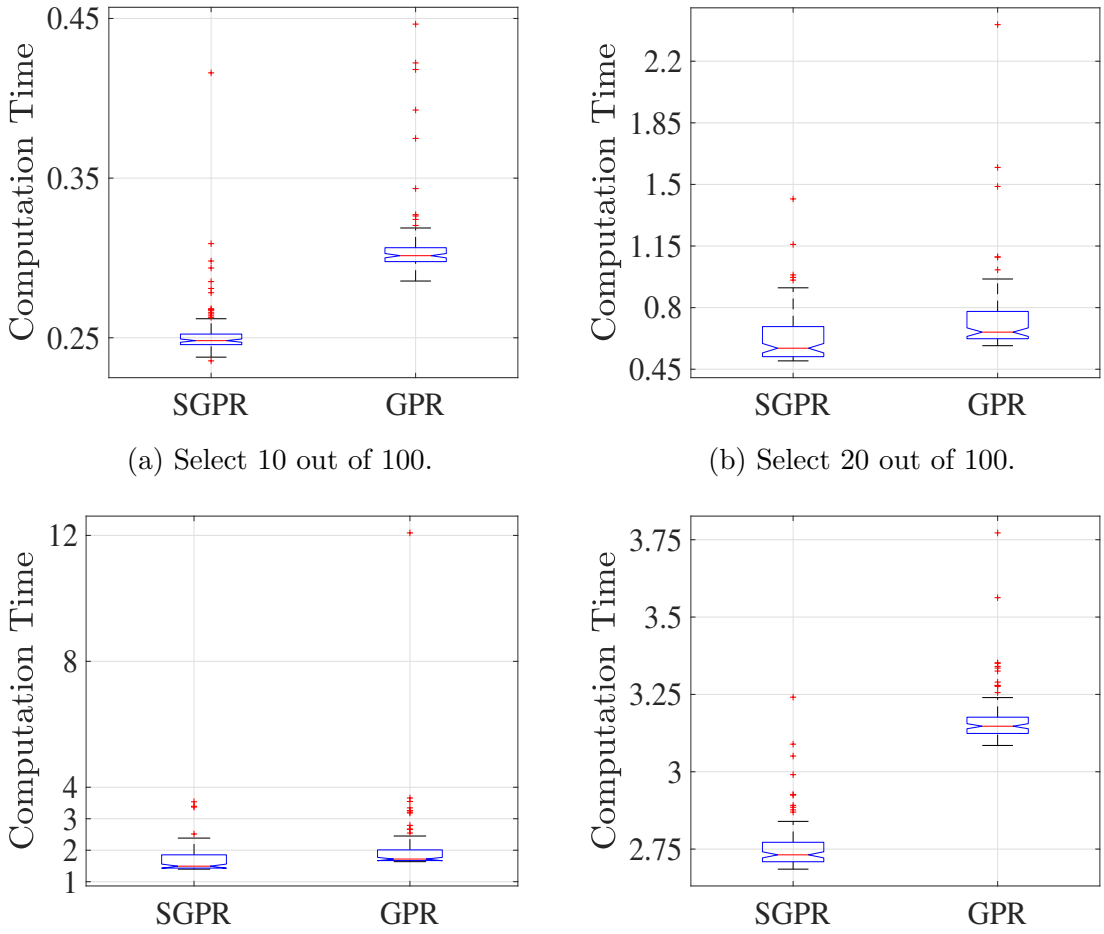
To demonstrate the performance of the proposed sequential GPR method, we conducted a Monte Carlo simulation to select different numbers of points from a series of random reference points. The reference points are generated as a series of random number which is uniformly distributed in the interval $(-2, 2)$. The first two sample points are randomly drawn from the set of reference points. Then, the proposed sequential GPR method is applied to sequentially select a number of points from the set of reference points. In the meanwhile, the conventional GPR is also applied to compare the performance with the proposed method. The values of the hyperparameters are fixed as $[1, 1, 1]$ in this example. Table 3.3 compares the mean values of the computation time from 100 runs of the Monte Carlo simulation. The computation time is recorded on a 2.7 GHz Intel Core i5 processor running MATLAB. All of the simulation results are illustrated as box plots in Fig. 3.3. The red line indicates the median values of 100 results. The top and bottom edges of the box indicate the 75th and 25th percentiles, respectively. The outliers which are represented as plus '+' marks are out of the main region of the coverage which is approximately 99.3 percent.

Regression methods	Computation time (s)			
	Select 10 out of 100	Select 20 out of 100	Select 10 out of 200	Select 20 out of 200
Conventional GPR	0.32	0.63	2.05	3.17
Sequential GPR	0.27	0.52	1.69	2.76

Table 3.3: Mean of computation time of 100 runs of Monte Carlo simulation.

3.5.3 Simulation Results

To evaluate the proposed optimal sample points selection method, it has been applied to estimated a variety of objects, such as human organs (kidney, liver), bottles and etc..



(a) Select 10 out of 100.

(b) Select 20 out of 100.

(c) Select 10 out of 200.

(d) Select 20 out of 200.

Figure 3.3: Comparison of the computation time using the conventional GPR and the proposed sequential GPR (SGPR). The unit of y axis is in second.

In this subsection, two objects with complicated shape, a rubber duck and the classic "Stanford bunny", are used to demonstrate the performance of the proposed method. We also compared the performance of the proposed method with the method which only considered the mutual information as a criterion to select sample points.

In both these two simulations, we first generate a scalar field, Υ, from a GP with a covariance function in Eq. (3.4), and let Eq. (3.1) to define the function value of such scalar field. Specifically, only the points with zero function value exist on the object surface.

At the beginning of both of these two simulations, we select the minimum and maximum values of the object surfaces on $x-$, $y-$ and $z-$ axis as six initial sample points. Based on these initial sample points, assuming the mean function value to be 0 ($\mathrm{m}(\bullet) = 0$), an estimated object surface, $\widehat{X}_o$, can then be constructed by Eq. (3.7). Only one point is supposed to be sampled on $\widehat{X}_o$ at each iteration. Since this sample point, $\hat{x}^*_{o_n}$, is not necessary on the object surface, the corresponding actual sample point, $x^*_{s_n}$, is determined as the intersection point between the actual object surface and the normal vector at $\bar{x}^*_{o_n}$. Then, such sampling process is iteratively executed until completing sampling process.

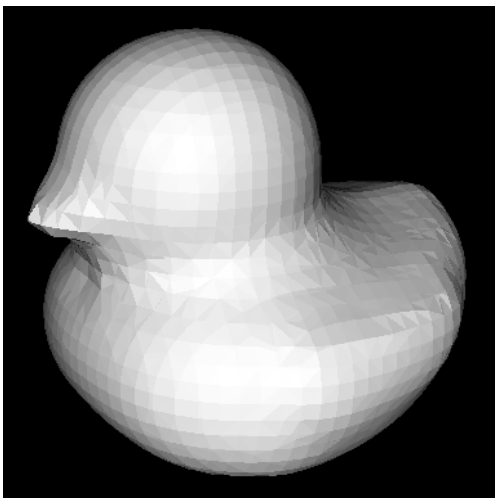


Figure 3.4: The rendering of the true shape of a rubber duck with $7,994$ reference data points.

For the rubber duck, the scalar field, Υ, is defined as $\Upsilon = \{(x, y, z) | -0.6 \leq x \leq 1, -0.7 \leq y \leq 0.7, -0.1 \leq z \leq 1.2\}$. The resolution of this field is set to be $41 \times 41 \times 41$. The point cloud of the true shape of the object that is contained within this scalar field is indicated as large dot marker '$\bullet$' in Fig. 3.4. Given the values of the hyperparameters by $\sigma_0 = 1$, $\sigma_l = 0.2$ and $\sigma^2_\varepsilon = 0.05$, the estimated shape of the object with $n_s = 50$ (0.63% of total reference data points) optimal sample points which are selected by applying the proposed method is illustrated in Fig. 3.5a. Figure 3.5b shows the left side of the estimated shape. The locations of the optimal sample points are indicated as circle marker '$\circ$' in Fig. 3.6a and Fig. 3.6b. Because we consider the curvature as a criteria to determine the location of the next sample point, the resulting sample points are intended to be selected from the high curvature regions, such as neck and mouth of the rubber duck. Consequently, the estimated shape around those areas are close to the true shape. To demonstrate

the advantages of the proposed method, we have also conducted simulations to compare the performance of the proposed method with the method which only uses the mutual information as a criterion to select the sample points. Applying the same values of the hyperparameters and selecting the same number of sample points, the resulting estimated shape of the object and the locations of the sample points are illustrated in Fig. 3.5c, Fig. 3.5d, Fig. 3.6c and Fig. 3.6d, respectively. In this case, the sample points should be selected at the locations in which can provide the maximum increase in the mutual information. In other words, the optimal sample points are most likely in the unobserved region. Hence, all the sample points are more evenly dispersed over the object. However, as shown in Fig. 3.5c and Fig. 3.5d, some geometric details of the object are lost, due to the lack of information on the high curvature areas.

In order to quantify the performance of the proposed algorithm, the Hausdorff distance [142] is used:

$$d_{\mathrm{H}}(X_o, \widehat{X}_o) = \max\{\tilde{d}_{\mathrm{H}}(X_o, \widehat{X}_o), \tilde{d}_{\mathrm{H}}(\widehat{X}_o, X_o)\}, \tag{3.35}$$

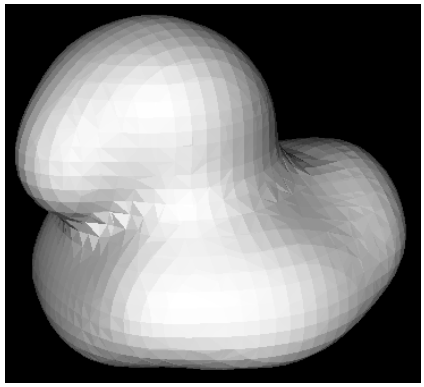
where

$$\tilde{d}_{\mathrm{H}}(X_o, \widehat{X}_o) = \sup_{x_o \in X_o} \inf_{q \in \widehat{X}_o} \|x_o - \hat{x}_o\|_2. \tag{3.36}$$

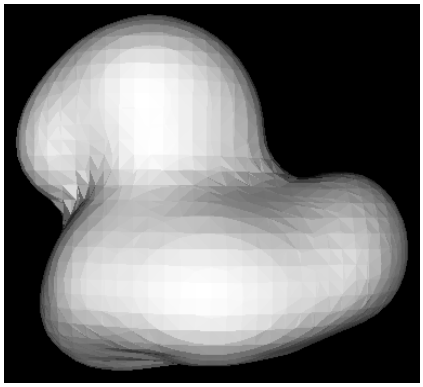
To see the effect of the hyperparamters, $\{\sigma_0, \sigma_l, \sigma_\epsilon\}$, we conducted a number of Monte Carlo simulations with different sets of hyperparameter values. Table 3.4 shows the Hausdorff distance in terms of the whole shape and that in terms of the featured areas (high curvature areas) with different sets of values of the hyperparameters.

Hyperparameters $\{\sigma_0, \sigma_l, \sigma_\epsilon^2\}$	Hausdorff Distance (d_H)			
	Whole shape		Featured areas	
	Proposed method	Using MI only	Proposed method	Using MI only
{1, 0.3, 0.05}	0.2076	0.1973	0.1506	0.1973
{1, 0.3, 0.01}	0.2118	0.1990	0.1693	0.1990
{1, 0.25, 0.05}	0.1754	0.1714	0.1383	0.1709
{1, 0.25, 0.01}	0.1846	0.1898	0.1549	0.1886
{1, 0.2, 0.05}	0.1543	0.1611	0.1054	0.1611
{1, 0.2, 0.01}	0.1732	0.1705	0.1469	0.1705

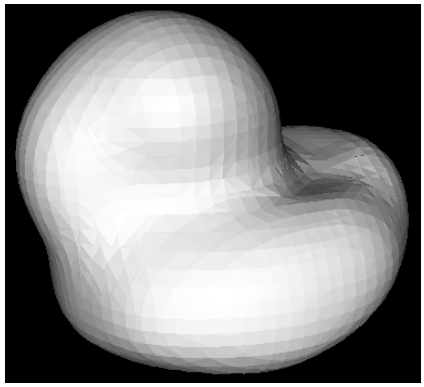
Table 3.4: Simulation results using rubber duck with different set of hyperparameters.



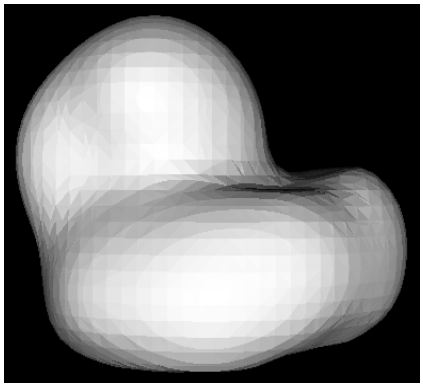
(a) Proposed method



(b) Proposed method

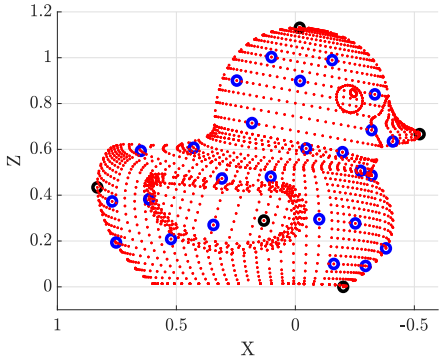


(c) Using MI only

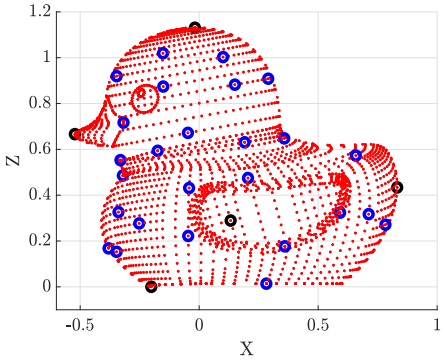


(d) Using MI only

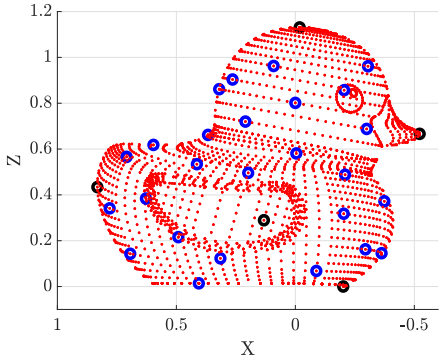
Figure 3.5: The estimated shapes of the rubber duck with only 0.63% of reference data points. Estimated shape using proposed method: (a) and (b). Estimated shape using MI only: (c) and (d).



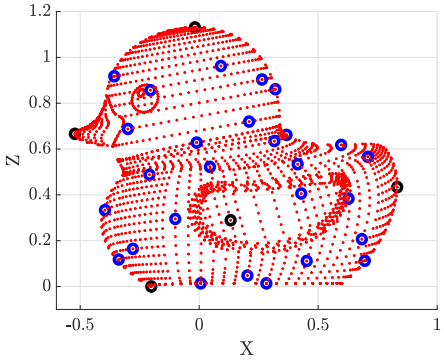
(a) Proposed method



(b) Proposed method



(c) Using MI only



(d) Using MI only

Figure 3.6: The locations of the optimal sample points selected by the proposed sample method: (a) and (b). The locations of the sample points selected based on mutual information only: (c) and (d).

The second simulation has been conducted using the Stanford bunny the true shape of which is shown in Fig. 3.7. For this case, the scalar field, Υ, is defined as $\Upsilon = \{(x, y, z)| -1.05 \leq x \leq 0.7, -0.65 \leq y \leq 0.7, 0.25 \leq z \leq 2\}$. The resolution of this field is set to be $51 \times 51 \times 51$. Given the values of the hyperparameters $\sigma_0 = 1$, $\sigma_l = 0.25$ and $\sigma_\epsilon^2 = 0.05$, and the variance of the measurement noise $\sigma_\varepsilon^2 = 0.01$, only $n_s = 50$ (0.36% of total reference data points) optimal sample points are selected by applying the proposed method and the MI only method to estimate the shape of the bunny. The locations of these optimal sample points are shown by the circle ($\circ$) markers in Fig. 3.8, and the rendering of the estimated shape of the object is also shown in Fig. 3.9, respectively. Using the mutual information as the only criterion, the main body of the bunny can be well estimated. However, the estimation of the ears is not quite accurate. Apparently, the ears of the bunny have relatively more complex geometric shapes compared to other parts of the shape. Taking the curvature into account for determination of the locations of the sample points, more sample points are selected from the ear region. As a consequence, the accuracy of the estimation of the ear region is significantly improved. If the number of the sample points is not limited, a better estimation in terms of the whole shape can be obtained by sampling more points.

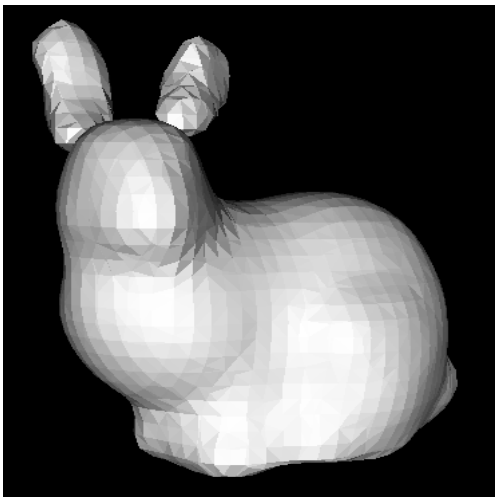
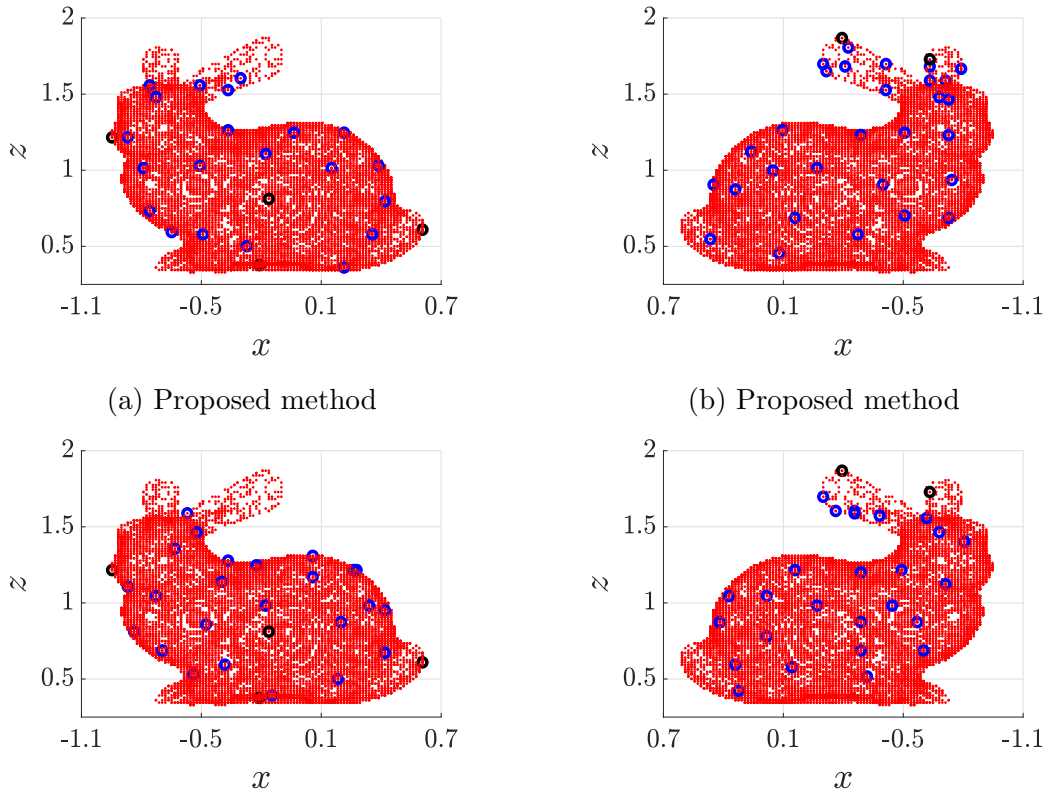


Figure 3.7: The true mesh realization of the Stanford Bunny generated from 13,775 reference data points.

