



HAL
open science

In-hand robotic tactile object recognition

Alex Vásquez

► **To cite this version:**

Alex Vásquez. In-hand robotic tactile object recognition. Robotics [cs.RO]. Université Pierre et Marie Curie - Paris VI, 2017. English. NNT : 2017PA066619 . tel-02408329

HAL Id: tel-02408329

<https://theses.hal.science/tel-02408329>

Submitted on 13 Dec 2019

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

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

A dissertation submitted to

PIERRE ET MARIE CURIE UNIVERSITY

Sorbonne Universités, Université Pierre et Marie Curie, Paris VI

Doctoral school of mechanics, acoustic, electronic and robotic science of Paris
*École doctorale des sciences mécanique, acoustique, électronique et robotique de
paris*

in partial fulfillment of the requirements for the Ph.D. degree

in

Robotics

by

Alex VÁSQUEZ

In-hand Robotic Tactile Object Recognition

to be defended on December 11th, 2017

Committee

Philippe FRAISSE	Professor at University of Montpellier 2	Referee
Daniel SIDOBRE	Senior Lecturer at the University Paul Sabatier	Referee
Kaspar ALTHOEFER	Professor at Queen Mary University of London	Examiner
Agnes ROBY-BRAMI	Researcher at Pierre et Marie Curie University	Examiner
Kevin BAILLY	Senior Lecturer at Pierre et Marie Curie University	Examiner
Véronique PERDEREAU	Professor at Pierre et Marie Curie University	Supervisor

Abstract

Robotic anthropomorphic hands are mostly used to reproduce the human dexterity in manipulation. Beyond the mechanical and control challenges that this represents, perceptive knowledge of the environment with which the hand interacts is key to ensure that dexterity is achieved. In this sense, tactile object recognition has become an important asset for manipulation systems. Regardless of the advances in this domain, it continues to be a valid subject of research today.

In this thesis, we propose a method to enable a robotic hand to quickly understand the geometrical nature of an object that has been handled by it. We intend to exploit every piece of information available from the moment the interaction between the hand and the object begins. Thus, aside from the static data obtained once the object has been fully grasped, the movements of the hand during the grasp execution will also be exploited.

As a first contribution, we propose the *proprioceptive shape signature*. This descriptor, based solely on proprioceptive data, is invariant to the size and pose of the object within the hand and it contains information about the global shape of the object almost as soon as the grasp execution ends.

As a second contribution, we propose a tool to extract information about the grasped object from the dynamic data generated during the grasp execution. For this, the movements of the fingers during the grasping process will be interpreted based on the grasp strategy.

Finally, we present a method to perform sequential object shape identification based on a collection of random forests. This method allows to update the recognition model as new shapes are desired to be identified. Thus, the time-consuming process of training the model from scratch is avoided.

Experiments to validate these methods are carried in both simulated and real environments. Synthetic data is used to train the learning algorithms, so that the tedious and time-consuming of data gathering process with real hands is avoided. Furthermore, a single grasp is needed to perform shape identification, contrary to most methods which need multiple grasps.

Contents

Abstract	i
Contents	ii
List of Figures	v
List of Tables	x
1 Introduction	1
1.1 This thesis	2
1.2 Outline and contributions	3
1.3 Related Publication	4
Journal articles	4
Conference publications	4
2 Proprioception for Object Shape Identification	6
Shape identification with touch sensors	7
Shape identification with proprioceptive sensors	8
Proposed contributions in this chapter	11
2.1 Human proprioception	12
2.2 Robotic proprioception for tactile shape identification	13
2.2.1 Proprioceptive data representation for a robotic hand	14
Finger configuration image <i>FCI</i>	14
2.2.2 Reproduction of enclosure grasping	15
2.3 Proprioceptive shape signature	16
2.3.1 Signature generation process	16
2.3.2 Proprioceptive signatures from the <i>FCI</i>	19
2.3.3 Signature enhancement process	19
2.4 Experimental Setup	20
2.4.1 Test shapes	21
2.4.2 Simulated Setup	21
Grasp	22
Data collection process	22
2.4.3 Real-hand setup	22
Grasp	24
Data collection process	24
2.5 Experimental evaluation of the signatures	24

2.5.1	Signature generation	25
2.5.2	Signatures invariance	27
2.5.3	Defining ranges for size and position	29
2.5.4	Estimation of the object angle of rotation	30
2.5.5	Modified shapes	33
2.6	Experimental evaluation of shape identification with signatures	33
2.6.1	Data division	34
2.6.2	Self-Organizing Maps learning algorithm	34
2.6.3	Support Vector Machine learning algorithm	37
	Position and shape recognition rates	37
2.7	Conclusion	39
3	Temporal Analysis of Proprioception	41
	Dynamic information from an exploration process	42
	Dynamic information from a grasping process	43
	Proprioceptive data as source of dynamic information	44
3.1	Temporal analysis of the grasping process	44
3.1.1	Bases of the temporal analysis	45
3.1.2	Recording the time-related information	46
3.1.3	Extracting information from image G	47
	Derivates	47
	Binarization	48
3.2	Non-contact finger dropout using temporal information	50
3.2.1	Position-based method	51
	Threshold definition	51
	Phalanx coupling dependence	52
3.2.2	Dynamic-based method	52
	Dynamic behavior of G_{bin}	53
3.2.3	Evaluation of the non-contact finger detection methods	54
	Simulation	54
	Real Platform	57
3.3	Proprioceptive signature correction	60
3.3.1	Modified procedure for signature generation	60
3.3.2	Evaluation of signature correction process	62
	Impact on the identification rate	62
	Shape identification with corrected signatures	62
3.4	Conclusion	66
4	Sequential Recognition of In-Hand Object Shape	69
4.1	Reducing the amount of instances	70
4.1.1	Tactile data interpretation	71
	Second-order information	71
	First-order information	72
4.1.2	Proprioceptive vs. contact normal	73
4.2	Addition of new instances to the training set	75
4.3	Neural Forests Collection	76
4.3.1	Multi-class shape identification	76

4.3.2	Soft trees with probabilistic split nodes	77
4.3.3	Online learning with recursive backpropagation	78
4.3.4	Sequential training	79
4.4	Experimental set-up	79
4.4.1	Data	80
4.4.2	Setting up the NF	80
4.5	Results and Discussion	81
4.5.1	Unimodal vs. multimodal shape recognition	81
4.5.2	Offline vs. sequential training	82
4.5.3	Shape identification with sequential training	84
4.5.4	Real hand results	85
4.6	Conclusion	86
5	Conclusions and future work	88
5.1	Proprioception for object-related information extraction	88
5.1.1	Static proprioceptive data	89
	Further work on proprioceptive shape signatures	89
5.1.2	Dynamic proprioceptive data	90
	Dynamic proprioceptive data for shape identification	90
5.2	Sequential shape identification using multi-modal inputs	91
5.3	Summary	92
A	From joint angles to central angles	93
B	Shadow Hand	94
B.1	Shadow Dexterous Hand	94
B.1.1	Sensors	95
B.1.1.1	Joint angle sensors	95
B.1.1.2	Joint force sensors	95
B.1.1.3	Tactile sensors	95
B.1.2	Control system	96
B.2	Shadow Hand Simulation Model	96
	Bibliography	97

List of Figures

1.1	Schematic drawing of a robot hand equipped with several types of contact sensor [1].	3
2.1	Methods using tactile object identification with sensor arrays.	8
2.2	Robotic hand and grippers equipped with proprioceptive sensors.	8
2.3	Methods combining proprioceptive and tactile data.	9
2.4	Benchmarking of the presented method.	11
2.5	Depiction of six manual exploratory procedures and their associated object property to be identified (in parenthesis) [2].	13
2.6	Image superposed to the hand so each pixel corresponds to one joint. Consequently, each column represents one finger and its pixels represent the joints of the finger. The thumb is not included in this representation.	14
2.7	Finger configurations (<i>Upper images</i>) and their corresponding Finger Configuration Images (lower images).	15
2.8	Lateral view of a finger. Red lines show the areas of contacts C_i onto which the contacts are detected. θ_i are the joint angles. T_i are the efforts on the joints i	16
2.9	Lateral view of a finger conformed to a circle. The red points represent the contact points of the phalanges. These points form an arc shown in black that subtends the angle θ' . θ is the angle between the phalanges forming the arc on the circle.	17
2.10	(A) Concentric circles showing the central angles of their corresponding intercepted arcs (\widehat{AB} and \widehat{CD}). Both arcs have the same length. (B) Continuous case of Fig. 2.10a. Infinite sequence of circles of different radii forming a shape (gray) and the manifold containing the arcs of the same length formed on each circle (blue). (C) Central angles subtended by their arcs contained in the manifold along the central segment $\overline{OO'}$	18
2.11	Signature modifications with the size and pose of the object.	19
2.12	Signature enhancement procedure.	20
2.13	Set of test shapes. The blue manifold on top of them is the result of equivalent arcs formed on the circles contained in the shape. Below each shape, the corresponding signature obtained by extracting the central angles from the manifold with respect to the central axis $\overline{OO'}$. The procedure was shown in Fig. 2.10.	21
2.14	Simulated models of the shapes on Gazebo.	22
2.15	Schema showing how the angle α was varied to rotate the shapes with respect to the normal to the palm.	22

2.16	Objects used for real experiments. In Set 2 , some of the objects were grasped on different sites to get different shapes. This is indicated by rectangles on those objects. Yellow is used for hyperboloid, red for cylinder and blue for cone.	23
2.17	Object angle estimation. Yellow lines correspond to the hand axis. Red line to the object axis. The angle of rotation is measured between the horizontal axis of the hand and the axis of the object.	24
2.18	Signatures obtained on both simulated (<i>upper row</i>) and real objects: Set 1 (<i>lower row</i>) and Set 2 (<i>middle row</i>). <i>from left to right</i> : cone, torus, sphere, hyperboloid, cylinder.	26
2.19	Measured signatures of the (a) cone, (b) torus, (c) sphere, (d) hyperboloid ,(e) cylinder after being enhanced. The signatures in <i>blue</i> the signatures that look like their corresponding theoretical pattern. The <i>gray</i> ones are the ones that not. In order to determine the similarities, the threshold difined in section 2.5.2 was used.	27
2.20	Dynamic Time Warping.	27
2.21	DTW-distances between generated signatures and theoretical signature patterns: (A) Simulation. (B) with real hand.	28
2.22	Integrals of the hyperboloid and cylinder resulting from the signature enhancement procedure. The dashed line is the threshold defined to distinguish the signatures of these two shapes. This threshold was computed using the mean of the mean values of the integrals of each shape. (a) Simulation. (b) with real hand.	29
2.23	Identification rate of rotated objects with respect to the angle of rotation.	29
2.24	Illustration of the location of the manifold formed by the fingers adapting to a torus located near the palm and the corresponding generated signature. The shape is cross-sectioned to facilitate the view.	30
2.25	Measured angle of the signatures versus the actual angle of rotation of the object in simulation.	31
2.26	Measured angle of the signatures versus the actual angle of rotation of the object in the real platform. The blue lines correspond to the linear regression model.	32
2.27	DTW distances computed for an ovoid and a hectagon pyramid. The theoretical signatures of a cone, torus, sphere, hyperboloid and cylinder were used.	33
2.28	Confusion matrices obtained with SOM using the simulated training and test sets.	35
2.29	Weight neighbor distances of the maps obtained with SOM algorithm using the simulated data.	36
2.30	Results of SOM using the real data sets.	36
2.31	Object angle estimation. Yellow lines correspond to the hand axis, red line to the object axis. The angle of rotation is measured between the horizontal axis of the hand and the axis of the object.	37
2.32	SVM results based on SI using simulated and real data sets.	38
2.33	SVM results with respect to the position of the object. SI are used as inputs from simulated data.	38
2.34	SVM results with respect to the size of the object. SI are used as inputs from simulated data.	38

3.1	(A) Setup used by Moll et al. in [3]. Two flat surfaces attached to a fixed point on one side and a robotic arm on the other. As the robotic arm moves, it makes the angle between the flat surfaces change. (B) Cross section of a robot finger with the stress sensors used by Howe et al. in [1]. (C) Neural-dynamic model used by Strub et al. in [4].	42
3.2	(a) Generation of the image G overtime. As the hand adapts to the object, the rows of G are been filled with the difference between the minimum and maximum gray scale values of the columns of the FCI	47
3.3	(A) Let us concentrate on the column on the right of the image G . In this figure, the illustration shows an example of what the image G would look like after the grasping execution is over. Each one of the signaled areas has its own meaning. <i>Area 1</i> : this area corresponds to the movement of the finger to reach the contact. <i>Area 2.1</i> : The finger adapts to the shape. <i>Area 2.2</i> : The finger has already adapted but other fingers haven't and the grasp execution continues. <i>Area 3</i> : Rows not filled because the grasp execution ends before reaching them. (B) Resulting binarized image after applying the Sobel filter to the image G	48
3.4	In the images (A) and (B) are illustrated the situations in which the lateral displacements of the shapes with respect to the hand prevent one finger to enter in contact with the object. The image (C) shows a sphere which is both too small and displaced so the little finger does not enter in contact with it.	50
3.5	(a) Generation of the image G_{bin} overtime by computing it at the frequency f_G as the hand adapts to the object.	53
3.6	Histograms illustrating the performance of the dropout non-contact finger methods based on the FCI temporal analysis for (A) position-based and (B) dynamic-based methods. The cases in which the methods detected a non-contact finger are shown in the <i>detected</i> bar, whereas the cases when no contact was measured on one finger are depicted in the <i>measured</i> bar. The blue proportion of the bars corresponds to cases when the methods performed accurately and the gray proportion, when they did not.	56
3.7	(A) Relative frequency distribution for the dynamic-based method with respect to the position-based method. (B) Cumulative frequency distribution. Blue bars correspond to the percentage of samples that saved more than 2.2 seconds using the dynamic-based method when compared to the position-based one.	58
3.8	Grasps performed to validate the non-contact finger dropout methods. Three objects were used in these experiments: (A) Cone, (B) Sphere and (C) Cylinder.	58
3.9	Histograms illustrating the performance of the dropout non-contact finger methods based on the FCI temporal analysis for (A) position-based method and (B) dynamic-based method. The cases in which the method detected a non-contact finger are shown in the <i>detected</i> bar, whereas the cases when no contact was measured on one finger are depicted in the <i>measured</i> bar. The blue proportion of the bars corresponds to cases when the methods performed accurately and the gray proportion, when they did not.	58

3.10	(A) Relative frequency distribution for the dynamic-based method with respect to the position-based method. (B) Cumulative frequency distribution. Blue bars correspond to the percentage of samples that saved more than 2.2 seconds using the dynamic-based method when compared to the position-based one.	59
3.11	Signature correction by dropping out information from non-contact fingers. On the <i>left</i> , all fingers are involved in the signature generation process. On the <i>right</i> , non-contact fingers are dropped out and the signature is corrected.	61
3.12	Confusion matrices showing the recognition rates. The upper row corresponds to the simulated results. The lower row corresponds to the results obtained with the real platform. The first column corresponds to the results obtained using SVM_{pos} . The second columns corresponds to the ones obtained with SVM_{dyn}	63
3.13	Confusion matrices showing the identification rates for non centered objects. The upper row corresponds to the simulated results. The lower row corresponds to the results obtained with the real platform. The first column corresponds to the results obtained using SVM_{pos} . The second columns corresponds to the ones obtained with SVM_{dyn}	64
3.14	Euclidean distances between the expected patterns and the measured signatures in simulation. The blue bars are obtained by comparing the measured signatures with their corresponding expected patterns. The gray bars correspond to the DTW-distances between the measured pattern and the non-corresponding patterns. The dashed-line is the threshold (55) established in chapter 2 for signature correspondence.	65
3.15	Confusion matrices showing the identification rates for non centered objects. The upper row corresponds to the simulated results. The lower row corresponds to the results obtained with the real platform. The first column corresponds to the results obtained using SVM_{pos} . The second columns corresponds to the ones obtained with SVM_{dyn}	65
3.16	Euclidean distances between the expected patterns and the measured signatures in the real platform. The blue bars are obtained by comparing the measured signatures with their corresponding expected patterns. The gray bars correspond to the DTW-distances between the measured pattern and the non-corresponding patterns. The dashed-line is the threshold (55) established in chapter 2 for signature correspondence.	66
4.1	Recognition rates for uni- and multimodal data.	71
4.2	Geometric information obtained from a finger as it explores a surface s of an object: finger position $\mathbf{p}(s)$ (zeroth-order), contact normal $\mathbf{n}(s)$ (first-order), and local curvature shown in red beneath the finger $\mathbf{c}(s)$ (second-order). Best view in color.	71
4.3	Contact normal and localization used in description-based representations.	73
4.4	Twenty-one hand scheme.	73
4.5	The values from literature plotted in log-log scales. The gray data points indicate thresholds obtained from zeroth-order information only. The black data points were obtained with real shapes. The gray lines describe the model itself. This data can be found in: Pont 1999 [5], van der Horst 2007 [6], Gordon 1982 [7], Henriques 2003 [8], Drewing 2006 [9].	74

4.6	Methods to update a discriminative appearance model: (a) Using a single positive image. As one can notice, this image is not centered on the object perfectly. (b) Using several positive images. This is a too rigid condition for a classifier to learn a boundary. (c) Using one positive bag consisting of several image patches to update a MIL classifier [10].	76
4.7	This image shows one tree of the hybrid random forests and neural networks.	78
4.8	Illustration of the shape rotation with respect to the z-axis of the hand. From left to right: 90°-Rotated Cylinder, 90°-Rotated Cone.	80
4.9	Objects used for tests with the real platform.	80
4.10	Average confusion matrices obtained using NF. Recognition rates are shown in %.	82
4.11	Confusion matrix for multimodal shape identification. Zeroth- and first-order information were concatenated in the feature vectors.	83
4.12	Recognition rate (%) obtained by computing the mean of ten tests. For each test, the order in which the objects were added to the training set was varied randomly. The tree corresponding to each object is updated 1500 times at the end of the training. Variance of the results is also shown. Outliers are shown in red for both training strategies.	83
4.13	Number of updates per tree vs. the number of objects for offline (all-at-once) and online (sequential) training.	84
4.14	Average confusion matrices obtained using NF and sequential training.	84
4.15	Matrix of recognition rate disturbances between shapes. Shapes on the vertical axis (Disturbances) are the new shapes being added to the training set. Shapes on the horizontal axis (Disturbed) are the shapes already present in the set being affected (Best view in color).	85
4.16	Confusion matrix obtained with a sequentially trained NF. Training was performed using simulated data. Data for testing was collected with the real Shadow hand.	86
5.1	Signatures measured in simulation for each one of the shapes used. The blue markers depict the points on the signatures. Those points correspond to the fingers that adapted to the same radius at the same position.	90
5.2	Images G_{bin} obtained for the different shapes in a simulated environment. From left to right: cone, torus, sphere, hyperboloid, cylinder.	91
A.1	Tangents l_1 and l_2 and the cords \overline{OA} and \overline{OC} forming a quadrilateral.	93
B.1	Shadow Hand used in experiments.	94
B.2	ATI nano 17 sensors adapted to the shadow hand.	96
B.3	Gazebo model of the Shadow Hand.	96

List of Tables

2.1	Tested dimensions and positions for each object. The column <i>Shape Dimension</i> shows the dimensions varying for each shape. In the column <i>Size</i> , the intervals of dimensions that were tested and the change rates. Images in the column <i>Position</i> correspond to the positions where the shapes were located. All dimensions are shown in <i>cm</i>	23
2.2	Ranges of size and positions in the hand attained for each one of the shapes.	31
B.1	Maximum and minimum joint angles that the joints from the index, middle, ring and little finger can reach. The other DOF of this hand are not shown here because they are set to a particular value. If needed, they will be specified in the corresponding chapters.	95

Chapter 1

Introduction

In-hand object manipulation, whether it is performed by a human being or by a robot, requires visual and haptic information in order to be performed efficiently. This information offers fundamental knowledge about the geometrical and physical characteristics of the manipulated object. Without it, the manipulation task becomes almost impossible to do by a robot, unless a perfect model of the environment in which it is performed is available.

Visual information is not absolutely required for all manipulation tasks. The lack of visual information certainly reduces the dexterity with which it can be accomplished, but some dexterity can still be achieved. This is the case when an individual writes a text message on his or her cellphone without looking at the screen or when someone plays with a straw with his or her fingers while talking to a friend at a coffee shop. The absence of haptic information would likely render these previously mentioned tasks nearly impossible.

In general, manipulating objects without tactile input is a task unachievable. This sense provides us with information about the object that the visual system cannot, such as its weight or the location of its center of mass. Even if geometrical information could be available through vision, occlusions happen once the object is grasped. For humans, this might not be a problem, but it is one of the drawbacks when vision is used for this matter on robotic hands.

Given the importance of tactile sensing for manipulation tasks, it is more frequent nowadays to find robotic manipulation systems equipped with sensors to provide some sort of haptic feedback to the controllers. Despite the advances in sensor technologies, robotic hands are still far from performing dexterous manipulation tasks as efficiently as human beings. Of course, it would be absurd to reduce this problem to tactile sensor

technology only. Limitations related to other disciplines such as control and electromechanical design of the hand also contribute to making this task a current challenge in robotics. However, tactile perception is at the core of most of the limitations imposed by those disciplines. For example, most control strategies for dexterous object manipulation need tactile input.

Robotic haptic perception is, on its own, a large field with limitations that are related to aspects other than the sensors themselves. Proper treatment of the sensor data is also a concern of the robotic community and many publications continue to be issued in important journals on that matter. A proper treatment of this data would make it possible to identify the characteristics of the object fundamental in order to develop an efficient manipulation strategy.

Identifying the shape of the manipulated object and having cues of its localization within the hand enhance the dexterity of the robotic hand. Tactile object recognition has particularly received a lot of attention during the last decades thanks to advances in machine learning algorithms.

This thesis work focuses on that concern and presents a novel approach for in-hand object shape identification based on haptic information. Unlike most works in this field, the proprioceptive information will be intensively exploited. Analogies to the human haptic system will be made to support the various usages of the information coming from the haptic sensors on a robotic hand. Experiments in simulated and real environments are performed to validate proper performance of the proposed method.

1.1 This thesis

In this thesis, we propose a method to identify the global shape of an object that has been passed on to a robotic anthropomorphic hand. To picture the task that has been identified as the goal of this thesis, imagine that an object has been passed on to a sight blinded individual. By the end of the grasp execution, he or she will already have collected enough information about the object to decide what manipulation strategy is more appropriate to further explore it. All this information is provided by the haptic system and it is collected from the moment the interaction with the object starts, which is the moment when the object enters in contact with the individual's hands. Thus, both the process of grasping and the final position of our hand, in their own way, contribute to the process of information acquisition.

Tactile sensing plays a particularly important role in reproducing this task with an anthropomorphic hand. Thus, robotic hands are frequently equipped with sensing

devices. These devices endow the hand with a haptic-like system and increase their dexterity. Howe [1] listed them as follows: tactile sensor arrays, fingertip force/torque sensors, dynamic tactile sensors and joint angle sensors (Fig. 1.1). He states that the contact information needed for manipulation depends very much on the task wanted to be achieved. This statement is also valid for tactile object recognition as each one of those types of sensors would provide different information about the grasped object.

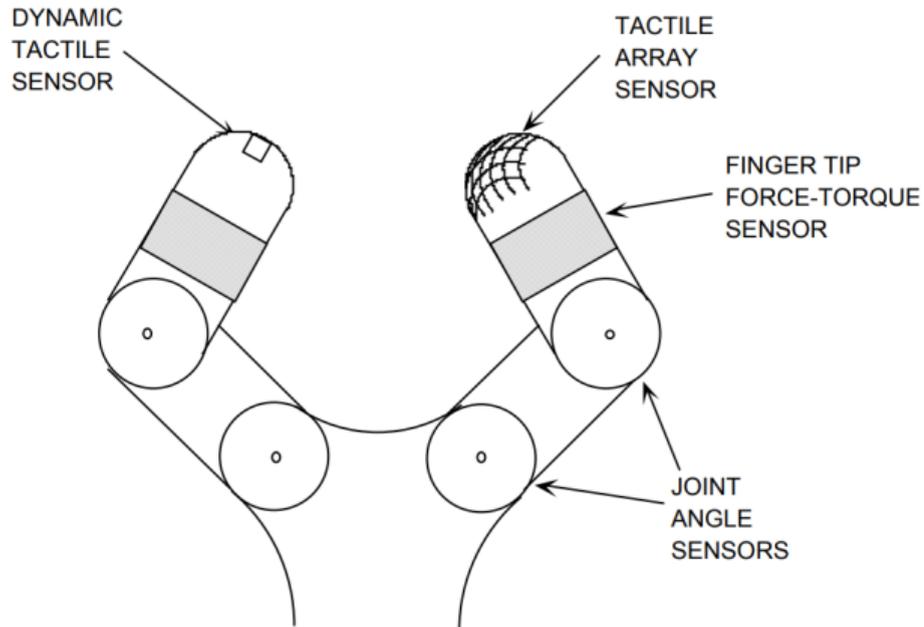


FIGURE 1.1: Schematic drawing of a robot hand equipped with several types of contact sensor [1].

For tactile object identification, the interpretation of the data coming from sensors used to be at the core of the research in the field. Much research work concentrated in building different representations or descriptors to identify different characteristics, mostly geometrical, of the object. However, with the latest advances of machine learning algorithms, this is almost no longer the case. The entire process of interpretation of the data is put on the learning algorithms. Despite the successful results obtained with this approach, improvement via this route push tactile recognition farther away from its original goal: providing input for the manipulation tasks.

Generally, one of the main purposes of this research is to endow a hand with haptic data representations that allow it to identify more than just shape information of the grasped object, but also estimate some information about its pose within the hand. Thus, the data issued from haptic sensors installed on the robotic hand can be fully exploited for both the identification task and subsequent manipulation tasks.

In order to achieve this goal, we will particularly exploit proprioception data. This type of information was our particular concern to make an actual contribution to the state-of-the-art in tactile object recognition. Proprioception has been very rarely used for tactile object recognition. When used, it has mostly a background role and it is combined with other sensing modes. Moreover, the notion of movement is better reflected on the proprioceptive data. Dynamic tactile data requires contact with the object, which proprioceptive movements can provide with cues of the object even if contact with it has not been reached.

1.2 Outline and contributions

- Chapter 2 describes the use of proprioception for shape identification in a robotic hand. Here, we propose the **proprioceptive shape signature**, which are a sort of shape descriptor generated from the proprioceptive data only. This descriptor is generated from a representation of the proprioceptive data of the robotic hand. A series of steps are presented to make these signatures invariant to the size and pose of the object in the hand. Experiments on both simulated and real environments are carried out and the quality of the signatures is measured. Finally, the signatures are tested for shape identification purposes.
- Chapter 3 concentrates on the exploitation of the **dynamic proprioceptive data** generated during the grasp execution. This method is exploited to detect the fingers that do not enter in contact with the grasped object. Experiments to validate the method are carried out. Moreover, the influence of dropping out the non-contact finger information from the feature vectors are analyzed.
- Chapter 4 presents a framework based on Random Forest algorithm for **sequential tactile shape identification**. This approach makes it possible to update the recognition model as new shapes are added to the identifiable set and training from scratch is avoided. In this chapter, the contact normals at the fingertips are measured and used as inputs for the RF algorithm. Experiments are carried out to compare the performance of both proprioceptive data and contact normal data, individually and combined.
- Chapter 5 concludes the thesis presenting some of perspectives for future work.

1.3 Related Publication

Journal articles

- Vasquez, A., Perdereau, V. (2017). Proprioceptive shape signatures for object manipulation and recognition purposes in a robotic hand. *Robotics and Autonomous Systems*.

Conference publications

- Vasquez, A., Kappassov, Z., Perdereau, V. (2016, October). In-hand object shape identification using invariant proprioceptive signatures. In *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on* (pp. 965-970). IEEE.
- Vasquez, A., Dapogny, A., Bailly, K., Perdereau, V. (2017, September). Sequential Recognition of In-Hand Object Shape using a Collection of Neural Forests. In *Intelligent Robots and Systems (IROS), 2017 IEEE/RSJ International Conference*. IEEE.

Chapter 2

Proprioception for Object Shape Identification

For human beings, tactile object recognition is an easy task to perform. According to the experiments presented by Klatzky et al. in [11], we are capable of recognizing both physical and geometrical properties of an object using tactile information. All this, with a remarkable accuracy as high as 94% and a rapidity of approximately 5s. Our ability to achieve this performance is mainly related to two abilities: tactile sensory perception and shape representation.

Tactile sensory perception relies on the somatosensory system. This system provides the brain with various information coming from many sensory channels [12]. The main sensory modes are touch, proprioception and temperature. For shape identification purpose, human beings exploit touch and proprioceptive information. Temperature sensory channels do not provide significant information about the shape of an object. Hence, they will not be considered in this work.

The **tactile shape identification** process is largely influenced by the **shape representation**. Without a proper representation, shape identification would be inaccurate even if the sensory perception functionality is intact. This was proved by Reed et al. [13] in a study about tactile agnosia (lack or loss of ability to recognize an object through touch). Results showed how a 65-year-old female subject, although having normal motor and intellectual skills, had issues in recognizing objects with her right hand due to an infarction on the left inferior parietal. In the conclusions, the researchers attributed this phenomenon to an impairment of the shape representation caused by a loss of tactile specific representations of shapes or high level shape features.

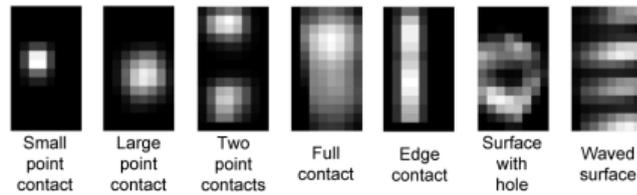
In robotics, attempts to reproduce these abilities can be found in the tactile object recognition literature. Many references document the efforts to endow robots with tactile sensory perception. In this regard, different types of sensors have been developed or adapted. Kappassov et al. [14] classified these sensors as touch (extrinsic) sensors and proprioceptive (intrinsic) sensors.

Shape identification with touch sensors

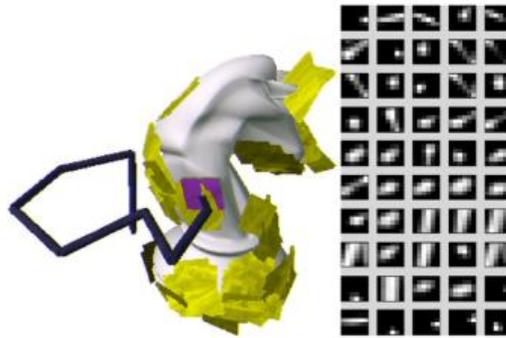
Touch sensors have received most of the attention; they have been largely diversified in terms of transduction methods. Some of them are based on the piezoresistive effect [15–18], which is a property of certain materials to change their resistance as they are mechanically deformed. Others are constructed using two electrodes separated by a dielectric material. This type of sensors takes the capacitance of this arrangement as a measure of the applied force/pressure [19–21]. Some other sensors exploit the piezoelectric effect, which is the property of certain materials to produce electric charges when subjected to a force/pressure [22–24]. Optical properties are also used in tactile sensor technologies. When so, the light reflection between two materials with different diffractive indices is exploited and changes in the light intensity are taken as a measure of the applied force/pressure [25–27]. There are some other methods implemented in tactile sensors, such as barometric measurements [28, 29], in which a liquid inside the tactile sensor is used as a propagation media for vibrations and a transducer measures the pressure value.

Many robotic anthropomorphic hands and manipulation systems are equipped with such touch sensors and several **shape representations** have been developed for the shape identification purpose based on the data issued from them. These representations can be classified into two categories: hand-crafted surface models and statistical patterns. The hand-crafted representations correspond to explicit geometrical models built from data that are matched to object descriptions (Fig. 2.1). In other words, patterns are created from *a priori* knowledge on the data generated by certain characteristics of the objects on the sensor arrays. In this sense, they mostly intend to build shape or surface models, such as edge, points, corners, etc., from pressure distributions on a sensor array [30–35]. The statistical representation exploit algorithms, such as bag-of-features (BoF), to classify the information from the pressure distributions [36–39]. These representations are not based on *a priori* knowledge on what the sensor data will look like, but a classification algorithm generates a model to identify the group to which each measurement corresponds. Both representations generally transform the pressure

distributions into images and profit from image processing techniques to detect image moments and/or create descriptors.



(A) Hand-crafted surface models from a Piezoelectric tactile sensor array [24]



(B) On the left, a serial arm with a tactile sensor array at the end-effector (in purple) palpating a shape. The yellow patches depicts the places where the sensor array has entered in contact with the object. On the right, sensor images extracted from contacts of the sensor array with the object. These images are used for a statistical based method [36]

FIGURE 2.1: Methods using tactile object identification with sensor arrays.

Shape identification with proprioceptive sensors

Regarding the **proprioceptive sensors**, the diversification of transduction methods is less important. Some of these sensors are shaft encoders based on either magnetic, optical or resistive properties. Magnetic encoders are composed of a magnetic disk and a hall sensor that detects the changes of the magnetic field as the disk rotates [40]. Such sensors are used in several robotic hands (Fig. 2.2), as the iCub humanoid robot hand [41], the Shadow Hand [42], the Utah/MIT dexterous hand [43] and the fluidic robotic hand developed at the Institute of Applied Computer Science of Karlsruhe [44]. Depending on the measurement method, encoders can be divided into two types: absolute and incremental. The absolute encoders provide with a unique position from the moment they are switched on. To the authors knowledge, this is the most common encoder embedded in robotic hands. Such is the case of the Allegro Hand [45], the Shunk SDH gripper [46], and NASA hand [47]. Incremental encoders, which provide electric pulses as the joint rotates, can also be found in some robotic hands [48, 49]. Other manipulation systems use accelerometers embedded on the grippers' links and process

their data through an Extended Kalman Filter to update the joint angles [50, 51]. Joint forces are also contributing to proprioception and most of the aforementioned hands count on their measurements too.

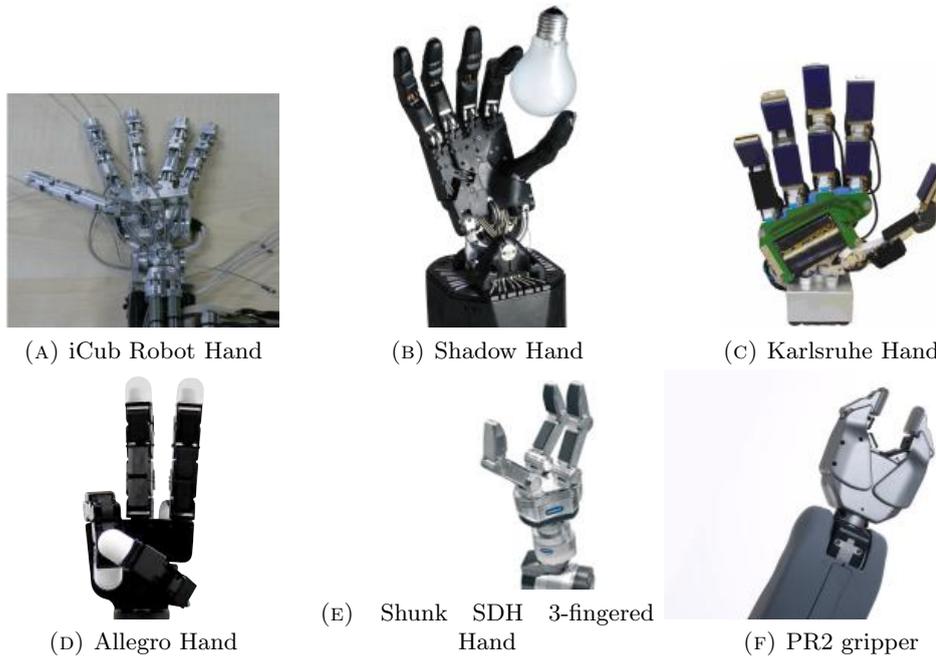


FIGURE 2.2: Robotic hand and grippers equipped with proprioceptive sensors.

