

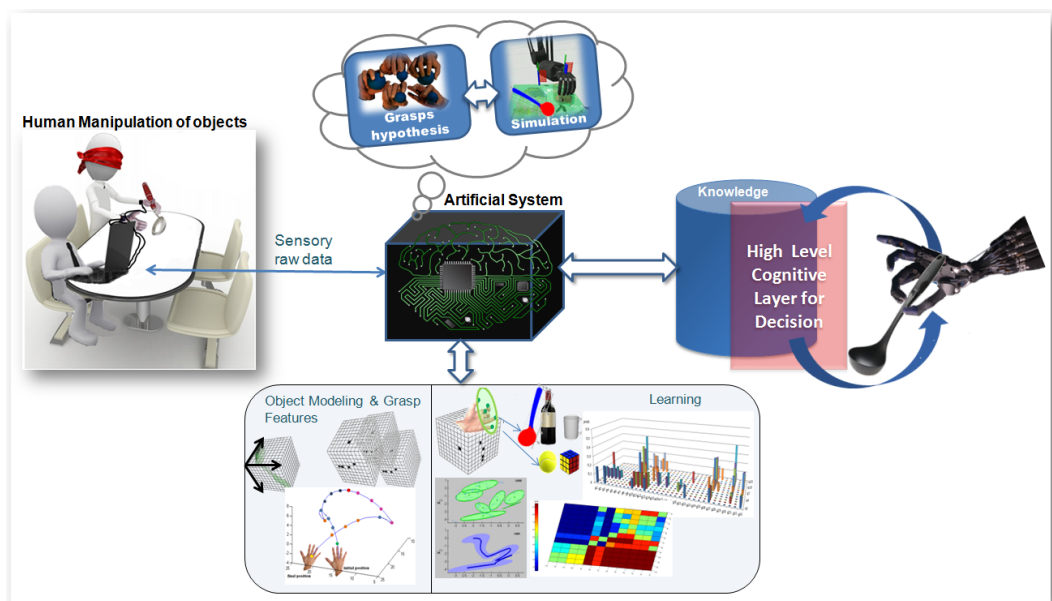


University of Coimbra

Faculty of Science and Technology

Department of Electrical and Computer Engineering

## Probabilistic Learning of Human Manipulation of Objects towards Autonomous Robotic Grasping



Ph.D. Thesis

*Diego Resende Faria*

Coimbra, August 2013



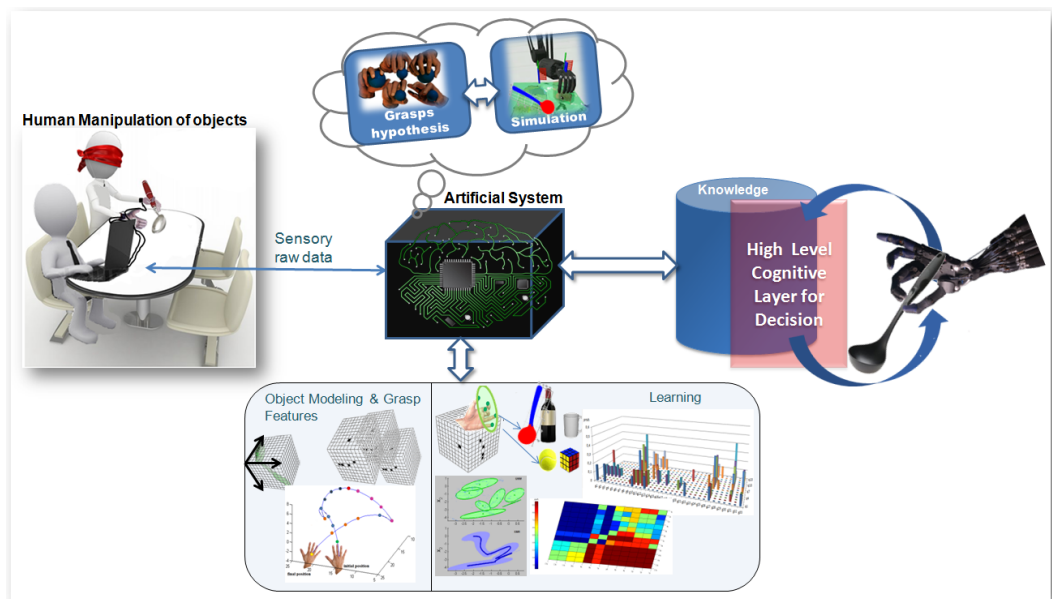


University of Coimbra

Faculty of Science and Technology

Department of Electrical and Computer Engineering

## Probabilistic Learning of Human Manipulation of Objects towards Autonomous Robotic Grasping



Thesis submitted:

to the Electrical and Computer Engineering Department of the Faculty of Science and Technology of the University of Coimbra in partial fulfillment of the requirements for the Degree of Doctor of Philosophy.





Research work developed under Supervision of

**Doctor Jorge Manuel Miranda Dias**

Associated Professor of the

Faculty of Science and Technology, University of Coimbra

and

**Doctor Jorge Nuno de Almeida e Sousa Almada Lobo**

Assistant Professor of the

Faculty of Science and Technology, University of Coimbra



*This work is dedicated for those that I love and are part of my life*



# Abstract

In this thesis we study how humans manipulate everyday objects, and construct a probabilistic representation model for the tasks and objects useful for autonomous grasping and manipulation by robotic hands. An object-centric probabilistic volumetric model is proposed to represent the object shape acquired by in-hand exploration. The object volumetric map is also useful to fuse the multimodal data and map contact regions, and tactile forces during stable grasps. This model is refined by segmenting the volume into components approximated by superquadrics modeling, and overlaying the contact points used taking into account the task context. A novel approach for object identification by human in-hand exploration of objects is proposed. Different contact points are associated to an object shape, modeled by mixture models, allowing the object identification through the set of hand configurations used during the in-hand exploration.

Humans excel when dealing with everyday manipulation tasks, being able to learn new skills, and to adapt in different complex environments. This results from a lifelong learning, and also observation of other skilled humans. To obtain similar dexterity with robotic hands, cognitive capacity is needed to deal with uncertainty. By extracting relevant multi-sensor information from the environment (objects), knowledge from previous grasping tasks can be generalized to be applied within different contexts. Based on this strategy, human demonstrations of manipulation tasks are recorded from both the human hand and object points of view. The multimodal data acquisition system records human hand and fingers 6D pose, finger flexure, tactile forces distributed on the inside of the hand, color images and stereo depth map, and also object 6D pose. From the acquired data, relevant features are detected concerning motion patterns, tactile forces and hand-object states. This will enable modeling a class of tasks from sets of repeated demonstrations of the same task, so that a generalized probabilistic representation is derived to be used for task planning in artificial systems.

In this research we also address an artificial system that relies on knowledge from previous human object grasping demonstrations to accomplish the objective of robot grasp synthesis for un-

known objects. A learning process is adopted to quantify probabilistic distributions and the uncertainty. These distributions are combined with preliminary knowledge towards inference of proper grasps given a point cloud of an unknown object. We designed a method that comprises a twofold process: object decomposition and grasp synthesis. The decomposition of objects into primitives is used, across which similarities between past observations and new unknown objects can be made. The grasps are associated with the defined object primitives, so that feasible object regions for grasping can be determined. The hand pose relative to the object is computed for the pre-grasp and the selected grasp.

The results presented in this thesis show that the in-hand exploration of object is useful to model and represent the object shape allowing its identification by the hand configurations during the exploration. The features extracted from human grasp demonstrations are sufficient to distinguish key patterns that characterize each stage of the manipulation tasks, ranging from simple object displacement, where the same grasp is employed during manipulation (homogeneous manipulation) to more complex interactions such as object reorientation, fine positioning, and sequential in-hand rotation (dexterous manipulation). We have validated our approach of grasp synthesis on a real robotic platform (a dexterous robotic hand). Results show that the segmentation of the object into primitives allows identifying the most suitable regions for grasping based on previous learning. The proposed approach provides suitable grasps, better than more time consuming analytical and geometrical approaches. Learning from human grasp demonstrations along with features extracted from objects is a useful way to endow a robotic dexterous hand with enough skills to autonomously grasp and manipulate novel objects.

# Resumo

Nesta tese estuda-se a forma como os seres humanos manipulam objectos do quotidiano, modelando-os de forma probabilística. Os modelos probabilísticos são também utilizados para a representação das tarefas e dos movimentos, com o objectivo de dotar mãos robóticas com elevado grau de autonomia. Um modelo volumétrico probabilístico centrado no objecto é proposto para representar a sua forma a partir de exploração táctil. O mapa volumétrico do objecto é também útil para a fusão de dados multimodais, mapas da região de contacto e força táctil durante uma pega estável. Este modelo é refinado através da segmentação do volume em componentes, aproximadas por superquádricas, e sobrepondo os pontos de contactos tendo em conta o contexto da tarefa. Uma nova técnica para identificação de objectos através de exploração táctil de objectos é proposta. Diferentes pontos de contacto são associados a forma de um objecto através de modelos de misturas, permitindo a identificação de um objecto através de um conjunto de configurações base que a mão toma durante a exploração.

Os seres humanos destacam-se quando se trata de tarefas de manipulação do quotidiano, sendo capaz de aprender novas habilidades ou adaptar-se a novos e complexos ambientes. Esta capacidade resulta de uma aprendizagem pessoal, mas também da observação de pessoas experientes. Para que mãos robóticas adquiram essa destreza, é necessária uma capacidade cognitiva, que as permitam lidar com as incertezas. Através da extracção de informação multisensorial do ambiente, o conhecimento de manipulações anteriores pode ser generalizado para a aplicação em diferentes contextos. Baseado nesta estratégia, as demonstrações de tarefas de manipulação são capturadas, tanto do ponto de vista do objecto como o da mão. O sistema de aquisição multimodal captura poses da mão e de cada um dos dedos, as medidas de flexão dos dedos, a força táctil distribuída na mão, imagens RGB, imagens de profundidade, e também a pose do objecto. Estes dados permitem modelar classes de tarefas a partir de conjuntos de demonstrações repetidas da mesma tarefa, para que um modelo probabilístico generalizado possa ser derivado e usado no planeamento de tarefas por parte de sistemas artificiais.

Nesta tese também se aborda a síntese da configuração geométrica da mão em tarefas de

manipulação, baseados no conhecimento adquirido através das demonstrações humanas. Um processo de aprendizagem é desenvolvido para quantificar as distribuições probabilísticas e a incerteza. Estas distribuições são combinadas com conhecimento prévio com o objectivo de se inferir a forma correcta de agarrar o objecto, dada uma nuvem de pontos de um objecto desconhecido. O método proposto é dividido em dois processos: decomposição do objecto e a síntese da configuração geométrica da mão para agarrar um objecto. A decomposição do objecto em primitivas permite que se usem semelhanças de observações passadas para novos objectos desconhecidos. As acções são associadas com as várias primitivas de um objecto, para que se determinem as regiões do objecto nas quais os objectos possam ser agarrados. A pose da mão relativa ao objecto é calculada para a fase de aproximação e contacto com o alvo.

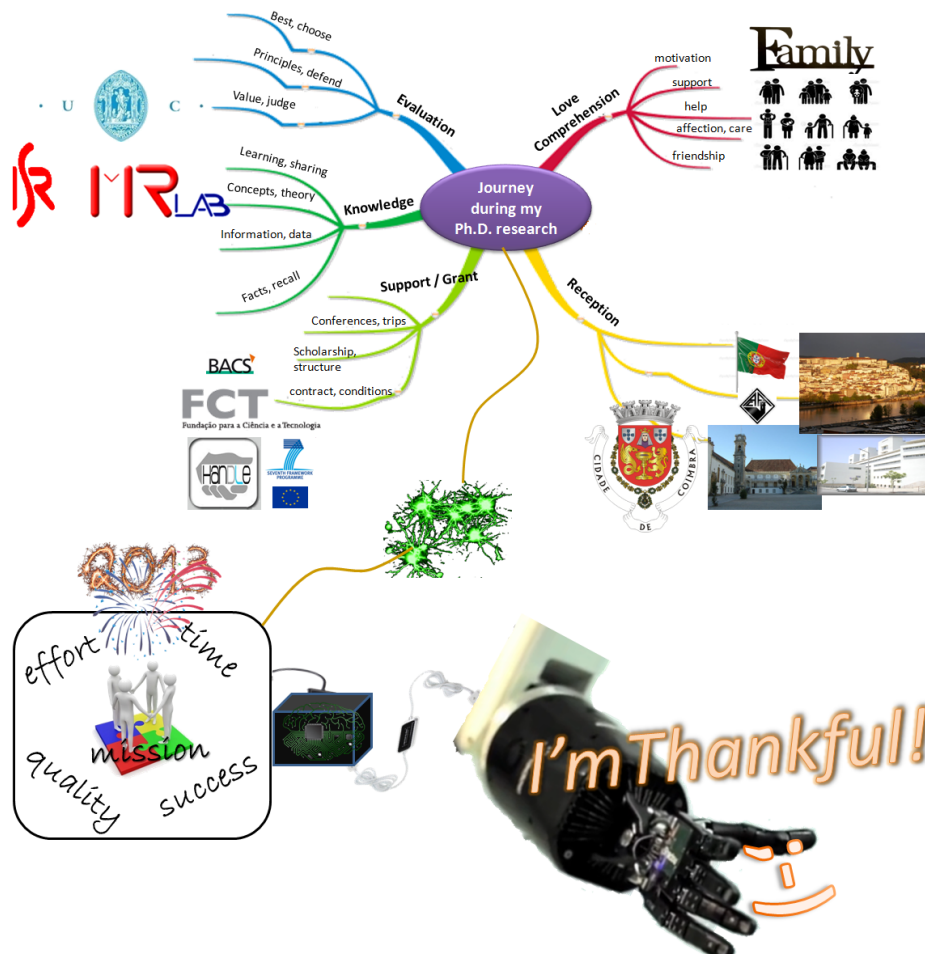
Os resultados apresentados nesta tese mostram que a exploração táctil de um objecto é útil para modelar e representar a forma do objecto, permitindo a sua identificação através da configuração geométrica da mão durante a exploração. As características extraídas a partir das demonstrações são suficientes para distinguir padrões chave que caracterizam cada etapa da tarefa de manipulação, desde a simples movimentação do objecto, onde a mesma pega é aplicada durante a manipulação (manipulação homogénea), até interações mais complexas, tais como reorientação do objeto, refinamento da pose ou rotação sequencial.

A nossa técnica de síntese de configurações da mão para agarrar um objecto foi validada numa plataforma robótica real. Os resultados demonstram que a segmentação de objectos em primitivas, permite identificar as regiões mais adequadas para os agarrar, tendo uma melhor performance que as aproximações analíticas. A aprendizagem através de demonstrações humanas, bem como as características extraídas de objectos, mostram vantagens na geração de muita autonomia de mão robótica permitindo a manipulação de novos objectos.



# Acknowledgment

I am grateful by the opportunity of coming to Portugal to start and develop my Ph.D. research. I had the pleasure of acquiring a valuable knowledge, facing new challenges, always full of “spirit of work” and cooperation to keep the quality and excellence of the work. I appreciate so much the advices of my advisor and co-advisor (Prof. J. Dias and Prof. J. Lobo), and also the love and support of all my family (specially Fernanda and my parents) including also the close relatives. I want to thank all team of the Mobile Robotics Lab and my colleagues at ISR-UC during this journey for the valuable discussion about science. I want to thank the HANDLE project consortium, because making part of this team was a pleasure to me. Finally, a big thanks for all jury members for the evaluation of my Ph.D. research, and you also, reader of this work, Enjoy it!





# Contents

<b>Abstract</b>	<b>i</b>
<b>Resumo</b>	<b>iii</b>
<b>Acknowledgment</b>	<b>v</b>
<b>List of Abbreviations</b>	<b>xxiii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 General Motivation . . . . .	4
1.2 Thesis Contributions . . . . .	5
1.3 Context of Current State of the Art . . . . .	7
1.3.1 Grasping from the Neuroscience point of View . . . . .	8
1.3.2 Grasping in Robotics . . . . .	10
1.4 Organization of the Thesis . . . . .	13
<b>2 Object Shape Representation</b>	<b>15</b>
2.1 Introduction . . . . .	15
2.2 Probabilistic Representation of Objects by In-Hand Exploration . . . . .	16
2.2.1 Single Hand Exploration of Static Objects . . . . .	18
2.2.2 In-Hand Exploration of Non-static Objects . . . . .	19
2.2.3 Probabilistic Volumetric Map Cells Updating . . . . .	21
2.2.4 Frame of Reference for Object-centric Representation . . . . .	26
2.2.5 Experimental Results: Object Shape by In-Hand Exploration . . . . .	26
2.3 Multimodality and Fusion . . . . .	30
2.3.1 Visual Cues to Complement the Object Model . . . . .	31
2.3.2 Bayesian Mixture Models . . . . .	35
2.3.3 Experimental Results . . . . .	37
2.4 Discussion . . . . .	38
<b>3 Segmentation and Modelling of Object Components</b>	<b>41</b>
3.1 Introduction . . . . .	41
3.2 Object Segmentation . . . . .	42
3.2.1 Mixture Distribution-based Segmentation . . . . .	44
3.2.2 Segmentation based on Major Axis Analysis . . . . .	48
3.3 Object Components: Primitives Detection . . . . .	52
3.3.1 Components Modelling using Basic Primitives: A Probabilistic Approach . . . . .	52
3.3.2 Object Component Modelling using Superquadrics . . . . .	57
3.4 Discussion . . . . .	59

<b>4</b>	<b>Identifying Objects from Hand Configurations</b>	<b>63</b>
4.1	Introduction	63
4.2	Learning Hand Configurations for Everyday Objects	65
4.2.1	Human Demonstrations	66
4.2.2	Mixtures of Contact Points Models and Signatures Extraction	67
4.2.3	Similarity Measure for the Contact Points	69
4.3	Object Identification	70
4.4	Experimental Results	72
4.5	Discussion	73
<b>5</b>	<b>Grasp Features from Human Demonstrations</b>	<b>77</b>
5.1	Introduction	77
5.2	Manipulation Task Database	78
5.2.1	Experimental Setup and Data acquisition	78
5.2.2	Data Storage	80
5.3	Grasp Detection from Contact Points Overlaid on the Object Model	82
5.4	Grasping Movements Recognition by Learning from Human Demonstration	85
5.4.1	3D Trajectories Segmentation: Curvatures and Hand Orientation Detection	86
5.4.2	Learning and Estimation for Trajectories Classification applying Bayesian Techniques	92
5.4.3	Experimental Results for 3D Trajectories	96
5.4.4	Simplifying for 2D Case: Hand Trajectories Segmentation and Classification	100
5.5	Manipulation Tasks Identification by Learning and Generalizing Hand Motions	104
5.5.1	Segmenting and Identifying Manipulation Stages	105
5.5.2	Motion Pattern: Finding Similarities	106
5.5.3	Trajectory Generalization for Task Representation	107
5.5.4	Tasks Identification	109
5.5.5	Experimental Results	111
5.6	Object-centric Framework for Manipulation Knowledge	112
5.7	Discussion	115
<b>6</b>	<b>Grasp Synthesis based on Human Grasp Demonstrations</b>	<b>119</b>
6.1	Introduction	119
6.2	Learning from Human Grasp Demonstrations	121
6.2.1	Overview of Bayesian Inference using the Learned Data	122
6.2.2	Experimental Setup for Data Acquisition	124
6.2.3	Learning Object Graspable Regions: Assigning Weights to Shape Primitives	124
6.2.4	Learning Suitable Objects Graspable Regions in Task-oriented Grasps	127
6.2.5	Learning Grasping Choice from Human Observations	132
6.2.6	Learning from Object Observations	136
6.2.7	Storing Learned Data	137
6.3	Grasp Synthesis	137
6.3.1	Using Decomposition Module in the Grasp Synthesis	138
6.3.2	Grasp Synthesis Module	139
6.4	Experimental Results	143
6.4.1	Simulated Tests	143
6.4.2	Tests in the Robotic Platform	145
6.5	Discussion	147
<b>7</b>	<b>Overall Conclusions and Future Work</b>	<b>149</b>

<b>A List of Publications</b>	<b>153</b>
<b>B Grasp List used in this Study</b>	<b>155</b>
<b>C SCamPol Toolbox for Matlab</b>	<b>157</b>
<b>Bibliography</b>	<b>163</b>



# List of Figures

1.1	Overview of how a general integration of subtopics of Manipulation/Grasping are done. In-hand exploration of objects and manipulation can be coupled during the exploration of unknown or partially known object to accomplish some manipulation tasks where the grasping or in-hand actions can be applied: [OC01] [FMLD12a] [FMLD12b] [FMLD10]. . . . .	4
2.1	Experimental setup area and the workspace for mapping (grid $35cm^3$ equally divided where each voxel is sized with $5cm^3$ ). . . . .	19
2.2	Polhemus Liberty Motion Tracking System [Pol]: Magnetic tracker sensors attached to the hand (fingertips and back of the hand). . . . .	20
2.3	Example of in-hand exploration where is needed to compute the transformation at each movement of the object to register all 3D points in the same frame of reference. The points belonging to the object surface are represented by the map in the workspace. . . . .	21
2.4	Examples of the probabilistic volumetric map. Left image: real object; middle image: partial volume of the object; left image: map of the full object model and contact points overlaid on the object surface (red voxels representing the contact points and blue voxel representing the centroid of the object to define its frame of reference). . . . .	23

2.5	BN for the occupancy grid model for object representation using in-hand exploration. Left image shows the labels: prior, posterior and respective distributions, yet not necessary in DBN representations. The variables are defined in terms of their notation and conditional dependence. The instantiation is defined with their parameters and the random variables that support the model are fully described (i.e. their significance and measurable space). Right image shows that contribution of the sensor on each finger through time made explicit using the Bayesian network formalism with plate notation applied to in-hand exploration of objects. . . . .	25
2.6	Raw data of 3D object models derived from in-hand exploration. Left to right: mug, sponge, bottle and wooden cat. . . . .	26
2.7	Computed probabilistic volumetric map of the wooden cat. Left image show all occupied cells of the object map. Right image shows the occupied cells using (2.6) with threshold = 0.6; and left image using threshold = 0.8. . . . .	27
2.8	Object representation using the probabilistic volumetric map: the first image is the object, followed by the probabilistic volumetric map where the darker cells are those ones with higher probability than the lighter ones. It shows the most explored region of the object. The last image is the map showing clear cells just those ones occupied (probability higher than the specified threshold 0.7). We can see the global shape derived from the in-hand exploration. . . . .	27
2.9	Object representation using the probabilistic volumetric map: sponge and its computed map. . . . .	28
2.10	Object representation using the probabilistic volumetric map: bottle and its computed map. . . . .	28
2.11	Object shape representation by in-hand exploration of a spray bottle. The first image (left to right) is the raw data (point cloud), next three images are different views of the voxels representation of the object shape, and the last image is the occupancy representation of the cells, the darkest ones represent the lower probabilities (less explored regions). . . . .	29



2.12	Example of registration process and mapping for moving objects. The first image shows the raw data of non-static object derived from the in-hand exploration of a wooden cat; middle image shows the point cloud after registration to a common frame of reference and then the last image is the computed probabilistic map in which the cells threshold $> 0.8$ . . . . .	29
2.13	on-the-fly object modelling derived from in-hand exploration and the rendering in Blender software. . . . .	30
2.14	Calibration strategy: Using a white tape on the sensor facilitates later to find the marker in the image to compute the 3D point given the left and right images corresponding to the 3D point of the sensor in its frame of reference. . . . .	32
2.15	Reprojection of 3D points of $\{P\}$ in the image plane $\{C\}$ . Left image shows how is collected the left and right images from the stereo camera as well as the 3D points from the magnetic tracker. The yellow dot represents the 3D point from $\{P\}$ to $\{C\}$ . Right image shows the reprojection from $\{P\}$ to $\{C\}$ after the in-hand exploration of the bottle. . . . .	33
2.16	Evolution of the rotation and translation matrices estimates according to the number of points used in the calibration process. . . . .	34
2.17	Probabilistic representation of the object global map (wooden cat) derived from in-hand exploration and vision. . . . .	38
2.18	Probabilistic representation of the object global map (bottle) derived from in-hand exploration and vision. . . . .	38
3.1	Everyday objects (wii-mote, mug, sponge, bottle, ladle, Nintendo nunchuck and spray bottle) segmentation using GMM clustering. These objects were acquired by different sensor modalities to test the segmentation. Top row: laser scanner; bottom row and last image (at right): in-hand exploration (bottle, sponge and spray bottle); RGB-D device (ladle and mug). . . . .	47
3.2	Examples of segmentation with little success of everyday objects using GMM clustering. Some segments cannot be considered as a good candidate region for grasping (based on a qualitative analysis). Some of the segmented regions are not suitable for subsequent approximation by a geometrical primitive. . . . .	47

3.3	Object Segmentation Definition. The object is segmented into three parts: top, middle and bottom. The segmentation takes into consideration the major axis (pc1: principal component), i.e., the axis with bigger length. . . . .	48
3.4	Results for the daily objects segmentation based on major axis. On left side is depicted the models acquired by in-hand exploration and on right side the models acquired by laser scanner. . . . .	50
3.5	Difference in the segmentation of a mug acquired by different sensors. Left image presents a mug acquired by in-hand exploration segmented into 3 components. Right image presents a mug acquired by laser scanner segmented into 2 components. . . . .	51
3.6	Learned histogram: sphere primitive. . . . .	53
3.7	Learned histogram: cylinder primitive. . . . .	54
3.8	Learned histogram: plane primitive. . . . .	54
3.9	Shape retrieval: left image shows the 3D points representing the object (bottle); middle image the GMM applied as segmentation; and the last image shows the recovered primitives for each of the object component. . . . .	56
3.10	Typical Superquadrics: $a_1 = a_2 = a_3 = 1$ , except ellipsoid. . . . .	57
3.11	Superquadrics models obtained for some segmented everyday objects. . . . .	59
4.1	Sensors used in our experimental setup: Polhemus Liberty Magnetic Tracking System and Tekscan Tactile sensor. . . . .	67
4.2	GMM and GMR process. Each cluster encloses demonstrations of hand configurations. A signature of the transition between the hand configurations is generated using the features in the latent space when is applied GMR. . . . .	69
4.3	Hypotheses Generation: at each demonstrated contact points during the in-hand exploration a hypotheses list of candidate objects identities are generated from the stored objects. . . . .	71
4.4	Sequences of contact points overlaid on the object surface during the demonstrations of hand design while the in-hand exploration was performed by an individual. For each sequence is identified the hand configuration given the contact points. . . . .	72
4.5	Most common hand configurations identified for the mug during the in-hand exploration. The grasps presented are following the taxonomy shown in the Human Grasping Database developed inside the GRASP project [GRA]. . . . .	74

4.6	Matching between object models using ICP method. The best matching between the new observation and the object stored in the database after the selection of candidates is the object in the red box (mug), RMSE = 0.0044. The best matching between the the laser scanner models was the object in the blue box (mug), RMSE = 5.5247. . . .	75
5.1	Global overview of the experimental area, data acquisition devices and objects available.	79
5.2	General representation of the data acquisition architecture implemented in the experimental area. . . . .	80
5.3	Schematic representation of the organization of folders (filled boxes) and files created during a data acquisition session; several session folders and device folders can be created. . . . .	81
5.4	Content of the data.xml file for the Polhemus Liberty device for tracker sensor 1. . .	82
5.5	Some grasps that were identified for the spray. . . . .	84
5.6	Top image: Example of a 3D trajectory of <i>pick-up and place</i> and possible curvatures along the trajectory; Bottom image: Example of hand orientation along the trajectory.	87
5.7	Top image: Illustration of trajectories normalization. Bottom image: Smoothed trajectory - Blue color represents the raw data; Red color represents the smoothed trajectory. . . . .	88
5.8	Top row: raw data representing trajectories of top-grasps actions during the reaching movement. Bottom row: Side-grasps. . . . .	92
5.9	Hand trajectory after the pre-processing step (smoothing and normalization): Top-grasp action. . . . .	92
5.10	Grasping learning tables: Mean histogram for top and side grasping actions - curvatures and hand orientation features. Each feature has a probability assigned to it at each segment (1 to 8). . . . .	94
5.11	Side-grasp trajectory (after smoothing and rescale). . . . .	97
5.12	Classification of a movement of top-grasp during the reaching step. Three different model of classification: 1. Bayesian model relying on curvatures, 2. Bayesian model relying on hand orientation, 3. Bayesian mixture model relying on the weights given by entropy for the two previous models. . . . .	99
5.13	Comparison graphic. Classification using two different types of features and the third one using weights reached by entropy. . . . .	99

5.14	Reach-to-grasp trajectories (raw data: inches measure). Left image: Top-Grasping; Right image: Side-Grasping. . . . .	102
5.15	Learning tables: Left image represents the Top-Grasping Learning Table $P(C GD)$ . The probabilities of the curvature down vary of 0.14 to 0.35. The probabilities of line vary of 0.16 to 0.57. The probabilities of curvature up vary of 0.19 to 0.66. The right image represents the Side-Grasping Learning Table $P(C GD)$ . The probabilities of curvature down vary of 0.16 to 0.4. The probabilities of line vary of 0.3 to 0.6. The probabilities of curvature up vary of 0.2 to 0.5. The sum of the the features down, line and up in each trajectory part must be 1. . . . .	103
5.16	Example of action phases in a simple homogeneous manipulation task, where the same grasp is employed during the manipulation. . . . .	106
5.17	Motion Patterns: Similarities detection in the action phases of the trajectories in a dataset of a manipulation task. . . . .	107
5.18	Regression applied on sub-regions of an action phase of a manipulation task. 2D view: left and middle images: x, y view; right image: x, z. Examples of quadratic and cubic order regression. . . . .	108
5.19	Left: Raw data(in inches): trajectories dataset (object displacement); Right: Trajectory segmentation by action phase. . . . .	109
5.20	Left: Extracted features (Cartesian positions); Middle: Relevant features (similarities among all trajectories); Right: Generalized trajectory by interpolation of the points as a function of arc length along a space curve (adopting parametric splines). . . . .	109
5.21	<i>Pick-up and lift</i> task representation at trajectory level: Interpolation using parametric splines using the Cartesian 3D coordinates of the selected features between the trajectories of the dataset of the same tasks. . . . .	110
5.22	Task features at trajectory level. . . . .	110
5.23	Task trajectory to be identified: <i>pick-up and place</i> . . . . .	112
5.24	Task classification results after 10 trials. Comparison of classification using multi-modal features and single feature for the same trials. . . . .	112
5.25	Hand trajectory (sensor attached to the back of the hand) during the in-hand manipulation task (pick-up the mug, rotate and release it). . . . .	113

5.26	Left image: Object trajectory during the in-hand manipulation task (in green colour); The blue circle shows the in-hand manipulation phase (object rotation along the trajectory); Right image: Graphs representing some transitions of hand shapes during the in-hand manipulation task. The nodes represent the locations of the fingertips and wrist. . . . .	114
5.27	Sequence of some contact points overlaid in the static representation of the object (volumetric map) during the manipulation task. The contact points are given by the fingertips locations during the in-hand manipulation phase. . . . .	114
5.28	Eye tracker. Left image: Subject performing a manipulation using the eye tracker; Right image: Typical output of the eye tracker. Red cross indicates the estimated gaze direction. . . . .	115
5.29	Human gaze during grasping and the contact points on the object surface. Task: Reaching and Grasping by the Object Handle. The visual gaze during the grasping shows that the human usually looks to the region of the object where will be performed the grasp. . . . .	116
5.30	Human gaze during grasping and the contact points on the object surface. Task: Reaching and Grasping the mug by side-grasp. This type of grasping was chosen due to the orientation of the object - it influences the type of grasping. . . . .	116
6.1	Overview of the grasp generator modules. . . . .	120
6.2	Overview of the learning process to assist the inference to search for candidate grasps for a given object model. . . . .	122
6.3	Bayesian network to represent a general model when a grasp type is estimated given some causes $Q, \mathcal{T}$ . . . . .	123
6.4	Sensors used in our experimental setup: Motion Tracking System, Tactile sensor and RGB-D sensor. . . . .	124
6.5	Examples of the statistical data acquired during the human grasp demonstrations. The statistics assist to weight the geometrical primitives as preference for grasping when dealing with a specific pair of quadrics. . . . .	127

6.6	Probability Distribution in the Learned Table: Pointing the preference given a pair of quadrics $q_i$ - $q_i$ , $i = \{1, \dots, n\}$ . The axes $\{x, y\}$ represents all possible pairs of quadrics $q_i$ that can compose an object. The probability assigned for the pair of quadrics is show through the color-map. . . . .	128
6.7	Examples of contact points of stable grasps from human demonstration on objects surfaces. The objects are: sponge, mug, wooden cat and bottle. . . . .	129
6.8	Examples of contact points of stable grasps from human demonstration on the object (spray bottle) surface. . . . .	129
6.9	Statistics computed from the observations. Three different tasks performed many times by different individuals. By analysing the probability distribution of the chosen primitives to perform the grasp, the object graspable part (given the task context) can be estimated. . . . .	131
6.10	Identification of object graspable component for the sponge, <i>wii-mote</i> , spray-bottle, mug and bottle. For these trials we have used only two components for each object. Each component has a probability of being graspable, the maximum a posteriori estimate indicates the graspable component in each context. . . . .	132
6.11	Grasp choice given the object quadrics: statistical data acquired by human demonstrations. The demonstrations were chosen from a grasp list [GRA] with 33 grasps types for 13 possible quadrics models $Q = \{box, cube, cuboid, cylinder, ellipsoid, sphere, octahedron, rounded box, rounded spinning-top, squared spinning-top, star spinning-top, variation1-sphere(spherical arch), variation2-sphere(butterfly shape)\}$ . . . . .	134
6.12	Probability Distribution: learned table from the statistics presented in Figure 6.11. Each grasp type has a probability of occurring given a quadric model. . . . .	135
6.13	Spray Bottle - Probability Distribution of $q_i$ being considered as an object component $c_i$ . . . . .	136
6.14	Frame of references adopted to generate the grasps pose relative to the object pose. Figure adapted from HANDLE Project Wiki page for the definitions of the ROS modules for the final demonstration of the project [HANb]. . . . .	140
6.15	Decompose view in a simulator. The images show all steps for the segmentation. Left image represents the raw object data from the Kinect sensor after removing the table-top. The right image shows the segmentation result of the unknown object achieved by our decompose module. . . . .	144

6.16	Results using the object point clouds to test our modules. The application returned some grasp associated with the geometrical shapes (quadrics $q_i$ ) of the object. The marked quadrics in red are the object parts with higher probability to be the graspable part, and the grasps associated with this specific part have a higher weight. The order of appearance of the grasp types indicates the most probable grasp for that component.	144
6.17	Selected grasp (grasp 27 from [GRA]: Quadpod) for the object (box) executed in a robotic platform.	145
6.18	Selected grasp executed in a simulator before the execution in the robotic platform.	146
6.19	Tests in the simulator to verify the valid grasps of the generated list.	146
B.1	Grasps list defined by the European GRASP project. More details and the complete taxonomy is available at [GRA].	155
C.1	Representation of an experimental setup with stereo camera and Polhemus rigidly mounted, and also the relevant frame of references.	158
C.2	Calibration strategy: using a white tape on the tracker sensor to acquire both 3D points, in the frame of reference of the tracker sensor, and also from the stereo camera by localizing the white mark in the images.	158
C.3	Graphic User interface: Main menu of the SCamPol toolbox.	158
C.4	Example of the images acquisition. The left images from the stereo camera are in the top row and the right images are in the bottom row. The yellow circles show the sensor location in the images.	159
C.5	Window View of the images acquired for calibration after being loaded.	160
C.6	Selection window: selection of the sensor region in the image to find its coordinate in the images (left and right) to allow the computation of the 3D point.	161
C.7	Graphic generated by the toolbox with the number of points used in the calibration and the value of re-projection errors in the scale of pixels.	161
C.8	Re-projection example: 3D points acquired from the tracker sensor in the image plane.	162





# List of Tables

2.1	Reprojection error values in pixels (average error and standard deviation) according to the number of points. . . . .	33
5.1	Hand Orientation extracted along the trajectory: Result of our application for the trajectory shown in Figure 5.9. The second column is the amount of features found in each segment; the third column is the corresponding probability of each feature. . . .	93
5.2	Curvatures extracted along the trajectory: Result of our application for the trajectory shown in 5.9. The second column is the amount of features found in each segment; the third column is the corresponding probability of each feature. . . . .	93
5.3	Classification using Curvatures ( <i>C</i> ) and Hand Orientation ( <i>O</i> ) for the trajectory shown in Figure 5.11. The trajectory was classified as side grasp with 98.32% using curvatures and 92% using hand orientation. The estimation for top or side grasps in each part of the trajectory is shown. . . . .	97
5.4	Result of 10 trials of Side-grasp trajectory. Two false Negative (less than 50%) on trial 3 and 5 using curvatures. The trials 4, 6 and 10 in hand orientation were considered as side-grasp, but with low probability, less than the threshold of 70%. Just one false negative (trial 3) was detected. Entropy was used to combine both features into a single classification model. The trials 5 and 10 were considered side-grasp with low probability. . . . .	100
5.5	Classification of Circle (2 first rows) and bye-bye (2 last rows) movements. . . . .	100
5.6	Trajectory Segmentation: Result for trajectory shown in Figure 5.14 (left image:top-grasp). . . . .	102
5.7	Trajectory Segmentation: Result for the trajectory shown in Figure 5.14, side grasp. .	102

5.8	Classification Result: Estimation of trajectory shown in Figure 5.14 (left image: top-grasp). At each trajectory part is shown the probability of the trajectory to be Top- or Side-Grasp. This trajectory was classified with probability of 87.12% as Top-Grasp. .	103
5.9	Classification Result: Estimation of trajectory shown in Figure 5.14 (right image: side-grasp). At each trajectory part is shown the probability of the trajectory to be Top- or Side-Grasp. This trajectory was classified with percentage of 83.70% as Side-Grasp. . . . .	104
5.10	Classification Result: 10 trials of Top-Grasping performed by 2 subjects. Blue color: probabilities > 70%; Red Colour: probabilities < 50%. In the trial 7 the trajectory was classified as Top-Grasp but with lower probability. . . . .	104
5.11	Classification Result: 10 trials of Side-Grasp performed by 2 subjects. Blue color: probabilities > 70%; Red Colour: probabilities < 50%. The trajectories in the trials 3, 5, 7 and 9 were classified as Side-Grasp with lower probabilities. . . . .	105
5.12	Classification Result . . . . .	112

# List of Abbreviations

2D, 3D, 6D - Two, Three, Six Dimensional

BMM - Bayesian Mixture Models

BVM - Bayesian Volumetric Map

CPS - Contact Points Space

DBN - Dynamic Bayesian Network

DoF - Degrees of Freedom

EM - Expectation Maximization

GMM - Gaussian Mixture Models

GMR - Gaussian Mixture Regression

GPLVMs - Gaussian Process Latent Variable Models

HMM - Hidden Markov Models

ICP - Iterative Closest Point

ISR-UC - Institute of Systems and Robotics-University of Coimbra

MGA - Maximum Grip Aperture

MMSE - Minimum Mean Square Error

NTP - Network Time Protocol

OOP - Object-Oriented Programming

PCA - Principal Component Analysis

PCoA - Principal Coordinate Analysis

RANSAC - Random Sample Consensus

RBC - Recognition by Components

ROS - Robot Operating Systems

SVD - Singular Value Decomposition

XML - eXtensible Markup Language



# Chapter 1

## Introduction



Robotics is increasingly moving towards the research and development of technologies that allow the introduction of robots in our daily life. The optimal robot assistant should share a human environment and be able to cope with human presence and interact in a very friendly way. To create such applications a number of problems need to be solved, including transposing the movements used in everyday tasks, as well as finding out how to interpret human interactions and how to use all this knowledge to create robots that can successfully act as assistants.

Intelligent robots are becoming part of our everyday lives and examples where they are usually employed are: caretakers for the elderly and for disabled people, assistants in surgery and patient rehabilitation and educational toys. The expectation of having intelligent robots lead us to think that in order for this to happen, the complexity of programming must be greatly reduced, and robot autonomy must become much more natural. This challenge is particularly relevant to a new generation of robots, which must interact with people, and operate in human environments.

In the human world, we are surrounded by many types of objects subject to manipulation by human hands. To ensure the robots have enough skills to interact with our environment, and in order

to make sure they handle objects proficiently, their hands must resemble our hands. An example of an advanced robotic hand that performs all 24 movements of the human hand is the Shadow Dexterous Hand produced by the Shadow Robot Company LTD [Sha]. Although this hand satisfies many degrees of freedom, the objects forms are infinite and the robotic hand will always encounter unknown objects. An autonomous robot hand will need to adapt to varying grasp tasks accurately in different situations. To overcome such a challenge, artificial cognitive skills are needed to enable a robotic platform to take decisions for the execution of each specific task, and also to adapt to a human environment.

Given that humans excel in manipulative tasks, which is a basic skill for our survival, and a key feature in our world of artefacts and devices made by human hands, then this study focuses on researching how humans manipulate everyday objects, and construct a probabilistic representation model for the tasks and objects useful to autonomous grasping and manipulation by robotic hands.

In this thesis, human demonstrations of predefined object manipulation tasks are recorded from both the human hand and from the object's point of view. The multimodal data acquisition system records hand and fingers 6D pose, finger flexure, tactile forces distributed on the inside of the hand, color images and stereo depth map, and also object 6D pose. From the acquired data, relevant features are detected concerning motion patterns, tactile forces and hand-object states. This will enable the modelling of a class of tasks from sets of repeated demonstrations of the same task, so that a generalized probabilistic representation is derived to be used for task planning in artificial systems. An object-centric probabilistic volumetric model is proposed to fuse the multimodal data and to map contact regions, gaze, and tactile forces during stable grasps. This model is refined by segmenting the volume into components approximated by geometrical primitives, and overlaying the contact points used taking into account the task context.

In this study, it is shown that the features extracted are sufficient to distinguish key patterns that characterize each stage of the manipulation tasks, ranging from simple object displacement, where the same grasp is employed during manipulation (homogeneous manipulation), to more complex interactions such as object reorientation, fine positioning, and sequential in-hand rotation (dexterous manipulation).

A system architecture for grasp synthesis is also presented in this thesis. The proposed approach relies on knowledge from previous human grasping of predefined objects. A learning process is adopted to quantify the probability distributions and the uncertainty over the human grasp experiences. These distributions are combined with preliminary knowledge of grasping choice for specific

---

object shapes towards inference of proper grasping given an object point cloud coming from the sensor observations. To accomplish our goal of generating grasp hypothesis given an unknown object, our system is designed in a twofold process: object decomposition and grasp synthesis. This way, after the object decomposition we can find suitable regions for grasping, as well as the candidate grasps for this object. The answers given by the system are validated by an artificial robotic dexterous hand given an unknown object.

The main tasks involved in this thesis regarding dexterous manipulation are:

- In-hand exploration of objects for shape representation and object identification;
- Modelling of contact points and hand configurations of stable grasps during human demonstrations;
- Planning grasp strategy (finding a suitable region on objects for grasping);
- Grasp synthesis for unknown objects;
- Learning and identification of hand trajectories and constraints during simple manipulation tasks.

Through tasks oriented grasps, it is possible to learn how to grasp specific objects. It is not a simple process to model a task, and it differs from one object to another. Empirical approaches were introduced to the grasp synthesis problem to avoid analytical techniques computational complexity. Empirical methods use learning algorithms to imitate human grasping strategies. Since commonly used objects have different shapes and sizes, generalizing these techniques to novel objects is not trivial. In this study, a generalization process to deal with unknown objects is performed by encompassing human demonstrations and object perception from previous familiar objects.