In this simulation, to demonstrate the advantage of the proposed objective function in terms of the multi-fidelity, the changes of the mean of the predicted variance ν_n and that of the weighting factor α as we select each sample point are shown in Fig. 3.10. As more sample points are selected, higher confidence (i.e., lower variance) of the estimation of the

target shape is expected, and the values of ν_n and α will be reduced. Consequently, the algorithm will focus more on exploiting the local areas with high curvature rather than exploring the whole 3D shape.



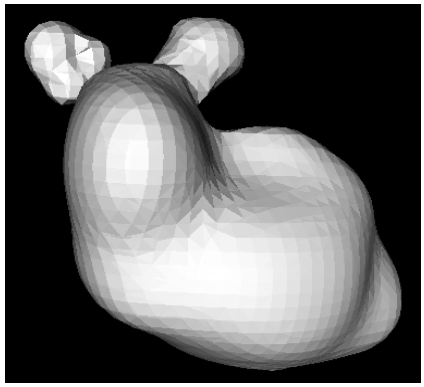
(a) Proposed method

(b) Proposed method

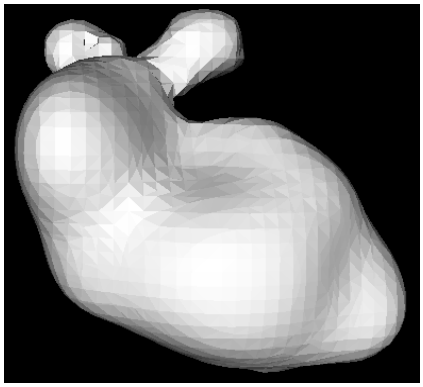
(c) Using MI only

(d) Using MI only

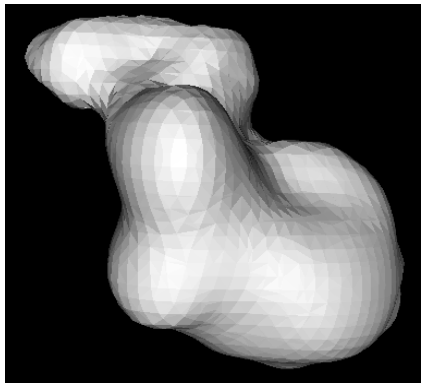
Figure 3.8: The locations of the optimal sample points selected by the proposed sample method: (a) and (b). The locations of the sample points selected based on mutual information only: (c) and (d).



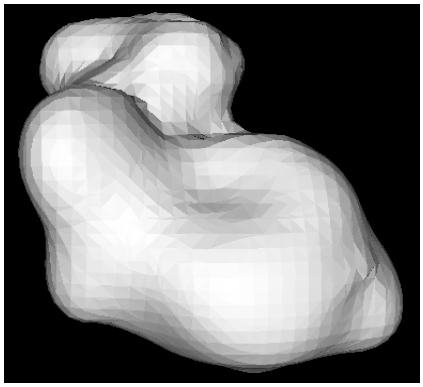
(a) Proposed method



(b) Proposed method

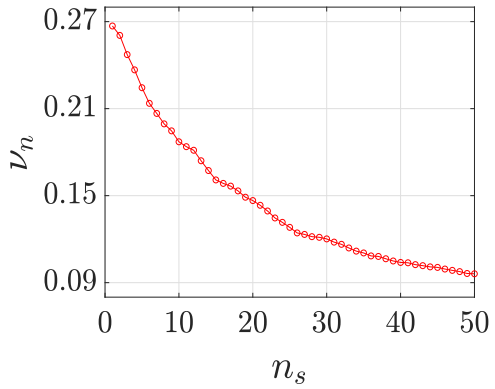


(c) Using MI only

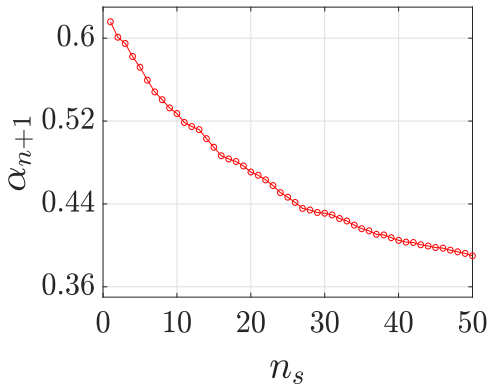


(d) Using MI only

Figure 3.9: The estimated shapes of the Stanford bunny with only 0.41% (including 6 initial sample points) of reference data points. Estimated shape using proposed method: (a) and (b). Estimated shape using MI only: (c) and (d).

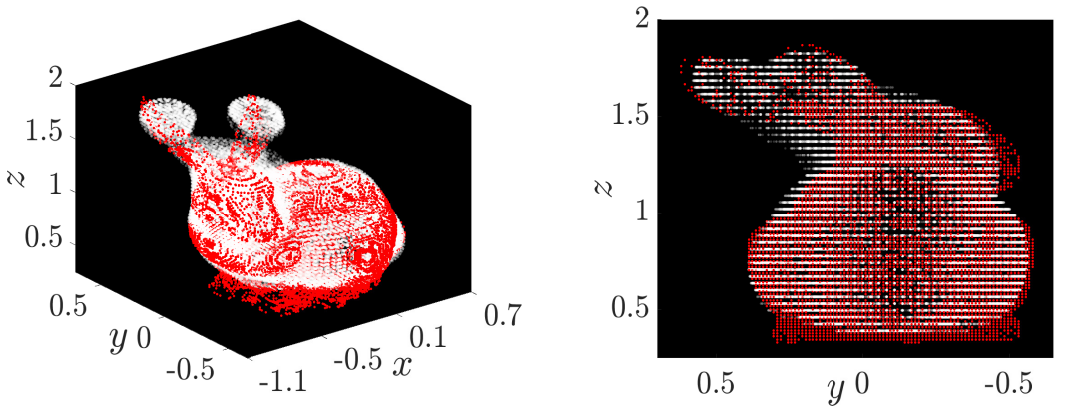


(a) Mean of predicted variance ν



(b) Weighting factor α

Figure 3.10: (a): The changes of the mean of the predicted variance, ν_n, with respect to the number of sample points. (b): The changes of the weighting factor, α, with respect to the number of sample points.



(a) Perspective view

(b) Front view

(c) Left side view

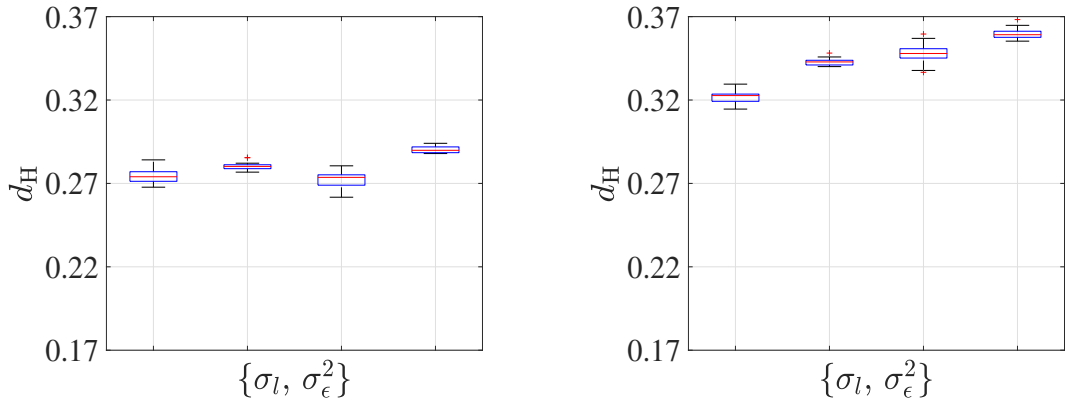
(d) Right side view

Figure 3.11: Uncertainty quantification for the Stanford bunny. Brighter spatial regions mean higher probability.

To see the effect of the hyperparamters, $\{\sigma_l,\ \sigma_\epsilon\}$ (we let $\sigma_0 = 1$ in all the Monte Carlo simulations), we also conducted a number of Monte Carlo simulations with different sets of hyperparameter values. Figure 3.12 shows the range of Hausdorff distance in terms of the whole shape and that in terms of the featured areas (high curvature areas) of 20 Monte Carlo runs using the proposed method and the conventional method. The average values of each set of runs are shown in Table 3.5.

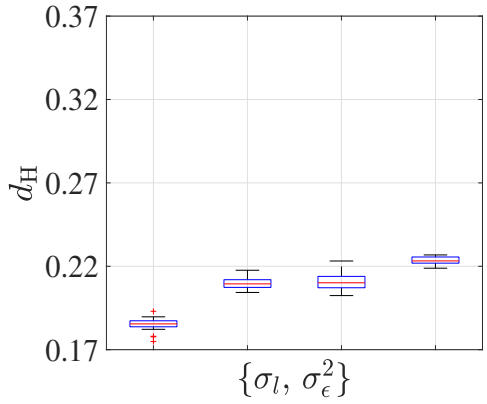
Hyperparameters $\{\sigma_0,\ \sigma_l,\ \sigma_\epsilon^2\}$	Average Hausdorff Distance (d_{H})			
	Whole shape		Featured areas	
	Proposed method	Using MI only	Proposed method	Using MI only
$\{1,\ 0.25,\ 0.05\}$	0.2732	0.3206	0.1866	0.2653
$\{1,\ 0.25,\ 0.01\}$	0.2798	0.3448	0.2105	0.2990
$\{1,\ 0.2,\ 0.05\}$	0.2726	0.3475	0.2095	0.3208
$\{1,\ 0.2,\ 0.01\}$	0.2887	0.3615	0.2231	0.3389

Table 3.5: Simulation results for the Stanford bunny: 20 Monte Carlo runs with different sets of hyperparameters.

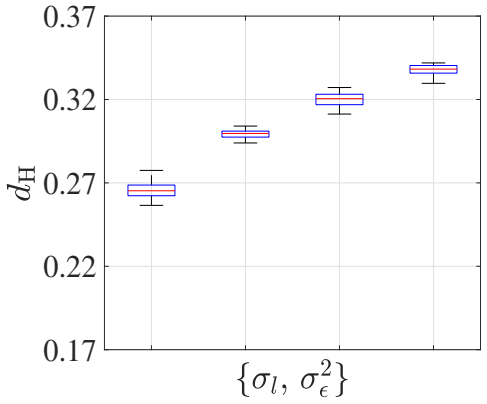


(a) Proposed method

(b) Using MI only



(c) Proposed method



(d) Using MI only

Figure 3.12: Simulation results of 20 Monte Carlo runs with different set of hyperparameters (from left to right: $\{0.25, 0.05\}$, $\{0.25, 0.01\}$, $\{0.2, 0.05\}$, $\{0.2, 0.01\}$) for the Stanford bunny. Hausdorff distance with respect to the whole shape is represented in (a) and (b), and that with respect to the featured ares is represented in (c) and (d).

3.5.4 Experimental Results

In this subsection, the proposed optimal sample points selection strategy is applied to tactile exploration using a real robot for estimating the shape of a teapot, as shown in Fig. 3.13. The hardware setup used in the experiment consists of a Barrett Whole Arm Manipulator (WAM) with 7 degree-of-freedom and a BarrettHand (BHand) equipped with tactile sensors on the fingertips and in the palm, as shown in Fig. 3.13.

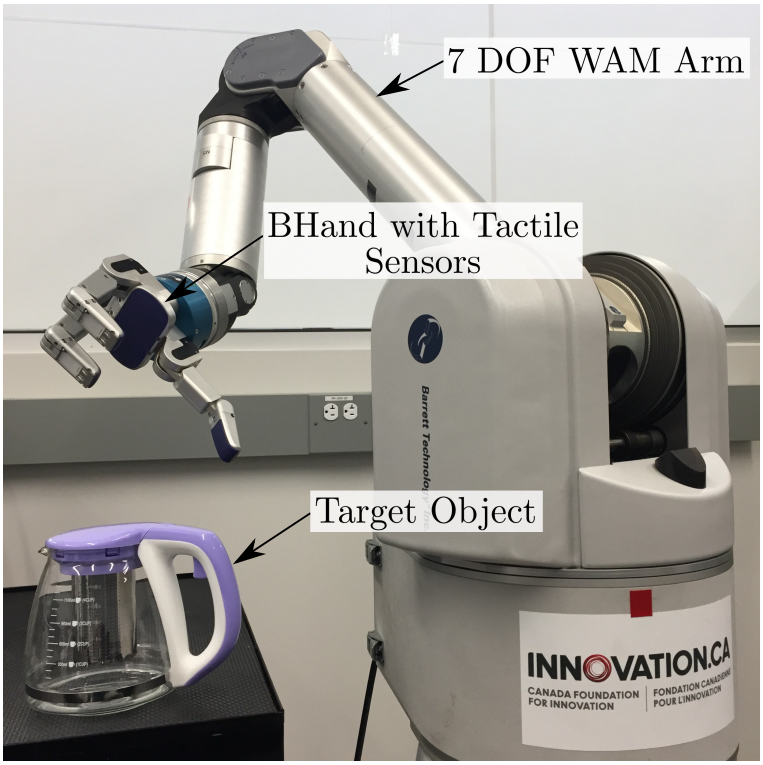


Figure 3.13: Experimental setup.

In this experiment, The teapot is placed within the workspace of the Barrett WAM. The scalar field that contains the teapot is defined as $\Upsilon = \{(x, y, z)| -0.4 \text{ m} \leq x \leq 0.16 \text{ m}, -0.7 \text{ m} \leq y \leq 0.5 \text{ m}, -0.3 \text{ m} \leq z \leq -0.15 \text{ m} \}$. We first select an initial sample point from the top of the teapot and determine the approximate height of the object. Then the WAM approaches to the teapot from positive and negative direction along x and y axis around median height and select four initial sample points. In addition, we also consider a point at the bottom of the teapot that has same x and y coordinates as the top point as one more initial sample point. Given the values of the hyperparameters $\sigma_l = 0.03$ and

$\sigma_\epsilon^2 = 0.05$, and the variance of the measurement noise $\sigma_\varepsilon^2 = 0.005$ for both proposed method and the conventional method, the locations of $n_s = 20$ sample points and the estimated shapes are illustrated in Figs. 3.15 and 3.16, respectively. Due to the complex structure of the handle part, more sample points are intended to be selected from that region using the proposed method than that using the conventional method. Note that, as introduced in Section 3.3.3, the (desired) next sample point, $\hat{x}^*_{o_{n+1}}$, may not be on the actual object surface. Figure 3.14 illustrates a scenario that $\hat{x}^*_{o_{n+1}}$ is outside the object surface. In this case, the fingertip approaches the object along the surface normal at $\hat{x}^*_{o_{n+1}}$ and takes a measurement until it touches on the object surface.

To evaluate the effect of the measurement noise, we also conducted 20 Monte Carlo runs. Figure 3.17 represents the mean of the predicted variance after selecting each sample point using the proposed method and the conventional one.

Since the target object is unknown, the model of the true shape is unavailable. As a consequence, the estimation error can not be quantified by using Hausdorff distance. In order to determine the sufficient number of sample points in terms of shape estimation, one can use Hausdorff distance to measure the difference between the estimated shape obtained from the previous step and that obtained from the current step. When the rate of change of the difference converges to a small value, the sampling process can then be terminated.

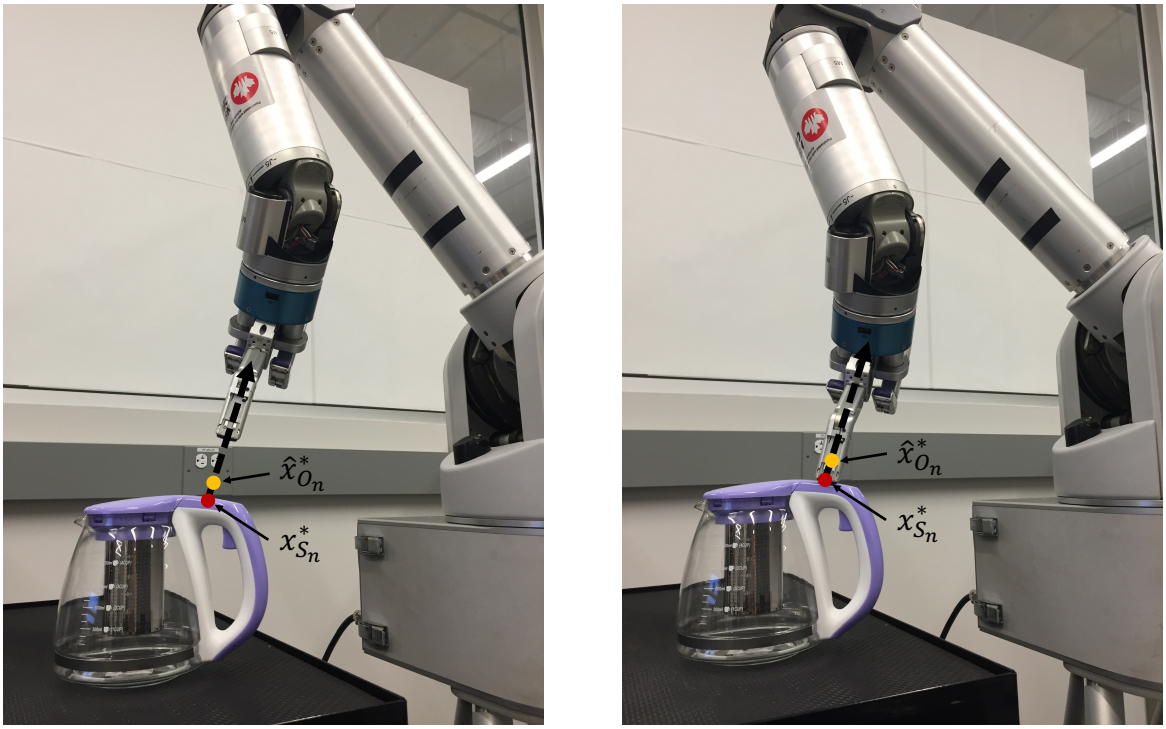
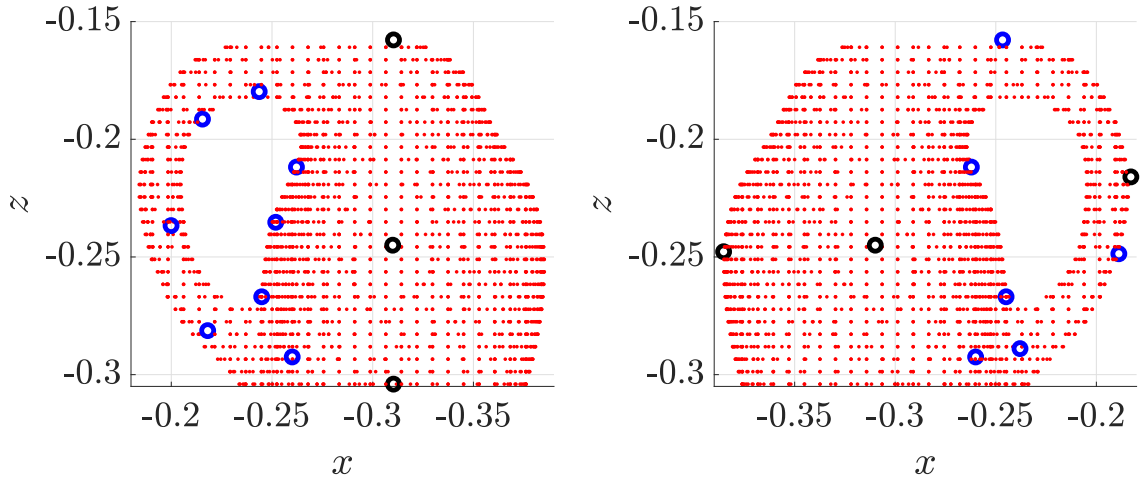
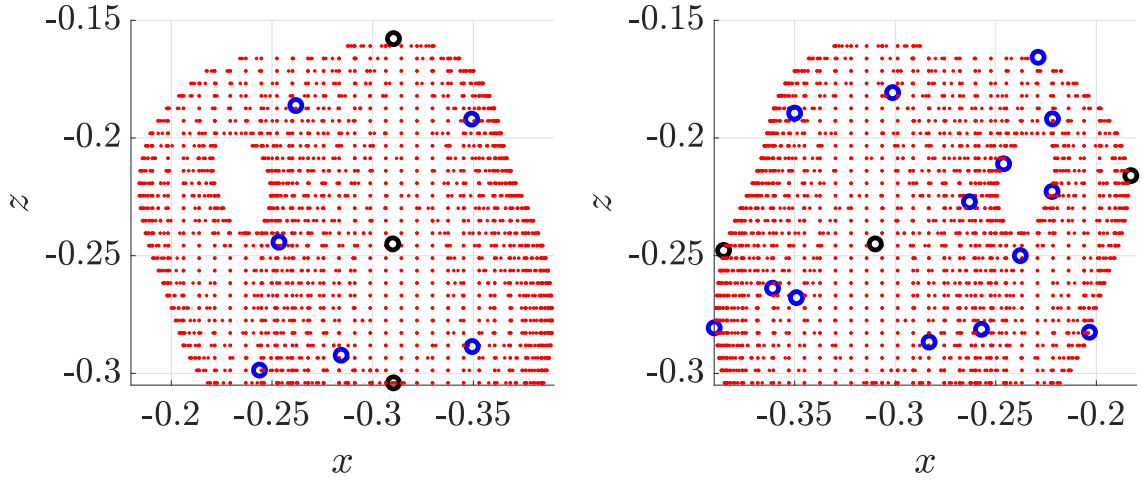


Figure 3.14: Using BarrettHand to select a sample point from the surface of the teapot.