Details about these proprioceptive sensors are rarely given in tactile shape identification literature and **proprioceptive shape representations** have not received as much attention as touch shape representations. Thus, proprioceptive data has played a background role in tactile shape identification; when found in literature, proprioceptive information is mostly combined with tactile information (Fig. 2.3) and several approaches can be cited as examples. One of the most common approaches for shape representation estimates contact point locations using kinematics information, which allows to generate a point cloud and match it to a hand-crafted pattern (3D shape models, or curves) [52–56]. Some works following a similar approach with contact normals will be subject of discussion in chapter 3. Another approach consists in building statistical patterns from the raw values of the joint angles [38]. Another technique is to add these measurements to the feature vector in order to add more dimensions to the tactile information [57–59]. Proprioception can also be found in combination with other sensory modalities, like vision [60] or hearing [61].

Although proprioception is largely used by humans to perform shape identification, **proprioceptive-only shape representations** are not abundant in the literature. Nearly all references to this are on statically-crafted representations obtained through learning algorithms. Johnsson [62] implemented Self-Organizing Maps (SOM) based on joint angle measurements from resistive encoders embedded on the LUCS Haptic Hand

II [63]. The author intended to identify two shape categories (cylindric and cubic) and estimate the size of the objects. During the experiments, five objects of different sizes from each shape category were used for training, resulting in a 50-sample training set. Results showed that the system was capable of grouping the objects according to their shape and ordering them according to their sizes. The system also showed to be able to generalize its adquired knowledge to six new objects not included in the training set. Even if the system was capable of identifying specific objects, its performance would have the tendency to diminish as the number of shape categories increases (examples to illustrate this phenomenon will be presented in chapter 4).

The same combination of SOM with raw values of joint angles was implemented by Ratnasingam et al. [64]. In this work, 25 objects from five different shape categories (cylindrical, cubic, disk, spherical, irregular) were identified. During the data collection process, each object was located in the hand on three different poses or orientations and 20 grasps were performed for every object in every orientation. 1500 samples were obtained in total from which 750 were used for training. To generate the feature vector, the authors concatenated three grasps of the same object in a vector. Each one of those grasps corresponded to one of the three chosen orientations of the object. This strategy to build the feature vector has a main drawback: the order in which the grasp data is concatenated needs to correspond to the order used in the training set. Thus, the sequence in which the grasps are performed has to be known beforehand. One of the important contributions of this work was the proof of usability of proprioceptive data for object shape identification. Results showed that 25 objects could be identified with samples noised up to 40dB. Okamura et al. [65] used both tactile and proprioceptive data to identify a group of shape patterns during tactile exploration and concluded that proprioceptive information was more robust to noise than tactile information. Furthermore, they concluded that tactile information was not needed to recreate object shape.

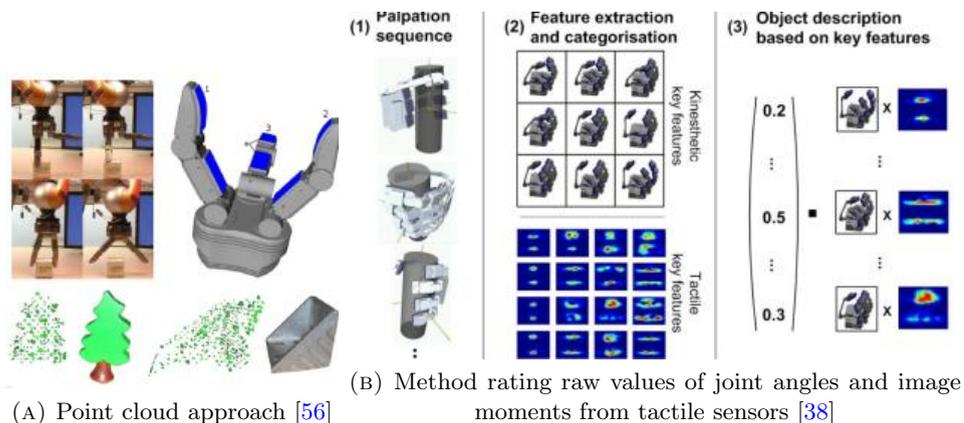


FIGURE 2.3: Methods combining proprioceptive and tactile data.

Proprioceptive data represented by joint torques have also been used for object recognition. That is the case in Bergquist et al. [66]. In this work, authors recorded joint torques during interactions with objects like lifting, shaking, dropping, crushing and pushing and used those recordings as inputs for a SOM. This was implemented on a Barrett arm equipped with a 3-fingered hand. Given a training set of 2500 samples coming from interactions with 50 objects, the algorithm showed to be able to recognize an object with an accuracy larger than 90% after five interactions with the object. Even if proprioception was used in this work, shape was not identified but specific object recognition prioritized. Most of these methods rely on learning algorithms to statistically separate the data and very few references on model-based patterns are found for proprioception-only methods. One of those is the work of Faria et al. [67], who correlated hand configurations during in-hand object exploration to generate hypotheses on a potential candidate object from a set.

Most of these methods achieve good results with accuracy rates higher than 85%. However, they present certain **limitations** in the context of tactile object identification. The first limitation is mostly related to the type of data used for the training process of the machine learning algorithms. Methods that use statically-crafted patterns based on raw values of the joint angles/torques require large training sets for the recognition algorithms to attain such results. Needing large training sets is not a problem *per se*, but building them is in the context of tactile object shape identification; getting large sets of data with a robotic hand can be a tedious and time consuming process. Furthermore, mechanical robustness could represent a limitation for some robotic hands as extended periods of usage might cause some damage to some of the mechanical parts. The second limitation is related to the recognition approach. Some of the cited works prioritize specific object recognition over shape identification. Even if tests on large numbers of objects are presented in each of the cited references, the objects used tend to have very similar global shapes. Therefore, multiple samples of each individual object need to be present in the training set. This worsen the first limitation as the training data set is consequently enlarged. Furthermore, a detrimental effect on the performance of the recognition algorithm might be observed for specific object recognition if the number of objects to be identified increases, as explained in [62]. The recognition approach is limited by another factor: the number of interactions needed before efficiently identifying the object. Some of the cited works need several grasps or interactions with the object to achieve the reported recognition rates.

Proposed contributions in this chapter

In this chapter, we present a method to perform global shape identification using a model-based shape representation solely based on the robotic hand proprioceptive data. With this approach, we intend to solve some of the aforementioned limitations of the cited methods for tactile object recognition based on proprioception. Fig. 2.4 illustrates the work flow of the method presented in this chapter. In this diagram, the main contributions of this method to the state of the art can be found:

- A representation of the proprioceptive data is created. To do so, human proprioception is emulated and the hand kinematics topology made intrinsic to this representation. This is an important contribution since it endows the robotic hand with a more intelligible representation of finger configurations and their movements.
- A model-based shape representation called *proprioceptive shape signature* is generated from the proprioceptive data. This is one of the main contributions of this thesis since all methods for tactile object recognition based on proprioceptive data use statistically-based shape representations. After passed through the block for disturbance suppression, these signatures are capable to describe the global shape of an object independently on its size and pose within the hand.

Based on these two main contributions, particularly the *proprioceptive shape signature*, several limitations of the cited methods can be resolved. First, global shape identification is prioritized over specific object recognition, contrary to the cited proprioception-based methods. This decreases the burden put on recognition algorithms when specific object recognition is performed, specially for large sets of objects. Identifying the global shape of an object is also convenient when a manipulation task is wanted to be performed since

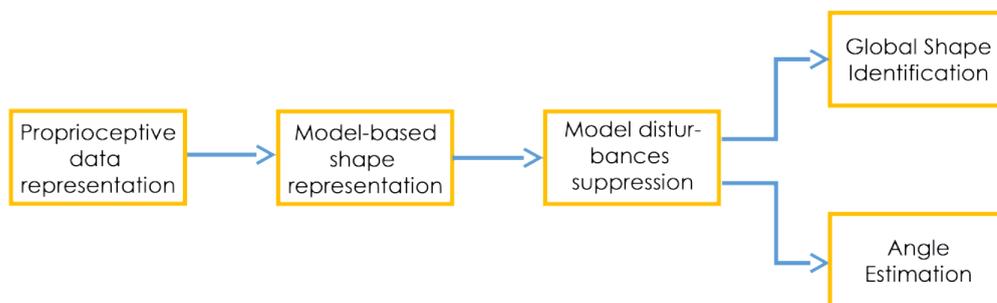


FIGURE 2.4: Benchmarking of the presented method.

geometrical information about the object is more important than specifically recognizing the object. Second, objects can be identified with a single grasp while most methods, both proprioception-based and tactile-based, need to perform more than one grasp. Third, as signatures are based on a data representation that respects the kinematics topology of the hand, some pose parameters can be estimated. In this thesis, we are going to specifically focus on the estimation of the angle of rotation of the object with respect to the normal to the palm.

Even if global shape identification is not a contribution by itself for tactile object recognition, it is for the proprioception-based methods. It also has two advantages: First, the amount of data needed to train learning algorithms, such as SOM, is reduced with respect to methods that perform specific object recognition. Thanks to the invariance of the signatures, the training set does not need to contain information about every single object to be able to identify its shape. Second, global shapes can be easily modeled in simulation, contrary to objects that surround us. Thus, synthetic data can be used to generate the training set corpus. This helps avoiding the tedious process of data collection with the real hand.

In this chapter, all these methodological and technical contributions will be detailed. First, the proprioceptive representation of a robotic hand used will be exposed. Conditions will be defined for this representation to embed information about the shape of the object. The generation process of shape signatures will be explained in detail. Next, the analysis of the finger configurations will then be presented as a tool for shape recognition. Finally, simulated results will be presented along with experimental results. In both cases, the Shadow Hand was used.

2.1 Human proprioception

In order to generate the shape proprioceptive signatures, some theoretical bases need to be set. In particular, human proprioception and its functioning in the context of tactile shape identification need to be understood and properly modeled for its exploitation in robotic shape identification.

As explained at the beginning of this chapter, the human body is equipped with many sensory systems and the one responsible for providing tactile cues regarding the environment that surrounds us is the somatosensory system. The receptors that form this system respond to different types of stimuli and contribute to different modes of perception (cutaneous and kinesthetic). The cutaneous receptors are distributed all over the skin surface. They are capable of detecting different types of information, such as

touch, vibrations, pressure and temperature. The kinesthetic receptors are distributed along the muscles, tendons and joints. They are responsible for sensing not only the body position and movement, which can be defined as proprioception, but also, according to Proske et al. [68], tension or force, efforts and balance. In the following, we will use the term proprioception to refer to the joint angle positions only.

Both perception modes, tactile and proprioception, have their weighted importance in the different haptic functions. According to Lederman et al. [69], when the haptic function is tactile shape identification, the relevance of the information provided by each perception mode is variable and depends on the geometrical properties of the object that is to be identified. Among those properties, the size is the one that influences the most the weighting process of the modes: when the size of the object requires the whole hand for its manipulation, proprioception becomes an important input for global shape identification. Based on this fact, the method presented in this chapter will focus on recognizing shapes of objects larger than a fingertip.

The other factor contributing to how the importance of the perception modes is weighted is the exploration procedure. Lederman et al. [2] associated different exploration procedures used in tactile object recognition to the object property to be identified (Fig. 2.5). These relationships were obtained from analyzing which exploratory procedure was used in experiments with individuals when asked to identify the different properties of the object.

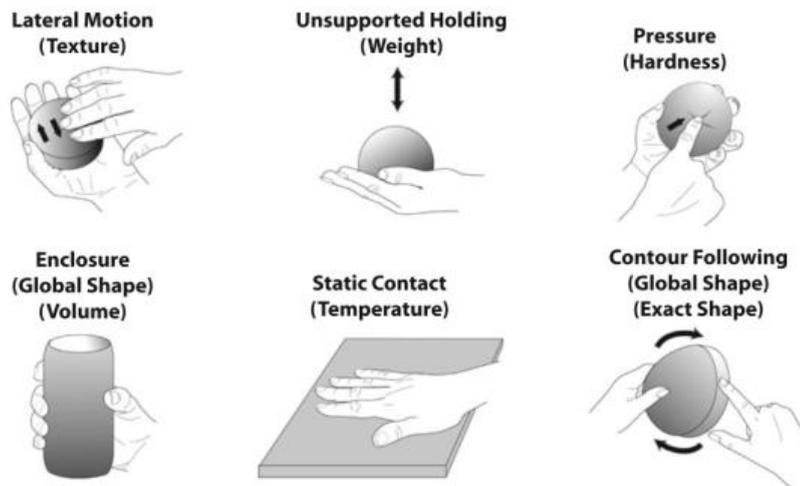


FIGURE 2.5: Depiction of six manual exploratory procedures and their associated object property to be identified (in parenthesis) [2].

Both touch and proprioception contribute to these strategies. However, proprioception becomes a particularly important input for shape identification when the exploratory procedure of *enclosure grasping* is employed. In this kind of grasp, the hand takes the object within its fingers enclosing the shape. In this scenario, proprioception carries

significant information about the shape of the object in the joint angles as their configurations describe it, to some extent. For this reason, the method presented in this chapter will reproduce an enclosure grasping with a robotic hand.

2.2 Robotic proprioception for tactile shape identification

Proprioceptive shape representations are almost absent in the literature. In this thesis work we want to reproduce, to some extent, the human use of proprioception for shape identification. This section will be dedicated to the procedures followed to achieve this, which are the main contributions of this chapter.

2.2.1 Proprioceptive data representation for a robotic hand

To endow a robotic hand with proprioception, a representation of its proprioceptive data needs to be created. As the goal is to perform shape identification, this representation should be based on kinesthetic data (joint angles). Based on the fact that human beings use mental images of the joint angles to detect features of the objects [70], our method will create an image to represent the joint angles of the hand. Since this representation will exhibit the finger configurations, we will call it the *Finger Configuration Image (FCI)*. The use of an image will also allow us to exploit image processing techniques to extract information from it, as detailed in further sections and chapters.

Since the *FCI* will represent the proprioception of the hand, it is important that it respects the definition of proprioception itself which implies the knowledge of the position of the neighboring parts of the body relative to each other. In the case of the robotic hand, it should represent the relative positions of the segments (phalanges) of the fingers with respect to each other. To fulfill this requirement, the *FCI* will be designed so that each of its pixels corresponds to a joint, as illustrated in Fig.2.6.

The thumb is discarded from this representation because it is generally in an opposed position with respect to the rest of the fingers when an object is grasped [71]. Moreover, according to Newell et al. [72], the information integrated across the other fingers yields a better representation of the surface of the object than the information gathered from the thumb. This makes the proprioceptive information coming from the thumb unreliable for shape identification. Thus, we will not include it in the *FCI*.

Given these theoretical basis about the use of proprioception for shape identification, the next section will introduce the methodology followed to generate the *FCI*.

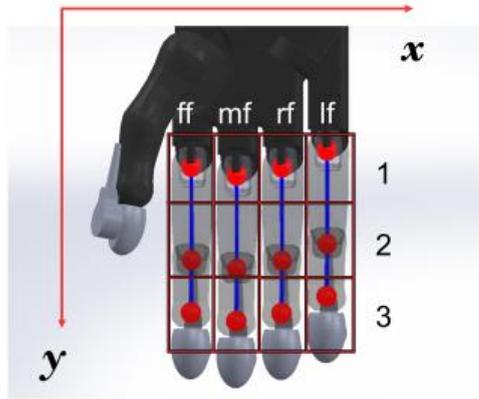


FIGURE 2.6: Image superposed to the hand so each pixel corresponds to one joint. Consequently, each column represents one finger and its pixels represent the joints of the finger. The thumb is not included in this representation.

Finger configuration image FCI

Filling the pixels of the FCI with the angles of the corresponding joints would not be different from using the raw values of the joint angles in cited methods for shape recognition. Even if some post signal treatment would allow to extract information from that image in such case, we want to replicate human proprioception. Consequently, the FCI is rather filled with information based on the kinematic topology of the fingers, which refers to the description of the configuration of its links and joints. In this approach, each finger is considered as a serial manipulator. Each joint angle is the sum of all joint angles downstream in the chain, the proximal phalanx being the first link in that chain. Based on this, the following equation is derived to fill the FCI :

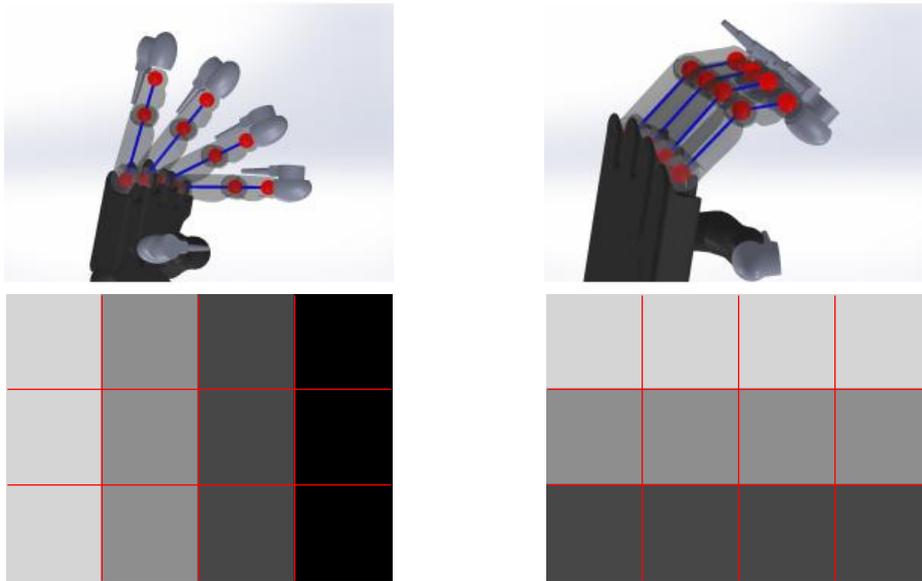
$$FCI(y, f) = k \sum_{i=1}^y \theta_{i,f} \quad (2.1)$$

where $FCI(y, f)$ is the value for the pixel on the row y and column f of the image FCI which corresponds to the representation of the corresponding joint angle shown in Fig. 2.6. $\theta_{i,f}$ is the angle of the joint i (row) of the finger f (column). k is a transformation constant from angle to gray scale value. The value of k depends on the bit depth of the pixels of the FCI . We define this value using the following expression:

$$k = \frac{P}{3} \quad (2.2)$$

where P is the maximum output level set by the bit depth in which the FCI was defined (e.g., 256 levels for a 8-bit image). It is divided by 3 so that there is an equal distribution of the grayscale value among the three phalanges (under the assumption that the three

joints share the same maximum value). The result is an image which pixels' values are updated when the joint angles of the concerned fingers change. Fig. 2.7 presents two examples of how hand configuration is represented in the image *FCI*. The image on the left shows that when the phalanges of a finger are aligned, the pixels corresponding to that finger share the same gray scale value. On the other hand, the image on the right shows that aligned fingers are represented as rows with the same grayscale value.



(A) Aligned phalanges (misaligned fingers). (B) Aligned fingers (Misaligned phalanges).
 FIGURE 2.7: Finger configurations (*Upper images*) and their corresponding Finger Configuration Images (*lower images*).

2.2.2 Reproduction of enclosure grasping

To reproduce the enclosure grasping with a robotic hand, several authors used an adaptive strategy. To illustrate how it works, let us consider a robotic finger as the one shown in Fig. 2.8. During grasping, θ_1 starts increasing until a contact is detected on any phalanx C_i upstream in the chain. If no contact is detected, the angle θ_1 will continue to increase until reaching its maximum value θ_1^{max} (this angle depends on the kinematics of the robotic hand being used). Once θ_1 stops, the same procedure is followed with θ_2 and so on with θ_3 afterwards.

Authors in [38] and [63] used this strategy and detected the contacts C_i based on touch sensors' readings during the grasp execution. Since we present this method to be solely based on proprioception, using contacts for the grasping strategy would be in contradiction with this statement. Thus, we propose to replace contact readings of C_i with joint torque readings T_i to detect contacts. In this case, a phalanx is stopped when the joint effort goes beyond a predetermined threshold.

This modification does not imply that tactile sensors cannot be used for this method to be valid. As a matter of fact, touch sensors used in [63] did not provide much information for shape identification and proprioception was fundamental for this task in the related work described in [62]. What this modification allows us to do is to present a method fully based on proprioception.

Algorithm 1 shows the procedure. For this enclosure grasping strategy, it is necessary that at least the proximal phalanx is not coupled with the rest of the phalanges. Otherwise, a different grasp strategy might be needed.

2.3 Proprioceptive shape signature

Given that the robotic hand has been endowed with a proprioception representation and an enclosure grasping strategy, we can now consider using proprioception for shape identification. For this, we are going to generate a model-based descriptor that we call *proprioceptive shape signature*, or simply *signature*. In the next sections we will explain the theoretical generation of a signature using the joint angles of fingers adapted on a shape. Then, the application of the *FCI* into the process of signature generation and finally, we will present a procedure to make the signature invariant to pose and size changes of the shape in the hand.

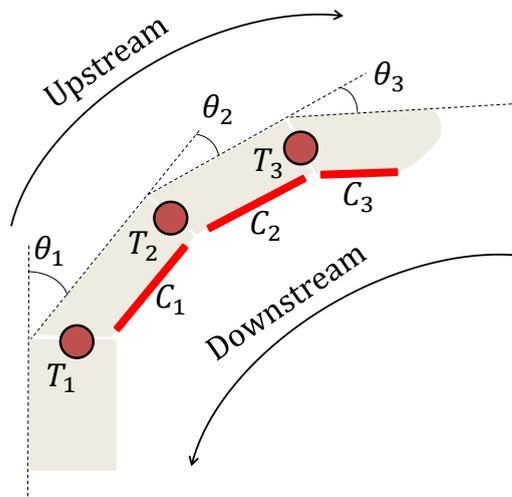


FIGURE 2.8: Lateral view of a finger. Red lines show the areas of contacts C_i onto which the contacts are detected. θ_i are the joint angles. T_i are the efforts on the joints i .

2.3.1 Signature generation process

When the enclosure grasp is achieved, each finger conforms to the shape of the object. Under this condition, global shape recognition can be identified. Even if corners and edges are considered the most salient characteristics in 3D haptic shape perception [73], they are not identifiable without tactile information. In the proprioception domain, any polygonal shape which cross-section provokes the joint angles to behave similarly will generate the same proprioceptive information. In this sense, proprioception alone cannot discriminate them (e.g. an hexagonal pyramid from a cone). Thus, for the sake of simplicity without losing the generality, this method assumes that when the enclosure grasp is performed around an object, each finger adapts to a circular shape.

As illustrated in Fig. 2.9, the central angle subtended by the arc formed by the finger on the circle is equal to the angle between the phalanges forming that arc (See Appendix A).

Algorithm 1 Pseudo Code for grasp strategy for one finger

Input:

T , vector containing the efforts detected on each joint
 θ , vector containing the joint angles to be sent to the hand. All elements are initialized at 0
 $\Delta\alpha$ Increment of the angles θ_i
 thres_T threshold of the joint efforts

begin

```

 $\theta = [0, 0, 0]$ 
 $i \leftarrow 1$ ;
 $flag = \text{true}$ 
while  $flag = \text{true}$  do
  GetTension( $T$ )
  if  $\exists n \in [i; 3] : T(n) > \text{thres}_T$  then
     $i \leftarrow n$ ;
    if  $i = 3$  then
       $flag \leftarrow \text{false}$ ;
    end if
  else
     $\theta_i \leftarrow \theta_i + \Delta\alpha$ 
    if  $\theta_i = \theta_i^{max}$  then
       $flag \leftarrow \text{false}$ ;
    end if
  end if
  SendAngleCommand( $\theta_i$ )
end while

```

end

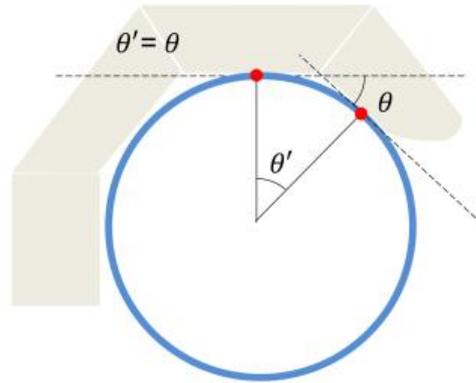


FIGURE 2.9: Lateral view of a finger conformed to a circle. The red points represent the contact points of the phalanges. These points form an arc shown in black that subtends the angle θ' . θ is the angle between the phalanges forming the arc on the circle.

The underlying association between the joint angle θ and the arc length s sets a relationship between the finger configuration and a geometric parameter of the object, namely the radius r of the enclosed circle, as follows:

$$s = r\theta \quad (2.3)$$

Since the length of the finger is constant, it can be stated that the arc length s is also constant. The angle θ and the radius r are inversely proportional variables.

Consider two fingers adapting to two different circles but forming arcs with the same length on each. Fig. 2.10a shows that the angle ($\angle AOB$) subtended by the arc on the larger circle is smaller than the one ($\angle COD$) on the smaller circle. Given that the joint angle can be known, the radius r of a circle can be inferred by 2.3. When this is applied to each finger, the radius of the different parts of the object shape can be obtained.

Each finger in contact to the object will generate a series (four, in the case of an anthropomorphic robotic hand) of arcs on the object surface. The arcs created by every finger forms a discrete representation of the contact surface. The contact surface can be reconstructed through interpolation between the series of arcs. To obtain a smooth surface, a cubic interpolation is adopted, so that small details on discontinuities on the surface of the object will not be represented there. Fig. 2.10b) illustrates the result of the interpolation between four circles of different radius. The blue manifold represents the contact region between the object and the fingers. This manifold can now be exploited to generate the *proprioceptive signature* of the shape.

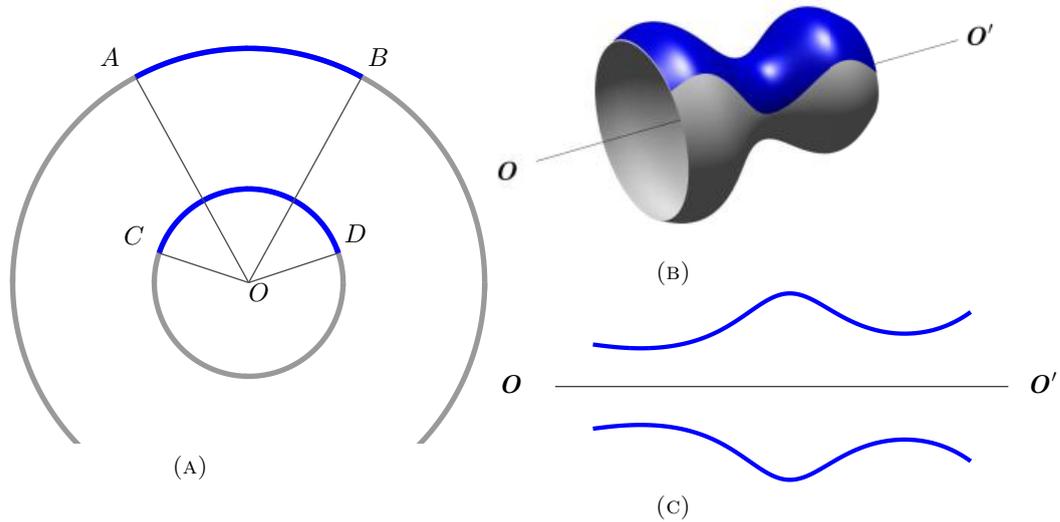


FIGURE 2.10: (A) Concentric circles showing the central angles of their corresponding intercepted arcs (\widehat{AB} and \widehat{CD}). Both arcs have the same length. (B) Continuous case of Fig. 2.10a. Infinite sequence of circles of different radii forming a shape (gray) and the manifold containing the arcs of the same length formed on each circle (blue). (C) Central angles subtended by their arcs contained in the manifold along the central segment OO' .

2.3.2 Proprioceptive signatures from the FCI

The FCI is now used to generate the proprioceptive signature of the object once the grasp is achieved and the fingers are conformed to the shape. For this, a two-step procedure is followed. First, the FCI is increased in size using cubic interpolation which softens the transition between the different radii to which each finger adapts. The size-increasing factor is chosen empirically. For this work, the chosen factor was 50, therefore, a 200×150 image resulted from this. Second, the signature $S(x)$ is built based on this interpolated FCI (FCI') using the following expression.

$$S(x) = \begin{cases} -\min FCI'_x \\ \max FCI'_x \end{cases} \quad (2.4)$$

which allows plotting two lines based on the minimum and maximum gray scale values in the corresponding column x of the FCI' . Since most robotic hands have coupled joints on the fingers, this representation integrates the information of those joints intrinsically. Thus, the lower line of the signature represents the angle of the proximal phalanx and the upper line represents the addition of the remaining angles (See Fig. 2.10c).

2.3.3 Signature enhancement process

The position of the fingers relative to each other will be similar for the same shapes, e.g., for a cone grasped along its main axis, the joint angles will be larger as its radius increases along this axis. This will be always the case regardless the radius of the cone. However, the absolute position of the finger will be affected by both the pose and the size of the object; therefore, the signatures are directly affected by those parameters. Fig. 2.11 illustrates how these parameters affect the signature. In case the object is rotated in the hand, so will be the signature (Fig. 2.11a). The vertical position (Fig. 2.11b) and the size (Fig. 2.11c) of the object similarly affect the signature: they make both lines to separate or get closer.

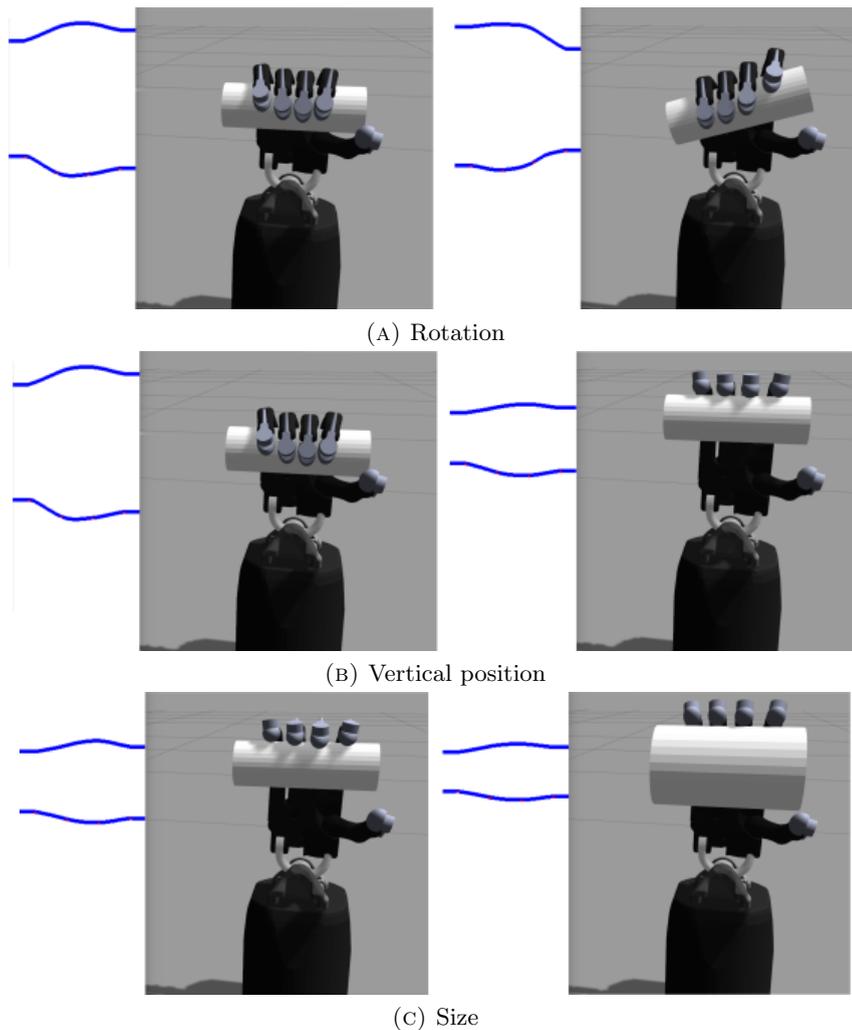


FIGURE 2.11: Signature modifications with the size and pose of the object.

In order to make the signatures invariant to the pose and size of the object, a series of modifications are separately performed on both lines of the signatures as follows:

First, the effect of the shape rotation is corrected. To do this, the points p_1 and p_2 on the ends of the signature (See Fig. 2.12b) are taken and the angle α is computed. To compute this angle, the slope of the line l that goes through those two points is used. Once the angle α has been computed, the signature is rotated to $-\alpha$ so that it becomes horizontal. Fig. 2.12b illustrates this step.

Second, the effect of the object size and vertical position on the signatures is corrected. Since this parameter causes the signature to be shifted vertically, the mean of the signature is computed and then subtracted from it. This will make the signature to be centered on the horizontal axis. This modification is depicted in Fig. 2.12c.

Third, the signature is normalized. This normalization is made for identification goals, as most machine learning algorithms require the input data to be normalized (Fig. 2.12d). The normalization range in this method will be $[-1; 1]$ following the standard normalization ranges.