To acquire knowledge on the previous mentioned topics, an artificial perception system is built and it can be used later to allow an artificial hand to accomplish different tasks, for instance, unknown object exploration and identification, as well as grasping planning and synthesis.

In summary, different subtopics of grasping will be presented in this thesis. The objectives of this research are threefold stages: the first is the in-hand exploration of objects to achieve the object model (Chapter 2) that is segmented and represented by some geometrical primitives (Chapter 3). This approach allows the object identification (Chapter 4), as well as to extract other object properties. The second stage of this thesis is the extraction of grasp features by observing how humans perform

a task (Chapter 5). We encompass the object model with human demonstrations (e.g., contact points) for learning purposes. The object model is used to learn object graspable regions and also to associate grasp types for a specific object. The methods and computational techniques presented in Chapters 2, 3 and 5 are used to assist the third stage of this thesis, a grasp synthesis system that is presented in Chapter 6, where some proposed methods were integrated into a single framework, accomplishing then the goal of generating candidate grasps for unknown objects to be executed in a robotic platform.

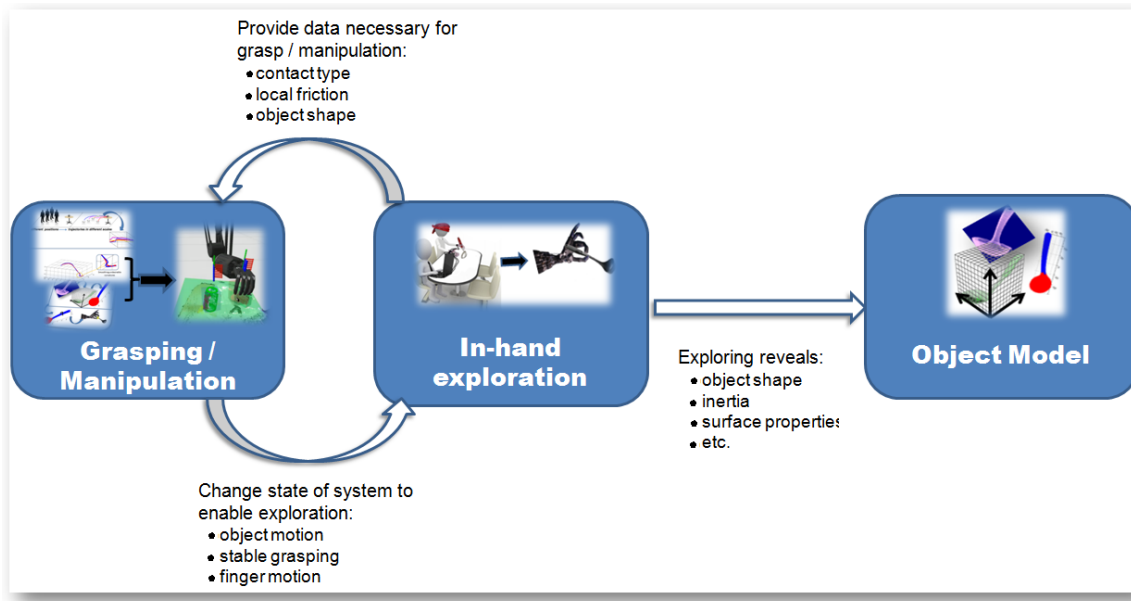


Figure 1.1: Overview of how a general integration of subtopics of Manipulation/Grasping are done. In-hand exploration of objects and manipulation can be coupled during the exploration of unknown or partially known object to accomplish some manipulation tasks where the grasping or in-hand actions can be applied: [OC01] [FMLD12a] [FMLD12b] [FMLD10].

However, the integration and relation of the objectives mentioned (all topics of this thesis) are done during different manipulation tasks. From a computational perspective, the integration of different modules, such as object perception and recognition, hand trajectories, task generalization and grasp synthesis into a common framework, provide a way to endow a robot with cognitive skills to interact with objects in complex tasks. This integration can be done as drawn in Figure 1.1 as referred by [OC01].

## 1.1 General Motivation

Humans are able to learn skills, adapt and interact in different complex environments using their rich source of sensory perception. Despite humans having different sensory perception, there is still



a world of sensory uncertainty. Perception is, in principle, multisensory; information from several modalities can be combined in case a single modality is not enough to reach a robust estimate [EB04]. When several modalities complement each other, the resulting sensory cooperation naturally leads to a more robust and complete estimate of the surrounding environment.

To deal with the uncertainty acquired by the artificial perception system for everyday tasks, in this study we adopt a probabilistic reasoning to acquire knowledge from human manipulation of objects, extracting the relevant patterns as well as properties of the surrounding environment (objects). Learning and inference are the main topics acquired from human demonstrations and then this priori knowledge is used to make generalizations in other situations. From the demonstrations, a quantified uncertainty (distributions) is achieved using the preliminary knowledge of the reasoning and the data coming from the observation of the phenomenon. After building the knowledge, a cognitive inference can be achieved through Bayesian inference.

Different problems in the field of grasping and manipulation are still open, such as intelligent in-hand manipulation of different objects, as well as full autonomous grasping of unknown objects. Humans studies on haptics perception demonstrated that human hands are very skilled to characterize, perceive and recognize using only the sense of touch. Humans use specific hand exploratory movements to extract features and characterize the in-hand manipulated objects [KLM85], [LK87]. Newell *et al.* [NETB01] showed that both visual and haptic object recognition is dependent on the orientation of the object relative to the observer, and that they thus complement each other and cooperate. The best view for recognising an object visually is the learned view (usually the front). The best view for object recognition by the haptic modality, however, is the side the fingers naturally explore the most, the back part of the object. This humans studies show us that by learning from human experiences we can extract relevant data towards improving robotic grasping.

## 1.2 Thesis Contributions

The following scientific question is addressed in this thesis: *"How can we endow an artificial system with appropriate cognitive skills (i.e., advanced perception capabilities) in order to grasp and manipulate everyday objects in the most autonomous and natural way possible?"*

The answer reached in this thesis involves understanding how humans manipulate everyday objects, and constructing a probabilistic representation model for the tasks (at a trajectory level) and objects. Therefore, in order to deal with any uncertainty from the sensors and surrounding envi-

ronment, these models are useful for decision-making in order to achieve successful grasping and manipulation. This research is focused on developing an artificial system with the adequate intelligence to allow the recovery of shape representation, identification of objects as well as reasoning on actions (movements) performed by humans hands. Relevant cues are extracted from human demonstration in order to replicate grasping of a variety of everyday objects and also skilled movements of reaching, grasping and object in-hand exploration. This is the key point in this research if it is to reach the mentioned goals by specific learning and inference on grasps and in-hand movements. To achieve the mentioned contributions, multimodal data feedback for grasping and in-hand exploration purposes are used. Moreover, the relevant features extracted are also used to build a database of learned data and the knowledge of uncertainty in the observations are processed towards making successful inferences.

To accomplish the objective of this thesis, the research falls into different grasping subtopics where the contributions are addressed as follows:

- A probabilistic representation method for object models (Chapter 2) and tasks (Chapter 5) that are important for autonomous robot grasping. A novel strategy for probabilistic representation of objects by in-hand exploration based on human demonstrations that supports multimodality and fusion is presented. The object volumetric map allows the representation of the partial volume of the object, as well as contact points modelling of stable grasps. The probabilistic representation model of tasks relies on the learning from 6D hand pose and trajectories for task identification taking into account the motion patterns.
- A novel approach for object identification by in-hand exploration based on learning from previous hand configurations associated with object models to generate hypotheses of object identities to be matched to an object database (Chapter 4);
- Strategies for object shape segmentation for grasping synthesis: Using data learned from human demonstrations of stable grasps and object modelling to identify suitable regions on objects for grasping based also on statistics of previously observed grasps (Chapters 3 and 5).

During the Ph.D. studies, our research led us to a set of publications in peer reviewed international conferences and journals. A set of publications related to the chapter's topic is listed in the end of each chapter of the thesis, and a complete list of publications is also presented in the Appendix

A. The Chapters 2, 3 and 4 brought forth publications covering the topics related to in-hand exploration of objects for shape representation using a probabilistic volumetric map, followed of the object segmentation into components, and later the object identification using the hand geometrical configurations. Some results presented in Chapter 5 about grasp features detection and learning from human grasp demonstrations for movements identification, modelling of contact points of stable grasps, and grasp types detection were published. The Chapter 6 brought forth publications which addresses stages and also an artificial system endowed with cognitive skills to allow a robot with intelligent behaviour by acquiring knowledge from human grasp demonstrations to reason about how to behave in response to an unknown object presented to the system with the objective of generating suitable grasps for the robot execution.

### 1.3 Context of Current State of the Art

In this section, computational theories of grasping, considering research results in the literature of neuroscience and robotics involving analytical and empirical approaches will be given. This research will, in essence, provide information of different topics inside manipulation area. These numerous techniques encode object modelling and identification by in-hand exploration. Another type analyses grasp strategies based on learning by human demonstrations where grasp features are learned to build a knowledge representation of tasks and objects. A global overview of different approaches is given in this chapter to present the related works that this study have been based on, in order to develop our ideas and contribute in this field. The methods and research results referenced in this chapter are important for the reader's comprehension about grasping and manipulation activities, employing various approaches to solve different challenges in the area. However, in each chapter of this thesis, more specific and detailed works are also presented, i.e., researches that are related to our contribution are analysed and a specific critical discussion comparing our proposed method with others methods is presented.

An important factor that needs to be considered in the development of robotic grasping is ensuring stability during the object grasping. There are many approaches for robotic grasping that tries to solve this problem, but various constraints are met. One is finding suitable stable grasps where the task requirements are involved, making the process more complex. To model a robotic grasping, generally, a set of constraints has to be satisfied. Firstly the robotic hand configurations, its finger

capabilities, must be considered. Secondly the object geometric features must be taken into account. Finally, the constraints of the task requirements must be analysed.

Specifying a set of contacts points on the object surface (taking also into consideration constraints) is called grasp synthesis. Grasp synthesis can be achieved by analytical or empirical approaches. Analytical approaches choose the finger positions and the hand configuration with kinematical and dynamical formulations. Thus, they generally optimize an objective function such as the grasp stability or the task requirements. On the other hand, empirical (knowledge-based) approaches use a learning strategy to choose a grasp depending on the task and on the object geometry. Different algorithms have been developed in grasp planning for two-dimensional objects [Liu00], [PF95]. In addition, grasp synthesis for three-dimensional objects are also an active research area because of the complex geometry and high dimensionality of the grasp space.

Stability is a necessary, but not a sufficient condition for a grasping strategy. When we reach out to grasp an object, we have a goal in our mind or a task to accomplish. Thus, in order to successfully perform the task, the grasp should also be compatible with the task requirements. A grasping strategy for unknown objects should consider the task-oriented grasps and has to deal with the variety of object shapes and sizes. It has to guarantee stability, task compatibility and adaptability to grasp novel objects and to answer to the famous question: "where should an unknown object be grasped in order to accomplish the task?".

In the next subsections different approaches to grasping are described, based on the neuroscientists' view of human manipulation activities, and also the artificial perception systems in robotics that have been developed to achieve autonomy and dexterity in grasping tasks.

### 1.3.1 Grasping from the Neuroscience point of View

Humans are capable of reaching and grasping objects with great dexterity. Behavioural consequences from the anatomical variations of the human hand due to adaptation have been studied in the neuroscience field. Modern studies focus on the relationship between the brain function and the hand. The hand has many functions whereby we reach for objects, grasp and lift them or manipulate or use them to act on other things. Napier's work [Nap80] first described the terms precision and power grips. The biomechanical and neurophysical constraints were explained using his model, as well as movement variations such as force, posture, duration and speed. The intended activity decided what type of grip was necessary. The techniques in behavioral neuroscience, neuroimaging and electrophysiology help

to reveal where exactly in the brain these processes have started. The sensorimotor transformation relates to the older views.

Many researchers have described that grasping with reference to the grip aperture which happens to be the separation between the thumb and index finger and the size of the maximum grip aperture (MGA) during prehension is linearly related to the size of objects to be grasped [Jea84], [Jea88], [SE08] and [Cas05].

A neurological study with humans has shown that the visual perception of object size, shape and orientation depends on visual pathways in the cerebral cortex that are separate from those mediating these same object properties in the control of goal directed grasping [GMJC91], [GMB<sup>+</sup>94].

Castiello and Jeannerod [CJ91] reported in their work the importance of previous learned knowledge in humans and in monkeys for visually guided grasping. This indicates that learning from previous knowledge is relevant for grasping new objects. The study suggests that object features are coded differently for their recognition and for their grasping and the knowledge or learning is relevant to object grasping. Thus, a grasping strategy should be able, using a learning algorithm, to grasp objects without recognizing them. It is clear that if we are able to recognize objects, we will also be able to associate a grasp to each object category. Because of the variety of object shapes and sizes, predicting every possible object the robot could encounter is impossible. This way, a robot will certainly have to grasp non-identified objects and so do humans. In such situations, what objects features may yield to a good grasp?

Humans can use and adapt skills learned and used in the past to grasp new objects. To account for this capacity, a theory of object recognition was put forth by Irving Biederman [Bie87] which extended previous work of Marr and Nishihara [MN78]. According to the Recognition By Components theory of Biederman, humans are able to recognize objects by separating them into geometric primitives. Biederman suggests that segmenting objects for their identification does not depend on our familiarity with these objects. Thereby, we conduct the same process for any object, whether it is familiar or unfamiliar. The author concludes that even nonsense objects may be identified by decomposing them into parts. Then, going through the robotics field, if an unknown object is decomposed into geometrical primitives, each primitive emphasises a specific grasp? Taking into consideration that learned previous knowledge is important as mentioned before, an unknown object can fall into the category of familiar object or even associated by similarities allowing possible candidate grasps for this object.

### 1.3.2 Grasping in Robotics

We are addressing in this research a subtopic of human movements recognition for object manipulation. Typically, in the literature the techniques for movement analysis have two main approaches. The first group represents the movements at the trajectory level and generalises the representation of the movements through the extraction of statistical regularities from several human demonstrations of the movements. The second group of approaches proposes a symbolic learning and encoding of movements based on the supervised labelling and segmentation of the primitives during the learning stage.

An example for the first class of approaches is provided by Calinon *et al.* [CGB07]. In their work, the extraction of continuous constraints from a set of demonstrations using different initial configurations of the manipulated object is described. The Cartesian trajectories are projected using Principal Components Analysis (PCA). The spatio-temporal constraints are then represented through Gaussian Mixture Models (GMM). The approach has been successful on a robotic platform that reproduces a generalised version (obtained using Gaussian Mixture Regression) of a demonstrated task. Another example from the same class is shown by Ogawara *et al.* [OTKH09] where a method to detect repeated motion patterns in a long motion sequence is developed. The approach considers that repeated motion patterns are structured information that can be obtained without knowledge of the context of motions. The method was evaluated and compared to other previous works that detect repeated interactions between humans and objects in every-day manipulation tasks. The method has shown a greater performance in terms of detectability and computational time. Pastor *et al.* [PHAS09] proposed an approach to learn motor skills from human demonstrations modelled using a set of differential equations - dynamic movement primitive (DMP) framework, and developed a library of movements by labelling each recorded movement according to the task and the context.

The typical approach of symbolic representation methods is to initiate primitive sequence detection in the human demonstrations stream of data, followed by pattern recognition methods which provide the most probable temporal sequence of primitives. An example of this technique is used by Kondo *et al.* [KUU08] that propose a method to describe in-hand manipulation demonstration movements by recognising a sequence of contact state transitions between the human hand and the manipulated object. The recognition algorithm is based on a Dynamic Programming approach by comparing the similarity of the contact state transition between an input sequence and template manipulation primitives.

The work by Krugger *et al.* [KHB<sup>+</sup>10] presents the automatic extraction of action primitives and the corresponding grammar from continuous movements of several human demonstrations of grasping tasks. The approach considers that all the actions can be described by a set of elementary building blocks and there are a set of rules (grammar) that define how these action primitives can be combined. The action primitives are represented by parametric Hidden Markov Models. One of the key elements of those platforms is their ability to handle and explore objects as shown by Klatzky and Lederman [KL90], Biederman *et al.* [Bie87] and Sahbani *et al.* [SEK09]. In this research we also explore a multi-sensory approach to estimate the regions of the object that are going to be grasped by analysing the visual gazing performed by the subject during the preliminary moments to the grasp execution, as proposed by Flanagan *et al.* [FBJ06].

Another related work is presented by Bohg *et al.* [BK10a] that explore the grasping movements as a combination of a descriptor based on visual shape context with a non-linear classification algorithm that leads to a detection of stable grasping points for a variety of objects. When it is assumed that an object with a similar graspable part can be grasped in the same manner, such as a handbag or a mug (both are composed by curved parts like cylinders), it is possible to segment the objects in primitives for grasp planning considering some shapes of single parts. The constituting parts of an object shape influence the choice of an object's graspable part, independently of their orientations. The relative size of the object component is very important to select the graspable part [EKS10] and [EKSP07].

In our work, a probabilistic description is used for the representation of 3D objects, which is then segmented into few parts by approximating each object part using superquadrics primitives and also using human demonstration of object graspable parts as proposed by El-Khoury *et al.* [EKSP07], but in this research we overlay automatically the human demonstration of stable grasps on object's surface to learn them, which is different to the mentioned work that uses only human demonstration of grasping choice given some objects components. In our representation we associate data on object graspable parts such as contact points and tactile force obtained from demonstrations of in-hand manipulation with successful grasps.

### **Imitation Learning: systems based on human demonstrations**

Empirical grasping methods avoid the computational complexity of analytical techniques by attempting to mimic human grasping strategies. Empirical strategies for grasp planning can be divided into

systems based on the observation of the object to be grasped and systems based on the observation of a human performing the grasp. The former techniques generally learn to associate object's characteristics with a hand pre-shape, while in the latter, a robot observes a human operator performing a grasp and then tries to imitate the same grasp. This technique is called learning by demonstration approach.

Different Learning-by-Demonstration frameworks were proposed in the literature where the robot observes the human performing a task and afterwards it is able to perform the task itself. One of the problems arising in human based learning settings is the one of measuring human performance. Many researchers use data gloves for mapping of human hand to artificial hand workspace and learn the different joint angles [FdSH98], [EK04] hand preshapes [KWSN05] or the corresponding task wrench space [AC08] in order to perform a grasp.

Robot programming by demonstration for grasping purposes is an active field in robotics. In [SRHR04], the authors use situated multimodal interaction to teach a robot. The system combines visual attention and gestural instruction by means of an interface for speech recognition and linguistic interpretation to allow multimodal task-oriented instructions. Others use kinesthetic demonstrations where the human teacher demonstrates the task by moving the robot arms and the sensor signals are learned and generalised [CGB07] to other tasks. Stereo vision is often used to track the demonstrators hand performing a grasp [HBZ06] or try to recognize its hand shape from a database of grasp images [RKK08]. Mirror neurons modelling is another alternative when observing an action for imitation, and it was also introduced to the grasping problem [OA02]. The research developed by Montesano *et al.* [MLBSV08] relies on a probabilistic approach for learning and imitation. Their work addresses the learning of affordances encoding the relationship between actions, objects and effects by the interaction of a robot with the environment. The visual perception plays an important role in their work.

We are motivated to focus our research on observing human demonstrations to rely our system on human grasp experiences, since different human skills can be identified during the manipulation tasks, and consequently trying to approximate and replicate them in artificial systems. However, we are still observing and extracting some object properties to obtain relevant features to achieve feasible results on grasp synthesis, and also on grasp movements identification.



### Systems based on Object Observations

Grasping strategies based on object observation analyse its properties and learn to associate them with different grasps. Some approaches associate grasp parameters or hand shapes to object geometric features in order to find good grasps in terms of stability [PMAT04], [LP05]. Others techniques learn to identify grasping regions in an object image [BK10a], [SDKN07], [SPLS08].

The authors [MKCA03] model objects on a set of basic geometrical primitives and then they define rules to generate a set of grasp starting positions and pre-grasp shapes that can then be tested on the object model. By using hand preshapes, this method can limit the huge number of possible hand configurations for grasp planning. The planner requires a manually constructed primitive decomposition of the object, so that Goldfeder *et al.* [GALP07] removed the need for a manual decomposition and introduced a multi-level superquadrics representation.

An automatic localization of a variety of differently shaped objects given a point cloud (cluttered setting) from a laser range sensor is proposed by [BV07]. The authors adopt superquadrics for shape recovery. The detection is based on a hierarchical RANSAC search, and a fitting criteria is used for voting quality purposes. Criteria for object shape and the relationship of object parts are used to generate hypothesis and results show that the proposed method is adequate for object localization in cluttered environment.

Lopez-Damian *et al.* [LDSA05] propose an iterative segmentation algorithm for grasping non-convex objects. They compute first the inertial axes of the whole object and used them to generate grasps on it. When failing to obtain valid grasps, the object decomposition process starts. At each iteration of the decomposition step, two components are obtained and the authors try to generate feasible grasps on them. The process is repeated until a grasp is found or the decomposition terminates.

Since the grasping problem sometimes demands many degrees of freedom, the previously mentioned methods use object decomposition into parts to define a small search space that is likely to contain many grasps. They do not attempt to find the grasp based on what humans choose when grasping an object, and that is consequently adapted to the task requirements.

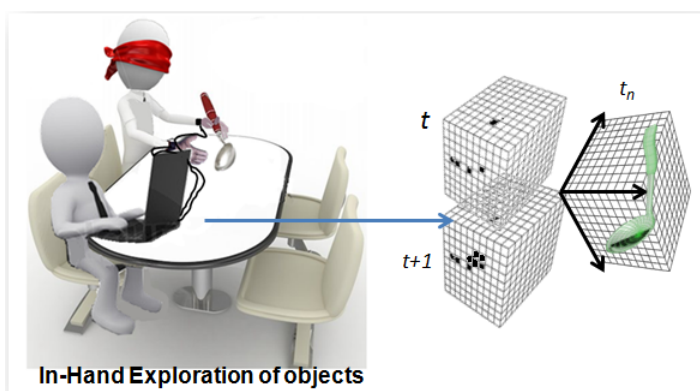
## 1.4 Organization of the Thesis

The following chapters will address in detail all aspects of the proposed approach. Chapter 2 introduces the proposed probabilistic framework to represent the full object model, partial volume of the object as well as the contact points of stable grasps acquired from human manipulation of objects.

The method is based on occupancy grid mapping using an object-centric representation. It is shown that multimodality is allowed to improve the object perception. Chapter 3 describes the object segmentation in single parts that is useful as a prelude to object grasping and shows that a good grasp is the result of the object graspable part identification. Chapter 4 proposes a method for object identification using cues from human in-hand manipulation of objects such as contact points and hand configurations to associate to hypotheses of candidate objects identities. Chapter 5 demonstrates the relevant features extracted from human manipulation of objects, including motion patterns, sequences of contact points, task modelling and identification which is useful information to assist in handling tasks. Chapter 6 addresses a proposed system that relies on human grasp experiences for robot grasp synthesis. Finally, the conclusion takes into consideration the advantages and limitations of the work, and addresses possible future work.

## Chapter 2

# Object Shape Representation



### 2.1 Introduction

Accurate modelling of the world (environment and its components) is important in autonomous robotics applications. More precisely, for grasping applications dealing with objects used in everyday tasks, the object information (intrinsic and extrinsic) acquired before the robot executes a task is crucial for grasp strategies. The object geometry (size and shape) play an important role in such applications, where its representation is also valuable for recognition into a class of known objects and also for identification of regions on the object surface proper for a stable grasp. Since the robotic end-effector usually relies on the knowledge of object geometry to plan or to estimate grasp candidates, the more accurate the geometry of the object, higher is the likelihood of success when estimating the candidate grasp for that object.

Different research fields (e.g., robotics and computer vision) still face the challenge of 3D object reconstruction and representation for recognition, localization and also for intervention tasks. In

case of object grasping, some approaches represent the object structure into polyhedral, which the object is represented as a finite number of flat faces. Based on some properties (constant normal and the position of a point on an object face), grasp synthesis adopts this object representation [Ngu87], [LDW99], [DLSX00]. However, representation of polyhedral objects are efficient for grasping when dealing with a low number of faces. Other object representations are based on the 3D point cloud distribution where meshes algorithms are computed to represent the surface of the objects for later compute algorithms to find contact points candidates based on the geometry representation [KDCI10], [RS07]. Other approaches directly matches the 3D point cloud into shapes primitives represented by superquadrics [MKCA03], [SEK09] to facilitate the grasp strategy. Usually the point cloud acquisition is derived from vision systems such as stereo cameras, time-of-flight cameras, multi-view from monocular camera, or also by other modalities such laser range finders (e.g., Hokuyo, Sick, etc.) or even from fusion (camera and laser), a basic example is the laser scanner (Konica Minolta). Several computer vision algorithms have been proposed for object modelling and recognition, for instance, space carving, octrees-based, shape from silhouette, etc.

This chapter presents the strategy for object shape representation using a probabilistic approach to deal with sensors uncertainty. The method used allows not only the full 3D shape representation, but also partial information of the object when less explored by the sensor as well as the contact points (fingers positions) on the object surface. The next sub-section presents the methodology adopted, as well as the advantages and disadvantages of the chosen approach; why it is important to acquire the object representation by combining multi-sensory information, and also the role of each individual modality, such as in-hand exploration of objects for representation and recognition.

## 2.2 Probabilistic Representation of Objects by In-Hand Exploration

The ability of manipulate different objects dexterously is one of the most well accomplished human skills. This skill is studied and pursued by researches in the robotic field with the objective of endowing a robot with this ability. Despite the different approaches found in the scientific papers which try to imitate the human dexterity and also the advanced robotic hands developed using new technologies, there are still differences between humans and robots in handling tasks. The ability of human manipulation involves different elements, such as hand, arm, eyes or head, where the human has many degrees of freedom and can easily deal with the control of those parts, whereas robots found

nowadays, do not have the control and skills that human do.

Humans use multiple sensory information to recognize objects. The best view for object representation for haptic modality, however, is the side the fingers naturally explore the most [NETB01]. Studies of human experiences in grasping and object exploration tasks can be applied in the robotic field, and so, endow a robot with similar skills through a generalization of these human abilities. Contour following is a common "exploratory procedure" that people use for determining the geometry of an object [KL90]. When performing in-hand exploration of objects, the key idea is to use the hand to extract object geometrical information. To achieve this goal, sensors are attached to the fingertips to acquire the hand movements on the object surface. To deal with the sensors uncertainty and real world noise, a probabilistic approach is used. This way, by computing the probability of the sensors 3D position at a specific location in the workspace (grid map equally divided into voxels), it is possible to know if that location belongs to the object surface.

Mapping techniques as occupancy grid [Mor88], [Elf89] has been used in the robotics field to describe an environment of a mobile robot. Two-Dimensional grid has been used for static indoor mapping as shown in [Thr02]. The idea is to verify the probability of each cell to be full or empty after the sensors observation. Probabilistic volumetric maps are also useful in the robotics field as presented in [RDC05] and [FCBD12]. The former provides means of integrating different occupancy belief maps in order to update a central multimodal map using a Bayesian filtering. A grid divides the workspace into equally sized voxels, and the edges are aligned with one of the axes of a reference coordinate frame. The coverage of each voxel given the sequence of batches of measurements is modelled through a probability density function. The probabilistic approach for building volumetric maps of unknown environments is also based on information theory. Each mobile robot uses an entropy gradient-based exploration strategy. A hierarchical Bayesian framework for multimodal active perception is presented by [FCBD12], where examples of data fusion (visual and auditory perception) are given. In this case, a Bayesian Volumetric Map (BVM) is introduced as a probabilistic framework for multimodal perception of 3D structure and motion which uses a log-spherical coordinate system to promote an egocentric trait for precision of the objects close to observer, in a bio-inspired way. The main motivations of using the probabilistic map are for a simpler way of static object reconstruction and representation (which is the purpose of this work: rigid objects representation); and because of the uncertainty of sensor noise caused by the real world (the sensor probability model depends on the characteristics of the sensor and the object being sensed).

During the in-hand exploration, the object might be moved or even released and re-grasped, for example, when one uses the other hand to assist the hand performing the in-hand exploration. This task becomes more complex when exploring objects not fixed in a specific position. To deal with moving objects during the in-hand exploration, the object rotation and translation need to be taken into consideration. Knowing the initial position of the object and the object displacements, transformations can be computed to have all points in the same frame of reference. In our case, we have a 6DoF sensor attached to the object so that we can map the hand contact points on an object centred frame of reference, and properly register the point clouds to build the object model. In the next two sections more details are given regarding the in-hand exploration using a single hand with a fixed object and later the strategy of exploring the object while the other hand is used to hold the object in such way that the exploration happens while the object is in movement.

### 2.2.1 Single Hand Exploration of Static Objects

In order to acquire the probabilistic representation of an object using a volumetric map, it is necessary to know a priori an estimated area where the object is placed for mapping. Some problems are addressed in this task, such as the need for the object to be completely static. Provided the sensors we are using to extract information about the object pose are magnetic trackers (Polhemus Liberty [Pol]), some limitation needs to be taken into account. After the initialization of the sensors, 3D points are constantly acquired so that when the hands are performing the exploratory procedure, if the fingers movements are not on the object surface, those generated 3D points probably do not belong to the object shape. If the involuntary movements (outside of the object surface) are sporadic, those 3D points will be ignored in the probabilistic map, since they do not have higher probability in their occurrence (the volumetric representation is based on probabilities higher than a threshold: 0.7). However, if a finger movement occurs with a higher frequency in the same region outside of the object surface, it can affect the result. Another possibility of this happening is when someone is exploring an object which is then moved from its original position. The final result will have a concentration of points representing the same part of the object in different positions. The solution is presented in the next subsection.

The setup for this experiment is composed of a wooden table, without any metallic parts, since the magnetic tracker is sensitive to nearby ferromagnetic materials. The rigid 3D object is fixed on the tabletop in a defined workspace. A workspace of  $35\text{cm}^3$  was defined on the table for the object

mapping. Each voxel of the volumetric map was defined to represent an area of  $0.5\text{cm}^3$  due to the sensor position resolution at  $30\text{cm}$  range is approximately  $1\text{mm}$  (as far the sensor is to the source, the resolution can vary a bit and the sensor error increases). During the exploratory procedure, at short period of discrete intervals the volumetric map is updated with the sensors measurements. Figure 2.1 shows the experimental setup area and Figure 2.2 shows the Polhemus magnetic tracker sensors attached to the hand.

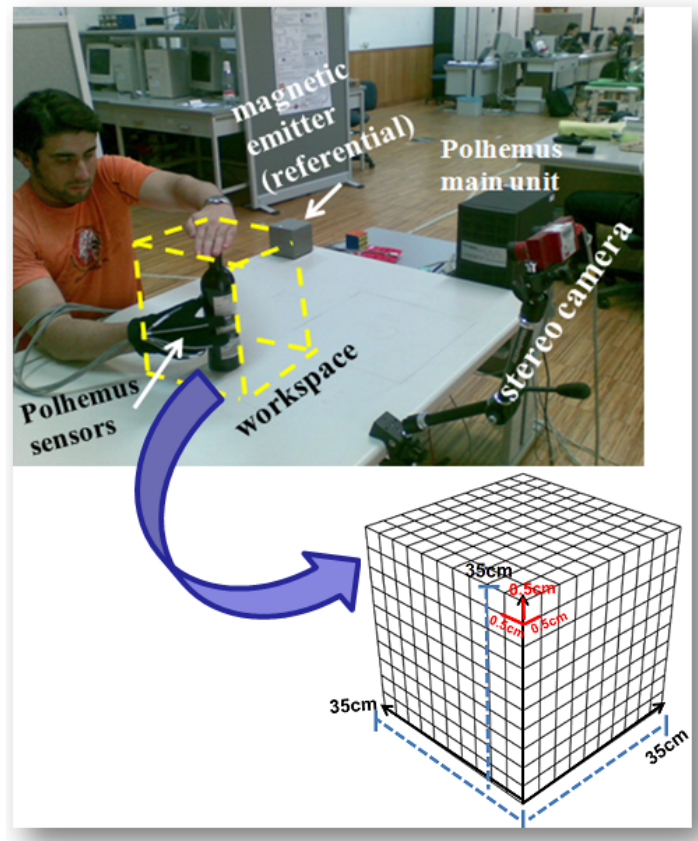


Figure 2.1: Experimental setup area and the workspace for mapping (grid  $35\text{cm}^3$  equally divided where each voxel is sized with  $5\text{cm}^3$ ).

### 2.2.2 In-Hand Exploration of Non-static Objects

The experimental setup for in-hand exploration of non-static objects follows the same structure as the single hand exploration. The difference with this new task is on the object position for mapping: the object does not need to be static any more. Since the object is allowed to move, it is possible to use the other hand (usually the left hand) to assist the right hand, holding the object for better exploration of the object. This task becomes more complex than the first one presented in the previous subsection due to the non-static objects. In this work the probabilistic map is used for the representation of the



Figure 2.2: Polhemus Liberty Motion Tracking System [Pol]: Magnetic tracker sensors attached to the hand (fingertips and back of the hand).

object shape, so there is a need to deal with the object rotations and translations during the in-hand exploration. Knowing the initial object position and the object displacements, we can compute the transformations to have all points in the same frame of reference (see figure 2.3). Given that the sensor attached to the object has 6DoF  $\{x, y, z, yaw, pitch, roll\}$ , we can compute the rotations and translation of the object. Before computing the probabilistic map in this context, we compute the transformations to have all points of the object in the same frame of reference. We compute the rotation matrix of the object in a specific instant using  $\alpha = yaw$  (rotation in  $z$  axis),  $\beta = pitch$  (rotation in  $y$ ) and  $\phi = roll$  (rotation in  $x$ ). To map the point cloud in the same frame of reference, for all points, we find the translation of the fingertip sensor to the object sensor and then we apply the rotation to that point:

$$p' = R_o t \quad (2.1)$$

where  $p'$  is the new position of the 3D point that we are mapping to the same frame of reference of the object sensor;  $R_o$  is the rotation matrix 3x3 of the object sensor and  $t$  the translation of the fingertip sensor to the object sensor. The rotation matrix can be built using  $(\alpha, \beta, \phi)$  given by the magnetic sensor representing the object rotation. This way, the rotation matrix is computed as follows:

$$R_{x,y,z}(\alpha, \beta, \phi) = R_x(\alpha) R_y(\beta) R_z(\phi) = \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) & 0 \\ \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} \cos(\beta) & 0 & \sin(\beta) \\ 0 & 1 & 0 \\ -\sin(\beta) & 0 & \cos(\beta) \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi) & \cos(\phi) \end{bmatrix}. \quad (2.2)$$



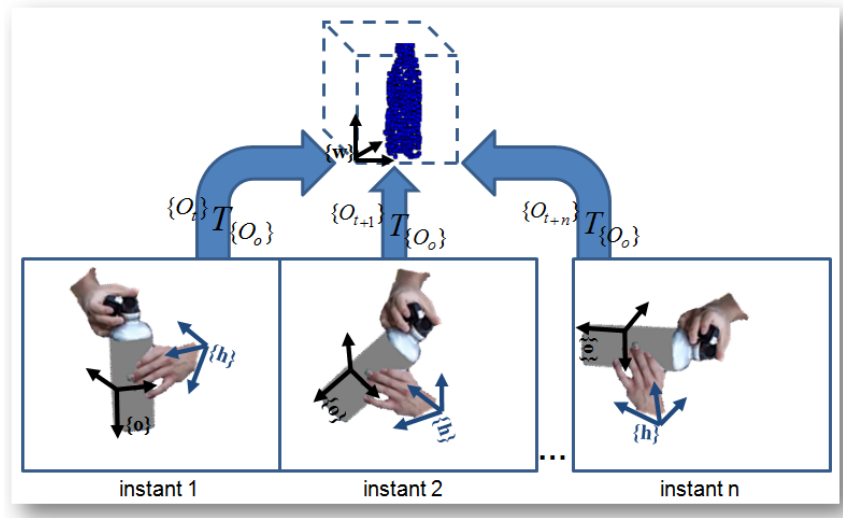


Figure 2.3: Example of in-hand exploration where is needed to compute the transformation at each movement of the object to register all 3D points in the same frame of reference. The points belonging to the object surface are represented by the map in the workspace.

### 2.2.3 Probabilistic Volumetric Map Cells Updating

Occupancy grids are discrete random fields, wherein each cell has an assigned value which represents the probability of the cell being occupied. The dimensions of the voxels define the spatial resolution of the representation. The edges of the grid are aligned with one of the axes of the world coordinate frame  $W$ . In this work, the map is a 3D grid comprised of a set of cells  $c \in \gamma$ , denoted as voxels, wherein each voxel is a cube with edge  $\varepsilon \in \mathbb{R}$ . The voxels divide the workspace into equally sized cubes with volume  $\varepsilon^3$  (see Figure 2.1). The occupancy of each individual voxel is assumed to be independent from the other voxels occupancy and thus  $O_c$  is a set of independent random variables as follows:

- $c \in M$ : Index a cell on the Map;
- $O_c \in |0, 1|$ : Binary value describing if the cell  $C$  is empty or occupied;
- $Z_c$ : In-hand exploration measurement that influences the cell  $c$ . It represents the measurements acquired from 5 sensors, each one returns the 3D location of each finger movement in the map;
- $P(O_c)$ : Probability distribution of preliminary knowledge describing the occupancy of the cell  $c$ , initially as a uniform distribution;
- $P(Z_c|O_c)$ : Probability density function corresponding to the set of measurements that influences

the cell  $C$  taken from the in-hand exploration measurements. This distribution is computed from the in-hand exploration sensor model.

The knowledge about the occupancy of a voxel  $c$  in the map  $M$ , after  $k$  measurements  $Z$  received from the sensors is represented by the probability density function  $P(O_c|Z_k^c)$ . Updating the 3D probabilistic representation of the manipulated object shape upon a new measurement  $Z_k$  means updating the probability distribution function  $P([O_c = 1]|Z_k^c)$  of the voxel  $c$  influenced by the measurement  $Z$ . Voxels are influenced by a measurement  $Z_k$  if the location associated with the sample computed from the sensor model  $P(Z_k^c|[O_c = 1])$  is contained in that voxel  $c$ . As analogously demonstrated in [RDC05], for each voxel  $c$ , the set of measurements  $Z_n^c$  contains the  $n$  measurements  $Z_k^c$  influencing a voxel  $c$ . The probability density function of the object shape representation of voxel  $c$  given the  $Z_n^c$  measurements influencing that voxel  $c$  is represented by  $P(Z_n^c|[O_c = 1])$ . To update the occupancy estimation of a cell in the map, the Bayes rule is applied:

$$P([O_c = 1]|Z_k^c) = \frac{P(Z_k^c|[O_c = 1])P([O_c = 1])}{P(Z_k^c|[O_c = 0])P([O_c = 0]) + P(Z_k^c|[O_c = 1])P([O_c = 1])}, \quad (2.3)$$

where  $P([O_c = 0]) = 1 - P([O_c = 1])$ ;  $P(Z_k^c|[O_c = 1])$  is given by the probability density function computed from the sensor model (more details on the sensor model are given in the next subsection) and  $P(Z_k^c|[O_c = 0])$  is a uniform distribution.

Assuming that consecutive measurements  $Z_k$  are independent given the cell occupancy, the following expression is obtained:

$$P([O_c = 1]|Z_n^c) = \beta P(O_c) \prod_{k=1}^n P(Z_k^c|[O_c = 1]), \quad (2.4)$$

where  $\beta$  is a constant representing a normalization factor ensuring that the left side of the equation sums up to one over all  $O_c$ .

Using the Bayesian formulation, the following equation can be written over the map updating:

$$P([O_c = 1]|Z_n^c) = P([O_c = 1]) \prod_{k=1}^n \frac{P(Z_k^c|[O_c = 1])}{\sum P(O_c)P(Z_k^c|O_c)}. \quad (2.5)$$

The cells occupancy in the map are probabilities that is updated over time as long as the sensors measurements are active. At the end of the in-hand exploration of the object, the cells are allowed to represent only two states: full or empty,  $O_c \in \{0, 1\}$ , so that a threshold is used for each cell to consider one of the two states:

$$O_c = \begin{cases} 0, & P(O_c|Z_n^c) < 0.7 \\ 1, & P(O_c|Z_n^c) \geq 0.7 \end{cases}. \quad (2.6)$$

Figure 2.4 shows an example of the probabilistic volumetric map and its utility. The map can be used to represent the full model of the object as well as partial volume of the object and contact points overlaid on the object surface derived from human demonstration of stable grasps.

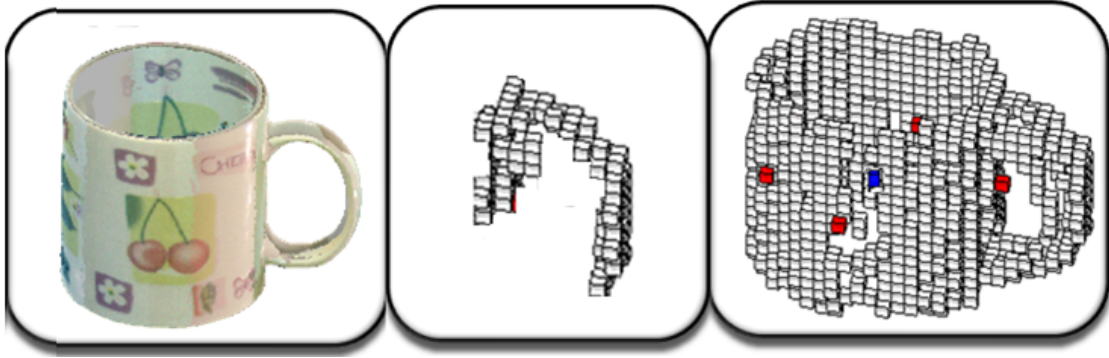


Figure 2.4: Examples of the probabilistic volumetric map. Left image: real object; middle image: partial volume of the object; left image: map of the full object model and contact points overlaid on the object surface (red voxels representing the contact points and blue voxel representing the centroid of the object to define its frame of reference).

The next subsection presents the probability density function acquired from the sensor model which is represented by a Gaussian distribution using the known sensor position error as the standard deviation and the sensors positions relative to the center of each cell in the map to model the normal distribution.

### Probability Density Function from In-Hand Exploration Sensor Model

This subsection will describe how the probabilistic map is affected by using the fingertip pose data. For in-hand exploration of objects, a magnetic sensor (Polhemus Liberty tracking sensor) is attached to each fingertip to acquire the shape of an object by the contour following procedure. The thumb and index fingers are the principal fingers for grip tasks and the index finger is responsible for opening and closing grip, allowing the thumb to maintain stability. For in-hand exploration, using only the thumb and index fingers, it is possible to achieve the object shape. Eventually, even with a static object or when the object is not static, that is, moving during the exploration, other fingers can be used (middle, ring and little fingers). The two main fingers are enough to cover the object shape through the movements around the object.