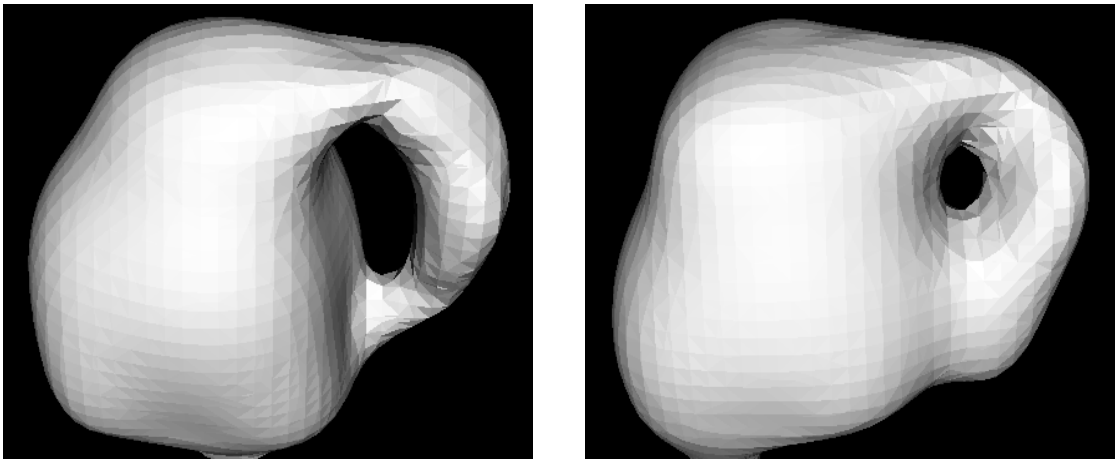


(a) Proposed method



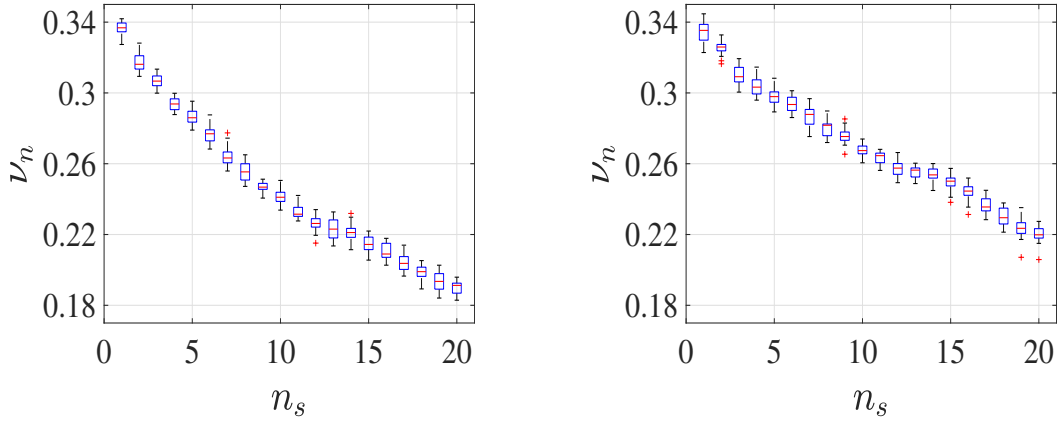
(b) Using MI only

Figure 3.15: The locations of the optimal sample points.



(a) Proposed method (b) Using MI only

Figure 3.16: The estimated shape of the teapot.



(a) Proposed method (b) Using MI only

Figure 3.17: The changes of the mean of the predicted variance with respect to the number of sample points for 20 Monte Carlo runs.

3.6 Conclusion

In this section, a novel approach to the problem of shape estimation using tactile exploration without any aid from visual or laser sensor has been presented where GPIS is used to provide an effective way to estimate the shape of an unknown object. An optimal sample point selection method is proposed to make the process of sample point selection more efficient and systematic such that the unknown shape can be estimated more accurately with minimum number of sample points. Two different criteria, mutual information and local curvature, are utilized to determine the locations of the sample points. Simulation results show that mutual information is a better criterion than variance in terms of providing information on the object shape.

By tuning the value of the weighting factor α the proposed objective function can adjust the priority between estimating the whole shape of the object and emphasizing on geometric features of the object. The proposed strategy realizes a novel multi-fidelity optimal sample points selection strategy which can provide more accurate shape estimation given the limited number of sample points available (using only less than 1% of reference data points in simulation). Simulation and experimental results demonstrate the benefit of proposed method for estimating complex geometric features more accurately than existing method (using MI as the only criterion) with the same number of sample points. Moreover, simulation results show that the proposed SGPR method can improve the computational efficiency of GPR.

Chapter 4

Knowledge-Based Grasp Planning Using Multi-Step Clustering

4.1 Introduction

In this chapter, a knowledge-based grasp planning method that takes the tasks into account is proposed to generate an appropriate grasp for a novel object which is similar to the training objects. Some training objects that have different dimensions but share similarities in both functional components and the whole shape are selected to form one object category. In addition, to improve the efficiency of grasp generation, four different hand poses are defined to accomplish specific tasks. The geometric information on the training objects and hand kinematic data are recorded via human demonstration and used to build up a training dataset. Then, a set of representative grasps are generated and form a codebook by using the proposed multi-step clustering algorithm from such training dataset. Every step of the clustering algorithm is developed based on batch learning SOM. To improve the accuracy and convergence rate of the proposed algorithm, a new initialization method is also proposed by using Gaussian kernel to measure the similarity between the training data. Moreover, the proposed clustering algorithm can automatically determine a suitable number of the clusters. Eventually, a good grasp is selected from the codebook for a novel object which is similar to the training objects to achieve a specific task, and the codebook is updated and improved with the information associated to this grasp execution.

4.2 General Framework

In this section, we briefly introduce the the basic idea of proposed approach. To focus on the grasp planning and the training data analyzing, we assume that the category to which the novel object belongs is known. In addition, the comparison between the novel object and the training objects from the same category is simplified as comparing their dimensions.

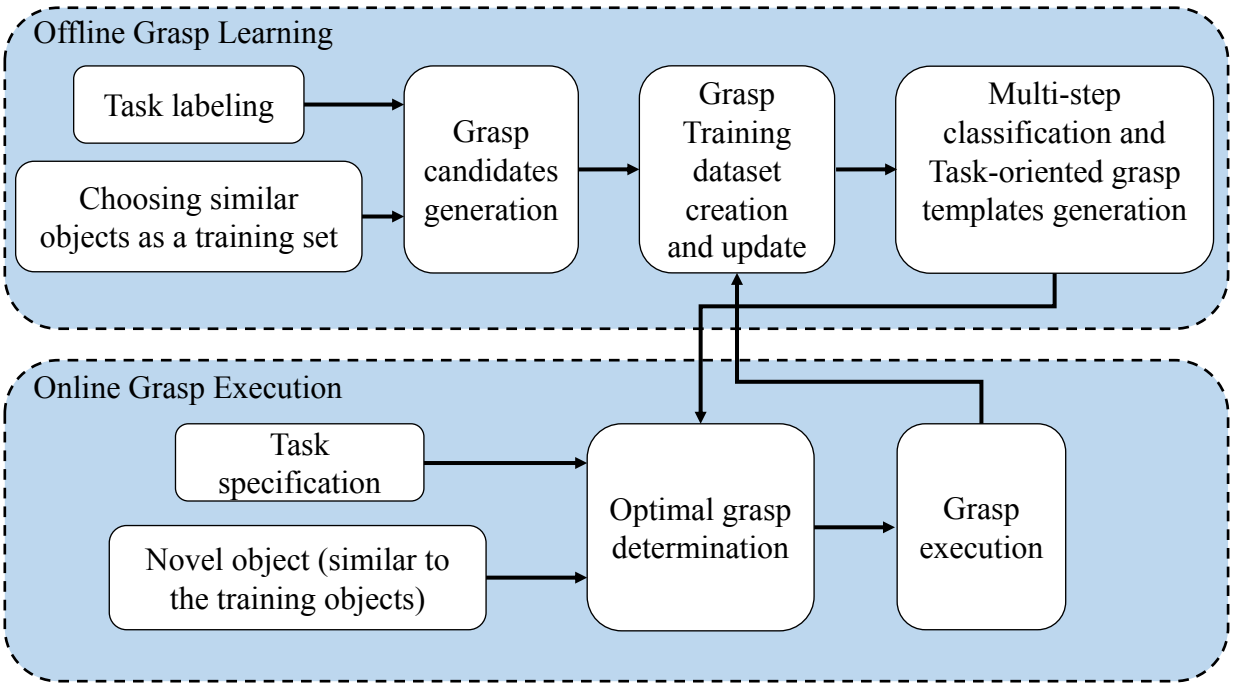


Figure 4.1: General workflow of the proposed approach.

As shown in Fig. 4.1, the approach is consisted of two parts: offline and online. In the offline, the first step is to select the training objects. The training objects should be similar in shape and can be used to accomplish the same tasks. Then, a number of task-oriented grasps are generated from each training object and recorded into a training dataset along with the geometric information on the training objects. Such training data is ran through a multi-step clustering algorithm. In the end of offline phase, the combination of the centroids that are generated in each step form a codebook which consists of a small set of representative grasps. During the online operation, comparing the shape similarity of the

target object to the training objects from the corresponding category, a representative grasp is selected from the codebook. The robot kinematic data associated to that particular grasp is utilized to the grasp execution for the target object. In the meanwhile, the codebook incorporates the geometric information on the novel object and automatically improved from such information.

4.3 Batch Learning Self-Organizing Map

This section reviews batch learning SOM that forms the basic structure of the proposed multi-step clustering algorithm in this thesis.

Let $\Psi = \{\psi_i | \psi_i \in \mathbb{R}^d\}$ represent the training dataset and each training data contained in this set is a d-dimensional vector. A finite set $\Omega = \{\omega_j | \omega_j \in \mathbb{R}^d\}$ denotes all the neurons in the competitive layer. Each neuron contains a weighting vector and has the same dimension as ψ_i. In general, there are four steps to perform batch learning algorithm: initialization, competition, cooperation and adaptation.

1. **Initialization**: In this step, all the weighting vectors contained in Ω are initialized with certain values (the details of the initialization algorithm will be introduced in Section 4.5.1).

2. **Competition**: Input the entire dataset, Ψ, to all the neurons at the same time. For each training vector, determine the closest neuron based on the Euclidean distance, i.e.,

$$\beta_i = \underset{j}{\operatorname{argmin}} \|\psi_i - \omega_j\|_2, \tag{4.1}$$

   where β_i denotes the index of the winning neuron corresponding to the training vector, ψ_i. This winning neuron will represent the corresponding training vector in the determination of the topological relation in the next step.

3. **Cooperation**: The most essential feature of SOM is that it utilizes the idea of topological neighborhood to define the relation between each neuron. There are some important properties of a topological neighborhood. The maximum of such neighborhood is at the the central neuron. It is symmetric about the associated central neuron and monotonically decreases as it gets far away from the central neuron. Specifically, the topological function of two neurons is defined as

$$h_{\beta_i,j} = \exp\left(-\frac{\|\omega_j - \omega_{\beta_i}\|_2^2}{2\sigma_h(t)^2}\right), \tag{4.2}$$

where σ_h (referred to the learning rate in some references) controls the variation rate of the correlation between two neurons. In general, σ_h decreases exponentially with time as

$$\sigma_h(t) = \sigma_h(0)\exp\left(-\frac{t-1}{t_{max}}\right), t = 1, ..., t_{max} \tag{4.3}$$

where t indicates the current iteration number, $\sigma_h(0)$ is the initial value of σ_h and t_{max} denotes the maximum number of iterations.

4. **Adaptation**: In this step, each neuron is updated through suitable adjustment based on its relation with every training vectors. Such relation is described by Eq. (4.2) in the previous step. The weight update is then expressed as

$$\omega_j^{new} = \frac{\sum_{i=1}^{N} h_{\beta_i,j}\psi_i}{\sum_{i=1}^{N} h_{\beta_i,j}}. \tag{4.4}$$

4.4 Grasp Candidates Collection

A set of objects which are in the same category but different in shape and size are selected to form the initial training objects for grasping. Using human demonstration, a number of grasp candidates are then generated for each training object to accomplish some certain tasks in experiment. Eventually, these grasp candidates are stored in a grasp training set with all training objects and specific tasks.

4.4.1 Task-Oriented Hand Poses

There are a large number of possible grasps that satisfy the *force-closure* and task conditions for a single object. However, only a few of them are used by human-beings, and those grasp can also be categorized into a small set of types [144]. Based on this idea, to improve the efficiency of the training process, we define four different hand poses to grasp different types of the objects for the BarrettHand. These hand poses, as shown in Fig. 4.2, and the corresponding target object shapes are defined as follow

- **Wrap grasp**: Two fingers are spread out, and the whole hand is fully open. This hand configuration is used to grasp the objects in the shape of box and cylinder objects from side. For instance, this hand pose can be used to grasp the main body of a mug and deliver it to another person while leaving the handle available to the person.

- **Two-finger thumb grasp**: This hand pose is generated by adjusting the wrap grasp. Two fingers are fully spread out, and the whole hand is half close. This pose is mainly used to grasp the elongated part of the object, such as the handle part of screwdrivers and that of hammers.

- **Ball grasp**: Two fingers are spread out with a certain angle, and the whole hand is fully open. This hand configuration is used to grasp the objects in the shape of ball and cylinder objects from top. For instance, this hand pose can be used to accomplish the pick-and-place task for a mug.

- **Disk grasp**: Two fingers are spread out with a certain angle, and the whole hand is half close. This hand pose is used to precisely grasp small objects in the shape of disk, such as the cap of a bottle. Moreover, it can also be used to accomplish the pick-and-place task for a bottle by grasping the bottleneck part.

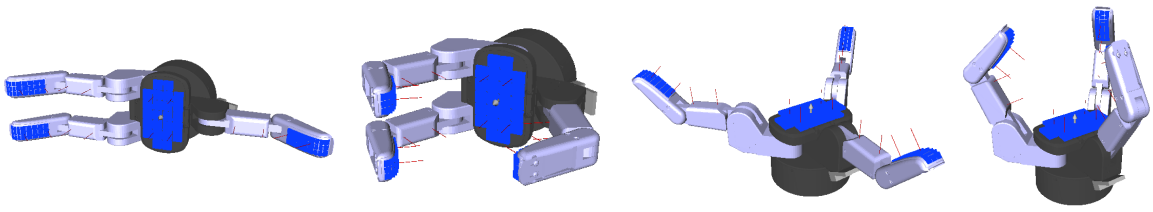


Figure 4.2: Four types of hand Poses. From left to right: wrap grasp, two-finger thumb grasp, ball grasp and disk grasp.

4.4.2 Grasp Training Set

A grasp training set, Ψ, is built to contain all the related information on the object dimension, specific tasks and the hand kinematics. Each training data vector in the set is then defined as

$$\psi_i = \left(\psi_i^{\mathcal{T}}, \psi_i^{\mathcal{D}}, \psi_i^{\mathcal{O}}, \psi_i^{\mathcal{R}}\right), \psi_i \in \Psi. \tag{4.5}$$

$\psi_i^{\mathcal{T}} \in \mathcal{T} \in \mathbb{R}^t$ denotes the desired task which belongs to a set of specific tasks learned in the experiments, such as pouring water, delivery and transportation. To encode different tasks into dataset, a series of binary numbers are used to represent each task. The value of t is determined based on the number of tasks. In addition, although some objects can be used to fulfill the same task (both bottles and mugs can be used to pour water from them), such task has to be encoded individually for each object category. $\psi_i^{\mathcal{D}} \in \mathcal{D} \in \mathbb{R}^3$ denotes the approaching direction of the manipulator when grasping the object. The approaching directions are used to represent the orientations of the objects. A finite set $\mathcal{D}$ contains a number of approaching directions that are performed in the experiments. $\psi_i^{\mathcal{O}} \in \mathcal{O} \in \mathbb{R}^3$ denotes the geometric information on the object. A set $\mathcal{O}$ contains the geometric information on all objects used for training. In this thesis, only length, width and height of the intended grasp part of the object are recorded. For instance, to complete a transportation task (i.e. pick-and-place) with a mug, only the dimensions of the main body are recorded. Such information is used to cluster the similar training objects and compare the similarity between the training objects and the target object. $\psi_i^{\mathcal{R}} \in \mathbb{R}^{17}$ denotes a hand kinematic data vector which consists of initial joint position and joint velocity of each finger, position and orientation (in quaternion) of the hand with respect to the center of the object and the strain-gauge data for each finger. Such information will be utilized to accomplish a certain grasping task.

4.5 Grasp Candidates Analysis

In this section, a multi-step classification algorithm is developed to analyze the grasp candidates and cluster similar candidates into one group. The classifier in each step is based on batch learning SOM. In addition, we also propose a novel initialization method to determine the initial neurons for each classifier. Furthermore, the proposed algorithm can automatically adjust the size of each classifier (the number of neurons) based on the information from grasping every novel objects. As a result, the entire grasp training set can be condensed into a small set of output centroids. Each centroid consists of one particular neuron from each step. The data contained in each centroid can be applied to the solution to grasping novel objects which is similar to the corresponding training objects.

4.5.1 Initialization for Clustering

As reviewed in Chapter 2, most of the existing initialization methods aim to search for such locations that they are close to the majority of data vectors and also as far as possible away from each other as initial centroids. Moreover, selecting outliers as initial centroids is not acceptable using those methods, since such outliers are normally the data corrupted by noise and far away from most of the training data vectors. In our case, however, each individual training data (i.e. grasp candidate), even the outliers, need to be considered as candidates of initial centroids, because each of them is associated with a particular object and the corresponding task. Therefore, the desired initial centroids should not only capture the density of the training data, but also take the outliers into account.