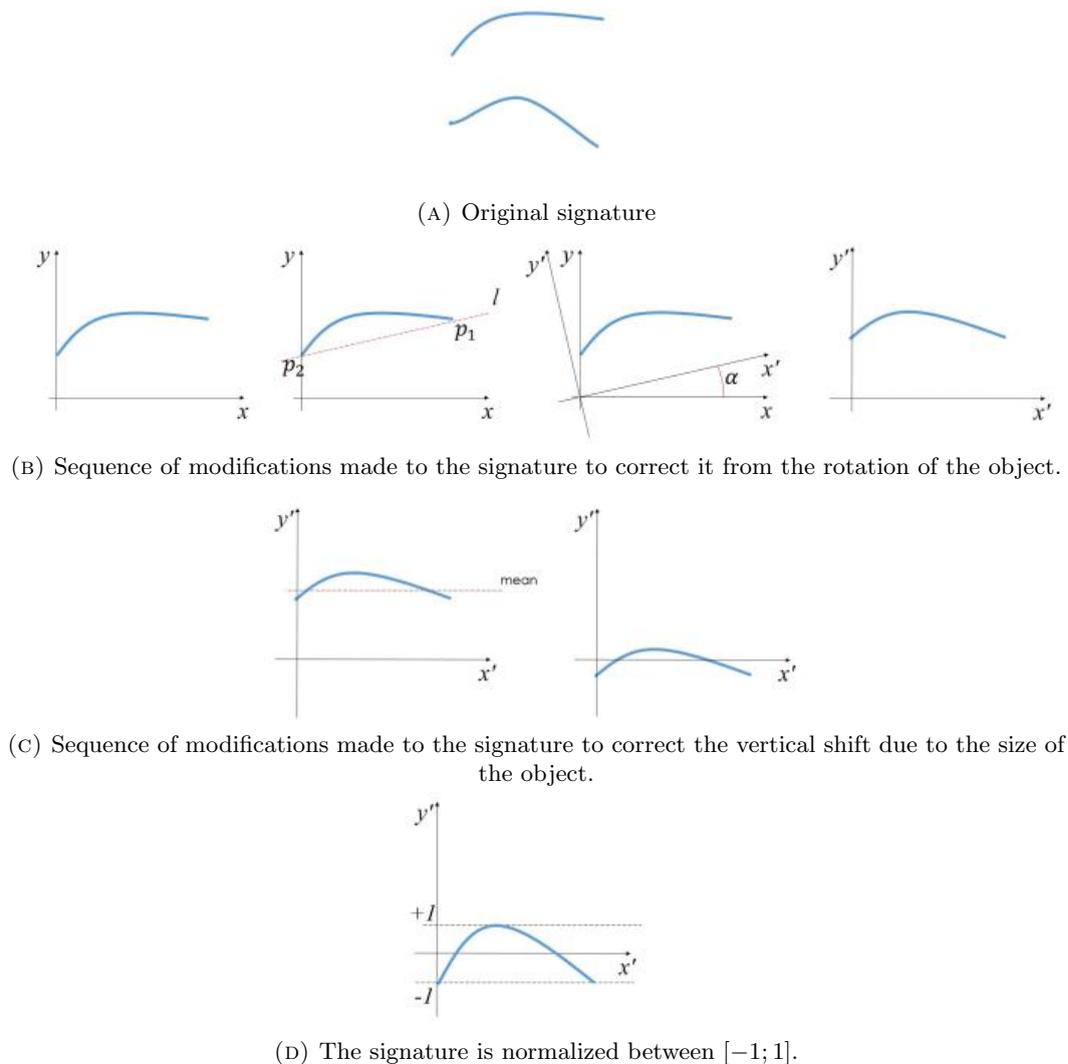


FIGURE 2.12: Signature enhancement procedure.

After this 3-step process, the effects of both the size and pose of the object have been discarded. The signature is in theory invariant to those parameters.

2.4 Experimental Setup

Experiments performed to test this method were carried in both simulated and real environments. In this section, we will present the setups used for both environments. For each setup, the procedures followed for the data collection and grasping execution will also be explained. Experiments concerning the following chapters will be based on the same setups and data collection process hereby presented. If additional information or procedure is added, it will be explained in the corresponding chapter. Before explaining the setups, the shapes chosen to validate this method will be introduced and their theoretical signatures shown to the reader.

2.4.1 Test shapes

To outline its performance, the proposed approach was tested on a set of five primitive shapes: cone, torus, sphere, one-sheeted hyperboloid and cylinder. The cone, sphere and cylinder have received a lot of attention in previous works. This has not been the case of the torus and hyperboloid, which makes them interesting to be added. Furthermore, household objects are frequently shaped as one of those five shapes. Some of these shapes were also chosen on their geometrical properties, such as convexity. This can be better observed in Fig. 2.13, where these shapes are illustrated with the signature they would theoretically generate. As can be noticed, they generate different signatures and almost all of the possible combinations of convexity in the signatures.

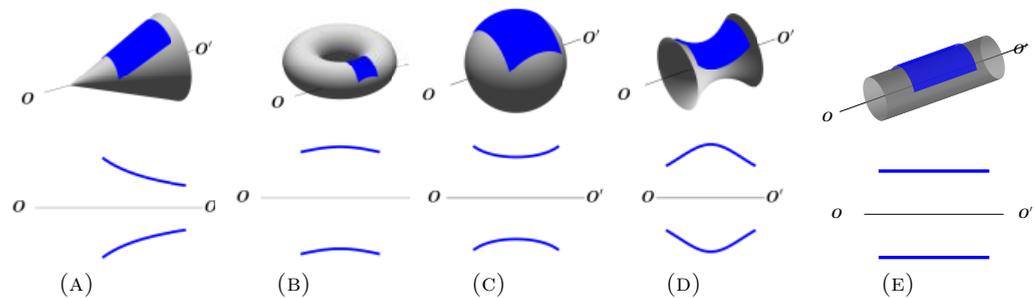


FIGURE 2.13: Set of test shapes. The blue manifold on top of them is the result of equivalent arcs formed on the circles contained in the shape. Below each shape, the corresponding signature obtained by extracting the central angles from the manifold with respect to the central axis $\overline{OO'}$. The procedure was shown in Fig. 2.10.

2.4.2 Simulated Setup

For experiments in simulation, the Gazebo model of the Shadow hand was used (Refer to Appendix B for details). Fig. 2.14 shows the shapes as simulated in the Gazebo simulator. The dimensions of these shapes can be modified, which we are going to take advantage of for data collection. In this environment, the shapes to be grasped were simulated as static which means that the physics of the simulated world do not apply to them. Thereby, their position is not changed by the interaction forces with the hand. On the contrary, the model of the hand was not static and not attached to any fixed point. Thus, the interaction forces between the model of the hand and the object may make the hand move from its initial position during the grasping execution.

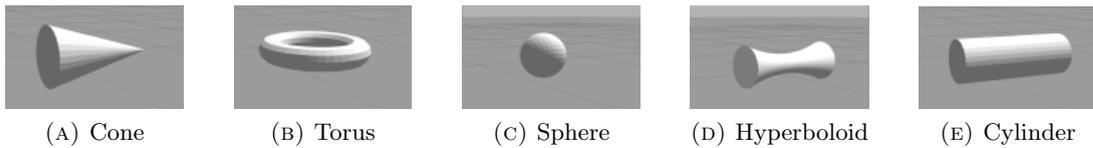


FIGURE 2.14: Simulated models of the shapes on Gazebo.

Grasp

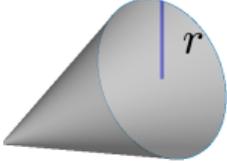
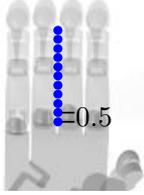
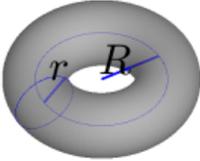
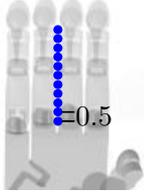
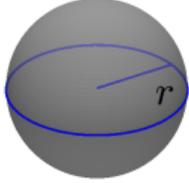
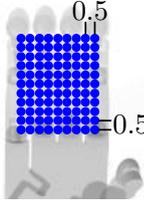
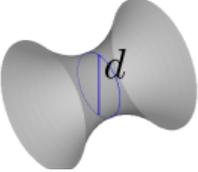
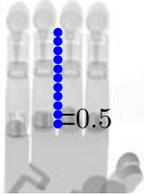
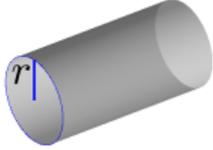
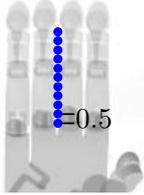
Forces at the joints cannot be measured in the simulated model of the Shadow hand. On the other hand, contacts on the surface of the phalanges can be detected. Thus, the contact-based version of the grasp strategy explained in section 2.2.2 will be employed and the magnitude of the contact ignored. To control the finger movements, the PID position controllers were performing.

Data collection process

In this thesis, it is claimed that signatures are made invariant to the size and pose of the object. Thus, both the dimensions and positions of the objects were varied during the data collection process to demonstrate this property. Specific information about the dimensions and positions used to collect the data are found in Table 2.1. The range of each parameter was chosen to be the widest possible. The goal of choosing such large ranges was to later define the ranges within which the presented method works accurately.

Regarding the rotation, this was tested with three of the shapes only: torus, hyperboloid and cylinder. Fig. 2.15 depicts how these shapes were rotated. The angle α was changed from 0° to 25° with changes of 5° .

TABLE 2.1: Tested dimensions and positions for each object. The column *Shape Dimension* shows the dimensions varying for each shape. In the column *Size*, the intervals of dimensions that were tested and the change rates. Images in the column *Position* correspond to the positions where the shapes were located. All dimensions are shown in *cm*.

Shape Dimensions	Size	Position
	$r = [1, 8]$ $\Delta r = 0.25$	
	$r = [1, 8]$ $\Delta r = 0.25$ R increased proportionally with r as ($R = 4 * r$)	
	$r = [1, 8]$ $\Delta r = 0.25$	
	$d = [2, 10]$ $\Delta d = 0.25$	
	$r = [1, 8]$ $\Delta r = 0.25$	

All combinations of sizes, positions and rotations were considered and a data base containing 10976 signatures was obtained after four simulation runs. The data from these simulations will be used in next chapters.

2.4.3 Real-hand setup

Real application tests were performed on a Shadow Hand described in Appendix B. Real objects with the same primitive shapes as the simulated ones in section 2.4.2 were used. Two sets of objects were formed: one of the sets (**Set 1**) with 3D-printed and

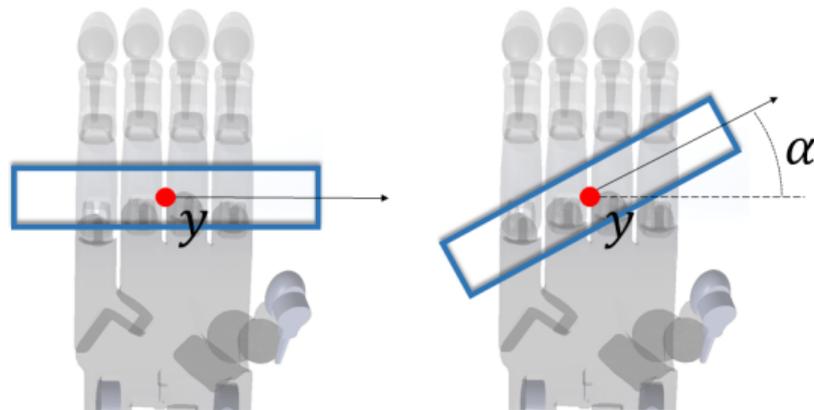
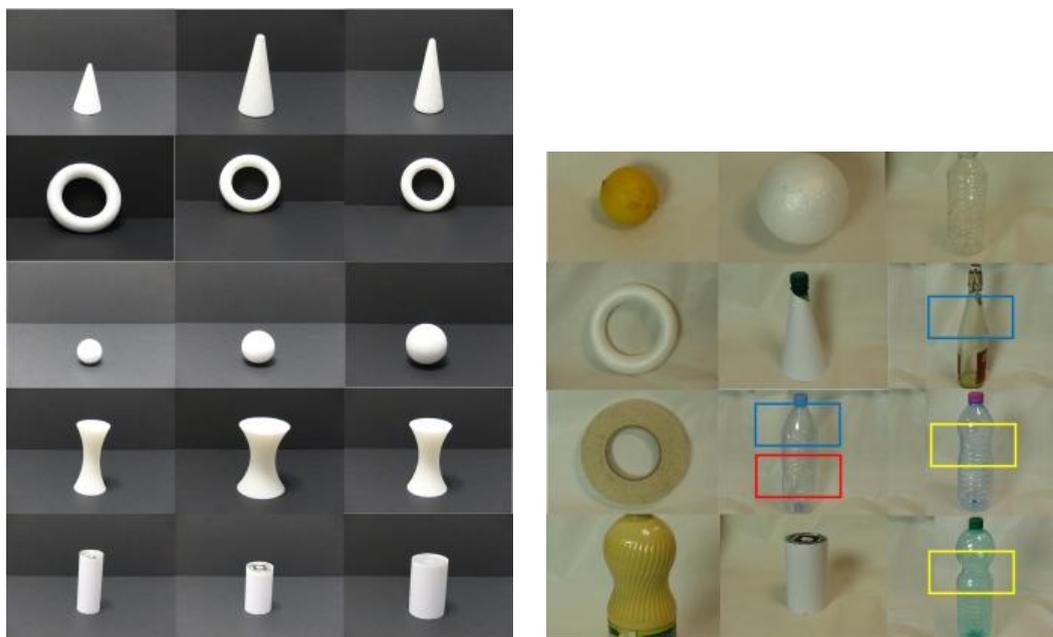


FIGURE 2.15: Schema showing how the angle α was varied to rotate the shapes with respect to the normal to the palm.

polystyrene shapes, three objects of different dimensions for each shape (Fig. 2.16a), the second set (**Set 2**) with everyday objects (Fig. 2.16b).



(A) **Set 1**

(B) **Set 2**

FIGURE 2.16: Objects used for real experiments. In **Set 2**, some of the objects were grasped on different sites to get different shapes. This is indicated by rectangles on those objects. Yellow is used for hyperboloid, red for cylinder and blue for cone.

Grasp

The joint-force based grasp strategy was used in the Shadow hand for experiments in the real environments. The force sensors embedded in this hand have a resolution of 30mN. The parameter thres_τ in Algorithm 1 was set to 100mN for all joints. This value was chosen empirically; it allows the fingers to use the force needed to move although remaining within the normal security values ($\pm 300\text{mN}$). To control the finger movements, the PID force/torque controllers were performing.

Data collection process

In the data collection process, the objects were passed on to the hand with different random positions and rotations. Interaction forces between the hand and the objects could make small changes on the position of the hand. Each object was grasped between 4 and 7 times for both **Set 1** and **Set 2**. A picture was taken after the grasp was executed, and this image was used to estimate the rotation angle of the object. This was done by computing the angle between the axis of the hand and the axis of the object as illustrated in Fig. 2.17.

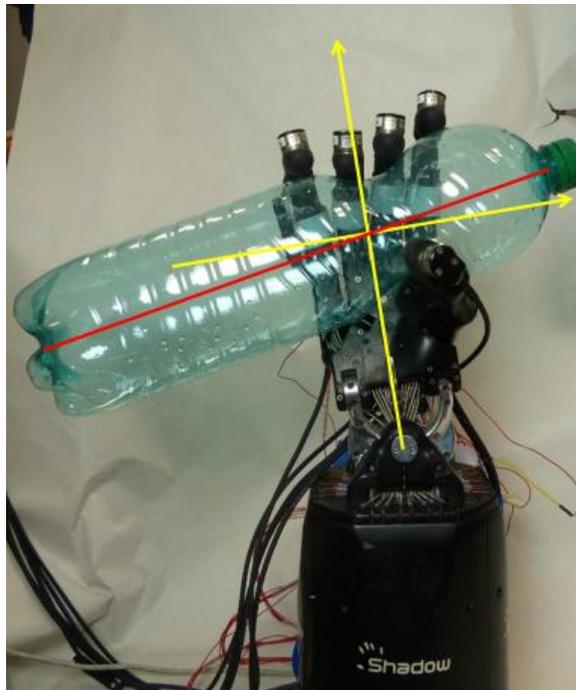


FIGURE 2.17: Object angle estimation. Yellow lines correspond to the hand axis. Red line to the object axis. The angle of rotation is measured between the horizontal axis of the hand and the axis of the object.

In this chapter, both sets were combined for all tests. In future chapters, this might not be the case. If so, it will be clearly indicated.

2.5 Experimental evaluation of the signatures

In this section, we are going to evaluate different aspects of the signature generation and quality. First, we will show the results obtained from generating the signature using the *FCI*, then, we will measure the invariance of the signatures to size and pose. The ranges within which the signatures are invariant will be defined. At last, we will estimate the rotation angle of the object within the hand. This evaluation will be carried on simulated and real data.

2.5.1 Signature generation

Fig. 2.18 shows the proprioceptive signatures obtained based on the *FCI*, as explained in section 2.3.2. Results from both environments, simulated and real, are presented in this figure. Changes exposed in section 2.3.3 can be noticed in these signatures: size and pose of the objects make the signatures change their vertical shift and inclination, respectively. Nevertheless, a visual inspection points out that these signatures keep certain similarity to their corresponding patterns shown in Fig. 2.13.

The resemblance with the corresponding patterns is not valid for all shapes. For instance, the signatures obtained for the cylinder have a certain similarity with the ones obtained for the hyperboloid (Fig. 2.19e). This is due to the kinematics of the hand and how the fingers are positioned with respect to each other in the Shadow hand, which was not taken into account in the *FCI*. Even if kinematics were taken into account, the generation straight horizontal lines on the signatures is theoretical and not likely to happen in experiments.

Fig. 2.19 illustrates the signatures obtained for each shape after the enhancement procedure has been performed on them. Through a visual inspection, the reader can infer that signatures of the same shapes show significant similarities regardless the changes of object size and the locations where the object was in the hand. Consequently, it can be concluded that signatures are invariant to the size and pose of the object within the hand. Since this is debatable for some shapes such as the sphere and the torus, the signatures will be compared in the next section to the theoretical patterns using DTW to measure the invariance more efficiently. Also, the threshold to discriminate them from non-corresponding signatures will be defined.

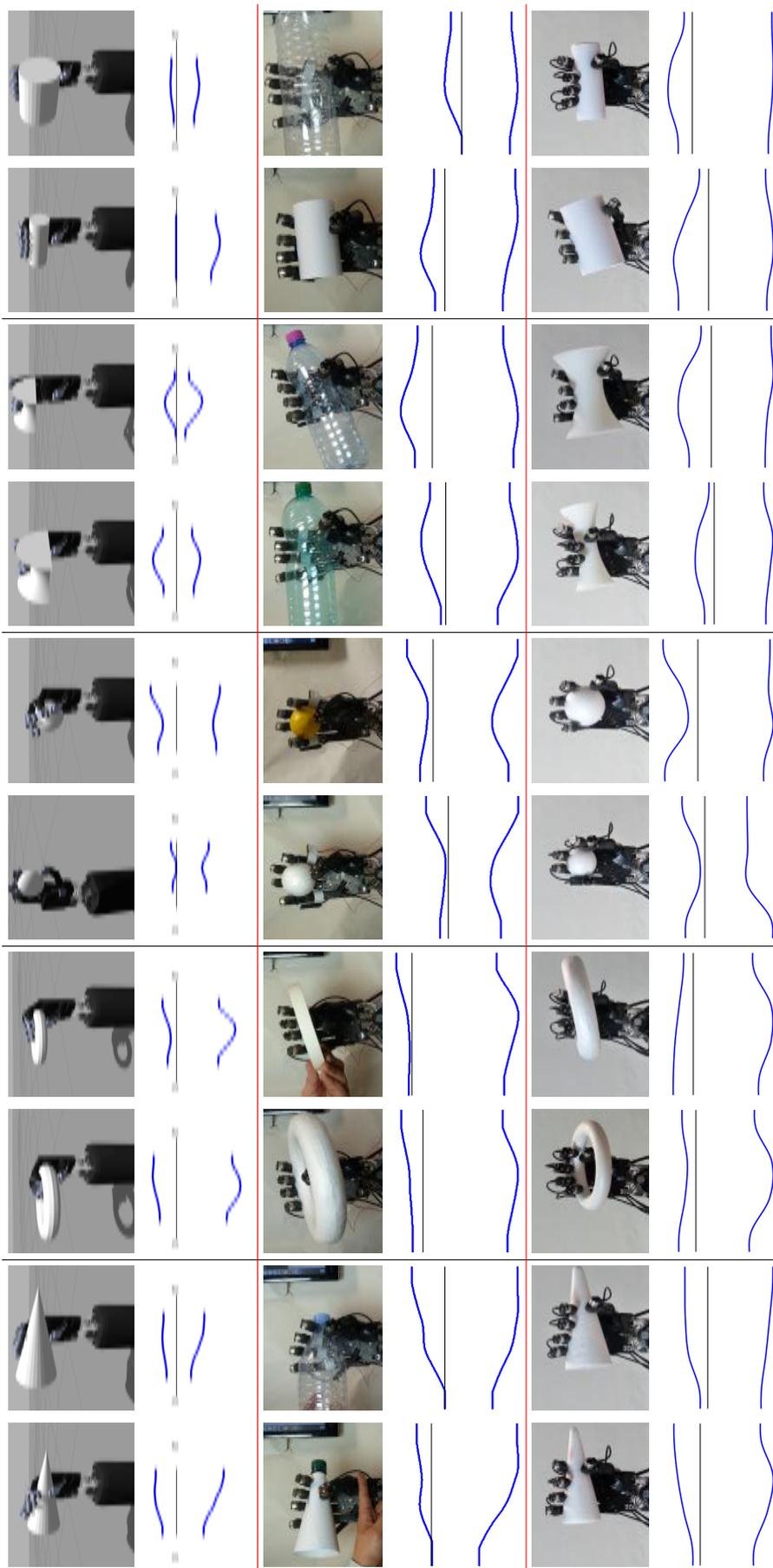


FIGURE 2.18: Signatures obtained on both simulated (*upper row*) and real objects:**Set 1** (*lower row*) and **Set 2** (*middle row*). *from left to right: cone, torus, sphere, hyperboloid, cylinder.*

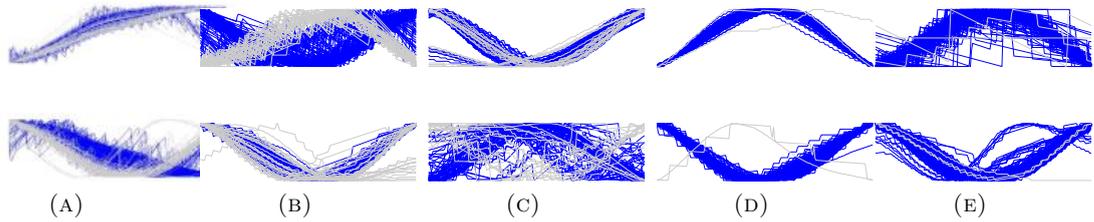
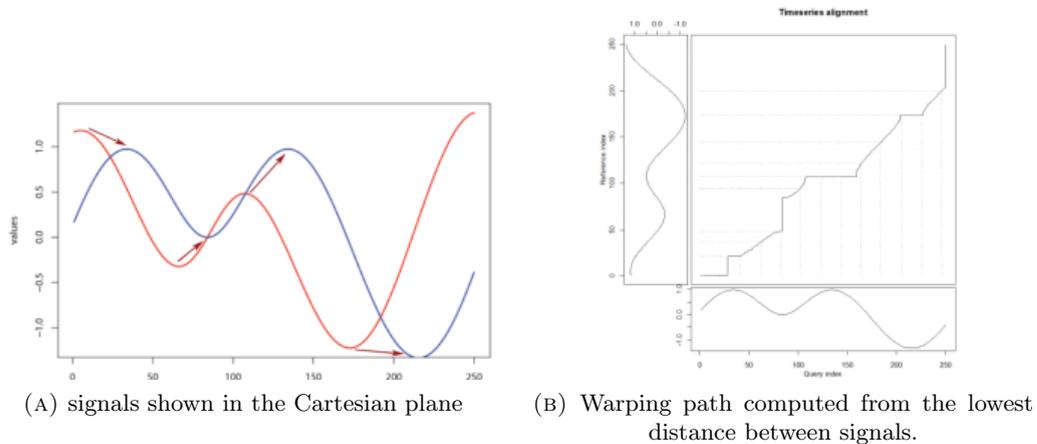


FIGURE 2.19: Measured signatures of the (a) cone, (b) torus, (c) sphere, (d) hyperboloid, (e) cylinder after being enhanced. The signatures in *blue* are the signatures that look like their corresponding theoretical pattern. The *gray* ones are the ones that not. In order to determine the similarities, the threshold defined in section 2.5.2 was used.

2.5.2 Signatures invariance

Hereby, we present the procedure followed to evaluate how much signatures are invariant with respect to the size and position of the shapes and the performance of the signature enhancement procedure. For this, each enhanced signature from measurements was compared to theoretical patterns shown in Fig. 2.13 using the Dynamic Time Warping (DTW) algorithm [74]. This algorithm allows measuring the similarities between two signals.



(A) signals shown in the Cartesian plane

(B) Warping path computed from the lowest distance between signals.

FIGURE 2.20: Dynamic Time Warping.

As a measure of similarity between two signals, DTW gives a distance-like measurement between them. Given two signals as shown in Fig. 2.20a, DTW computes the warping path by matching the points of the signals that are the closest to each other (Fig. 2.20b).

Results of this comparison are illustrated in Fig. 2.21. As one can observe, most shapes showed lower DTW-distance when compared to their corresponding theoretical signature. For reasons explained in the previous section, the cylinder is an exception to this statement. Simulated results illustrated in Fig. 2.21a show this even more clearly: the DTW-distances between the measured signatures and the corresponding pattern of the cylinder were very high. For the real objects, the results are shown in Fig. 2.21b.

Based on the results obtained with the simulated data, we decided to compare the signatures of the cylinder with the pattern of the hyperboloid, which explains why the DTW-distances are low for the cylinder. It confirms the similarity between both shapes signatures.

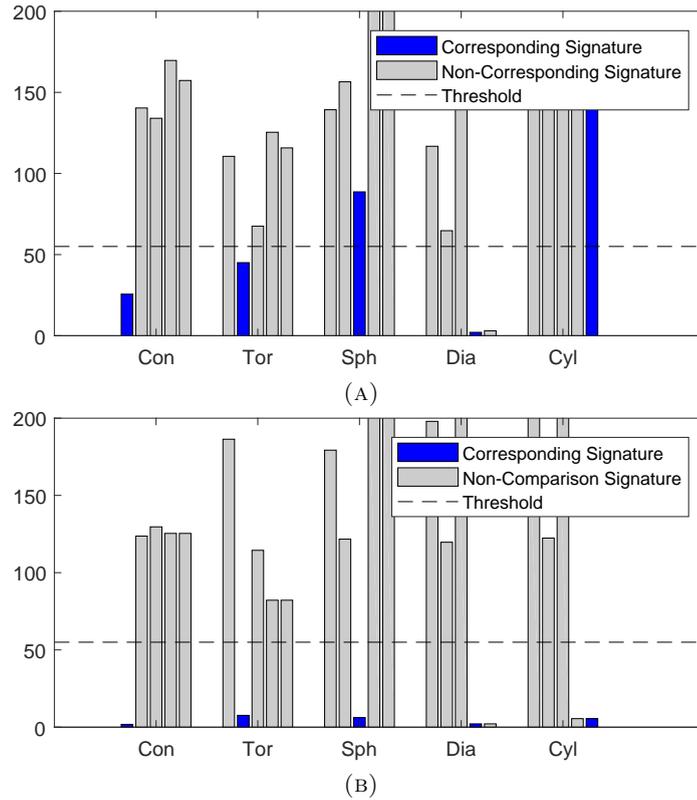


FIGURE 2.21: DTW-distances between generated signatures and theoretical signature patterns: (A) Simulation. (B) with real hand.

The differences between the shapes of the cylinder and the hyperboloid must have an effect on the signatures as well. As the cylinder is flat, its signature should be less curved than the hyperboloid's one. This should be reflected in the integrals of both signatures. To confirm this, after the second step of the signature enhancement procedure, the integral of the signatures was computed. Results (Fig. 2.22) showed that this hypothesis was true for both simulated and real signatures: the hyperboloid signatures presented a higher integral. Both simulated and real data behaved similarly (i.e., the integrals of the hyperboloid tends to be larger than those of the cylinder) but integral values were not the same in both environments. A threshold was defined to discriminate these signatures. This threshold was chosen so that at least 75% of the data of each shape could be discriminated.

Based on the results of the comparison of the signatures with the expected patterns, a threshold was defined for the DTW-distances to determine whether a signature corresponds to a specific pattern or not. To do so, only the simulated results were taken into account. This threshold had to be lower than all distances between non-corresponding

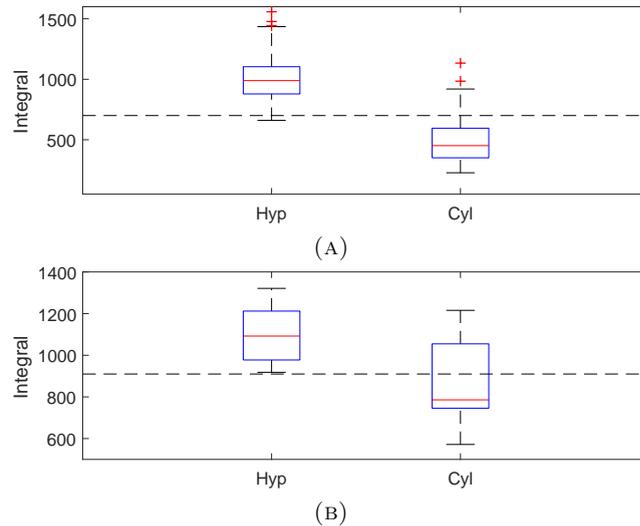


FIGURE 2.22: Integrals of the hyperboloid and cylinder resulting from the signature enhancement procedure. The dashed line is the threshold defined to distinguish the signatures of these two shapes. This threshold was computed using the mean of the mean values of the integrals of each shape. (a) Simulation. (b) with real hand.

signatures. Thus, results shown in Fig. 2.21a were analyzed. It was observed that for most of the signatures, the mean distances between the signatures and the corresponding patterns fell below a threshold of 50, except for the sphere which mean goes up to around 80. For the non-corresponding signatures, the lowest distances were about 60. Thus, the threshold was fixed to 55. This threshold will be used in the results presented in the following sections.

In order to further evaluate the signature enhancement procedure, the effects of the inclination of the objects on the DTW-distances was analyzed. Fig. 2.23 shows the percentage of measured signatures that correspond to the theoretical patterns as a function of the inclination angle. These results correspond to signatures measured in simulation. As shown in the referred figure, the signatures match their corresponding pattern with an accuracy of 70% for most shapes independently of the inclination angle. An exception to this is the torus, but still, never lower than 60%. This can be attributed to the signatures of this shape that resembles the hyperboloid (shown in gray in Fig. 2.19b).

2.5.3 Defining ranges for size and position

To define the ranges within which the presented method works best, results from the last section were used. Every signature that matched with its expected pattern increased the probability of generating the correct signature for a specific pair of size and pose. These probabilities were computed for each shape and put into a matrix which dimensions depend on the number of parameters changing during the tests. For

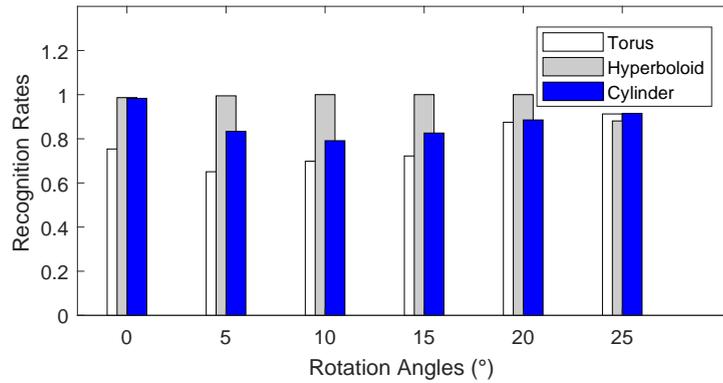


FIGURE 2.23: Identification rate of rotated objects with respect to the angle of rotation.

instance, in the case of the cone, only the radius and the vertical position changed. Thus, the matrix is two-dimensional. Once the matrix is filled with the probabilities, the means of the probabilities contained in every possible continuous interval for each dimension of the matrix were computed. Only those for which the mean was higher than 80% were kept. Then, the largest intervals from the chosen ones were considered to reflect the best possible performance of the method.

Table 2.2 shows the ranges of the radius obtained for each shape. It also shows the interval of positions on the hand considered for each shape. The position range defined for the torus shows that this shape is better recognized on upper position. This is due to the fact that given a horizontal pose of the torus in the hand, the contact zone is not located on the top of the shape as shown Fig. 2.13 but rather on its inner part. Consequently, the generated signature is the same as the hyperboloid one, as confirmed by the Fig. 2.24. This explains why the lower positions are not included in the position range.

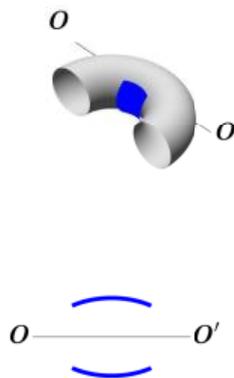
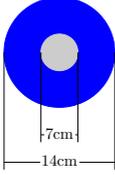
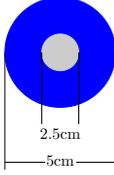
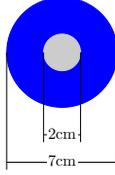
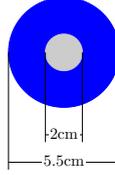
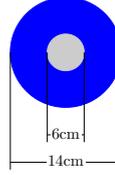
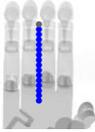
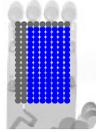
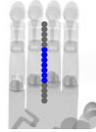
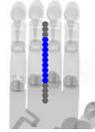


FIGURE 2.24: Illustration of the location of the manifold formed by the fingers adapting to a torus located near the palm and the corresponding generated signature. The shape is cross-sectioned to facilitate the view.

In the experiments with real objects, their sizes and locations were chosen to fall within these ranges, which explains why the results depicted in Fig. 2.21b show that all shapes fall below the threshold established in section 2.5.2.

TABLE 2.2: Ranges of size and positions in the hand attained for each one of the shapes.

Cone	Torus	Sphere	Hyperboloid	Cylinder
				
				

2.5.4 Estimation of the object angle of rotation

Thanks to how the signatures are generated, information about the pose of the object within the hand can be estimated. The angle of rotation α is particularly relevant as the signature inclination has a direct relationship with it. Indeed, the rotation angle computed in step 1 of the signature enhancement procedure can be exploited and taken as a measure of the object rotation.

Fig. 2.25 illustrates the results corresponding to the estimation of the rotation angle α . In these images, the measured angle of the signature is plotted against the actual inclination angle of the shape in simulation.

The behavior of the results matches with a linear relationship between computed and actual angles. For this reason, a linear regression was used to create the model to estimate the rotation angles. The performance of this model, created using simulated data, was tested using data obtained with the real platform. Results are shown in Fig. 2.26. In order to determine how well the model fits the data, the coefficient of determination R^2 is computed as follows:

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \quad (2)$$

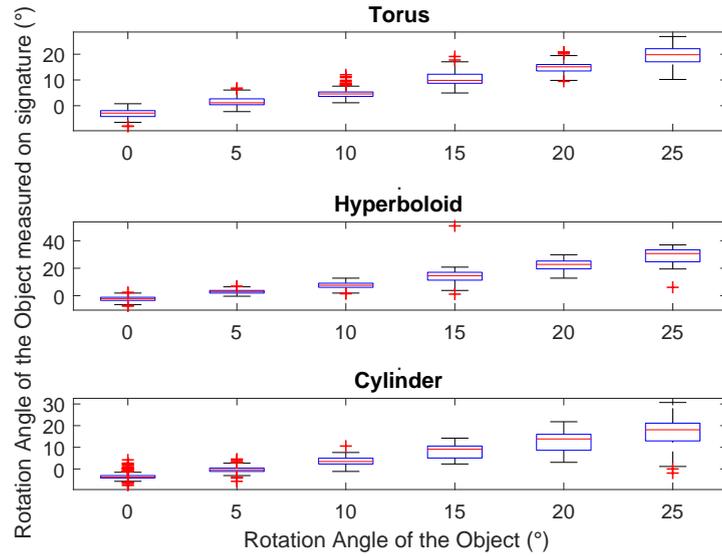


FIGURE 2.25: Measured angle of the signatures versus the actual angle of rotation of the object in simulation.

where n is the number of samples y , \hat{y}_i represents the estimated values and \bar{y} is the mean of y . This coefficient falls between $[0, 1]$, 1 being the perfect fit. The R^2 obtained for the torus, hyperboloid and cylinder were 0.61, 0.49 and 0.63, respectively. Thus, the models fit sufficiently well to be used for rotation angle estimation. These results depicts an interesting feature because it adds another advantage to the signature, such as the capability of estimating the pose of the object within the hand.