Each magnetic sensor attached to the fingertips returns the 3D coordinates of the finger location based on the sensor frame of reference (source/emitter of the Polhemus Liberty tracking system). The frame rate of each sensor was defined to be up to 15Hz. During the data acquisition, a workspace ( $35cm^3$ ) is defined in the experimental area for mapping. The grid space is divided into equally sized voxels (also denoted as cell) with  $0.5cm^3$ . During the displacement of each finger on the object surface, it is possible to identify in which grid cell that measurement is inserted. Due to the size of each cell, relative to the standard deviation of the magnetic tracking sensors measurements (up to 3 mm), inside each cell a 3D isotropic Gaussian probability distribution is defined,  $P(Z_k^c|O_c)$ , centred at the cell central point with the standard deviation  $0.3cm$  and mean value equal to the central point coordinates of the cell. In other words, this means that the model attempts to ensure that, upon receiving a measurement from the sensor attached to the fingertip, the closer the finger position is to the center of a specific cell of the map, the more probable that cell is occupied. Furthermore, during the object surface exploration, the more often that the finger passes through that cell, the cell probability is updated with higher certainty in which that given point position actually belongs to the object surface. The probability that a measurement belongs to a cell is given by a normal distribution using the known sensor position error as the standard deviation and the sensors positions relative to the center of each cell in the map as follows:

$$P(Z_k^c|O_c) = \frac{1}{(2\pi)^{3/2}|\Sigma|^{1/2}} e^{(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T\Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu}))}, \quad (2.7)$$

where  $P(Z_k^c|O_c)$  represents the probability distribution of the sensor measurement given a specific cell  $O_c$ ;  $|\Sigma|$  represents the determinant of  $\Sigma$  (sensor noise variation). It can also represent a scalar value. Due to the normalization, (2.7) takes the form:

$$P(Z_k^c|[O_c = 1]) = \exp\left(-\frac{(x-u_x)^2 + (y-u_y)^2 + (z-u_z)^2}{2\sigma^2}\right), \quad (2.8)$$

where  $(x, y, z)$  are the coordinates of the 3D point on the object surface and  $u$  is the central coordinate of the cell (for each axis).

Later on, the sensor model is used in (2.3) to update the probability of the cell in the map as exemplified previously.

The in-hand exploration of objects can be performed by using the thumb and the other fingers, i.e. the occupancy grid can be influenced by them over time, thus, expanding on the model for

cell update, the contribution of the sensor on each finger through time can be made explicit on the decomposition as follows:

$$P({}^0Z_{thumb}, \dots, {}^T Z_{thumb}, {}^0Z_i, \dots, {}^T Z_N, O_c) = P(O_c) \prod_{t=0}^T P({}^t Z_{thumb} | O_c) \prod_{i=1}^N P({}^T Z_i | O_c), \quad (2.9)$$

where  $T$  represents the current time instant and  $N = 4$  the remaining four fingers of the hand. This process for cell update over time recursively (i.e. initially using the cell probability as a uniform distribution: empty or occupied, and later the cell probability - updated with the Bayes rule - is used as prior for the next update), represents a Bayesian network.

The Bayesian network (BN) representation of the formalism applied to the decomposition of the joint distribution in which the sensor model was used is given on Figure 2.5. The plate notation relies on assumptions of duplicated sub-graph as many times as the associated repetition number (in this particular case the hand fingers); the variables in the sub-graph are indexed according to the repetition number; the links that cross a plate boundary are replicated for each sub-graph repetition; the distributions are in the joint distribution as an indexed product of the sequence of variables. This representation of plate notation is a useful add-on for Bayesian networks, introduced by Buntine [Bun94]. Bayesian formalisms for probabilistic model construction and some BN examples of occupancy grid model can also be seen in chapter 3 of the book [FD13] by Ferreira and Dias.

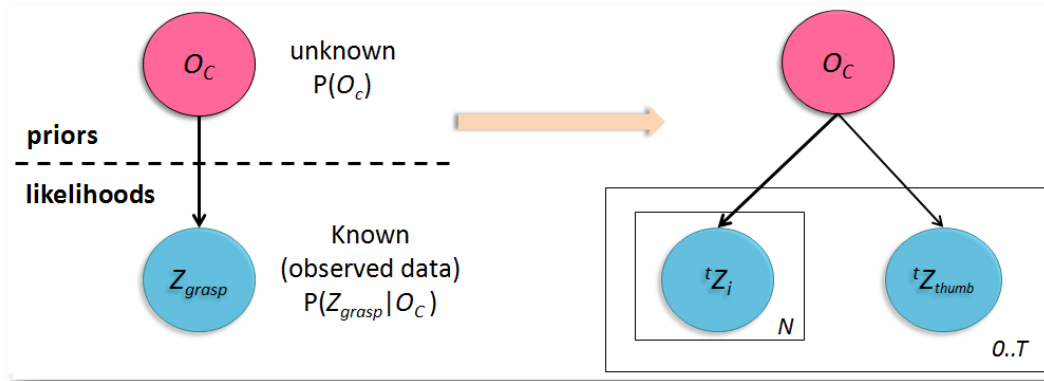


Figure 2.5: BN for the occupancy grid model for object representation using in-hand exploration. Left image shows the labels: prior, posterior and respective distributions, yet not necessary in DBN representations. The variables are defined in terms of their notation and conditional dependence. The instantiation is defined with their parameters and the random variables that support the model are fully described (i.e. their significance and measurable space). Right image shows that contribution of the sensor on each finger through time made explicit using the Bayesian network formalism with plate notation applied to in-hand exploration of objects.

### 2.2.4 Frame of Reference for Object-centric Representation

We are adopting an object-centric representation by estimating the frame of reference of each object by its geometrical properties. For that, we compute the 3D moment invariants to find the centroid of the point cloud which depends on the distribution of the points of the object surface. The centroid will be located at the densest part of the point cloud.

The 3D moment invariants are a measure of the spatial distribution of the mass of a shape. Let  $p(x, y, z)$  be a local continuous density function which is represented by the probability of a voxel to be occupied (e.g., occupied when  $p(x, y, z) \geq 0.7$ ; empty, otherwise). To estimate the location of the centroid of the point cloud, we first compute the zero<sup>th</sup> moment (sum of the voxels' probabilities) followed by the first moments for each axis  $\{x, y, z\}$  (sum of the product of all  $x$  by the probability of the respective voxel being occupied; the same for  $y$  and  $z$ ). Then the centroid  $\{cx, cy, cz\}$  is computed by the normalization of each  $c$  by the zero<sup>th</sup> moment. The centroid is useful not only to define the frame of reference of the object, but it can be used later with the contact points location to estimate how the object can be grasped.

### 2.2.5 Experimental Results: Object Shape by In-Hand Exploration

Everyday objects were used for in-hand exploration, where some objects have simpler geometrical shapes and others slightly more complex regarding concavity. Figure 2.6 demonstrates the raw data acquired during the in-hand exploration of objects. The raw data represents the objects 3D models before the computation of the probabilistic volumetric map.

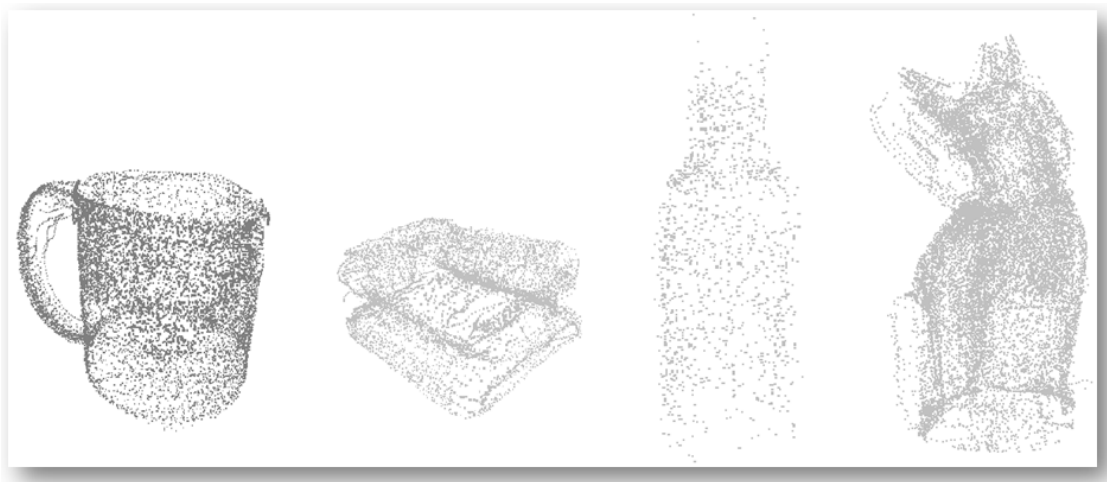


Figure 2.6: Raw data of 3D object models derived from in-hand exploration. Left to right: mug, sponge, bottle and wooden cat.

Figure 2.7 shows the computed probabilistic volumetric map of the wooden cat in different views and using different thresholds to decide whether the voxel is full (occupied) or empty. The cat maps are represented by the occupied cells.

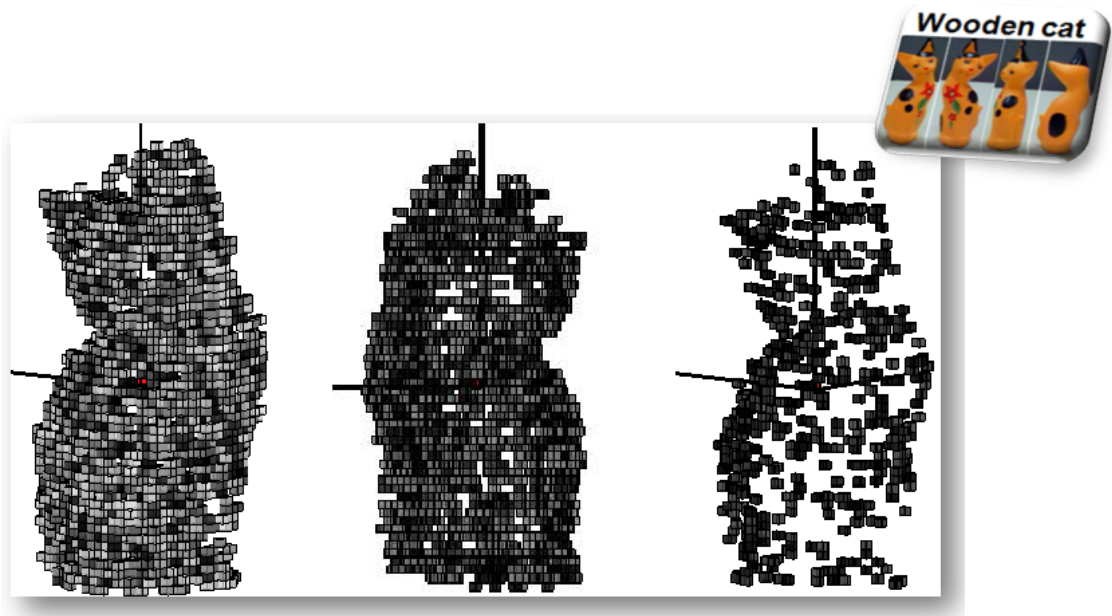


Figure 2.7: Computed probabilistic volumetric map of the wooden cat. Left image show all occupied cells of the object map. Right image shows the occupied cells using (2.6) with threshold = 0.6; and left image using threshold = 0.8.

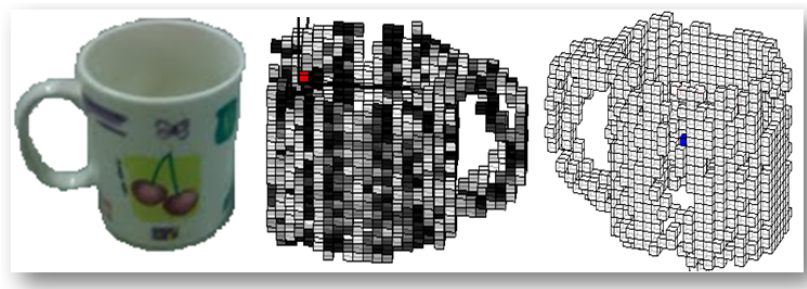


Figure 2.8: Object representation using the probabilistic volumetric map: the first image is the object, followed by the probabilistic volumetric map where the darker cells are those ones with higher probability than the lighter ones. It shows the most explored region of the object. The last image is the map showing clear cells just those ones occupied (probability higher than the specified threshold 0.7). We can see the global shape derived from the in-hand exploration.

In Figure 2.8, the achieved probabilistic map with occupied cells for the mug can be seen. The first image shows the real object, the middle image shows the occupied cells. The darker voxels represent the most explored regions. Due to the way the object was explored (vertical movements: top-down) we can see some pattern represented by the darker vertical voxels. Note that parallel to the



darker lines, from one side of the mug to the another side, we have the same pattern due to the thumb and index finger which were always in a parallel position during the exploratory movement. The last image shows the occupied cells (clear cells).

The probabilistic map demonstrated in Figure 2.9 shows the occupied cells for the sponge. In this case we are not dealing with the softness of the object, the exploratory procedure, contour following was performed in a soft way avoiding the deformation of the object. Here our interest is in rigid representation of the objects.

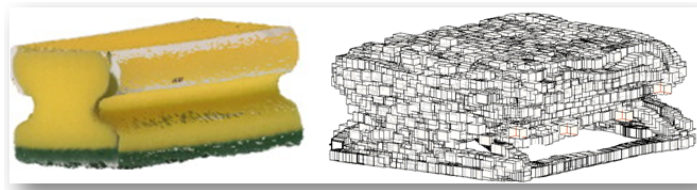


Figure 2.9: Object representation using the probabilistic volumetric map: sponge and its computed map.

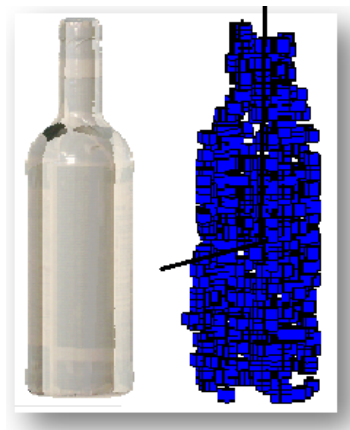


Figure 2.10: Object representation using the probabilistic volumetric map: bottle and its computed map.

Figure 2.10 shows the map achieved for the bottle. Figure 2.11 shows the in-hand exploration result of a spray bottle. Figure 2.12 shows an example derived from in-hand exploration with the non-static object as explained in subsection 2.2.2. In this case, different to the single hand exploration, the subject that is performing the in-hand exploration can use both hands, one to explore, and another one to assist holding the object during the exploratory procedure. It makes the object non-static, moving during the exploration passing by different rotations and translations which makes it necessary to have a registration process to have all point cloud in the same frame of reference. In the middle image some noise can be seen. This is due to sensors noise in the rotation data  $\{roll, pitch, yaw\}$  during the



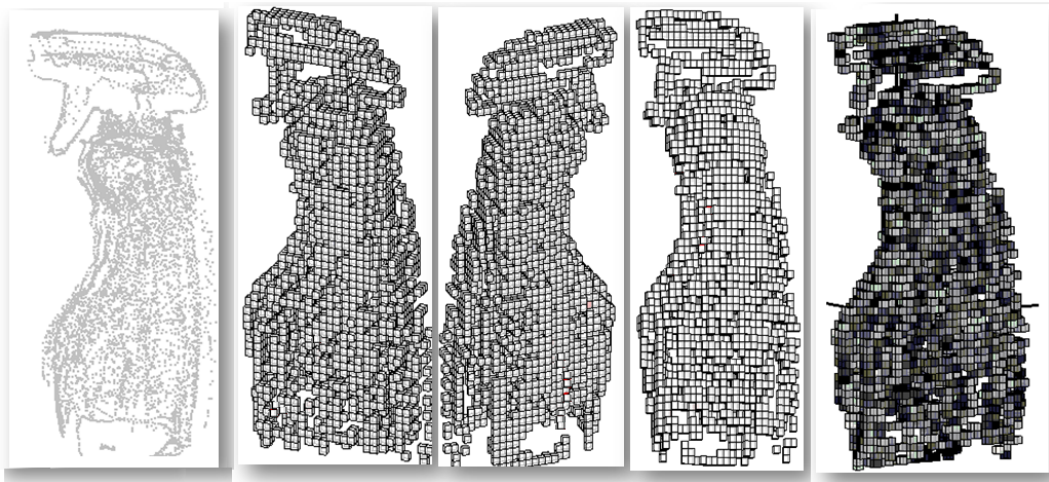


Figure 2.11: Object shape representation by in-hand exploration of a spray bottle. The first image (left to right) is the raw data (point cloud), next three images are different views of the voxels representation of the object shape, and the last image is the occupancy representation of the cells, the darkest ones represent the lower probabilities (less explored regions).

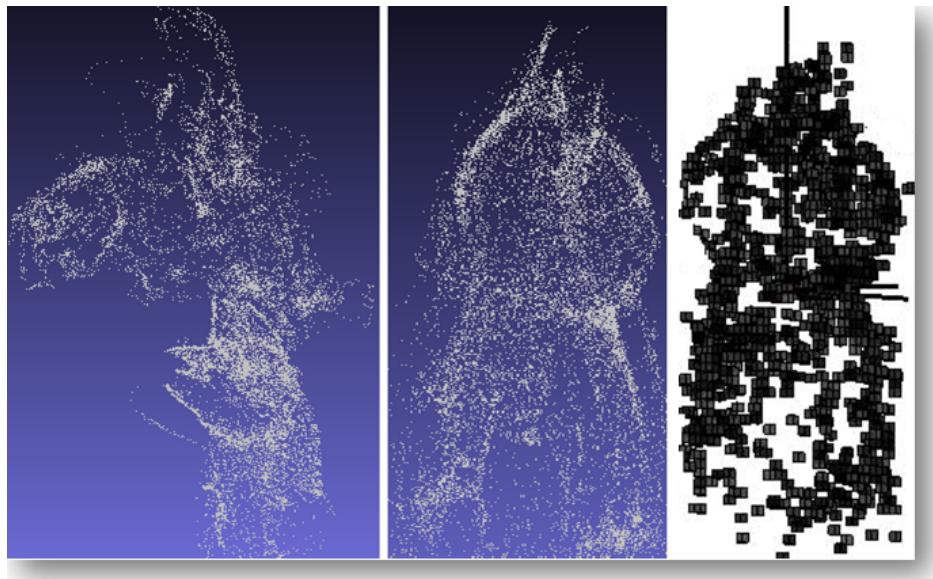


Figure 2.12: Example of registration process and mapping for moving objects. The first image shows the raw data of non-static object derived from the in-hand exploration of a wooden cat; middle image shows the point cloud after registration to a common frame of reference and then the last image is the computed probabilistic map in which the cells threshold  $> 0.8$ .

data acquisition.

The results show that valid objects models can be obtained. The volumetric map also provides information concerning the contact points and the most explored region of the object. Even with the limitations of single hand exploration of static objects, it is possible to reduce some noise caused by involuntary movements out of the object surface along the exploration by ignoring cells with low

probability.

An on-the-fly viewer on Blender (a free and open-source 3D computer graphics software) was developed. At same time as the data is acquired from in-hand exploration the probabilistic map is computed and the result is shown through the Blender viewer. Since Blender works with python language, a script to compute the volumetric map was developed in python and returns the occupancy grid probabilities to be updated and visualized in Blender software. Figure 2.13 shows the interface and rendering of the probabilistic volumetric map during in-hand exploration.

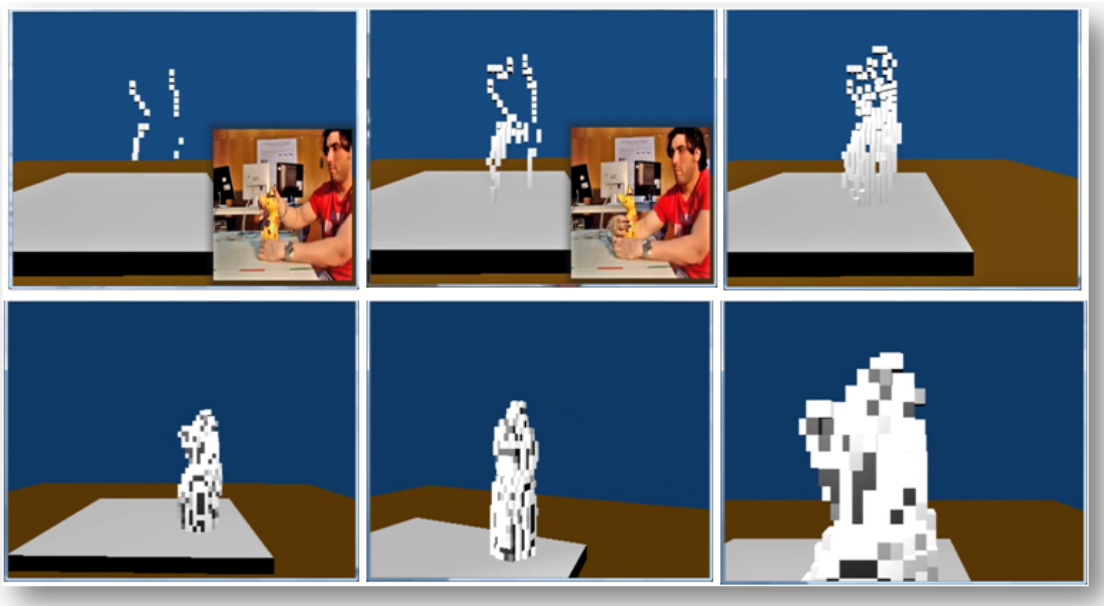


Figure 2.13: on-the-fly object modelling derived from in-hand exploration and the rendering in Blender software.

## 2.3 Multimodality and Fusion

Dealing with multiple sources of information from different sensory modalities when forming a percept is known as multimodality. The process of transforming a single percept from multiple sources of information is known as fusion. It is often seen as a product of models and can be used when the underlying models are defined independently so that they can be combined to form a shared variable.

Different sensors can be used to cooperate and assist the volumetric map. For instance, tactile information is a good option to filter undesirable outliers, that is, points outside the object surface. This can be done by using the tactile information when the positional data points are returned from the magnetic tracker sensors (attached to the fingertips). The 3D points are valid just when

the tactile sensors are activated, this means that the fingers are really touching the object surface, this way filtering out the points of the hand configuration transitions from the object surface during the exploration. It can be done because the data acquisition process is distributed and has synchronized time stamps for the data. The tactile sensing device consists of 360 sensing elements (Tekscan Grip System sensor [Tek]) which are distributed along the hand palm and fingers surface. The sensing elements are grouped into 15 regions, corresponding to different areas of the hand (distal, proximal and palm). Each of these regions can be defined as activation level states,  $R \in \{NotActive, LowActive, HighActive\}$ . When the volumetric map is used to demonstrate a stable grasp (fingertips positions on the object surface), instead of to represent only the full model of the object, the force of each fingertip can be associated to the cell in the map and this information can be kept for learning and later to reproduce the stable grasp by an artificial dexterous hand.

The next subsections present how to use other sensors modalities and how to build the volumetric model for data fusion to be used, for instance, with visual information to compute the cells probability in the map beside the sensors used for in-hand exploration.

### 2.3.1 Visual Cues to Complement the Object Model

The ability of human manipulation involves different elements such as hand, arm, eyes or head where the human has many degrees of freedom and can easily deal with the control of those parts, which is not an easy task for robots, because they do not have the control and skills that humans do. Humans use multiple sensory information to deal with and to recognize objects. Human studies [NETB01] show the importance of multimodality, more specifically when combined haptic and visual cues. Multimodal perception is used by humans to estimate the identity and properties of objects. The combination of these two modalities can be used to clarify ambiguous situations. The best view of the object when using visual cues is when the humans look at the direction straight ahead to the object, and the best view for object representation for the haptic modality, is the side the fingers naturally explore the most.

Using the probabilistic volumetric map to combine the visual and haptic information needs to work in the same frame of reference to update the global map, since local maps for each modality are acquired. The sensor model for visual information is used as the likelihood in Bayesian formulation to update the global map.

### Sensors Calibration

The calibration step allows us to work in the same frame of reference when dealing with different sensors. A calibration step was proposed for the Polhemus Liberty 240/8 magnetic tracking device and Videre STH-MDCS3-9cm stereo camera. But it can be easily adapted for other sensors that provide point cloud, for instance, Kinect device.

The first step of this calibration is to acquire the intrinsic and extrinsic parameters of the stereo camera. Then it is necessary to have the two sets of point clouds, one for each sensor modality. The magnetic sensors return the 3D points related to its frame of reference, so that to have the corresponding 3D points from the camera, the magnetic sensor needs to be identified in the left and right image of the camera to compute the corresponding 3D point. To make this task easier, a white tape is used on the sensor to facilitate the identification of the marker in the image, and so acquire the 3D point after the camera calibration (Figure 2.14). The intrinsic and extrinsic parameters as well as the stereo camera calibration is performed following [Bou] using the camera calibration toolbox for Matlab.

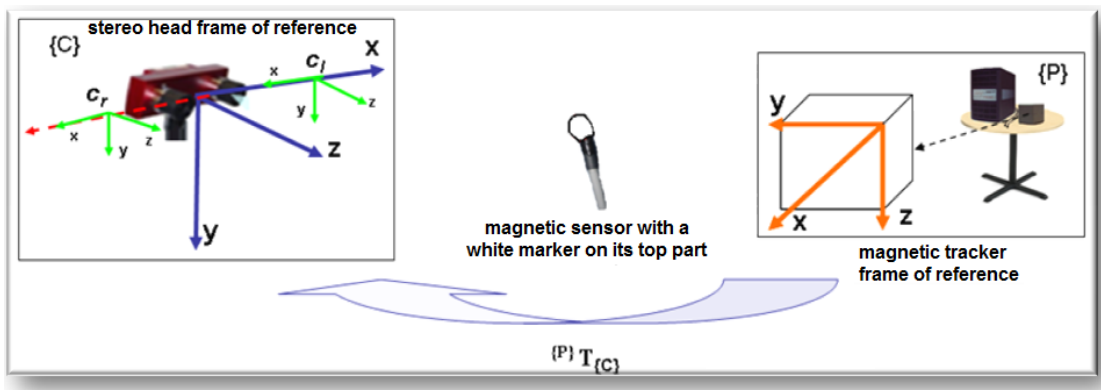


Figure 2.14: Calibration strategy: Using a white tape on the sensor facilitates later to find the marker in the image to compute the 3D point given the left and right images corresponding to the 3D point of the sensor in its frame of reference.

Thirty images (left and right) were acquired simultaneously with the 3D point from the tracker device sensor in different positions and orientations. The tracker sensor was attached at a tripod on a red piece of paper for easy displacement and easy localization in the image. This idea is originally inspired from auto calibration method between multi-cameras by [SMP05] and also based on the method of [ANP<sup>+</sup>09] where a laser pointer was used to get different viewpoints in the image for the calibration.

The frame of reference of the stereo camera and the magnetic tracker,  $\{C\}$  and  $\{P\}$  respectively, are rigid to each other. Collecting two sets of 3D corresponding points in two coordinate references,

${}^c p = \{{}^c p_i | i = 1, \dots, N\}$  and  ${}^p p = \{{}^p p_i | i = 1, \dots, N\}$ , then the 3D point from  $\{P\}$  to  $\{C\}$  is given by:

$${}^c p = {}^p R_c {}^p p + {}^p t_c. \quad (2.10)$$

To compute  ${}^p R_c$  and  ${}^p t_c$  (rotation and translation matrices of the homogeneous transformation) Arun's method [AHB87], also adopted by [ANP<sup>+</sup>09], has been used which is based on an algorithm to find the least-squares solution of  $R$  and  $t$  using singular value decomposition (SVD) of a 3x3 matrix.

Figure 2.15 shows the result of the calibration. Left image: the magnetic tracker sensor attached to a tripod during the calibration, and the reprojection of its 3D point is represented as yellow point in the image plane. Right image: The reprojection of in-hand exploration of the bottle in the image plane. Red dots represents the thumb and blue dots the index finger movements on the object surface.

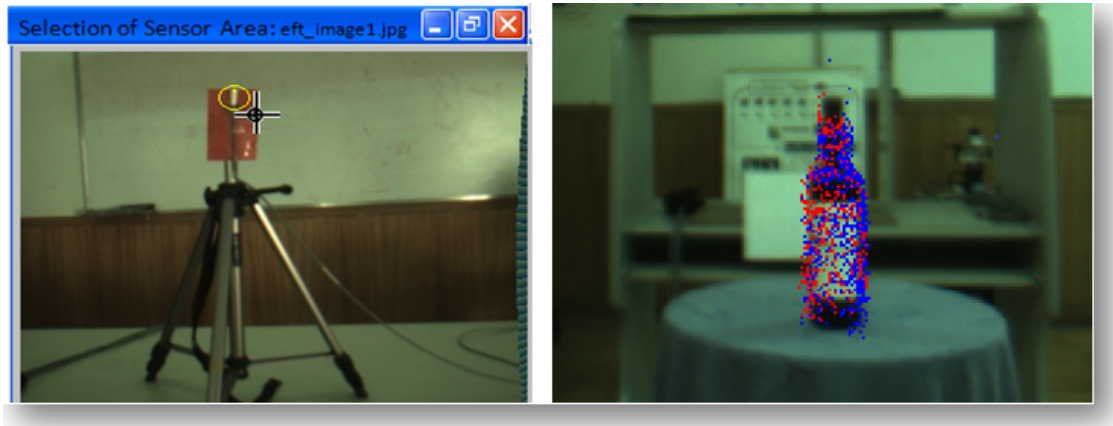


Figure 2.15: Reprojection of 3D points of  $\{P\}$  in the image plane  $\{C\}$ . Left image shows how is collected the left and right images from the stereo camera as well as the 3D points from the magnetic tracker. The yellow dot represents the 3D point from  $\{P\}$  to  $\{C\}$ . Right image shows the reprojection from  $\{P\}$  to  $\{C\}$  after the in-hand exploration of the bottle.

Figure 2.16 shows the evolution of the rotation and translation matrices estimates (left and right images respectively) by the calibration according to the number of points used.

Table 2.1 shows the average reprojection error values, in pixels, according to the number of 3D points used. The average error of the proposed calibration decreases when the method uses a higher number of points. It is possible to consider that for  $N = 20$  points the calibration method is stable.

Table 2.1: Reprojection error values in pixels (average error and standard deviation) according to the number of points.

	N = 7	N = 10	N = 13	N = 15
AE	12.363	8.917	7.333	6.491
SD	3.450	3.092	2.923	2.825

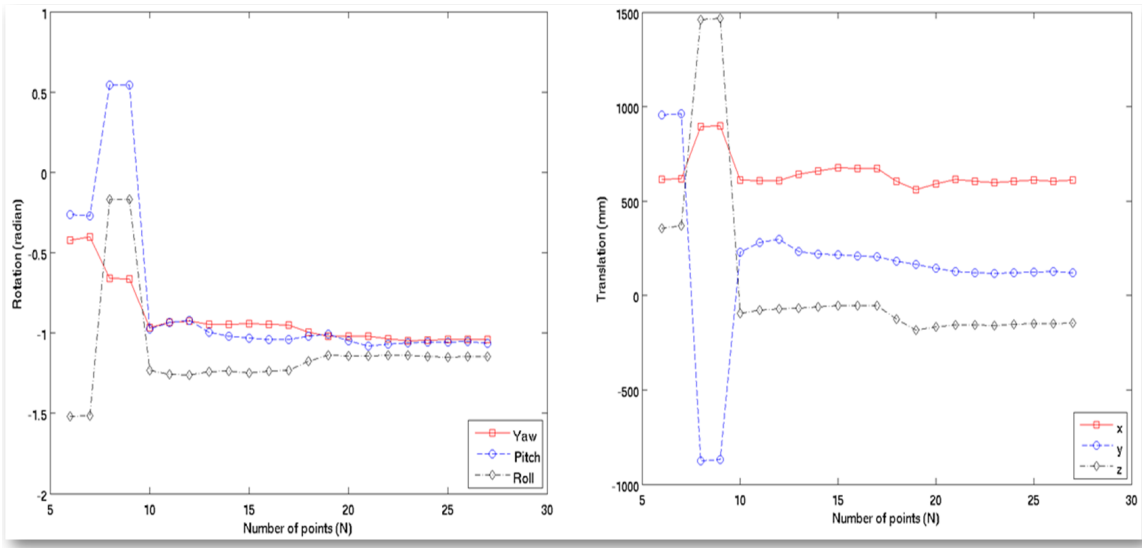


Figure 2.16: Evolution of the rotation and translation matrices estimates according to the number of points used in the calibration process.

A calibration toolbox for Matlab was developed to compute the transformation between the frames of references, to plot the average errors as well as plot of the points reprojection in another frame of reference, for instance, data from in-hand exploration to the image plane. More details about it can be seen in the annex of this thesis (appendix C).

### Vision Sensor Model

The variables presented in equation (2.5) to estimate the probability of each cell is kept. In case of using other sensors such as vision, the sensor model  $P(Z_{vision}|O_C)$  needs to be defined. Visual systems are usually implemented as deterministic algorithms returning the visual properties like range values. In the 3D world we have the position as  $X_k$  that represents the  $\{x, y, z\}$  coordinates of a point  $p$  and the range measurement  $d_j$  taken by the sensor and the position magnitude of vectors  $\vec{r}_{k,i}$  with the direction of the projection line corresponding to each measurement. Adopting the solution proposed by [RDC05], the voxel's occupancy belief can be defined as Gaussian distribution. This distribution uses the distance between the sensor and the detected obstacle, the distance between the sensor and the voxel's centre, following a linear model for the standard deviation. This solution relies on sensor calibration to estimate global values for sensor model parameters to achieve the linear model.

Using different sensors, the joint distribution decomposition of the relevant variables shows the dependency assumptions according to Bayes' rule, and the posterior is the probability distribution on each cell of the map using  $P(O_C|Z_{vision}Z_{grasp})$  for each voxel. This way, the demonstrated equa-



tions (2.3) to (2.5) can be adapted, adding to the model the likelihood correspondent to the visual information. The general model can be represented as follows:

$$P([O_c = 1] | Z_{vision,k}^C Z_{grasp,k}^C) = \alpha P(Z_{vision,k}^C | [O_c = 1]) P(Z_{grasp,k}^C | [O_c = 1]) P([O_c = 1]). \quad (2.11)$$

### 2.3.2 Bayesian Mixture Models

Bayesian modelling has often been used for multimodal fusion [CDB10]. Usually the Bayesian models follows assumptions of the naive Bayesian fusion model, in which the probability distribution over each sensation is independent of the others given the phenomenon. Maximum Likelihood Estimation (MLE) and also complete posterior distribution are usually employed to deal with multimodality and fusion.

Mixture models are known as distributions of parametric forms with multiple components. Generally, the probability distributions are Gaussian distributions. It can be seen as a basic tool to build a model. Clustering and classification are the most common use of mixture models.

Here, the mixture model allows the combination of different sensors models into one. This way, a global map can be updated after receiving the sensors measurements. Such models are weighted sum of unimodal probability distributions in order to yield a desired multimodal probability distribution as follows:

$$P(A) = \sum_{i=1}^N w_i \times P_i(A), \quad (2.12)$$

where  $N$  is the number of components (here represented by number of sensors);  $w_i$  is the weight of each component  $P_i(A)$ , and  $\sum_{i=1}^N w_i = 1$ .

#### Entropy as Confidence Level

The Shannon entropy (information theory)  $H$  as demonstrated in [CT91] is used as a measure of the uncertainty associated with a random variable. Here, entropy was adopted as a confidence level of the sensor models to update the global probabilistic volumetric map by computing weights to perform late fusion as mixture models. The weights are achieved using each entropy value computed as demonstrated in (2.13) for each local map (vision and in-hand exploration). In a Bayesian framework, each model contributes to the result of the inference in proportion to its probability. The mixture

model is presented directly as weighted sums of the distributions, then the combination of different models into one can be achieved. The intention is to look at the different sensor data to know the confidence of each sensor.

Through Bayesian techniques, we can implement the sensor fusion and use entropy  $H$  as a confidence level. A confidence variable  $w$  will be used as the weight for each sensor. The weight  $w$  can be expressed as a prior  $P(w)$  in the Bayesian rule. For each sensor (each local map), we can compute the entropy of the posterior probabilities as follows:

$$H(P([O_c = 1]|Z)) = -\sum_c P([O_c = 1]|Z) \log(P([O_c = 1]|Z)), \quad (2.13)$$

where  $P([O_c = 1]|Z)$  represents the posterior probability of the occupancy of each cell in the map achieved by a specific sensor. The variable  $Z$  represents the sensors measurements and  $c$  is the index of each grid cell.

Through the entropy  $H$  we can achieve the probability distribution of the weights of each sensor as follows:

$$w = 1 - \left( \frac{h}{\sum_{i=0}^n H_i} \right), \quad (2.14)$$

where  $w$  is the weight result;  $h$  is the current value of entropy that is being transformed in a weight;  $i$  is the index for each entropy value computed by (2.13).

Given the confidence of the occupied cells achieved by each sensor, we can fuse the sensors belief multiplying each local map to the correspondent sensor's weight achieved by the entropy. For each cell of the volumetric map we can compute the mixture model belief for local maps fusion:

$$P([O_c = 1]|Z_1, \dots, Z_S) = \sum_{i=1}^S P(w_i) P([O_c = 1]|Z_i), \quad (2.15)$$

where  $S$  represents the number of sensors.

Using (2.15), we update a global map with the probability distribution of each cell achieved by different sensors for data fusion. Employing entropy as confidence level we will be sure of the confidence of each sensor, that is, which is more reliable and then we build the global map from local maps (vision and in-hand exploration) with more certainty of the measures of the sensors. The only concern that needs to be taken into consideration when using the proposed methodology is the computational cost due to the necessity of calculating (2.13) and (2.15) for each cell.



A calibration between the sensors is needed to work with the local maps and the global one in the same frame of reference.

### 2.3.3 Experimental Results

To update the global map of the object, after the sensors calibration has been achieved, each local map is acquired separately (in-hand exploration and visual information). The global map is updated as described in the previous subsection 2.3.2. The local map acquired from in-hand exploration of objects is on-the-fly, and as long as the sensors return the data, the local map is updated. The local map achieved by visual information is achieved after computing the 3D points. Before that, a process of camera calibration is needed for it be possible to compute the correspondence of the the points in the left and right images from the stereo camera. For that, the Small Vision System (SVS) library (developed by Videre Design) [Vid] combined with OpenCV (computer vision library) [ope] were used. Details of the calibration process and the triangulation algorithm to compute the correspondence can be found in the SVS manual [Vid].

Figure 2.17 shows the result of the global map for the wooden cat. From left to right is shown: the image of the wooden cat; the object map achieved by in-hand exploration; the textured point cloud from stereo camera; the combined point cloud (from in-hand exploration and visual information) achieved after the calibration process; and the global map with information from both modalities. In the last image, the occupied cells threshold  $\geq 0.8$ . Using the calibration process we can work in the same frame of reference and then we can retrieve the texture of the object by the visual information. Blue color represents data from in-hand exploration and the other textured voxels are visual information. The blue voxels are those which the visual information had lower confidence compared to the in-hand exploration modality.

Figure 2.18 shows the result of the global map for the bottle. The top row shows the bottle and different views of the combined point cloud after the calibration process (allowing both point cloud in the same frame of reference). The bottom row shows two different views of the global map containing both modalities where the blue voxels are those ones with higher confidence from in-hand exploration modality and the other textured voxels are those ones where visual information prevailed.

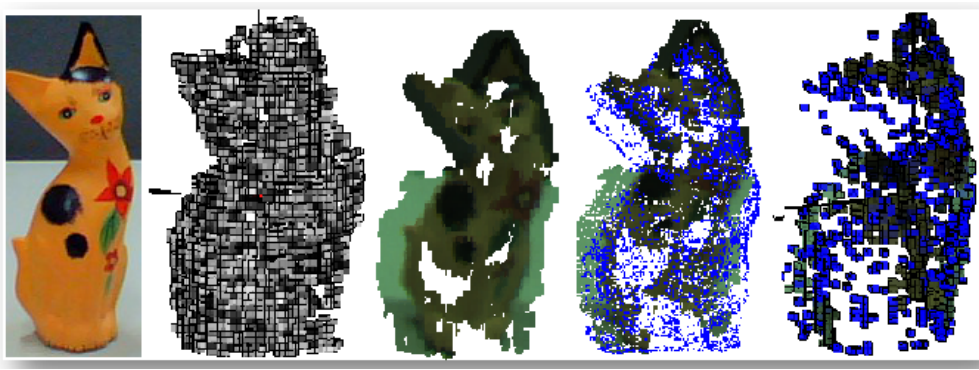


Figure 2.17: Probabilistic representation of the object global map (wooden cat) derived from in-hand exploration and vision.

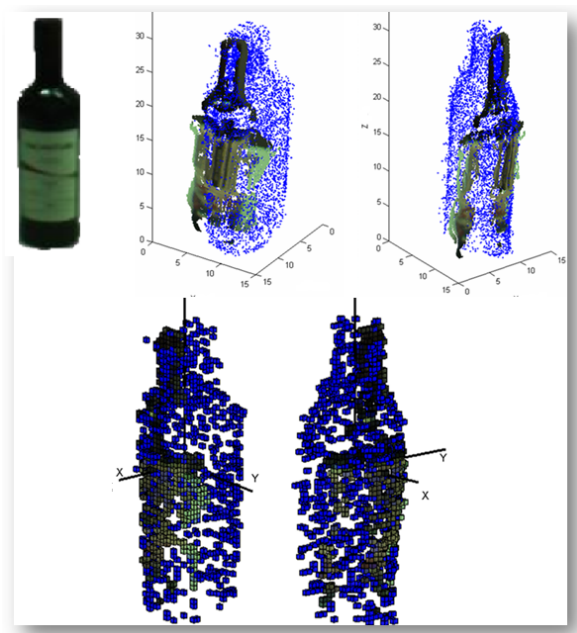


Figure 2.18: Probabilistic representation of the object global map (bottle) derived from in-hand exploration and vision.

## 2.4 Discussion

Through in-hand exploration of objects, suitable models are achieved by computing the probabilistic occupancy grid method. The probabilistic representation for 3D objects was presented and showed how given the sensors measurement of in-hand exploration, the probabilistic map is computed. The object centroid is computed to define the object frame of reference for object-centric representation. Two ways of in-hand exploration are presented: single hand exploration of static objects and in-hand exploration of non-static objects when usually the individuals use the left hand to assist the other hand for exploration. The results show that it is possible to achieve valid models of the object surface. We

can deal with the problem of moving objects along the in hand-exploration, making the registration of the points cloud to the same frame of reference, but we still can find some problems such as noise in the angles measure. The object-centric representation does not fall into the memory and computational cost problem (as happens sometimes in 3D mapping of environments) due to the area limit defined for object mapping, which is suitable to represent daily objects (intention of this work) for grasping in everyday tasks.

In this approach, data from different sensors is allowed to improve the object model by means of fusing the multimodal perception into a single percept. When various sources of information are involved to compute the volumetric map, each can sometimes lead to significantly different individual percepts. This way, both modalities (in-hand exploration and visual information) can complement each other to have a better representation of the object shape. The relative importance of different cues can be determined based on the mixture distribution after computing the weight function by Entropy to reach the sensors measurements confidence level. In this work, ambiguous information coming from different modalities is described as a mixed distributions to update a global map of the object.

The outputs of this work can be used in different robotic applications by integrating the object information and human demonstrations of manipulation tasks to search for stable grasps and other features (grasp transitions, estimated regions on objects for stable grasp given the task context, etc.) towards improving the autonomous robotic dexterous manipulation. The object map can also be used to overlay the partially observed volume of the object with data about hand-object contact points and tactile forces suggesting suitability for grasp planning since a unified model has the relevant observed information on how to grasp the object.

The publications related to this chapter's subject, in-hand exploration of objects, are listed as follows:

#### **Journal**

- Diego R. Faria, Ricardo Martins, Jorge Lobo, Jorge Dias. "Extracting Data from Human Manipulation of Objects Towards Improving Autonomous Robotic Grasping". *Robotics and Autonomous Systems*, Elsevier, Volume 60, Issue 3, March 2012, Pages 396-410, 2012.