Based on the idea of the proposed optimal sample points selection method introduced in Chapter 3, we propose a novel initialization method that selects some representative training data vectors as the initial neurons. The selected neurons are determined by measuring the similarities between the training data vectors. Specifically, we use Gaussian kernel to measure the similarity, because it can produce a large response when two data vectors are close to each other and exponentially fall off when two data vectors are apart. In addition, as a nonlinear function of Euclidean distance, Gaussian kernel can also provide some useful properties such as smoothness (i.e. the Gaussian function is infinitely differentiable). Due to these features of Gaussian kernel, it is well suited for clustering problems. Gaussian kernel used in this thesis is defined as the classical squared exponential function with only one hyperparameter, $\sigma_{\mathbf{k}}$. It is given by

$$\mathbf{k}(\psi_i, \psi_j) = \exp\left(-\frac{\|\psi_i - \psi_j\|_2^2}{2\sigma_{\mathbf{k}}^2}\right). \tag{4.6}$$

This equation is very similar to Eq. (4.2). The only difference is that $\sigma_{\mathbf{k}}$ is a constant value and does not change with time. The desired neurons should be most dissimilar to each other and most similar to unselected data. Specifically,

$$\psi_i^* = \underset{\psi_i \in \Psi \setminus \Psi^*}{\operatorname{argmin}} \, \Xi_i \tag{4.7}$$

where

$$\Xi_i = \frac{\frac{1}{J}\sum_{j=1}^{J} \mathbf{k}(\psi_i, \psi_j^*)}{\frac{1}{N-J-1}\sum_{l=1}^{N-J-1} \mathbf{k}(\psi_i, \psi_l)}, \tag{4.8}$$

where ψ_i^* denotes the i-th optimal neuron, $\Psi = \{\psi_1, ..., \psi_N\}$ represents the original training dataset, $\Psi^* = \{\psi_1^*, ..., \psi_J^*\}$ represents all selected neurons, and $\Psi \backslash \Psi^*$ denotes the set of all unselected training data vectors. The details of the implementation of the proposed initialization method to select a total of K neurons are shown in Table 4.1.

training:	(1) Original dataset, $\Psi = \{\psi_1, ..., \psi_N\}$ (2) Number of initial neurons, K
Output:	K optimal initial neurons, $\Psi^* = \{\psi_1^*, ..., \psi_K^*\}$
1:	**if** $j = 1$ (i.e. the first neuron) **do**
2:	**for** every unselected data training, ψ_i, **do**
3:	**Calculate** $\frac{1}{N-1} \sum_{l=1}^{N-1} \mathbf{k}(\psi_i, \psi_l)$
4:	**end for**
5:	**Determine** the optimal neuron, ψ_i^*, as
6:	$\psi_i^* = \text{argmin}_{\psi_i \in \Psi \backslash \Psi^*} \frac{1}{\frac{1}{N-J-1} \sum_{l=1}^{N-J-1} \mathbf{k}(\psi_i, \psi_l)}$
7:	**else**
8:	**for** every unselected training data vector, ψ_i, **do**
9:	**Calculate** Ξ_i by Eq. (4.14)
10:	**end for**
11:	**Determine** the optimal neuron, ψ_i^*, by Eq. (4.7)
12:	**end if**
13:	**Update** $\Psi^* = \Psi^* \cup \psi_i^*$
14:	**Stop**, if K neurons are selected. Otherwise, back to Step 1.

Table 4.1: Proposed initialization method

4.5.2 Multi-Step Classification

Grasping is the most fundamental skill that allows human to interact environment and manipulate tools to achieve desired tasks. However, there are still some factors influencing the human grasp choice and the corresponding action. Among these factors, there are four major ones which characterize human grasping the most [24]:

- **Desired Tasks**: The tasks need to be accomplished, such as pour water from some objects.

- **Object types**: The geometric information on the target object.
- **Object poses**: The position and orientation of the target object.
- **Grasp types**: An appropriate task-oriented hand pose and the corresponding motion of the hand.

Based on these criteria, we develop a multi-step clustering algorithm which consists of three connected steps to cluster the grasp candidates into a small set of groups. Assuming the shape of the target object and its corresponding category are known, each individual step corresponds to one of the factors listed above. The training data vectors represent the grasp candidates collected from the experiments and are defined by Eq. (4.5). Because these vectors are clustered based on different criteria in each step, the number of the clusters (i.e. the number of neurons) in each step are different.

In the first step, the training data vectors, Ψ (indicated in grey in Fig. 4.3), are clustered into K_t groups (indicated in orange in Fig. 4.3) based on the specific tasks. Specifically, only $\psi_i^{\mathcal{T}}$ is considered as the training data in this layer. The value of K_t is defined as the number of tasks learned in the experiments. At the beginning of this step, K_t initial neurons, $\omega_{k_t}^{\mathcal{T}}$ $(k_t = 1, ..., K_t)$, are selected using the proposed initialization method. The value of $\omega_{k_t}^{\mathcal{T}}$ is then updated using batch learning SOM which is introduced in Section 4.3.

In the second step, each group from the previous step is further clustered into $K_d^{k_t}$ groups (indicated in yellow in Fig. 4.3) based on the approaching direction, $\psi_i^{\mathcal{D}}$. The value of $K_d^{k_t}$ depends on the number of data vectors which are centered around $\omega_{k_t}^{\mathcal{T}}$. One popular method to determine the number of clusters based on the number of data in statistics is the 2^K rule which is given by

$$2^{K_d^{k_t}} \geq N_{k_t}, \tag{4.9}$$

where N_{k_t} denotes the total number of training vectors surround $\omega_{k_t}^{\mathcal{T}}$. Then, the value of $K_d^{k_t}$ can be determined by

$$K_d^{k_t} = \lceil \log N_{k_t} \rceil. \tag{4.10}$$

According to the 2^k rule, we then use the proposed method to select $K_d^{k_t}$ initial neurons $\omega_{k_t,k_d^{k_t}}^{\mathcal{D}}$ in each group. The value of $K_d^{k_t}$ will be further adjusted based on results from batch learning SOM.

Following the same procedure stated in the second step, each group from the second step is again clustered into $K_o^{k_t,k_d} r$ groups (indicated in blue in Fig. 4.3) based on the geometric information on objects, $\psi_i^{\mathcal{O}}$, and the hand kinematic data, $\psi_i^{\mathcal{R}}$. Eventually, the output of clustering is a set of centroids and each centroid consists of a series of connected neurons

$$\Omega = (\omega_{k_t}^{\mathcal{T}}, \omega_{k_t,k_d}^{\mathcal{D}}, \omega_{k_t,k_d,k_{or}}^{\mathcal{OR}}), \tag{4.11}$$

where

$$\begin{aligned} k_t &= 1, ..., K_t, \\ k_d &= 1, ..., K_d^{k_t}, \\ k_{or} &= 1, ..., K_{or}^{k_t,k_d}. \end{aligned}$$

$\omega_{k_t,k_d,k_{or}}^{\mathcal{OR}}$ is the combination of $\omega_{k_t,k_d,k_{or}}^{\mathcal{O}}$ and $\omega_{k_t,k_d,k_{or}}^{\mathcal{R}}$. $\omega_{k_t,k_d,k_{or}}^{\mathcal{O}}$ will be used as a decision to determine a suitable grasp, $\omega_{k_t,k_d,k_{or}}^{\mathcal{R}}$, for a novel object. Ω is then written into a codebook for the online grasp generation and execution. The workflow of the proposed multi-step classification is illustrated in Fig. 4.3.

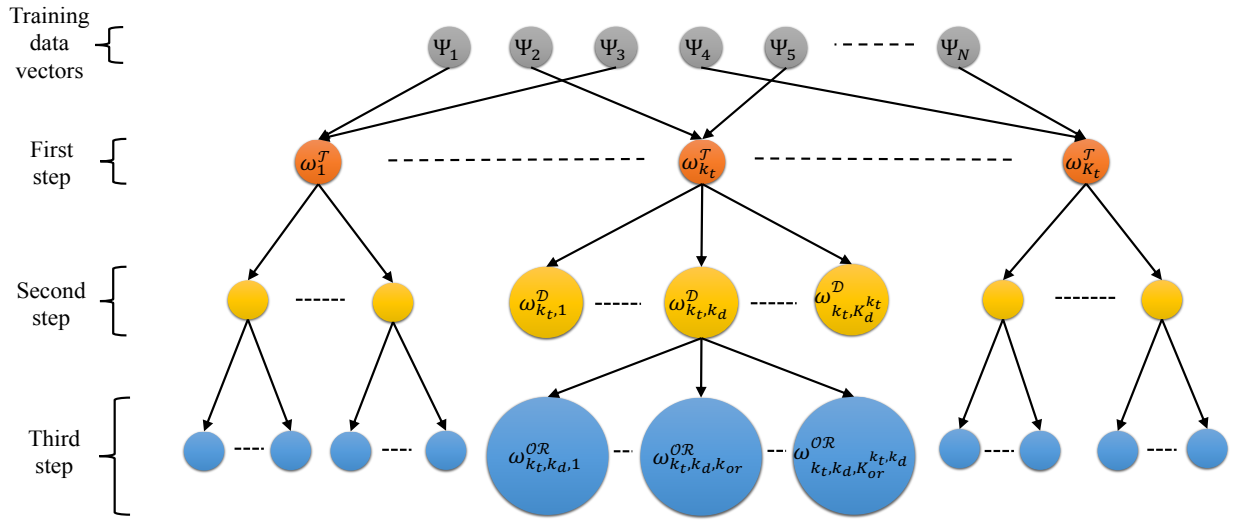


Figure 4.3: The workflow of the proposed multi-step clustering algorithm. The training data vectors represent the grasp candidates collected from the experiments. Each step is a SOM, and the neurons in different step are associated with a different piece of the training vectors. Eventually, each output centroid consists of one particular neuron from each step.

4.5.3 Determination of Cluster Number

As mentioned in the precious section, the numbers of the initial neurons in the second and third steps are determined based on the 2^K rule. Although it is an efficient method to be implemented, the number of the clusters may not be optimal for analyzing the data. Moreover, the manipulators should keep improving the grasping skill as it interacts with more and more objects. In other words, the training set should constantly adapt and update with the related information from each grasp execution.

In this thesis, we propose an auto-growing algorithm to address the issue of identifying the an appropriate number of clusters. A number of initial neurons are first selected based on the 2^K rule. Once these neurons are updated using batch learning SOM, the quality of classification is measured by the distortion function for each cluster. The distortion function is given by

$$\mathbb{J}_j = \sum_{i}^{N} 1\{\beta_i = j\}\|\psi_i - \omega_j\|_2^2, j = 1, ..., K, \tag{4.12}$$

where $1\{\bullet\}$ is defined as an indicator function. Specifically, $1\{\bullet\} = 1$ if the argument inside is true, and 0 otherwise, i.e. $1\{True\} = 1$ and $1\{False\} = 0$. For instance, $1\{2 = 2\} = 1$ and $1\{5 = 3\} = 0$. Then, let $\underline{\iota}$ denote the mean threshold value, and let $\bar{\iota}$ denote the max threshold values. The procedure to insert a neuron into or remove a neuron from the clustering algorithm is shown in Table 4.2.

1:	**for** every cluster, $j = 1$ to K, **do**
2:	**if** any$(1\{\beta_i = j\}\|\psi_i - \omega_j\|_2^2 > \bar{\iota})$ **do**
3:	**Determine** $\beta_{insert} = \text{argmax}_{i \in N}\, 1\{\beta_i = j\}\|\psi_i - \omega_j\|_2^2$
4:	**Insert** $\psi_{\beta_{insert}}$ as an addition neuron to the clustering algorithm
5:	**elseif** all$(1\{\beta_i = j\}\|\psi_i - \omega_j\|_2^2 < \bar{\iota})$ & $\mathbb{J}_j > \underline{\iota}$ **do**
6:	**Remove** ψ_j
7:	**end if**
8:	**end for**

Table 4.2: Proposed auto-growing clustering algorithm.

4.6 Grasp Selection and Dataset Update

Using the codebook generated from the proposed clustering algorithm, the process of generating a task-oriented grasp for a novel object is presented in this section.

4.6.1 Task-Oriented Grasp Selection

We assume the shape of the novel object is known and can be simply represented by length, width and height. The required input data vector consists of a specific task, desired approaching direction of the hand and the geometric information of the object (length, width and height) and is defined as

$$\Psi_{novel} = \{\psi^{\mathcal{T}}_{novel}, \psi^{\mathcal{D}}_{novel}, \psi^{\mathcal{O}}_{novel}\}. \tag{4.13}$$

The process of selecting a suitable grasp from the codebook involves three steps of decision making, as shown in Fig. 4.4. Each decision is made by comparing a particular data in Ψ_{novel} with the neurons in the corresponding step. Eventually, the group that Ψ_{novel} belongs to can be determined. Moreover, the robotic kinematic data contained in the neuron of that group is retrieved and applied to grasp the target object.

4.6.2 Training Dataset Update

The codebook can be updated from each time grasping a novel object. If a novel object can be successfully grasped by the selected grasp from the codebook, the data contained in Ψ_{novel} will be stored into the training dataset. In addition, the updated training dataset is ran through the proposed multi-step clustering algorithm. As a result, the values contained in each neuron are refined. In other words, the relation between each representative grasp and the geometrics of its corresponding object is enhanced. In the case that the selected grasp is failed to grasp the novel object, human assistance is required. The grasp is then inputted to the training dataset so that the codebook can be improved from the experience of failure.

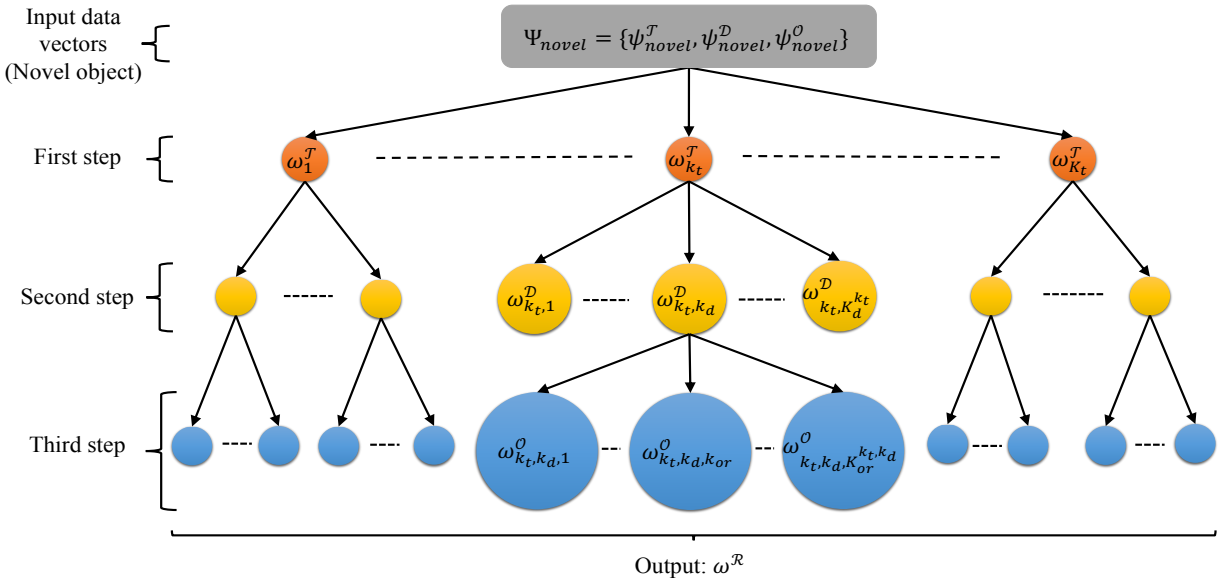


Figure 4.4: The process of selecting a suitable grasp for a novel object.

4.7 Simulation and Experimental Results

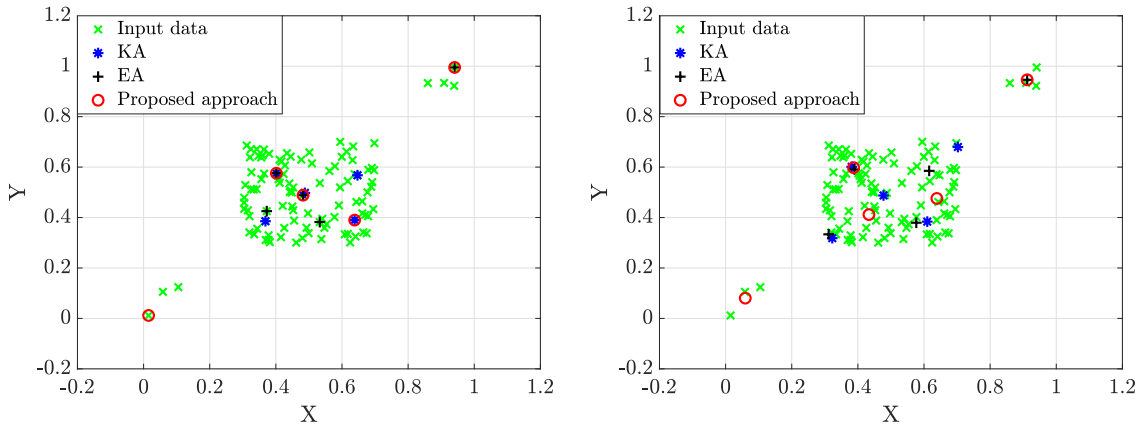
In this section, the performance of the proposed initialization method is first validated by comparing with the Kaufman approach in Section 4.7.1. The advantages of using Gaussian kernel rather than Euclidean distance to measure the similarity between data is also demonstrated within this section. The proposed auto-growing method is then simulated and compared with the density-based approach in Section 4.7.2. In order to demonstrate the performance of the proposed knowledge-based grasp planning framework, we train our manipulator (Barrett WAM equipped with BHand) to learn how to grasp a mug to achieve some desired tasks from experience. The robotic kinematic data is gathered from human manipulation assistance while performing two specific tasks (transportation and delivery/pouring water) with a set of different mugs. Such data is then recorded with the dimensions of the training mugs and the tasks to build up a training dataset. The details of how the training data is collected and prepared are presented in Section 4.7.3. Finally, the experiment results of running the proposed multi-step clustering algorithm on the training dataset for generating a suitable grasp for three new mugs are presented in Section 4.7.4.

4.7.1 Comparing of Different Initialization Methods

To demonstrate the advantages of the proposed initialization method, we have conducted two simulations with different 2D datasets to compare the performance of the proposed method with the Kaufman approach (KA) method and a Euclidean distance based approach (EA). Both datasets consist of one or more regions with high data density and some sparse outliers. The Euclidean distance based approach just simply replaces Gaussian kernel with l_2 norm in Eq (4.14) as

$$\Xi_{i,Euclidean} = \frac{\frac{1}{J}\sum_{j=1}^{J} \|\psi_i - \psi_j^*\|_2}{\frac{1}{N-J-1}\sum_{l=1}^{N-J-1} \|\psi_i - \psi_l\|_2}, \tag{4.14}$$

and selects the initial centroids by Eq (4.7).



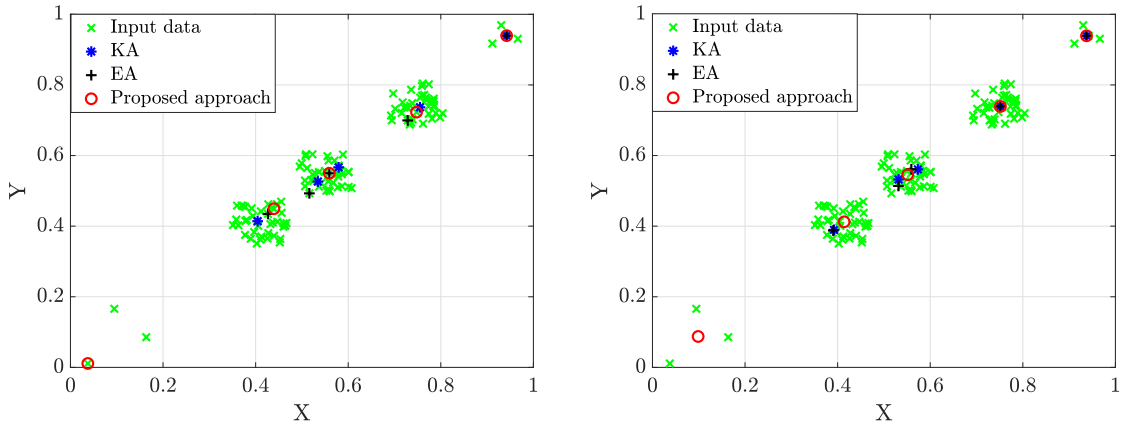
(a) Initial centroids (b) Updated centroids from 2 iterations

Figure 4.5: Comparison of different initialization methods using test dataset 1.