2.5.5 Modified shapes

In this section, it is desired to evaluate the performance of the presented method when the shapes to be identified do not correspond to geometric primitives. In other words, to evaluate the behavior of the signatures when edges or distortions are present on the shape. For this, an ovoid and a pyramid with a heptagon base were chosen as modifications of the sphere and cone, respectively. Results are shown in Fig. 2.27. As can be observed for the pyramid, its signature corresponded to the cone one, confirming what was said in section 2.3 about proprioception not being able to detect edges and generating the same signatures for shapes without edges. The ovoid generated signatures similar to that of a sphere or a torus. When data was analyzed in more detail, we realized that it depends on the position of the ovoid within the hand. Thus, for lower positions, a signature similar to that of a sphere would be generated, and for upward locations in the hand, the signature becomes similar to that of a torus.

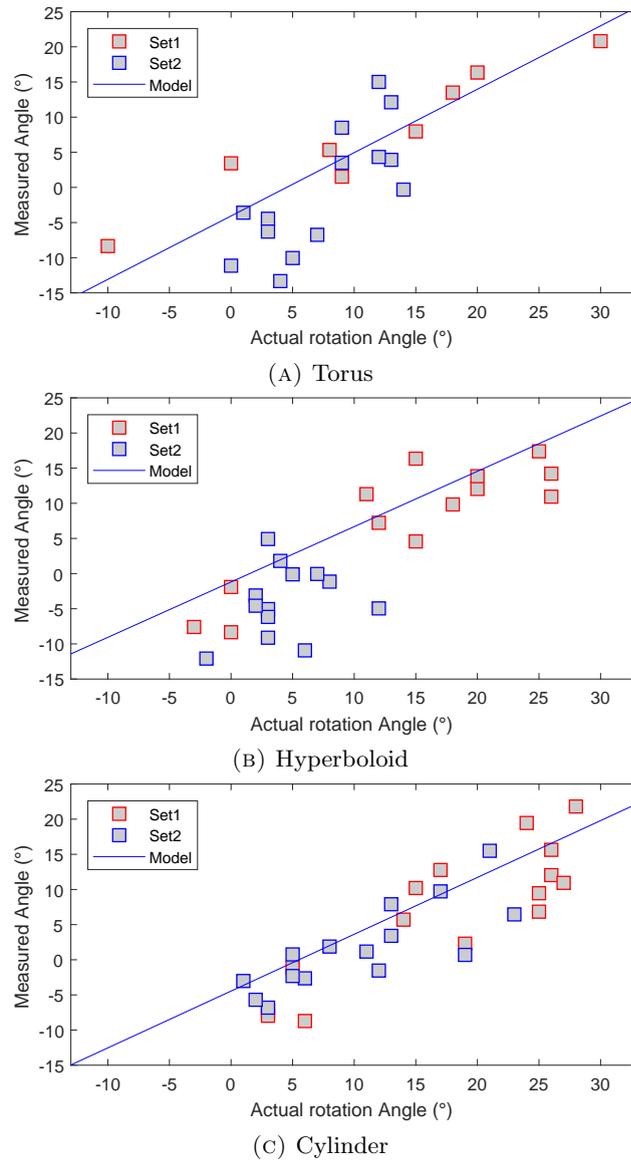


FIGURE 2.26: Measured angle of the signatures versus the actual angle of rotation of the object in the real platform. The blue lines correspond to the linear regression model.

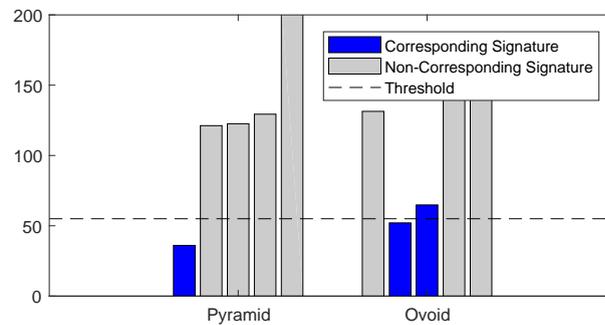


FIGURE 2.27: DTW distances computed for an ovoid and a hectagon pyramid. The theoretical signatures of a cone, torus, sphere, hyperboloid and cylinder were used.

2.6 Experimental evaluation of shape identification with signatures

For object shape identification, features were extracted from the enhanced signatures. In this thesis work, points from each line of the signatures were taken. These points were equidistant on the x-axis. Their coordinates on the y-axis were concatenated into a vector. Here, the number of the chosen points were 12 in each line, which results in a 24-element feature vector. This feature vector will be the input for machine learning algorithms. In this chapter, two machine learning algorithms will be used: Self-organizing maps (SOM) and Support Vector Machine (SVM). Next sections will briefly introduce these two algorithms.

2.6.1 Data division

In order to evaluate the performance of the signatures for object shape identification, a separation of the data into two sets is needed: training set and test set. Thus, the simulated data will be divided into two sets. The signatures that will be considered in this section correspond to those falling within the ranges defined for object size and position in section 2.5.

As common practice, the *training set* will contain 30% of the samples. This set should contain an equal amount of samples of each shape in order to avoid bias and overfitting. Since the amount of samples is not the same for all shapes, the 30% will be computed with respect to the shape that has the fewer amount of samples.

The *test set* will contain all the samples that are not included in the training set. Balance is also important in the test set. Thus, results will be computed as follows: ten confusion matrices will be computed and for each confusion matrix, the number of samples will correspond to 70% of the shape that has the fewer samples. Samples will be chosen randomly. The results presented here will correspond to the mean of the ten computed confusion matrices.

The data collected on the real platform will not be used for training. Thus, this data will be put in a single test set. This data will be used as inputs for the SVM and SOM trained with the simulated data.

2.6.2 Self-Organizing Maps learning algorithm

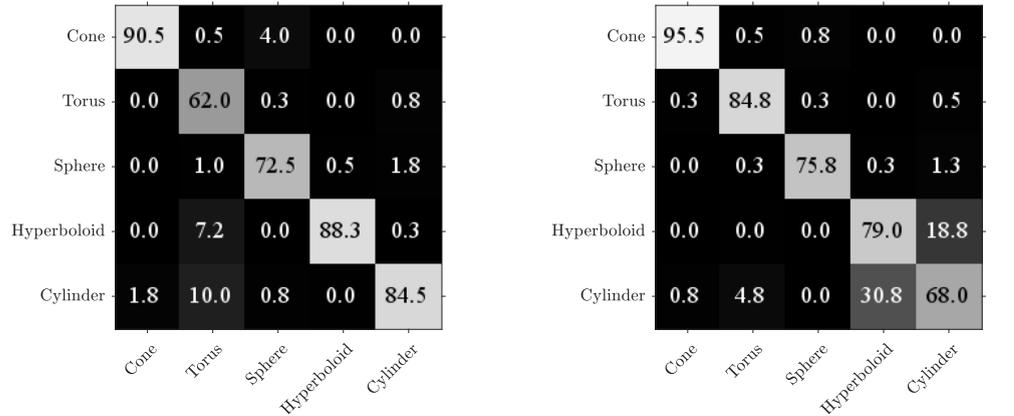
Most previous works on shape identification based on proprioception only exploit SOM to perform object shape identification. This choice is made based on the fact that SOM emulates the functioning of the sensory cortex. To achieve such functioning, the SOM algorithm emulates the topographical organization of the sensory cortex in the mammalian brain: neurons can be driven by stimuli coming from neighboring neurons [75] and similar stimuli can excite similar areas of the sensory cortex. SOM algorithm serves as a clustering method based on artificial neural networks to create a topographical representation of the input data. It is called "self-organizing" because it exploits unsupervised competitive learning [76]. The neurons of the SOM are organized in a 2-dimensional manner so that each neuron is connected to its neighboring neurons only. There is a weight vector associated with each neuron which length will be the same as the input vector.

As the SOM is initialized, the weight vectors of each neuron are randomly assigned. To create an organized map in the training phase, the SOM algorithm has two stages: the competitive stage and the cooperative stage. In the competitive stage, the Euclidean distance between the weight vector of each neuron and the values contained in the input vector are computed. The neuron with the smallest Euclidean distance to the input vector is chosen as the "winner". Thus, the weights of the "winner" neurons will be adapted to increase its similarity to the input vector. The cooperative stage is in charge of adapting the weights of the neighboring neurons so that the formed maps are topographically ordered. By doing so, the neighboring neurons respond to similar inputs.

The following analysis of the performance of this algorithm will be based on two criteria: the identification rate and the weight distances of neurons. The weight distances allow to determine how close the weights of two neurons are. Close weights depict neurons belonging to the same cluster. This can tell us how efficiently the algorithm was capable of clustering the feature vectors. For this, the neural network toolbox from matlab [77] will be implemented.

Signatures were used as inputs to this algorithm. To take into account the results from the previous sections, the integral of the signatures will be added to the 24-element feature vector, which results in a 25-element feature vector. Fig. 2.28 shows the performance of SOM when the raw values of the joint angles (JA) or the signatures (SI) are used as inputs.

As can be observed, the confusion matrices show that the SI (Fig. 2.29b) perform slightly better than the JA (Fig. 2.28a). However, it is important to point out that

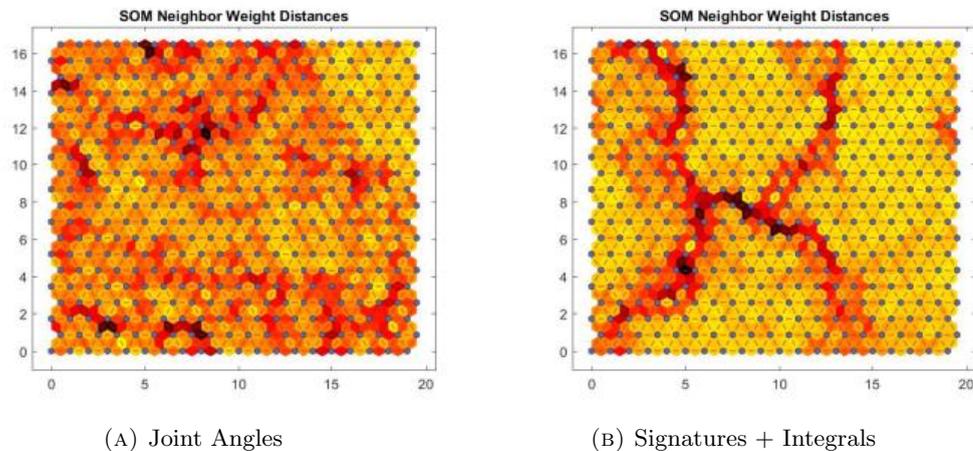


(A) Confusion matrix (%) using raw values of joint angles (JA) (B) Confusion matrix (%) using signatures and integrals (SI)

FIGURE 2.28: Confusion matrices obtained with SOM using the simulated training and test sets.

JA discriminated the hyperboloid and cylinder with more success, which proves that information from these shapes are different. These differences are lost in the signature during the normalization step of the enhancement process. Nevertheless, integrals of the signatures contain the differences and allowed the SOM to successfully discriminate both shapes.

The weight neighbor distances between the neurons of the SOM's are depicted in Figures 2.29a and 2.28b. The image corresponding to SI depicts a clear separation of four clusters (because the hyperboloid and cylinder are mixed in the same cluster). This is not the case when JA are used: no clear separation is observed.



(A) Joint Angles (B) Signatures + Integrals

FIGURE 2.29: Weight neighbor distances of the maps obtained with SOM algorithm using the simulated data.

For the real data, the SOM model trained with simulated data were exploited. Once more, SI (Fig. 2.30b) performed generally better than JA (Fig. 2.30a). In these results, discrimination between the hyperboloid and cylinder is less successful than in simulated

results. In the case of SI, it is related to the fact that the values of the integrals are different with simulation and real platform (Fig. 2.22).

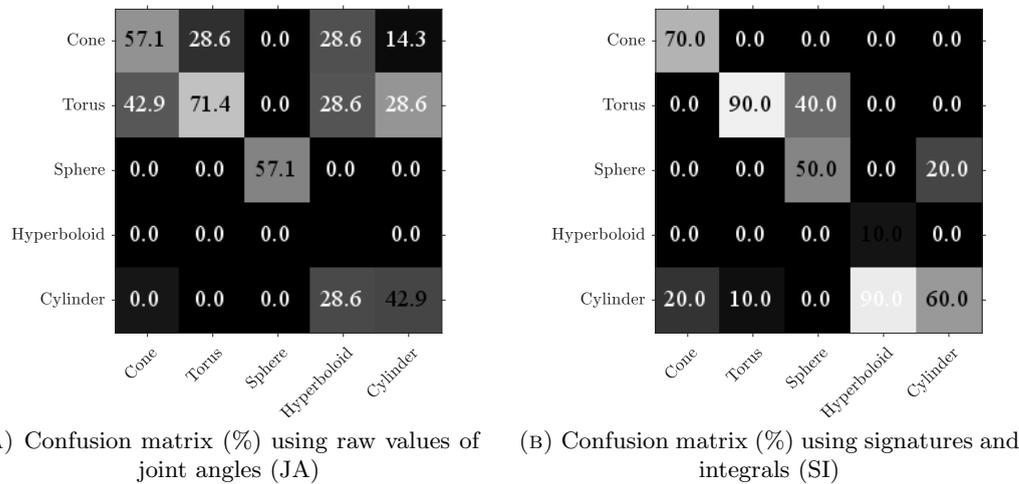


FIGURE 2.30: Results of SOM using the real data sets.

Based on this analysis, SI perform better than JA as inputs for object shape identification. Thus, the next experiments will be made with the SI data only.

2.6.3 Support Vector Machine learning algorithm

We will make tests to evaluate the performance of the signatures with the Support Vector Machine (SVM) algorithm [78]. SVM is defined as hyperplanes that separate the training data by a maximal margin such as shown in Fig. 2.31. The training instances that are the closest to the hyperplane are called support vectors.

To achieve separation of instances, SVM takes the input vectors and passes them through a kernel function to project them into a high dimensional feature space. The hyperplanes defined in this feature space correspond to more complex decision boundaries in the original space. According to Tong et al. [79], two types of kernels are commonly used: polynomial kernel and radial basis kernel. The first generates polynomial boundaries in the original space while the second induces boundaries by placing weighted Gaussians upon key training instances.

To implement this algorithm, the OpenCV library [80] will be exploited. The `trainauto` will be used for training. This function chooses the optimal kernel to separate the data empirically.

When the `trainauto` function had signatures as inputs, it found the radial basis kernel the most optimal. The confusion matrices obtained using the SVM are shown in Fig. 2.32. Results issued from simulated data depict a better performance compared to those issued from SOM in the same situation (Fig. 2.29b). Moreover, SVM performed better

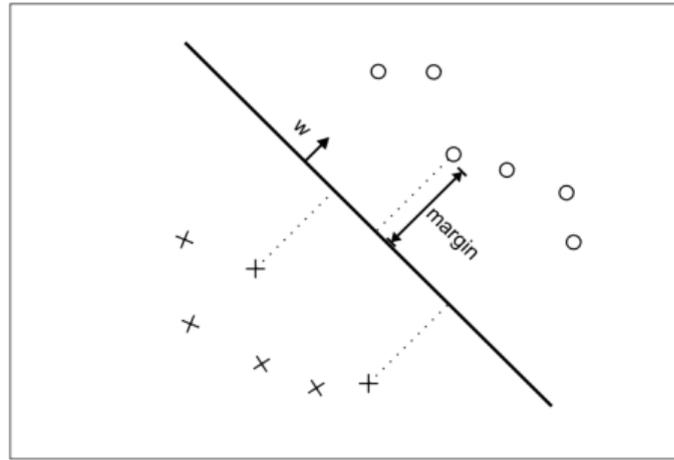
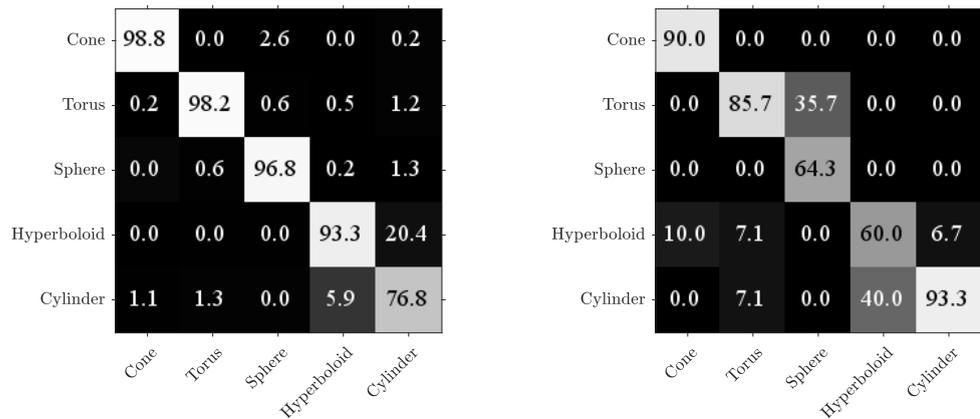


FIGURE 2.31: Object angle estimation. Yellow lines correspond to the hand axis, red line to the object axis. The angle of rotation is measured between the horizontal axis of the hand and the axis of the object.

than SOM on real data with a simulated-data-trained model (Fig. 2.32b). Results presented later in this sections are based on the results summarized in the confusion matrices.



(A) Confusion matrix (%) using simulated data (B) Confusion matrix (%) using real data

FIGURE 2.32: SVM results based on SI using simulated and real data sets.

Position and shape recognition rates

To further analyze the results obtained with SVM, the identification rates with respect to size and position of the objects were computed. Fig. 2.33 illustrates the results obtained for each shape with respect to position. Results agree with both the ranges obtained in section 2.5.3 and the recognition rates obtained with SVM.

As for the results with respect to the size of the shapes, it is shown in Fig. 2.34 that the recognition rate changes with respect to the radius of the shape within the ranges

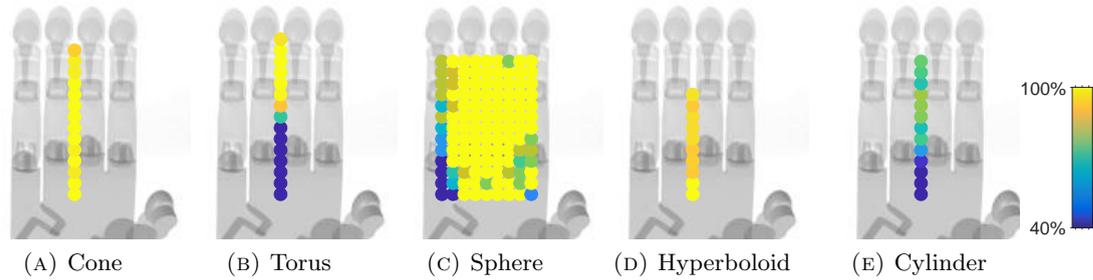


FIGURE 2.33: SVM results with respect to the position of the object. SI are used as inputs from simulated data.

defined in section 2.5.3. The cylinder showed the lowest recognition rates which is due to the confusion with the hyperboloid. Due to the scale used to illustrate these results, the performance of the method for the cylinder seems drastically lower than for the other shapes. However, the difference is 20% only, which still makes this method valid for cylinder identification. In any case, in chapter 4 we are going to present how to improve these results.

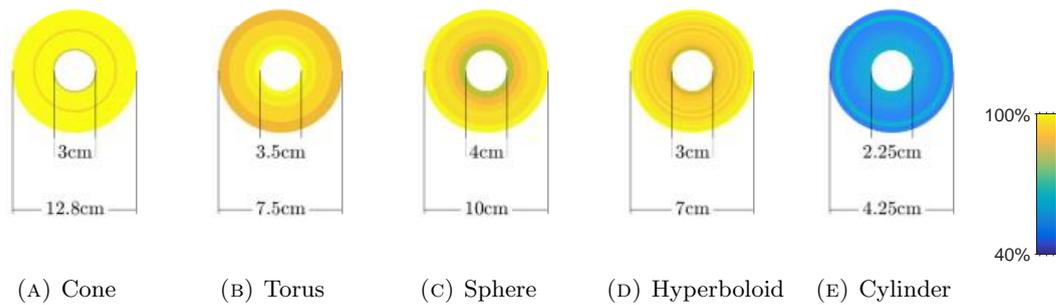


FIGURE 2.34: SVM results with respect to the size of the object. SI are used as inputs from simulated data.

2.7 Conclusion

In this chapter, proprioceptive data was used to identify the shape of a grasped object in a robotic hand. For this task, a representation of the joint angles of the fingers was created in which the finger configurations were imprinted. This representation allowed to find patterns related to the grasped shapes, thus to generate the so-called proprioceptive shape signatures. It was shown that these signatures have the potential of being profitable for both object recognition and manipulation tasks because they contain information on the global shape and pose of the object within the hand.

Experiments were carried out on both simulated and real hands. Results obtained in both cases were compared and showed to behave similarly. A comparison of the signatures with the expected patterns based on DTW showed that measured signatures of a given shape have a clear correspondence to the theoretical one of the same shape,

and a very low correspondence to those of different shapes. When shapes look similar to each other, such as cylinder and hyperboloid, and cannot be distinguished by their normalized signatures only, another parameter extracted from the signature (its integral) can be used to solve this problem. The cylinder is a flat shape and the integral of its signature is lower than the integral of the signature of the hyperboloid, which shape is concave. Thus, using both the signature and its integral, it can be concluded that they exclusively correspond to a specific shape, which is beneficial for shape identification tasks.

Signatures showed little variance within large ranges of size and pose (position and orientation) for each shape since the effects of both parameters on the signatures were discarded by the signature enhancement procedure. This brings several key features to the presented method. First, the angle of inclination of the object in the hand can be estimated through the inclination of the signature. Second, instead of focusing on specific object recognition, signatures allow shape identification for any object within the ranges defined in Table 2.2. Third, the signatures are generated and enhanced in about 10ms. Fourth, no palpation is needed, as a single grasp is enough for the shape to be identified. Thus, this method can be used in real-time for manipulation tasks where physical knowledge on the object is more important than its identification.

Even if the method was basically designed for solid shapes, results showed that objects with some elasticity and flexibility could also be identified. Such is the case of the plastic bottles from **Set 2** used in experiments, which shapes were successfully identified despite the deformations that forces applied to them caused.

Regardless the satisfactory results, this method present some limitations. The lack of tactile information makes it difficult to determine whether all fingers are in contact with the object or not. This is not a limitation of the method *per se*, but more of the proprioceptive data. In case this occurs, information contained in the signature would not provide proper information for shape identification. Furthermore, the position ranges within which this method works for all shapes was not the same. Each shape had a different position range. Tactile data or larger exploration would fix this problem.

In future chapters, contacts will be detected using proprioceptive information based on a tool develop to perform temporal analysis of the joint angle changes during the grasp execution. Furthermore, a generalization of the ranges with respect to position will also be possible thanks to the implementation of contact normals on the fingertips. We will show that by integrating this data to the feature vector, the confusion between the hyperboloid and the cylinder will no longer be an issue in this thesis work.

Chapter 3

Temporal Analysis of Proprioception

The perceptive performance of the haptic system is thought to be exclusively based on the capability of the sensors that conform it (cutaneous, thermal and kinesthetic sensors). However, Lederman et al. [2] suggested that it is also related to hand movements. By actively grasping or manipulating an object, variations of the sensory inputs are generated allowing a better perception of the spatial layout and structure of the objects. The performance of the haptic system are directly related to the generation of these variations and when minimal sensory variation is generated, the haptic system generally displays a poor performance [81–83].

There is an association between the hand movements, the sensory input and the type of information desired. In this sense, humans use stereotyped hand movement patterns to provoke specific sensory variations and, thus, obtain specific information about the object. Those hand movements are called *exploratory procedures* (EP). In chapter 2, we presented some associations established between a set of EPs and certain properties of the object, such as texture, hardness and shape. In the case of the shape, two EPs were associated with it: contour following and enclosure grasping.

Each of those EP, contour following and enclosure grasping, provides a different description of the object shape: contour following is associated with the exact shape of the object while enclosure grasping is associated with its global shape. This is because they employ different hand movements and therefore, generate different sensory variations. Thus, dynamic information generated during an EP contains cues of a particular dimension of its associated object property. The following paragraphs offer a brief explanation on how dynamic information has been exploited in robotics.

Dynamic information from an exploration process

Both contour following and enclosure grasping have been implemented on different manipulation systems. However, only contour following has been used to generate dynamic information. This is completely normal since its definition states it to be a dynamic EP. In order to reproduce it with robots, several approaches have been developed based on edge following. The techniques for edges detection depend on the nature of the data extracted from the tactile sensors. Some approaches convert the data coming from sensors arrays into images. This makes it possible to exploit image processing techniques, such as geometrical moments or Hu moments, to detect the edges [31, 84, 85]. Others take the contact force and orientation readings from sensor arrays to obtain edge information [86, 87].

Usage of dynamic information in the field of tactile shape identification has not been limited to contour following. Some authors have implemented other exploration procedures to generate dynamic data and extract information about the object. Moll et al. [3] utilized a manipulation system consisting of two flat palms covered with tactile sensors (Fig. 3.1a). To explore the object, they moved the flat palms to make the object rotate. They reconstructed the shape of the object by monitoring the haptic sensor data as the object is rotated. A set of differential equations were used to model the dynamics of the object. Based on the motion of the contact points, the authors were able to prove the observability of the shape, the motion and the center of mass of the object. Strub et al. [4] combined tactile information with kinematics inputs to build a representation of the shape of an object. They explored the object by rotating it between two robotic fingers and trained a neural-dynamic model (Fig. 3.1b) based on haptic exploration. Results showed that the algorithm was able to model two different n-gons from tactile data while rotating them within the robotic hand. Okamura et al. [65] used a finger sliding over a surface. In this work, six pattern features were defined (cusp, step, bump, pit, ridge and ravine). The trajectory of the contact between the finger and the surface was used to build a model of those patterns.

The cited methods proved dynamic data to be robust for object shape feature rendering. Object pose estimation should also be possible with most of these methods since contact position is vital for them to work. Thus, these methods showed that dynamic information represents a rich source of information on both the shape and the pose of the explored object. However, their implementation within the framework of this thesis is limited due to the EP they used.

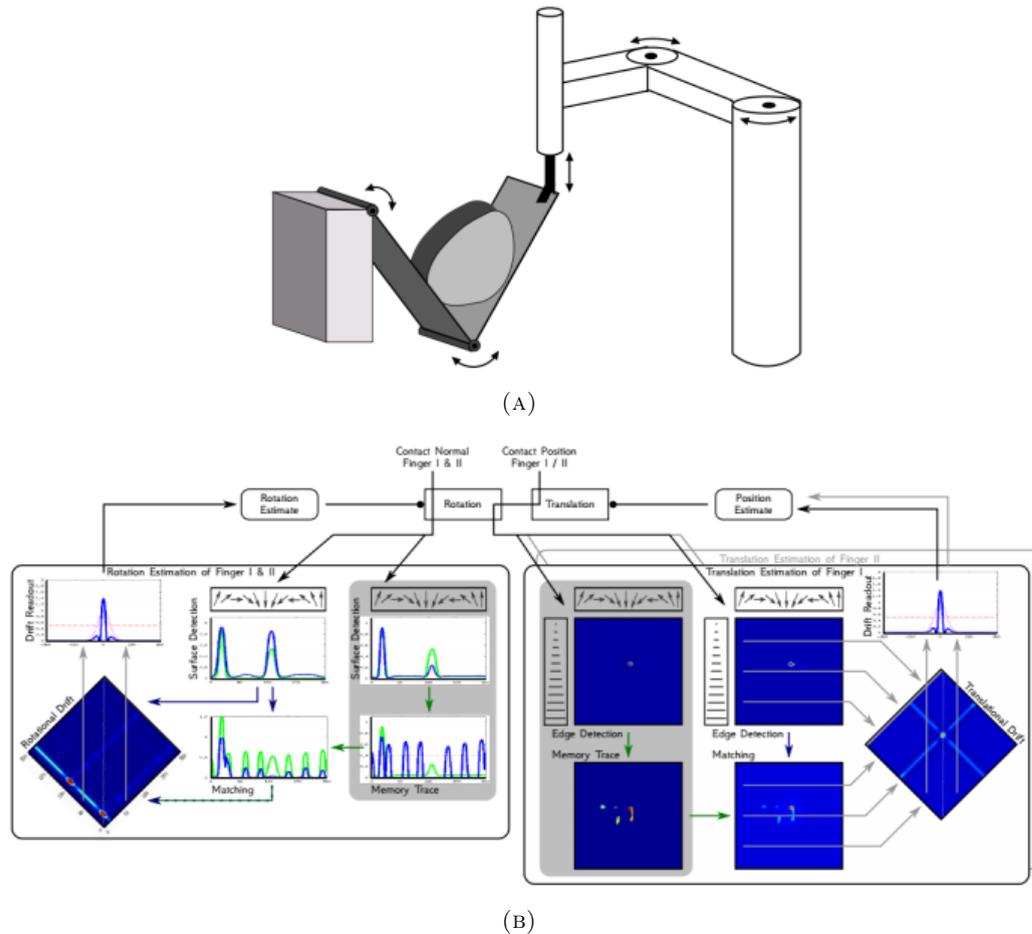


FIGURE 3.1: (A) Setup used by Moll et al. in [3]. Two flat surfaces attached to a fixed point on one side and a robotic arm on the other. As the robotic arm moves, it makes the angle between the flat surfaces change. (B) Cross section of a robot finger with the stress sensors used by Howe et al. in [1]. (C) Neural-dynamic model used by Strub et al. in [4].

First, most of them were not implemented on robotic anthropomorphic hands. To do so, important modifications would need to be done. Second, for some of these exploration procedures, the objects were not manipulated but explored and their only goal was to perform tactile object recognition. This is not a disadvantage *per se*, but it represents a conflict with our goal of performing tactile shape identification to enhance the manipulation dexterity. These two disadvantages are overcome with the second EP that can be used for tactile shape identification: the enclosure grasping.

Dynamic information from a grasping process

Enclosure grasping can be easily performed with almost any robotic hand. It also represents the initial interaction before further manipulation of the grasped object is carried out. It has therefore been largely used in tactile object recognition methods. That was the case of the works presented in the previous chapter. That includes the

cited works in the introductory paragraphs and the method proposed in this thesis. The results presented by these works were incontestably good, regardless of the limitations that their approach may have had. However, for the effects of this chapter, we will focus our interest on the type of data they used.

The methods that used enclosure grasping built shape representations on data generated *after* the grasp was completed. This means that the hand has stopped moving and the collected data is static. This aligns with the consideration of enclosure grasping as a static exploration procedure. However, not all haptic research agrees with such conclusion. In his work on proprioception [88], McCloskey expressed that 'every position is arrived at through a movement and every movement causes a change in position'. Thus, not only final finger configurations give information about the shape and pose of the object, but also the movements to reach those configurations.

Despite the promising information contained in the movements during the grasping process, references exploiting it are nonexistent. Furthermore, among the works using this grasping as EP, very few performed object pose estimation. This is particularly challenging for the methods based on touch information only. To extract dynamic information from the grasping process, proprioception should be particularly considered because it is responsible for providing position and movement cues [89].

Proprioceptive data as source of dynamic information

In this chapter, we propose to extract information from the proprioceptive data generated during the enclosure grasping procedure. A tool will be developed to analyze the fingers movements during the grasp execution. This tool represents the main contribution of this chapter and its applications may go from shape representation to feature extraction.

Further discussions about the different applications of this tool will be carried out in chapter 5. In this thesis, we focused on a particular application: the non-contact finger detection. To the author's knowledge, this is an issue that has not been addressed by any of the cited works on tactile object recognition. We consider this an important issue since most methods using enclosure grasping assume that all fingers must be in contact with the object. Therefore, their approaches may fail if this assumption is not met. Non-contact fingers will introduce false information leading to incorrect shape representations. Recognition rates will consequently be negatively affected.

In the following sections, we will outline the context of the presented method. First, we will explain how the Finger Configuration Image (*FCI*) is used to extract movement

information during the grasp execution. Then, a procedure will be presented to interpret the data extracted from the *FCI*. After that, two methods for non-contact finger detection will be introduced. Finally, experiments to evaluate these methods and their impact on the recognition rates will be presented. Both simulated and real results will be included.

3.1 Temporal analysis of the grasping process

The framework of this research work was described in chapter 1. This framework states that the interaction between an object and the robotic hand would be restricted to a single grasp. If a human being was to perform this same interaction, they would start collecting information about the object from the moment it enters in contact with their hand. Thus, the haptic information obtained *during* and *after* the grasping process would be exploited.

To extract data from both periods of time, we will divide the grasping process into two phases: the dynamic phase and the static phase. The dynamic phase concerns the data generated *during* the grasping process itself while the static phase relates to the moment *after* the grasp is completed. As already mentioned, we will focus in this work on the dynamic phase.

Our hypothesis about the dynamic phase containing information about the object is based on the following statement: as the robotic hand adapts to the object, the movements of its fingers will be constrained by the object that is being grasped, more specifically, by its shape and its pose within the hand. These constraints will therefore imprint cues about both the shape and the pose of the object on the dynamic information.

To exploit the dynamic phase, the proprioceptive information contained in the Finger Configuration Image (FCI) will be exploited. Since this image is updated online, the temporal analysis of its information is possible during the grasp execution. In this sense, an analysis of the changes in the *FCI* during the grasp execution will be performed. These changes will be recorded and depicted in an image designated as *G*. Once the grasp is fully executed, *G* will be used to obtain information about the grasped object such as its position and size.

3.1.1 Bases of the temporal analysis

As previously expressed, the *FCI* and grasp strategy are the foundations for the temporal analysis of proprioceptive data proposed in this chapter. Therefore, a complete understanding of how they are combined for this analysis is essential.

First, for what the method presented here is concerned, the grasp strategy described in chapter 2 will continue to be used. This grasp strategy imposes that the joint angles change separately and sequentially starting from the closest to the palm. Thus, only one joint angle will change at a time and it will not stop moving until a contact is detected or its maximum value is reached.

Second, the gray-scale values of the *FCI* do not represent measurements of independent joint angles. Instead, a relationship is established between the pixels of each column. This relationship makes the grayscale value of each row depend on the one of the previous row in its column.

When these two principles are combined, the behavior of the *FCI* is the following: as the adaptive grasp starts, all pixels of each column in *FCI* share the same grayscale value because the finger is straight. As they move, when a contact is detected, the moving joint stops and the following joint upstream in the chain starts moving. This causes the related pixel of the row to change. Thus, the pixels of a column will no longer share the same value.

3.1.2 Recording the time-related information

In order to record the variations depicted by the *FCI* during the grasp execution, a new image G is created which size is determined beforehand. The number of columns is the same as the *FCI* because the variations are computed for each column of the *FCI*. The number of rows $rows_G$ is related to some parameters of the robotic hand and the system that controls it. Some of those parameters are:

- The maximum angle each joint can reach ($\theta_{proximal}$, θ_{middle} and θ_{distal}). This parameter is related to the robotic hand itself and established by the grasping strategy.
- The rates at which the joint angles change ($\Delta\theta_i$) for each phalanx i . These parameters allow indicating the time taken by the hand to finish the grasping process. They are established by the grasp strategy.

- The frequency (f_G) at which the variations of the *FCI* are sampled in G . The higher this frequency is, the more rows will be needed.