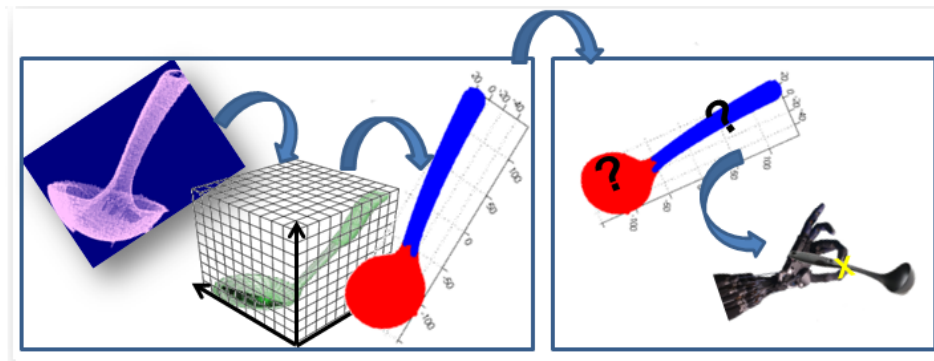
#### **International Conferences**

- Diego R. Faria, Ricardo Martins, Jorge Lobo, Jorge Dias. "Probabilistic Representation of 3D Object Shape by In-Hand Exploration". In *Proceedings of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS'10 - Taipei, Taiwan - October 2010*.
- Diego R. Faria, Ricardo Martins, Jorge Dias. "Grasp Exploration for 3D Object Shape Representation using Probabilistic Map". in *Proceedings of DoCEIS'10 - Doctoral Conference on*

Computing, Electrical and Industrial Systems. Costa da Caparica - Portugal, February, 2010.  
Springer - ISBN: 978-3-642-11627-8.

## Chapter 3

# Segmentation and Modelling of Object Components



### 3.1 Introduction

Humans usually identify object parts in order to choose a suitable region to grasp and they can easily distinguish between things they have seen in the past and novel objects. The RBC theory [Bie87] motivated us to adopt methods of object segmentation, since it reveals that humans are able to identify objects by segmenting them into shapes (geons). Geons are composed of different shapes primitives (e.g., cylinders, sphere, cones, etc.) that can be assembled in various ways to form an unlimited amount of objects. These geons are derived qualitatively using attributes of generalized cylinders, describing characteristics of its shape, symmetry and size. Biederman [Bie87] suggests that segmenting objects for their identification does not depend on our familiarity with these objects. Thus, the same process for any object can be done, whether it is familiar or not. If we see an unfamiliar object,

despite its unfamiliarity, we are able to identify this object by segmenting it into parts at regions of deep concavity, looking for known or familiar geons.

Based on this study, the intention is to apply a segmentation process on the object derived from the object global shape representation. Then an approximation of geometrical primitives is applied on the components of the object given by the segmentation process. The importance behind these processes applied on objects is to use information of everyday objects, such as global shape and its segmented components, to acquire the probability distribution of a graspable part given by human demonstrations. The segmentation of the object into components and their approximation using superquadrics decreases the huge amount of potential grasps for that specific object part. This knowledge can be extended for "unknown" objects to estimate the object location and the candidate grasp given the information previously acquired from similar objects. Since an unknown or new object is segmented and approximate to a known geometrical primitive, the system can consider this object similar to another already observed to plan possible grasp after identify suitable regions on the object for grasping.

The next subsections present the process for object segmentation and geometrical primitives modelling for each component of the object.

## 3.2 Object Segmentation

In this section we will introduce two different methods that can be applied to segment an object to find suitable components for robot grasping. In the first approach, we are clustering the output of the probabilistic volumetric map (point cloud acquired by in-hand exploration) or from another modality (e.g., RGB-D camera), to find the possible object components. By clustering, we can achieve outlier removal and we can also keep the position and size information of the object. According to the points cloud structure, using the known method of Gaussian mixture models (GMM), we can find the most suitable clustering that will represent a component of the object. The second method of segmentation is based on the major axis of the object to find the three possible components (top, middle and bottom parts of the object), by analysing the magnitude of the object in each axis direction.

The objective of the segmentation is to simplify the object shape into components, modelling each segment by geometrical primitives. Thus, from an "unknown" object, it will fall into a familiar class of objects by looking for the combination of geometrical primitives already known. Even when the object is too complex in its shape, at least one primitive can be recognized to make it possible to

generate candidate grasps for that similar or identified primitive reusing the knowledge from previously known objects. The main problem of 3D object segmentation into parts is to decompose the complete object surface into different useful regions for grasping.

Within the literature of 3D mesh segmentation, there are two main approaches that satisfy this condition, shape-based and boundary-based approaches. Shape-based approaches, known also as primitive based approaches, decompose objects into parts according to similarity between the shapes of parts models and objects parts [SLM94], [LJS97], [GB93], [DPR92], [CJB03]. Before segmentation, these approaches define a set of model shapes, such as a cylinder, a cuboid and a cone. They then generate a hypothesis of the object representation as an assembly of shapes chosen from the defined set of shapes. A measure of similarity between the hypothesis and the real object shape is then computed. If this measure is above a threshold, another hypothesis is generated. Otherwise, the segmentation process is terminated and the desired part representation is obtained. The advantage of such approaches is that part segmentation and part identification are performed simultaneously. On the other hand, their main problem is the possible non-uniqueness of the decomposition. For example, an object roughly shaped as a cylinder may be represented as one cylinder or also as the assembly of two cylinders with the same diameter.

Boundary-based approaches find first object boundaries. A common strategy in this kind of segmentation is to compute surface features which contrast boundary and non-boundary points and decompose the object into parts at boundary points. While many researchers have addressed the problem of 3D model segmentation, we can find three main features that are used in all boundary-based approaches: surface curvature, concaveness estimation and electrical charge physical features. The authors in [WL97], presented a physics-based part segmentation approach. The novelty of this method is that part boundaries are determined by using the idea of electrical charges instead of traditional curvatures for each vertex. The disadvantage of this method is the high computational cost involved in computing electrical charges. On the other hand, the curvature estimation for 3D meshes is not a trivial operation, as it is mathematically defined for a smooth surface only [MP77]. Most of the existing algorithms are computationally expensive [MW99], [PRF02], [RB02], [RKS00]. The authors in [ZPKG02] proposed a simple segmentation algorithm using Gaussian curvature analysis and more recently, a 3D mesh watershed-based segmentation algorithm using Gaussian curvature and concaveness estimation have also been proposed by [CG06].

Various approaches using object shape segmentation or approximation for grasping purposes

can be found in the literature. In [MKCA03], the authors model the objects as set of simple shape primitives (e.g., spheres, cylinders, cones and boxes). Then rules are defined to generate a set of grasp starting positions and pre-grasp shapes that can then be tested on the object model. The method can limit the huge number of possible hand configurations for grasp planning by using hand pre-shapes. The authors proposed planner required a manually constructed primitive decomposition of the object, then [GALP07] removed the need for a manual decomposition and introduced a multi-level superquadrics representation. The proposed iterative segmentation algorithm by [LDSA05] is applied for grasping non-convex objects. Firstly the inertial axes of the whole object is computed, afterwards candidate grasps are generated. When there is failure to obtain valid grasps, the object decomposition process starts. At each iteration of the decomposition step, two components are obtained and the authors try to generate feasible grasps on them. The process is repeated until a grasp is found or the decomposition terminates.

Considering that grasping problem induces a huge number of degrees of freedom, some methods use object decomposition into parts to define a small search space that is likely to contain many grasps, and others are strictly directed at the object geometry to compute possible grasps. Usually these methods based on object models do not attempt to find suitable grasps based on knowledge acquired by human demonstration for learning. The human demonstration strategy usually does not pay attention to the object model when dealing with grasps, obtaining only the grasp and movements characteristics to grab an object, and it is consequently adapted to the task requirements. The objective here is to pay attention to the object, by segmenting the object and later modelling the components by using geometrical primitives approximation. This will allow to overlay the human grasp demonstration on the object regions (e.g., contact points of stable grasps on the surface) to learn the object graspable parts.

### 3.2.1 Mixture Distribution-based Segmentation

In this subsection, we are addressing a segmentation method by means of Gaussian Mixture Models given the object point cloud. It is another alternative that allows for the search for segments on the object which are candidate regions for grasping. The estimation of the parameters (e.g., mean, covariance matrix and weight) of each individual Gaussian density function (cluster) is accomplished by the Expectation Maximization (EM) algorithm, also known as *EM clustering*, which is an iterative method that attempts to find the maximum likelihood estimator of a parameter. A global parameter



that needs to be set is the maximum number of clusters  $k_{max}$ . An optimal  $k_{max}$  can be estimated by MDL (Minimum Description Length) penalty function [Ris78] on the input data. In our case we pre-set  $k_{max} = 3$ , since we have observed that it is sufficient for hand-held everyday objects.

Let a set of points (i.e., object point cloud represented by a matrix of points) be  $\mathbb{P} \in \mathbb{R}^3$ , generated independent and identically distributed (i.i.d.) by a mixture of  $k$  Gaussians, and  $\wp_j \subset \mathbb{P}$  representing a subset of the point cloud, that is, a specific cluster  $j = \{1, \dots, k\}$ . Each set or subset of 3D points encloses many 3D points,  $\mathbf{P}(x, y, z) = \mathbf{P}_i \in \wp_j \subset \mathbb{P}$ . The entire set of parameters is denoted as  $\theta = \{(w_j, \mu_j, \Sigma_j)\}_j^k$ , where  $\mu_j$  represents the mean of a specific cluster  $j$  (also represented as  $\wp_j$ ), the covariance matrix is represented by  $\Sigma_j$ , and  $w_j$  represents the weight of the cluster, which specifies how likely each Gaussian is selected. The derivation of EM algorithm to estimate the set of GMM parameters  $\theta$ , for  $\mathbb{P}$  (input) and any  $\mu_j, \Sigma_j$ , is denoted as Gaussian according to the following expression:

$$\phi(\wp_j | \mu_j, \Sigma_j) \triangleq \frac{1}{(2\pi)^{d/2} |\Sigma_j|^{1/2}} \exp\left(-\frac{1}{2}(\wp_j - \mu_j)^T \Sigma_j^{-1} (\wp_j - \mu_j)\right). \quad (3.1)$$

The *pdf* for the combination of the  $k$  models to search for the most likely combination  $\theta$  of models to explain the observed data is achieved by (3.2). This means a learning of mixture models, so that we are searching for the combination of the proper clusters that better describes the input data  $\mathbb{P}$ , achieving the subset of points  $\wp_j$ , representing the proper cluster  $j$ , where  $j = \{1, \dots, k\}$ .

$$P(\wp_j | \theta) = \sum_{j=1}^k w_j \phi(\wp_j | \mu_j, \Sigma_j), \quad (3.2)$$

where  $w_j > 0$ ,  $\sum_k^j w_j = 1$  and  $\theta = \{(w_j, \mu_j, \Sigma_j)\}_j^k$ .

To summarize the EM algorithm that estimates the GMM parameters, we apply then the following steps, first compute the initial log-likelihood, that is used later to check the convergence of the EM algorithm:

$$\ell^{(0)} = \frac{1}{n} \sum_1^n \log\left(w_j^{(0)} \phi(\wp_j | \mu_j^{(0)}, \Sigma_j^{(0)})\right), \quad (3.3)$$

where  $n$  is the amount of samples contained in  $\wp_j$ ; the initial estimates  $w_j^{(0)}, \mu_j^{(0)}, \Sigma_j^{(0)}, j = \{1, \dots, k\}$  can be randomly chosen. During the initialization, we can take some  $k$  of the object point cloud  $\mathbb{P}$  as the first estimate of the cluster mean, setting the first estimate of the covariances to be the identity matrices, and the first guess at the weights  $w_i = \dots = w_k = 1/k$ , which is common when using this

algorithm. A better alternative commonly used, and also adopted in our work, is using the K-means algorithm to provide a good initialization for the EM. The **E** (Expectation) step is achieved by (3.4). Let  $\gamma_{ij}^m$  be the estimate at the  $m^{\text{th}}$  iteration of the probability that the  $i^{\text{th}}$  sample was generated by the  $j^{\text{th}}$  Gaussian component (cluster), as demonstrated as follows:

$$\gamma_{ij}^m = \frac{w_j^{(m)} \phi(\mathcal{D}_j | \mu_j^{(m)}, \Sigma_j^{(m)})}{\sum_{j=1}^k w_j^{(m)} \phi(\mathcal{D}_j | \mu_j^{(m)}, \Sigma_j^{(m)})}, \quad i = \{1, \dots, n\}. \quad (3.4)$$

To facilitate the representation of the next formulas, we use a notational simplification, denoting the total membership weight of the  $j^{\text{th}}$  cluster as  $s_j^{(m)}$  as follows:

$$s_j^{(m)} = \sum_{i=0}^n \gamma_{ij}^{(m)}. \quad (3.5)$$

Consequently, the **M** (maximization) step is given by:

$$w_j^{(m+1)} = \frac{s_j^{(m)}}{n}, \quad (3.6)$$

$$\mu_j^{(m+1)} = \frac{1}{s_j^{(m)}} \sum_{i=0}^n \gamma_{ij}^{(m)} \mathcal{D}_j, \quad (3.7)$$

$$\Sigma_j^{(m+1)} = \frac{1}{s_j^{(m)}} \sum_{i=0}^n \gamma_{ij}^{(m)} \left( \mathcal{D}_j - \mu_j^{(m+1)} \right) \left( \mathcal{D}_j - \mu_j^{(m+1)} \right)^T, \quad (3.8)$$

where the maximization step is computed for all clusters  $j = \{1, \dots, k\}$ . Afterwards, the new log-likelihood is computed to verify the convergence of the algorithm  $|\ell^{(m+1)} - \ell^{(m)}| > \delta$  (pre-set threshold) as follows:

$$\ell^{(m+1)} = \frac{1}{n} \sum_1^n \log \left( w_j^{(m+1)} \phi(\mathcal{D}_j | \mu_j^{(m+1)}, \Sigma_j^{(m+1)}) \right). \quad (3.9)$$

All steps mentioned above are summarised in Algorithm 1. More details about the theory and use of the EM algorithm and the GMM learning can be found in [GC11].

Afterwards, each cluster generated by the EM clustering is represented as a segmented region of the object that can be used as a candidate region for grasping. Examples of the segmentation using the GMM method are shown in Figure 3.1.

The success of the GMM method depend on how the clusters are generated, taking also into consideration the amount of clusters. Sets with a larger number of points have a significant impact on

**Algorithm 1:** EM Algorithm for Estimating GMM Parameters

- 1 Inputs: Object point cloud  $\mathbb{P}$
- 2 **Initialization:** Choose the initial estimates  $w_j^{(0)}, \mu_j^{(0)}, \Sigma_j^{(0)}, j = \{1, \dots, k\}$ , and compute the initial log-likelihood as demonstrated in eq. (3.3).
- 3 **while**  $|\ell^{(m+1)} - \ell^{(m)}| > \delta$  (*pre-set threshold*) **do**
- 4     **E step:** For  $j = 1, \dots, k$ , compute  $\gamma_{ij}^{(m)}$  and  $s_j^{(m)}$  as exemplified in eq. (3.4)-(3.5)
- 5     **M step:** For  $j = 1, \dots, k$ , compute the new estimates:  $w_j^{(m+1)}, \mu_j^{(m+1)}$  and  $\Sigma_j^{(m+1)}$  as demonstrated in eq. (3.6)-(3.8)
- 6     **Convergence step:** compute the new log-likelihood  $\ell^{(m+1)}$  as shown in eq. (3.9)
- 7 Outputs:  $\theta = \{(w_j, \mu_j, \Sigma_j)\}_j^k$

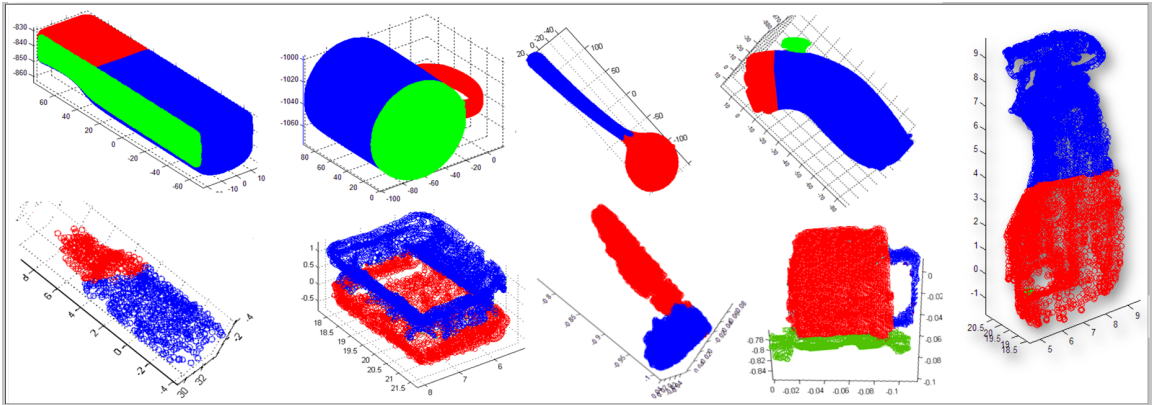


Figure 3.1: Everyday objects (wii-mote, mug, sponge, bottle, ladle, Nintendo nunchuck and spray bottle) segmentation using GMM clustering. These objects were acquired by different sensor modalities to test the segmentation. Top row: laser scanner; bottom row and last image (at right): in-hand exploration (bottle, sponge and spray bottle); RGB-D device (ladle and mug).

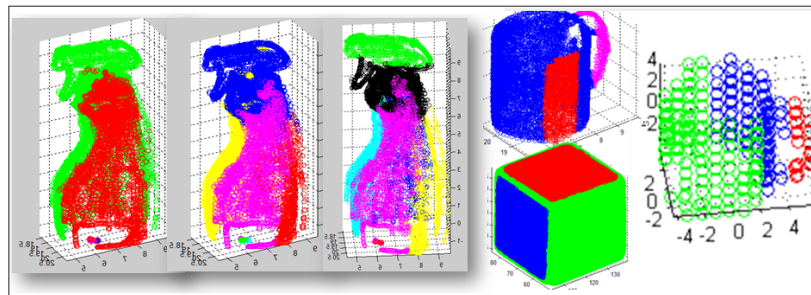


Figure 3.2: Examples of segmentation with little success of everyday objects using GMM clustering. Some segments cannot be considered as a good candidate region for grasping (based on a qualitative analysis). Some of the segmented regions are not suitable for subsequent approximation by a geometrical primitive.

the algorithm's processing time, due to the iteration steps to estimate the parameters of each cluster.

When  $k_{max}$  is defined to be more than three clusters, then the results for everyday objects are not so

satisfactory, because sometimes an object with too many segments does not present reasonable candidate regions for grasping. Some examples of segmentation with little success (based on a qualitative analysis) are presented in Figure 3.2.

### 3.2.2 Segmentation based on Major Axis Analysis

For the segmentation, we are assuming that all everyday objects are composed of a maximum of three components: Top; Middle and Bottom, if the object size analysed in the major axis satisfies a defined threshold  $\delta$  of size. Otherwise, the object will be segmented into two parts (top and bottom), or in the case of a small object, then it will be considered as a single segment (comprised of only one shape primitive). The threshold  $\delta$  for segmentation were found heuristically, where three segments are achieved when the object size is bigger than 7cm. Figure 3.3 shows the example of the segmentation strategy. The object frame of reference is found based on its center (origin), and the axes are defined as  $\{x,y,z\}$ : right-hand rule ( $x$ : index finger pointing to front;  $y$ : middle, pointing to the left; and  $z$  thumb, pointing to up position). For the segmentation we assume the major axis is in a vertical position, having the segments as top, middle and bottom.

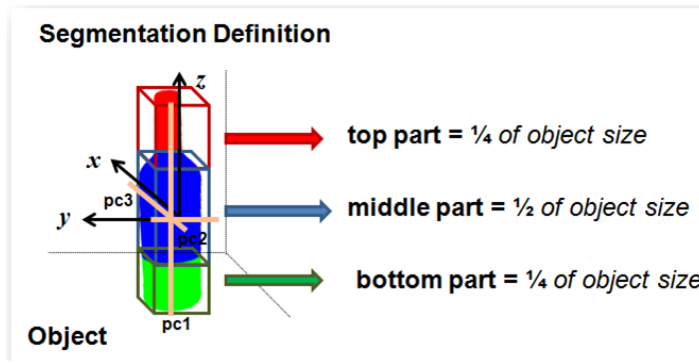


Figure 3.3: Object Segmentation Definition. The object is segmented into three parts: top, middle and bottom. The segmentation takes into consideration the major axis (pc1: principal component), i.e., the axis with bigger length.

The segmentation process is based on the idea of methods that analyses the major axis, such as the known method in the state of the art, Principal Coordinate Analysis (PCoA), which is a related statistical technique often used in information visualization for exploring similarities or dissimilarities of the data. The idea here is simple; we arrange the data by the major axis based on distance measures. More specifically, given a set of the 3D points  $\mathbb{P}$  that form an object, where a 3D point follows the notation  $\mathbf{P}_i = \mathbf{P}(x,y,z,r,g,b) \in \mathbb{P}$ , we search for the axis vector with higher magnitude. By analysing

the points in each axis  $\{x, y, z\}$ , we can search the points with maximum and minimum coordinate values to compute the object length in the Cartesian space, using the distance between these points. Let  $\vec{e}$  be a vector with the points at maximum and minimum coordinate in a specific axis  $\{x, y, z\}$ , which may take the following forms:  $\mathbf{e}_x = \{x_{min}, x_{max}\}$ ,  $\mathbf{e}_y = \{y_{min}, y_{max}\}$ ,  $\mathbf{e}_z = \{z_{min}, z_{max}\}$ . Then the higher magnitude is computed as follows:

$$\|\mathbf{e}\| = \sqrt{\mathbf{e}_i^2 + \mathbf{e}_j^2}, \quad (3.10)$$

where  $i$  is the first element of  $\mathbf{e}$ , representing the point at the minimum coordinate in a specific axis and  $j$  is the second element of  $\mathbf{e}$ , representing the point at the maximum coordinate in the same axis. Then we search for the major axis  $a$  as follows:

$$a = \begin{cases} \{x\}, & \|\mathbf{e}_x\| > \|\mathbf{e}_y\| > \|\mathbf{e}_z\| \\ \{y\}, & \|\mathbf{e}_y\| > \|\mathbf{e}_x\| > \|\mathbf{e}_z\| \\ \{z\}, & \|\mathbf{e}_z\| > \|\mathbf{e}_x\| > \|\mathbf{e}_y\| \end{cases} \quad (3.11)$$

Afterwards, the segmentation is applied based on the object axis vector with higher magnitude. Let  $\mathcal{B}$  be a specific boundary (top or middle or bottom region) of the object point cloud. Each 3D point  $\mathbf{P}_i$  will belong to a specific region (i.e.,  $\mathcal{R}_t$ : top;  $\mathcal{R}_m$ : middle;  $\mathcal{R}_b$ : bottom), if this point is inside of that region boundary. The boundary verification is achieved by the following steps:

$$\mathbf{P}_i \in \mathcal{R}_t : (\mathbf{P}_i^a \geq a_{max} - \mathcal{B}_t^a), \quad (3.12)$$

$$\mathbf{P}_i \in \mathcal{R}_b : (\mathbf{P}_i^a \leq a_{min} + \mathcal{B}_b^a), \quad (3.13)$$

$$\mathbf{P}_i \in \mathcal{R}_m : (a_{min} + \mathcal{B}_b^a) < \mathbf{P}_i^a < (a_{max} - \mathcal{B}_t^a), \quad (3.14)$$

where  $\mathbf{P}_i$  is a point that belongs to the object point cloud  $\mathbb{P}$ ,  $i = \{1, \dots, n\}$ ;  $\mathbf{P}_i^a$  is the point's coordinate in the major axis  $a$ ;  $a_{max}$  and  $a_{min}$  are the points in the major axis at the maximum and minimum coordinate, respectively;  $\mathcal{B}_t^a$  is the boundary for  $\mathcal{R}_t$  in  $a$ ;  $\mathcal{B}_b^a$  is the boundary for  $\mathcal{R}_b$  in  $a$ ;  $\mathcal{B}_m^a$  is the boundary for  $\mathcal{R}_m$  in  $a$ . The region boundaries can be found as follows:

$$\mathcal{B}_t = \begin{cases} \frac{|a_{max} - a_{min}|}{4}, & |a_{max} - a_{min}| \geq 7cm \\ \frac{|a_{max} - a_{min}|}{2}, & 5cm \leq |a_{max} - a_{min}| < 7cm \\ |a_{max} - a_{min}|, & |a_{max} - a_{min}| < 5cm \end{cases} \quad (3.15)$$

$$\mathcal{B}_m = \frac{|a_{max} - a_{min}|}{2}, \quad (3.16)$$

$$\mathcal{B}_b = \mathcal{B}_t, \quad (3.17)$$

where the boundaries  $\mathcal{B}_m$  and  $\mathcal{B}_b$  are used only if the object size is bigger than a determined threshold  $\delta$ . Algorithm 2 shows a summary version for object segmentation.

Figure 3.4 presents the results achieved for some everyday objects to validate the segmentation method. The objects point clouds on the left were achieved by in-hand exploration (as explained in Chapter 2) and on the right, by laser scanner.

Figure 3.5 presents two different cases of segmentation. For the mug object, the two scanning methods had two distinct results. This will happen for objects that do not have a clear major axis, and have similar dimensions along two axes. For these cases, the sensors and noise characteristics can have a bigger impact, and some smoothing pre-processing might be required. The segmentation method for the mug acquired by the laser scanner indicated the y direction as the the major axis. Thereby, in this situation, we changed the parameter to segment the mug into two components instead of three, which has resulted a more coherent segmentation. Within this situation, the mug's handle is separated from the body, unlike segmenting it into three components that would result the mug's body divided into two parts.

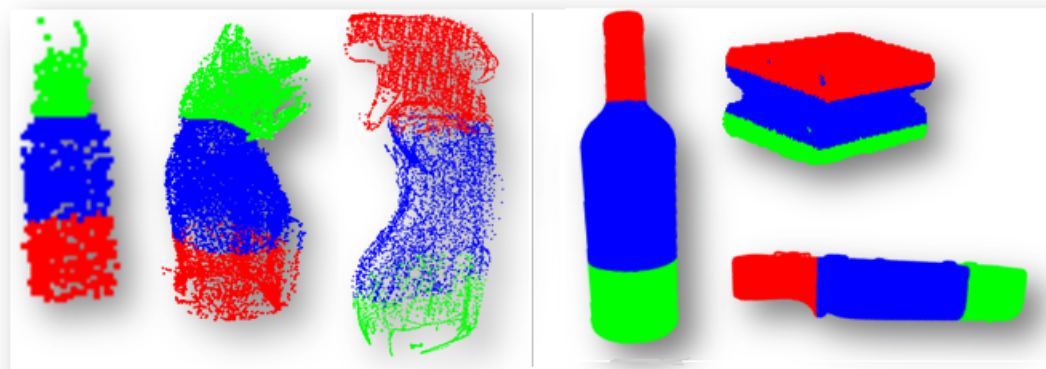


Figure 3.4: Results for the daily objects segmentation based on major axis. On left side is depicted the models acquired by in-hand exploration and on right side the models acquired by laser scanner.

**Algorithm 2:** Object Components Segmentation Algorithm based on Major Axis Analysis

- 1 Input: Object Point Cloud =  $\mathbb{P}$ ;
- 2  $\forall \mathbf{P}(x, y, z, r, g, b) = \mathbf{P}_i \in \mathbb{P}$ , search for the points with maximum and minimum coordinates values in each axis  $\{x, y, z\}$ , and build the vectors:  
 $\mathbf{e}_x = \{x_{max}, x_{min}\}$ ,  $\mathbf{e}_y = \{y_{max}, y_{min}\}$ ,  $\mathbf{e}_z = \{z_{max}, z_{min}\}$  ;
- 3  $\forall \mathbf{e}$  compute their magnitudes ( $\|\mathbf{e}_x\|$ ,  $\|\mathbf{e}_y\|$ ,  $\|\mathbf{e}_z\|$ );
- 4 Compare the magnitudes of the vectors  $\mathbf{e}_x$ ,  $\mathbf{e}_y$ ,  $\mathbf{e}_z$  and keep the the biggest one considering its reference  $x$  or  $y$  or  $z$  as the major axis  $a$  ;
- 5 Verify the object size (in distance, e.g. *cm*) in the major axis, and search the regions Top:  $\mathcal{R}_t$ ; Middle:  $\mathcal{R}_m$  and Bottom:  $\mathcal{R}_b$ , by computing the boundaries  $\mathcal{B}_t$ ,  $\mathcal{B}_m$ ,  $\mathcal{B}_b$  as demonstrated in eq. (3.15)-(3.17).
- 6 Segment the object by labelling the points in red if  $\mathbf{P}_i \in \mathcal{R}_t$ , or in blue if  $\mathbf{P}_i \in \mathcal{R}_m$  or in green if  $\mathbf{P}_i \in \mathcal{R}_b$  by computing eq. (3.12)-(3.14).
- 7 Outputs: Object Segments (top, middle and bottom) =  $\mathbb{P}_{top}, \mathbb{P}_{mid}, \mathbb{P}_{bot}$

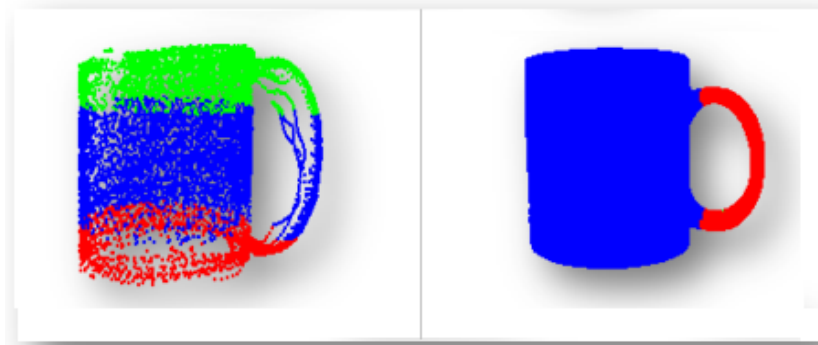


Figure 3.5: Difference in the segmentation of a mug acquired by different sensors. Left image presents a mug acquired by in-hand exploration segmented into 3 components. Right image presents a mug acquired by laser scanner segmented into 2 components.

The segmentation of everyday objects into three components can describe different candidate regions for grasping. Searching for optimal or plausible contact points on the entire geometry of the object (e.g., searching on the mesh) is time consuming. For real applications on a robotic hand, the mesh would first have to be computed, followed by additional computations of stable grasping regions. In our case, by segmenting the objects into three components, we can approximate each segment by a geometrical primitive (e.g., quadrics) allowing an association with previously observed candidate grasps for each geometrical primitive.

We have an implicit assumption that the grasps are adaptive (i.e., synergies of the fingers). This means that an approximated grasp type is adjusted, thus, avoiding the association of an exact object's geometry with an exact grasp geometry. Then, our strategy of object decomposition is sufficient

and with high possibility of grasping success for everyday objects, which justifies our segmentation method.

### 3.3 Object Components: Primitives Detection

This section presents two methods to approximate the object shape into primitives. The first method is based on learning and classification of geometrical shapes and the second is the well known method of superquadrics modelling. The first method depends on a learning process of possible primitives, but is still an effective approach of shape retrieval that can be acquired by a probabilistic classification using Bayesian techniques. This approach is limited to a set of learned shapes. Here we are using basic geometrical primitives, such as cylinder, plan and sphere to test the method. The second method (superquadrics modelling) is used for more complex shapes that can be employed to fit a point cloud into various shapes in a robust way. The second method is useful since the high number of shape possibilities and the fitting step is on-the-fly which does not need a learning process.

#### 3.3.1 Components Modelling using Basic Primitives: A Probabilistic Approach

The data acquired from in-hand exploration is used to match the data with some basic geometrical primitives, such as sphere, cylinder and plane. After the segmentation process, for example, using the clustering segmentation, the object components can be approximated within basic geometrical primitives. Using a probabilistic approach we are able to learn the basic primitives. Given the 3D points of basic primitives, features are computed from the covariance matrix, extracting then three eigenvalues to characterize each one. For that, a normalization of these values are computed as follows:

$$e_i = \frac{\lambda_i}{\lambda_{max}}, \quad (3.18)$$

where  $e$  represents the normalized eigenvalue;  $i$  represents an index for all eigenvalues found for each type of primitive and  $\lambda_{max}$  is the resulting sum between the three eigenvalues found for each primitive type. After this normalization, the maximum and minimum new eigenvalues of each type of primitive are used in the learning phase.

For the learning phase, 20.000 random synthetic primitives for each type of primitive were generated using the Matlab environment, inserting some Gaussian noise on the primitives to perform the



learning phase based on histogram techniques. In the learning phase, a histogram for each primitive accumulates all maximum and minimum eigenvalues corresponding to each primitive (Figures 3.6, 3.7, 3.8).

To compute a bivariate histogram, a matrix of dimension  $10 \times 10$  was created. For each observation, the normalized eigenvalues ( $e_{max}$  and  $e_{min}$ ) are used and they correspond to the  $x$  and  $y$  axes. After analysing all observation for each type of primitive, three histograms are generated. Each histogram is then normalized representing the learned distribution to be used as likelihood during the classification. Given a set of observation to represent a class of basic primitive  $s$ , we have the probability of each pair of feature,  $e_{max}$  and  $e_{min}$  to represent each primitive, so that we have  $P(E = \{e_{max}, e_{min}\} | s)$ . To understand the general classification model some definitions are given as follows:

- $s$  is a known shape from all possible  $S$  (e.g., cylinder, sphere and plane);
- $e_{max}$  is a certain value of feature, representing the maximum normalized eigenvalue;
- $e_{min}$  is a certain value of feature, representing the minimum normalized eigenvalue.

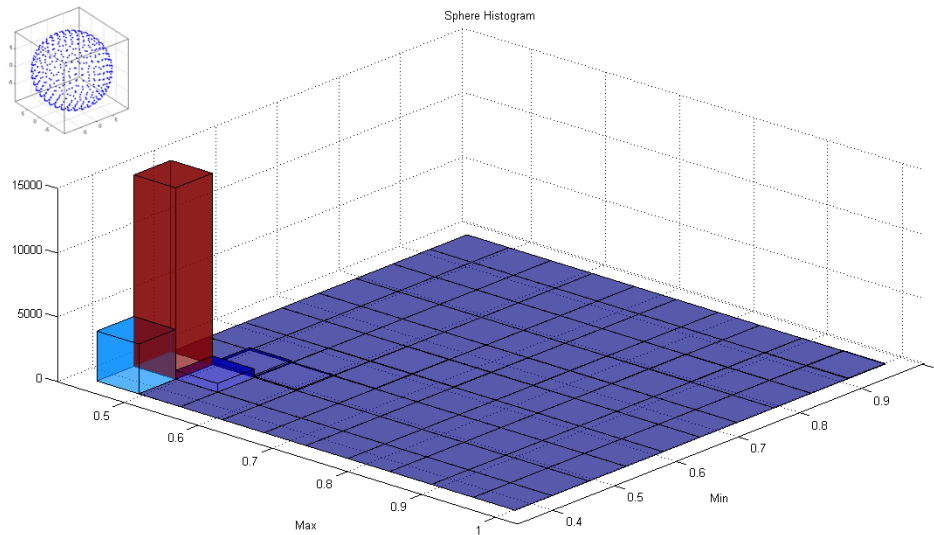


Figure 3.6: Learned histogram: sphere primitive.

Learning the probability distribution  $P(E = \{e_{max}, e_{min}\} | s)$  for each known primitive and knowing the priors (uniform distribution), Bayes' rule can be applied for the classification as follows:

$$P(s | E = \{e_{max}, e_{min}\}) = \frac{P(E = \{e_{max}, e_{min}\} | s)P(s)}{\sum_j P(E = \{e_{max}, e_{min}\} | s_j)P(s_j)}. \quad (3.19)$$

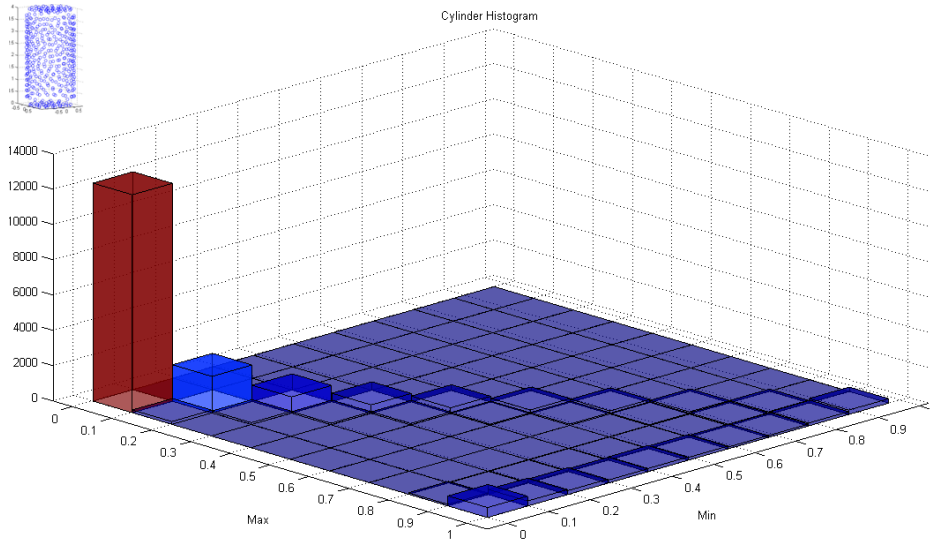


Figure 3.7: Learned histogram: cylinder primitive.

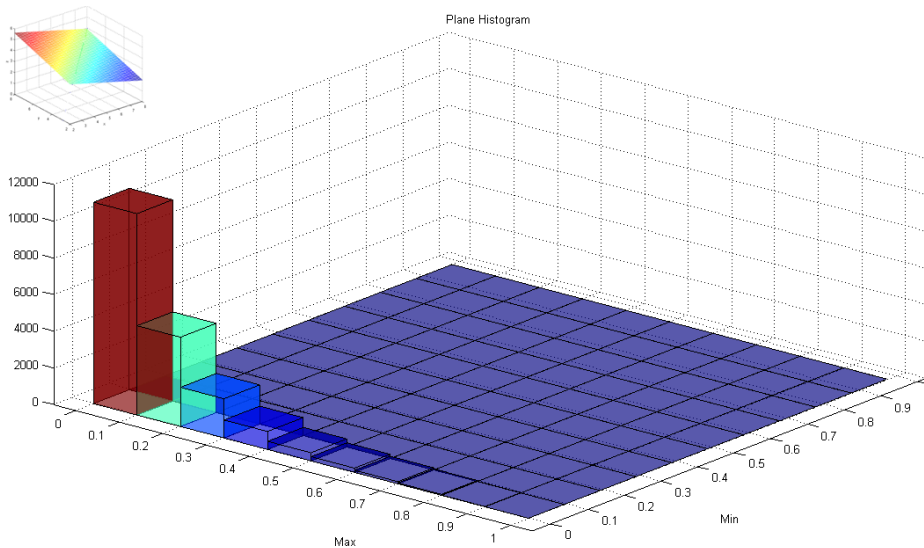


Figure 3.8: Learned histogram: plane primitive.

After the classification of each primitive (retrieving the correct shape given a point cloud), the pose of each primitive is needed, so that we compute the rotation and scale of each shape. For that, we use the algorithm proposed by [NDJR<sup>+</sup>09] that is used to retrieve shapes for novelty detection in robotic maps. It finds the shape that better approximates to an ideal basic shape from  $\Psi_{shape}$ . They use the mathematical space of the Gaussian mixture model which is described by the covariance and mean of the Gaussian functions. The Gaussian mixture associated with the 3D points is denoted as  $\Pi$ . The shape retrieval algorithm is based on the covariance matrices matching. The best model of the shape and the rigid transformation  $T$  with respect to an ideal shape is the main idea of the algorithm.

Gaussian functions are matched with each basic shape which measures the similarity between their covariance matrices,  $d_\Psi = \{d_{sphere}, d_{cylinder}, d_{plane}\}$ . The minimum value of  $d_\Psi$  determines the shape that best approximates to the point cloud, just as a rigid transformation.

Covariance matching is a basic task in measurement design [FM99]. The main goal is to obtain a distance measurement of two covariance matrices. The space of covariance matrices is not a vector space and therefore a standard arithmetic difference does not measure the difference between them. But covariance matrices are symmetric and positive semi-definite and then they can be formulated using a distance based on Riemannian metric. They use the distance measure described by Foerstner and Moonen [FM99] which is defined as follows:

$$d(\Sigma_1, \Sigma_2) = \sqrt{\sum_{i=1}^N \ln^2 \lambda_i(\Sigma_1, \Sigma_2)} \quad (3.20)$$

where  $\Sigma_1$  and  $\Sigma_2$  are the two input covariance matrices,  $\lambda$  represents the generalized eigenvalues of  $\Sigma_1$  and  $\Sigma_2$ , and  $N$  is the dimensionality of the matrices. Considering  $\Sigma_1$  as the covariance of the Gaussian function which identify a shape to be recognized and  $\Sigma_2$  as the covariance of a basic primitive. To consider possible rotations and scaling changes of the model, it must be noted that:

$$\Sigma_i = T \Sigma_j T^T = (R \cdot L) \Sigma_j (R \cdot L)^T \quad (3.21)$$

where  $T$  represents the rigid transformation applied to the ideal geometric primitive, which is composed of scale and rotation matrices,  $R$  and  $L$ . In this approach, the translation is directly known with the mean information of each Gaussian and rotation is the known rotation matrix with three degrees of freedom using Euler angles. The matrix  $L$  represents the diagonal matrix with scale for each axis. The equation (3.20) can be minimized using a least squares minimization method based on Levenberg-Marquardt algorithm, which modifies the rotation and scaling matrices in each iteration. A starting guess of the parameters is required to reduce the number of iterations needed to converge and remove local minima situations. The algorithm uses a good approximation to the rigid transformation  $T$  according to the eigenvectors values of the two covariance matrices.

This mentioned method could be used to match the shape beyond of only rotation and scale. However, we are using this method just to match the rotation and scale of each classified shape due the probabilistic classification by the Bayesian model reaching sufficient and attractive results for simple and few types of shape primitives.

After some trials with 3000 cases (1000 randomly for each shape), the classification model

reached a satisfactory result of 97.53%. The small percentage of false negative occurred with the cylinder and sphere primitives. The problem found in the classification was some confusion with the cylinder and sphere when the cylinder diameter was similar to its height so that it was classified as a sphere. Sometimes the sphere was classified as a cylinder due to the noise, it makes the radius varying more than the tolerable. Another case was the cylinder being classified as a plane, it happens when the cylinder height is close to zero, that is, when the radius is much bigger than the height.

An example of the shape retrieval regarding scale and orientation is presented in Figure 3.9.

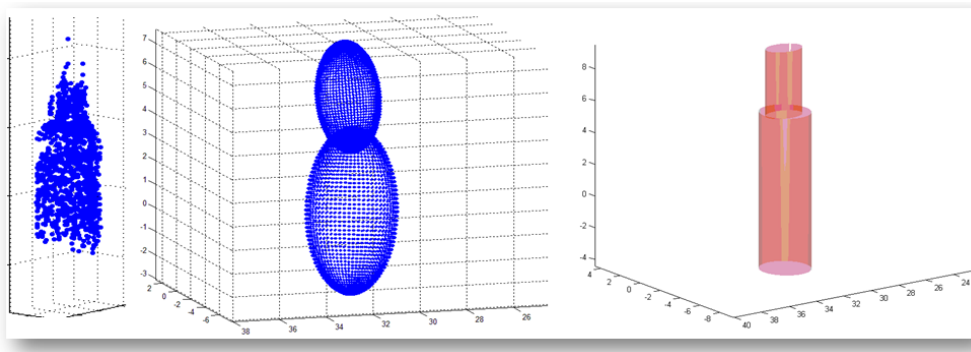


Figure 3.9: Shape retrieval: left image shows the 3D points representing the object (bottle); middle image the GMM applied as segmentation; and the last image shows the recovered primitives for each of the object component.