In the first dataset, 100 data points are uniformly distributed within a 2D space ($0.3 \leq x \leq 0.7$, $0.3 \leq y \leq 0.7$) and additional seven data points are located in two different clusters far away from the majority of the data points. All these input data points are indicated as the cross ('$\times$') markers in Fig. 4.5. Following the procedure of KA introduced in Chapter 2, the selected starting centroids using KA are indicated as the asterisk '$*$' markers in Fig. 4.5. This result shows that KA can provide a good performance in terms of accurate determination of the region with high data density. The outliers, however, are completely ignored. The initial centroids selected using EA are indicated as the plus

('+') markers. One of the clusters contained sparse outliers is captured and yet the other one is omitted. On the contrary, when $\sigma_{\mathbf{k}} = 1$, both clusters contained sparse outliers are well captured using the proposed approach (selected centroids are indicated as the circle 'o' markers). Figure 4.5b shows the updated centroids from 2 iterations of running batch learning method.

In the second dataset, 3 clusters are created and each of them contains 100 uniformly distributed data points. In addition, seven sparse outliers are randomly selected from two regions which are away from the clusters. Given $\sigma_{\mathbf{k}} = 1$, the simulation result further demonstrates that the proposed approach is capable of capturing the regions with the outliers, as shown in Fig. 4.7.

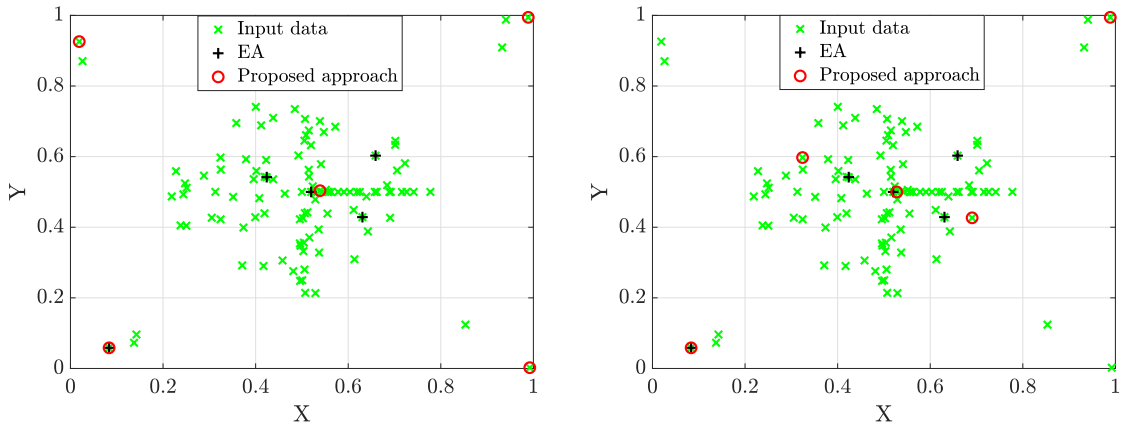


(a) Initial centroids

(b) Updated centroids from 2 iterations

Figure 4.6: Comparison of different initialization methods using test dataset 2.

Moreover, to identify the sensitivity of the proposed approach to the value of $\sigma_{\mathbf{k}}$, we create another dataset which consists of 100 data points uniformly distributed in the center region of a 2D space and 10 sparse outliers located in four different regions. By setting $\sigma_{\mathbf{k}} = 0.3$, all the regions with outliers can be well captured. However, when let $\sigma_{\mathbf{k}} = 0.5$, only two out of four regions can be captured. This result implies that in order to identify all the regions with outliers, we have to carefully tune the value of $\sigma_{\mathbf{k}}$.



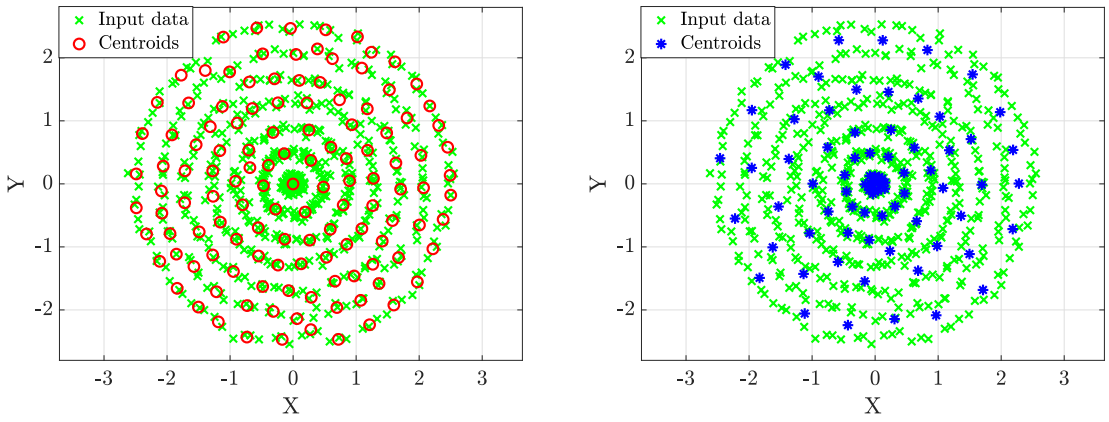
(a) $\sigma_{\mathbf{k}} = 0.3$ (b) $\sigma_{\mathbf{k}} = 0.5$

Figure 4.7: Simulation results by setting different values of $\sigma_{\mathbf{k}}$.

4.7.2 Comparing of Different Growing Methods

To evaluate the performance of the proposed auto-growing algorithm, the proposed algorithm is applied to cluster two synthetic datasets, a multiply concentric circles dataset and a flower shape dataset, while satisfying two different conditions. In the simulations with both dataset, the selected centroids are refined by running through batch learning 3 times in each iteration.

In the first dataset, 700 data points are uniformly distributed in seven concentered circles, and each circle contains 100 data points. The data points are indicated as the cross ('×') markers in Fig. 4.8 and Fig. 4.9. The growing process starts with one initial centroid which is selected using the proposed initialization method. Given $\underline{\iota} = 0.2$ and $\bar{\iota} = 0.3$, 126 centroids are selected in 128 iterations using the proposed auto-growing algorithm, as shown in Fig. 4.8a. The distribution of the selected centroids at different stage in the growing process are illustrated in Fig. 4.9.To demonstrate the advantages of the proposed method, a density-based approach introduced in Chapter 2 (i.e., insert a centroid in the region with relative high data density and remove a centroid from the region with relative low data density) is also applied to cluster the same dataset. By running the density-based approach with the same number of iteration as the proposed method, only 78 centroids are selected and neither of two conditions are satisfied. As illustrated in Fig. 4.8b, most of the centroids are selected from the center of the space in which the majority of data points locate as expected.



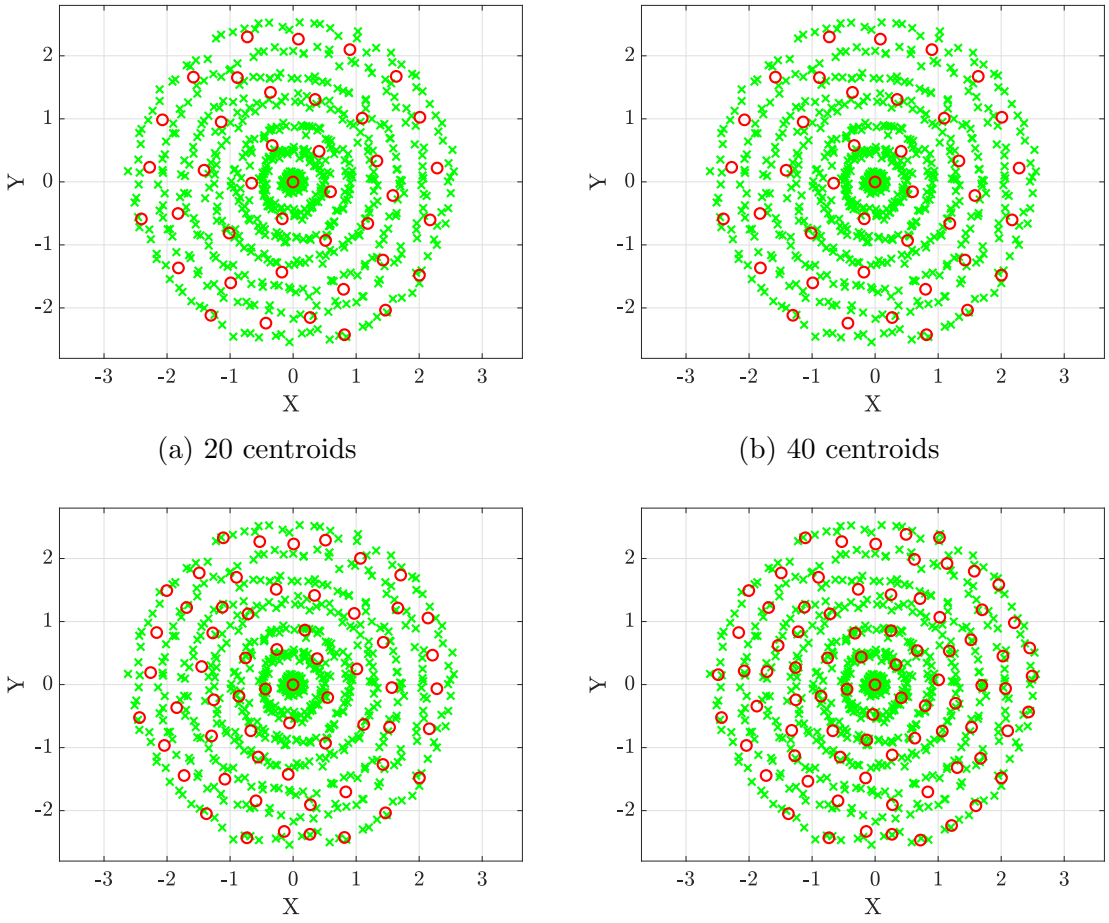
(a) Proposed approach (b) Density-based approach

Figure 4.8: Comparison of different auto-growing methods with multiply concentric circles dataset.

Since the dataset is randomly generated, we conduct a number of Monte Carlo simulations for the input data points. In the Monte Carlo simulation, we run both the density-based method and the proposed method 50 iterations. In the meanwhile, the mean of distortion function value of each cluster and the mean of the maximum Euclidean distance between an input data point and its corresponding centroid are recorded. Table 4.3 compares the performance of these two approaches from 100 runs of the Monte Carlo simulation. All of the simulation results are illustrated as box plots in Fig. 4.10.

	Mean of Distortion Function values	Maximum of Euclidean Distance
Density-based approach	0.4627	0.8513
Proposed approach	0.2851	0.3919

Table 4.3: Mean of computation time of 100 runs of Monte Carlo simulation.



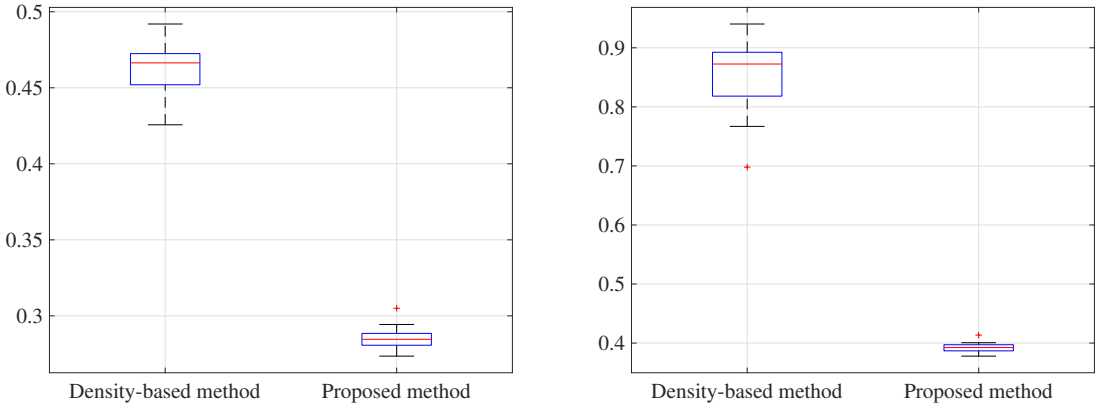
(a) 20 centroids

(b) 40 centroids

(c) 60 centroids

(d) 80 centroids

Figure 4.9: Different stage in the growing process.



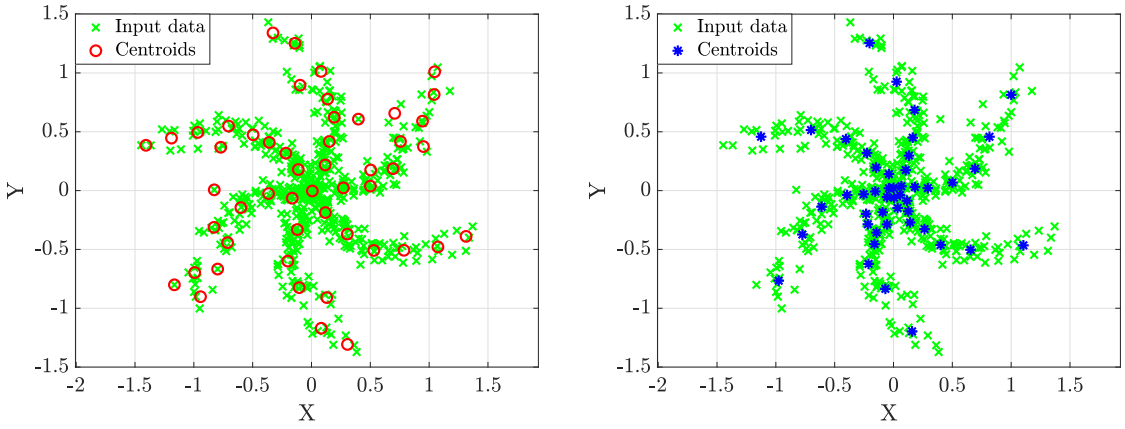
(a) Mean of distortion function values (b) Maximum of Euclidean distance

Figure 4.10: Simulation results of 100 Monte Carlo runs with different input data points.

In the second dataset, 600 data points are randomly distributed to form a flower shape. Given $\underline{\iota} = 0.1$ and $\bar{\iota} = 0.2$, 51 centroids are selected in 57 iterations using the proposed auto-growing algorithm, as shown in Fig. 4.11a. The distribution of the selected centroids at different stage in the growing process are illustrated in Fig. 4.13. As shown in Fig. 4.11b, by running the density-based approach with same number of iteration, 47 centroids are selected. We also conduct a number of Monte Carlo simulations for the input data points and run both method 50 iterations. Table 4.4 compares the performance of these two approaches from 100 runs of the Monte Carlo simulation. All of the simulation results are illustrated as box plots in Fig. 4.12.

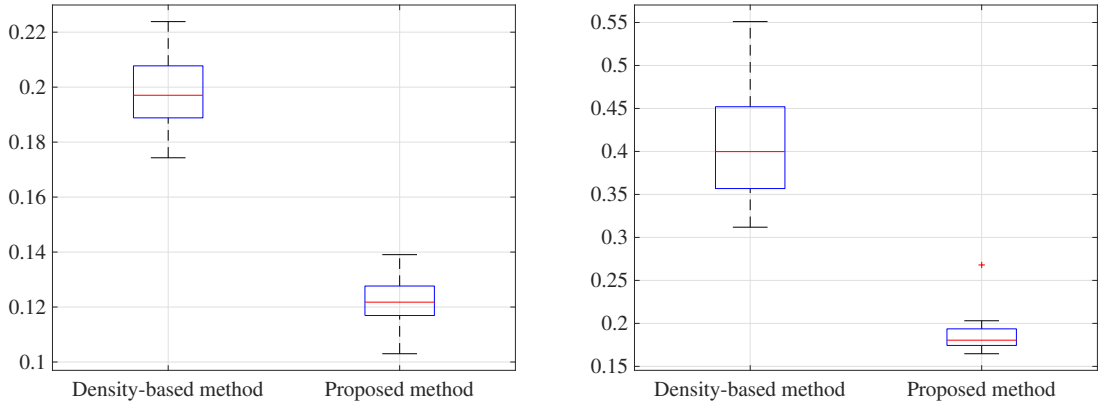
	Mean of Distortion Function values	Maximum of Euclidean Distance
Density-based approach	0.1982	0.4065
Proposed approach	0.1218	0.1866

Table 4.4: Mean of computation time of 100 runs of Monte Carlo simulation.



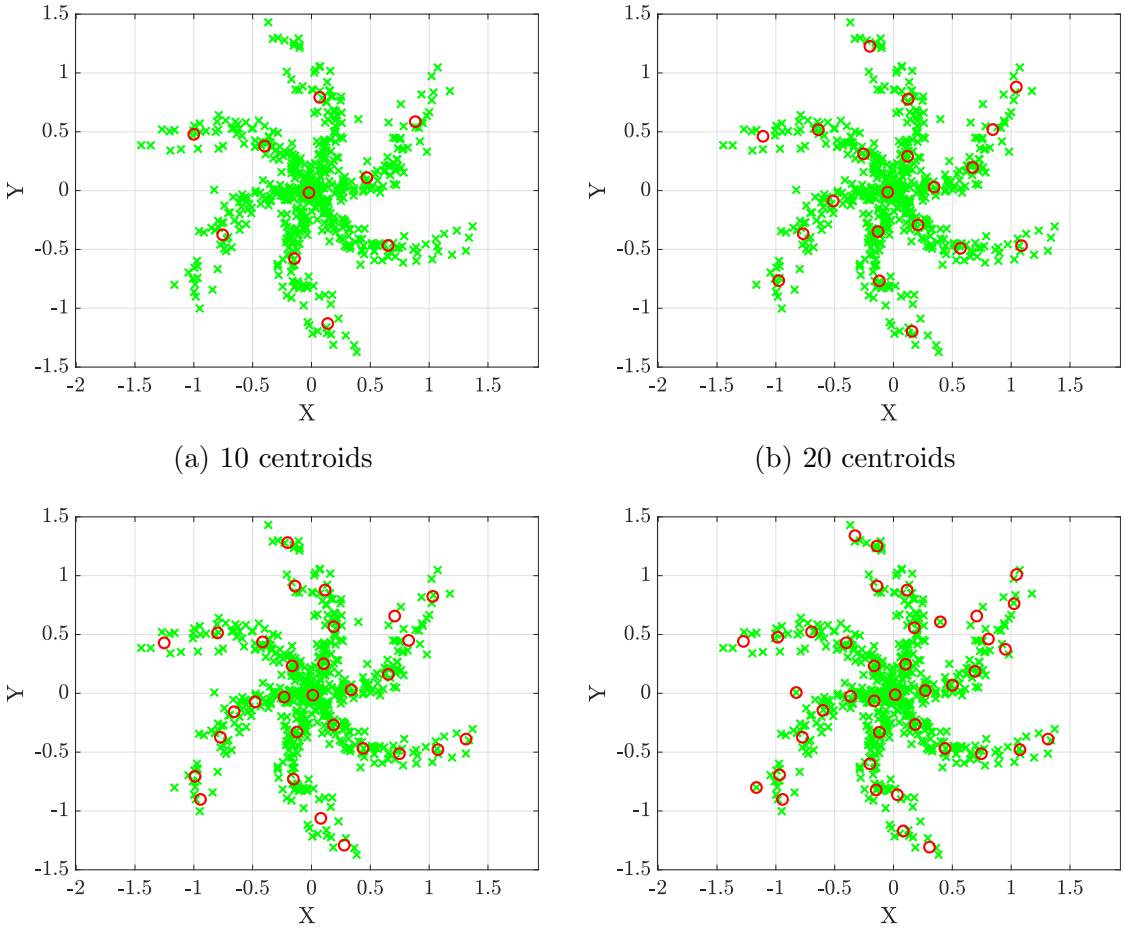
(a) Proposed approach (b) Density-based approach

Figure 4.11: Comparison of different auto-growing methods with a flower shape dataset.