These three parameters define the number of rows of G as follows:

$$rows_G = f_G \times \sum_{i=proximal}^{distal} \frac{\theta_i}{\Delta\theta_i} \quad (3.1)$$

It is advised that the recording process of the variations in the *FCI* during the grasp execution is performed at the same sampling frequency as the *FCI*. Also, it should start at the same instant as the grasp execution process. The variations of the gray-scale values in the *FCI* are recorded using the following expression:

$$G(y_t, x) = max_x(FCI_t) - min_x(FCI_t) \quad (3.2)$$

where the right term is the difference between the maximum and minimum gray-scale values in each column x of the *FCI* at the instant t during the grasp execution. Each time this difference is computed for each column x over time, the obtained vector is stored in the row y_t of G . In y_t , the subscript t indicates the dependence between the progression on the rows y as the time passes by.

The recording ends with the grasp execution process. The increasing differences between joint angles of each finger are reflected in the image G as a whitening of the corresponding columns (gradual increase of the grayscale value of the pixels of that column downward the rows).

In Fig. 3.2, the recording process is illustrated using the simulated model of the Shadow Hand. For this simulation, the sampling frequency f_G was set to 10Hz, the increment change to $20^\circ/seg.$ for the proximal phalanx and $25^\circ/seg.$ for the middle and distal phalanges. The maximum angles of each joint ($\theta_{proximal}$, θ_{middle} and θ_{distal}) to 85° .

3.1.3 Extracting information from image G

Once the recording is over, the generated G is processed to extract useful information. Fig. 3.3a shows the image resulting from the recording process shown in Fig 3.2. By visually analyzing this image, three areas can be identified and they are directly associated to the different stages of the grasping execution process. Starting from the top rows, which are the rows first filled during the recording process, an area of dimmed

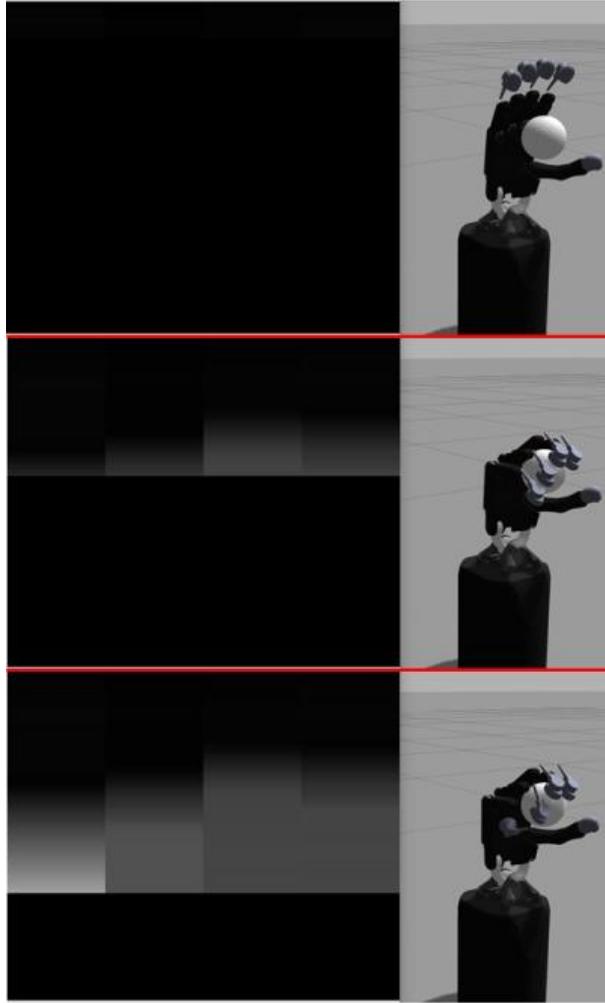


FIGURE 3.2: (a) Generation of the image G overtime. As the hand adapts to the object, the rows of G are been filled with the difference between the minimum and maximum gray scale values of the columns of the FCI .

pixels is observed. This area corresponds to the moments where the fingers seek to reach the object. Then a lighter area starts to appear, which depicts the folding of the finger after contact has been reached. Once the grasp strategy ends, the recording process is stopped and a third dimmed area is observed. This third area does not contain information about the finger movements.

Derivates

Focusing on the lighter area, a smooth change of the grayscale values is observed. This transition depicts the finger adaptation to the shape of the object. Once one finger has completely adapted, the grayscale value of its column becomes constant until the rest of the fingers finish adapting. In order to extract the zones of the image G where the finger adapts to the object, the pixels where the grayscale values increase need to

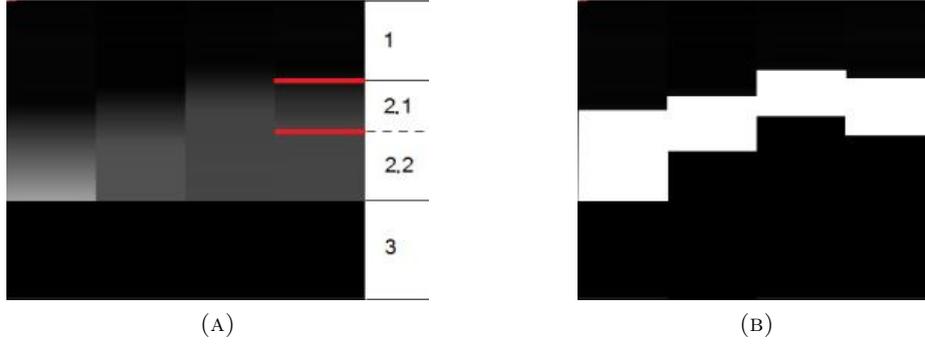


FIGURE 3.3: (A) Let us concentrate on the column on the right of the image G . In this figure, the illustration shows an example of what the image G would look like after the grasping execution is over. Each one of the signaled areas has its own meaning. *Area 1*: this area corresponds to the movement of the finger to reach the contact. *Area 2.1*: The finger adapts to the shape. *Area 2.2*: The finger has already adapted but other fingers haven't and the grasp execution continues. *Area 3*: Rows not filled because the grasp execution ends before reaching them. (B) Resulting binarized image after applying the Sobel filter to the image G .

be located. This is made by computing the derivatives of this image along the vertical axis. For this, a Sobel operation is applied to G . (Fig. 3.3b). This operator is usually used for edge detection in image processing and it is basically an differentiation operator that computes the gradient of the image intensity.

Binarization

After applying the Sobel filter to G , the resulting image is binarized. The threshold \mathbf{thres}_G will depend on the smallest $\Delta\theta_i$ established by the grasp strategy. Thus, \mathbf{thres}_G is computed as follows:

$$\mathbf{thres}_G = 0.90 \times 8 \times \frac{\min(\Delta\theta_i)}{f_G} : \Delta\theta_i \in [\Delta\theta_{middle}, \Delta\theta_{distal}] \quad (3.3)$$

where the constant 8 results from the convolution of the Sobel operator with the image G . This equation uses only the joint angles of the middle and distal phalanges. The joint angle of the proximal phalanx is not included in this equation because, as this phalanx moves, the joint angles of the middle and distal phalanges remain zero, and thus, the value $G(y_t, x)$ from equation 3.2 remains zero too. The threshold is fixed to 90% (which explains the constant 0.90) to compensate for the inaccuracies of sampling synchronization of the system.

This equation is defined as such because its goal is to detect the gradient between continuous pixels of the columns of G . These differences are basically established by the change rate of the angles $\Delta\theta_i$ but reflected on the image G as grayscale values, which

original values are modified by the Sobel operator. This threshold is used to binarize the image G , resulting in the image G_{bin} . This process is presented in Algorithm 2.

Algorithm 2 Pseudo Code for the FCI temporal analysis

Input:

FCI , instantaneous FCI .

thres_G , threshold expressed in equation (6).

Output:

G_{bin} , FCI temporal evolution binary image.

begin

$flag = false$

$n \leftarrow 1$;

while $flag = false$ **do**

$j \leftarrow 1$

for each column j of FCI **do**

$G(n, j) = \max FCI_j - \min FCI_j$

end for

$n \leftarrow n + 1$

 Update(FCI)

if GraspIsOver **then**

$flag = true$

end if

end while

$G_{diff} = \text{Sobel}(G)$

$G_{bin} = \text{binarize}(G_{diff}, \text{thres}_G)$

end

Since the kinematics are not considered in the FCI , a precise computation of the movements followed by each finger is not possible based on the image G_{bin} only. However, given the relationship of this image with the different stages of the grasp strategy (Section 3.1.3), it can be exploited to somehow interpret the movements executed by the fingers.

In Fig. 3.3a, the amount of rows in each area is proportional to the time taken by the fingers to complete the associated stage of the grasping strategy. This assumption implies that the sampling rate and the joint angle increments ($\Delta\theta_i$) are constant or nearly constant for each phalanx i . By knowing $\Delta\theta_i$, the amount of degrees the joint angles have changed on every row of G can be computed as follows:

$$Deg/pixel_i = \frac{\Delta\theta_i}{f_G} \quad (3.4)$$

Based on the values assigned to the parameters involved in this equation (section 3.1.2), the proximal phalanx moves approximately 2° for each sample taken from FCI to fill the image G shown in Figure 3.3a while the middle and distal phalanges move $2,5^\circ$ for each sample taken from FCI .

Given this interpretation, the final position of each finger and the time it took to adapt to the shape are intrinsically contained in the image G_{bin} . By extracting this information, both the size and pose of the object within the hand can be estimated. However, using the information coming from all fingers might pose a problem in situations in which one of the fingers did not enter in contact with the object. In such situation, incoherence will be present in the extracted information. Therefore, before estimating the size and pose of the object, a decision must be taken on whether a finger should be taken into account or not. In the next section, the method developed to solve this problem will be presented and explained.

3.2 Non-contact finger dropout using temporal information

One of the applications proposed in this thesis for the temporal analysis of proprioception is the detection of non-contact fingers. Most works on tactile object recognition assume that all fingers are in contact with the object once the grasp is over. However, this might not always be true. The object might be too small with respect to the hand for all fingers to be in contact with it. Also, a lateral displacement of the object might prevent one of the fingers from adapting to it. Such cases are illustrated in Fig. 3.4.

When non-contact fingers are included in the data for shape identification, the shape representation can be modified to the point of not corresponding to any of the expected patterns. Consequently, the recognition rate of learning algorithms might be directly affected. Therefore, any information coming from the non-contact fingers should be neglected.

Taking tactile information might seem the most logical solution to this situation. The detection of a contact would immediately be translated as a contact reached with the manipulated object. However, this approach has an important limitation: the finger might reach contact with itself or another finger. It is therefore proposed to use proprioceptive information to overcome this limitation.

In the absence of tactile information, proprioception can offer several cues for contact detection during the grasping execution. These cues can be extracted from position or movements of the fingers. Since this information can be obtained from the image G_{bin} , we are going to use that image for non-contact finger detection.

In this section, two methods to detect the non-contact fingers are developed based on position and movement cues, respectively. Experiments will be carried out to evaluate the performance of both methods in simulation and with a real robotic hand.

3.2.1 Position-based method

In this method, it is proposed to perform non-contact finger detection based on finger position. In general, the following hypothesis is made: as the finger finishes adapting to the shape of the object, contact information can be inferred from its adapting movements. To illustrate this, the following case is considered: if the movements correspond to a full closure of the finger, then no contact was reached.

Thus, this method performs an analysis of the movements made by the finger to reach its position. To do so, it exploits the information contained in the image G_{bin} once the grasping process is over. The interpretation of this image exposed in section 3.1.3 is used. More precisely, the position and the size of the white areas of the image G_{bin} are measured to detect the non-contact fingers. For this, a threshold needs to be defined for both the position and the size of the white areas.

Threshold definition

According to the grasping strategy implemented in the proposed method, when a finger does not enter in contact with the object, its joint angles do not change until the ones downstream the chain reach their maximum values. This is reflected in G_{bin} , and thus, can be detected. For this, the equation 3.4 is used to compute the number of pixels the first and second areas would have when the maximum joint angles ($\theta_{proximal}$, θ_{middle} and θ_{distal}) are attained in the grasping process. Two thresholds can be defined: one for the position of the finger (\mathbf{thres}_{pos} , associated with the first area) and the other one for the finger adaptation stage (\mathbf{thres}_{adapt} , associated with the second area). These thresholds are computed with the following expressions:

$$\mathbf{thres}_{pos} = \frac{\theta_{proximal}}{Deg/pixel_{proximal}} \quad (3.5)$$

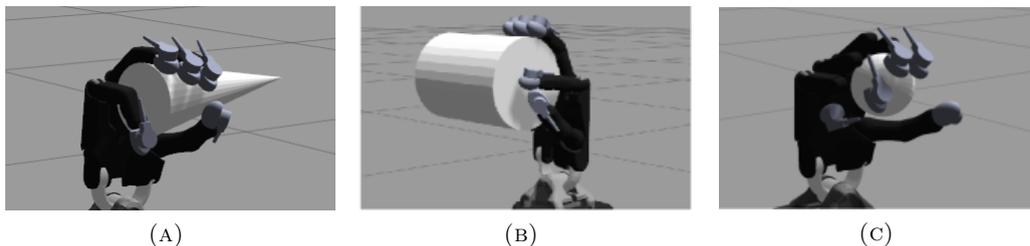


FIGURE 3.4: In the images (A) and (B) are illustrated the situations in which the lateral displacements of the shapes with respect to the hand prevent one finger to enter in contact with the object. The image (C) shows a sphere which is both too small and displaced so the little finger does not enter in contact with it.

Algorithm 3 Position-based method for dropout of non-contact fingers**Input:**

G , FCI temporal evolution binary image
 $\text{thres}_{\text{pos}}$, Threshold of pixels for the first area
 $\text{thres}_{\text{length}}$, Threshold of pixels for the second area

Output:

\mathcal{C} , set containing the indexes of contact fingers

begin

for each column j of G **do**

$x_{\text{init}} = \min(\{i \in [1, \text{rows}_G] : G_{\text{bin}}(i, j) = 1\})$

$x_{\text{end}} = \max(\{i \in [1, \text{rows}_G] : G_{\text{bin}}(i, j) = 1\})$

if $x_{\text{init}} < \text{thres}_{\text{pos}}$ or $(x_{\text{init}} - x_{\text{end}}) < \text{thres}_{\text{length}}$ **then**

$\mathcal{C} \leftarrow j$

else

Dropout j

end if

end for

end

$$\text{thres}_{\text{adapt}} = \frac{1}{2} \sum_{i=\text{middle}}^{\text{distal}} \frac{\theta_i}{\text{Deg}/\text{pixel}_i} \quad (3.6)$$

When the grasping strategy ends, the number of rows engaged in each area is measured. Algorithm 3 illustrates the procedure followed to measure it and dropout fingers, which will be contained in the set \mathcal{D} . The fingers are dropped out if both thresholds are simultaneously exceeded.

Phalanx coupling dependence

If coupling between phalanges is present in the hand, some modifications must then be made to equations 3.5 and 3.6. For example, equation 3.6 implies that θ_{middle} and θ_{distal} are not coupled and have different changing rates ($\Delta\theta_i$). However, the Shadow hand used for the experiments shown in this work had these two joint angles coupled, so, they always shared the same value. Consequently, their changing rates $\Delta\theta_i$ were the same. If this equation was used, the threshold would be larger than the value needed to detect the non-contact fingers. To correct this and adapt it to our hand, a modification to equation 3.6 was made and the threshold was computed as follows:

$$\text{thres}'_{\text{adapt}} = \frac{\theta_{\text{middle}}}{\text{Deg}/\text{pixel}} \quad (3.7)$$

Using equations 3.5 and 3.7, the thresholds are $\mathbf{thres}_{\text{pos}} = 42$ and $\mathbf{thres}'_{\text{adapt}} = 34$). In order to measure the performance of the presented method for non-contact fingers dropout, experiments were carried out on the simulated hand and the real platform. Section 3.2.3 will explain the procedure followed in those experiments.

3.2.2 Dynamic-based method

In this section, a method is proposed to perform non-contact finger detection based on an online analysis of the movements of the fingers during the grasp execution. The following hypothesis is stated: for a continuous and smooth-profile object, finger movements should not be abruptly different from each other.

For this method, an analysis of the movements of all fingers is performed online. Each finger is not looked at individually, but associations between their movements are established. While making these associations, the grasping strategy and the geometrical features of the test objects are taken into account.

Changes in the image G will be analyzed during the grasp execution. This allows detection to be done before the grasping process is over. Therefore, time is saved and grasp corrections can be made sooner, if desired.

For this method, algorithm 2 was modified so that the image G_{bin} is computed at the same frequency f_G . This image is then computed in real-time as the grasp is executed. Based on this condition, the Fig. 3.2 will rather look like the image shown in Fig. 3.5.

Dynamic behavior of G_{bin}

In more detail, the algorithm uses the changes of the subsequent rows n of G_{bin} to detect the non-contact fingers. Its functioning is divided into two phases. The first phase corresponds to the rows $\{n : n \leq \mathbf{thres}_{\text{pos}}\}$ and the second one to the rows $\{n : n > \mathbf{thres}_{\text{pos}}\}$. Two sets \mathcal{C} and \mathcal{D} are created that contain the columns j of the vector G_{bin}^n corresponding to the contact and non-contact fingers, respectively.

In the first phase, each column in G_{bin}^n is included in either \mathcal{C} or \mathcal{D} , depending on whether the finger is in contact or not with the object. The following rules are set to include a finger in \mathcal{C} :

- Every column j in G_{bin}^n that is equal to 1 will be included in \mathcal{C}

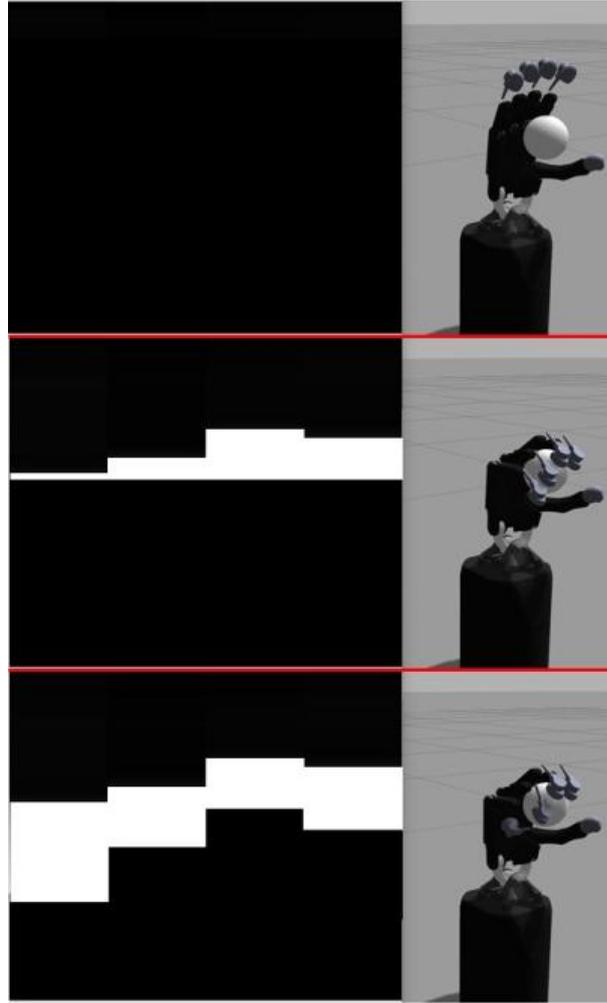


FIGURE 3.5: (a) Generation of the image G_{bin} overtime by computing it at the frequency f_G as the hand adapts to the object.

- If two fingers enter in contact with the object, all other fingers between them will probably do so as well. Thus, given two columns j^m and j^n already contained in \mathcal{C} , every other column between them is automatically included in \mathcal{C} as well. The function `ConsecutiveTouch` is created to detect this condition. It verifies that all indexes j in the set \mathcal{C} are consecutive ($j_{n+1} - j_n = 1 \forall j \in \mathcal{C}$). Otherwise, it redefines the set \mathcal{C} so that $\mathcal{C} = \{j \in \mathbb{N} : j_{first} \leq j \leq j_{last}\}$

Before presenting the criteria to fill in the set \mathcal{D} , let us introduce the function `AdaptOver`. This function monitors which fingers have ended their adaptation process. It verifies that columns j contained in \mathcal{C} have changed their state from 1 to 0. In such cases, the finger is considered as adapted, and the function `AdaptOver` includes it in the set \mathcal{A} .

Now, the following rules are set to include the columns j in the set of non-contact fingers \mathcal{D} :

- When a finger has finished adapting, the fingers next to it that have not started their adapting process will be considered as non-contact. Consequently, their corresponding columns j will be included in the set \mathcal{D} . The function `NotAdapted` is responsible for detecting such situation. It takes $j \in \mathcal{A}$ and verifies if its neighbors j_{+1} and j_{-1} are contained in \mathcal{C} . Otherwise, it includes it in \mathcal{D} .
- If the `threspos` is exceeded, all columns $j \notin \mathcal{C}$ are included in \mathcal{D} .

For $n > \text{thres}_{\text{pos}}$, the previous rules are no longer valid. The enclosure grasping will continue to be executed until all fingers in \mathcal{C} have adapted ($\mathcal{A} \subseteq \mathcal{C}$). As long as the grasp continues to be executed, the behavior of the columns contained in \mathcal{D} is observed by `PositionBased` function which exploits the function `AdaptOver` to detect if the fingers in \mathcal{D} have ended their adaptation process. If they did, it verifies that the thresholds established by the Position-based method are respected. The columns that respect these thresholds are taken out of \mathcal{D} and included in \mathcal{C} .

This method allows to stop the grasping execution as soon as all fingers in \mathcal{C} finish adapting to the object. As a consequence, time of the grasping execution is reduced. Section 3.2.3 will present the set of experiments made to validate this method and a comparison of the performance between position- and dynamic-based methods.

3.2.3 Evaluation of the non-contact finger detection methods

In this section, we will carry out a set of experiments in order to validate the presented methods to detect the non-contact fingers. A comparison of their performance will be discussed. For this, both simulated and real results will be taken into account.

Simulation

In the simulated environment, the data collection process included three of the five shapes used in the previous chapter (cone, hyperboloid and cylinder). The collected data of the sphere was also included since it contains non-contact fingers cases. The torus was not included in these experiments for two reasons. First, in cases in which its size was too little, fingers adapt to the whole shape rather than to one part of the ring. Second, if moved laterally, the finger meant to be the non-contact finger enters in contact regardless its size.

In order to collect the data, the horizontal positions of the cone, hyperboloid and cylinder were shifted from $x = 0$ so that one of the fingers does not enter in contact with

Algorithm 4 Dynamic-based method for dropout of non-contact fingers**Input:**

G_{bin}^n , newly computed row n of the image G_{bin}
 $\text{thres}_{\text{pos}}$, Threshold of pixels for the first area.

Output:

\mathcal{C} , set containing the indexes of contact fingers.

begin

while GraspExecution = true **do**

$[G_{bin}^n, n] \leftarrow \text{UpdateGraspProcess}$

$\mathcal{C}_{temp} = \{i \in [1; 4] : G_{bin}^n(i) = 1 \wedge i \notin \mathcal{C}\}$

$\mathcal{D}_{temp} = \{i \in [1; 4] : G_{bin}^n(i) = 0 \wedge i \notin \mathcal{D}\}$

$\mathcal{C} \leftarrow \mathcal{C}_{temp} \cup \mathcal{C}$

$\mathcal{A} \leftarrow \text{AdaptOver}(\mathcal{D}_{temp}, \mathcal{C})$

if $n \leq \text{thres}_{\text{pos}}$ **then**

 ConsecutiveTouch(\mathcal{C})

for every element j in \mathcal{A} **do**

$D \leftarrow \text{NotAdapted}(j, \mathcal{C})$

end for

else

$\mathcal{B} \leftarrow \text{AdaptOver}(\mathcal{D})$

for every element j in \mathcal{B} **do**

 PositionBased(j)

end for

 empty(\mathcal{B})

if $\mathcal{A} \subseteq \mathcal{C}$ **then**

 GraspExecution = false

end if

end if

end while

end

the object. As for the sphere, the shape was shifted in the horizontal axis within the range of $x = [-2, 2]$ cm. Thus, both cases (little objects or laterally shifted objects) were included. The vertical position y and the sizes of the shapes were within the decisive ranges defined in chapter 2. The Δr of each shape was computed so that the number of samples was roughly the same for each shape. Thus, 1787 (450/shape) samples were obtained and used in these experiments.

To measure the performance of both methods, the contact information given by the simulator was compared to the results obtained with the presented methods. The results are shown in Fig. 3.6 in the form of a two-bar histogram. The *detected* bar corresponds to the number of non-contact fingers detected by our methods. The blue section of this bar depicts the number of matches between the contact information given by the simulator and the results obtained with our methods. The gray section represents the false positives. The *measured* bar represents the amount of non-contact fingers successfully detected by the methods (in blue).

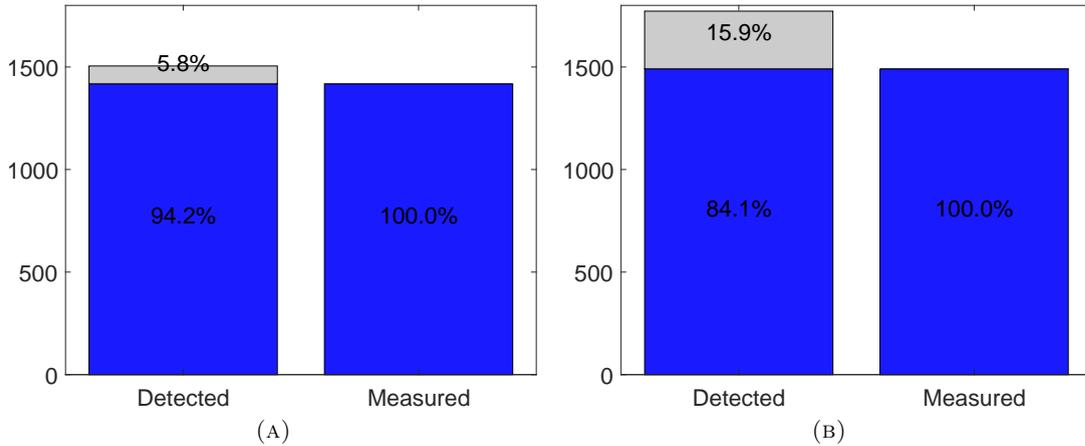


FIGURE 3.6: Histograms illustrating the performance of the dropout non-contact finger methods based on the *FCI* temporal analysis for (A) position-based and (B) dynamic-based methods. The cases in which the methods detected a non-contact finger are shown in the *detected* bar, whereas the cases when no contact was measured on one finger are depicted in the *measured* bar. The blue proportion of the bars corresponds to cases when the methods performed accurately and the gray proportion, when they did not.

Let us start by looking at the numbers that correspond to the histograms in Fig. 3.6 and explain their nature. In this figure, the *detected* bar depicts that the position and dynamic methods detected 1591 and 1819 non-contact fingers, respectively. These numbers include the false positives. The *measured* bars indicate that among the 1787 samples, 1433 non-contact fingers were detected. This difference of 354 samples is due to two factors. First, the range of x -positions of the sphere was not always enough to prevent fingers to enter in contact with the object. Thus, the non-contact finger scenario did not occur in 270 samples of the sphere. Second, a frequent situation presented itself in which the finger did not adapt to the object but its lateral side touched the shape. Thus, a contact was measured. This situation is due to the fact that the hand is not attached to a fixed reference and interacting forces with the object can alter its position. The other 84 samples were attributed to this situation. This proves the limitation of using contact information to detect non-contact fingers.

Let us now interpret these numbers and begin with the *detected* bars for both methods. The position-based method (Fig. 3.6a) exhibits a good performance by having a rate of false-positives of 5,8%. In the case of the dynamic-based method, a slightly larger number of false positives was detected (15,9%). This gives an advantage to the position-based method.

The *measured* bars show that both methods were capable of successfully detecting the non-contact fingers in 100% of the cases. If both performances were compared, the dynamic-based method exhibited a slightly lower accuracy than that of the position-based one. However, the dynamic-based method is still advantageous with respect to the position-based one since it allows to reduce the time of the grasp execution.

In order to measure the amount of time T_{diff} saved by using the dynamic-based method with respect to the position-based one, the following expression was used:

$$T_{diff} = \frac{n_{position} - n_{dynamic}}{f_G} \quad (3.8)$$

where $n_{position}$ and $n_{dynamic}$ are the rows n of the image G_{bin} at which the grasp execution process would stop when using the position- or dynamic-based method, respectively. T_{diff} was computed for every sample in which the method successfully detected the non-contact finger.

Fig. 3.7 shows the relative and cumulative frequency distributions of the time reductions of the dynamic-based method with respect to the position-based. As can be observed in these plots, the dynamic-based method accurately predicts the non-contact fingers 2.2 seconds before the end of the grasp execution in 50% of the cases.

Even though these results correspond to the simulation set-up, they represent a solid reference to show the advantage of the dynamic-based method.

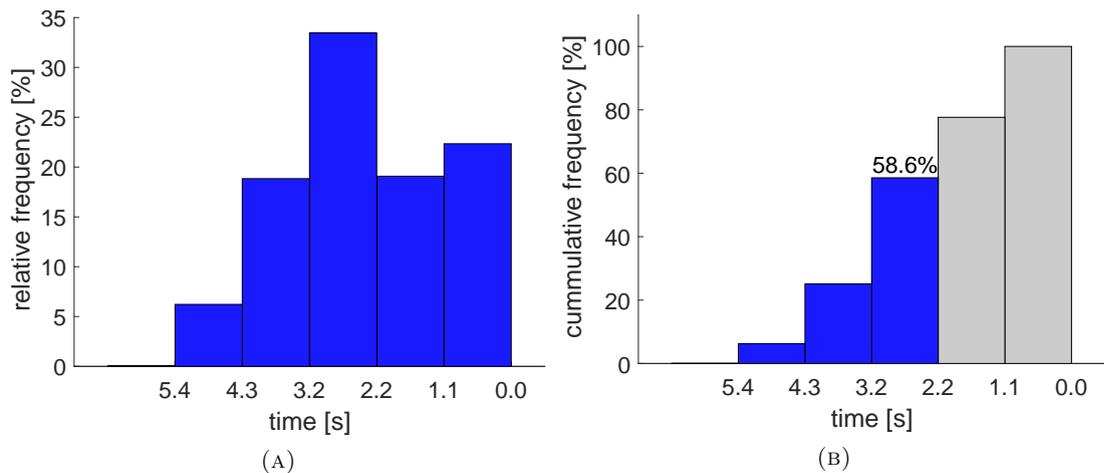


FIGURE 3.7: (A) Relative frequency distribution for the dynamic-based method with respect to the position-based method. (B) Cumulative frequency distribution. Blue bars correspond to the percentage of samples that saved more than 2.2 seconds using the dynamic-based method when compared to the position-based one.

Real Platform

To evaluate the performance on the real platform, the experiments carried out in simulation were reproduced with the real hand following the same procedure and using the same platform as experiments described in chapter 2. The experiments were done on three shapes: cone, sphere and cylinder. Objects are from the **Set 1** defined in chapter 2. As shown in Fig. 3.8, the grasp was executed so that the little finger would not enter

in contact with the object. Five samples were collected for each object, giving a data set of 45 samples.

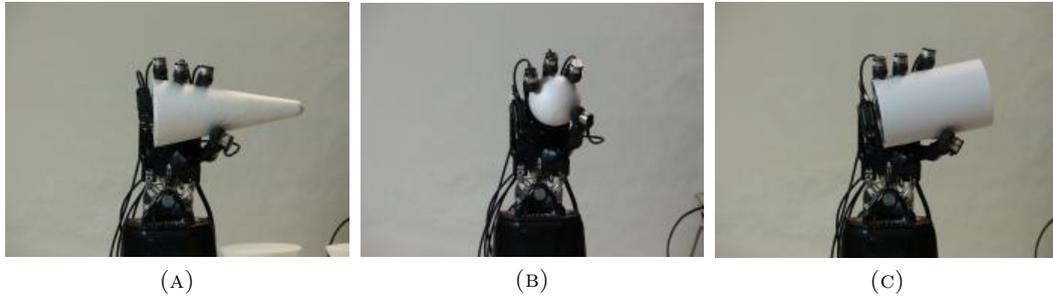


FIGURE 3.8: Grasps performed to validate the non-contact finger dropout methods. Three objects were used in these experiments: (A) Cone, (B) Sphere and (C) Cylinder.

Results of both the position- and dynamic-based methods on the real platform are shown in Fig. 3.9. The position-based method detected the non-contact fingers in only 60% of times, while the dynamic-based method detected 100% of them. Thus, the dynamic-based method performed better than the position-based one.

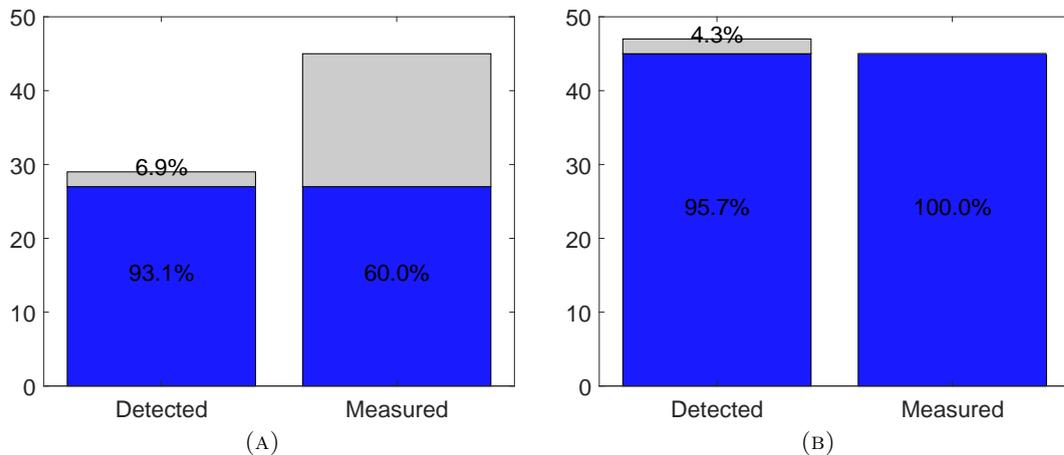


FIGURE 3.9: Histograms illustrating the performance of the dropout non-contact finger methods based on the *FCI* temporal analysis for (A) position-based method and (B) dynamic-based method. The cases in which the method detected a non-contact finger are shown in the *detected* bar, whereas the cases when no contact was measured on one finger are depicted in the *measured* bar. The blue proportion of the bars corresponds to cases when the methods performed accurately and the gray proportion, when they did not.

The differences with respect to the results obtained in simulation can be attributed to control perturbances present in the real hand. More specifically, when the experiments were conducted, the middle and distal phalanges of the little finger would take longer to start closing. This is due to the controller tuning: the effort needed by the little finger to start moving was particularly higher than the other fingers. As the little finger starts closing, it accelerates and moves faster than it should, making its corresponding white area on the image G_{bin} shorter. Therefore, the position-based method was unable to detect the non-contact finger because the $\text{thres}_{\text{adapt}}$ was not exceeded every time. An adaptation of the thresholds to this perturbation would fix the problem.

This situation does not affect the dynamic-based method because, by the time the little finger starts closing, it is already detected as non-contact finger and dropped out.