This method has been tested due to the eigenvalues derived from the covariance matrices (GMM density functions) may denote the variance of the samples representing the shape primitives. We assume the maximum and minimum eigenvalues as features, because they are able to distinguish the basic geometric shapes through the Cartesian coordinates of a point cloud in such a way that after a learning phase, they can preserve a class separability criterion based on the scatter of the samples.

The results showed us that even using simulated shapes generated randomly with noise for the learning phase, we reached good classifications with real and synthetic data. However, this approach is still limited to those mentioned basic primitives. If the shapes primitives possibilities are increased, a new learning process for each primitive has to be done. This does not happen with the superquadrics fitting method, because it adopts an iterative change of shape parameters allowing different shapes possibilities without a learn phase for each new primitive. To deal with unknown objects with more complex shapes, superquadrics (subsection 3.3.2) has proven to be a good shape approximation method as it will be explained in the next subsection.

### 3.3.2 Object Component Modelling using Superquadrics

Having segmented the object, we now want to model each segment as a geometric simple shape or primitive. In this work we use superquadrics [Bar81], a technique that models a rich variety of shapes (e.g., Figure 3.10), and that facilitates computing parameters that enclose important cues, such as scale and orientation. Superquadrics has been used for 3D object modelling [SLM94] and for segmentation of point cloud [CJB03], in robotics (novelty detection) [JNR<sup>+</sup>10] and successfully in other works for grasping purposes [MKCA03] [EKSP07] [BV07].

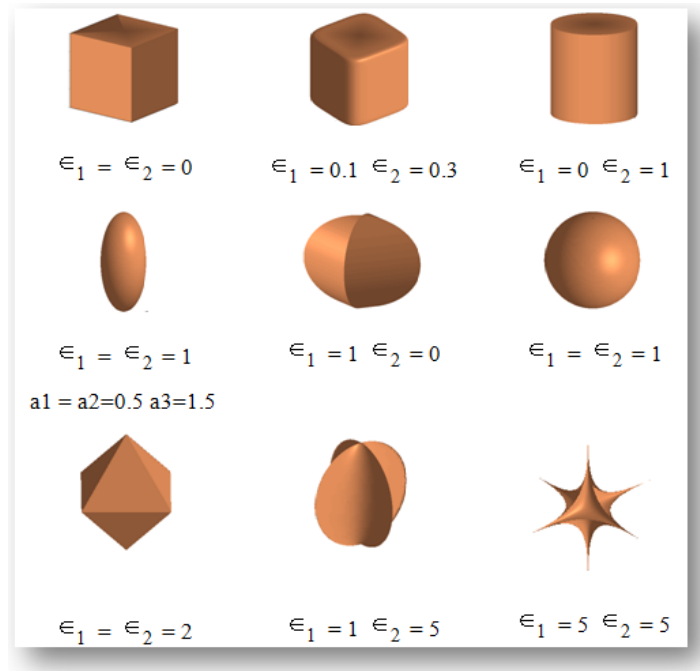


Figure 3.10: Typical Superquadrics:  $a_1 = a_2 = a_3 = 1$ , except ellipsoid.

The superquadrics models are expressed by a function  $f : \mathbb{R}^3 \rightarrow \mathbb{R}$  as:

$$f(x, y, z) = \left( \left( \frac{x}{a_1} \right)^{\frac{2}{\epsilon_2}} + \frac{y^2}{a_2^2} \right)^{\frac{\epsilon_2}{\epsilon_1}} + \left( \frac{z}{a_3} \right)^{\frac{2}{\epsilon_1}}, \quad (3.22)$$

where  $\epsilon_1$  and  $\epsilon_2$  are the parameters for shape;  $a_1, a_2$  and  $a_3$  are the scale factors on the  $\{x, y, z\}$  axes. This form provides information on the position of a 3D point relative to the superquadric surface. The implicit function  $f(x, y, z)$  partitions the space into three regions: the point  $\mathbf{P}(x, y, z)$  lies on the surface if  $f(x, y, z) = 1$ , if  $f(x, y, z) < 1$  then the point is inside, and outside when  $f(x, y, z) > 1$ . Even if the five parameters of the model are compact, it allows to deal with a large variety of shapes such as cylinders, spheres, ellipsoids, parallelepipeds and others. The shape parameters can be constrained

to have, for example, just convex shapes (when  $\epsilon_1 < 2$  and  $\epsilon_2 < 2$ ).

The recovery of the superquadrics from a point cloud is represented in a global coordinate system. Thus, we have another 6 parameters to express the rotation (Euler angles  $(\phi, \theta, \psi)$ ) and translation  $(p_x, p_y, p_z)$ . The function can also be expressed as  $f(x, y, z, \Lambda)$ , where the set of the 11 parameters can be represented as  $\Lambda = \{a_1, a_2, a_3, \epsilon_1, \epsilon_2, p_x, p_y, p_z, \phi, \theta, \psi\}$ , representing three parameters for scale in each axis, two parameters for shape variation; and six parameters representing the translation and rotation in each axis, respectively.

After the segmentation process, the set points of each object component will be approximated by a superquadric shape primitive. To estimate the parameters of the superquadric model, the gradient least-square minimization of an error-of-fit function based on Levenberg-Marquardt method [JLS00] is used as follows:

$$\min_{\Lambda} \sum_{i=1}^n (\sqrt{a_1 a_2 a_3} (f^{\epsilon_1}(x_i, y_i, z_i; \Lambda) - 1))^2, \quad (3.23)$$

where  $\sqrt{a_1 a_2 a_3}$  are constraints used to find the smallest superquadric based on the scale parameters. The power  $\epsilon_1$  makes the error metric independent of the superquadric shape. More details can be found in [JLS00].

An important aspect for the success of the superquadrics fitting is on the initialization of the method. This can influence the fitting which concerns the number of iteration, which determines the local minimum for the convergence of the method. Thus, we are using an initial pose based on the matrix  $M$  that represents the center of gravity and the central moments computed from the input data. The shape parameters are initialized as an ellipsoid,  $\epsilon_1 = \epsilon_2 = 1$ . The scale factors are based on the computed eigenvalues of the inertia matrix  $M$ .

For each object component (segmented region), a superquadric model is generated. Figure 3.11 shows the superquadrics models generated for the segments of some everyday objects, simplifying the object parts shape into geometrical primitives.

More parameters can be computed to represent a superquadric model, the 11 parameters already explained, and a further 4 parameters that can be included in this set of parameters, such as the centroid of the superquadric  $\{c_x, c_y, c_z\}$  and its volume  $v$ .

We have limited the set of geometrical primitives that can compose everyday objects into the following superquadrics models: box and its variation (rounded box), cube, cuboid, cylinder, ellipsoid, sphere, octahedron, spinning-top (squared, rounded and star shape) and variation 1 of the

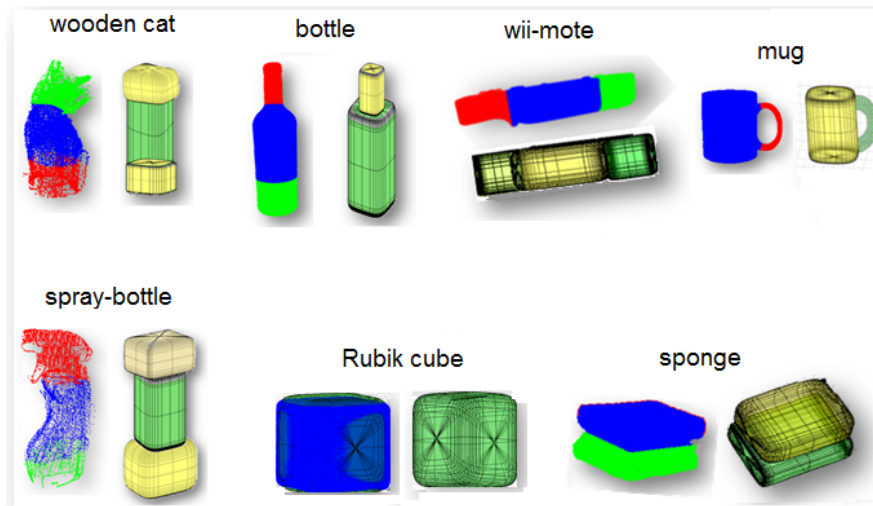


Figure 3.11: Superquadrics models obtained for some segmented everyday objects.

sphere (arch) and variation 2 of the sphere (butterfly shape). This was done because these thirteen superquadrics models are sufficient to approximate and describe a big variety of everyday objects for grasping.

The superquadrics fitting results and processing time depend on the point cloud size. The superquadrics models can represent an object shape even when the input is a partial volume of the object. For the everyday objects used in this work, we have achieved results that were satisfactory and useful for the subsequent grasping synthesis. We can achieve a processing time for a point cloud acquired from an RGB-D sensor of under a second on average, using a standard computer (e.g., a Laptop with intel core i3-3500M processor, 2.26GHz, 4GB DDR3 Memory).

This process of decomposition (segmentation and object component modelling) will be used during the learning phase as well as during the grasp synthesis when the artificial system faces an unknown object for grasping as described in Chapter 6. These processes will assist the grasp synthesis to choose the proper grasp for the decomposed part of the object. The method matches new observed geometric primitives with the nearest previously recorded primitives, and uses the corresponding observed human grasps to have a small set of candidate grasps.

### 3.4 Discussion

In this research we are dealing with everyday objects proper for grasping applications. The objects that are used here satisfy some requirements, such as: big enough to be detectable by the sensor, small enough to be held fully in the hand (even power grasp), afford several grasp configurations, complex

enough not to be easily manipulated by a simple gripper, simple enough to be able perform actions with the hand, simple in shape, existing in several slightly different shapes, possible to match to a similar or familiar object in case of an unknown for the robot.

Both segmentation can be applied for everyday objects. The choice of the method depends on the need of the application, for instance, if we are dealing with time processing, basic shapes, number of points composing the object, etc. A question can be raised here, when and why is it necessary to segment an object? The answer is simple, to approximate regions on the object in simple shapes to find suitable grasps.

Using the segmentation method based on the major axis we could achieve satisfactory results for everyday objects. This method is fast even for an object point cloud with a large number of points, unlike other methods that are more time consuming, which gives a clear advantage of the method based on the major axis. We can also verify the success of this method by a qualitative analysis (comparing with human choice, asking how a subject would segment that specific object).

Although the segmentation method based on the main axis uses a fixed number of components, with regions already set (top, middle and bottom), we could observe that grasp types can be associated with these regions, thereby allowing the robotic hand to be able to grasp one of these regions. For everyday objects, this algorithm proved to be efficient. However, for more complex objects, lets say objects that have more concave or convex regions, or even with many handles, the method can also work, as long as we correctly define the boundaries for the segmentation.

Adopting this method based on the major axis, we will always have at least a segment that can represent a feasible candidate region for grasping, and consecutively, we will find the candidate grasps for that region, which validates our method as sufficient and feasible for grasping purposes.

This method is quite fast even for a large number of points (about 10msec), unlike GMM that for the same set acquired from the same sensor takes much longer (about 5 seconds). The GMM method is automatic in the selection of the clusters (object regions), searching for the best fitting given the object point cloud. However, the parameters estimation can be slow due to the number of iterations necessary when using a large number of points, making the algorithm convergence slower.

After the object segmentation, methods for shape approximation are evaluated. In a first approach, features (eigenvalues) are detected to represent basic primitives such as sphere, cylinder and plane. Acquiring the features of each shape, a learning phase was performed based on histogram techniques that represents the likelihood in the Bayesian model for shape classification. This first ap-



proach of shape classification is simple and efficient when we have a limited number of geometrical primitives. In the second approach, the components segmented are used to model each component into superquadrics models, which has the advantage over the first approach of the number of possible shapes just changing the parameters of the superquadrics, without requiring a learning step for each possible primitive. The concern of the superquadrics modelling is the processing time when a large number of points are used during the fitting process to find the best option between the superquadrics shapes possibilities. However, for everyday objects we achieve an acceptable processing time (on average varying between 2 and 3 seconds) when dealing with a point cloud with approximately 60000 points (for each component).

The object representation (segmentation and shape approximation) is a good strategy to generate candidate grasps. When the shape is associated for each segment of the object, we are recovering the object pose parameters  $\{x, y, z, yaw, pitch, roll\}$ . This process of representation enables us to decrease the huge amount of possible grasps into a set of candidate grasps for a specific geometrical primitive. The next chapters will better explain the adopted strategy to generate candidate grasps for a specific object component.

The publications related to this chapter's subject, object decomposition and components modelling, are listed as follows:

#### **Journals**

- Diego R. Faria, Ricardo Martins, Jorge Lobo, Jorge Dias. "Extracting Data from Human Manipulation of Objects Towards Improving Autonomous Robotic Grasping". *Robotics and Autonomous Systems*, Elsevier, Volume 60, Issue 3, March 2012, Pages 396-410, 2012.
- Diego R. Faria, Pedro Trindade, Jorge Lobo, Jorge Dias. "Knowledge-based Reasoning from Human Grasp Demonstrations for Robot Grasp Synthesis". **Under Review**: *Robotics and Autonomous Systems*, Elsevier, 2013.

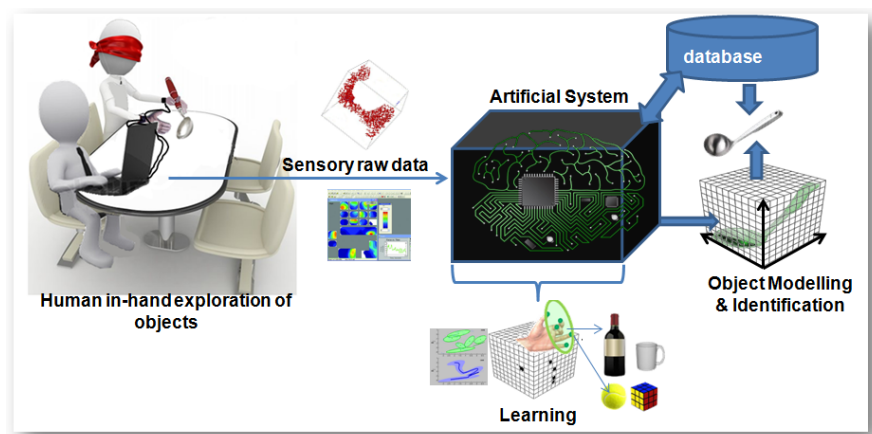
#### **International Conference**

- Diego R. Faria, Jose Prado, Paulo Drews Jr., Jorge Dias. "Object Shape Retrieval through Grasping Exploration". In the 4th European Conference on Mobile Robots, ECMR'09, Mlini/-Dubrovnik, Croatia, September 2009, pp.43-48.



## Chapter 4

# Identifying Objects from Hand Configurations



### 4.1 Introduction

Several studies have been carried out in neuroscience/psychology to better understand human perception [LK87] [KL90]. Haptic perception is an important mechanism in which humans get knowledge about the properties of unknown objects. Humans have the ability to manipulate different objects dexterously using different exploratory movements for object haptic perception, such as contour following to extract the global shape of the object, lateral motion to perceive the texture, pressure movement to extract the softness characteristics of an object, static contact to perceive the temperature, enclosure (e.g., grabbing a glass by side power-grasp), and unsupported holding to perceive the object weight [LK87] [KL90].

Artificial perception systems are required by robotic systems to navigate and interact with the surrounding environment and persons. Human perception is studied and pursued by researchers in the robotic field to endow a robot with this ability. Typically, the planning of a robotic object identification and recognition tasks start by estimating an initial model of the object by obtaining data through the vision systems. Other approaches are dedicated to the estimation of the superficial characteristics of the object such as texture and stickiness [OCLBC11], others to find objects global shapes by robotic hand [BWB<sup>+</sup>07]. This chapter will discuss the second group of approaches for identification using a probabilistic approach.

In [LSN<sup>+</sup>11] is proposed an algorithm for surfaces frictional properties estimation to classify objects, while the surface is explored by a robotic finger equipped with a force/torque sensor. In [OC01] is proposed a method to identify different types (cusp, step, bump) of superficial features during the lateral sliding of a robotic finger. Some other works focus their analysis of the perceived object representation on estimating the global shape of the object and finding suitable object regions for stable grasping. In [CSPB11] only tactile information is used to evaluate the deformation signature of several objects in order to discriminate the internal state (empty, full, open, close) and identity of those objects. [NGW10] presents an approach for haptic object recognition using an anthropomorphic robot hand which identifies objects from the haptic sensor data acquired by palpation sequences. Given the sequence of sensor data, features are extracted from tactile patterns to describe the object by key features. Here, the work differentiates, going further from these previous related works by not only using the finger movements or tactile information to acquire the object shape, but we are learning and identifying possible hand configurations that can be associated with object shapes to select a strong candidate from a hypothesis of object identities.

Canonical grasps from human demonstrations is presented by [dGSF06]. The authors proposed an approach to learn grasp affordances by modelling the hand pose by mixture distributions. The main objective was to learn the reach-to-grasp actions to set the proper affordance for an object. Human hand actions representations for programming grasping actions is the goal of the approach presented by [RFKK10]. A hand posture space is represented by a low dimensional space. Gaussian Process Latent Variable Models (GPLVMs) were used to model the lower dimensional manifold of human hand motions during object grasping which is useful for grasping actions modelling, mapping and recognition. Based on these mentioned works, the proposed work here uses also mixture distributions to model the hand configuration for a specific object to find similarities between a hand pose and

a object shape to achieve some hypotheses of object identity when the object is being explored. Modelling of actions or tasks performed with the object is not taken into consideration.

The aim of the work described in this chapter is to use cues from the hand kinaesthetic sensory, distal fingers segments positions and movements, for retrieving object intrinsic information as well as to find the object identity. By adopting a probabilistic representation model of the object (probabilistic volumetric map) and contact points on the object surface generated during in-hand exploration, some characteristics of object shape associated with the hand configuration can be learned. These characteristics are acquired by observing examples of some possible grasps for specific objects (by using the contact points). The learning of hand configurations (grasps taxonomies) associated for specific everyday objects is achieved through mixture distribution-based representation. Different contact points associated with an object shape can be represented in a latent space and lie on a lower dimensional non-linear manifold in the contact points space which is suitable for modelling and recognition. From the multiples clusters are generated, each has an important representation of possible hand configurations for a specific object. Using the mixture distributions, a signature is extracted by Gaussian Mixture Regression (GMR) to represent a candidate object given the hand configuration during the in-hand exploration of objects. Acquiring a compact representation that describes and associates hand configurations to candidate object shapes, improves the hypothesis belief for object identification. Knowing the functionality of how object shapes are grasped is meaningful in terms of description, so that, when an object is being explored, by using the hand configurations on the object surface, we can identify candidate objects based on these inputs.

In order to achieve our goals and to propose more efficient and solid ways of object identification using dexterous manipulation through kinaesthetic stimuli, we are adopting probabilistic methods. The benefit of the proposed approach is the belief acquired to search for a strong candidate from the possible hypotheses for the object identification.

## 4.2 Learning Hand Configurations for Everyday Objects

An object's geometry plays an important role in robotic applications, where its representation is also valuable for identification into a class of known objects and also to search for regions on the object surface proper for a stable grasp. In this research, the object is associated with possible hand configurations (i.e., valid grasp types for this object). Differently of chapter 2, the probabilistic volumetric map is used here to represent the locations (occupied voxels in the grid) of the contact points that

are also partial volume of the object surface during the in-hand exploration to assist in the object identification.

The key of this work is, whilst a subject is manipulating an object by means of kinaesthetic sensory modality, the artificial system generates and updates candidate object identities for the presented contact points. For that, a learning phase is performed to associate possible taxonomies of grasp types (i.e., hand configurations formed by contact points) to object shapes. Previous information of these grasps taxonomies are demonstrated by human individuals. The next subsection presents the strategy of the demonstration.

### 4.2.1 Human Demonstrations

A study in which several grasp taxonomies were analysed (robotics, biomechanics and medicine) has been carried out by [FPS<sup>+</sup>09] and then some grasp taxonomies were evaluated. These taxonomies were developed within the European project GRASP [GRA]. Based on the taxonomies proposed in that work, we are considering some of the taxonomies in this study to associate some hand configurations with object shapes.

Humans demonstrators (five male right handed individuals) participated to provide examples of some grasp taxonomies for some objects. The intention was to build a knowledge repository of contact points (fixed/static hand configuration) for some specific objects. Each individual has attached six Polhemus [Pol] magnetic sensors to the hand, one on each fingertip to record the 6DoF (position and orientation)  $\{x, y, z, yaw, pitch, roll\}$  of each sensor and another in the wrist to compute the relative position of each fingertip with respect to the wrist. The pose of the hand is defined as the fingers position relative to the palm. Each set of contact points are then represented in an 18 dimensionality space (6 sensors, each one  $\in \mathbb{R}^3$ ).

Tactile sensors are also used here to assist the tracker sensors (Polhemus Liberty) in a simple way. Since the tracker sensors when active return the positional data, and in a specific task such as exploration of the object, just the contact points when the hand is static forming a hand shape on the object surface is important for this work. Thus, we ignore the tracker sensors data during the finger transitions, because it is not relevant here. The tactile sensors assist to filter the tracker sensors data, using only the data that is acquired when the tactile sensors are active. We can easily do that since our data acquisition process is distributed and with synchronized timestamps for the data. The tactile sensing device consists of 360 sensing elements (Tekscan Grip System sensor [Tek]) which are

distributed along the hand palm and fingers surface. The sensing elements are grouped into fifteen regions as presented in Figure 4.1, corresponding to different areas of the hand. Each of these regions can be defined as activation level states,  $R \in \{NotActive, LowActive, HighActive\}$ .

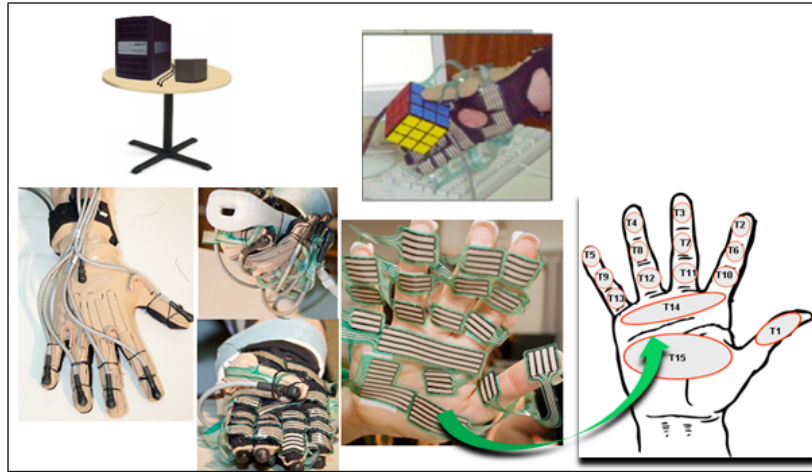


Figure 4.1: Sensors used in our experimental setup: Polhemus Liberty Magnetic Tracking System and Tekscan Tactile sensor.

Some everyday objects with simple shapes were used for the hand configurations demonstrations, such as a mug, bottle, Rubik cube, tennis ball and a ladle. We have asked for each subject to perform the in-hand exploration of the object using seven hand configurations for each object. Two different ways of recording the object were performed to be used in the learning phase. The first one used fixed grasps belonging to the taxonomy defined in [GRA] as previous mentioned, which the recording session started when the subject already had the hand configuration on the object surface and after two seconds in this fixed position the recording session was finished. In the second one, the subjects performed seven hand configurations sequentially, and the grasp transitions between one grasp to another one were also recorded. In this last one, the subject performed the demonstrations by the sequence of movements around the object and usually it takes no more than 15 seconds for each individual demonstrating the seven grasp types for each object.

#### 4.2.2 Mixtures of Contact Points Models and Signatures Extraction

The contact points space is built from the human demonstrations for everyday objects, where the contact points are the occupied cells in the object map ( $P(Z_n^c | [O_c = 1])$ ). Features in the latent space are extracted to find signatures of possible hand configurations associated with object shapes. Multiples clusters are computed given the observations using Gaussian Mixture Models (GMM) distributions.

Each specific distribution of contact points for a specific object is represented by a density function in the mixture models. Furthermore, the use of multiple density functions stores any covariance that may exist between hand configurations and objects. This work, therefore employs, the mixture distribution-based representation by means of GMM. The steps to compute the density function of the mixture  $g$  is similar to the equations presented for object segmentation by using GMM explained in Chapter 3, equations (3.1) to (3.9). Here, the difference is on the input data of the GMM model, instead of using the full object point cloud  $\mathbb{P}$ , we are using a set of contact points representing the hand configurations overlaid on the object model, defined as a matrix of points,  $\mathbb{C} \in \mathbb{R}^3$ . The Gaussian densities is previously set, where the number maximum of clusters  $k_{max} = 4$  to enclose the contact points for a specific object model. The reason for adopting 4 clusters is related to the possible grasp categories as presented in the study of grasp taxonomies [FPS<sup>+</sup>09] and also available on the website of the GRASP project [GRA]. The estimation of the parameters of each individual density function (Gaussian) and the weight variables are accomplished by using the Expectation Maximization (EM) algorithm, in the same way as previous explained in Chapter 3.

Figure 4.2 shows examples of the clustering process which is the demonstrated hand configurations associated with the objects, for later being generated a signature for each object, resulting from the hand configurations. Each cluster encloses demonstrations of one or more hand configurations (similar taxonomy) during the in-hand exploration.

The measure of similarity between the contact points is achieved by using mixture density functions. Since we have a probabilistic model through GMM in the latent space, we can extract contact points signatures (CPS) by a generalization process achieved by GMR. Then, a specific trajectory (signature) is generated based on the demonstrated hand configurations for a specific object. The GMR over a stochastic retrieval process provides a suitable way of reconstruct sequence from a Gaussian model. Researches in different fields, such as robotics and machine learning have used the statistical models (mixture distribution-based and local weighted regression) for learning, representation and generalization of data [CGB07], [CGJ96].

The main idea here is to extract from the latent space a generalized representation of hand configurations (formed by contact points) for each object, so that the closest similarity between the input and the generalized hand configuration signature can be found.



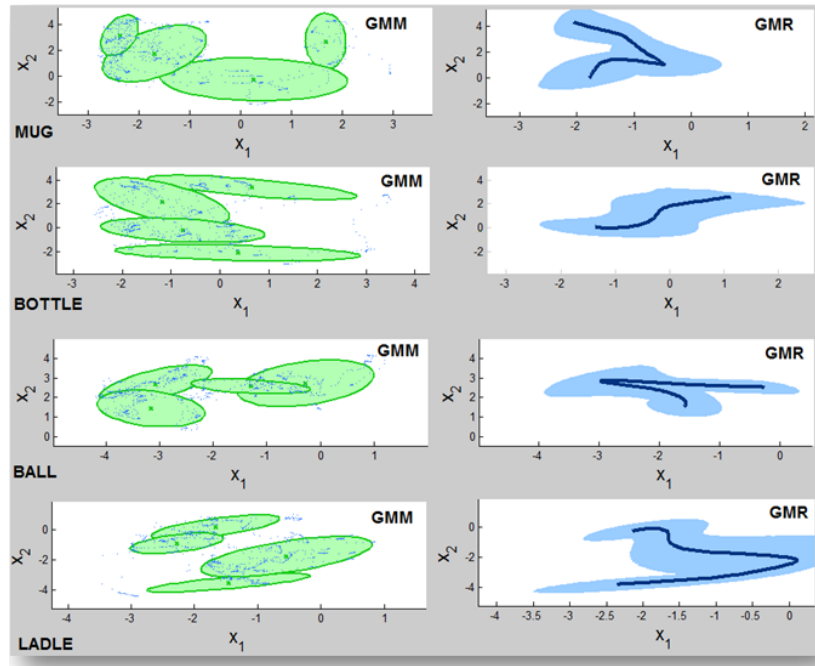


Figure 4.2: GMM and GMR process. Each cluster encloses demonstrations of hand configurations. A signature of the transition between the hand configurations is generated using the features in the latent space when is applied GMR.

### 4.2.3 Similarity Measure for the Contact Points

The similarity measure is verified by comparing how likely the new observation  $\zeta$  (set of contact points forming a hand configuration) is to the mixture distribution-based representation  $\xi$  (signature of the hand-state class  $\mathbb{C}$ ) achieved by the computation of GMM/GMR. This way, we can identify which class the demonstrated contact points belong to. Here, the class is defined as hand-object, that is, the possible hand configurations for an object shape. For that, we are basing our approach on [RFKK10], adapting it for our specific case.

From computing the probability  $P(\zeta|\xi)$  we can infer how probable is a new observation belonging to a specific signature, where a set of contact points  $\zeta$  is generated by the model  $\zeta$  being similar to the signature  $\xi$ . Since we have a probabilistic model for each  $\mathbb{C}$ , through the GMM representation achieving  $P(\xi|\mathbb{C})$ , we can compute how likely the  $\zeta$  is generated for  $\xi$  by achieving  $P(\zeta|\xi)$  enclosing the GMM parameters as follows:

$$P(\zeta|\xi) = \sum_{j=1}^k w_j^{\xi} \phi(\zeta | \mu_j^{\xi}, \Sigma_j^{\xi}), \quad (4.1)$$

$$P(\zeta|\xi) = \prod_{\forall \varsigma \in \xi} P(\varsigma|\xi), \quad (4.2)$$

then the similarity function  $\hat{s}$  is computed by averaging the two entities:

$$\hat{s}(\zeta, \xi) = \frac{P(\zeta|\xi) + P(\xi|\zeta)}{2}, \quad (4.3)$$

so that the minimum distance between the result of  $\hat{s}$  to a specific class of hand-object  $\xi$  point to a class that the new observation belongs to. It is necessary to compute (4.3) between the new observation to all possible signatures. The minimum distance is achieved by  $\min f(\hat{s})$ . The distance between the result of  $\hat{s}$  and all possible hand-object  $\xi_i$  signature is computed as follows:

$$\min_{i \in \{1, \dots, N\}} f(\hat{s}) = |\hat{s} - P(\xi_i|\zeta)|. \quad (4.4)$$

Equation (4.1) states that the probability of a hand configuration given the contact points  $\varsigma$  belonging to a hand-object class is modelled as a weighted mixture of Gaussians by computing the GMM as shown in Chapter 3. The mixture of probabilities of the contact points is generated by the model  $\xi$  as presented in (4.2). The probability of a new hand configuration being generated by a GMR model is computed following these equations above. By comparing those probabilities we can estimate which is the most likely hand-object class that generated the contact points to try to find the most probable object shape for that hand configuration.

### 4.3 Object Identification

The process of object identification starts by verifying the first demonstration of contact points forming a hand configuration whilst the object is being explored by the in-hand exploration procedure. At each hand configuration demonstrated we can search for possible object candidates identities. As long as the in-hand exploration of an object is increasing, the list of possible objects (from the database) is updated based on candidate objects with high occurrence (those objects associated with a hand configuration). The hand configuration during in-hand exploration is compared with the learned signatures by the similarity measure. Later the objects with higher occurrences are more probable than those that appeared less during the exploration. This process reduces the hypotheses of object identity not matching the partial point cloud formed by contact points with all objects in the database.

The probability distribution to find hypotheses of possible candidate objects is computed by

simply counting the number of occurrences of the objects that are listed during the exploration of the partial volume of the object shape.

The object database is composed of a set of models of everyday objects as mentioned before. Each object is represented by the 3D Cartesian coordinates in the frame of reference of the sensor that acquired the object model. The 3D object models were acquired by a 3D laser scanner (Konica Minolta Vivid 910) and also acquired with full in-hand exploration of the object. The idea of having two representations of the object is to guarantee that if not fitted to the object model with a high degree of reliability (generated by the laser scanner) we have the approximated model achieved by the in-hand exploration.

Figure 4.3 shows an example of the hypotheses generation (objects selection) for the demonstrated contact points during the in-hand exploration of an object.

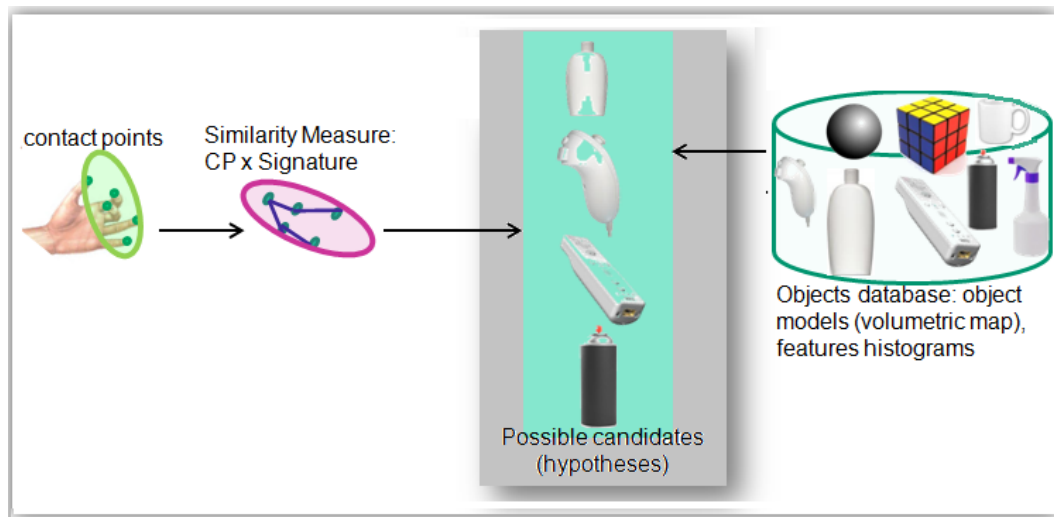


Figure 4.3: Hypotheses Generation: at each demonstrated contact points during the in-hand exploration a hypotheses list of candidate objects identities are generated from the stored objects.

In this work, the identification is an estimation process to find the most probable object. Our focus is to identify the objects given the hand configuration, so that the main contribution is from the application of the selection of objects from the database using the GMM/GMR process. For the selection process of candidate objects, we process the raw data from in-hand exploration into the wrist frame of reference to find invariance to compute the hand configurations.

To match the partial volume of the object acquired from in-hand exploration, we are using the data acquired using the probabilistic volumetric map in the object workspace. The set of contact points represent partial volume of the object surface, then the point cloud is matched against the 3D

models of the object stored in the database that was pre-selected during the hypothesis generation. The matching is done using the classical algorithm Iterative Closest Point (ICP) first introduced by [BM92]. By minimizing the difference between two clouds of points we can achieve the best match. We are computing the Root Mean Square Error (RMSE) to estimate the best matching by choosing the minimum RMSE resultant from all matching.

## 4.4 Experimental Results

A few sequences of contact points that form the hand configurations are overlaid in the full volume of the object as presented in Figure 4.4. Thus, we are using the same strategy to identify the hand configuration by the contact points. In that example, four sequences of hand configurations were detected during the in-hand exploration of object, as well as the partial volume of the object model. The results demonstrated that we can represent the global shape of the object by using the contact points, and also the hand configurations during the exploration.

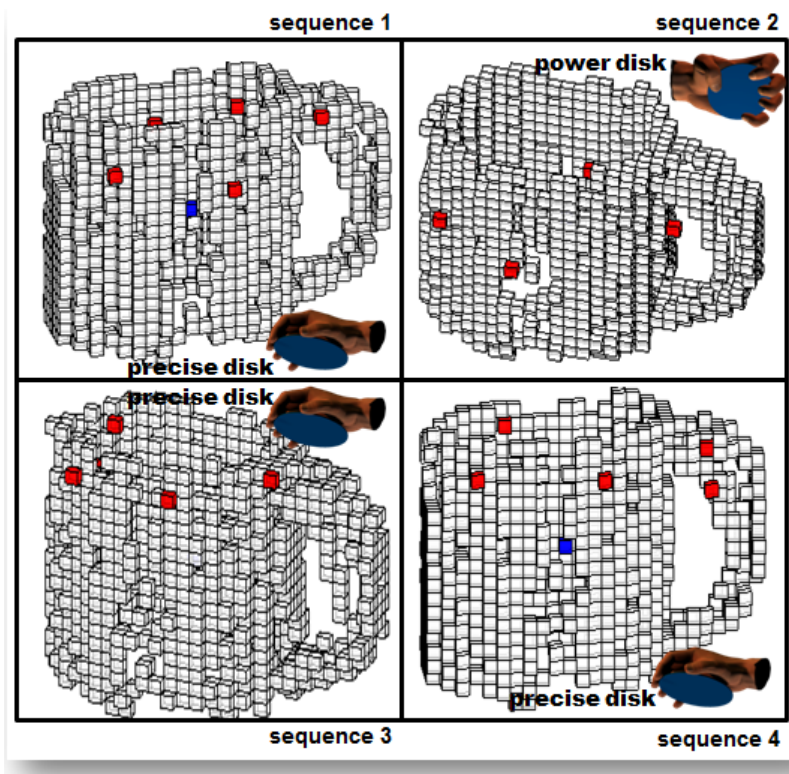


Figure 4.4: Sequences of contact points overlaid on the object surface during the demonstrations of hand design while the in-hand exploration was performed by an individual. For each sequence is identified the hand configuration given the contact points.

We state that using five sequences of hand configurations for the objects that we are dealing

with in our database, we can achieve the hypotheses to identify correctly the object that is being explored. In the example given in this work (mug), after observing some in-hand exploration performed by five subjects, the most common hand configuration detected for the mug is presented in Figure 4.5. This figure represents the result after using the similarity measure to identify possible candidate objects associated to each hand configuration. We can select more than one object for each hand configuration resultant from (4.3) and (4.4). Afterwards the probability of occurrences of all objects listed during the in-hand exploration is computed as shown in Figure 4.5. The probability distribution for each object is independent from each other.

Figure 4.6 shows the result of the matching of the new observation (partial in-hand exploration of a mug) to the pre-selected objects based on hand configurations. The gray point clouds are the 3D models stored in the database. The green color is the partial volume acquired during the in-hand exploration. The objects in the top row are the selected objects from the similarity measure using the signatures achieved during the learning phase. The bottom row presents the most probable object model (mug) acquired from laser scanner and the less probable model (ladle) acquired from the laser scanner. In the top row we can see that selected objects models are full models from in-hand exploration as well as from laser scanner (rubik cube). The object model, mug (in-hand exploration) in the red box indicates the best match between all models. The object model (mug, laser scanner) in the blue-box is the matching between all models acquired from laser scanner. The RMSE for all objects, from left to right, top to down: bottle-mug = 0.2730; mug-mug = 0.0044; mug-cube = 14.2485; mug-mug (laser scanner) = 5.5247; mug-ladle = 22.5571.

In order to have satisfactory results without mistakes, it is better to perform the matching between models acquired from the same sensors (for the stored objects in the database). This means that, if we perform a matching between an object from in-hand exploration with an object model from laser scanner, the others matching should follow the same strategy, matching between the in-hand exploration and the other models acquired from laser scanner, since we have different models acquisition for the same object.

## 4.5 Discussion

The perception acquired by human hands (haptic: kinaesthetic, cutaneous, thermal) plays an important role in human life during everyday tasks when prehensile and manipulation activities are performed to acquire objects' intrinsic and extrinsic information. Using cues from human in-hand

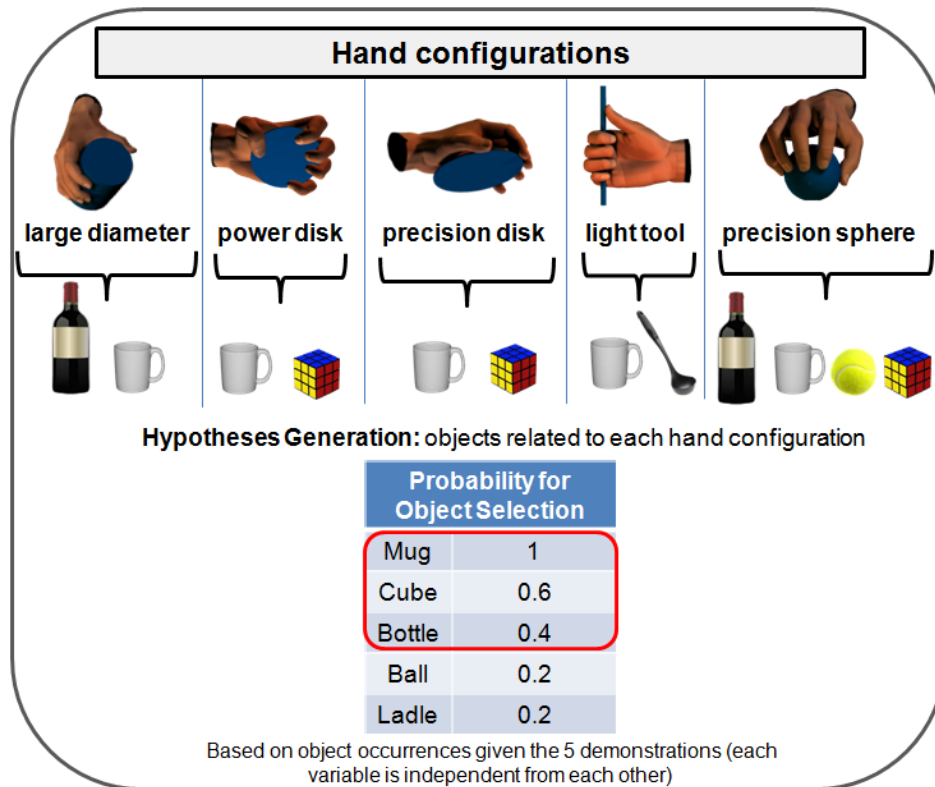


Figure 4.5: Most common hand configurations identified for the mug during the in-hand exploration. The grasps presented are following the taxonomy shown in the Human Grasping Database developed inside the GRASP project [GRA].

exploration of objects, hand configurations can be learnt and associated with object shapes to derive suitable models to identify the manipulated objects.

A probabilistic 3D grid-based method for representation of the partial volume of the object derived from in-hand exploration as well as the set of contact points on the object surface are used. By means of mixture distribution-based signatures are learned and generated using a sequence of hand configurations for a specific object. Later using a similarity measure function, we can compute the probability of a new observation being associated with object shapes. This process allows a selection of candidate object identities to reduce the amount of objects for matching. This way, we have hypotheses generation, which discards the less probable objects, keeping just the strong candidates found by the computed weights based on the probability distribution of the objects acquired when a list of candidate objects is generated and updated during the exploration. The accumulated set of contact points (partial volume of the object shape) during the object in-hand exploration is matched to those ones selected from the database (most probable candidate objects). Then the object identification is achieved by matching between point clouds (in-hand exploration of object and the selected

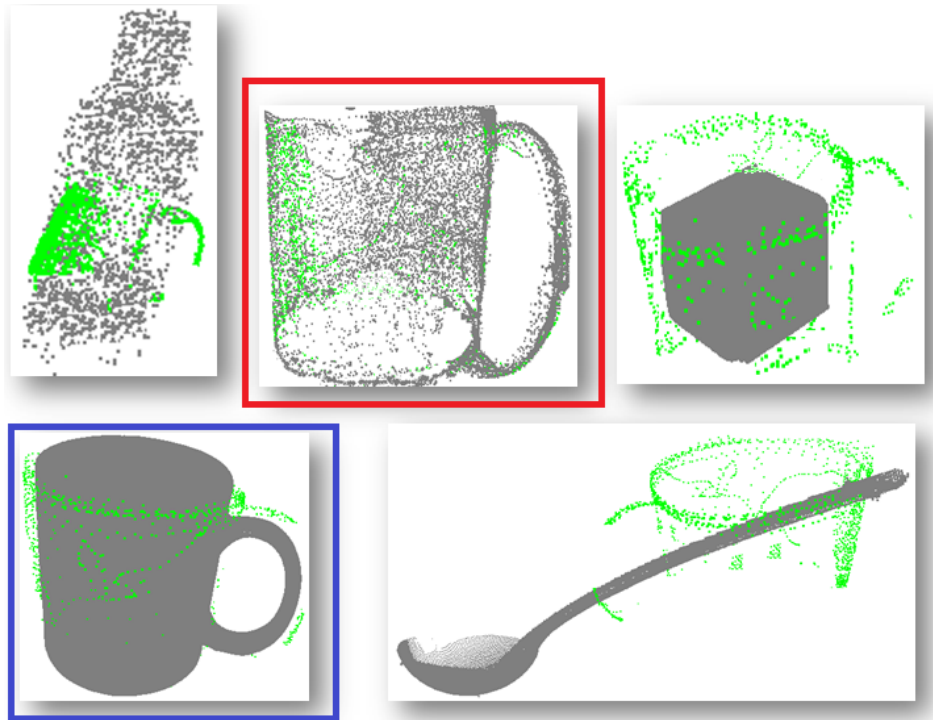


Figure 4.6: Matching between object models using ICP method. The best matching between the new observation and the object stored in the database after the selection of candidates is the object in the red box (mug),  $RMSE = 0.0044$ . The best matching between the the laser scanner models was the object in the blue box (mug),  $RMSE = 5.5247$ .

models from the database).