(a) Mean of distortion function values (b) Maximum of Euclidean distance

Figure 4.12: Simulation results of 100 Monte Carlo runs with different input data points.



(a) 10 centroids

(b) 20 centroids

(c) 30 centroids

(d) 40 centroids

Figure 4.13: Different stage in the growing process.

4.7.3 Experimental Setup

The proposed approach is tested on a Barrett WAM with a three-fingered Barrett Hand, the same manipulator and robotic hand we used in the tactile exploration. We consider the case where the target object is located on a flat surface and within the operation range of WAM. We consider a mug category which consists of eight different mugs varying in shape and size.The mugs used for the training in the experiment are illustrated in Fig. 4.14. The dimensions of the main body of each mug is listed in Table 4.5.

Figure 4.14: Training objects used in physical experiments.

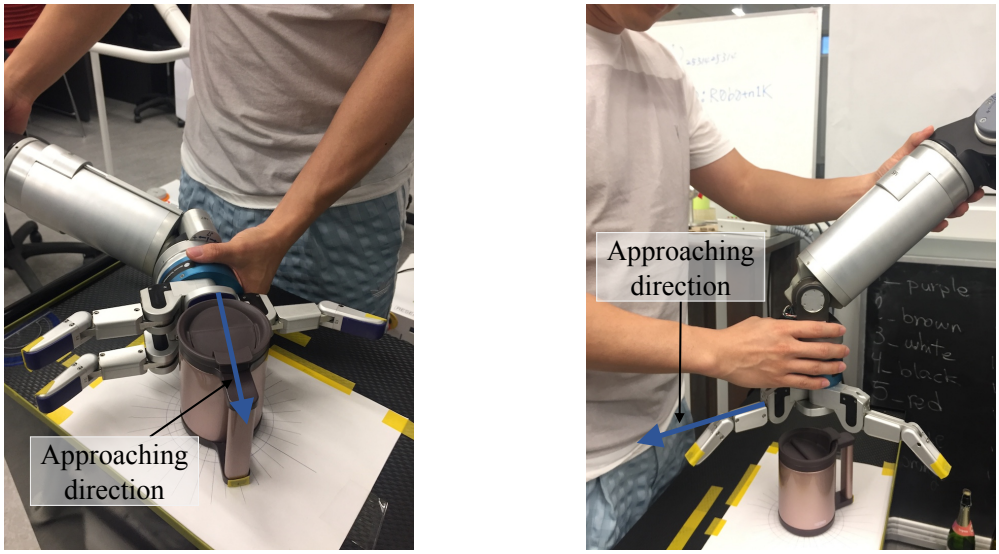
Object description	Dimension of main body (cm)		
	Length	Width	Height
Thermos	8.0	7.9	11.6
Blue (Corelleo)	8.6	8.7	10.1
Black	8.2	8.2	9.6
Maxwell	9.0	8.9	9.4
Stitch	9.8	9.0	11.9
Doraemon	8.9	8.9	15.3
Donald duck	9.1	9.1	12.2
Pluto	11.4	11.3	9.4

Table 4.5: Dimensions of the training objects.

Two tasks (delivery and transportation) are performed with human assistance for each training mug, as shown in Fig 4.15. To accomplish the delivery task, the body of the mug should be grasped from the side while leaving the handle open and available to be received by another person, as shown in Fig. 4.16b. In addition, when the height of the mug is higher than the size of BHand, we will teach the robot to grasp the upper part of the mug, as shown in Fig. 4.16a. To accomplish the transportation task, the body of the mug should be grasped from the top while avoiding to touch the lip of the mug, as shown in Fig. 4.17a. Moreover, none of the finger should hit or block the handle of the mug, as shown in Fig. 4.17b. Specifically, the thumb finger should point to the opposite direction of the handle.

The training data, ψ, collected in the experiment and the dimension of each data are shown in Table 4.6. Since there are two tasks performed in the training phase, we define the delivery task as $[0, 1]$ and the transportation task as $[1, 0]$. As mentioned in Section 4.4.2, the approaching direction of BHand is used to describe the orientation of the object. More specifically, when approaching from the side, the approaching direction is defined as the pointing direction of BHand (as indicated in Fig. 4.15a) with respect to the base of WAM. When approaching from top, the pointing direction of the base of the thumb (as indicated in Fig. 4.15b) with respect to the base of WAM is defined as the approaching direction. The initial joint position of each finger are predefined by the human trainer, since different task associates to different hand pose as introduced in Section 4.4.1. One thing needs to be noted is that the joint velocity of each finger should be carefully tuned, as the three fingers of BHand can not form an equilateral triangle if they move in the same velocity. Therefore, the velocity of the thumb has to be a little lower than that of the other two fingers. Both position and orientation of BHand are defined with respect to the base of WAM, and the orientation is expressed as quaternion. The stain-gauge data can be acquired from the corresponding sensors equipped with BHand. Eventually, we collect 36

grasps for each training mug per task (576 grasps in total) and generate 200 representative grasps using the proposed multi-step clustering algorithm ($\sigma_{\mathbf{k}} = 1$).

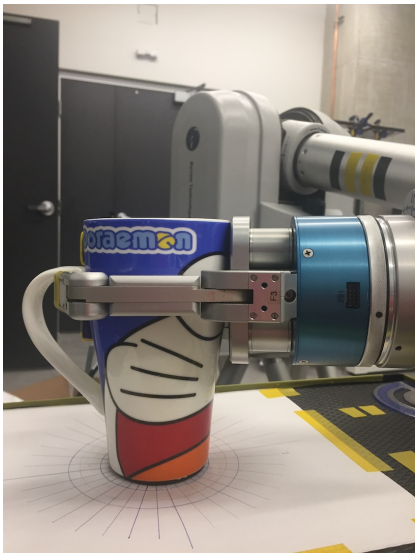


(a) Perform delivery task (b) Perform transportation task

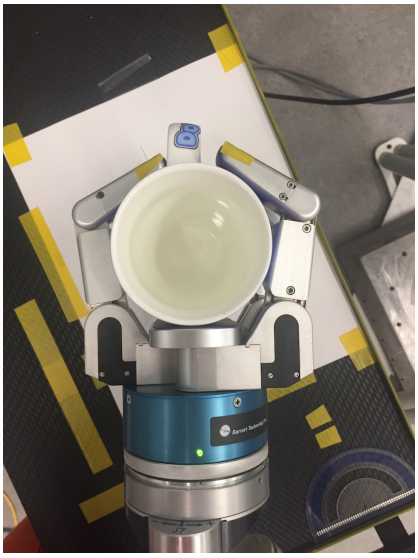
Figure 4.15: Perform delivery and transportation task with human assistance.

Description of training data, ψ	Dimension of training data
Task	$\mathbb{R}^2$
Object dimension	$\mathbb{R}^3$
Approaching direction	$\mathbb{R}^3$
Initial joint positions of BHand	$\mathbb{R}^4$
Joint velocity of each finger	$\mathbb{R}^3$
Position of BHand	$\mathbb{R}^3$
Orientation of BHand	$\mathbb{R}^4$
Strain-gauge	$\mathbb{R}^3$

Table 4.6: Data collected in training experiment.



(a) Side view



(b) Top view

Figure 4.16: A demonstration of grasping from side to accomplish a delivery task.

(a) Side view

(b) Top view

Figure 4.17: A demonstration of grasping from top to accomplish a transportation task.

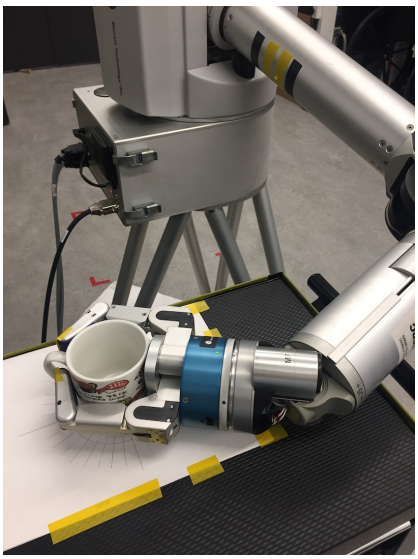
4.7.4 Experimental Results

Once the training dataset is successfully built up, we evaluate the proposed approach by performing task-oriented grasp planning on three novel mugs which are shown in Fig. 4.18. The dimensions of these mugs are listed in Table 4.7. These dimensions and the orientation of the target mug are assumed to be known. All these mugs are placed on a flat surface with different orientations. Some example grasps for the text objects are shown in Fig. 4.19, Fig. 4.20 and Fig. 4.21. In the experiment, all three test mugs can be successfully grasped using the task-oriented grasp which is selected from the codebook. One important observation as demonstrated in Fig. 4.21b is that even though the target mug is not in a standard geometric shape, it can still be grasped using the proposed approach.

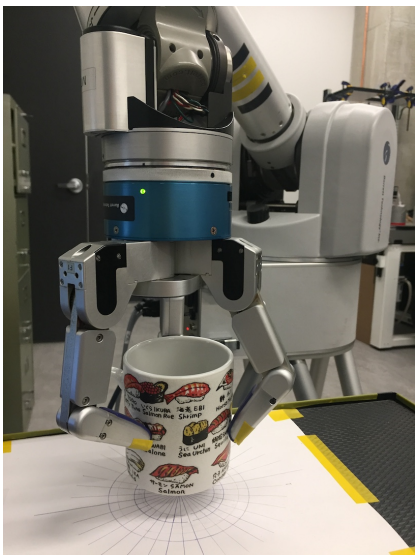
Figure 4.18: Three novel objects used to test the proposed approach.

Object description	Dimension of main body (cm)		
	Length	Width	Height
Sushi	8.0	8.0	8.9
Piglet	9.1	9.1	15.1
Stormtrooper	10.1	9.2	9.4

Table 4.7: Dimensions of the test objects.

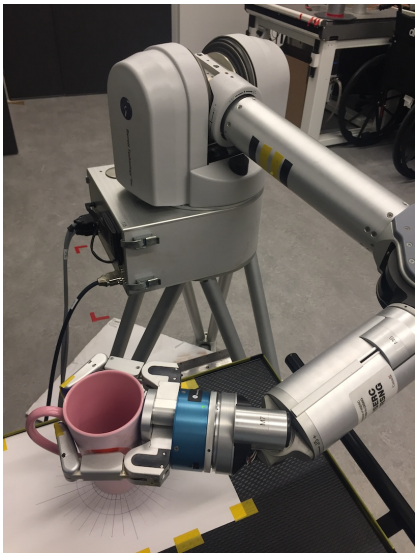


(a) Accomplish the delivery task



(b) Accomplish the transportation task

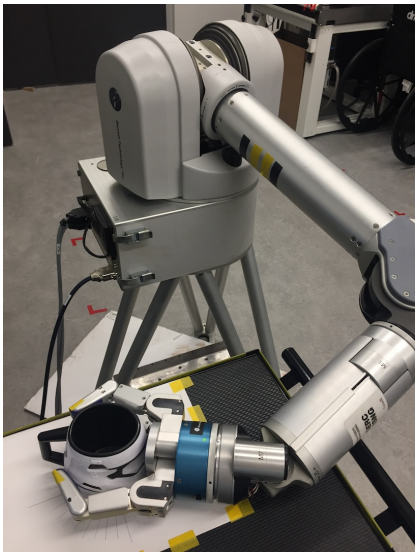
Figure 4.19: Task-oriented grasps for the Sushi mug.



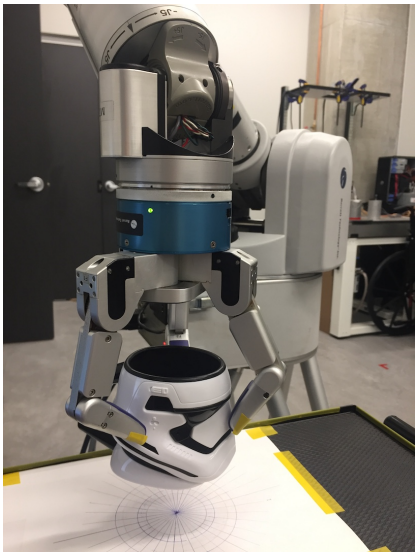
(a) Accomplish the delivery task

(b) Accomplish the transportation task

Figure 4.20: Task-oriented grasps for the Piglet mug.



(a) Accomplish the delivery task



(b) Accomplish the transportation task

Figure 4.21: Task-oriented grasps for the Stormtrooper mug.

4.8 Conclusion

In this section, a knowledge-based approach to the problem of task-oriented grasp planning has been presented. A multi-step clustering algorithm is proposed to analyze the training data and generate a small set of representative grasp based on some specific criteria. By using Gaussian kernel to measure the similarity between the training data, a novel method to determine the initial centroids is also proposed. Simulation results show the benefit of the proposed method for capturing the outlier more accurately then a classic approach (Kaufman approach). Moreover, an auto-growing algorithm for automatic determining the number of clusters is developed. Simulation results validate the advantages of the propose algorithm in terms of accuracy and efficiency. Moreover, the proposed knowledge-based approach is evaluated in a realistic experiment in which a suitable grasp is generated for a novel object to successfully complete a specific task.

Chapter 5

Contributions and further work

5.1 Summary of contributions

In this thesis, a novel sampling approach for GPIS to estimate unknown shapes using tactile exploration under limited sampling points without visual or laser sensors. By tuning the weighting factors, the objective function can adjust the priority between estimating the whole shape of the object and emphasizing on geometric features of the object. The proposed strategy realizes a novel multi-fidelity optimal sample points selection strategy which can provide more accurate shape estimation given the limited number of sample points available (using only 0.41% of reference data points in simulation). Simulation and experimental results are provided to demonstrate the benefit of proposed method for estimating complex geometric features more accurately than existing methods with the same number of sample points.

This thesis also presents an effective grasp planning that uses learned grasping knowledge on a set of training objects to improve the efficiency of the process of generating successful grasps for novel objects. The proposed multi-step clustering algorithm can automatically produce a set of representative grasps while taking manipulation tasks into account. The proposed clustering algorithm is also improved by applying the proposed initialization method. Moreover, the proposed approach is capable of continuously enhancing the grasping skills as more and more objects are manipulated in a real environment. Experimental results demonstrate the advantages of proposed methods in terms of accuracy, efficiency and robustness for grasping a novel object to accomplish specific tasks.

5.2 Publications

The research carried out for this thesis has been, or is intended to be, published in the following.

1. Yang, Shiyi, Soo Jeon, and Jongeun Choi. "Multi-Fidelity Sampling for Rapid Shape Estimation with Tactile Exploration." Under review for IEEE Transactions on Industrial Informatics.

2. Yang, Shiyi, and Soo Jeon. "Recursive Path Planning and Wind Field Estimation for Precision Airdrop." *Journal of Guidance, Control, and Dynamics* 42, no. 6 (2019): 1429-1437.

3. Yang, Shiyi, Nan Wei, Soo Jeon, Ricardo Bencatel, and Anouck Girard. "Real-time optimal path planning and wind estimation using gaussian process regression for precision airdrop." In *2017 American Control Conference (ACC)*, pp. 2582-2587. IEEE, 2017.

4. Yang, Shiyi, Soo Jeon, and Jongeun Choi. "Level-set based greedy algorithm with sequential gaussian process regression for implicit surface estimation." In *ASME 2016 Dynamic Systems and Control Conference*, pp. V002T25A001-V002T25A001. American Society of Mechanical Engineers, 2016.

5.3 Further work

Opportunities for further research extending from the contributions of this thesis have been identified, and are discussed in the following.

5.3.1 Shape Estimation Improvement

Robustness is one of the most important criterion needs to be concerned in the the implementation of the proposed sampling method. To further evaluate the performance of the proposed method, more objects with various shape should be considered.

The tactile exploration strategy used in Chapter 3 is completely dependent on the tactile sensor feedback data. However, the tactile sensor only is not sufficient to provide enough information when dealing with the estimation of other properties of the object and

that of the environment. Hence, more sensors should be utilized, such as vision, strain gage and etc.. As a part of potential future work, the integration of sensing information from different types of sensors can be further studied.

5.3.2 Object Deformability Estimation

The research presented in this thesis is designed with an assumption that the objects being handled are rigid and do not deform when grasped. In reality, however, many objects are non-rigid, such as plastic bottles, aluminum cans, paper tea cups and etc.. Therefore, in addition to the information about the object shape, the knowledge of deformable property of the object or even part of an environment is also crucial to the problem of object grasping and manipulation.

In most of the existing approaches, the interaction between the object surface and the robotic arm is observed only by a vision camera. People, however, often acquire more knowledge of the object properties through the direct contact with the object rather than through the vision perception. It reveals that instead of using vision sensors to detect the deformation of the object, the displacement of the fingertip can also be used to determine the deformation according to the tactile sensor (or strain gage joint-torque sensor) feedback. More specifically, the sensor feedback from the tactile sensors (or strain gage joint-torque sensors) can more precisely describe the interaction with the object than that from the vision sensors. The BHand is equipped with tactile sensors and strain gage joint-torque sensors that can provide such information. Hence, such sensor feedback needs to be efficiently utilized.

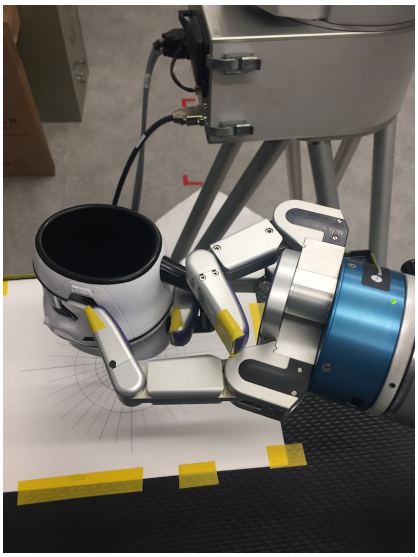
One typical research study [1] introduced in Section 2.2 proposes an idea to represent the deformability of the object as a probabilistic model so as to be estimated by using GPR. This work, however, only concerns the deformability in 1D space where a constant normal force is applied to a 2D flat surface. One possible research is that we can extend this idea to 3D space with 3D objects. Moreover, the proposed estimation method which is presented in Section 3 defines the measurement y as the value of an implicit function to represent if the point is on the surface of the object. In the case of estimating the deformability of the object, the measurement y should be properly defined so that the GPR can be applied to model the deformability.

5.3.3 Grasping Skills Enhancement

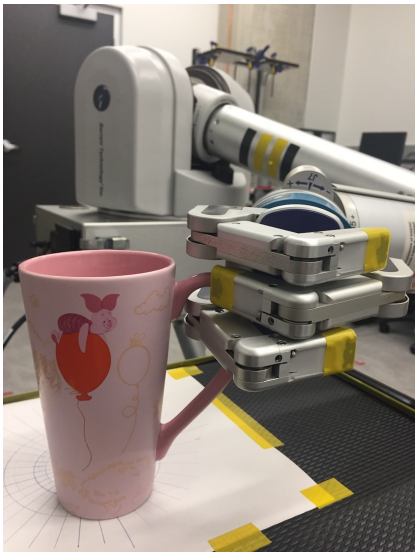
Grasping the handle part of a mug is much harder than grasping the main body of a mug. All the hand poses presented in Section 4.4 can not produce a *force-closure* grasp and guarantee the stability of the grasp. In order to address this technical issue, we develop a special hand pose as shown in Fig. 5.1a. One of the two fingers inserts into the handle and the thumb touches on the main body to provide extra support to the grasp. This particular hand pose can be successfully implemented when the handle of the mug is small. However, when the gap space between the handle and the main body is large, the proposed hand pose is failed to completing the desired task even with the human assistance, as shown in Fig. 5.1b. The mechanical design of BHand is one of reasons to cause the failure. Furthermore, this result indicates that we need a more specific and careful grasp planning for any complicated tasks.