With respect to the time-related performance, Fig. 3.10 shows the results in the form of relative and cumulative frequency distributions. As can be observed, the results correspond to those obtained in simulation. In this case, 69,2% of the samples saved at least 11s. This difference has increased with respect to the simulated results as a consequence of the delay of the little finger to start adapting.

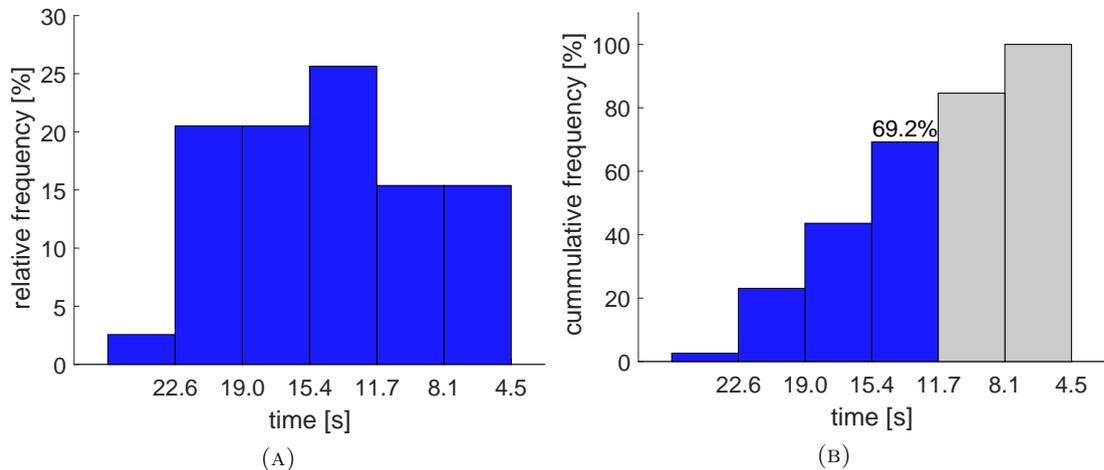


FIGURE 3.10: (A) Relative frequency distribution for the dynamic-based method with respect to the position-based method. (B) Cumulative frequency distribution. Blue bars correspond to the percentage of samples that saved more than 2.2 seconds using the dynamic-based method when compared to the position-based one.

Both method showed how position and movement proprioceptive cues can be used for non-contact finger detection. Generally speaking, the dynamic-based method performed better than the position-based method in both simulated and real environment. It was also confirmed that dynamic-based method saves grasp execution time, which might be beneficial for manipulation tasks. The position-based method also showed good results and its usage should not be yet neglected for possible applications with other robotic hands.

3.3 Proprioceptive signature correction

Dropping out non-contact fingers can be exploited for several purposes and improvements of the work presented in this manuscript. The correction of the signatures is one of them. This section introduces a modification of the signature generation process explained in chapter 2. Experiments and results will be presented to show the advantages of this modification on the shape identification rates.

3.3.1 Modified procedure for signature generation

For the signature generation process described in chapter 2, it had been assumed that objects were large enough for all fingers to be in contact with them when grasped. However, in section 3.2, it was demonstrated that this was not the case for all shapes and locations. For this reason, some signatures did not correspond to the expected patterns. This translated into limitations of the method for shape identification based on proprioceptive signatures.

In this section, we will introduce a modification to the signature generation process presented in chapter 2. This modification will allow the proprioceptive signature to overcome its limitation related to non-contact fingers.

Because non-contact fingers can be successfully detected using both position- and dynamic-based methods, we perform the signature generation after one of these algorithms has detected the non-contact fingers. The results of this modification is shown in algorithm 5. In this manner, non-contact fingers will not be taken into account in the signature generation process. By doing this, the generated signature will correspond to its pattern. Three conditions are established in order for this combination to be coherent with the purpose of signatures: identifying the shape of the object. These conditions are the following:

- At least three fingers have to be detected as contact fingers. This is because a minimum of three fingers are needed in order for the proprioceptive signatures to have concavities that differentiate them. If less fingers are involved in the signature generation process, they would not carry enough information to discriminate the different shapes.
- Contact fingers should be in continuous sequence and no non-contact fingers should be between them. This is related to the fact that the objects used in this work, and most objects in real environment, have continuous or one block shapes. Thus, if more than 2 fingers are in contact with the object, they must be consecutive fingers. Based on this principle, the middle or the ring finger will always be in contact with the shape.
- If one of the previous conditions is not respected, it is assumed that an error occurs in the process of grasp execution. Thus, no signature will be generated for shape identification

When these conditions are fulfilled, a modified *FCI* is created, which is called *FCI'*. This *FCI'* contains only those columns corresponding to the fingers that enter in contact

with the object. Once this image is filled, then the signature generation algorithm can be executed with this image as input. This process is illustrated in the Algorithm 5.

Algorithm 5 Signature generation after dropping out fingers

Input:

\mathcal{C} , set containing indexes of contact fingers
 FCI , Finger configuration image

Output:

S , proprioceptive signature.

begin

$j_{min} = j^* \in \mathcal{C} : j^* \leq j \forall j \in \mathcal{C}$

$j_{max} = j^* \in \mathcal{C} : j^* \geq j \forall j \in \mathcal{C}$

$l = |\mathcal{C}|$

if $j_{max} - j_{min} - 1 = l$ and $l \geq 3$ **then**

for each column j of \mathcal{C} **do**

$FCI' \leftarrow FCI(\mathcal{C})$

end for

$S = \text{GenerateSignature}(FCI')$

else

 Grasp Error

end if

Fig. 3.11 depicts a comparison between the signature generated with the unmodified (left) and modified (right) signature generation process. As can be observed, when the non-contact finger is dropped out, the signature changes and its resemblance to the expected pattern increases significantly. After their generation, the signatures go through the signature enhancement procedure.

3.3.2 Evaluation of signature correction process

In this section, we are going to evaluate the influence that the modification of the signature generation process has on the shape identification based on proprioceptive shape signatures. First, we are going to show the impact on the identification rates obtained in chapter 2. Then, the performance of the corrected signatures for shape identification will be tested.

Impact on the identification rate

Hereby, the impact of the dropping out of non-contact fingers have on the identification rate will be analyzed. Only results of simulation are presented.

Simulation. Data collected in chapter 2 is re-used and dropout methods are applied to it. The division of this data in training and test sets are the same as in chapter 2.

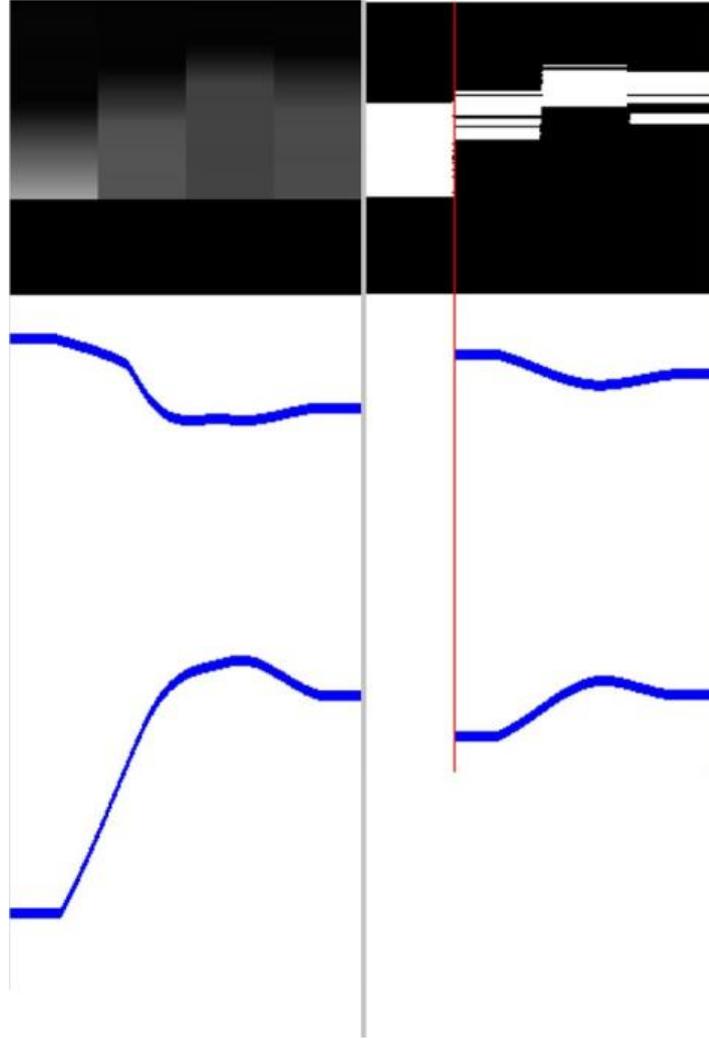


FIGURE 3.11: Signature correction by dropping out information from non-contact fingers. On the *left*, all fingers are involved in the signature generation process. On the *right*, non-contact fingers are dropped out and the signature is corrected.

Two SVM-models were trained with this data (SVM_{pos} for the position-based method and SVM_{dyn} for the dynamic-based one). The results are compared to those obtained in chapter 2.

Fig. 3.12 shows the confusion matrices obtained with this simulated data. The image on the left corresponds to the position-based method and the one on the right to the dynamic-based method.

When compared to the results obtained in chapter 2, recognition rates are not affected because the sample set used for these results was taken making sure all fingers enter in contact with the object. The only exception to this is the sphere because it was moved laterally preventing some fingers to enter in contact with it. Its recognition rate increased by around 10% for both dropout methods.

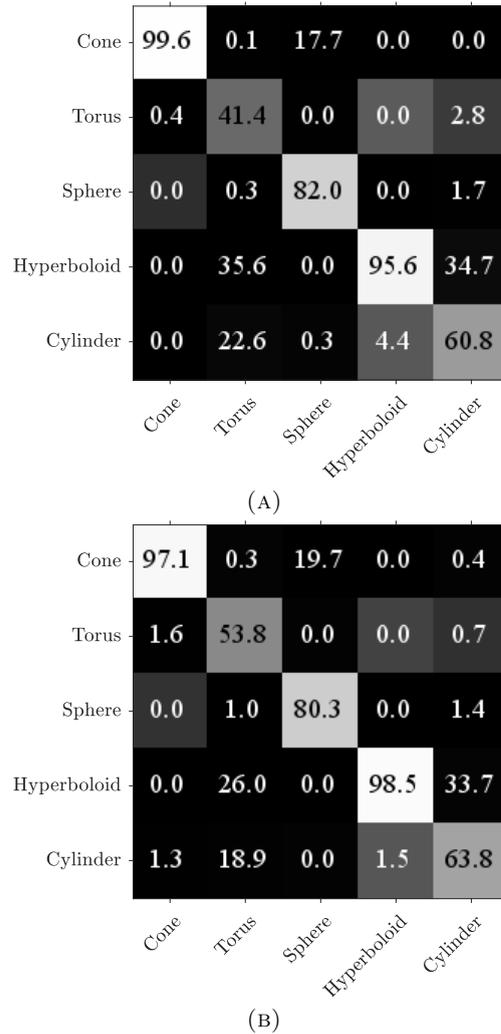


FIGURE 3.12: Confusion matrices showing the recognition rates. The upper row corresponds to the simulated results. The lower row corresponds to the results obtained with the real platform. The first column corresponds to the results obtained using SVM_{pos} . The second column corresponds to the ones obtained with SVM_{dyn} .

Shape identification with corrected signatures

This section is dedicated to analyze the performance of both the non-contact finger dropout method and the identification algorithms. Both simulated and real platform results will be presented.

Simulation. The data generated following the procedures presented in section 3.2.3 was used with the SVM_{pos} and SVM_{dyn} . Results are shown in Fig. 3.13. As can be observed, the identification rates are mostly lower than those obtained with experiments where all fingers were involved. This might seem contradictory with the fact that 100% of the non-contact fingers were successfully detected by both position- and dynamic-based methods. However, the false negatives and the displacements of the hand provoked by the interaction forces during the grasping process caused these discrepancies.

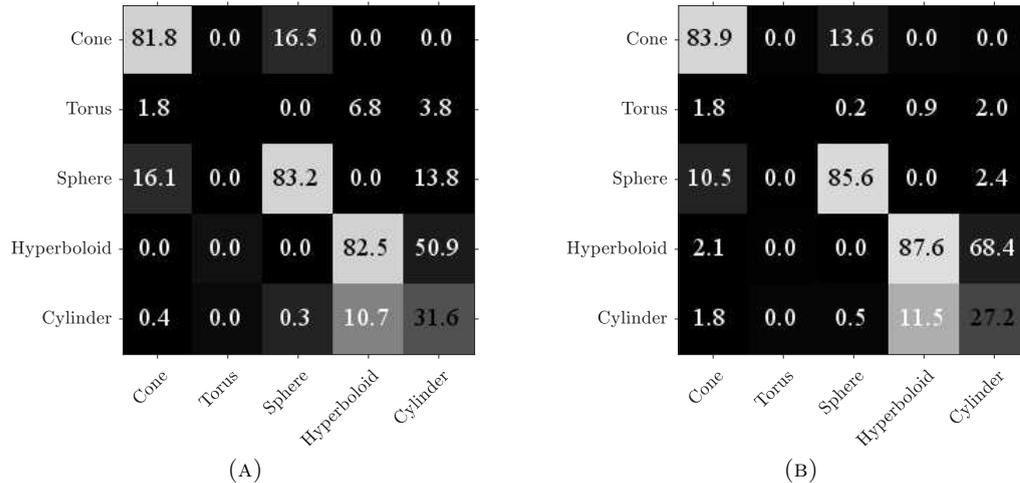


FIGURE 3.13: Confusion matrices showing the identification rates for non centered objects. The upper row corresponds to the simulated results. The lower row corresponds to the results obtained with the real platform. The first column corresponds to the results obtained using SVM_{pos}. The second columns corresponds to the ones obtained with SVM_{dyn}.

To further analyze this data, the Dynamic Time Warping (DTW) algorithm was used again to compute the similarities between the measured signatures and the expected patterns. These similarities were computed for both non-corrected and corrected signatures so that a comparison could be made between both cases. The same procedure as chapter 2 was used for this. As observed in Fig. 3.14a, non-corrected signatures do not correspond to any of the expected patterns and their corresponding DTW-distance exceeds the established threshold. In the case of the sphere, the threshold is respected because all fingers are in contact with the object in most signatures contained in its data.

On the other hand, the accumulated distance for all shapes is lower than the threshold established in chapter 2. This is verified for both the position-(Fig. 3.14b) and the dynamic-based methods (Fig. 3.14c). This explains why both SVM_{pos} and SVM_{dyn} also show good performance for corrected signatures.

Real platform. The data collected during the experiments carried out in section 3.2.3 with the real platform was used as input for both SVM_{pos} and SVM_{dyn} trained with simulated data. Results are shown in Fig. 3.15. Since the position-based method had a poor performance on this data, it directly affects the identification rates of the SVM_{pos}. As observed in the results on Fig. 3.15a, the cone was mostly confused with the sphere. This can be attributed to two factors. The first one is related to the aforementioned situation in which the thumb interferes with the adaptation of the index finger on the cone. The second reason is that when the signature of the cone is not corrected, it looks more like the expected pattern of the sphere.

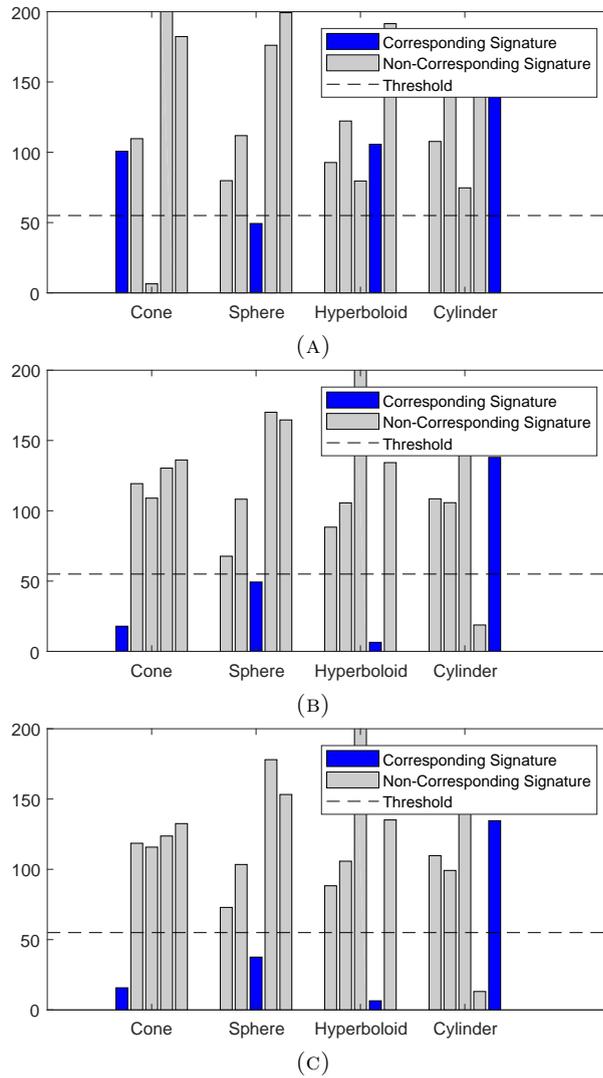


FIGURE 3.14: Euclidean distances between the expected patterns and the measured signatures in simulation. The blue bars are obtained by comparing the measured signatures with their corresponding expected patterns. The gray bars correspond to the DTW-distances between the measured pattern and the non-corresponding patterns. The dashed-line is the threshold (55) established in chapter 2 for signature correspondence.

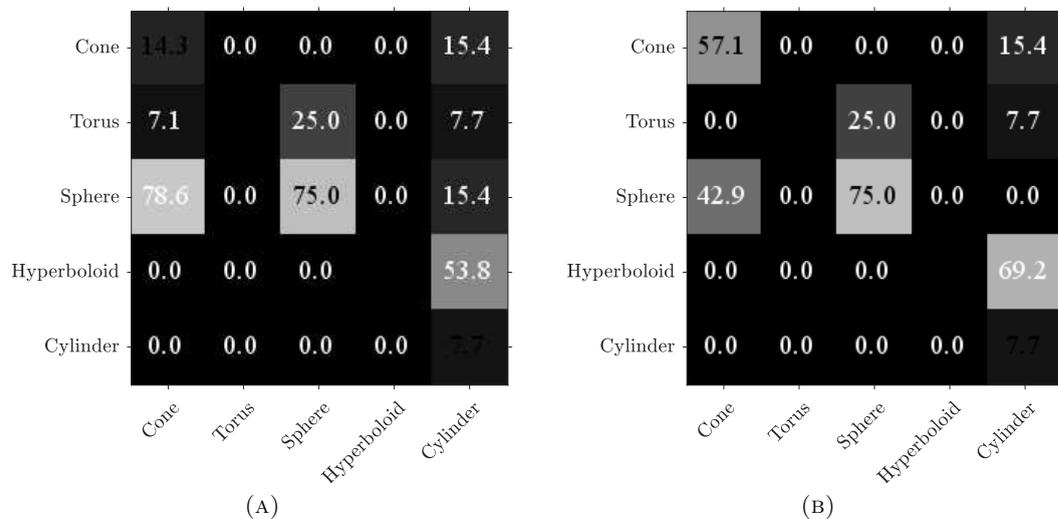


FIGURE 3.15: Confusion matrices showing the identification rates for non centered objects. The upper row corresponds to the simulated results. The lower row corresponds to the results obtained with the real platform. The first column corresponds to the results obtained using SVM_{pos} . The second columns corresponds to the ones obtained with SVM_{dyn} .

In the case of the dynamic-based method, the identification rates of the shapes depict a better performance. In this case, the confusion of the cone with the sphere is more related to the first reason explained in the prior paragraph. Fig. 3.16 depicts the DTW distances measured after the dropout non-contact finger methods were applied.

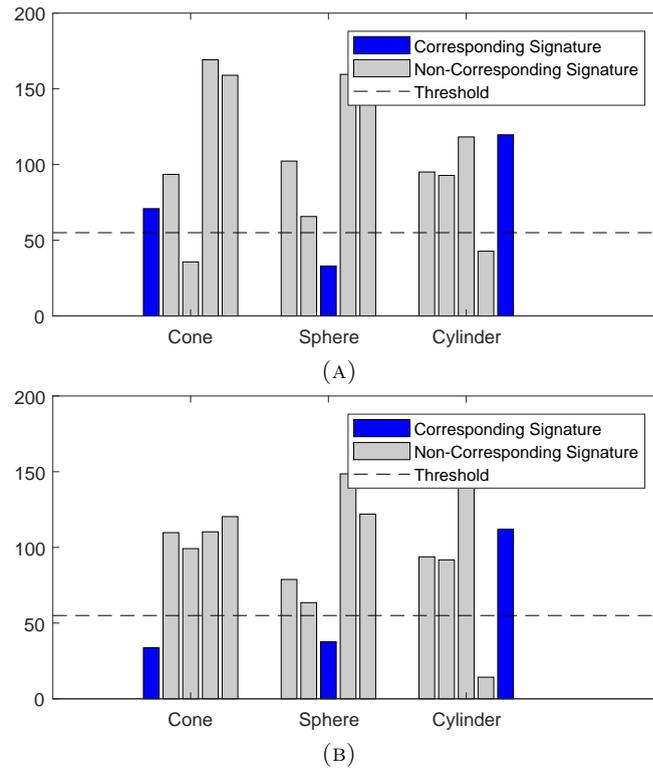


FIGURE 3.16: Euclidean distances between the expected patterns and the measured signatures in the real platform. The blue bars are obtained by comparing the measured signatures with their corresponding expected patterns. The gray bars correspond to the DTW-distances between the measured pattern and the non-corresponding patterns. The dashed-line is the threshold (55) established in chapter 2 for signature correspondence.

As can be observed in Fig. 3.16a, the position-based method did not succeed to dropout all fingers in the cone data, which explains the confusion of the cone with the sphere. On the other hand, the dynamic-based method was capable of dropping out the non-contact fingers more efficiently, which corresponds to the identification rates obtained with SVM_{dyn} shown in Fig. 3.16b.

3.4 Conclusion

In this chapter, we introduced a novel method to perform a temporal analysis of proprioception based on the Finger Configuration Image. This analysis allowed to determine the movements followed by the fingers during the grasping strategy execution. For this, changes on the *FCI* were recorded in an image G . Based on the interpretation of the information in this image, two methods were presented to detect or predict if one

of the fingers did not adapt to the object: the position-based method and the dynamic-based method. The first one allowed detecting the non-contact fingers once the grasping execution was over. The second one allowed predicting the non-contact fingers before the grasping execution ended.

When the performance of both methods was compared in simulation, they presented similar and good performance achieving to detect the non-contact fingers at all times. This same situation could not be reproduced in the experiments with the real platform. In these experiments, the performance of the position-based method was considerably lower as it detected only 60% of the non-contact fingers while the dynamic-based method maintained the performance reached in the simulated environment. The performance of the position-based method was particularly affected by technical issues regarding the hand controllers and results showed that the dynamic-based method was robust enough to maintain its performance in such situations. However, the position-based method should not be discarded since its performance can be considerably improved by fixing the problem with the controllers.

Beyond the robustness and better performance of the dynamic-based method with respect to the position-based one, it presents another advantage. This advantage is related to the fact that this method allows to detect the non-contact fingers before the grasping execution is over. This is particularly convenient as the grasp could be rapidly corrected if one of the fingers is detected as non-contact. Results showed, in both simulated and real experiments, that the dynamic-based method saves considerable amount of the grasping-execution time in more than 58% of the samples.

These methods were also exploited to correct the generated signatures in cases when one finger do not enter in contact with the object. To evaluate the performance in such application of the presented methods to dropout the non-contact fingers, both SVM and DTW algorithms were used. First, two SVM were trained following the same schema presented in chapter 2. For this, these SVMs were trained using the data obtained with the position- (SVM_{pos}) and dynamic-based (SVM_{dyn}) methods. Simulated results showed that, by dropping out the non-contact fingers, shapes can be successfully identified with both methods. Both methods depicted similar performance with simulated data. However, this was not the case with the data collected with the real platform.

Since the position-based method had a poor performance detecting the non-contact fingers on the real data, it directly affected the recognition rate of SVM_{pos} . This was not the case with SVM_{dyn} . Even if the results are not as good as those obtained in simulation, the dynamic-based method is considered to be valid for real applications.

Similar results were obtained using the DTW algorithm. In simulation, both methods performed similarly. However, this algorithm confirmed why the cone was being highly confused with the sphere when its signature is not corrected by the position-based method. Once more, the performance of the dynamic-based method was also better under this analysis.

Generally, the method to predict the non-contact fingers in a grasp execution presented in this chapter depicts a good and profitable performance on both simulated and real environments. When applied to correct the signatures, it did not necessarily solved the limitations of the signatures presented in chapter 2: the hyperboloid and the cylinder are still confused. However, this problem will be treated in the following chapter.

Chapter 4

Sequential Recognition of In-Hand Object Shape

Due to the unpredictable nature of realistic environments, tactile object recognition is confronted to several challenges. Putting aside the challenges imposed by manipulation and exploration tasks or data collection procedures, the challenges that we will address here are related to the recognition algorithm, more specifically, the training phase they go through before being able to perform any recognition.

Before a learning algorithm is capable of making predictions or identifying an instance, it goes through a training phase. During this training phase, the algorithm receives as input a set of samples representative of the different instances that it will have to identify. Realistic environments work in a manner that might affect the efficiency of the training phase of a learning algorithm. First, the amount of objects that the manipulation system will possibly handle can be very large. Second, new objects might be frequently introduced in the environment.

In this chapter, we will formulate a framework that will allow the training process to perform efficiently under both conditions. In this framework, the large amount of instances will be reduced by focusing on identifying the global shape of the objects instead of their exact shape. This was already discussed in chapter 2. Hereby we will discuss it further and a more robust solution will be presented. This framework will also propose a solution to the challenge imposed by the addition of new instances to the training set. This challenge has received very little attention in the tactile object recognition literature. We will address it by using an online training technique. Further explanations will be presented in the following sections.

In order to present how the proposed framework tackles each one of the aforementioned conditions, the structure of this chapter will be slightly different from the previous ones. This should allow the reader to have a broader view with an optimal focus on the contributions of this chapter. In this sense, the two following sections will present a state-of-the-art regarding the solutions for each condition and the contributions of this work on those aspects.

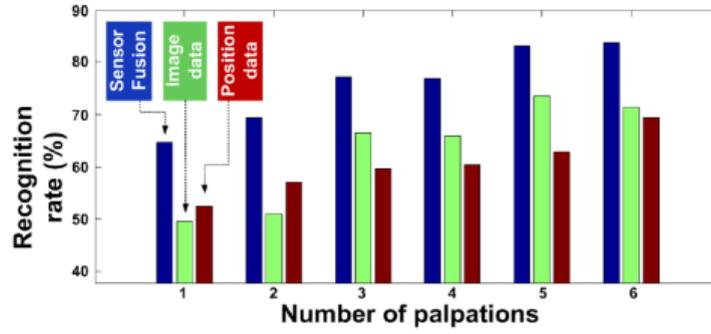
4.1 Reducing the amount of instances

Dealing with a big amount of objects to be recognized is a challenge for identification algorithms. The more shapes are added to the identifiable set of objects, the lower is the reported recognition rate. In [37], the recognition rate went from 96% to 84% when 8 objects were added to a set of 13. The same behavior was reported in [64], where recognition rates were reported to be 100% and 89% for sets of 5 and 25 objects, respectively.

This issue was discussed in chapter 2, and our approach to address it was to focus on global shape identification instead of specific object recognition. By doing so, the amount of objects that can be recognized is considerably increased with no detriment to the recognition rate. In this method, a unimodal shape recognition approach was implemented based on the so-called shape proprioceptive signatures. Remarkable results were obtained and the method showed to be robust enough to be applied to robotic hands.

In order to preserve the performance of the presented method while increasing the amount of shapes that can be identified with it, new dimensions need to be added to the shape representation. To do so, a new mode of information will be exploited and added to proprioceptive signatures, i.e., a multimodal shape identification approach will be introduced now. This decision is driven by the fact that other authors proved that multimodal data performs better than unimodal data. This is the case in [38], where the authors showed that proprioception and tactile sensing perform better combined than when exploited individually (See Fig. 4.1a). Same scenario was shown in [90], where different types of tactile and proprioceptive data were used for object recognition identification. Fig. 4.1b shows the results of this research. Notice how the researchers achieve a better performance by combining all the inputs.

In order to choose the type of additional data to include in the proprioceptive signatures, an analysis of the human haptic system was done. As discussed in chapter 2, the haptic system can provide with both cutaneous (touch) and kinesthetic (proprioception)



(A) Image taken from [38]

DATASET	RECOGNITION RATE (%)
All data	49.3
Without skin sensors	41.0
Only 6 axis F/T	38.0
Motor angles + reference angles + currents + spring displacements	31.5
Only skin sensors	17.0

(B) Image taken from [90]

FIGURE 4.1: Recognition rates for uni- and multimodal data.

information. So far, proprioception usefulness for shape identification has been object of study in this manuscript. Thus, the additional data to include in proprioception will be taken from tactile information. Let us now discuss how touch can be interpreted for the shape identification task.

4.1.1 Tactile data interpretation

To develop a shape representation of the explored object based on touch, our brain interprets the tactile information in two manners: as a first- or as a second-order information [91]. Fig. 4.2 depicts a finger sliding on a surface profile and the tactile and proprioceptive data are interpreted. Each representation contributes in a different way to shape identification, depending on the object scale with respect to the hand as well as the features that need to be identified.

Second-order information

During haptic exploration, a finger presses on the shape of the object causing the finger skin to deform. The second-order information $c(s)$ is the detected pressure gradient of

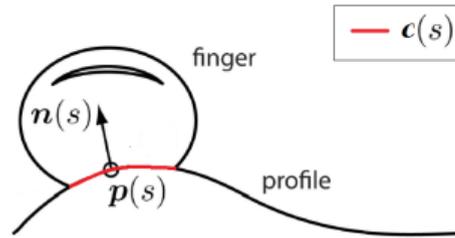


FIGURE 4.2: Geometric information obtained from a finger as it explores a surface s of an object: finger position $\mathbf{p}(s)$ (zeroth-order), contact normal $\mathbf{n}(s)$ (first-order), and local curvature shown in red beneath the finger $\mathbf{c}(s)$ (second-order). Best view in color.

the skin as it is deformed. This type of information is very useful to identify local shape features such as edges, corners and bumps. According to Wheat et al. [92], surfaces with curvatures lower than $-5, 8\mathbf{m}^{-1}$ and larger than $+4, 9\mathbf{m}^{-1}$ can be discriminated with a single finger pad from a flat surface with a 75% success rate. Note that the curvature is inversely proportional to the radius of curvature. So, in terms of radius of the shape, the second order information can detect curvatures of radius lower than $-0.17\mathbf{m}$ and larger than $+0, 20\mathbf{m}$.

In robotics, the second-order information has received a lot of attention with the development of pressure sensor arrays and many shape representations have been developed based on them (refer to chapter 1 for more detailed information). This information was not exploited in this PhD work for two reasons:

1. The present work is focused on global shape identification. Local features like bumps, edges and corners would be rather useful for manipulation purposes or specific object recognition.
2. The curvatures of the objects used in this work do not correspond to the ranges within which the second order information is the most accurate. Moreover, their sizes go beyond the fingertip scale which is the scale at which the second order information performs best.

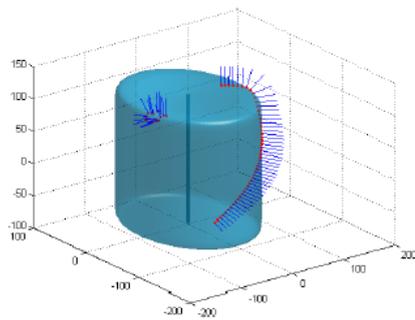
It is however prudent to say that using this information would enlarge the scope of application of the presented method. This will be further discussed in chapter 5.

First-order information

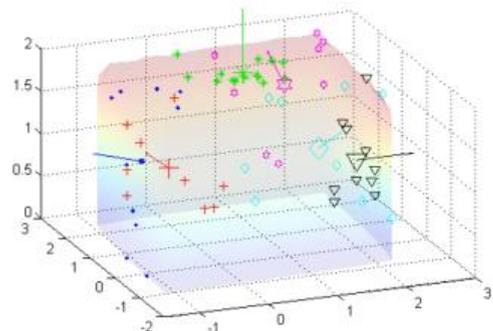
Another interpretation that our brain does with respect to the tactile information is related to the slope of the touched surface. This can also be seen as the normal to the contact surface $\mathbf{n}(s)$. According to Pont. et al. [93], the contact normals are

the dominant information to detect curvatures in the range $[-4; +4]\mathbf{m}^{-1}$. Furthermore, Wijntjes et al. [91] concluded from their research work that the orientation of contact normals dominates the curvature identification. They state that this is true for exploration lengths that go from the fingertip size to the arm length. Based on this conclusion, we decided to exploit the contact normals in this thesis.

In robotics literature, several shape representations have been developed based on contact normals. Most of them combine proprioception (zeroth-order information) with contact normals. In the case of contact normals, most approaches focus on model-based representations. Grimson et al. [94] used all possible contact normal orientations and locations that can be generated on a set of known polyhedral objects to generate a tree of feasible interpretations of this data. In order to prune the information and match it to an object, they established a set of constraints for the combination of the distances between the contacts and the directions of the contact normals. Jin et al. [95] computed the parameters of a super-quadratic equation using contact normal information. To do so, they first divided the shape into several patches using the k-means unsupervised clustering method on the contact normals and classified the object with a one-vs-one classifier constructed with binary Gaussian Process classifier. A similar work for super-quadratic estimation was presented by Bierbaum et al. [96]. Most of these approaches have not been implemented in robotic hands and a large amount of data is needed to achieve the results reported by the concerned researchers. This can be translated into a time-consuming exploration algorithm and sensor dependent procedure.



(A) Image taken from [96]



(B) Image taken from [95]

FIGURE 4.3: Contact normal and localization used in description-based representations.

The statistically-based representations are less frequent in literature in this matter. Schmitz et al. [90] used joint angles, sensor array and contact normal information arranged in a feature vector as inputs of a deep neural network. In this work, experiments were made on a Twenty-one hand [97] equipped with tactile sensor arrays, joint angles and force/torque sensors on the fingertips (Fig. 4.4). For the data collection process, 20 different objects of different shapes and hardness were grasped at least 20 times using

a power grasp. The feature vectors were formed combining 4 grasps, resulting in a training set composed of 5000 instances. The recognition rate reported by the authors was 88%. In spite of these results, the same limitations presented in chapter 2 remain; several grasps are needed before the object can be accurately identified. Moreover, no information whatsoever regarding the pose of the object within the hand is given.

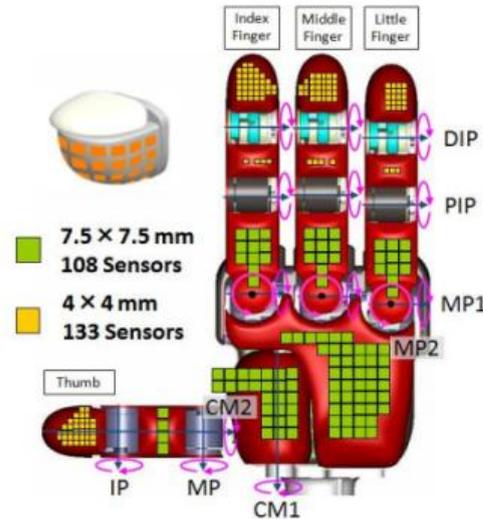


FIGURE 4.4: Twenty-one hand scheme.