Results are presented for human manipulation of objects, but this can also be applied to artificial hands, although we have not addressed the hand control, only the object identification. The results also suggest that the methodology adopted has the potential and is possible to acquire satisfactory results using cues from kinaesthetic stimuli during in-hand exploration of objects.

The publication related to this chapter's subject, identifying objects by in-hand exploration, is given as follows:

#### **International Conference**

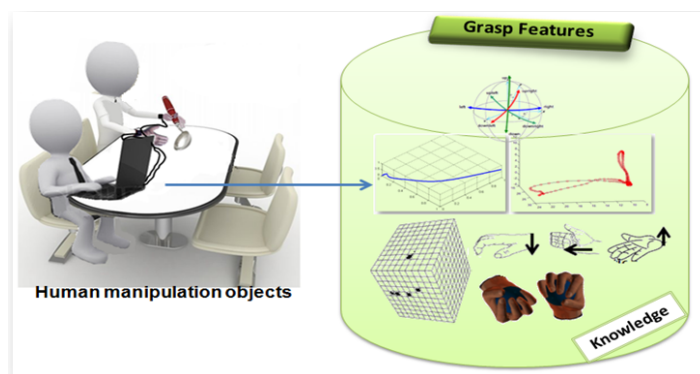
- Diego R. Faria, Jorge Lobo and Jorge Dias. "Identifying Objects from Hand Configurations during In-hand Exploration". In proceedings of the 2012 IEEE International Conference on Multisensor Fusion and Information Integration (MFI 2012), Hamburg, September, 2012.





## Chapter 5

# Grasp Features from Human Demonstrations



### 5.1 Introduction

One of the key elements of the performance of robotic platforms is the ability to perform autonomous grasping, manipulation, exploration and characterization of objects that are not completely known/familiar. Autonomous grasping and learning by imitation are topics that have been the focus of interest of many research groups in robotics for decades. This research aims to learn and model the human dexterity to endow a robot with such skills. The main objectives inside grasping strategy are to ensure stability and the ability of grasping unknown objects. Therefore, we focus on observing the human grasping performance to extract relevant features for learning, allowing then inferences in different grasping context.

In robotics, the analysis of human movements has been applied in research areas related with

task learning by imitation of human demonstrations. This approach is based on the principles described by several studies from human developmental sciences that, humans can acquire skills by watching and analysing others performing tasks. The challenge of using the human grasp demonstrations to model the manipulation strategies that will be performed by robotic platforms consists of building the bridge between the observed performances and the reproduction of movements that will produce the same effect.

In this chapter we present a framework to extract relevant information from human demonstration using multimodal data overlaid with object information, having both the perspective of the object state during the manipulation task and the perspective of the human performing the manipulation. Identifying different stages of a manipulation task and characterizing each phase of the task is important so as to retain the context in which different grasps and forces were used. One of the main elements in a manipulation task is the object being manipulated, and the effect of the human hand actions on the transformation of the object status from the starting conditions to the task goal. The object-centric probabilistic volumetric model is used here to represent the contact regions and forces that enabled successful grasps during the human demonstrations. The object centric framework will facilitate future matching for an artificial system observing objects and searching for cues on how to grasp it, taking into account the task context.

The next sections present the feature extraction implemented for the multimodal data from human demonstrations which builds upon these features to segment and identify manipulation stages and derive a generalized task representation, as well as the probabilistic map representation of contact and tactile data for successful stable grasp. The manipulation knowledge acquired from human demonstrations can be unified with object information into a framework to be used in grasping planning strategies and autonomous grasping.

## **5.2 Manipulation Task Database**

### **5.2.1 Experimental Setup and Data acquisition**

In this research the human grasp demonstrations play an important role in the ability of learning and identification of manipulation tasks, which includes hand and object movements, contact points of stable grasps, etc. The acquired data is used to model and extract the relevant aspects of the human demonstrations, providing the inputs for the methods presented in this chapter to represent the ma-

nipulation tasks, such as homogeneous and dexterous manipulation tasks [KI95]. The experimental activities with humans executing manipulation tasks are performed in our experimental area presented in Figure 5.1. The experimental area is equipped with multiple data acquisition devices in order to capture the different types of data used by humans to perform successful manipulation tasks. The system records human gaze, hand and fingers 6D pose, finger flexure, tactile forces distributed on the inside of the hand, colour images and stereo depth map. Using objects instrumented with inertial and force sensors, 6D pose and tactile forces on the object are also captured. This experimental area setup was built inside the Mobile Robotics Laboratory (MRLab), Institute of Systems and Robotics-University of Coimbra (ISR-UC), as part of the European project HANDLE [HANb].

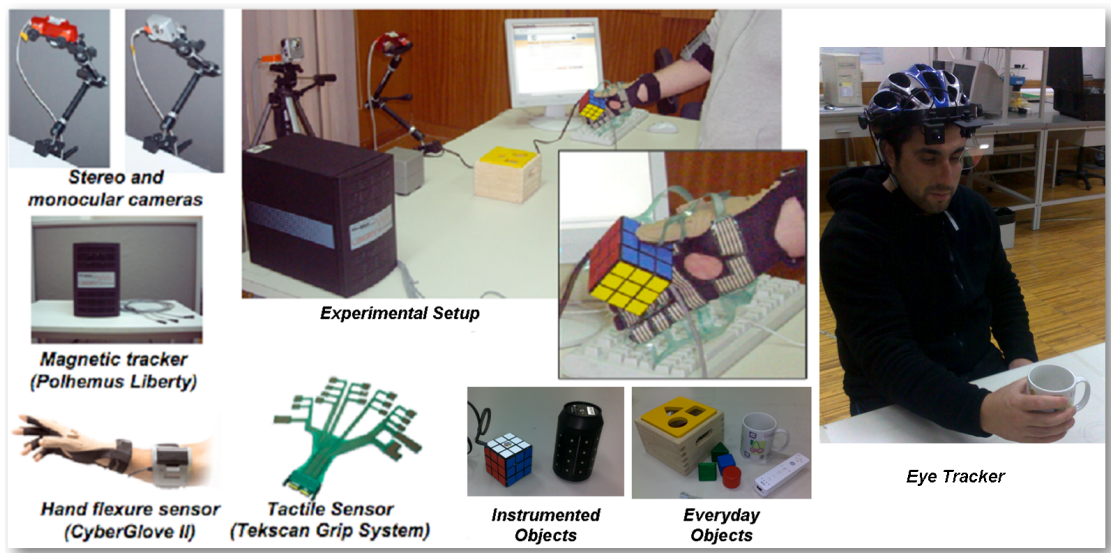


Figure 5.1: Global overview of the experimental area, data acquisition devices and objects available.

As previously mentioned, the data acquisition was implemented as a distributed and synchronized acquisition generating the timestamps for each sensor. We have used the Network Time Protocol (NTP) to have all computer clocks synchronized. A hierarchy of the computers was developed, where the server commands the clients, sending messages to communicate between them. Each client has one or more sensor plugged into it. Using the sockets programming, the server sends messages to all clients at same time to start the acquisition in a synchronized way. The same happens to stop the acquisition. This way, it is possible to access the multimodal data and correlate the data, finding the corresponding frames in different sensor modalities. Figure 5.2 shows a sketch of our data acquisition.

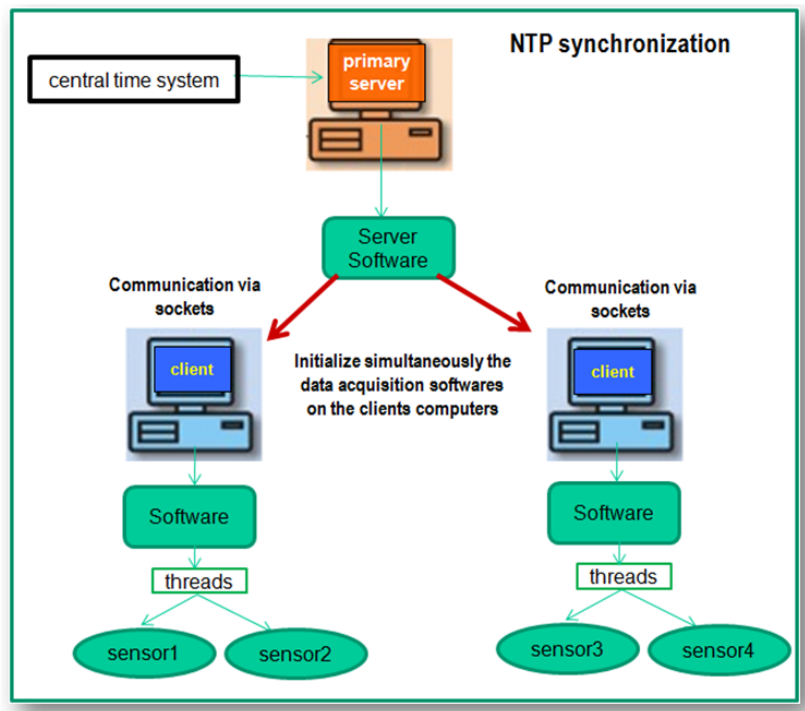


Figure 5.2: General representation of the data acquisition architecture implemented in the experimental area.

### 5.2.2 Data Storage

The data acquisition along with the database developed at ISR-UC was used in this research. In the database, different tasks achieved by human demonstrations can be found, such as simple manipulation tasks (e.g., object displacement), in-hand manipulation tasks where dexterous hand and finger movements are important (e.g., hand writing, pick-up an object and rotate it, task using screwdriver, etc.). An online database, the Handle Project - Data Collection Database [HANa], is publicly available with the collected datasets.

The data storage was developed in order to share the collected datasets. We have used the XML (eXtensible Markup Language) structure which allows us to define some standard for data reading and recording. The online database allows the upload and download of the XML structure with all acquired data from recording sessions of various systems. The XML structure allows the categorization of data so that an application can make a more consistent search. Some advantages of this structure are: advanced database search, flexible development of web applications, data integration from different sources, etc. Non-web applications also benefit from this format, simplifying the data set analysis and manipulation, for instance when loading a dataset into Matlab or another tool, a consistent set or subset can be selected, with associated setup and calibration information with the underlying raw

data in a simple direct format, such as text files and standard image formats.

During the data acquisition, each scenario can have several sessions. The application generates one XML file per data acquisition session for each sensor used. For each data acquisition session, a set of folder and files are created in order to store the acquired data. The folders and files are created according to a specific and pre-defined structure as shown in Figure 5.3.

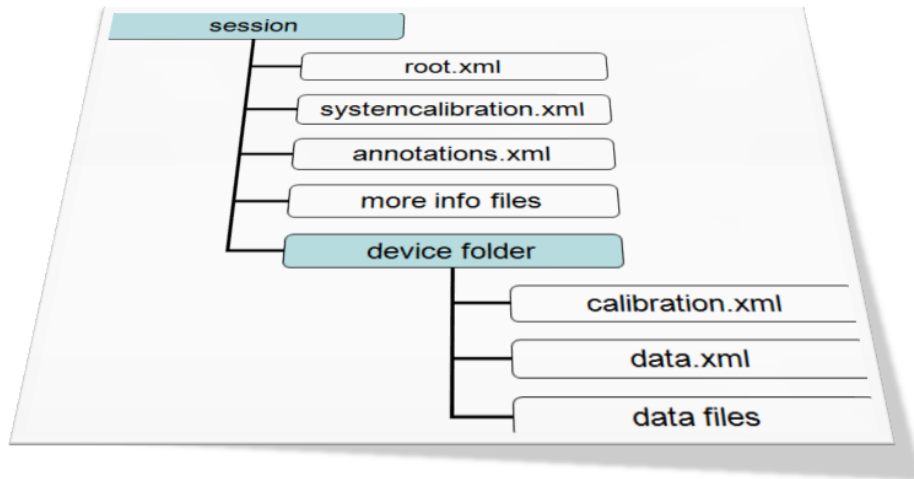


Figure 5.3: Schematic representation of the organization of folders (filled boxes) and files created during a data acquisition session; several session folders and device folders can be created.

The `root.xml` file stores the global information of the session acquisition regarding number of sensors used, timestamps format, folders for each device where the XML files are saved, and other information about calibration file (folders and file names), data and time of the recorded session, and if there is an annotation file to describe the actions phases and movements primitives/actions of the manipulation task. The structure and hierarchy of the `data.xml` file depends of the type of device it refers to. The specific structure of the elements of the various types of devices depends on the sensors returned data. An example is shown in Figure 5.4. The other devices follow the same idea, but with the information of the specific sensors, for instance, stereo camera instead of returning  $\{x, y, z, roll, pitch, yaw\}$  like the magnetic tracker, in this case, the file has information about left and right images names and location (frame 1 to frame  $N$  for each camera).

Given a specific scenario and a specific task, after the sensors data are recorded to reach a task goal, a user can use some tools to make manual annotations on what is happening in the scene as a task description to be stored in the database. The annotations provide a descriptive ground truth of real actions in the collected data. In the Handle project, the work team at ISR-UC has developed an on-line tool for annotation based on the images to describe each, or a set, of timestamps related to the

```

<dataset>
- <units>
  <X>inch</X>
  <Y>inch</Y>
  <Z>inch</Z>
  <YAW>degree</YAW>
  <PITCH>degree</PITCH>
  <ROLL>degree</ROLL>
</units>
- <data>
  - <rawdata timestamp="67">
    <X>7,269000</X>
    <Y>11,927000</Y>
    <Z>0,984000</Z>
    <YAW>75,236000</YAW>
    <PITCH>0,760000</PITCH>
    <ROLL>101,339996</ROLL>
  </rawdata>
  - <rawdata timestamp="131">
    <X>7,269000</X>
    <Y>11,927000</Y>
    <Z>0,984000</Z>
    <YAW>75,236000</YAW>
    <PITCH>0,760000</PITCH>
    <ROLL>101,339996</ROLL>
  </rawdata>
</data>
</dataset>

```

Figure 5.4: Content of the data.xml file for the Polhemus Liberty device for tracker sensor 1.

sensors signals. An automatic tool was also developed in partnership with other partners of the project to analyse the sensors signals and automatically annotate the primitives (actions and movements) of each task.

### 5.3 Grasp Detection from Contact Points Overlaid on the Object Model

Some grasping strategies for robotic systems are based on analysing the object geometric properties to fit suitable grasps, others on learning from human demonstrations using specific objects. Research on human motion and grasping has been carried out for automatic generation of grasping strategy such as [AA88], [KI95], [KFUM09]. Usually many approaches try to find a successful grasp given a 3D object model. Some of them associate the object model in specific geometrical primitives, [MKCA03] or fit to superquadrics model, [GALP07]. Thus, it is possible to set a specific number of candidate grasps for each object component. In [HK08] for instance, potential grasps are searched through cues provided by the primitives that were associated with a specific object. The authors in [RV08] compute grasp points based on the center of mass of the object's top surfaces. The object models are acquired

based on range images. Tests with a real robotic hand were not accomplished, but simulations were carried out to find a proper pose of the end-effector for those contact points. Saxena *et al.* [SDKN07] proposed a system that infers where to grasp an object using visual information. They apply machine learning techniques to train a grasping point model on labelled synthetic object images. Bohg *et al.* [BK10b] shows the analysis of grasping as a combination of a descriptor based on visual shape context with a non-linear classification algorithm that leads to a detection of stable grasping points for a variety of objects. The approach developed by [LP05] depends on the availability of a 3D object model to find a suitable grasp as a shape matching problem between the hand and the object. They use a database of human grasp examples and the object shape features are used to match against the hand postures.

In this section is explained how the grasp types are detected using the contact points of stable grasps overlaid on the object map. The fingertip locations on the object map cells (i.e., object surface) are the contact points used to detect the hand configurations. It allows to associate the grasp types with the object model or its components (geometrical primitives) as explained in the grasp synthesis chapter (Chapter 6).

To detect a grasp type automatically, we rely on the fingertip 6D pose relative to the wrist. For this, each finger has attached to it a tracker sensor, and an additional sensor is placed on the wrist. This allows us to compute the hand configurations defined in the grasp list [GRA]. To have the fingertip 6D data into the wrist coordinate system, we compute for each fingertip  $f = \{x, y, z, roll, pitch, yaw\}$  (contact points) the following step:

$$f_{new} = \{(f_x - w_x) + w_x, (f_y - w_y) + w_y, (f_z - w_z) + w_z, \alpha_w, \beta_w, \gamma_w\}, \quad (5.1)$$

where  $f$  represents each fingertip in  $\{x, y, z\}$  (in the frame of reference of the tracker sensor) and  $w$  the wrist coordinates;  $\alpha_w, \beta_w, \gamma_w$  are the angles roll, pitch and yaw of the wrist.

Later, the grasp type detection by using contact points on the object surface is achieved based on the transformed data representing a hand configuration. A grasp type is identified as a squared mean distance value between the fingers as shown in (5.3). As mentioned before, the grasp type used in this work are the ones defined in the grasp list adopted from [GRA], so that each discrete grasp type has an identity by using the hand configuration as explained next. First the Euclidean distances between thumb and index finger are computed, followed of the distances of the thumb and middle, thumb and ring, thumb and little, index and middle, middle and ring, and finally ring and little.

$$D_v = \frac{1}{N} \sum_{k=1}^N (d_{pq})_k^2, \quad (5.2)$$

where  $d_{pq} = \sqrt{(p_x - q_x)^2 + (p_y - q_y)^2 + (p_z - q_z)^2}$ , and  $p$  and  $q$  represent the Cartesian coordinates of two fingers (e.g., thumb and index, or index and middle, and so on). Given a new observation of stable grasp, after computed the contact points distances, we can associate it to a pre-defined grasp by a similarity measure to search for the closest grasp similarity:

$$\hat{g}_s = \min_{i \in \{1, \dots, N\}} f(\hat{g}_s) = |D_v - \delta_i|, \quad (5.3)$$

where  $D_v$  is the mean distance computed given the contact points;  $\{\delta_1, \dots, \delta_n\}$  are the grasping thresholds, i.e., each grasp is represented by a  $\delta$  value. Many observations (different subjects) of the same grasp value was computed (5.3) and an average for  $\delta$  was achieved to represent a learned grasp. The result  $\hat{g}_s$  is the grasp that will represent a new observation of contact points.

The grasp detection steps are presented in Algorithm 3. Figure 5.5 shows some demonstrations adopted in one of the objects for learning and the identified grasps types.

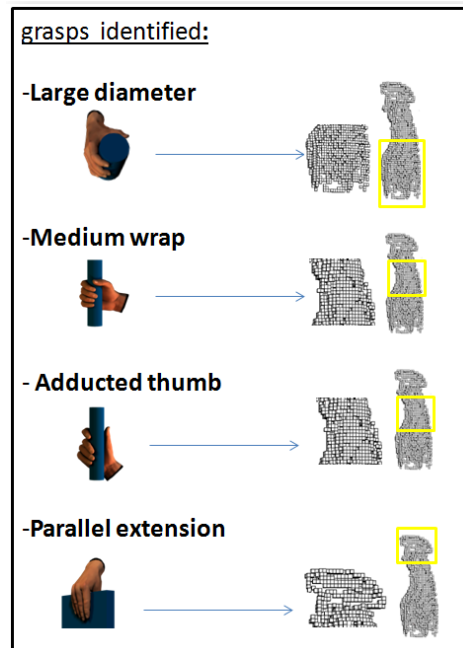


Figure 5.5: Some grasps that were identified for the spray.

A few sequences of contact points that form the hand configurations are presented in Figure 4.4 in Chapter 4. The sequences are overlaid in the full volume of the object computed in the object volumetric map. The four sequences presented are identified by using the contact points in the wrist



**Algorithm 3:** Grasp Type Detection

- 
- 1 Inputs: 6D fingertips contact points (acquired when the tactile sensors are active, i.e. touching the object)
  - 2 For each fingertip: Compute a transformation of the fingertip 6D data  $f = \{x, y, z, roll, pitch, yaw\}$  into the wrist coordinate system as demonstrated in eq. (5.1);
  - 3 Compute the Euclidean Distances  $d$  between the thumb fingertip and the other 4 fingertips:  $d(t, i), d(t, m), d(t, r), d(t, l), t = \text{thumb}, i = \text{index}, m = \text{middle}, r = \text{ring}, l = \text{little}$  ;
  - 4 Compute the Euclidean distance between the other fingertips  $d(i, m), d(m, r), d(r, l)$  ;
  - 5 Compute the grasp value by an averaged sum of the squared Euclidean distances between the fingertips  $D_v = \frac{1}{n} \sum_{k=1}^n (d_{i,j})_k^2$  ;
  - 6 Search for the minimum distance  $D_v$  and the learned grasping thresholds  $\{\delta_1, \dots, \delta_n\}$ , by computing a similarity function  $\hat{g}_s: \min_{i \in \{1, \dots, N\}} f(\hat{g}_s) = |D_v - \delta_i|$
  - 7 Output: Grasp type
- 

frame of reference as explained previously. According to the grasp taxonomy: sequences 1, 3 and 4 were identified as precision disk and sequence 2 is power disk. The examples presented show the potential of the probabilistic volumetric map applied for in-hand exploration of objects as well as for human demonstration of stable grasp.

In the next chapter we will show how to build a learning phase based on probability tables to distinguish what kind of grasp is more probable to happen in each specific situation and also the object region that was chosen for the grasp. For that, some trials relying on human grasp demonstrations of how to grasp an object given the object model will be presented.

## 5.4 Grasping Movements Recognition by Learning from Human Demonstration

Some of the most performed actions by humans in their daily activities involve the handling of objects for a specific task. The study of human reach-to-grasp and manipulation movements are important for researches of different areas. In computer science field, hand trajectories segmentation and classification are useful for human-machine interaction using gestures to interact with machines, for example, the hand can be used as a computer mouse. Various theories have been proposed for predicting hand trajectories. Hand trajectory segmentation and classification are also useful in the robotics field for imitation learning for human-robot interaction. Typically, the global hand's trajectory during a manipulation task can be segmented into different stages: reach, lift, transport and release [FBJ06]. In

this section we focus our attention on grasping movements. The intention is the development of an automated system for trajectories segmentation and classification by a probabilistic approach. In this work, the estimation and classification of reach-to-grasp movements when someone is performing the grasping is shown. By analysing these movements we are able to understand some human behaviours during the hand journey to reach and grasp an object. This information can be used to endow robots using the movements before the object manipulation, i.e., the capability of a robot recognizing how a human grasps an object to imitate the human action. This methodology can also be applied for gesture recognition tasks for human-robot interaction.

In order to learn and characterize the hand trajectories, we are discretizing them into significant changes in direction along the trajectories, hereafter named curvatures, and also detecting some pre-defined hand orientations with respect to a vertical reference. In this work, we are working with hand trajectories in 3D space. With our experimental setup, we have 6D pose data at 30Hz given by the tracker device that is attached to the fingertips and on the back of the hand. A smoothing mean filter with a centred window of 9 samples is used, followed by a 0 to 1 scale normalization using the initial and final points as reference. The trajectory is then segmented into action phases. The hand trajectory curvatures and hand orientation along each phase will be used to characterize each segment, so as to identify all the phases of the human manipulation demonstrations.

Figure 5.6 shows examples of features extraction along a hand trajectory: top image illustrates the curvatures along a 3D trajectory (*pick-up and place*); bottom image shows the hand orientation along the same trajectory.

### 5.4.1 3D Trajectories Segmentation: Curvatures and Hand Orientation Detection

A pre-processing step is applied in the raw data (hand trajectory) to smooth the possible noise. Due to the sensors (magnetic tracker) precision and resolution (spatial resolution: 8mm and angular resolution: 0.15 degrees), small shaky movements can be seen as noise when dealing with curvatures detection. To avoid this, a simple mean filter is applied (at each four previous and four forward neighbours of each point) to smooth the trajectory. A trajectory normalization is also applied to rescale all trajectories to the same size (0 to 1). This way, the hand movements that are not started in a pre-defined initial position can also be recognized by the trajectory shape when we detect the discretized curvatures. Figure 5.7 illustrates the pre-processing steps applied to the 3D hand trajectories.

In this work we are considering the discretization of curvature along 8 key directions, i.e.,

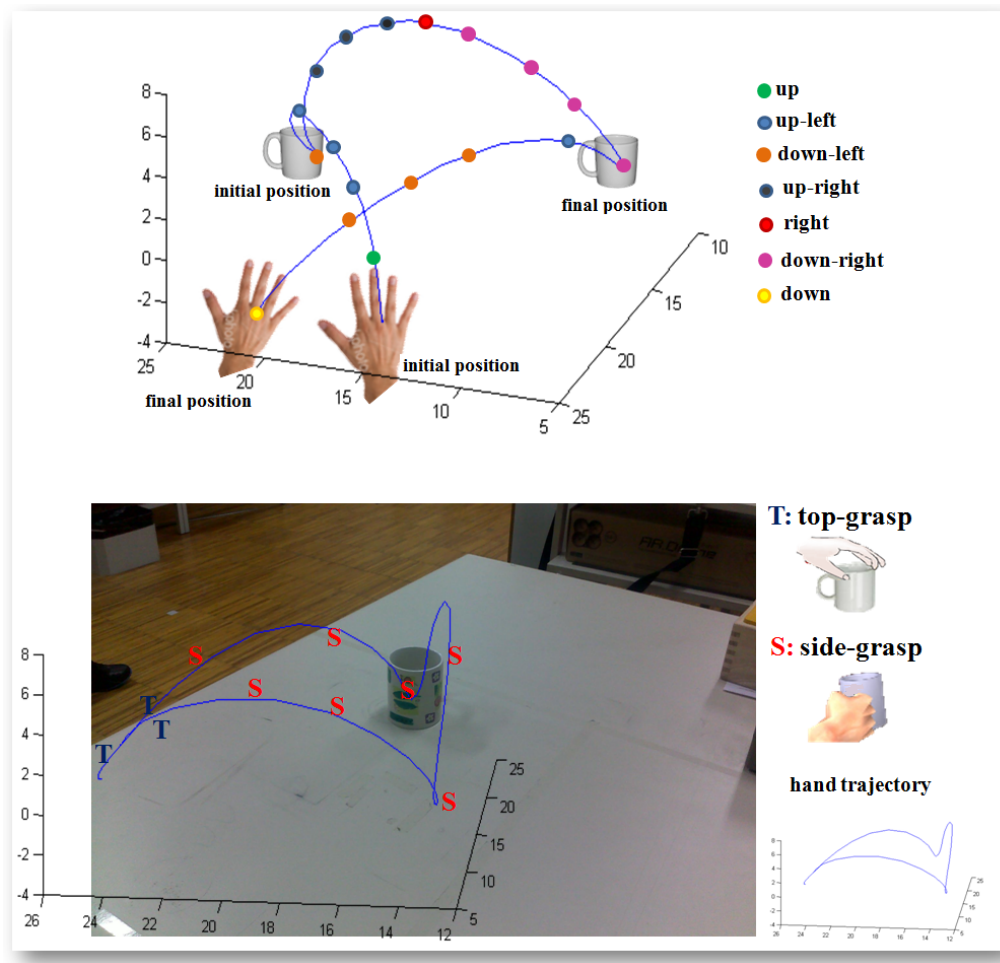


Figure 5.6: Top image: Example of a 3D trajectory of *pick-up and place* and possible curvatures along the trajectory; Bottom image: Example of hand orientation along the trajectory.

$c \in C = \{up, down, left, right, up-left, up-right, down-left, down-right\}$  along the hand trajectory. These are derived from the trajectory with a threshold on the level of significant change that triggers a new feature. The hand orientation is represented as  $o \in O = \{top, side, hand-out\}$ , and derived from the plane formed by three fingers (index, middle and ring finger).

As long as the trajectory is in 3D space, for better curvature detection we can work in cylindrical  $(r, \theta, h)$  or spherical coordinate system  $(r, \theta, \varphi)$ . Using two points of the trajectory we have the vectors representation in 3D space. The angle formed between these two vectors by the projection on  $(x, y)$  plane we achieve the  $\theta$  angle which give us the azimuth information, if the angle is increasing, we have the direction *left*, or if it is decreasing we reach the direction *right*. The same 2 vectors and their formed angles by the projection on  $(z, y)$  plane, we can achieve  $\varphi$  angle for pitch (tilt) information. In a 3D space we can make some combinations of the possible directions, for example, given the  $h$  information we have *up* and *down* directions; given the  $\theta$  angle, we have *left* and *right* directions;

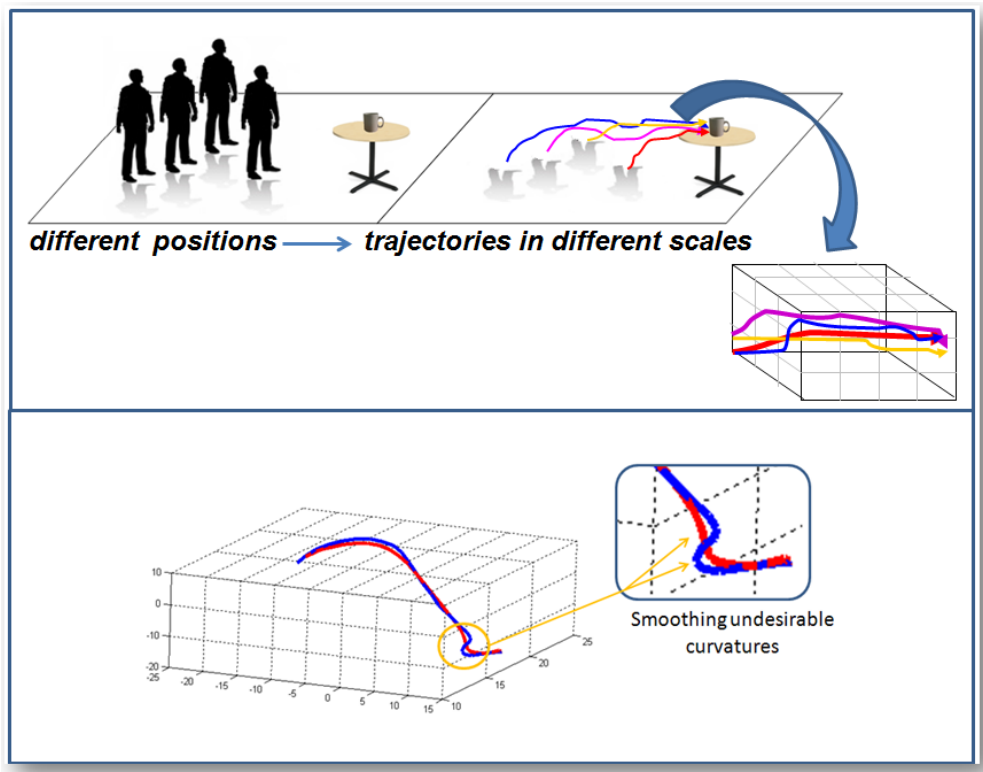


Figure 5.7: Top image: Illustration of trajectories normalization. Bottom image: Smoothed trajectory - Blue color represents the raw data; Red color represents the smoothed trajectory.

given the radius information  $r$ , we have *further* and *closer* directions, so that we can have several combinations of features. The height information  $h$  is achieved in a simpler way using the cylindrical coordinate system, calculating the difference between the  $z$  axis values from both points. In spherical coordinate system just the  $\theta$  angle can not give us the height or diagonals movements, being necessary to also verify the radius  $r$ , if it is increasing or decreasing and there are no changes in  $\varphi$  angle, this way, we reach this information. To know the directions *up* or *down*, there are changes on  $\varphi$ , and the variables  $r$ ,  $\theta$  remains the same. In cylindrical coordinate system we need to combine  $r$ ,  $\theta$  and  $h$  to know features like *up-right*, *up-left*, *down-right* and *down-left*. The next steps demonstrate how to achieve  $(r, \theta, \varphi)$  in spherical coordinate system:

$$r_1 = \sqrt{x_1^2 + y_1^2 + z_1^2}, \quad (5.4)$$

$$\sin\varphi = \frac{\sqrt{x_1^2 + y_1^2}}{r_1}, \quad (5.5)$$

$$\cos\varphi = \frac{z_1}{r_1}, \quad (5.6)$$

$$\varphi_1 = \arctan 2(\sin \varphi, \cos \varphi), \quad (5.7)$$

$$\cos \theta = \frac{x_1}{\sqrt{x_1^2 + y_1^2}}, \quad (5.8)$$

$$\sin \theta = \frac{y_1}{\sqrt{x_1^2 + y_1^2}}, \quad (5.9)$$

$$\theta_1 = \arctan 2(\sin \theta, \cos \theta). \quad (5.10)$$

Then with the second vector acquired by the second 3D point we follow the same steps that are used in (5.4) to (5.10) achieving then  $r_2$ ,  $\varphi_2$  and  $\theta_2$ . After that, we achieve the  $\theta$  angle and pitch information (height) given by  $\varphi$  angles as follows:

$$h = r_2 \cos \varphi_2 - r_1 \cos \varphi_1, \quad (5.11)$$

$$\theta = \theta_2 - \theta_1. \quad (5.12)$$

By adopting the cylindrical coordinate system, to find the height information, we can simplify the computation eliminating the equations (5.4) to (5.7) and we can replace the equation (5.11) by:

$$h = z_2 - z_1. \quad (5.13)$$

Then, to find the feature  $c \in \{up, down, left, right, up-right (UR), up-left (UL), down-right (DR), down-left (DL)\}$  we use the following rules:

$$c = \begin{cases} \mathbf{Up}, & \text{height} > 0 \text{ and } \theta \sim 0 \text{ and } r_{(x,y)} \sim 0 \\ \mathbf{Down}, & \text{height} < 0 \text{ and } \theta \sim 0 \text{ and } r_{(x,y)} \sim 0 \\ \mathbf{Right}, & \text{height} = 0 \text{ and } \theta < 0 \text{ and } r_{(x,y)} = 0 \\ \mathbf{Left}, & \text{height} = 0 \text{ and } \theta > 0 \text{ and } r_{(x,y)} = 0 \\ \mathbf{UR}, & \text{height} > 0 \text{ and } \theta < 0 \text{ and } r_{(x,y)} \sim 0 \\ \mathbf{UL}, & \text{height} > 0 \text{ and } \theta < 0 \text{ and } r_{(x,y)} \sim 0 \\ \mathbf{DR}, & \text{height} < 0 \text{ and } \theta > 0 \text{ and } r_{(x,y)} \sim 0 \\ \mathbf{DL}, & \text{height} < 0 \text{ and } \theta < 0 \text{ and } r_{(x,y)} \sim 0 \end{cases}, \quad (5.14)$$

where  $r_{(x,y)}$  is the radius in cylindrical coordinate system represented in  $(x,y)$  plane. It is computed as follows:

$$r_{(x,y)} = r_{2(x,y)} - r_{1(x,y)}, \quad (5.15)$$

where  $r_1$  and  $r_2$  are given by:

$$r_{1(x,y)} = \sqrt{x_1^2 + y_1^2}, \quad (5.16)$$

$$r_{2(x,y)} = \sqrt{x_2^2 + y_2^2}. \quad (5.17)$$

If  $h, \theta$ , and  $r$  are equal to zero, then there is no movement.

Splitting the trajectory in some parts, in a simpler way just defining the slices (e.g., 8 parts), or in a more complex way, detecting first the manipulation stages (explained in the next section), we can characterize the trajectory, so that each part can differentiate the type of grasp. After curvatures detection, the probability distribution of these features in each part of the trajectory is computed as follows:

$$P(c_i) = \frac{o}{N}, \quad (5.18)$$

where  $o$  represents the occurrences a specific curvature  $c_i$  in a specific hand displacement (trajectory part) and  $N$  is the total of curvatures found in each part, i.e., the sum of the total of occurrence of all  $c_i$ .

Regarding hand orientation, using three sensors on three fingertips we can approximate the

hand plane computing its orientation to find out if it represents top, side-grasp or hand-out. The three parallel fingers (index, middle and ring) usually remain parallel in the most part of the hand shapes for grasping, so it is a good example to form the hand plane. The hand orientation is achieved by computing the norm of the hand plane when we verify the angle formed using the hand norm related to the vertical axis ( $z$ ) of the magnetic tracker frame of reference. The hand orientation is computed given three points in each segment of the trajectory, then using the total occurrence of the discretized orientations (top-grasp, side-grasp and hand-out) in each segment, we can achieve the probability distribution as follows:

$$P(o_i) = \frac{o}{N}, \quad (5.19)$$

where  $o$  represents the occurrences of hand orientation  $o_i$  a specific trajectory segment and  $N$  represents the total of occurrences of all hand orientations found in that specific trajectory segment.

In the test case, first we are demonstrating grasping movements during the reaching phase until the grasp action. For this specific work, we have used a dataset with 140 trajectories for reaching movements, where 70 trajectories were performed for each type of movement: top-grasp and side-grasp. The movements were performed by 7 subjects, each subject demonstrated 10 times each type of movement. For each observation of our dataset, a XML file that stores the characterization of the trajectory is created, i.e., the segmentation information: amount of detected features and their probabilities in each trajectory segment. Two XML files for each trajectory were created, one with curvatures information and another with hand orientation. This information is useful to perform the histogram learning that will be used in the classification step. Figure 5.8 shows examples of reaching trajectories (Cartesian; inches measure) for pickup and place task for top-grasp (top-row) and side-grasp actions (bottom row). Figure 5.9 shows an example of the normalized and smoothed trajectory of top-grasp action after the pre-processing phase. Table 5.1 shows the result of trajectory segmentation by hand orientation acquired from the first trajectory (top-row) exemplified in Figure 5.9. The same process shown in Table 5.1 is done for the segmentation by curvatures as shown in Table 5.2.

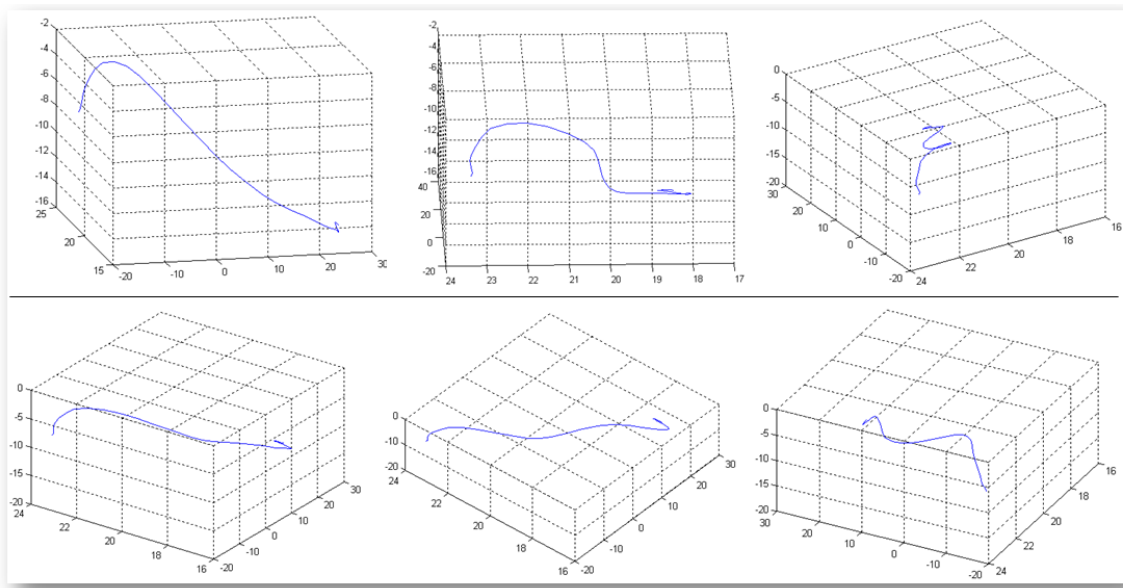


Figure 5.8: Top row: raw data representing trajectories of top-grasps actions during the reaching movement. Bottom row: Side-grasps.

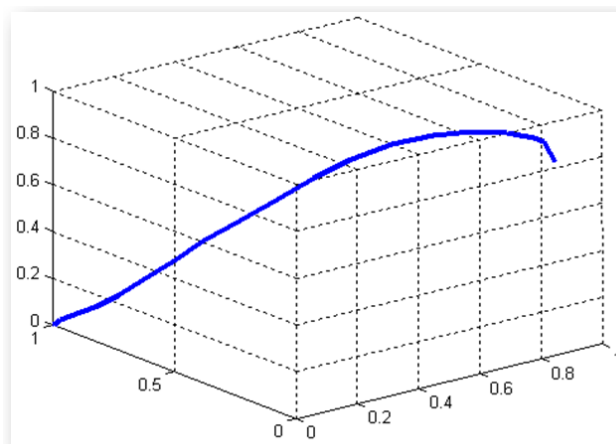


Figure 5.9: Hand trajectory after the pre-processing step (smoothing and normalization): Top-grasp action.

#### 5.4.2 Learning and Estimation for Trajectories Classification applying Bayesian Techniques

The learning phase is based on histogram of the discretized features. Some studies have prompted us to apply Bayesian method to classify human movements. Computational models for human perception and action have been explored by researches. Some studies about the human brain reports that Bayesian methods have achieved success in creating computational theories for perception and sensorimotor control [KP04].

In the learning phase all trajectories of our dataset session are analysed, where each observation



Table 5.1: Hand Orientation extracted along the trajectory: Result of our application for the trajectory shown in Figure 5.9. The second column is the amount of features found in each segment; the third column is the corresponding probability of each feature.

Trajectory Parts	Hand Orientation Side-Top	Hand Orientation Probab. Side-Top
1	5-4	0.56-0.44
2	3-8	0.28-0.72
3	4-7	0.37-0.63
4	3-8	0.28-0.72
5	2-10	0.17-0.83
6	1-11	0.08-0.92
7	1-13	0.07-0.93
8	1-16	0.06-0.94

Table 5.2: Curvatures extracted along the trajectory: Result of our application for the trajectory shown in 5.9. The second column is the amount of features found in each segment; the third column is the corresponding probability of each feature.