The objects used for training are only mugs. Different tasks using objects with more complicated shapes should be used to further evaluate the performance of the proposed method. In the meantime, since the proposed neural network is capable of auto-improving and evolving, it can be enhanced as more and more objects are grasped.

The experiments conducted in this thesis are all on a Barrett WAM with a three-fingered BHand. One possible future work is to generalize the proposed knowledge-based grasp planning for other dextrous robotic hands, such as Kinova gripper KG-3 or hands equipped with more fingers.



(a) Grasp the handle of the Stormtrooper mug.



(b) Grasp the handle of the piglet mug.

Figure 5.1: Examples of grasping handle of mugs.

References

[1] S. Caccamo, P. Güler, H. Kjellström, and D. Kragic, "Active perception and modeling of deformable surfaces using gaussian processes and position-based dynamics," in *IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*. IEEE, 2016, pp. 530–537.

[2] J. Bohg, M. Johnson-Roberson, B. Len, J. Felip, X. Gratal, N. Bergstrm, D. Kragic, and A. Morales, "Mind the gap - robotic grasping under incomplete observation," in *IEEE International Conference on Robotics and Automation (ICRA)*, May 2011, pp. 686–693.

[3] G. M. Bone, A. Lambert, and M. Edwards, "Automated modeling and robotic grasping of unknown three-dimensional objects," in *IEEE International Conference on Robotics and Automation (ICRA)*, May 2008, pp. 292–298.

[4] O. Williams and A. Fitzgibbon, "Gaussian process implicit surfaces," *Gaussian Proc. in Practice*, pp. 1–4, 2007.

[5] A. Krause, A. Singh, and C. Guestrin, "Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies," *Journal of Machine Learning Research*, vol. 9, no. Feb, pp. 235–284, 2008.

[6] S. Ottenhaus, P. Weiner, L. Kaul, A. Tulbure, and T. Asfour, "Exploration and reconstruction of unknown objects using a novel normal and contact sensor," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 1614–1620.

[7] 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. 1850015, 2018.

[8] 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)*, Sep. 2017, pp. 65–72.

[9] T. Yoshikawa, "Multifingered robot hands: Control for grasping and manipulation," *Annual Reviews in Control*, vol. 34, no. 2, pp. 199–208, 2010.

[10] S. El-Khoury and A. Sahbani, "Handling objects by their handles," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, no. POST_TALK, 2008.

[11] J. Bohg, A. Morales, T. Asfour, and D. Kragic, "Data-driven grasp synthesis - a survey," *IEEE Transactions on Robotics*, vol. 30, no. 2, pp. 289–309, 2013.

[12] E. Nikandrova and V. Kyrki, "Category-based task specific grasping," *Robotics and Autonomous Systems*, vol. 70, pp. 25–35, 2015.

[13] D. Meike and L. Ribickis, "Energy efficient use of robotics in the automobile industry," in *15th International Conference on Advanced Robotics (ICAR)*, June 2011, pp. 507–511.

[14] A. P. Advincula, "Robot-assisted laparoscopy: Basic principles equipment and instrumentation," *Emerging Technologies in Women's Health-Robotic Surgery in Gynecology*, vol. 1, p. 3, 2011.

[15] E. Guizzo and E. Ackerman, "How rethink robotics built its new baxter robot worker," *IEEE spectrum*, 2012.

[16] B. Graf, U. Reiser, M. Hägele, K. Mauz, and P. Klein, "Robotic home assistant care-o-bot® 3-product vision and innovation platform," in *IEEE Workshop on Advanced Robotics and its Social Impacts (ARSO)*. IEEE, 2009, pp. 139–144.

[17] K. Yamazaki, R. Ueda, S. Nozawa, M. Kojima, K. Okada, K. Matsumoto, M. Ishikawa, I. Shimoyama, and M. Inaba, "Home-assistant robot for an aging society," *Proceedings of the IEEE*, vol. 100, no. 8, pp. 2429–2441, 2012.

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

[19] N. F. Lepora, K. Aquilina, and L. Cramphorn, "Exploratory tactile servoing with active touch," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 1156–1163, 2017.

[20] D. M. Wolpert, J. Diedrichsen, and J. R. Flanagan, "Principles of sensorimotor learning," *Nature Reviews Neuroscience*, vol. 12, no. 12, pp. 739–751, 2011.

[21] P. Hebert, N. Hudson, J. Ma, and J. Burdick, "Fusion of stereo vision, force-torque, and joint sensors for estimation of in-hand object location," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2011, pp. 5935–5941.

[22] T. Supuk, T. Bajd, and G. Kurillo, "Assessment of reach-to-grasp trajectories toward stationary objects," *Clinical biomechanics*, vol. 26, no. 8, pp. 811–818, 2011.

[23] T. Feix, I. M. Bullock, and A. M. Dollar, "Analysis of human grasping behavior: Object characteristics and grasp type," *IEEE transactions on haptics*, vol. 7, no. 3, pp. 311–323, 2014.

[24] ——, "Analysis of human grasping behavior: Correlating tasks, objects and grasps," *IEEE transactions on haptics*, vol. 7, no. 4, pp. 430–441, 2014.

[25] A. Bicchi and V. Kumar, "Robotic grasping and contact: A review," in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*, vol. 1. IEEE, 2000, pp. 348–353.

[26] N. Curtis and J. Xiao, "Efficient and effective grasping of novel objects through learning and adapting a knowledge base," in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2008, pp. 2252–2257.

[27] J. Bohg, K. Welke, B. León, M. Do, D. Song, W. Wohlkinger, M. Madry, A. Aldóma, M. Przybylski, T. Asfour, *et al.*, "Task-based grasp adaptation on a humanoid robot," *IFAC Proceedings Volumes*, vol. 45, no. 22, pp. 779–786, 2012.

[28] H. B. Hmida, C. Cruz, F. Boochs, and C. Nicolle, "Knowledge base approach for 3d objects detection in point clouds using 3d processing and specialists knowledge," *International Journal on Advances in Intelligent Systems*, vol. 5, 2013.

[29] M. Meier, M. Schopfer, R. Haschke, and H. Ritter, "A probabilistic approach to tactile shape reconstruction," *IEEE Transactions on Robotics*, vol. 27, no. 3, pp. 630–635, 2011.

[30] R. Fabio, "From point cloud to surface: the modeling and visualisation problem," in *International Workshop on Visualization and Animation of Reality-based 3D Models.*, vol. 34, 2003, p. 5.

[31] R. B. Rusu, Z. C. Marton, N. Blodow, M. Dolha, and M. Beetz, "Towards 3d point cloud based object maps for household environments," *Robotics and Autonomous Systems*, vol. 56, no. 11, pp. 927–941, 2008.

[32] C. H. Esteban and F. Schmitt, "Silhouette and stereo fusion for 3d object modelling," *Computer Vision and Image Understanding*, vol. 96, no. 3, pp. 367–392, 2004.

[33] J. Varley, C. DeChant, A. Richardson, A. Nair, J. Ruales, and P. Allen, "Shape completion enabled robotic grasping," *arXiv preprint arXiv:1609.08546*, 2016.

[34] A. T. Miller, S. Knoop, H. I. Christensen, and P. K. Allen, "Automatic grasp planning using shape primitives," in *IEEE International Conference on Robotics and Automation (ICRA)*, vol. 2, 2003, pp. 1824–1829.

[35] C. Goldfeder, P. K. Allen, C. Lackner, and R. Pelossof, "Grasp planning via decomposition trees," in *Proceedings 2007 IEEE International Conference on Robotics and Automation*, April 2007, pp. 4679–4684.

[36] S. Dragiev, "An object representation and methods for uncertainty-aware shape estimation and grasping," 2014.

[37] K. Huebner, S. Ruthotto, and D. Kragic, "Minimum volume bounding box decomposition for shape approximation in robot grasping," in *2008 IEEE International Conference on Robotics and Automation*, May 2008, pp. 1628–1633.

[38] J. Tegin, J. Wikander, S. Ekvall, D. Kragic, and B. Iliev, "Experience based learning and control of robotic grasping," in *IEEE-RAS International Conference on Humanoid Robots, Workshop: Towards Cognitive Humanoid Robots*, 2006.

[39] K. Huebner and D. Kragic, "Selection of robot pre-grasps using box-based shape approximation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2008, pp. 1765–1770.

[40] G. Barequet and S. Har-Peled, "Efficiently approximating the minimum-volume bounding box of a point set in three dimensions," *Journal of Algorithms*, vol. 38, no. 1, pp. 91–109, 2001.

[41] A. Pasko, V. Adzhiev, A. Sourin, and V. Savchenko, "Function representation in geometric modeling: concepts, implementation and applications," *The Visual Computer*, vol. 11, no. 8, pp. 429–446, 1995.

[42] G. Yngve and G. Turk, "Robust creation of implicit surfaces from polygonal meshes," *IEEE transactions on visualization and computer graphics*, vol. 8, no. 4, pp. 346–359, 2002.

[43] A. Adamson and M. Alexa, "Approximating and intersecting surfaces from points," in *Proceedings of the 2003 Eurographics/ACM SIGGRAPH symposium on Geometry processing*. Eurographics Association, 2003, pp. 230–239.

[44] S. Dragiev, M. Toussaint, and M. Gienger, "Gaussian process implicit surfaces for shape estimation and grasping," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2011, pp. 2845–2850.

[45] J. Mahler, S. Patil, B. Kehoe, J. Van Den Berg, M. Ciocarlie, P. Abbeel, and K. Goldberg, "Gp-gpis-opt: Grasp planning with shape uncertainty using gaussian process implicit surfaces and sequential convex programming," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 4919–4926.

[46] J. F. Blinn, "A generalization of algebraic surface drawing," *ACM transactions on graphics (TOG)*, vol. 1, no. 3, pp. 235–256, 1982.

[47] B. Curless and M. Levoy, "A volumetric method for building complex models from range images," in *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*. ACM, 1996, pp. 303–312.

[48] Y. Ohtake, A. Belyaev, M. Alexa, G. Turk, and H.-P. Seidel, "Multi-level partition of unity implicits," in *Acm Siggraph 2005 Courses*. ACM, 2005, p. 173.

[49] F. Steinke, B. Schölkopf, and V. Blanz, "Support vector machines for 3d shape processing," in *Computer Graphics Forum*, vol. 24, no. 3. Wiley Online Library, 2005, pp. 285–294.

[50] J. Tegin, S. Ekvall, D. Kragic, J. Wikander, and B. Iliev, "Demonstration-based learning and control for automatic grasping," *Intelligent Service Robotics*, vol. 2, no. 1, pp. 23–30, 2009.

[51] M. Nieuwenhuisen, J. Stueckler, A. Berner, R. Klein, and S. Behnke, "Shape-primitive based object recognition and grasping," in *7th German Conference on Robotics (ROBOTIK)*, May 2012, pp. 1–5.

[52] Z. C. Marton, D. Pangercic, N. Blodow, J. Kleinehellefort, and M. Beetz, "General 3d modelling of novel objects from a single view," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct 2010, pp. 3700–3705.

[53] C. Dune, E. Marchand, C. Collowet, and C. Leroux, "Active rough shape estimation of unknown objects," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sept 2008, pp. 3622–3627.

[54] V. Lippiello, F. Ruggiero, B. Siciliano, and L. Villani, "Visual grasp planning for unknown objects using a multifingered robotic hand," *IEEE/ASME Transactions on Mechatronics*, vol. 18, no. 3, pp. 1050–1059, June 2013.

[55] J. Ilonen, J. Bohg, and V. Kyrki, "Fusing visual and tactile sensing for 3-d object reconstruction while grasping," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2013, pp. 3547–3554.

[56] 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.

[57] S. Dragiev, M. Toussaint, and M. Gienger, "Uncertainty aware grasping and tactile exploration," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2013, pp. 113–119.

[58] K. Chalupka, C. K. Williams, and I. Murray, "A framework for evaluating approximation methods for gaussian process regression," *Journal of Machine Learning Research*, vol. 14, no. Feb, pp. 333–350, 2013.

[59] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2005.

[60] D. J. MacKay, *Information theory, inference and learning algorithms*. Cambridge university press, 2003.

[61] S. Vasudevan, F. Ramos, E. Nettleton, and H. Durrant-Whyte, "Gaussian process modeling of large-scale terrain," *Journal of Field Robotics*, vol. 26, no. 10, pp. 812–840, 2009.

[62] S. T. O?Callaghan and F. T. Ramos, "Gaussian process occupancy maps," *The International Journal of Robotics Research*, vol. 31, no. 1, pp. 42–62, 2012.

[63] N. D. Lawrence, "Gaussian process latent variable models for visualisation of high dimensional data," in *Advances in neural information processing systems*, 2004, pp. 329–336.

[64] S. Keerthi and W. Chu, "A matching pursuit approach to sparse gaussian process regression," in *Advances in neural information processing systems*, 2006, pp. 643–650.

[65] J. Quiñonero-Candela and C. E. Rasmussen, "A unifying view of sparse approximate gaussian process regression," *Journal of Machine Learning Research*, vol. 6, no. Dec, pp. 1939–1959, 2005.

[66] J. Quinonero-Candela, C. E. Rasmussen, and C. K. Williams, "Approximation methods for gaussian process regression," *Large-scale kernel machines*, pp. 203–224, 2007.

[67] A. Banerjee, D. B. Dunson, and S. T. Tokdar, "Efficient gaussian process regression for large datasets," *Biometrika*, vol. 100, no. 1, pp. 75–89, 2012.

[68] E. Snelson and Z. Ghahramani, "Local and global sparse gaussian process approximations," in *Artificial Intelligence and Statistics*, 2007, pp. 524–531.

[69] Y. Shen, M. Seeger, and A. Y. Ng, "Fast gaussian process regression using kd-trees," in *Advances in neural information processing systems*, 2006, pp. 1225–1232.

[70] S. Park, S. K. Mustafa, and K. Shimada, "Learning based robot control with sequential gaussian process," in *IEEE Workshop on Robotic Intelligence In Informationally Structured Space (RiiSS)*, 2013, pp. 120–127.

[71] M. F. Huber, "Recursive gaussian process regression," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2013, pp. 3362–3366.

[72] N. Jamali, C. Ciliberto, L. Rosasco, and L. Natale, "Active perception: Building objects' models using tactile exploration," in *IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, 2016, pp. 179–185.

[73] Z. Yi, R. Calandra, F. Veiga, H. van Hoof, T. Hermans, Y. Zhang, and J. Peters, "Active tactile object exploration with gaussian processes," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016, pp. 4925–4930.

[74] S. Lazebnik, C. Schmid, and J. Ponce, "A maximum entropy framework for part-based texture and object recognition," in *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, Oct 2005, pp. 832–838 Vol. 1.

[75] C. E. Shannon, "A mathematical theory of communication," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 5, no. 1, pp. 3–55, 2001.

[76] C.-W. Ko, J. Lee, and M. Queyranne, "An exact algorithm for maximum entropy sampling," *Operations Research*, vol. 43, no. 4, pp. 684–691, 1995.

[77] C. Currin, T. Mitchell, M. Morris, and D. Ylvisaker, "Bayesian prediction of deterministic functions, with applications to the design and analysis of computer experiments," *Journal of the American Statistical Association*, vol. 86, no. 416, pp. 953–963, 1991.

[78] N. Cressie, *Statistics for spatial data*. John Wiley & Sons, 2015.

[79] S. Yang, S. Jeon, and J. Choi, "Level-set based greedy algorithm with sequential gaussian process regression for implicit surface estimation," in *ASME 2016 Dynamic Systems and Control Conference*. American Society of Mechanical Engineers, 2016, pp. V002T25A001–V002T25A001.

[80] S. Ottenhaus, M. Miller, D. Schiebener, N. Vahrenkamp, and T. Asfour, "Local implicit surface estimation for haptic exploration," in *IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*. IEEE, 2016, pp. 850–856.

[81] 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.

[82] N. Pestell, J. Lloyd, J. Rossiter, and N. F. Lepora, "Dual-modal tactile perception and exploration," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 1033–1040, 2018.

[83] R. M. Murray, Z. Li, S. S. Sastry, and S. S. Sastry, *A mathematical introduction to robotic manipulation*. CRC press, 1994.

[84] B. Siciliano and O. Khatib, *Springer handbook of robotics*. Springer, 2016.

[85] Y.-H. Liu, "Qualitative test and force optimization of 3-d frictional form-closure grasps using linear programming," *IEEE Transactions on Robotics and Automation*, vol. 15, no. 1, pp. 163–173, 1999.

[86] D. Ding, Y.-H. Liu, and S. Wang, "Computing 3-d optimal form-closure grasps," in *IEEE International Conference on Robotics and Automation (ICRA)*, vol. 4, 2000, pp. 3573–3578.

[87] J.-W. Li, H. Liu, and H.-G. Cai, "On computing three-finger force-closure grasps of 2-d and 3-d objects," *IEEE Transactions on Robotics and Automation*, vol. 19, no. 1, pp. 155–161, 2003.

[88] Y.-H. Liu, M.-L. Lam, and D. Ding, "A complete and efficient algorithm for searching 3-d form-closure grasps in the discrete domain," *IEEE Transactions on Robotics*, vol. 20, no. 5, pp. 805–816, Oct 2004.

[89] M. A. Roa and R. Suárez, "Computation of independent contact regions for grasping 3-d objects," *IEEE Transactions on Robotics*, vol. 25, no. 4, pp. 839–850, 2009.

[90] R. Krug, D. Dimitrov, K. Charusta, and B. Iliev, "On the efficient computation of independent contact regions for force closure grasps," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2010, pp. 586–591.

[91] R. Suárez, J. Cornella, and M. R. Garzón, *Grasp quality measures*. Institut d'Organització i Control de Sistemes Industrials, 2006.

[92] M. A. Roa and R. Suárez, "Grasp quality measures: review and performance," *Autonomous robots*, vol. 38, no. 1, pp. 65–88, 2015.

[93] B.-H. Kim, S.-R. Oh, B.-J. Yi, and I. H. Suh, "Optimal grasping based on non-dimensionalized performance indices," in *Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium (Cat. No.01CH37180)*, vol. 2, 2001, pp. 949–956 vol.2.

[94] M. A. Roa, R. Koiva, and C. Castellini, "Experimental evaluation of human grasps using a sensorized object," in *4th IEEE RAS & EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob)*. IEEE, 2012, pp. 1662–1668.

[95] E. Chinellato, A. Morales, R. B. Fisher, and A. P. Del Pobil, "Visual quality measures for characterizing planar robot grasps," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 35, no. 1, pp. 30–41, 2005.

[96] Y. Zheng, "An efficient algorithm for a grasp quality measure," *IEEE Transactions on Robotics*, vol. 29, no. 2, pp. 579–585, 2013.

[97] M. R. Cutkosky, "Grasping and fine manipulation for automated manufacturing," Ph.D. dissertation, December 1985.

[98] I. Biederman, "Recognition-by-components: a theory of human image understanding." *Psychological review*, vol. 94, no. 2, p. 115, 1987.

[99] S. El-Khoury, A. Sahbani, and V. Perdereau, "Learning the natural grasping component of an unknown object," in *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2007, pp. 2957–2962.

[100] D. Rao, Q. V. Le, T. Phoka, M. Quigley, A. Sudsang, and A. Y. Ng, "Grasping novel objects with depth segmentation," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2010, pp. 2578–2585.

[101] H. O. Song, M. Fritz, D. Goehring, and T. Darrell, "Learning to detect visual grasp affordance," *IEEE Transactions on Automation Science and Engineering*, vol. 13, no. 2, pp. 798–809, 2015.

[102] Y. Shiraki, K. Nagata, N. Yamanobe, A. Nakamura, K. Harada, D. Sato, and D. N. Nenchev, "Modeling of everyday objects for semantic grasp," in *The 23rd IEEE International Symposium on Robot and Human Interactive Communication*. IEEE, 2014, pp. 750–755.