In this thesis work, a statistical-based representation was used. The direction of the force sensors was exploited to perform object shape identification. The magnitude of the force sensor was neglected, since this information would not be useful for shape identification. The three components of the force sensed by the fingertips were concatenated, resulting in a 12-element feature vector for the normals. The learning algorithm used to create this representation was discussed in section 4.2.

4.1.2 Proprioceptive vs. contact normal

Research has shown that the first-order (contact normal) information is quite robust and sufficient to perform curvature identification [5, 93, 98]. Therefore, the following question arises: why continuing to use proprioceptive inputs if contact normals are sufficient for shape identification? To answer this question, let us take a look at Fig. 4.5. This figure shows the model developed by Wijntjes et al. [91] to describe how the minimum curvatures that can be discriminated with each order of information change with respect to the explored surface. Each order of information is represented with a different curve. The curve depicting lower values is taken as the most robust one because this means that it is capable of recognizing lower curvatures. This is the case of the first-order information curve. Thus, the authors concluded that it is the most robust one.

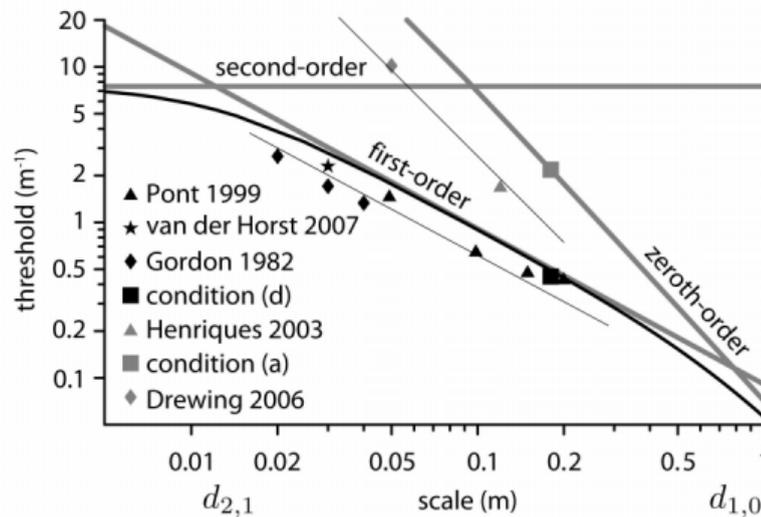


FIGURE 4.5: The values from literature plotted in log-log scales. The gray data points indicate thresholds obtained from zeroth-order information only. The black data points were obtained with real shapes. The gray lines describe the model itself. This data can be found in: Pont 1999 [5], van der Horst 2007 [6], Gordon 1982 [7], Henriques 2003 [8], Drawing 2006 [9].

This conclusion does not necessarily mean that the zeroth- and second-order information do not provide cues about the curvature. It rather states that for curvature discrimination, humans will probably base their prediction on the first-order information. Thus, the curvature cues given by proprioception can represent complementary information to the contact normal.

In this thesis, the use of proprioception gains legitimacy when the radii of the shapes and the length of the explored surface are taken into account. The largest radius for which this method works best was reported to be 0,14m (equivalent to a curvature of approximately $7,15\text{m}^{-1}$) in Chapter 2. The length of the explored surface for an enclosure grasping should be about 0,13m. Thus, according to the model illustrated in Fig. 4.5, proprioceptive information is capable of providing curvature cues for the setup used in this thesis.

It is also important to point out that this model is based on fingertip exploration. None of the results was obtained with full-hand exploration procedures. Curvature detection on the fingertips is not sufficient for 3D shape identification. Extrapolation of this model to full hand exploration requires a complex work that goes beyond the scope of this thesis. If taken as is, the robotic hand would need to have distributed sensors to compute all contact normals. However, most hands are equipped with force sensors at the fingertips only. Therefore, contact normals at the fingertips are going to be combined with proprioceptive information hereafter.

4.2 Addition of new instances to the training set

This framework proposes to deal with the addition of objects to the training set. This challenge directly concerns the learning algorithm. In this chapter, we propose to address it with a focus on the training strategy. Most of the cited papers regarding tactile object recognition perform an offline training, also called batch training. Such strategy supposes that the amount of objects to be identified is known *a priori* and the corpus of the training set is generated for those objects only. Using this type of training does not pose any problem when the amount of objects to be recognized does not change. However, in a realistic environment, the algorithm would need to be trained from scratch every time a new object is added to the training set. This could be time consuming as the amount of objects increases.

Online training techniques allow overcoming this problem. As defined by Rakhlin [99], *roughly speaking, online methods are those that "process one datum at a time"*. This processing makes it possible to update the model with new data, avoiding to perform the whole training process from scratch. This type of training is particularly interesting when:

- The data contained in the training set is too large to be handled by the system.
- The application is inherently online.

Many sequential training algorithms have been applied to different applications. Most of them concern vision problems, specially visual tracking. Ross et al. [100] used an appearance model of the object to track objects and adapted it to changes in pose, view angle, and illumination. This model was based on an online updated eigenspace using the Sequential Karhunen–Loeve (SKL) algorithm [101]. Babenko et al. [10] used an adaptive appearance model to track the position of an object in a video. Tracking took into account the position of the object in the first frame of the video. In order to update the position, the authors developed an online version of Multiple Instance Learning (MIL) [102]. The use of MIL is the main contribution of this work. Traditional methods would label individual samples which causes the appearance model to be updated with suboptimal positives and ends up drifting the tracker and degrading the model (Fig. 4.6). MIL, on the other hand, labels a bag of samples instead. A bag will be labelled as positive if at least one of the samples contained in it is positive. A bag is labeled negative if all the samples in it are negative. This allows the model to better handle ambiguities in the samples and correct the labeling of the individual images contained in the bags. Other online training techniques have been presented [103–105] and their applications mainly concern object tracking and recognition [106–108].

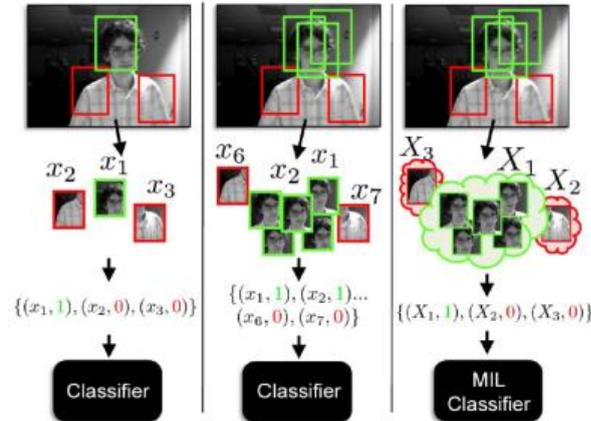


FIGURE 4.6: Methods to update a discriminative appearance model: (a) Using a single positive image. As one can notice, this image is not centered on the object perfectly. (b) Using several positive images. This is a too rigid condition for a classifier to learn a boundary. (c) Using one positive bag consisting of several image patches to update a MIL classifier [10].

To the author's knowledge, application of sequential training for tactile shape identification is not found in literature. This can be explained by the fact that this task is not necessarily inherently online, like object tracking. However, advantages of sequential training can be exploited in this field as well. In this chapter is presented a novel framework based on Neural Forests (NF) to sequentially add new instances or new shapes to the training set (i.e. open set recognition). Thus, the model can be sequentially trained as new instances are available, instead of training it from scratch. This allows reducing time of training as new shapes are added because the number of iterations to update the model is less compared to the offline training technique. The developed framework is detailed in the next sections.

4.3 Neural Forests Collection

One of the main contributions presented in this chapter is related to the use of an online training technique for tactile shape identification. As mentioned earlier in this chapter, all approaches cited in this manuscript rely on an offline training. In this section, is described the architecture of the Neural Forests Collection that will be used for tactile shape identification. We will specifically explain how the proposed framework can be trained sequentially. First, a general model for multi-class identification will be presented. Then, an explanation of the Neural Forests algorithm will be given followed by a formalization of its use for online learning. Finally, the procedure for sequential training will be exposed.

4.3.1 Multi-class shape identification

Let us start with a general multi-class model based on an one-vs-all classifier. Our main goal is to discriminate K different shapes, where K is given *a priori*. In this context, for an unknown object shape \mathbf{O}_k represented by its feature vector \mathbf{S}_i , the final class assignment \hat{k} is computed as follows:

$$\hat{k} = \operatorname{argmax}_k \{p(\mathbf{O}_k|\mathbf{S}_i)\} \quad (4.1)$$

where $p(\mathbf{O}_k|\mathbf{S}_i)$ denotes the probability of shape class k . This corresponds to the case in which an one-vs-all classifier is geared towards discriminating this shape class. For the training phase, the instances of the shapes are labeled in a binary manner. Thus, we consider all instances of shape k as positive by setting the ground truth $y_{i,k}^* = 1$ and all instances of other shapes as negative: $y_{i,k}^* = 0$. We then train a Neural forest to achieve this binary classification task.

4.3.2 Soft trees with probabilistic split nodes

For the purpose of this chapter, we are going to implement a Neural Forests (NF) [109]. NF were recently introduced as a hybrid of random forests and neural networks. It consists in ensembles of balanced probabilistic trees that each contains an ensemble of split and leaf nodes. Fig. 4.7 illustrates a scheme of one tree and each tree is a classifier. The numbered circles are called split nodes while the circles at the bottom of the tree are called leaf nodes. As can be observed, the split nodes distribute the probabilities into two outputs. Thus, the split node number 1 divides the tree in two subtrees: right subtree and left subtree. The depth of the tree is defined by the number of split nodes between the input channel and the leaf nodes. In the case of a binary NF dedicated towards discriminating a shape k , the probability is computed as follows.

$$p(\mathbf{O}_k|\mathbf{S}_i) = \sum_t \sum_{l \in \mathcal{L}} \mu^l(\mathbf{S}_i) y_t^l \quad (4.2)$$

where the prediction y_t^l of leaf l of tree t is either 0 or 1, and is randomly sampled during the initialization. μ^l denotes the probability to reach leaf node l , which is defined as a product of continuous split probabilities associated to each probabilistic split node n , that are parametrised by Bernoulli random variables $d^n \in [0, 1]$. The so-called activation $d^n(\mathbf{S}_i)$ for node n is defined as the output of a neuron layer (a Sigmoid function with weights $\{\beta_j^n\}$ and bias $-\theta^n$):

$$d^n(\mathbf{S}_i) = \sigma\left(\sum_{j=1}^f \beta_j^n S_{ij} - \theta^n\right) \quad (4.3)$$

Taking the expected value (which corresponds to an infinite number of samples from tree t), an example \mathbf{S}_i goes to the right subtree associated to node n with a probability given by the activation function $d^n(\mathbf{S}_i)$ and to the left subtree with probability $1 - d^n(\mathbf{S}_i)$. From a decision tree perspective, the successive activations $d^n(\mathbf{S}_i)$ thus define a soft routing through the tree, with oblique hyperplanes in the feature space.

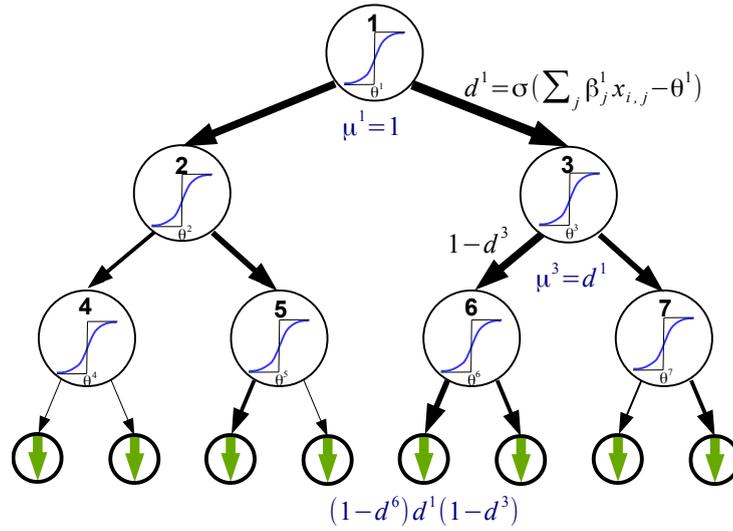


FIGURE 4.7: This image shows one tree of the hybrid random forests and neural networks.

4.3.3 Online learning with recursive backpropagation

The basis provided by the two previous sections is now going to be used to develop the online learning framework we will use in this chapter. For this, the work presented in [109] will be implemented, which suggests a two-step optimization procedure: first the split nodes are optimized using Stochastic Gradient Descent (SGD) and recursive backpropagation, from the bottom to the root of the trees. Then, the leaf nodes are jointly optimized for all trees in the forest using least-square minimization while the split nodes are left untouched. However, the latter step requires using all the samples at once, and therefore does not allow online training. Thus, in our case, we solely update the split nodes and the prediction nodes y_t^l remain constant during training. We found that such optimization was sufficient to obtain satisfying accuracy on a variety of classification and regression benchmarks, while enabling a faster, fully-online learning procedure.

The training phase is performed by sequentially presenting the feature vectors \mathbf{S}_i of randomly picked training instances i . A forward pass through the trees provides the values of the probabilities $\mu^n(\mathbf{S}_i)$ and activation $d^n(\mathbf{S}_i)$ for every node n and tree t . Particularly, the prediction error ϵ_t^l for a leaf node l of tree t can be computed as the cross-entropy loss between this ground truth value $y_{i,k}^*$ and the leaf prediction y_t^l stored in that leaf. The prediction error for the whole tree is thus equal to:

$$\epsilon_t(\mathbf{S}_i) = \sum_l \mu^l(\mathbf{S}_i) \epsilon_t^l \quad (4.4)$$

Hence, for any parameter ϕ^n (*i.e.* a feature weight β_j^n or the threshold value θ^n), the parameter update is given by equation (4.5) (with α the learning rate hyperparameter).

$$\phi^n \leftarrow \phi^n - \alpha \frac{\partial \epsilon_t(\mathbf{S}_i)}{\partial \phi^n} \quad (4.5)$$

Moreover, the derivatives of ϵ_t w.r.t. parameter ϕ^n can be calculated recursively:

$$\frac{\partial \epsilon_t(\mathbf{S}_i)}{\partial \phi^n} = \mu^n(\mathbf{S}_i) \frac{\partial d^n(\mathbf{S}_i)}{\partial \phi^n} (\epsilon_+^n(\mathbf{S}_i) - \epsilon_-^n(\mathbf{S}_i)) \quad (4.6)$$

with $\epsilon_-^n(\mathbf{S}_i)$ and $\epsilon_+^n(\mathbf{S}_i)$ the errors for the left and right subtrees, respectively:

$$\begin{cases} \frac{\partial d^n(\mathbf{S}_i)}{\partial \theta^n} &= -d^n(\mathbf{S}_i)(1 - d^n(\mathbf{S}_i)) \\ \frac{\partial d^n(\mathbf{S}_i)}{\partial \beta_j^n} &= S_{ij} d^n(\mathbf{S}_i)(1 - d^n(\mathbf{S}_i)) \end{cases} \quad (4.7)$$

And, the error backpropagated up to node n is:

$$\epsilon^n = d^n(\mathbf{S}_i) \epsilon_+^n(\mathbf{S}_i) + (1 - d^n(\mathbf{S}_i)) \epsilon_-^n(\mathbf{S}_i) \quad (4.8)$$

Thus, the error can be backpropagated from the bottom to the tree root and the hyperplanes corresponding to each split node can be updated accordingly.

4.3.4 Sequential training

Traditionally, offline training of the learning algorithm is implemented in shape identification. In this technique, the whole recognition model is trained from scratch every time a new shape is to be recognized. Instead, we propose to implement a sequential

learning technique. For a NF, since the structure of the trees is fixed *a priori* and the leaf node prediction settings y_t^l for binary classification do not depend on the number of shapes K , it is fairly easy to update an existing model for the recognition of a novel shape. Formally, let's assume that we trained a NF collection for the purpose of discriminating K shapes, and that we now want to add one shape, transforming our problem into a $K + 1$ recognition problem. In order to do so, we have to:

1. Train a new NF to discriminate the shape $K + 1$ from the K previous ones
2. Fine-tune the K existing binary classifiers to specifically discriminate shape K against shape $K + 1$.

Step 1 can be directly performed by using the procedure described above. Step 2 is done by applying a number N_u of class-wise balanced SGD updates, using equations (4.4), (4.5), (4.6), (4.7) and (4.8), thus adapting the oblique hyperplanes to the new class distributions.

4.4 Experimental set-up

Contact normals are used to perform shape identification with the framework proposed in this chapter. Therefore, the following section will present the procedure to collect the data corresponding to the contact normals. After that, is presented how iterations will be made in the sequential training.

4.4.1 Data

In order to test the proposed sequential training, both simulated and real data were used. The simulated data was obtained following the data collection procedure presented in chapter 2. The contact normals at the fingertips were measured with the simulated model of the ATI-nano 17 (See Appendix B). The same shapes as in previous chapters were used. The cone and cylinder were rotated 90° about the Z-axis so that the hand grasps them on their ends. This counted as new shapes which made a total of 7 different shapes (Fig. 4.8). Positions and sizes of these shapes were varied as indicated in chapter 2.

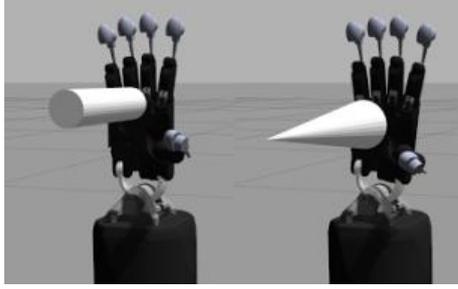


FIGURE 4.8: Illustration of the shape rotation with respect to the z-axis of the hand. From left to right: 90°-Rotated Cylinder, 90°-Rotated Cone.

For the data obtained with the real platform, the set of objects used for these experiments is shown in figure 4.9. Some objects of this set are also contained in **Set 1** and **Set 2** from chapter 2. The data collection procedure for the real platform presented in chapter 2 was used. Each object was grasped between 10 and 12 times.



FIGURE 4.9: Objects used for tests with the real platform.

4.4.2 Setting up the NF

Two NFs were generated, one for batch training and the other one for sequential training. For the results to be comparable regardless of the training approach, the hyperparameters of both NF were set so that an equal amount of updates is applied to the sequential and offline learning models. Therefore, for each object, a forest of 10 trees with a depth of 6 was used. As the shapes were added to the training set, the amount of updates was increased as indicated by (4.9). The starting number of updates was set to 1715 for two shapes. This number corresponds to the minimum number of updates needed to reach an average recognition rate higher than 90% for two classes.

$$Updates_{new} = Update_{old} \left(\frac{1}{nbShapes} + 1 \right) \quad (4.9)$$

where $Updates_{new}$ is the amount of updates to be performed for each tree as a new shape is added to the training set, $Updates_{old}$ corresponds to the number of updates made before this new shape was added, and $nbShapes$ is the amount of shapes present

in the training set before the new one was added. This equation ensures that every tree is updated the same amount of times for each shape.

4.5 Results and Discussion

In this section, the set of experiments carried out will be presented. First, we will analyze the performance obtained with different haptic inputs. Analogies with the human haptic system will be formulated. The NF used for this experiment is trained offline. After, the sequential training will be compared with the batch training. This comparison will be based on the recognition rates and the time taken for both techniques to complete the training. Finally, deeper analysis on the performance of the sequential training will be done. Results obtained with the real platform will be presented at the end of this section.

4.5.1 Unimodal vs. multimodal shape recognition

The performance of the zeroth- and first-order information for robotic shape identification was tested on NF. Each type of information was used individually (unimodal) as input for the NF. An offline training was performed to focus on the performance of the haptic inputs only. Fig. 4.10 shows the confusion matrices depicting the performance of NF for each input. The number of samples for each shape is different. Hence, 10 confusion matrices are computed using the same amount of samples for all shapes and then averaged.

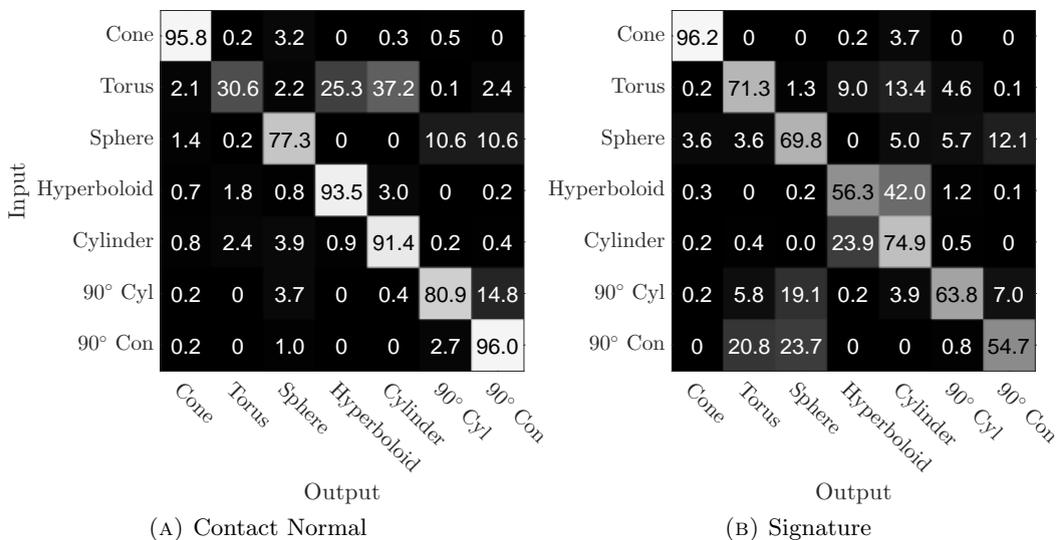


FIGURE 4.10: Average confusion matrices obtained using NF. Recognition rates are shown in %.

Results showed that the torus is highly confused with the hyperboloid and cylinder when the contact normals are used as inputs. In the case of the hyperboloid, the confusion can be attributed to the fact that the torus located in lower positions generates the same contact normals as an hyperboloid. Thus, it is confirmed that contact normals measured only at the fingertips do not offer enough information to discriminate 3D shapes with similar curvature profiles. The confusion of these two shapes with signatures was expected since similar results were obtained in chapter 2. However, results depict that proprioception offers enough information to double the recognition rate of the contact normals.

The confusion between the cylinder and hyperboloid when zeroth-order information (proprioceptive signatures) continues to happen (Fig. 4.10b). The differences with respect with the results obtained in chapter 2 are due to the absence of the integrate of the signature in the feature vector. This relates to the fact that proprioception is not capable of discriminating low curvatures. On the other hand, contact normals discriminate these two shapes with high accuracy (Fig. 4.10a). This is because these two shapes present different curvature profile, which allows contact normals measured at the fingertips to be sufficient for discrimination.

To some extent, zeroth- and first-order information are complementary to each other. To prove this, a new RF was trained using both types of information concatenated in the same feature vector (multimodal). This resulted in a 36-element feature vector. The confusion matrix in Fig. 4.11 shows the obtained results. As expected, recognition rates increased for almost all shapes. Torus continued to be confused with the hyperboloid and cylinder. However, its recognition rate was brought up to 75,6%. It is important to note that these results include the positions that were discarded by the ranges of accuracy of the signatures in chapter 2.

4.5.2 Offline vs. sequential training

This section is dedicated to evaluating and comparing the performance of offline and sequential training techniques for object shape identification using multimodal information. For this, objects were sequentially added to the training set in ten randomly chosen orders starting from two objects up to seven. Results (Fig. 4.12) showed that both techniques performed similarly for most cases and converged to the same result as the amount of objects increased.

These results make sequential training advantageous because an existing identification model can be retrained to identify a new object without needing the complete original training set.

Cone	96.5	0	0	0.2	3.2	0.2	0	
Torus	0.3	75.8	1.4	13.0	9.0	0.5	0	
Sphere	2.1	1.9	70.9	0.2	2.9	6.4	15.6	
Hyperboloid	0.2	0.1	0.2	97.0	2.1	0.3	0.2	
Cylinder	0.4	0.4	0.1	9.9	89.0	0.2	0	
90° Cyl	0.8	0.2	6.8	0	1.6	77.8	12.8	
90° Con	0	0.4	7.7	0	0	8.5	83.4	
		Cone	Torus	Sphere	Hyperboloid	Cylinder	90° Cyl	90° Con

Output

FIGURE 4.11: Confusion matrix for multimodal shape identification. Zeroth- and first-order information were concatenated in the feature vectors.

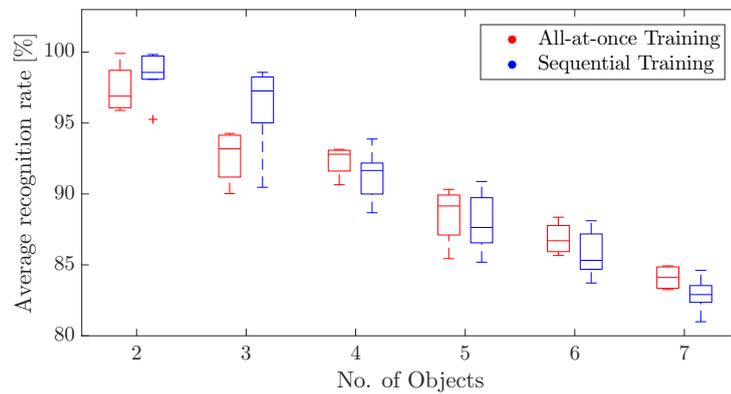


FIGURE 4.12: Recognition rate (%) obtained by computing the mean of ten tests. For each test, the order in which the objects were added to the training set was varied randomly. The tree corresponding to each object is updated 1500 times at the end of the training. Variance of the results is also shown. Outliers are shown in red for both training strategies.

In order to compare the time each training technique took to be completed, see Fig. 4.13. As one can notice, sequential training takes less time to be completed. This is because it allows to update the model as new shapes are added to the training set. Also, the distance between these two curves increases with the amount of object, indicating that the more objects there are, the more advantageous sequential training is.

4.5.3 Shape identification with sequential training

Fig. 4.14 shows the confusion matrix obtained using a sequentially trained NF. This confusion matrix shows the identification rates for each shape once all shapes have been added to the training set. To compute these recognition rates, ten tests were made in which the order of adding the objects to the training set was different and randomly

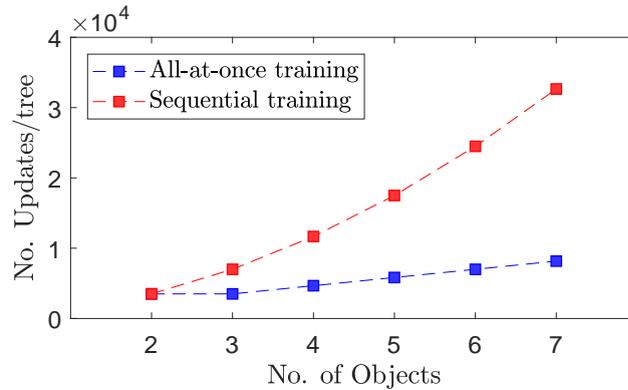


FIGURE 4.13: Number of updates per tree vs. the number of objects for offline (all-at-once) and online (sequential) training.

chosen. Results are comparable to those obtained with the offline training technique (Fig. 4.11) confirming that the proposed method performs as well as traditional training methods for object shape identification.

		Sequential						
		Cone	Torus	Sphere	Hyperboloid	Cylinder	90° Cyl	90° Con
Input	Cone	96.8	0.1	0.1	0.3	2.6	0.0	0.0
	Torus	0.5	75.0	0.8	13.9	9.0	0.6	0.3
	Sphere	2.3	2.9	59.8	0.0	2.4	6.9	25.6
	Hyperboloid	0.1	0.1	0.3	95.4	3.7	0.4	0.1
	Cylinder	0.6	0.3	0.1	10.1	88.5	0.4	0.0
	90° Cyl	0.8	0.5	5.3	0.1	1.7	75.6	16.1
	90° Con	0	0.9	3.8	0.0	0.2	5.9	89.1
			Cone	Torus	Sphere	Hyperboloid	Cylinder	90° Cyl
Output								

FIGURE 4.14: Average confusion matrices obtained using NF and sequential training.

We thought it would be interesting to see how the recognition rates of each shape changes as new shapes are added to the training set. This allowed us to determine how each shape affects the recognition rate of the others during the sequential training. Fig. 4.15 illustrates the results of this experiment. The positive values in the matrix depict an increase of the recognition rate. Analogously, a decrease in the recognition rate is depicted by a negative value. The **disturbances** correspond to the newly added shape while **disturbed** correspond to the shapes that were already present in the training set.

In most cases no significant changes were detected and less than 10% changes are reported. The exceptional cases where large perturbations were detected correspond to: first, the shapes generating similar zeroth- and first-order information (e.g. the

Disturbances	Cone	0.0	0.0	0.0	0.4	-1.3	-0.2	0.0
	Torus	-0.1	0.0	0.0	0.2	-0.8	-0.1	-2.5
	Sphere	-0.2	-1.1	0.0	0.2	-0.4	-29.9	-12.5
	Hyperboloid	-0.3	-4.8	-1.2	0.0	-15.9	5.4	7.6
	Cylinder	-0.4	-11.6	-1.1	-16.9	0.0	7.5	-1.6
	90° Cyl	0.0	0.5	-9.3	3.2	2.3	0.0	-12.4
	90° Con	0.4	1.9	-34.7	4.0	-0.9	-9.3	0.0
			Cone	Torus	Sphere	Hyperboloid	Cylinder	90° Cyl
		Disturbed						

FIGURE 4.15: Matrix of recognition rate disturbances between shapes. Shapes on the vertical axis (**Disturbances**) are the new shapes being added to the training set. Shapes on the horizontal axis (**Disturbed**) are the shapes already present in the set being affected (Best view in color).

sphere, the 90°-R cone and the 90°-R cylinder), and second, shapes generating similar zeroth-order information (e.g. the hyperboloid and the cylinder). Regardless of these perturbations, the performance of the sequential training was still similar to the ones obtained with the offline training technique (see section 4.5.2).

4.5.4 Real hand results

For the experiments with the real hand, a sequentially trained NF was used. Training of this NF was based on simulated data only. Results are shown in Fig. 4.16. The trees corresponding to the 90°-R cone and the 90°-R cylinder were neglected so that the same shapes as in chapter 2 were present in the confusion matrix.

When compared with results obtained with SVM in chapter 2, some differences can be noticed. The sphere is no longer confused with the torus. This can be particularly attributed to the use of a first-order information. The torus reduced its recognition rate, and the reasons for that were discussed in section 4.5.1.

4.6 Conclusion

A method was presented for object shape identification using Neural Forest (NF) Collection sequentially trained. The methodology presented in this chapter aimed to address the challenges faced by tactile object recognition in realistic environments.

Input	Cone	90.0	10.0	0	0	0
	Torus	0	63.6	0	36.4	0
	Sphere	0	0	100.0	0	0
	Hyperboloid	0	0	0	100.0	0
	Cylinder	0	0	0	25.0	75.0
		Cone	Torus	Sphere	Hyperboloid	Cylinder
		Output				

FIGURE 4.16: Confusion matrix obtained with a sequentially trained NF. Training was performed using simulated data. Data for testing was collected with the real Shadow hand.

In order to mitigate the impact of dealing with a large amount of objects, a multimodal approach for global shape identification was presented. The haptic data used in this approach was chosen based on an analysis of the functioning of the human haptic system. This analysis demonstrated that first-order information was the most robust source of information for curvature discrimination. For this reason, it was decided to include this information in the shape representation.

For the shape representation based on contact normal information, the raw values of the orientation of the contact normal vectors were used and the NF performed the statistical representations. This representation performed well for every shape present in the training set, except for the torus. This particular outcome results from the fact that contact normal information was measured at the fingertips only. Thus, only 2D features (curvature) can be detected, not 3D features (shape). This limitation is not due to the representation, but to the amount of information present in it. If contact normals were measured on other points of the hand, this limitation would certainly be overcome.

The proprioceptive signatures continued to be used for shape identification in this chapter. Regardless of this information being judged as the least robust for curvature discrimination in haptic literature, we concluded that this information is of usefulness in the context of this research. Results in chapter 2 further proved it.

By comparing results obtained with this data in an unimodal shape identification approach, it was shown how both information complement each other. When combined, an improvement of the recognition rates was achieved. Furthermore, unimodal results showed certain similarities with the human haptic system, i.e., contact normals

performed better than proprioception in discriminating shapes with smooth curvatures, such as the hyperboloid with respect to the cylinder.

The challenge of the dynamic training set in realistic environment was approached by working on the training technique. Sequential training was implemented. This differs to most approaches which use batch training. When these training techniques were compared, sequential training showed an advantage over batch training as it allowed to considerably reduce the training time with respect to batch training. With respect to recognition rates, both gave similar results. Globally speaking, they both converged to approximately the same recognition rate for 7 objects. We presented an analysis on how the recognition rates were affected by adding new shapes to the training set. Results showed that only shapes generating similar zeroth- and first-order information affect each other. Otherwise, the disturbances are less significant.

NF sequentially trained with simulated data showed to work well in identifying shapes described by data obtained with the real platform. Some minor differences were found with results reported in chapter 2 using SVM and explanation of those differences were presented. However, the overall performance is similar, proving sequentially trained NF to be usable for shape identification. This training technique allows reducing the training time as new objects are added, which is usually the case in realistic environments.

Chapter 5

Conclusions and future work

For robotic hands to reproduce the human dexterity in manipulation tasks, they require to count on perception systems that allow them to identify certain properties of the objects they are handling. Besides the sensors, these perception systems are composed by robust representations of the objects.

This thesis focused on creating representations needed for robotic hands to collect information from haptic inputs. Such information refers to both the global shape of the object and its location within the hand. The considered scenario consisted in a robotic hand receiving and grasping a solid non-deformable object.

The robotic hand used the sensor information it collected during its interaction with the object to create a haptic representation. Even if both tactile and proprioceptive inputs were utilized, a great portion of the presented work was dedicated to the exploitation of the proprioception. Chapter 2 and 3 presented how proprioceptive data was exploited.

5.1 Proprioception for object-related information extraction

The proprioception was at the core of the research presented in this thesis. A representation of the proprioceptive data of a robotic hand was created. This representation mimics some of the principles of human proprioception, such as, the hand kinematics topology. By doing so, the possibility of implementing it for more than just shape identification purposes was targeted. Moreover, a method was presented to use both static and dynamic data obtained from the hand during the grasp execution process.

5.1.1 Static proprioceptive data

Chapter 2 focused on static proprioceptive data. Once the hand had grasped the object, it used proprioceptive data to generate the *proprioceptive shape signatures*. Since we wanted to focus on parameters that could actually be used for manipulation tasks, we designed a signature enhancement process to make it invariant to the size and pose of the objects within the hand.

An evaluation of the quality of the signatures measured in both simulated and real environments proved that signature invariance is achieved within large ranges of sizes and poses. We also examined the capabilities of the signatures to give accurate estimations of the rotation of the object within the hand. With these results, we proceeded to evaluate the signature performance for shape identification. These results depicted object recognition rates up to 90%.

Further work on proprioceptive shape signatures

Proprioceptive shape signatures are generated under the condition that enclosure grasping is used. In [2], it was demonstrated that this exploration procedure was used for both global shape and volume identification. Therefore, further exploitation of the signature involves object size estimation.