Trajectory Parts	Curvatures amount D-U-L-R-UL-UR-DL-DR	Curvatures Probab. D - U - L - R - UL - UR - DL - DR
1	1-2-0-0-0-3-1-1	0.120-0.260-0.00-0.00-0.000-0.38-0.12-0.120
2	1-4-0-0-2-4-1-1	0.080-0.300-0.00-0.00-0.150-0.30-0.08-0.080
3	1-5-0-0-2-3-0-1	0.080-0.420-0.00-0.00-0.160-0.25-0.00-0.080
4	1-1-0-0-1-6-1-1	0.090-0.090-0.00-0.00-0.090-0.54-0.09-0.090
5	1-1-0-0-1-4-0-1	0.125-0.125-0.00-0.00-0.125-0.50-0.00-0.125
6	1-1-1-2-0-1-1-4	0.090-0.090-0.09-0.18-0.000-0.09-0.09-0.360
7	3-0-1-0-1-0-2-3	0.300-0.000-0.10-0.00-0.100-0.00-0.20-0.300
8	5-3-2-1-3-1-1-2	0.278-0.167-0.11-0.05-0.167-0.05-0.05-0.110

represents a simple task: pick up the object. Given a set of observations to represent a type of Grasping  $G$ , at some displacement  $D$ , we have the probability of each type of curvature  $C$  in each part of a trajectory represented as  $P(C|G, D)$ . The same rule is used for hand orientation learning, so that we have  $P(O|G, D)$  where  $O$  represent all possible hand orientations. Here, the learned table is a mean histogram calculated from all top grasp and all side grasp probability tables acquired during the extraction process.

Each grasp type has its specific learning table. Figure 5.10 shows the Grasping Learning Tables (mean histograms) obtained after analysing all trajectories of our dataset, the curvatures features histogram and hand orientation histograms for each class of grasping, respectively.

Due to this learning process adopting histogram techniques, some features might have zero probability, because they never have been observed, i.e., a few types of curvatures or hand orientations will never happen to some specific trajectory. Whenever these features with zero probability occur

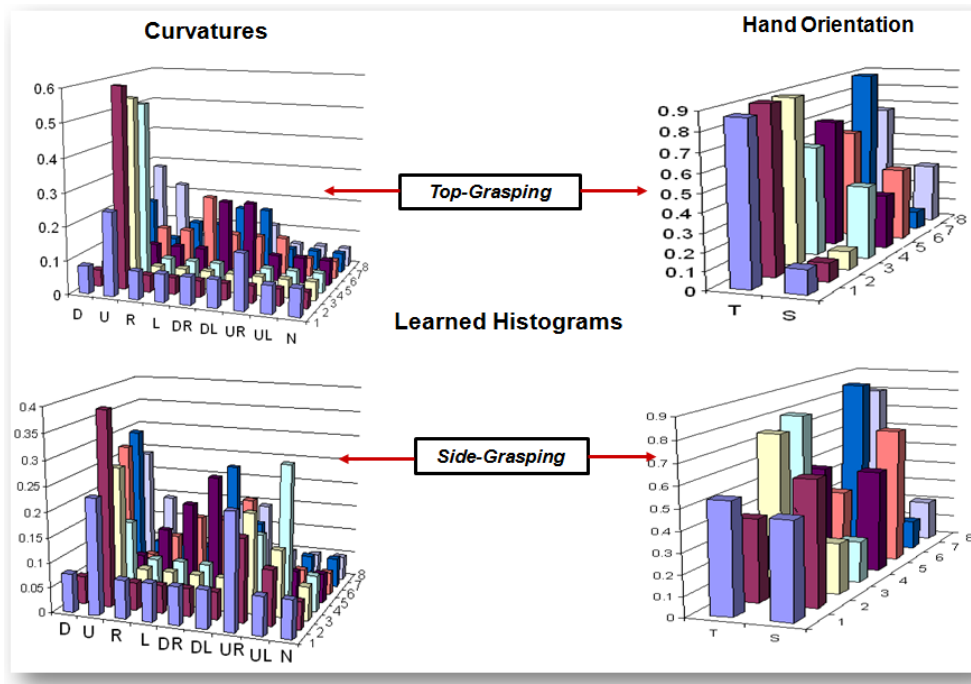


Figure 5.10: Grasping learning tables: Mean histogram for top and side grasping actions - curvatures and hand orientation features. Each feature has a probability assigned to it at each segment (1 to 8).

in the classification step, the corresponding hypothesis will receive also a zero probability. Since for the inference, the classifier is continuous, based on a multiplicative update of beliefs, these zeros would lead to a definite out-rule of the hypothesis. To avoid this problem we are using the Laplace Succession Law to produce a minimum probability for non-observed evidences, as follows::

$$\forall n_i = 0, P_{min}([n_i = 0]) = \frac{n_i + 1}{N + \chi} = \frac{1}{N + \chi}, \quad (5.20)$$

where  $P_{min}([n_i = 0])$  is the resulting minimum probability that will be assigned to the non-observed features ( $n_i = 0$ );  $\chi$  represents the total number of features (i.e., for all possible features, curvatures  $C = 9$  and orientation  $O = 3$ );  $n_i$  is a specific feature (the non-observed feature: curvature or hand orientation);  $N$  represents the total of occurrences (sum of all occurrences of features).

Bayesian classification models have already proven their usability in gesture recognition systems [RD07]. Based on this study, a Bayesian classification of grasp types analysing reach-to-grasp movements is presented here. The estimation and classification of a type of grasp happens along a trajectory that is being performed by a human subject. In each hand displacement (a segment of the trajectory) the probability of each type of grasp is updated, i.e., the application informs us which grasping is more probable to happen by the higher probability between top and side grasp variables.

During the classification, the estimation of each grasp type is computed, showing the probability of each grasp type at each hand displacement (corresponding by 1/8 of the trajectory). To better understand the general grasp classification model, some definitions are done:  $g$  is a known grasp from all possible  $G$  (Grasp types);  $c$  is a certain value of feature  $C$  (Curvature types);  $i$  is a given index from all possible hand displacement that composes the distance  $D$ , representing the trajectory size, where  $i = 1/8$  of  $D$ . The probability  $P(c|g, i)$  that a feature  $C$  has certain value  $c$  can be defined by learning the probability distribution  $P(C|G, D)$ . Knowing  $P(c|G, i)$  and the prior  $P(G)$  we are able to apply the Bayes rule and compute the probability distribution for  $G$  given the hand displacement  $i$  of the learned table and the feature  $c$ . Initially, the prior  $P(G)$  is assumed as a uniform distribution, and during the classification its value is updated according to the Dynamic Bayesian Network (DBN) theory:

$$P(G_{k+1}|c_{k+1}, i) \propto P(c_{k+1}|G, i)P(G). \quad (5.21)$$

We assume the same model of classification for hand orientation which is differentiated just by segmentation information, that is, the hand orientation instead of curvatures, where  $o$  is a certain value of feature  $O$  (hand orientation for side and top grasp). Knowing  $P(o|G, i)$  and the prior  $P(G)$  we apply Bayes rule as follows:

$$P(G_{k+1}|o_{k+1}, i) \propto P(o_{k+1}|G, i)P(G). \quad (5.22)$$

We formulate the equation in a recursive way (DBN). The posterior probability of a previous trajectory part becomes the prior for the next trajectory part (next hand displacement). Assuming that each hand displacement can find new curvatures and new hand orientations, then we can express the on-the-fly behaviour by using the index  $k$  that represents a certain hand displacement performed by the person in the reach-to-grasp movement. The rule for classification is based on it being necessary the highest probability value reaching a certain threshold (e.g., 0.7). We expect that a reach-to-grasp movement that is being performed by a subject to grasp the mug by top or side grasp will produce a grasp hypothesis with a significant probability. The Maximum a Posteriori (MAP) estimate is adopted to identify the grasp type.

Entropy (information theory) is used in this work as confidence level in the same way as explained in Chapter 2, here trying to improve and achieve a better classification based on results of

previous classification. After analysing the classifications results of trajectories by hand orientation and by curvatures as shown in 5.21 and 5.22, we can apply entropy to verify the best classification between both. For that, a confidence variable will be used as weight  $w \in \{w_1, \dots, w_n\}$  for each model of classification. The weight  $P(w)$  is used in the Bayesian mixture model as follows:

$$P(G|F, D) = \sum_{j=1}^n P(w_j) \times P(g_j|f, i), \quad (5.23)$$

where  $P(g|f, i)$  represents the classification result of each hand displacement computed as shown in 5.21 and 5.22,  $f$  represents  $o$  or  $c$  in the equation. Variable  $N$  is the number of components (here represented by types of classification);  $w_i$  is the weight of each component  $P(w_j)$ , and  $\sum_{i=1}^N w_i = 1$ . Each kind of classification is multiplied for its corresponding weight based on analyses of previous classification. Given the confidence of classification, we fuse the classification belief using the weights reached by the entropy.

### 5.4.3 Experimental Results for 3D Trajectories

The discretized features used in this work are sufficient to characterize and identify hand trajectories for reaching movements, and they can also be applied for identification of more complex movements (different task contexts), which encloses other stages, such as reaching, lift, transport and release.

All trajectories used for the learning stage were performed following a protocol in the experimental area. The subject is in front of the table and the object is located on the table. The right-handed subject has 6 sensors attached to the hand (on the fingertips and one on the wrist). An initial position of the hand is marked on the table. All trajectories have to be started from this initial position and when the trajectory is finished (it is considered when the hand touches the object), the hand has to come back to the initial position. For the test case, we are using a mug. Its orientation is a side position to the subject's field of view. This means that the mug has the handle part turned to the right side in the subject's field of view.

Figure 5.11 shows a side grasp trajectory performed by a right-handed subject and Table 5.3 shows the answer of the application along this trajectory classifying it by two independent features, first by using curvatures detection and the second one using hand orientation. It shows the probability updated by Bayes rule for both variables (top and side) in each part of the trajectory. The final probability in the last part of the trajectory (in the 8th part) is the result of the classification.

Comparing this case of trajectory shown in Figure 5.11, we can see that both classifications

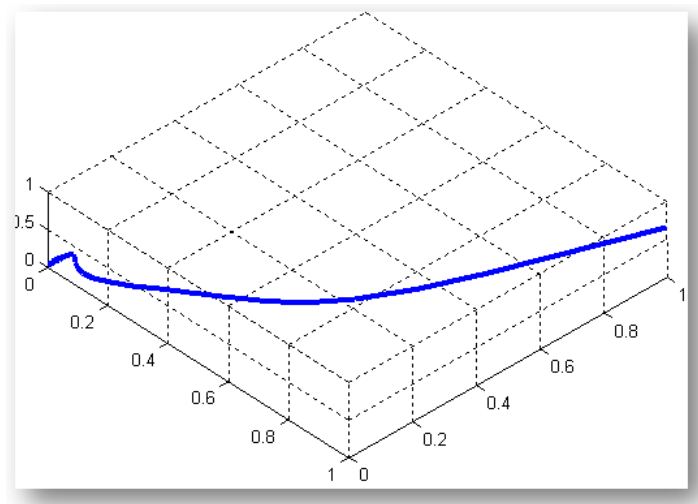


Figure 5.11: Side-grasp trajectory (after smoothing and rescale).

Table 5.3: Classification using Curvatures ( $C$ ) and Hand Orientation ( $O$ ) for the trajectory shown in Figure 5.11. The trajectory was classified as side grasp with 98.32% using curvatures and 92% using hand orientation. The estimation for top or side grasps in each part of the trajectory is shown.

Trajectory Part	Top%(C)	Side%(C)	Top%(O)	Side%(O)
1	34	66	19.10	80.90
2	34	66	4.76	95.24
3	34	66	4.76	95.24
4	0.68	99.32	4.76	95.24
5	0.68	99.32	4.76	95.24
6	0.68	99.32	8.25	91.75
7	0.68	99.32	10.83	89.17
8	<b>1.68</b>	<b>98.32</b>	<b>8.00</b>	<b>92.00</b>

achieved good performance classifying the trajectory correctly. In this experiment the classification by curvatures was better than by hand orientation. It happened because when adopting the hand orientations, the big difference between top and side grasp happened just in the end of the trajectory, when the subject was almost touching the object for the initial grasp. The results showed us that the reaching movements are similar before the grasp. In the case of curvatures detection, even the reaching movements being similar, more differences are detected due to the variation of the directions during the movement, since we can detect different types of curvatures given a sequence of two points of the trajectory. The curvatures can be detected in different ways, for example, at time  $(t + 1, \dots, t + 10)$ , or at trajectory segments of  $1cm, \dots, 10cm$ . In this work, we have used at each two points of the trajectory, when there is a variation (i.e., when the first point is different from the second point). Thus, we could detect more differences between one individual to another, being possible to better

characterize the movement. Therefore, our model is asynchronously updated based on the directions or orientation changes at each distance sequence (segment) of the trajectory. During the classification can be made independent of the segments stipulated, but at each hand displacement or in a temporal way (at each time instant  $t, t + 1$ ). The idea of using the trajectory segments was considered in this work to have a probability distribution at a sequence of changes in direction.

Following the protocol, two right-handed subjects, around 25 to 30 years old, have performed reach-to-grasp movements to test our application. Table 5.4 shows the results of the classification of 10 trials of side grasp using curvatures features, using hand orientation features and combining them using entropy as confidence level (weight) in a Bayesian mixture model, enclosing both class of features (likelihoods) into a single classification model. The false negative values in the classification using curvatures features happened due to the side-grasp trajectory are similar to the top-grasp. The classification using curvatures features when positive achieved higher values than the classification using hand orientation features. However, using hand orientation features, we did not have false negative values. Using the entropy  $H$  in these trials to reach an uncertainty measurement, we could assign weights for each classification model (each one using a specific class of feature), achieving the following weights:  $P(w_{curv}) = 0.61$  and  $P(w_{hor}) = 0.38$ .

Figure 5.12 presents a graphic with the classification results of a movement to perform a top-grasp action, where three models are used to classify the same trajectory. The first classification is using a Bayesian model relying on the likelihood with discretized curvatures features, the second with hand orientations features and another adopting the entropy weights in Bayesian Mixture Model to use the previous two likelihoods into a single model. The classification can be seen in each sample (segment or hand displacement) of the trajectory. Figure 5.13 shows a graphic representing the comparison between the three types of classification. The results show us that using the entropy belief is a kind of balance between the classification using the two types of features for the trials shown in Table 5.4.

The application was developed using the language C++. For these trials, laptop HP Pavilion dv5000, AMD Turion 64, 2.0Ghz, 1Gb of RAM was used. The segmentation process and classification are performed in real time.

To test the efficiency of the proposed classification framework, another class of trajectory was employed, typical trajectory for gesture recognition for human-robot interaction, or even for human-computer interaction. For that, more movements were learned (e.g., bye-bye and circle), with 30

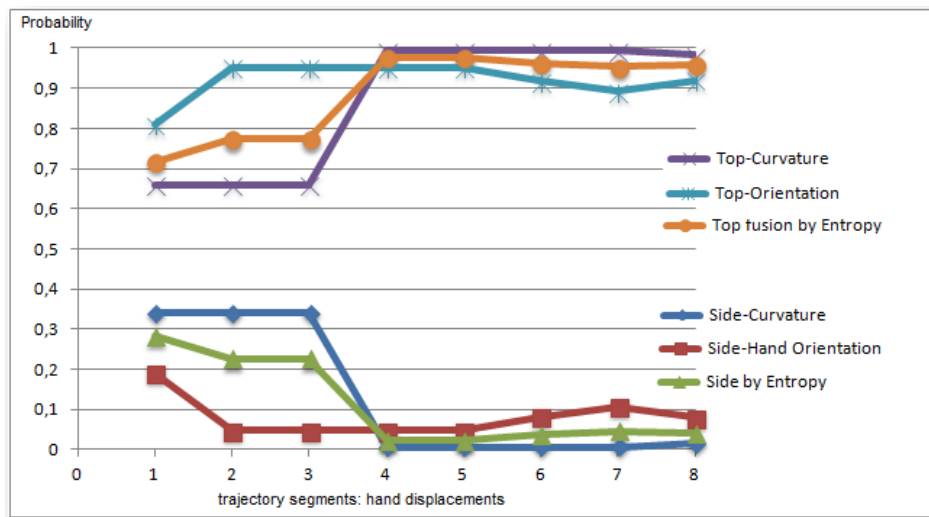


Figure 5.12: Classification of a movement of top-grasp during the reaching step. Three different model of classification: 1. Bayesian model relying on curvatures, 2. Bayesian model relying on hand orientation, 3. Bayesian mixture model relying on the weights given by entropy for the two previous models.

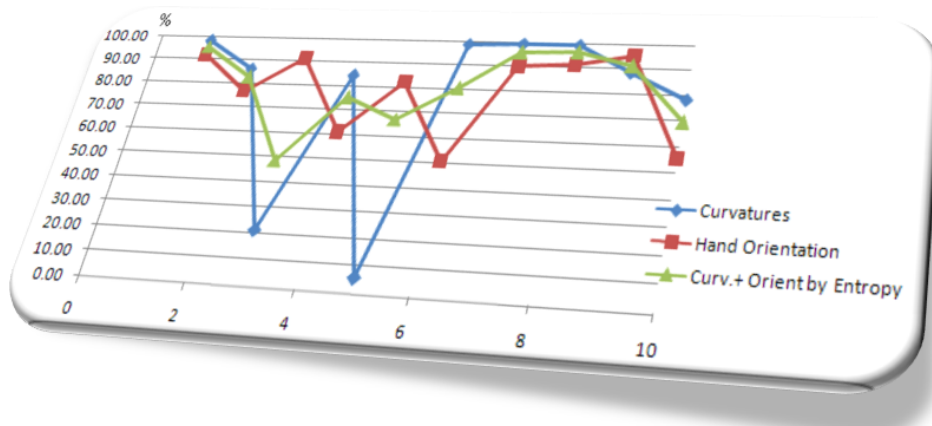


Figure 5.13: Comparison graphic. Classification using two different types of features and the third one using weights reached by entropy.

observations for both. Table 5.5 shows in the two first rows the classification of circles movements performed 10 times by two different subjects (first and second row, respectively), and the last two rows show the classification of the bye-bye movement after performed 10 times by these two subjects (third and fourth row, respectively). The results are similar to the top and side-grasp classification. In 10 trials, we have found a false negative for each movement.

The application for segmentation and classification of reach-to-grasp movements using two different ways of changes detection along the 3D trajectory showed to be a good alternative to distinguish the hand trajectories. Entropy as uncertainty measurement was applied to reach a confidence

Table 5.4: Result of 10 trials of Side-grasp trajectory. Two false Negative (less than 50%) on trial 3 and 5 using curvatures. The trials 4, 6 and 10 in hand orientation were considered as side-grasp, but with low probability, less than the threshold of 70%. Just one false negative (trial 3) was detected. Entropy was used to combine both features into a single classification model. The trials 5 and 10 were considered side-grasp with low probability.

Trial	1-Classification using Curvatures	2-Classification using Hand Orientation	3-Entropy to combine both features
1	98.32%	92.00%	95.85%
2	86.63%	76.93%	82.86%
3	<b>21.67%</b>	91.53%	<b>48.81%</b>
4	84.69%	<b>61.12%</b>	75.52%
5	<b>5.78%</b>	82.53%	<b>67.41%</b>
6	99.33%	<b>51.22%</b>	80.63%
7	99.68%	90.43%	96.08%
8	99.97%	91.53%	96.68%
9	88.98%	95.69%	91.58%
10	78.67%	<b>55.98%</b>	<b>69.85%</b>

Table 5.5: Classification of Circle (2 first rows) and bye-bye (2 last rows) movements.

Trial	1-Classification using Curvatures	2-Classification using Hand Orientation	3-Entropy to combine both features
1	95.30%	80.65%	89.40%
2	86.53%	78.95%	83.54%
3	87.59%	76.52%	82.90%
4	89.84%	82.92%	86.91%

level giving weights for both classifications for their fusion in a mixture model. The results have shown that using the weights reached from entropy for a joint classification has balanced the results, improving some classification when its probability is too low. The Bayesian techniques have shown to be an efficient way of classification for grasping movements.

#### 5.4.4 Simplifying for 2D Case: Hand Trajectories Segmentation and Classification

A simplification can be done to work with trajectories in 2D case. Here, to exemplify the proposed method, we are following the same initial idea of the last three previous subsections regarding the pre-processing step to smooth and normalize the trajectories as well as the learning and classification using Bayesian techniques. The main difference is on the discretized features detection. The focus of this work is on the changes detection in directions along the 2D trajectory, here also named as curvature detection. In this subsection we are not using the hand orientation. The sensors used to acquire the data were the magnetic tracker device, in the same way as the 3D case, but could be used



with monocular camera, tracking the hand trajectory and using the 2D data.

The changes detection along the trajectories is given by the computation of 2<sup>nd</sup> order derivative. For curvature detection along the trajectory, we decided to simply split the trajectory into segments as demonstrated in the 3D case, and afterwards the curvatures detection is made in each segment. It is done due to the on-the-fly classification that is performed during the hand displacement, i.e., to estimate and classify the trajectory that is being performed at each hand displacement, updating the trajectory classification. Initially we tested to split the rescaled trajectories into 1/4 and 1/8 of the trajectory size to verify the quality of the results.

Giving three points at each trajectory segment, we compute the second order derivative being able to detect the curvatures as expressed below:

$$d_1 = \begin{cases} \frac{y_2 - y_1}{x_2 - x_1}, & (x_2 - x_1) \neq 0 \quad \text{and} \quad (y_2 - y_1) \neq 0; \\ 0, & (y_2 - y_1) == 0 \quad \text{or} \quad (x_2 - x_1) == 0 \end{cases}, \quad (5.24)$$

$$d_2 = \begin{cases} \frac{y_3 - y_2}{x_3 - x_2}, & (x_3 - x_2) \neq 0 \quad \text{and} \quad (y_3 - y_2) \neq 0; \\ 0, & (y_3 - y_2) == 0 \quad \text{or} \quad (x_3 - x_2) == 0 \end{cases}, \quad (5.25)$$

$$Curvature = d_2 - d_1, \quad (5.26)$$

where  $d_1$  and  $d_2$  are the first and second derivative respectively and  $x_i$ ,  $y_i$  represent the Cartesian coordinates of the points. The curvature value is discretized given a determined threshold:

$$k = \begin{cases} -1, & curvature < -0,7 \quad \Rightarrow \quad down \\ 0, & -0,7 < curvature < 0,7 \quad \Rightarrow \quad line \\ 1, & curvature > 0,7 \quad \Rightarrow \quad up \end{cases}, \quad (5.27)$$

where  $k$  is the discretized curvature value. The threshold value was found in an empirical way. After some tests with threshold values and analysing the trajectory shape, we could find a value that satisfactorily returned the trajectories curvatures. By now, in this 2D case, the curvatures are limited to *down*, *up*, and when there is no significant changes in the direction, named here as *line*.

Figure 5.14 shows examples of reach-to-grasp trajectories for top and side-grasp performed and plotted in a 3D view. The trajectories were performed following the same protocol of the 3D case explained in the previous sub-section using the same object (mug) for the grasp action. Tables 5.6 and 5.7 show the curvature detection along the trajectories presented in Figure 5.14, top-grasp and

Table 5.6: Trajectory Segmentation: Result for trajectory shown in Figure 5.14 (left image:top-grasp).

Slices	Curv. Amount	Curv. Probab.
	D-L-U	D-L-U
1	3-2-5	0.3-0.2-0.5
2	3-2-2	0.43-0.285-0.285
3	3-2-1	0.5-0.333-0.167
4	1-1-3	0.2-0.2-0.6
5	1-1-2	0.25-0.25-0.5
6	1-0-3	0.25-0-0.75
7	1-0-2	0.333-0-0.667
8	3-0-2	0.6-0-0.4

Table 5.7: Trajectory Segmentation: Result for the trajectory shown in Figure 5.14, side grasp.

Slices	Curv. Amount	Curv. Probab.
	D-L-U	D-L-U
1	4-4-6	0.29-0.29-0.42
2	3-0-2	0.6-0-0.4
3	3-0-2	0.6-0-0.4
4	2-1-1	0.5-0.25-0.25
5	2-1-1	0.5-0.25-0.25
6	2-1-1	0.5-0.25-0.25
7	2-1-1	0.5-0.25-0.25
8	2-3-1	0.333-0.5-0.167

side-grasp respectively.

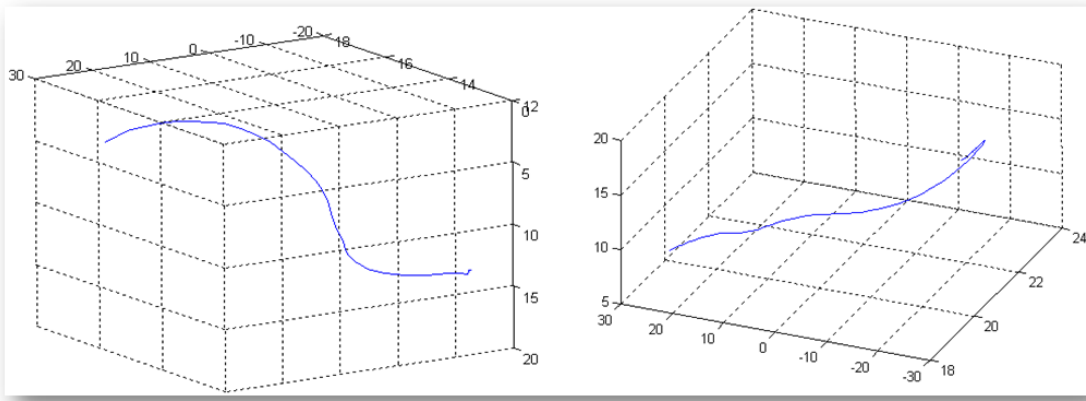


Figure 5.14: Reach-to-grasp trajectories (raw data: inches measure). Left image: Top-Grasping; Right image: Side-Grasping.

The learning phase and classification model follow the same steps of the 3D case, as explained in equation (5.21). The learned tables  $P(C|GD)$  computed by histogram techniques after analysing human demonstrations of the hand trajectories for top-grasp and side grasp actions are shown in Figure 5.15.

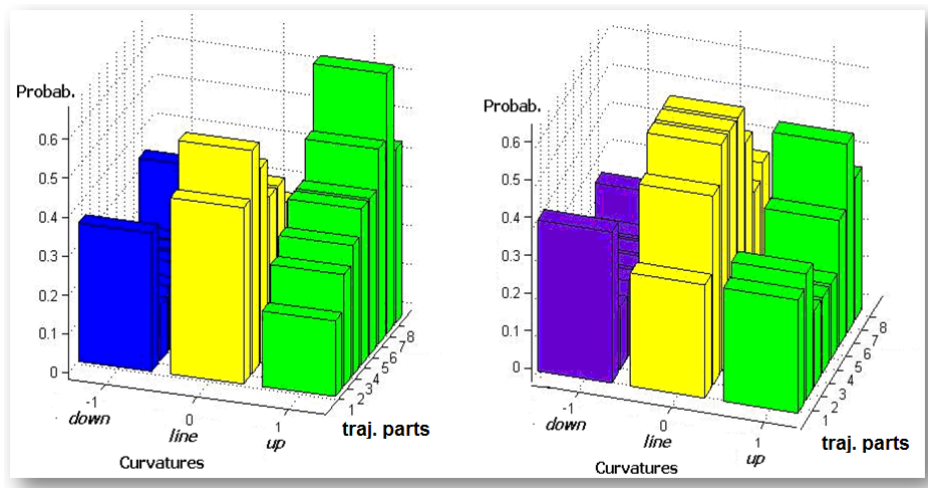


Figure 5.15: Learning tables: Left image represents the Top-Grasping Learning Table  $P(C|GD)$ . The probabilities of the curvature down vary of 0.14 to 0.35. The probabilities of line vary of 0.16 to 0.57. The probabilities of curvature up vary of 0.19 to 0.66. The right image represents the Side-Grasping Learning Table  $P(C|GD)$ . The probabilities of curvature down vary of 0.16 to 0.4. The probabilities of line vary of 0.3 to 0.6. The probabilities of curvature up vary of 0.2 to 0.5. The sum of the the features down, line and up in each trajectory part must be 1.

Table 5.8: Classification Result: Estimation of trajectory shown in Figure 5.14 (left image: top-grasp). At each trajectory part is shown the probability of the trajectory to be Top- or Side-Grasp. This trajectory was classified with probability of 87.12% as Top-Grasp.

Slices	Curv. Amount	Curv. Probab.	TG%	SG%
	D-L-U	D-L-U		
1	3-2-5	0.3-0.2-0.5	47.001	53.009
2	3-2-2	0.43-0.285-0.285	43.193	56.807
3	3-2-1	0.5-0.333-0.167	38.787	61.213
4	1-1-3	0.2-0.2-0.6	55.894	44.106
5	1-1-2	0.25-0.25-0.5	71.707	28.293
6	1-0-3	0.25-0-0.75	79.174	20.826
7	1-0-2	0.333-0-0.667	83.523	16.477
8	3-0-2	0.6-0-0.4	<b>87.120</b>	12.880

Tables 5.8 and 5.9 show the answers of the framework along the trajectories presented in Figure 5.14 classifying them. The application updates the probability of the variables Side-Grasp and Top-Grasp demonstrating which type of grasp is more probable at each hand displacement (trajectory part).

Following the same strategy as the 3D case, 2 right-handed subjects performed the reach-to-grasp trajectories to test the framework. After 10 trials we have observed the top-grasp movements were classified as having better results than side-grasp. This happened due to the side-grasp trajectories being performed with more variance between the movements. Table 5.10 shows the performance

Table 5.9: Classification Result: Estimation of trajectory shown in Figure 5.14 (right image: side-grasp). At each trajectory part is shown the probability of the trajectory to be Top- or Side-Grasp. This trajectory was classified with percentage of 83.70% as Side-Grasp.

Slices	Curv. Amount D-L-U	Curv. Probab. D-L-U	TG%	SG%
1	4-4-6	0.29-0.29-0.42	47.00	53.00
2	3-0-2	0.6-0-0.4	43.19	56.81
3	3-0-2	0.6-0-0.4	38.78	61.22
4	2-1-1	0.5-0.25-0.25	38.78	61.22
5	2-1-1	0.5-0.25-0.25	38.78	61.22
6	2-1-1	0.5-0.25-0.25	38.78	61.22
7	2-1-1	0.5-0.25-0.25	38.78	61.22
8	2-3-1	0.333-0.5-0.167	16.30	<b>83.70</b>

Table 5.10: Classification Result: 10 trials of Top-Grasping performed by 2 subjects. Blue color: probabilities > 70%; Red Colour: probabilities < 50%. In the trial 7 the trajectory was classified as Top-Grasp but with lower probability.

Trial	Probability	True Positive	False Negative
1	77.17%	x	
2	71.71%	x	
3	85.54%	x	
4	88.38%	x	
5	85.11%	x	
6	<b>38.79%</b>		x
7	<b>55.89%</b>		
8	83.52%	x	
9	87.11%	x	
10	<b>38.06%</b>		x

of 10 trials of Top-Grasp trajectories and Table 5.11 trials of Side-Grasp highlighting the probabilities in the classification, the true positive and false negative rate.

The results show that it is possible to achieve correct classification, although with a low rate of classification. However the classification results are not superior compared to the 3D case. This is explained due to the limitation of features detection along the trajectory, i.e., limited types of changes in direction.

## 5.5 Manipulation Tasks Identification by Learning and Generalizing Hand Motions

An important issue for modelling and recognition of human actions and behaviours are the motion patterns found during some activity. In different daily tasks the motion assumes an important key

Table 5.11: Classification Result: 10 trials of Side-Grasp performed by 2 subjects. Blue color: probabilities > 70%; Red Colour: probabilities < 50%. The trajectories in the trials 3, 5, 7 and 9 were classified as Side-Grasp with lower probabilities.

Trial	Probability	True Positive	False Negative
1	76.90%	x	
2	78.70%	x	
3	56.81%		
4	28.29%		x
5	54.20%		
6	83.70%	x	
7	52.99%		
8	76.20%	x	
9	59.13%		
10	37.13%		x

point to describe a specific action. The variety of human activity in an everyday environment is very diverse; the same way that repeated performances of the same activity by the same subject can vary, similar activities performed by different individuals are also slightly different. The basic idea behind this is if a particular motion pattern appears many times in long-term observation, this pattern must be meaningful to a user or to a task. In this section manipulation tasks at trajectory level to find similarities (significant patterns) given by multiple observations is the focus of the work. The intention is to learn and to generalize a specific task by the hand movement including fingers motion, as well as the object trajectory along the task for its recognition. This application is useful for task recognition in robot imitation learning and it can be applied in such a way that the generalized movements can be used in other contexts by a robot. We are not going through the imitation part, but we are focusing on the ability of learning and generalization.

### 5.5.1 Segmenting and Identifying Manipulation Stages

Segmenting a task in action phases can help us to characterize each movement of a task, as well as to understand the behaviour of the hand in each phase. By knowing the action phases of a task, we can discriminate easily a fixed grasp task (homogeneous manipulation) from a dexterous task due to the action transitions. Simple tasks (e.g., object displacement) can be composed of the following action phases: reach, load, lift, hold/transport, unload and release. A dexterous task is characterized by having the in-hand manipulation phase, where fine movements are performed with the intention of re-configuring the object state while it is being held by the hand. Dexterous tasks are usually composed of the following action phases: reach, load, lift, in-hand manipulation, unload and release.

Figure 5.16 illustrates an example of action phases in a simple homogeneous manipulation task, where the in-hand manipulation is replaced by a fixed grasp transport phase.

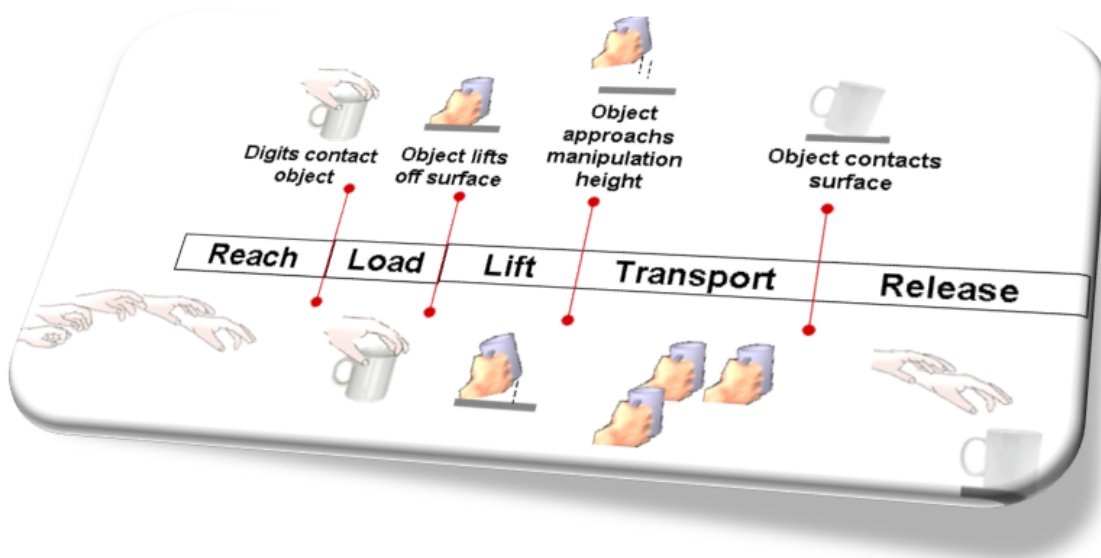


Figure 5.16: Example of action phases in a simple homogeneous manipulation task, where the same grasp is employed during the manipulation.

By observing the multimodal data, some assumptions can be made to find those phases during a task. For example, in the reaching phase, there is no object movement, the load phase is active when there is tactile information, and the transport phase when the object is moving. Since we have a synchronized data acquisition, by using the timestamps, we can analyse the multimodal data to know the state of each sensor in a specific time. Another option is segmenting by a probabilistic classification. Since we can extract features from the sensor signals, we can learn from multiple observations and then characterize each phase in a probabilistic way. Dealing with the uncertainty of sensor noise due to the real world is a reason for adopting a probabilistic approach to automatically classify the action phases.

### 5.5.2 Motion Pattern: Finding Similarities

An important step to model the human actions and behaviours is the motion pattern detection during an activity. In this work we are focusing on similarities (significant patterns) given by many observations of human actions. By looking for similarities among the features of a dataset of trajectories, it is possible to represent the dataset by its relevant features as illustrated in Figure 5.17. The relevant information are repeated motion patterns that are used to generate a generalized trajectory.

In this work, we use some grasp classes to estimate the grasp type along the task to esti-

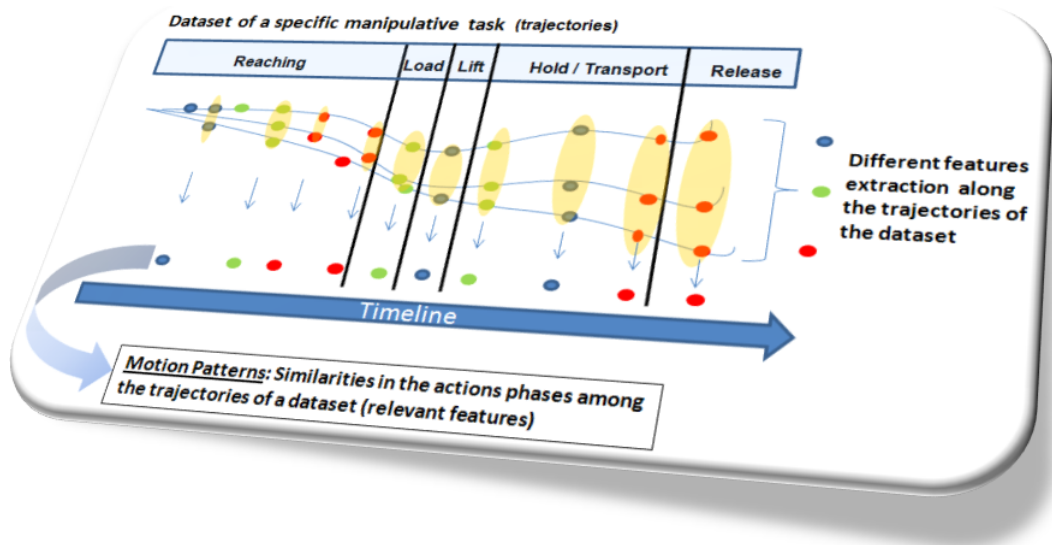


Figure 5.17: Motion Patterns: Similarities detection in the action phases of the trajectories in a dataset of a manipulation task.

mate the grasp transitions when a human is manipulating the object. In each task it is necessary to identify the types of the defined grasping/gesture and then compute the probability distribution  $P(\text{Grasp}|\text{Observation})$  of each one along the action phases of the task for each trajectory by analysing the grasping occurrences.

The dataset of trajectories are aligned temporally such as demonstrated in [CGB07], applying as a pre-processing step Dynamic Time Warping (DTW), a pattern-based method that allows the sequential information description of the data by the temporal distortion between different examples [SC78]. The next step is to detect the features and compute the probability distribution of the feature occurrences. Then similarities in all trajectories of a dataset are found, i.e., features with high probability (high occurrence in all the trajectories). A threshold is set on this probability to obtain a set of relevant features. The representation of a dataset of a specific task at trajectory level is given by the general form of the data. It is obtained after selecting the relevant features and then applying a regression on the spatial information of the relevant features.

### 5.5.3 Trajectory Generalization for Task Representation

There are some possibilities to achieve the general form (a smoothed trajectory) of a dataset of trajectories. The first one is an interpolation applied after the features selection (similarities between trajectories) as a function of arc length along a space curve using parametric splines. The second way is using the spatio-temporal information of all features extracted from all trajectories of a dataset,

where a polynomial regression is applied to fit the data to have a smoothed trajectory. The polynomial regression can be a good choice due to the curvilinear response during the fit and it can be adjusted because it is a special case of multiple linear regressions model. In case of applying regression, to have a correct fit, the regression needs to be done locally, at subregions of the trajectory due to the shape of the trajectories. In general, for our data, a cubic order polynomial regression is enough for the fitting. In this type of curvilinear regression, the choice of degree and the evaluation of the quality of the fitting depend on an empirical analysis. Although polynomial regression fits a non-linear model to the data, as a statistical estimation problem, it is linear, in the sense that the regression function is linear in the unknown parameters that are estimated from the data. It is based on least square fitting.

The general model of polynomial regression is given below:

$$Y_i = \beta_0 + \beta_1 x_i + \beta_{11} x_i^2 + \varepsilon_i \quad (5.28)$$

where  $x_i = X_i - \bar{X}$  and  $\varepsilon$  is an unobserved random error with mean zero conditioned on a scalar variable;  $\varepsilon$  can be computed as error of least square fitting;  $\beta$  minimizes the least square error. Examples of regression in sub-regions of hand trajectories is shown in Figure 5.18.

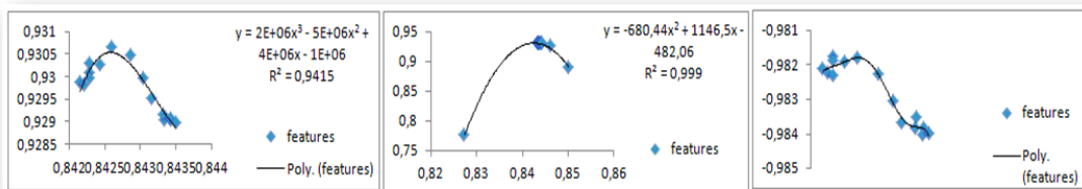


Figure 5.18: Regression applied on sub-regions of an action phase of a manipulation task. 2D view: left and middle images: x, y view; right image: x, z. Examples of quadratic and cubic order regression.

The hand motion generalization is useful to represent a task. For each dataset we intend to have a generalized data that can be used to endow a robot to perform the generalized movement. Each task will be represented by the generalized hand trajectory combining with the learned force intensities, grasp transition and contact points for stable grasp in each action phase of the task.

Figure 5.19 (left image) shows the raw data of the used dataset corresponding to the task: pick-up a mug and place it in another position (hand trajectories); and the right image shows the detected action phases using the sensors information. Figure 5.20 (left image) shows an example of the 3D positions of the features extracted (curvatures: trajectory directions) from all observations before



finding similarities; middle image presents relevant features selection by analysing the probability distribution of the features to know which type of feature is more relevant, later after computing the least square among all features points of the trajectories dataset we can estimate the coordinates of them; last image shows an example of interpolation of the features points as a function of arc length along a space curve by parametric splines.

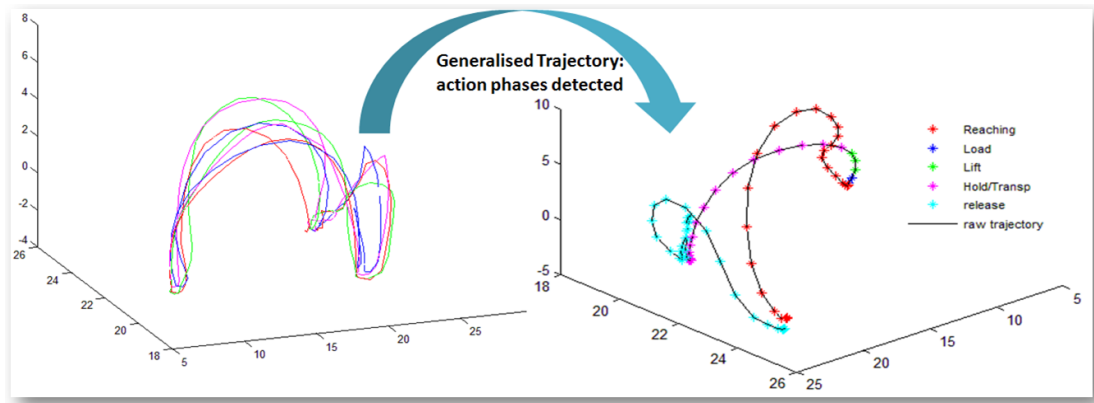


Figure 5.19: Left: Raw data(in inches): trajectories dataset (object displacement); Right: Trajectory segmentation by action phase.