[103] A. Ten Pas and R. Platt, "Localizing handle-like grasp affordances in 3d point clouds," in *Experimental Robotics*. Springer, 2016, pp. 623–638.

[104] M. Kokic, J. A. Stork, J. A. Haustein, and D. Kragic, "Affordance detection for task-specific grasping using deep learning," in *2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids)*. IEEE, 2017, pp. 91–98.

[105] K. Fang, Y. Zhu, A. Garg, A. Kurenkov, V. Mehta, L. Fei-Fei, and S. Savarese, "Learning task-oriented grasping for tool manipulation from simulated self-supervision," *arXiv preprint arXiv:1806.09266*, 2018.

[106] D. Song, K. Huebner, V. Kyrki, and D. Kragic, "Learning task constraints for robot grasping using graphical models," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2010, pp. 1579–1585.

[107] D. Song, C. H. Ek, K. Huebner, and D. Kragic, "Multivariate discretization for bayesian network structure learning in robot grasping," in *2011 IEEE International Conference on Robotics and Automation*. IEEE, 2011, pp. 1944–1950.

[108] Y. Bekiroglu, D. Song, L. Wang, and D. Kragic, "A probabilistic framework for task-oriented grasp stability assessment," in *2013 IEEE International Conference on Robotics and Automation*. IEEE, 2013, pp. 3040–3047.

[109] H. Dang and P. K. Allen, "Semantic grasping: Planning robotic grasps functionally suitable for an object manipulation task," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 1311–1317.

[110] N. Vahrenkamp, L. Westkamp, N. Yamanobe, E. E. Aksoy, and T. Asfour, "Part-based grasp planning for familiar objects," in *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*. IEEE, 2016, pp. 919–925.

[111] M. Stark, P. Lies, M. Zillich, J. Wyatt, and B. Schiele, "Functional object class detection based on learned affordance cues," in *International conference on computer vision systems*. Springer, 2008, pp. 435–444.

[112] A. T. Miller and P. K. Allen, "Graspit! a versatile simulator for robotic grasping," *IEEE Robotics Automation Magazine*, vol. 11, no. 4, pp. 110–122, Dec 2004.

[113] E. K. née Nikandrova and V. Kyrki, "Task-specific grasping of simiiar objects by probabiiistic fusion of vision and tactiie measurements," in *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*. IEEE, 2015, pp. 704–710.

[114] C. Goldfeder, M. Ciocarlie, Hao Dang, and P. K. Allen, "The columbia grasp database," in *2009 IEEE International Conference on Robotics and Automation*, May 2009, pp. 1710–1716.

[115] C. Borst, M. Fischer, and G. Hirzinger, "Grasping the dice by dicing the grasp," in *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 4, Oct 2003, pp. 3692–3697 vol.3.

[116] L. Kaufman and P. J. Rousseeuw, *Finding groups in data. an introduction to cluster analysis*, 1990.

[117] P. Arora, S. Varshney, *et al.*, "Analysis of k-means and k-medoids algorithm for big data," *Procedia Computer Science*, vol. 78, pp. 507–512, 2016.

[118] R. Detry, C. H. Ek, M. Madry, J. Piater, and D. Kragic, "Generalizing grasps across partly similar objects," in *2012 IEEE International Conference on Robotics and Automation*. IEEE, 2012, pp. 3791–3797.

[119] R. Detry, C. H. Ek, M. Madry, and D. Kragic, "Learning a dictionary of prototypical grasp-predicting parts from grasping experience," in *2013 IEEE International Conference on Robotics and Automation*. IEEE, 2013, pp. 601–608.

[120] T. Kohonen, "The self-organizing map," *Proceedings of the IEEE*, vol. 78, no. 9, pp. 1464–1480, 1990.

[121] Y. Chen, B. Qin, T. Liu, Y. Liu, and S. Li, "The comparison of som and k-means for text clustering." *Computer and Information Science*, vol. 3, no. 2, pp. 268–274, 2010.

[122] J. Kangas, "Sample weighting when training self-organizing maps for image compression," in *Proceedings of 1995 IEEE Workshop on Neural Networks for Signal Processing*. IEEE, 1995, pp. 343–350.

[123] Z. Yu and M. Lee, "Real-time human action classification using a dynamic neural model," *Neural Networks*, vol. 69, pp. 29–43, 2015.

[124] H. Hua, "Image and geometry processing with oriented and scalable map," *Neural Networks*, vol. 77, pp. 1–6, 2016.

[125] D.-W. Huang, R. J. Gentili, G. E. Katz, and J. A. Reggia, "A limit-cycle self-organizing map architecture for stable arm control," *Neural Networks*, vol. 85, pp. 165–181, 2017.

[126] T. Kohonen and P. Somervuo, "How to make large self-organizing maps for nonvectorial data," *Neural networks*, vol. 15, no. 8-9, pp. 945–952, 2002.

[127] H. Matsushita and Y. Nishio, "Batch-learning self-organizing map with weighted connections avoiding false-neighbor effects," in *The 2010 international joint conference on neural networks (IJCNN)*. IEEE, 2010, pp. 1–6.

[128] E. Forgey, "Cluster analysis of multivariate data: Efficiency vs. interpretability of classification," *Biometrics*, vol. 21, no. 3, pp. 768–769, 1965.

[129] J. MacQueen *et al.*, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, vol. 1, no. 14. Oakland, CA, USA, 1967, pp. 281–297.

[130] D. Arthur and S. Vassilvitskii, "k-means++: The advantages of careful seeding," in *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*. Society for Industrial and Applied Mathematics, 2007, pp. 1027–1035.

[131] M. E. Celebi, H. A. Kingravi, and P. A. Vela, "A comparative study of efficient initialization methods for the k-means clustering algorithm," *Expert systems with applications*, vol. 40, no. 1, pp. 200–210, 2013.

[132] J. M. Pena, J. A. Lozano, and P. Larranaga, "An empirical comparison of four initialization methods for the k-means algorithm," *Pattern recognition letters*, vol. 20, no. 10, pp. 1027–1040, 1999.

[133] R. Tibshirani, G. Walther, and T. Hastie, "Estimating the number of clusters in a data set via the gap statistic," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 63, no. 2, pp. 411–423, 2001.

[134] A. Ng, "Clustering with the k-means algorithm," *Machine Learning*, 2012.

[135] B. Fritzke, "A growing neural gas network learns topologies," in *Advances in neural information processing systems*, 1995, pp. 625–632.

[136] Q. Zhang, C. Zhu, L. T. Yang, Z. Chen, L. Zhao, and P. Li, "An incremental cfs algorithm for clustering large data in industrial internet of things," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 3, pp. 1193–1201, June 2017.

[137] K. Doherty, R. Adams, and N. Davey, "Treegng-hierarchical topological clustering," in *Procs of ESANN 2005: European Symposium on Artificial Neural Networks*, 2005.

[138] N.-Q. Doan, H. Azzag, and M. Lebbah, "Growing self-organizing trees for autonomous hierarchical clustering," *Neural Networks*, vol. 41, pp. 85–95, 2013.

[139] H. Liu and X.-j. Ban, "Clustering by growing incremental self-organizing neural network," *Expert Systems with Applications*, vol. 42, no. 11, pp. 4965–4981, 2015.

[140] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle, "Surface reconstruction from unorganized points," in *Proceedings of the 19th Annual Conference on Computer Graphics and Interactive Techniques*, New York, NY, 1992, pp. 71–78.

[141] M. Pauly, M. Gross, and L. P. Kobbelt, "Efficient simplification of point-sampled surfaces," in *IEEE Visualization, 2002. VIS 2002.*, Oct 2002, pp. 163–170.

[142] D. P. Huttenlocher, G. A. Klanderman, and W. J. Rucklidge, "Comparing images using the hausdorff distance," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no. 9, pp. 850–863, Sep. 1993.

[143] H. N. Do, A. Ijaz, H. Gharahi, B. Zambrano, J. Choi, W. Lee, and S. Baek, "Prediction of abdominal aortic aneurysm growth using dynamical gaussian process implicit surface," *IEEE Transactions on Biomedical Engineering*, 2018.

[144] M. Cai, K. M. Kitani, and Y. Sato, "Understanding hand-object manipulation with grasp types and object attributes." in *Robotics: Science and Systems*, vol. 3. Ann Arbor, Michigan;, 2016.

[145] D. Kappler, J. Bohg, and S. Schaal, "Leveraging big data for grasp planning," in *IEEE International Conference on Robotics and Automation (ICRA)*, May 2015, pp. 4304–4311.

APPENDICES

Appendix A

Sequential Gaussian Process Regression

The aim of this appendix is to demonstrate the details of process to derive Eq. (3.29) in Section 3.4. For the sake of simplifying the notation, x and X, when used in the equations, will represent $f(x)$ and $f(X)$ in this appendix.

Suppose we sequentially select three sample points (x_2, x_3 and x_4), and let $X_1 = \{x_1\}$ (the subscript indicates the number of sample point contained in the set) denote the initial sample set. Then, the conditional variance of x_2 given X_1 can be obtained by Eq. (3.8)

$$\mathbb{V}\left(x_2 \mid X_1\right) = \Sigma_{x_2x_2} - \Sigma_{x_2X_1}\Sigma_{X_1X_1}^{-1}\Sigma_{X_1x_2}, \tag{A.1}$$

where the covariance matrices $\Sigma_{x_2x_2}$, $\Sigma_{x_2X_1}$ and $\Sigma_{X_1X_1}$ can be obtained by Eq. (3.4). Adding x_2 into set X_1, the conditional variance of x_3 given X_2 ($X_2 = \{x_1, x_2\}$) can be expressed as

$$\mathbb{V}\left(x_3 \mid x_1, x_2\right) = \Sigma_{\tilde{x}_{3|1}\tilde{x}_{3|1}} - \Sigma_{\tilde{x}_{3|1}\tilde{x}_{2|1}}\Sigma_{\tilde{x}_{2|1}\tilde{x}_{2|1}}^{-1}\Sigma_{\tilde{x}_{2|1}\tilde{x}_{3|1}}, \tag{A.2}$$

where $\tilde{x}_{3|1}$ and $\tilde{x}_{2|1}$ can be obtained by Eqs. (3.27) and (3.28)

$$\begin{aligned}\tilde{x}_{3|1} &= x_3 - \mathbb{E}[x_3 \mid X_1] \\ &= x_3 - \left(\mathrm{m}(x_3) + \Sigma_{x_3X_1}\Sigma_{X_1X_1}^{-1}\left(X_1 - \mathrm{m}(X_1)\right)\right),\end{aligned} \tag{A.3}$$

$$\begin{aligned}\tilde{x}_{2|1} &= x_2 - \mathbb{E}[x_2 \mid X_1] \\ &= x_2 - \left(\mathrm{m}(x_2) + \Sigma_{x_2 X_1}\Sigma_{X_1 X_1}^{-1}\left(X_1 - \mathrm{m}(X_1)\right)\right)\end{aligned} \tag{A.4}$$

Also from Eq. (3.7), we can derive the expected value of $\tilde{x}_{3|1}$ and $\tilde{x}_{2|1}$ as

$$\begin{aligned}\mathbb{E}\left[\tilde{x}_{3|1}\right] &= \mathbb{E}\left[x_3 - \mathbb{E}[x_3 \mid X_1]\right] \\ &= \mathrm{m}(x_3) - \mathbb{E}\left[\mathrm{m}(x_3) + \Sigma_{x_3 X_1}\Sigma_{X_1 X_1}^{-1}\left(x_1 - \mathrm{m}(x_1)\right)\right] \\ &= \mathrm{m}(x_3) - \left(\mathrm{m}(x_3) + \Sigma_{x_3 X_1}\Sigma_{X_1 X_1}^{-1}\left(\mathrm{m}(X_1) - \mathrm{m}(X_1)\right)\right) \\ &= 0,\end{aligned} \tag{A.5}$$

$$\begin{aligned}\mathbb{E}\left[\tilde{x}_{2|1}\right] &= \mathbb{E}\left[x_2 - \mathbb{E}[x_2 \mid X_1]\right] \\ &= \mathrm{m}(x_2) - \mathbb{E}\left[\mathrm{m}(x_2) + \Sigma_{x_2 X_1}\Sigma_{X_1 X_1}^{-1}\left(X_1 - \mathrm{m}(X_1)\right)\right] \\ &= \mathrm{m}(x_2) - \left(\mathrm{m}(x_2) + \Sigma_{x_2 X_1}\Sigma_{X_1 X_1}^{-1}\left(\mathrm{m}(X_1) - \mathrm{m}(X_1)\right)\right) \\ &= 0.\end{aligned} \tag{A.6}$$

Taking $\tilde{x}_{3|1}$ and $\tilde{x}_{2|1}$ with their expected values, we can derive $\Sigma_{\tilde{x}_{3|1}\tilde{x}_{3|1}}$ as

$$\begin{aligned}\Sigma_{\tilde{x}_{3|1}\tilde{x}_{3|1}} &= \mathbb{E}\left[\left(\tilde{x}_{3|1} - \mathbb{E}[\tilde{x}_{3|1}]\right)\left(\tilde{x}_{3|1} - \mathbb{E}[\tilde{x}_{3|1}]\right)^{\intercal}\right] \\ &= \mathbb{E}[\left(\left(x_3 - \mathrm{m}(x_3)\right) - \Sigma_{x_3 X_1}\Sigma_{X_1 X_1}^{-1}\left(X_1 - \mathrm{m}(X_1)\right)\right) \\ &\quad \left(\left(x_3 - \mathrm{m}(x_3)\right) - \Sigma_{x_3 X_1}\Sigma_{X_1 X_1}^{-1}\left(X_1 - \mathrm{m}(X_1)\right)\right)^{\intercal}] \\ &= \mathbb{E}\left[\left(x_3 - \mathrm{m}(x_3)\right)\left(X_1 - \mathrm{m}(X_1)\right)^{\intercal}\right] \\ &\quad - \mathbb{E}\left[\Sigma_{x_3 X_1}\Sigma_{X_1 X_1}^{-1}\left(X_1 - \mathrm{m}(X_1)\right)\left(x_3 - \mathrm{m}(x_3)\right)^{\intercal}\right] \\ &\quad - \mathbb{E}\left[\left(x_3 - \mathrm{m}(x_3)\right)\left(X_1 - \mathrm{m}(X_1)\right)^{\intercal}\Sigma_{X_1 X_1}^{-1}\Sigma_{X_1 x_3}\right] \\ &\quad + \mathbb{E}\left[\Sigma_{x_3 X_1}\Sigma_{X_1 X_1}^{-1}\left(X_1 - \mathrm{m}(X_1)\right)\left(X_1 - \mathrm{m}(X_1)\right)^{\intercal}\Sigma_{X_1 X_1}^{-1}\Sigma_{X_1 x_3}\right] \\ &= \Sigma_{x_3 x_3} - \Sigma_{x_3 X_1}\Sigma_{X_1 X_1}^{-1}\Sigma_{X_1 x_3} - \Sigma_{x_3 X_1}\Sigma_{X_1 X_1}^{-1}\Sigma_{X_1 x_3} + \Sigma_{x_3 X_1}\Sigma_{X_1 X_1}^{-1}\Sigma_{X_1 x_3} \\ &= \Sigma_{x_3 x_3} - \Sigma_{x_3 X_1}\Sigma_{X_1 X_1}^{-1}\Sigma_{X_1 x_3}.\end{aligned} \tag{A.7}$$

Applying the same method, we can also derive $\Sigma_{\tilde{x}_{3|1}\tilde{x}_{2|1}}$ and $\Sigma_{\tilde{x}_{2|1}\tilde{x}_{2|1}}$ as

$$\begin{aligned}\Sigma_{\tilde{x}_{3|1}\tilde{x}_{2|1}} &= \mathbb{E}\left[\left(\tilde{x}_{3|1} - \mathbb{E}[\tilde{x}_{2|1}]\right)\left(\tilde{x}_{3|1} - \mathbb{E}[\tilde{x}_{2|1}]\right)^{\intercal}\right] \\ &= \Sigma_{x_3x_2} - \Sigma_{x_3X_1}\Sigma_{X_1X_1}^{-1}\Sigma_{X_1x_2},\end{aligned} \tag{A.8}$$

$$\begin{aligned}\Sigma_{\tilde{x}_{2|1}\tilde{x}_{2|1}} &= \mathbb{E}\left[\left(\tilde{x}_{2|1} - \mathbb{E}[\tilde{x}_{2|1}]\right)\left(\tilde{x}_{2|1} - \mathbb{E}[\tilde{x}_{2|1}]\right)^{\intercal}\right] \\ &= \Sigma_{x_2x_2} - \Sigma_{x_2X_1}\Sigma_{X_1X_1}^{-1}\Sigma_{X_1x_2}.\end{aligned} \tag{A.9}$$

Now, plugging Eqs. (A.7), (A.8) and (A.9) into Eq. (A.2), we obtain

$$\begin{aligned}\mathbb{V}(x_3 \mid x_1, x_2) = \Sigma_{x_3x_3} &- \Sigma_{x_3X_1}\Sigma_{X_1X_1}^{-1}\Sigma_{X_1x_3} \\ &- \frac{\left(\Sigma_{x_3x_2} - \Sigma_{x_3X_1}\Sigma_{X_1X_1}^{-1}\Sigma_{X_1x_2}\right)^2}{\Sigma_{x_2x_2} - \Sigma_{x_2X_1}\Sigma_{X_1X_1}^{-1}\Sigma_{X_1x_2}}.\end{aligned} \tag{A.10}$$

Notice that $\Sigma_{X_1x_2}$ and $\Sigma_{X_1X_1}$ have been calculated by Eq. (A.1). Furthermore, the denominator in Eq. (A.10) is nothing else but the conditional variance of x_2 given X_1, which has also been calculated by Eq. (A.1).

Adding x_3 into the X_2, the conditional variance of x_4 given X_3 ($X_3 = X_2 \cup x_3$) can be derived with the same procedure as

$$\begin{aligned}\mathbb{V}\left(x_4 \mid X_2, x_3\right) &= \Sigma_{\tilde{x}_{4|2}\tilde{x}_{4|2}} - \Sigma_{\tilde{x}_{4|2}\tilde{x}_{3|2}}\Sigma_{\tilde{x}_{3|2}\tilde{x}_{3|2}}^{-1}\Sigma_{\tilde{x}_{3|2}\tilde{x}_{4|2}} \\ &= \Sigma_{x_4x_4} - \Sigma_{x_4X_2}\Sigma_{X_2X_2}^{-1}\Sigma_{X_2x_4} \\ &\quad - \frac{\left(\Sigma_{x_4x_3} - \Sigma_{x_4X_2}\Sigma_{X_2X_2}^{-1}\Sigma_{X_2x_3}\right)^2}{\Sigma_{x_3x_3} - \Sigma_{x_3X_2}\Sigma_{X_2X_2}^{-1}\Sigma_{X_2x_3}}.\end{aligned} \tag{A.11}$$

where the denominator in Eq. (A.11) can be directly obtained from Eq. (A.2). Furthermore, $\Sigma_{x_3X_2}$, $\Sigma_{X_2x_3}$ and $\Sigma_{X_2X_2}$ can also be expressed as

$$\Sigma_{x_3X_2} = \Sigma_{X_2x_3}^{\intercal} = \begin{bmatrix}\Sigma_{x_3x_1} & \Sigma_{x_3x_2}\end{bmatrix}, \tag{A.12}$$

$$\Sigma_{X_2X_2} = \begin{bmatrix} \Sigma_{X_1X_1} & \Sigma_{X_1x_2} \\ \Sigma_{x_2X_1} & \Sigma_{x_2x_2} \end{bmatrix}. \tag{A.13}$$

Notice that all of entries in $\Sigma_{x_3X_2}$, $\Sigma_{X_2x_3}$ and $\Sigma_{X_2X_2}$ are from previous calculation. In addition, Equation (3.26) can also be derived using the same procedure.

In summary, this appendix presents the procedure to derive the proposed sequential GPR. By applying the proposed method, the computation complexity can be reduced to $O((n-1)^3)$