To estimate the size of the objects based on the signatures, a data-driven model approach could be used. With this approach, kinematic information of the hand would not be required. Instead, only the generated signatures would be used to build the model.

This model should employ signatures that have not gone through the enhancement procedure. The vertical shift and distance between both lines of the signatures contain information about the size of the object. The enhancement procedure eliminates and normalizes those parameters, which is why enhanced signatures cannot be used for size estimation.

The vertical shift and the separation between the lines of non-enhanced signatures change with respect to the vertical position and the size of the grasped object. Data shows a proportional correspondence between those pairs of parameters. In other words, for a given radius and a vertical position, the vertical shift and the separation between the lines of the signatures will remain the same. This is independent of the shape of the object.

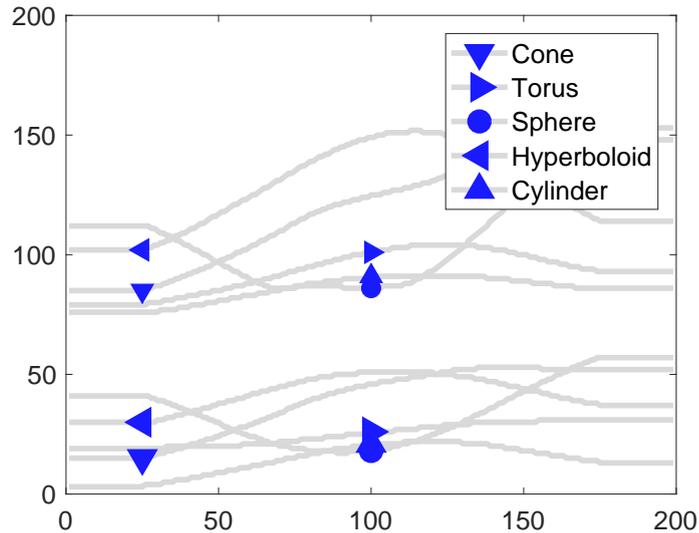


FIGURE 5.1: Signatures measured in simulation for each one of the shapes used. The blue markers depict the points on the signatures. Those points correspond to the fingers that adapted to the same radius at the same position.

5.1.2 Dynamic proprioceptive data

As discussed in chapter 3, dynamic information from the grasping process was extracted and a tool was proposed to interpret it. The application given to this tool consisted of the detection of the non-contact fingers. This is an important issue in the object-shape identification field since most of the methods neglect to consider that this situation might occur. The consequences of using a shape representation with erroneous information from non-contact fingers could highly affect the recognition rates of the learning algorithms.

Two methods were presented for non-contact finger detection. The first one, which is position-based, took into account the movements of the fingers during the hand grasping process. The second one, defined as dynamic-based, analyzed the movements in real time and predicted the non-contact fingers before the grasping process was over.

Experiments, also carried out in simulated and real environments, showed that both methods performed well. However, the dynamic-based method had an advantage over the position-based one, since it saves time by predicting the non-contact fingers before the grasp execution process ends. Experiments were carried out to test the advantages of detecting the non-contact fingers for shape identification. Results showed that by dropping out the non-contact fingers the capacity of the signatures to perform well was extended.

Dynamic proprioceptive data for shape identification

As stated in chapter 3, shape information can potentially be extracted from dynamic data generated during the grasping process. The applications for this data are complementary to the static data. Thus, these perspectives are more related to improvements of the work presented in the previous chapters.

The signatures proved to successfully identify the shapes of objects. Information coming from the dynamic phase of the enclosure grasping should also be usable for tactile shape identification. To prove this, consider the examples of the images G_{bin} shown in Fig. 5.2. Each one of these images corresponds to one of the five shapes considered in this thesis work. As can be visually observed, both the position and the size of the white areas in the images G_{bin} show different patterns.

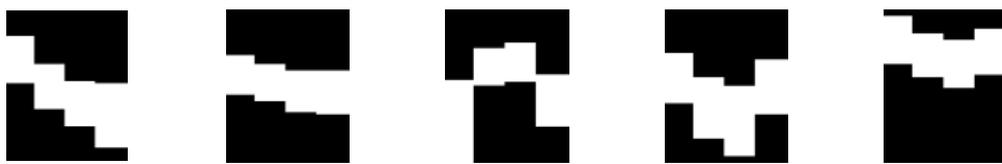


FIGURE 5.2: Images G_{bin} obtained for the different shapes in a simulated environment. From left to right: cone, torus, sphere, hyperboloid, cylinder.

5.2 Sequential shape identification using multi-modal inputs

In chapter 4, a framework was presented to sequentially train a collection of neural forests. This framework allowed to include new shapes into the set of identifiable shapes without needing to train the model from scratch. Instead, the model was updated as new shapes were added, making it possible to reduce the training time with respect to batch training techniques.

Both proprioceptive shape signatures and contact normals were used as inputs for this algorithm. The contact normals were measured at the fingertips of the robotic hand. Analogies with the human haptic system were made based on the results obtained with these inputs. In the case of the signatures, the confusion of the hyperboloid with the cylinder was attributed to its low curvature along its main axis. Similarly, a human would not be able to identify low curvatures using proprioception only. Contact normals are the main source of information in this case. This same scenario reproduced itself with the robotic hand: contact normal information was efficient in discriminating the hyperboloid from the cylinder.

The sequential training performed similarly to the batch training technique in terms of recognition accuracy. However, it reduced the training time by approximately 75%, in the case of having seven shapes in the training set. This represents an important

advantage of the sequential training method since, in realistic environments, new objects might always appear and the system should be able to incorporate them in its set of identifiable objects. Moreover, when there is a vast amount of objects, it would make a batch training technique extremely time consuming if implemented.

In future work, it would be interesting to integrate sensor array readings to the inputs of this algorithm. That would increase the complexity of the shapes that can be identified. Furthermore, information obtained by grasping each shape from different angles could be put in the same instances.

5.3 Summary

This research was intended to broaden understandings of robotic tactile object recognition. The solutions proposed here were meant to provide both shape and pose estimation of the object. The capacities of proprioception as source of information for this task were proved. A tool was created to endow robotic anthropomorphic hands with proper representations of their proprioception. Based on this tool, both static and dynamic proprioceptive information during a grasp execution were exploited for different purposes.

The static information, measured after finishing the grasping process, was exploited to create the proprioceptive shape signatures. This descriptor made it possible to identify global shapes of objects independently of their size and position within the hand. Signatures also made it possible to measure the rotation angle of the object within the robotic hand.

In this thesis, it was shown that information collected during the grasping process also contains information about a given object. Thus, a tool was developed to extract information from that process and detect non-contact fingers. Our results revealed the benefits of this tool for shape identification by enhancing the generation of the signatures with static information.

Finally, this thesis proposes a framework to sequentially train a Random Forest algorithm. Results showed that this training strategy offers similar results as the traditionally used batch training but training time was considerably reduced with respect to the batch training strategy.

The work presented in this thesis could still be further developed and its application could be in manipulation tasks, object size and pose estimation, robotic prosthesis feedback design, multimodal object recognition and so on.

Appendix A

From joint angles to central angles

This appendix shows how central from a grasped circle are equivalent to the angle subtended by the arc formed by the contact points between the finger and the circle. In the Fig. A.1, points O,A,B and C define a quadrilateral. The sum of its internal angles is equal to 360° .

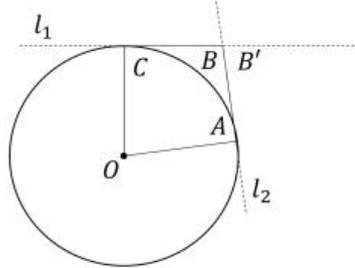


FIGURE A.1: Tangents l_1 and l_2 and the cords \overline{OA} and \overline{OC} forming a quadrilateral.

$$\angle O + \angle A + \angle B + \angle C = 360^\circ \quad (1)$$

Let l_1 and l_2 be tangents to the circle O at the points A and C . Hence, the cords \overline{OA} and \overline{OC} are orthogonal to l_1 and l_2 , respectively and

$$\angle O + \angle B = 180^\circ \quad (2)$$

Because $\angle B'$ and $\angle B$ are supplementary angles,

$$\angle B' + \angle B = 180^\circ \quad (3)$$

Substituting (2) in (3) demonstrates that

$$\angle O = \angle B' \tag{3}$$

Appendix B

Shadow Hand

In this appendix, we will be describing the experimental setups that will be used for all the experiments to test the methods proposed in this manuscript. Except in cases of modifications for specific purposes, all chapters refer to this appendix for practical experiments explanations.

Most of the experiments were carried in both simulated and real environments. Therefore, the information about both environments will be given here. Furthermore, the procedures followed in each environment for the data collection process will also be explained.

B.1 Shadow Dexterous Hand

The Shadow Dexterous Hand was used in experiments (Fig. B.3). This hand [42] is a 24-DOF system that reproduces closely the kinematics of the human hand. All fingers of the robotic hand are the same length, however, they are not aligned. Each one of them is located at a different height with respect to the forearm. The thumb has a different length and kinematics from the rest of the fingers. Just like the human hand, the middle and distal phalanges are coupled so that the middle joint has always a greater or equal angle to the angle of the distal joint. The kinematics of the hand are optimized so that it resembles the human kinematics as much as possible. Table 1.1 shows the limits of each one of the joints.



FIGURE B.1: Shadow Hand used in experiments.

TABLE B.1: Maximum and minimum joint angles that the joints from the index, middle, ring and little finger can reach. The other DOF of this hand are not shown here because they are set to a particular value. If needed, they will be specified in the corresponding chapters.

Joints	Degrees		Notes
	Minimum	Maximum	
Proximal joint	0°	90°	
Middle joint	0°	90°	coupled
Distal joint	0°	90°	

B.1.1 Sensors

Sensors play an important role in object shape identification since they provide with the information of the environment that can be used later to extract information about the grasp objects. The Shadow hand is equipped with several types of sensors allowing to measure different parameters.

B.1.1.1 Joint angle sensors

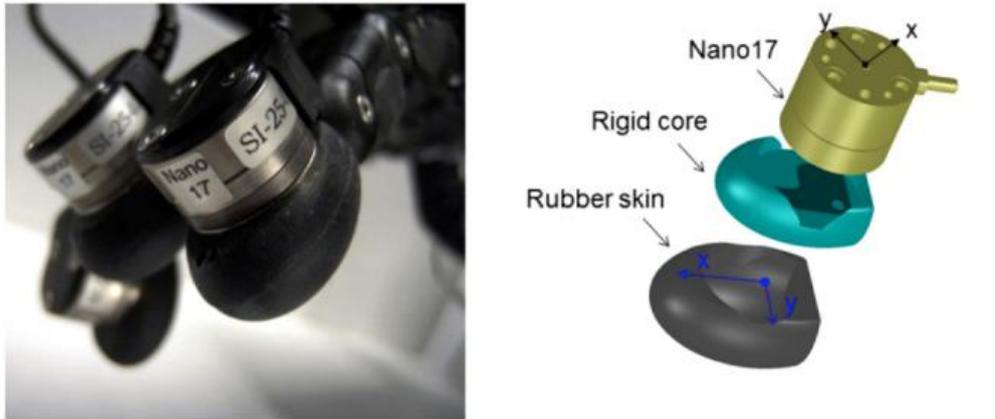
The Shadow Hand is equipped with absolute encoders based on magnetic principles: a hall sensor measures the intensity of magnetic field of a magnetic disk and sends the data in a raw form to the computer. The raw data is then transformed at the host to determine the position of the joints based on a calibration previously made. The resolution of these sensors is approximately 0.2° .

B.1.1.2 Joint force sensors

The transmission from the motors to move the fingers is made through tendons. A individual force sensor measures the force on each pair of sensors and it measures the difference of tension between the two sensors and not the absolute tension.

B.1.1.3 Tactile sensors

The Shadow Hand is also equipped with 6-axis force/torque ATI nano 17 sensors [110]. These sensors are located in the fingertips of the hand (Fig. B.2a). These sensors have been covered with an ellipsoidal solid core covered with deformable rubber skin (Fig. B.2b). Liu et al. [111] developed an algorithm to estimate contact location, the direction and the magnitude of the friction and normal forces and the local torque generated at the surface of the rubber skin based on the information coming from the ATI nano 17. This algorithm works for single point contact only and it showed an error mean squared error inferior to 0.55mm. Thus, the data coming from it is reliable for shape identification.



(A) Real ATI nano sensors

(B) Layers put onto the ATI nano sensor to make it look like a fingertip.

FIGURE B.2: ATI nano 17 sensors adapted to the shadow hand.

B.1.2 Control system

Grasping is important to get usable data for tactile shape identification. Thus, hand controllers are important to be known to plan a suitable grasping strategy. This hand can be controlled via ROS. Smart Motors are used to actuate the joints and they are driven by PWM. A PID controller is used and it can be set to control force control or position controller. More complex controllers can be implemented via Robotic Operating System (ROS). The joints of the hand are controllable to $\pm 1^\circ$.

B.2 Shadow Hand Simulation Model

For experiments in simulation, the model of the Shadow Hand on Gazebo [112] was used. This model has mostly the same characteristics of the real hand in terms of dimensions. However, it presents small differences. One of them regards the kinematics. The coupling between the middle and distal phalanges does not always fulfill the expected behavior. The distal joint angles may be larger than that of the middle phalanx.



FIGURE B.3: Gazebo model of the Shadow Hand.

This hand does not count on joint force sensors, therefore, force control cannot be made on this hand. The 6D force/torque sensors on the fingertip are simulated. For this, the contacts detected on the fingertip by the simulator are averaged in direction and magnitude. Thus, a single contact is given by this model as well. It is also possible to detect contacts on all links of the hand. In spite of having this information available, they are recorded as raw values.

Bibliography

- [1] Robert D Howe and Mark R Cutkosky. Dynamic tactile sensing: Perception of fine surface features with stress rate sensing. *IEEE transactions on robotics and automation*, 9(2):140–151, 1993.
- [2] Susan J Lederman and Roberta L Klatzky. Hand movements: A window into haptic object recognition. *Cognitive psychology*, 19(3):342–368, 1987.
- [3] Mark Moll and Michael A Erdmann. Dynamic shape reconstruction using tactile sensors. In *Robotics and Automation, 2002. Proceedings. ICRA '02. IEEE International Conference on*, volume 2, pages 1636–1641. IEEE, 2002.
- [4] C. Strub, F. Wörgötter, H. Ritter, and Y. Sandamirskaya. Correcting pose estimates during tactile exploration of object shape: a neuro-robotic study. In *4th International Conference on Development and Learning and on Epigenetic Robotics*, pages 26–33, Oct 2014.
- [5] Sylvia C. Pont, Astrid M. Kappers, and Jan J. Koenderink. Similar mechanisms underlie curvature comparison by static and dynamic touch. *Perception & Psychophysics*, 61(5):874–894, 1999. ISSN 1532-5962.
- [6] Bernard J. van der Horst and Astrid M. L. Kappers. Curvature discrimination in various finger conditions. *Experimental Brain Research*, 177(3):304–311, Mar 2007.
- [7] Ian E Gordon and Victoria Morison. The haptic perception of curvature. *Attention, Perception, & Psychophysics*, 31(5):446–450, 1982.
- [8] Denise YP Henriques and John F Soechting. Bias and sensitivity in the haptic perception of geometry. *Experimental Brain Research*, 150(1):95–108, 2003.
- [9] Knut Drewing and Marc O Ernst. Integration of force and position cues for shape perception through active touch. *Brain research*, 1078(1):92–100, 2006.

- [10] Boris Babenko, Ming-Hsuan Yang, and Serge Belongie. Robust object tracking with online multiple instance learning. *IEEE transactions on pattern analysis and machine intelligence*, 33(8):1619–1632, 2011.
- [11] Roberta L Klatzky, Susan J Lederman, and Victoria A Metzger. Identifying objects by touch: An “expert system”. *Perception & psychophysics*, 37(4):299–302, 1985.
- [12] P. Dougherty. “chapter 2: Somatosensory systems” from neuroscience online, 1997. URL <http://neuroscience.uth.tmc.edu/s2/chapter02.html>.
- [13] Catherine L Reed, Richard J Caselli, and Martha J Farah. Tactile agnosia: Underlying impairment and implications for normal tactile object recognition. *Brain*, 119(3):875–888, 1996.
- [14] Zhanat Kappassov, Juan-Antonio Corrales, and Véronique Perdereau. Tactile sensing in dexterous robot hands. *Robotics and Autonomous Systems*, 74:195–220, 2015.
- [15] A. Bierbaum, K. Welke, D. Burger, T. Asfour, and R. Dillmann. Haptic exploration for 3d shape reconstruction using five-finger hands. In *2007 7th IEEE-RAS International Conference on Humanoid Robots*, pages 616–621, Nov 2007.
- [16] O. Kerpa, K. Weiss, and H. Worn. Development of a flexible tactile sensor system for a humanoid robot. In *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No.03CH37453)*, volume 1, pages 1–6 vol.1, Oct 2003.
- [17] Wataru Fukui, Futoshi Kobayashi, Fumio Kojima, Hiroyuki Nakamoto, Nobuaki Imamura, Tadashi Maeda, and Hidenori Shirasawa. High-speed tactile sensing for array-type tactile sensor and object manipulation based on tactile information. *Journal of Robotics*, 2011, 2011.
- [18] W. Robotics. Tactile sensors, . URL <https://www.weiss-robotics.com/en/start-page/>.
- [19] D. Johnston, Ping Zhang, J. Hollerbach, and S. Jacobsen. A full tactile sensing suite for dextrous robot hands and use in contact force control. *Robotics and Automation, 1996. Proceedings., 1996 IEEE International Conference on*, 4:3222–3227, 1996.
- [20] Alexander Schmitz, Perla Maiolino, Marco Maggiali, Lorenzo Natale, Giorgio Cannata, and Giorgio Metta. Methods and technologies for the implementation of large-scale robot tactile sensors. *IEEE Transactions on Robotics*, 27(3):389–400, 2011.

- [21] J. M. Romano, K. Hsiao, G. Niemeyer, S. Chitta, and K. J. Kuchenbecker. Human-inspired robotic grasp control with tactile sensing. *IEEE Transactions on Robotics*, 27(6):1067–1079, Dec 2011.
- [22] K. Motoo, T. Fukuda, F. Arai, and T. Matsuno. Piezoelectric vibration-type tactile sensor with wide measurement range using elasticity and viscosity change. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1946–1951, Oct 2006.
- [23] Sadao Omata and Yoshikazu Terunuma. New tactile sensor like the human hand and its applications. *Sensors and Actuators A: Physical*, 35(1):9 – 15, 1992.
- [24] Dirk Goger, Nicolas Gorges, and Heinz Worn. Tactile sensing for an anthropomorphic robotic hand: Hardware and signal processing. In *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, pages 895–901. IEEE, 2009.
- [25] M. K. Johnson and E. H. Adelson. Retrographic sensing for the measurement of surface texture and shape. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1070–1077, 2009.
- [26] S. Begej. Planar and finger-shaped optical tactile sensors for robotic applications. *Robotics and Automation, IEEE Journal of*, 4(5):472–484, 1988.
- [27] Peter Kampmann and Frank Kirchner. Integration of fiber-optic sensor arrays into a multi-modal tactile sensor processing system for robotic end-effectors. *Sensors*, 14(4):6854–6876, 2014.
- [28] SynTouch. The biotac. URL <http://www.syntouchllc.com/Products/BioTac/>.
- [29] Yaroslav Tenzer, Leif P Jentoft, and Robert D Howe. Inexpensive and easily customized tactile array sensors using mems barometers chips. *IEEE Robot. Autom. Mag*, 21(3):89–95, 2014.
- [30] Zhanat Kappassov, Juan-Antonio Corrales Ramon, and Véronique Perdereau. *ZMP Features for Touch Driven Robot Control via Tactile Servo*, pages 234–243. Springer International Publishing, Cham, 2017.
- [31] N. Chen, H. Zhang, and R. Rink. Edge tracking using tactile servo. In *Proceedings 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems. Human Robot Interaction and Cooperative Robots*, volume 2, pages 84–89 vol.2, Aug 1995.
- [32] H. Zhang and N. N. Chen. Control of contact via tactile sensing. *IEEE Transactions on Robotics and Automation*, 16(5):482–495, Oct 2000.

- [33] R. Andrew Russell. Object recognition by a 'smart' tactile sensor. In *Proceedings of the Australian Conference on Robotics and Automation*, pages 93–8, 2000.
- [34] Nicolas Gorges, Peter Fritz, and Heinz Wörn. Haptic object exploration using attention cubes. *KI 2010: Advances in Artificial Intelligence*, pages 349–357, 2010.
- [35] P. K. Allen and P. Michelman. Acquisition and interpretation of 3-d sensor data from touch. *IEEE Transactions on Robotics and Automation*, 6(4):397–404, Aug 1990.
- [36] Z. Pezzementi, E. Plaku, C. Reyda, and G. D. Hager. Tactile-object recognition from appearance information. *IEEE Transactions on Robotics*, 27(3):473–487, June 2011.
- [37] Alexander Schneider, Jürgen Sturm, Cyrill Stachniss, Marco Reisert, Hans Burkhardt, and Wolfram Burgard. Object identification with tactile sensors using bag-of-features. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pages 243–248. IEEE, 2009.
- [38] Nicolas Gorges, Stefan Escalda Navarro, Dirk Göger, and Heinz Wörn. Haptic object recognition using passive joints and haptic key features. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 2349–2355. IEEE, 2010.
- [39] Zachary Pezzementi, Caitlin Reyda, and Gregory D Hager. Object mapping, recognition, and localization from tactile geometry. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 5942–5948. IEEE, 2011.
- [40] K. Miyashita, T. Takahashi, and M. Yamanaka. Features of a magnetic rotary encoder. *IEEE Transactions on Magnetics*, 23(5):2182–2184, Sep 1987.
- [41] Italien Institute of Technologie. icub robot. URL <http://www.icub.org/index.php>.
- [42] Shadow Robot Company. Dexterous shadow hand. URL <http://www.shadowrobot.com/products/dexterous-hand/>.
- [43] Steve Jacobsen, E Iversen, D Knutti, R Johnson, and K Biggers. Design of the utah/mit dextrous hand. In *Robotics and Automation. Proceedings. 1986 IEEE International Conference on*, volume 3, pages 1520–1532. IEEE, 1986.
- [44] A Schulz, Christian Pylatiuk, Artem Kargov, Reinhold Oberle, and Georg Bretthauer. Progress in the development of anthropomorphic fluidic hands and their applications. In *Mechatronics & Robotics*, pages 936–941, 2004.

-
- [45] Owonik Robotics. Allegro hand, . URL <http://www.simlab.co.kr/Allegro-Hand.htm#Features>.
- [46] Shunk. Servo-electric 3-finger-gripping hand. URL <http://www.schunk-modular-robotics.com/en/home/products/servo-electric-3-finger-gripping-hand-sdh.html>.
- [47] Chris S Lovchik, HA Aldridge, and Myron A Driftler. Design of the nasa robonaut hand. 1999.
- [48] K. Kaneko, K. Harada, and F. Kanehiro. Development of multi-fingered hand for life-size humanoid robots. In *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pages 913–920, April 2007.
- [49] Prensilia. S.R.L. Ih2 azzurra hand. <http://www.prensilia.com>, 2016.
- [50] Hugh HS Liu and Grantham KH Pang. Accelerometer for mobile robot positioning. *IEEE Transactions on Industry Applications*, 37(3):812–819, 2001.
- [51] Willow Garage. Pr2 robot. URL <http://www.willowgarage.com/pages/pr2/overview>.
- [52] R. Ibrayev and Y. b. Jia. Surface recognition by registering data curves from touch. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 55–60, Oct 2006.
- [53] P. K. Allen and K. S. Roberts. Haptic object recognition using a multi-fingered dextrous hand. In *Proceedings, 1989 International Conference on Robotics and Automation*, pages 342–347 vol.1, May 1989.
- [54] S. Caselli, C. Magnanini, and F. Zanichelli. Haptic object recognition with a dextrous hand based on volumetric shape representations. In *Multisensor Fusion and Integration for Intelligent Systems, 1994. IEEE International Conference on MFI '94.*, pages 280–287, Oct 1994.
- [55] A. M. Okamura and M. R. Cutkosky. Haptic exploration of fine surface features. In *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No.99CH36288C)*, volume 4, pages 2930–2936 vol.4, 1999.
- [56] Martin Meier, Matthias Schopfer, Robert Haschke, and Helge Ritter. A probabilistic approach to tactile shape reconstruction. *IEEE Transactions on Robotics*, 27(3):630–635, 2011.
- [57] Claudius Strub, Florentin Wörgötter, Helge Ritter, and Yulia Sandamirskaya. Correcting pose estimates during tactile exploration of object shape: a neuro-robotic

- study. In *Development and Learning and Epigenetic Robotics (ICDL-Epirob), 2014 Joint IEEE International Conferences on*, pages 26–33. IEEE, 2014.
- [58] S. Luo, W. Mou, K. Althoefer, and H. Liu. Iterative closest labeled point for tactile object shape recognition. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3137–3142, Oct 2016.
- [59] Lorenzo Natale and Eduardo Torres-Jara. A sensitive approach to grasping. In *Proceedings of the sixth international workshop on epigenetic robotics*, pages 87–94, 2006.
- [60] Alexander Bierbaum, Kai Welke, D Burger, Tamim Asfour, and Rüdiger Dillmann. Haptic exploration for 3d shape reconstruction using five-finger hands. In *Humanoid Robots, 2007 7th IEEE-RAS International Conference on*, pages 616–621. IEEE, 2007.
- [61] Jivko Sinapov, Taylor Bergquist, Connor Schenck, Ugonna Ohiri, Shane Griffith, and Alexander Stoytchev. Interactive object recognition using proprioceptive and auditory feedback. *The International Journal of Robotics Research*, 30(10):1250–1262, 2011.
- [62] Magnus Johnsson and Christian Balkenius. Experiments with proprioception in a self-organizing system for haptic perception. *Towards Autonomous Robotic Systems*, pages 239–245, 2007.
- [63] Magnus Johnsson and Christian Balkenius. Lucs haptic hand ii. *LUCS Minor, Lund University Cognitive Studies-Technical Reports*, 9, 2006.
- [64] S. Ratnasingam and T. M. McGinnity. Object recognition based on tactile form perception. In *2011 IEEE Workshop on Robotic Intelligence In Informationally Structured Space*, pages 26–31, April 2011.
- [65] Allison M Okamura and Mark R Cutkosky. Feature detection for haptic exploration with robotic fingers. *The International Journal of Robotics Research*, 20(12):925–938, 2001.
- [66] Taylor Bergquist, Connor Schenck, Ugonna Ohiri, Jivko Sinapov, Shane Griffith, and Alexander Stoytchev. Interactive object recognition using proprioceptive feedback. In *Proceedings of the 2009 IROS Workshop: Semantic Perception for Robot Manipulation, St. Louis, MO*, 2009.
- [67] Diego R Faria, Jorge Lobo, and Jorge Dias. Identifying objects from hand configurations during in-hand exploration. In *Multisensor Fusion and Integration for Intelligent Systems (MFI), 2012 IEEE Conference on*, pages 132–137. IEEE, 2012.

- [68] Uwe Proske and Simon C Gandevia. The proprioceptive senses: their roles in signaling body shape, body position and movement, and muscle force. *Physiological reviews*, 92(4):1651–1697, 2012.
- [69] Susan J Lederman and Roberta L Klatzky. Haptic perception: A tutorial. *Attention, Perception, & Psychophysics*, 71(7):1439–1459, 2009.
- [70] Julien Voisin, Yves Lamarre, and C Elaine Chapman. Haptic discrimination of object shape in humans: contribution of cutaneous and proprioceptive inputs. *Experimental Brain Research*, 145(2):251–260, 2002.
- [71] John R Napier. The prehensile movements of the human hand. *Journal of bone and joint surgery*, 38(4):902–913, 1956.
- [72] Fiona N Newell, Marc O Ernst, Bosco S Tjan, and Heinrich H Bülthoff. Viewpoint dependence in visual and haptic object recognition. *Psychological science*, 12(1):37–42, 2001.
- [73] Myrthe A Plaisier, Wouter M Bergmann Tiest, and Astrid ML Kappers. Salient features in 3-d haptic shape perception. *Attention, Perception, & Psychophysics*, 71(2):421–430, 2009.
- [74] Richard Bellman and Robert Kalaba. On adaptive control processes. *IRE Transactions on Automatic Control*, 4(2):1–9, 1959.
- [75] Marc M Van Hulle. Self-organizing maps. In *Handbook of Natural Computing*, pages 585–622. Springer, 2012.
- [76] Teuvo Kohonen. The self-organizing map. *Neurocomputing*, 21(1):1–6, 1998.
- [77] The Mathworks [®]. Self organizing maps. URL <https://fr.mathworks.com/help/nnet/ref/selforgmap.html>.
- [78] Corinna Cortes and Vladimir Vapnik. Support vector machine. *Machine learning*, 20(3):273–297, 1995.
- [79] Simon Tong and Daphne Koller. Support vector machine active learning with applications to text classification. *Journal of machine learning research*, 2(Nov):45–66, 2001.
- [80] OpenCV. Support vector machines. URL http://docs.opencv.org/2.4/doc/tutorials/ml/introduction_to_svm/introduction_to_svm.html.
- [81] Lochlan E Magee and John M Kennedy. Exploring pictures tactually. *Nature*, 283(5744):287–288, 1980.

- [82] Sheldon Cashdan. Visual and haptic form discrimination under conditions of successive stimulation. *Journal of Experimental Psychology*, 76(2p1):215, 1968.
- [83] PE Bryant and I Raz. Visual and tactual perception of shape by young children. *Developmental Psychology*, 11(4):525, 1975.
- [84] Alan D Berger and Pradeep K Khosla. Edge detection for tactile sensing. In *Proc. SPIE's Cambridge Symposium on Optical and Optoelectronic Engineering: Advances in Intelligent Robotics Systems*, 1988.
- [85] Tri Cong Phung, Yong Seok Ihn, Ja Choon Koo, and Hyouk Ryeol Choi. An enhanced edge tracking method using a low resolution tactile sensor. *International Journal of Control, Automation and Systems*, 8(2):462–467, 2010.
- [86] Ning Chen, Hong Zhang, and R Rink. Edge tracking using tactile servo. In *Intelligent Robots and Systems 95. 'Human Robot Interaction and Cooperative Robots', Proceedings. 1995 IEEE/RSJ International Conference on*, volume 2, pages 84–89. IEEE, 1995.
- [87] Kitti Suwanratchatamane, Ryo Saegusa, Mitsuharu Matsumoto, and Shuji Hashimoto. A simple tactile sensor system for robot manipulator and object edge shape recognition. In *Industrial Electronics Society, 2007. IECON 2007. 33rd Annual Conference of the IEEE*, pages 245–250. IEEE, 2007.
- [88] D I McCloskey. Kinesthetic sensibility. *Physiological Reviews*, 58(4):763–820, 1978. URL <http://physrev.physiology.org/content/58/4/763>.
- [89] Barry C Stillman. Making sense of proprioception: the meaning of proprioception, kinaesthesia and related terms. *Physiotherapy*, 88(11):667–676, 2002.
- [90] A. Schmitz, Y. Bansho, K. Noda, H. Iwata, T. Ogata, and S. Sugano. Tactile object recognition using deep learning and dropout. In *2014 IEEE-RAS International Conference on Humanoid Robots*, pages 1044–1050, Nov 2014.
- [91] Maarten WA Wijntjes, Akihiro Sato, Vincent Hayward, and Astrid ML Kappers. Local surface orientation dominates haptic curvature discrimination. *IEEE transactions on haptics*, 2(2):94–102, 2009.
- [92] AW Goodwin, KT John, and AH Marceglia. Tactile discrimination of curvature by humans using only cutaneous information from the fingerpads. *Experimental brain research*, 86(3):663–672, 1991.
- [93] Sylvia C Pont, Astrid ML Kappers, and Jan J Koenderink. Haptic curvature discrimination at several regions of the hand. *Perception & Psychophysics*, 59(8):1225–1240, 1997.

- [94] W. Grimson and T. Lozano-Perez. Model-based recognition and localization from tactile data. In *Proceedings. 1984 IEEE International Conference on Robotics and Automation*, volume 1, pages 248–255, Mar 1984.
- [95] M. Jin, H. Gu, S. Fan, Y. Zhang, and H. Liu. Object shape recognition approach for sparse point clouds from tactile exploration. In *2013 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 558–562, Dec 2013.
- [96] A. Bierbaum, I. Gubarev, and R. Dillmann. Robust shape recovery for sparse contact location and normal data from haptic exploration. In *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3200–3205, Sept 2008.
- [97] Tewndy-One. Tewndyone hand. URL http://www.twendyone.com/tech3_e.html.
- [98] Hanifa Dostmohamed and Vincent Hayward. Trajectory of contact region on the fingerpad gives the illusion of haptic shape. *Experimental Brain Research*, 164(3):387–394, 2005.
- [99] S. Rakhlin. *Online Methods in Machine Learning: Theory and Applications*. MIT. URL <http://www.mit.edu/~rakhlin/6.883/#notes>.
- [100] David A Ross, Jongwoo Lim, Rwei-Sung Lin, and Ming-Hsuan Yang. Incremental learning for robust visual tracking. *International Journal of Computer Vision*, 77(1):125–141, 2008.
- [101] A Levey and Michael Lindenbaum. Sequential karhunen-loeve basis extraction and its application to images. *IEEE Transactions on Image processing*, 9(8):1371–1374, 2000.
- [102] Oded Maron and Tomás Lozano-Pérez. A framework for multiple-instance learning. In *Advances in neural information processing systems*, pages 570–576, 1998.
- [103] Nan-Ying Liang, Guang-Bin Huang, Paramasivan Saratchandran, and Narasimhan Sundararajan. A fast and accurate online sequential learning algorithm for feedforward networks. *IEEE Transactions on neural networks*, 17(6):1411–1423, 2006.
- [104] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul):2121–2159, 2011.
- [105] Jyrki Kivinen, Alexander J Smola, and Robert C Williamson. Online learning with kernels. *IEEE transactions on signal processing*, 52(8):2165–2176, 2004.

- [106] Kuang-Chih Lee and David Kriegman. Online learning of probabilistic appearance manifolds for video-based recognition and tracking. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 852–859. IEEE, 2005.
- [107] Li Fei-Fei, Rob Fergus, and Pietro Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. *Computer vision and Image understanding*, 106(1):59–70, 2007.
- [108] Qing Wang, Feng Chen, Jimei Yang, Wenli Xu, and Ming-Hsuan Yang. Transferring visual prior for online object tracking. *IEEE Transactions on Image Processing*, 21(7):3296–3305, 2012.
- [109] Peter Kotschieder, Madalina Fiterau, Antonio Criminisi, and Samuel Rota Buló. Deep neural decision forests. In *International Conference on Computer Vision*, pages 1467–1475, 2015.
- [110] ATI Industrial Automation. F/t sensor: Nano17. URL http://www.ati-ia.com/products/ft/ft_models.aspx?id=Nano17.
- [111] Hongbin Liu, Kien Cuong Nguyen, Véronique Perdereau, Joao Bimbo, Junghwan Back, Matthew Godden, Lakmal D Seneviratne, and Kaspar Althoefer. Finger contact sensing and the application in dexterous hand manipulation. *Autonomous Robots*, 39(1):25–41, 2015.
- [112] Gazebo. Gazebo simulator. URL <http://www.gazebosim.org>.