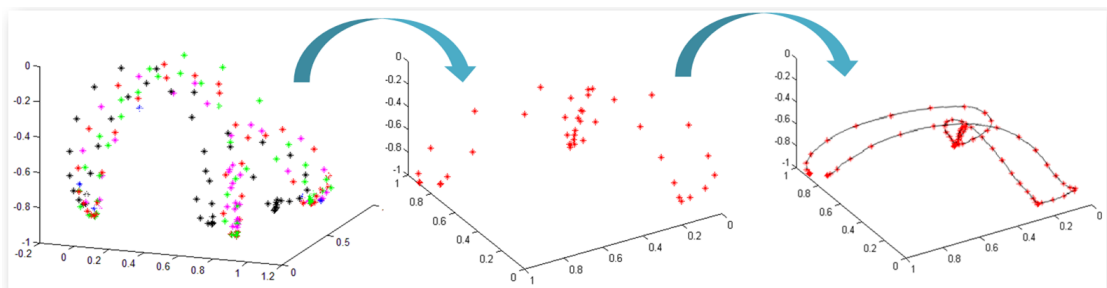


Figure 5.20: Left: Extracted features (Cartesian positions); Middle: Relevant features (similarities among all trajectories); Right: Generalized trajectory by interpolation of the points as a function of arc length along a space curve (adopting parametric splines).

Figure 5.21 shows the interpolation using parametric splines using the Cartesian 3D coordinates of the selected features after finding similarities between trajectories of a dataset of the task pick-up a mug and lift it (pour task).

#### 5.5.4 Tasks Identification

In the same way as the 3D hand trajectory classification as presented in previous section, here the task is represented as a complete hand trajectory involving different manipulation stages, as well as the object trajectory. Other relevant information can be used to identify a task, such as the grasp

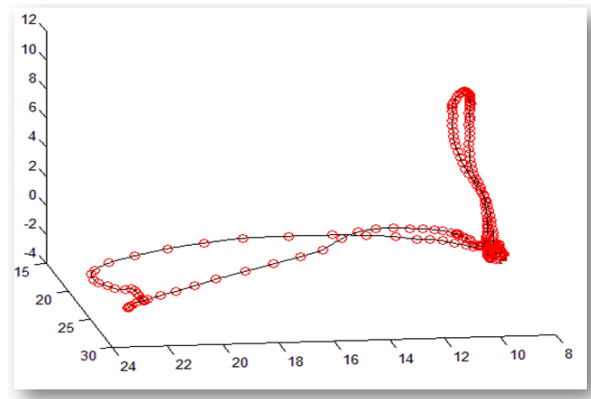


Figure 5.21: *Pick-up and lift* task representation at trajectory level: Interpolation using parametric splines using the Cartesian 3D coordinates of the selected features between the trajectories of the dataset of the same tasks.

transitions during each manipulation stage which demands the detection of grasp types as explained before in this chapter. The learning phase is based on histogram techniques as presented in the previous section using the defined and discretized features. Figure 5.22 illustrates the task features used at trajectory level for the task identification.

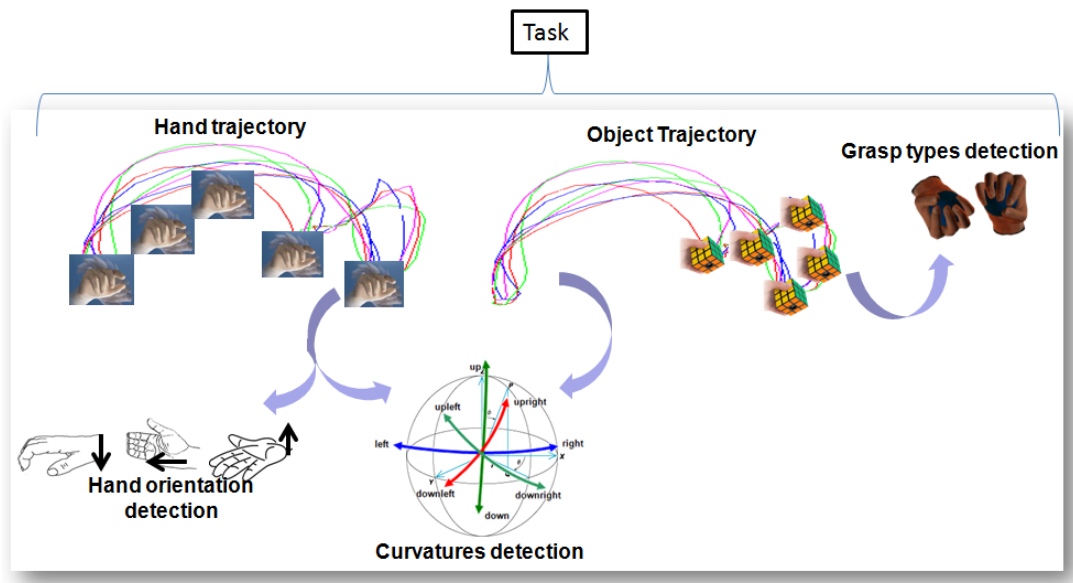


Figure 5.22: Task features at trajectory level.

First of all, it is important to identify if the task that will be classified has all action phases of the learned tasks and then it is possible to know if the task falls into the class of simple tasks or more complex tasks which includes the in-hand manipulation stage. Then, applying a continuous classification based on multiplicative updates of beliefs by Bayesian techniques taking into consideration the learned observations (relevant features of the general form of signals), we can identify a given task

into a class of the learned tasks.

Let  $t_g$  be a known task goal from all possible tasks  $\mathcal{T}$ ;  $c$  is a certain value of feature  $C$  (Curvature types) found in the hand trajectories;  $C_{obj}$  represents curvatures found in the object trajectories;  $G$  represents the learned grasping type;  $o$  is a certain value of feature  $O$  (hand orientation types) and  $i$  is a given index from all possible action phases  $A$ . The probability  $P(c|t_g, i)$  that a feature  $C$  has certain value  $c$  can be defined by learning the probability distribution  $P(C|G, A)$ ;  $P(o|t_g, i)$  of feature  $O$  learning  $P(O|\mathcal{T}, A)$ ;  $P(g|t, i)$  learning  $P(G|\mathcal{T}, A)$  and  $P(c_{obj}|t_g, i)$  of feature  $C_{obj}$  learning  $P(C_{obj}|\mathcal{T}, A)$ . Then the task classification model is represented as follows:

$$P(\mathcal{T} | c, c_{obj}, o, g, i) = \frac{P(c|\mathcal{T}, i)P(c_{obj}|\mathcal{T}, i)P(o|\mathcal{T}, i)P(g|\mathcal{T}, i)P(\mathcal{T})}{\sum_j P(c|\mathcal{T}, i)P(c_{obj}|\mathcal{T}, i)P(o|\mathcal{T}, i)P(g|\mathcal{T}, i)P(\mathcal{T})}. \quad (5.29)$$

Note that, here differently of the the previous section where we treat 3D hand trajectories (reaching movements), we are encompassing different likelihoods in the Bayesian formulation. We notice that adopting many likelihoods for the Bayesian model we can achieve a better classification than using just a specific class of feature in the model. The multimodal information helps to improve the classification reducing the ambiguities on the sensors signals.

### 5.5.5 Experimental Results

The trajectories that were used to test the task classification are *pick-up and place* and *pick-up and lift* using a mug as an everyday object. Here we follow the same protocol (to perform a task) and experimental setup as the 3D hand trajectory classification presented in the previous section. Given a new hand trajectory and the corresponding object trajectory, we want to identify the task. The classification variables (possible types of tasks, the learned ones) are updated in each action phase.

Figure 5.23 exemplifies a new observation, that is, the hand trajectory that is going to be identified (pick-up the object and place it). The learned tasks fall into the simple task class, *pick-up and place* and *pick-up and lift* the object to pour. Table 5.12 shows the result of the classification of a new observation of *pick-up and place*.

Figure 5.24 shows more results regarding task classification. The left image shows the behaviour on-the-fly of the classification at each time  $t$  to recognize if the task is *pick-up and place* (pp) or *pick-up and lift* (pl). The right image shows 10 trials of task classification using the proposed classification method presented in (5.29) enclosing multimodal features compared to a simple classification model using just a single type of features (curvatures), as presented in the previous section.

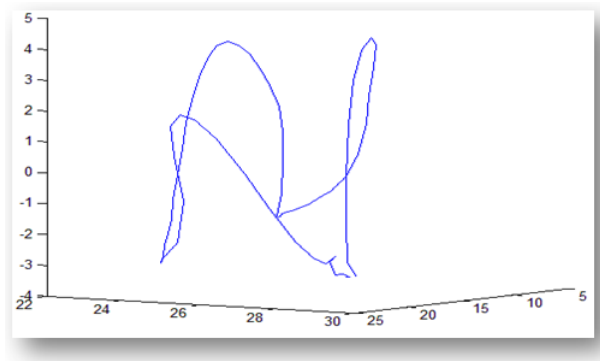


Figure 5.23: Task trajectory to be identified: *pick-up and place*.

Table 5.12: Classification Result

Action Phases	Pick-up and Place %	Pick-up and Lift %
Reaching	45.00	55.00
Load	48.10	51.90
Lift	59.32	40.68
Transport	69.83	30.17
Release	<b>78.00</b>	<b>22.00</b>

The results show that the multimodality ensures a better classification.

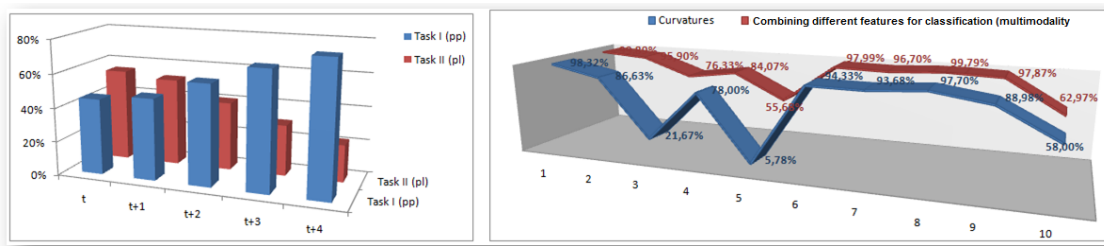


Figure 5.24: Task classification results after 10 trials. Comparison of classification using multimodal features and single feature for the same trials.

## 5.6 Object-centric Framework for Manipulation Knowledge

In the previous sections we addressed how features extracted from hand trajectories and contact points of stable grasps during in-hand manipulation could enable the segmentation and classification of the action phases, and also how an object probabilistic volumetric map could be used to plan the grasp strategies. This section presents how is possible to unify them into a single framework that associates the object probabilistic model and the hand approach vectors, initial grasps, and sequencing of grasps during in-hand dexterous manipulation. The rational behind this framework is that when confronted with objects for the first time, the artificial system can perform a match of the observed partial shape

with the volumetric map, and use the data in the framework to key the possible approach trajectories and grasps span for manipulating the object.

When searching in the framework for object graspable parts we also need to take into account the task context. Humans not only make some type of segmentation and identification of object parts in order to choose the best place to grasp, but are also task oriented in this choice.

From the multimodal data, we are able to extract relevant information of the performed manipulation tasks, such as different phases of the manipulation during the hand trajectory. From the hand trajectories and tactile data we identify the grasp types and transitions. These are mapped onto the object probabilistic volumetric model, so as to retain the relevant data from human demonstrations, concerning both the manipulation and object characteristics. The object centred framework will facilitate future matching for an artificial system observing objects and searching for cues on how to grasp it, and also taking into account the task context.

Figures 5.25 to 5.27 present some of the data collected in this framework. Figure 5.25 shows the hand trajectory during the in-hand manipulation task. The task is pick-up the object, rotate and repose it in another location. Figure 5.26 (left image) shows the object trajectory, where it is possible to visualize the object rotation during its trajectory for the same task; and the right image shows some transitions of hand shape during the same task. Figure 5.27 shows the object point of view, that is, the sequence of some contact points location during the in-hand manipulation phase.

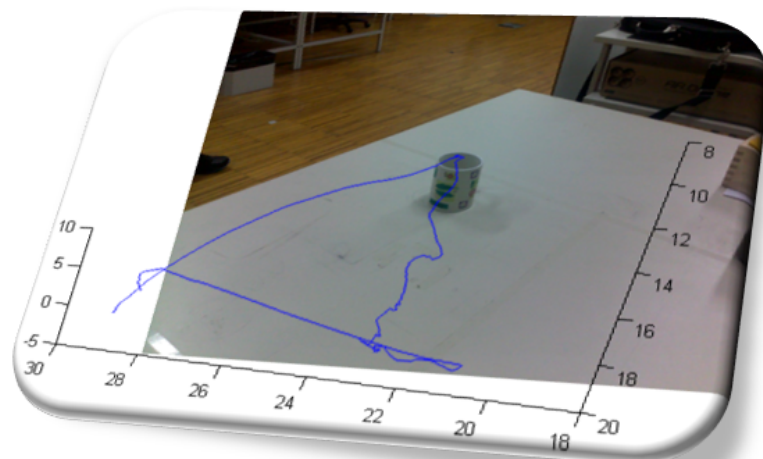


Figure 5.25: Hand trajectory (sensor attached to the back of the hand) during the in-hand manipulation task (pick-up the mug, rotate and release it).

In this work, the analysis of the stable grasps executed by humans during manipulation tasks is performed using a multimodal approach, in order to capture the multiple signals and strategies.

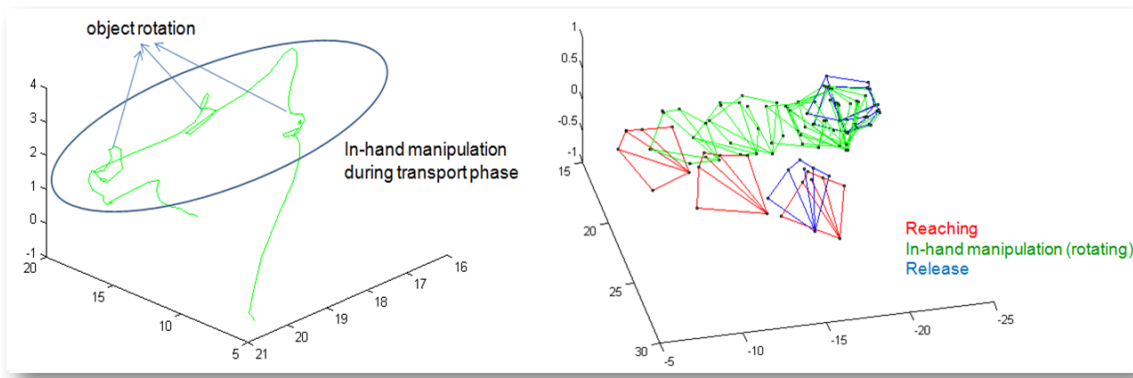


Figure 5.26: Left image: Object trajectory during the in-hand manipulation task (in green colour); The blue circle shows the in-hand manipulation phase (object rotation along the trajectory); Right image: Graphs representing some transitions of hand shapes during the in-hand manipulation task. The nodes represent the locations of the fingertips and wrist.

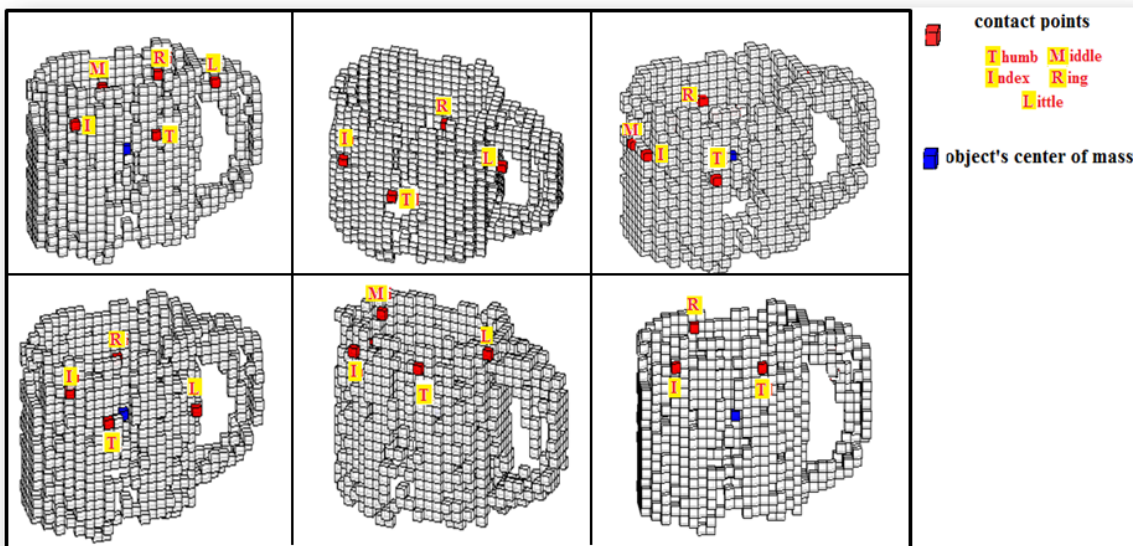


Figure 5.27: Sequence of some contact points overlaid in the static representation of the object (volumetric map) during the manipulation task. The contact points are given by the fingertips locations during the in-hand manipulation phase.

An object-centric probabilistic volumetric model is used to represent the multimodal data and map contact regions, gaze and tactile forces during stable grasps. One aspect that characterizes the manipulation task is the trajectory described by the hand (fingers, palm and wrist) to reach and contact the object, in order to perform the initial stable grasp. The location of the contact points of the fingertips on the object surface are acquired using Polhemus Liberty motion tracking system.

The biological signals related to tactile inputs are also relevant to perform the fine control of the manipulation tasks. The information about the level of activity of each region of the hand, during the contact with the object while the initial stable grasp, is acquired using the tactile sensing array

Tekscan Grip System.

The gaze has been used as an analysis tool of physiological responses to stimuli as an indication of cognition. The gaze in response to visual, auditory or cognitive stimulus is measured, during the manipulation task, using an SMI iView eye tracker device. The gaze provides important cues about the strategies used to find and anticipate the appropriated region of the object to be grasped. The eye tracking system uses infra-red illumination and computer-based image processing. The pupil is detected and after calibration, the pupil centre location is translated into gaze data. The gaze direction is mapped by the system in the scene images by a red cross as presented in Figure 5.28.

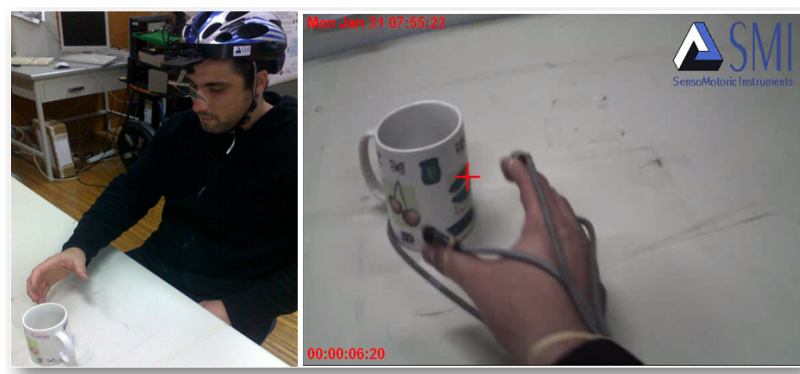


Figure 5.28: Eye tracker. Left image: Subject performing a manipulation using the eye tracker; Right image: Typical output of the eye tracker. Red cross indicates the estimated gaze direction.

Another way of using the object model is by using an eye tracker to estimate the regions of the object that are observed by the subject while doing the reach motion planning and during the load phase. Figure 5.29 shows some snapshots of the estimated observed regions during the manipulation of a mug, placed in a configuration where the handle of the mug is completely visible by the subject. The volumetric map represents both the observed regions of the mug and the regions which were effectively grasped. Figure 5.30 represents the results achieved in a situation where the handle of the mug was not completely visible to the subject. Although, during initial instants, the attention of the subject is captured by the partially visible handle of the mug, due to its inaccessibility, the subject chooses to grasp the mug using a side grasp applied to the lateral regions of the mug.

## 5.7 Discussion

This Chapter presented how to extract features from human demonstration from multimodal data for grasping movements, task recognition, and how to encompass human demonstration of stable grasps



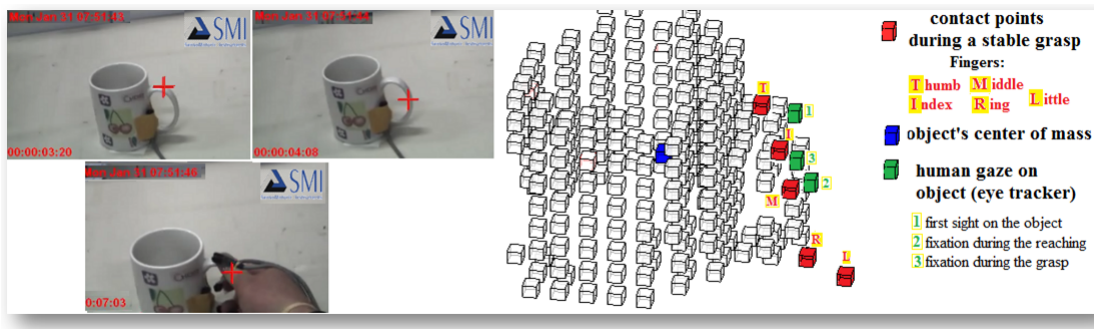


Figure 5.29: Human gaze during grasping and the contact points on the object surface. Task: Reaching and Grasping by the Object Handle. The visual gaze during the grasping shows that the human usually looks to the region of the object where will be performed the grasp.

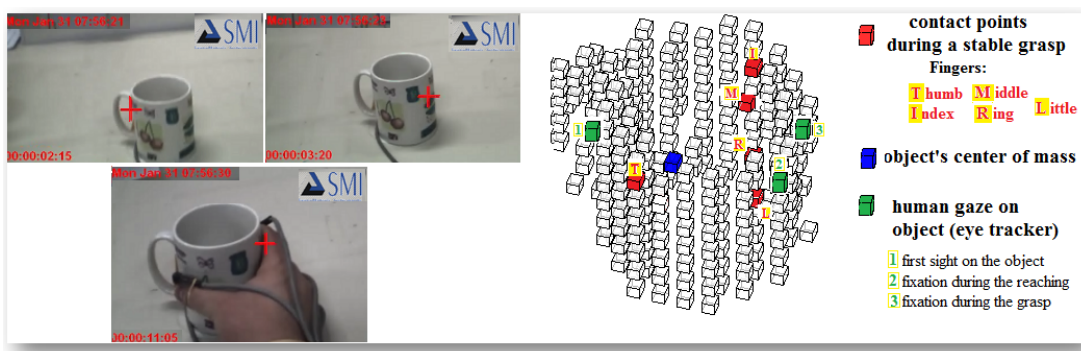


Figure 5.30: Human gaze during grasping and the contact points on the object surface. Task: Reaching and Grasping the mug by side-grasp. This type of grasping was chosen due to the orientation of the object - it influences the type of grasping.

with object model representation. The presented work starts from simple hand trajectories to more complex tasks involving in-hand manipulation of objects.

A consistent database with human demonstrations of manipulation tasks (from simple to more complex tasks including in-hand manipulation of objects) was used to test the proposed methods. The database was developed inside the European Handle project [HANb] at ISR-UC and is available on-line for the scientific community. The experimental setup with a distributed multimodal data acquisition used in this work was also developed inside the Handle project at ISR-UC. Using multimodal data to learn from human demonstrations of manipulation tasks, we can learn and derive suitable models of manipulation tasks and of the manipulated objects that can be used to endow an artificial dexterous hand to perform manipulation tasks. From the motion patterns, a generalized probabilistic representation for each type of task was derived. Results show the successful break down of action phases along a trajectory, as well as the suitability of the selected features as descriptors for the probabilistic approach used in task identification.



The utility of the object probabilistic volumetric map was shown in this chapter to overlay the partially observed volume of the object with data about human visual gaze when initiating a grasp task, hand-object contact points and tactile forces. Results of this representation suggest its suitability for grasp planning since a unified model has the relevant observed information on how to grasp the object.

Adopting a probabilistic framework we can correctly estimate and identify characteristics on the object model as well as grasping movements. The advantage of using probabilistic approaches is the way of dealing with the uncertainty of the sensors data allowing the reasoning and inference with high confidence based on previous observed knowledge.

The publications related to this chapter's subject, grasp features from human demonstrations and their applications, are listed as follows:

#### **Journal**

- Diego R. Faria, Ricardo Martins, Jorge Lobo, Jorge Dias. "Extracting Data from Human Manipulation of Objects Towards Improving Autonomous Robotic Grasping". *Robotics and Autonomous Systems*, Elsevier, Volume 60, Issue 3, March 2012, Pages 396-410, 2012.

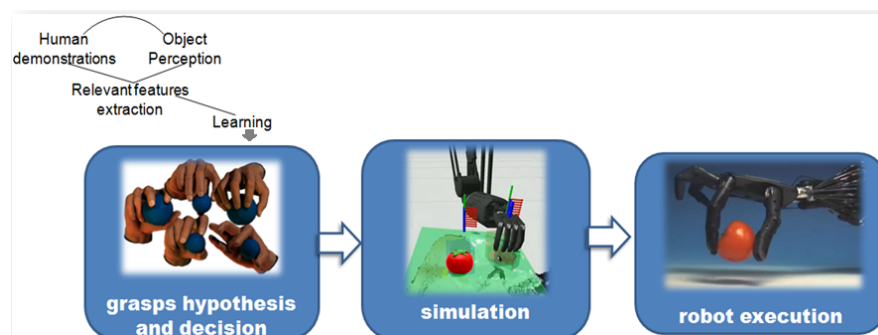
#### **International Conferences**

- Ricardo Martins, Diego R. Faria, Jorge Dias. "Representation framework of perceived object softness characteristics for active robotic hand exploration". In *Proceedings of 7th ACM/IEEE HRI'2012 - Workshop on Advances in Tactile Sensing and Touch based Human-Robot Interaction*, Boston, USA, March 5-8, 2012
- Jafar Hosseini, Diego R. Faria, Jorge Lobo, Jorge Dias. "Probabilistic Classification of Grasping Behaviours using Visuo-haptic Perception". In *Proceeding of 3rd Doctoral Conference on computing, Electrical and Industrial Systems (DoCEIS'12)*, Costa da Caparica, Portugal, 2012. *IFIP Advances in Information and Communication Technology*, Volume 372/2012, 241-248.
- Diego R. Faria, Ricardo Martins, Jorge Lobo, Jorge Dias. "Manipulative Tasks Identification by Learning and Generalizing Hand Motions". In *Proceedings of DoCEIS'11 - 2nd Doctoral Conference on Computing, Electrical and Industrial Systems*. Costa da Caparica - Portugal, February, 2011. *IFIP Advances in Information and Communication Technology*, Volume 349/2011, 173-180.
- Diego R. Faria, Ricardo Martins, Jorge Dias. "Learning Motion Patterns from Multiple Observations along the Actions Phases of Manipulative Tasks". *Proceedings of Workshop on Grasping Planning and Task Learning by Imitation: IEEE/RSJ IROS'2010*, - Taipei, Taiwan - October 2010.
- Ricardo Martins, Diego R. Faria, Jorge Dias. "Symbolic Level Generalization of In-hand Manipulation Tasks from Human Demonstrations using Tactile Data Information". *Proceedings of Workshop on Grasping Planning and Task Learning by Imitation: IEEE/RSJ IROS'2010*, - Taipei, Taiwan - October 2010.
- Diego R. Faria, Jorge Dias. "3D Hand Trajectory Segmentation by Curvatures and Hand Orientation for Classification through a Probabilistic Approach". In *Proceedings of the 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS'09*, St. Louis, MO, USA - October 2009.

- Diego R. Faria, Hadi Aliakbarpour, Jorge Dias. "Grasping Movements Recognition in 3D Space Using a Bayesian Approach", in Proceedings of ICAR'2009 - The 14th International Conference on Advanced Robotics - Munich, Germany, June 22-26, 2009. Print ISBN: 978-1-4244-4855-5.
- Diego R. Faria, Ricardo Martins, Jorge Dias, "Human reach-to-grasp generalization strategies: a Bayesian approach" - Workshop at Robotics Science and Systems 2009: "Understanding the Human Hand for Advancing Robotic Manipulation" - July 28, 2009 - Dillon Eng Seattle, WA, USA
- Diego R. Faria, Jorge Dias. "Bayesian Techniques for Hand Trajectory Classification", RECPAD 2008 - 14th Portuguese Conference on Pattern Recognition, Coimbra-Portgal, 31st October, 2008.
- Diego R. Faria, Jorge Dias. "Hand Trajectory Segmentation and Classification Using Bayesian Techniques", Workshop on "Grasp and Task Learning by Imitation" 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, Acropolis Convention Center, Nice, France Sept, 22-26, 2008, pp.44-49.

## Chapter 6

# Grasp Synthesis based on Human Grasp Demonstrations



### 6.1 Introduction

Humans are able to learn new skills, and to adapt to different complex environments and interact with objects (including unknown) to manipulate them. This results from a lifelong learning, and also observation of other skilled humans. To obtain similar dexterity with robotic hands, cognitive capacity is needed to deal with uncertainty. By extracting relevant multi-sensor information from the environment (objects), knowledge from previous grasping tasks can be generalized to be applied within different contexts. Based on this strategy, we show in this chapter that learning from human experiences is a way to accomplish our goal of robot grasp synthesis for unknown objects. We address an artificial system that relies on knowledge from previous human object grasping demonstrations. A learning process is adopted to quantify probabilistic distributions and uncertainty. These distributions are combined with preliminary knowledge towards inference of proper grasps given a point cloud of

an unknown object. In this chapter, we designed a method that comprises a twofold process: object decomposition and grasp synthesis. The decomposition of objects into primitives is used, across which similarities between past observations and new unknown objects can be made. The grasps are associated with the defined object primitives, so that feasible object regions for grasping can be determined. The hand pose relative to the object is computed for the pre-grasp and the selected grasp.

To accomplish the objective defined in this chapter, we are encompassing many of the methods presented in previous sections into a single framework. Figure 6.1 depicts an overview of our proposed approach

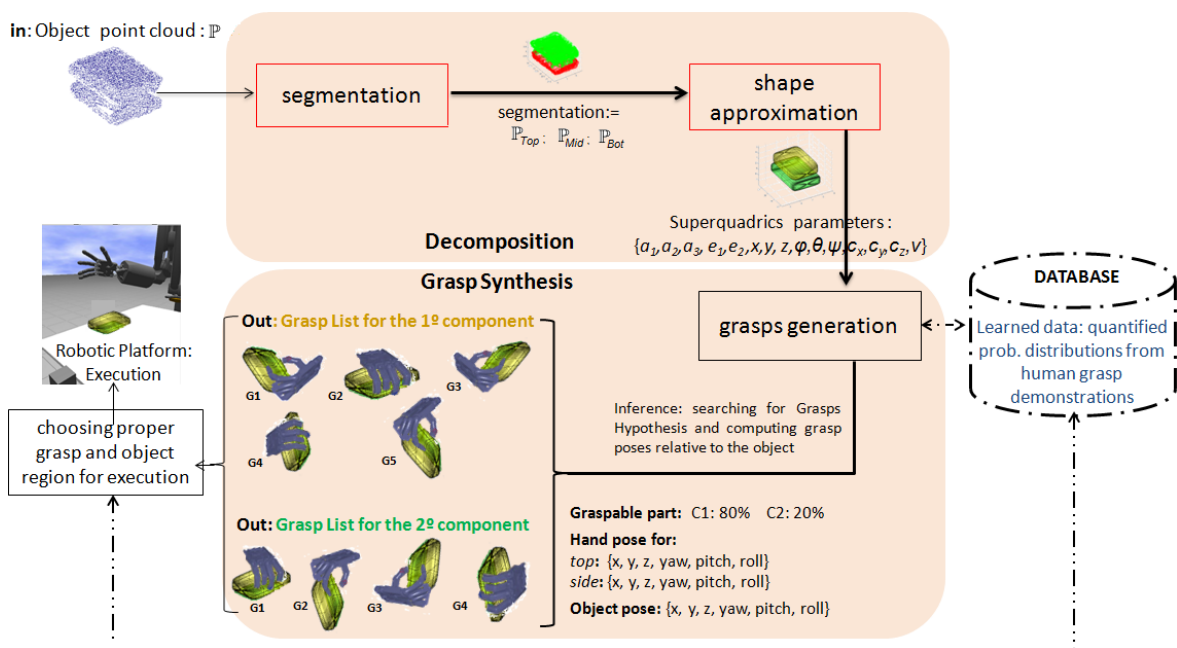


Figure 6.1: Overview of the grasp generator modules.

The next section will present the learning strategy using the human grasp demonstrations that is based on the Bayesian techniques presented in previous chapters for reasoning, i.e., inferring how to generate proper grasps for a specific object, and estimating the proper region on the object for grasping. Then, the grasp synthesis architecture that encloses the steps mentioned before to develop the decompose module, where methods of object segmentation and shape modelling given an object point cloud are adopted, as explained in Chapter 3, that was also used in the learning process and when the system is applied in a real context using the robotic platform. Later, the generation of grasp pose relative to the object will be presented within the grasp synthesis module, followed by the the experimental results, including simulations and also an example in a real application using a dexterous robotic hand. In the end of this chapter we will draw some conclusions and present

directions for future work about this topic.

## 6.2 Learning from Human Grasp Demonstrations

In this section we address the learning from human grasp demonstration that will assist the grasp hypothesis generation, which we are basing on previous works, such as the models presented in Chapter 5 to build the learning strategy.

The learning phase follows a probabilistic approach, and in general finds a model that describes the dependency of one random variable on another one. Let  $q_i \in Q, i = \{1, \dots, n\}$  be possible object regions (quadrics) and  $g_i \in G, i = \{1, \dots, n\}$  be the possible grasp types, then the dependency is defined by a conditional probability distribution  $P(Q|G)$  that is the probability density function (*pdf*) of a random variable representing one of the target classes given the random variable representing the input vector (features). In other words, this means that, given the space of possible inputs  $Q$  and the possible targets space  $G$ , an estimate of the class that encloses the input space is given by a classification model resulting  $P(G|Q)$ .

The human grasp demonstrations result in a dataset  $\mathcal{D}$  with  $N$  labelled examples coming from the learning phases as presented subsequently in the following subsections. An example of a labelled dataset is given by the possible candidate grasps  $G$  associated with each quadric  $q_i$  representing an object region. Each  $g_i$  represents a discrete grasp type that is included in a list of 33 grasp types [GRA], which is used in this work. Each grasp type can be found using the hand fingertips 6D data given in the wrist coordinate system, forming a hand configuration. We have a database that includes the datasets of the labelled examples coming from the observations and it also includes the learned probability tables achieved during the learning phases. From the observed data, inferences can be made to assist the grasp synthesis to estimate possible grasps given a quadric model, or to search for possible candidate regions on the object to perform a grasp, or even the candidate regions on the object given a task context  $\mathcal{T}$ . In general, a hypothesis is represented as  $h : q_i \rightarrow G$  that can correctly predict the set of possible grasps  $G$  associated with each  $q_i$ , which means that the classifier (inference model) is represented as a function  $f : q_i \rightarrow G$  that assigns a class label  $g_i \in G, i = \{1, \dots, n\}$  associated to the object region for grasping  $q_i \in Q, i = \{1, \dots, n\}$ . The inference models are based on Bayesian techniques that will be explained later.

Figure 6.2 depicts an overview of the learning and inference strategies adopted in this work. In the next subsections, the steps for learning and inference based on human grasping experiences will

be described .

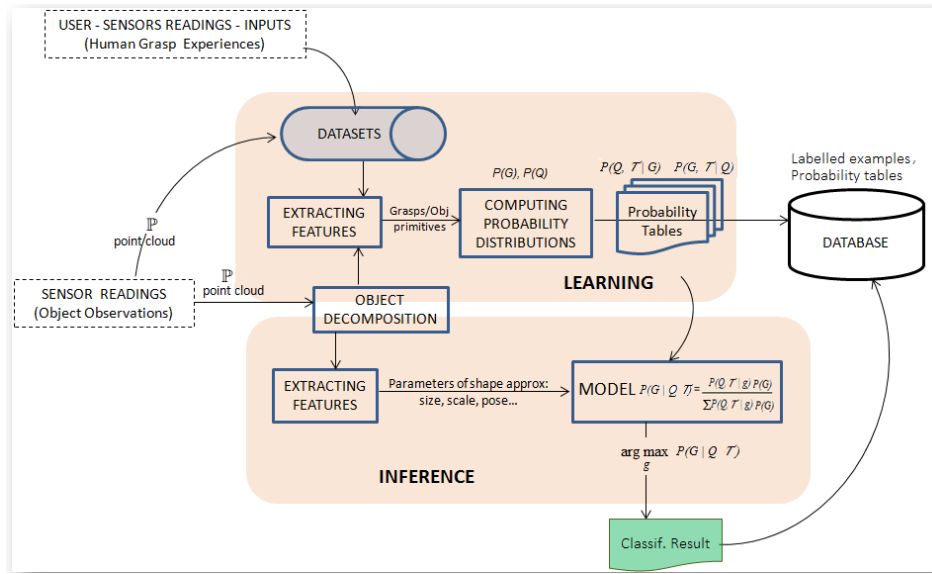


Figure 6.2: Overview of the learning process to assist the inference to search for candidate grasps for a given object model.

### 6.2.1 Overview of Bayesian Inference using the Learned Data

Here, we first give a general explanation of how we perform inference adopting Bayesian theory applied in the context of grasping. Later, in the next subsections on learning, we will show the models we defined that use the learned likelihoods useful for our grasp synthesis system, taking decisions on which are the most probable grasps and suitable regions for grasping given the object model.

Probabilistic techniques such as Bayesian theory is used to support the decision given the learned likelihood. A strong assumption is that all inputs are mutually independent of each other given the class label (e.g., grasps types associated with object regions under a task context). When adopting a Dynamic Bayesian Network (DBN) for the learned likelihood, the joint probability distribution is represented as a set of random variables. The set of parameters in a DBN encloses the conditional probability distribution of the random variables and the learned probability tables. Then using the Markov condition, each node is stated as independent of its non-descendants given its parents. Figure 6.3 presents a general example of an application in a grasp context to better understand the models. It represents the probability of a grasp type happening when some events occur, such as when the artificial system finds a specific object region, represented as a quadric  $q_i \in Q, i = \{1, \dots, n\}$ , and inside of a task context  $\mathcal{T}$ . The events (parents) represent the set of parameters  $Q, \mathcal{T}$  that trigger

an effect (node  $G$ ).

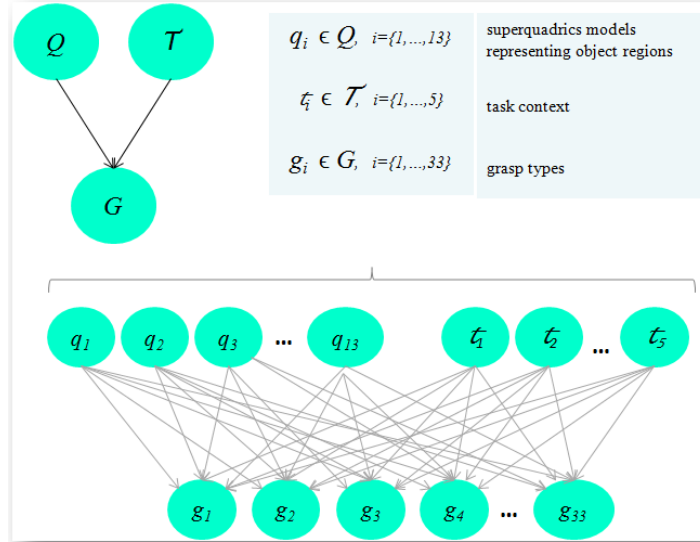


Figure 6.3: Bayesian network to represent a general model when a grasp type is estimated given some causes  $Q, T$ .

The DBN can be used as a classifier that gives the posterior probability distribution of the class node  $G$  given the values of other attributes (set of events). The model represented in Figure 6.3 can be expressed as the posterior distribution  $P(G|Q, T)$  given the observations enclosing the random variables  $Q, T$  as follows:

$$P(G|Q, T) = \frac{P(Q, T|G)P(G)}{\sum_j P(Q_j, T|G)P(G_j)}, \quad (6.1)$$

then, using the MAP estimation  $\underset{g_i \in G}{\operatorname{argmax}} P(G|QT)$ , we have the classification result. The dynamics in the BN is when the system's state at time  $t$  depends only on its immediate predecessor  $t - 1$ . The time instant  $t$  of the system evolves over time according to the system dynamics that is specified by the conditional density function  $P(G_{t+1}|G_t)$ .

The general models mentioned in this section were given to exemplify the techniques adopted for inference using the learned data to assist the grasp synthesis. The learned likelihoods used in this work were built based on histogram techniques. The idea is to rely on statistical data to achieve a successful estimate. The learning phases and inferences used in our artificial system are explained in the next subsections.

### 6.2.2 Experimental Setup for Data Acquisition

The experimental activities with humans executing grasping tasks are performed in our experimental area with multiple data acquisition devices (Figure 6.4) in order to capture how humans perform successful grasps. The data acquired is used to model and extract the relevant aspects of the human demonstrations, as well as providing input for the methods presented in this work. For that, we are reading data from two perspectives: the human hand and the object. In the first case, finger 6D pose using a magnetic tracking system and the tactile forces distributed on the inside of the hand are used. For the object, a point cloud model is used, obtained from in-hand exploration, as explained in Chapter 2, from an RGB-D sensor and also off-line from a laser scanner sensor. An online database, the Data Collection Database [HANa] of the HANDLE project, is publicly available with the datasets collected.



Figure 6.4: Sensors used in our experimental setup: Motion Tracking System, Tactile sensor and RGB-D sensor.

### 6.2.3 Learning Object Graspable Regions: Assigning Weights to Shape Primitives

In this section we address how to assign a weight (based on human statistics) to an object shape primitive for an initial grasping, without paying attention to the task context. The objective is to search for the shape primitive that has assigned more weight (between the three components of the object) to describe this region as suitable for grasping.

Through the human grasp demonstrations, we analyse the human choice to find the object graspable component given the three geometrical primitives that compose the object shape. We are biasing the geometrical primitives using the statistical data by quantifying the human grasp demon-



stration with a probability distribution based on histogram techniques. This way, given an unknown object and its three components, we will have weights distributed for each primitive to know which one is the best part as a candidate region for grasping. Afterwards, we have built a learned table with information of primitives preferences based on their weights. Adopting a verification of dual-combination of shapes, we verify the object components weights to know which component is the graspable part, by comparing the weight of the first object component with the second, later the component with the bigger weight from the previous verification is compared with the third component of the object. The learned table from the observations is a probability table of a dual-combination of geometrical primitives, so that later an estimate can be made to select which part of the object is the best one for grasping when the grasp synthesis system faces different geometrical primitives on the object.

A heuristic rule is also used for biasing the geometrical primitive representing the bottom part of the object if its pose is in a vertical position, which decreases the probability of the most suitable region for grasping. If the object pose is in a horizontal position, then this rule does not apply to the bottom part, because it can be a candidate for grasping in the same way as the other object regions.

During this learning process, we analyse the human preference for grasping given a set of primitives that can compose an object. A questionnaire was made to find out the human's choice given two shapes primitives. A set of geometrical shapes was shown to the subjects, and they were instructed to point (grasp) the shape that is the graspable choice (easier to grasp or could be grasped in more different ways than the other one) in the subject's point of view. The primitives used to observe the human's choice are the set of the defined superquadrics for this work as described in Chapter 3.

The set of shapes were shown to each subject (demonstrated two shapes per time), and to facilitate the human choice, the subject could grasp and interact with both shape primitives, and later the subject had to decide which one is preferable for grasping, if those two shapes were part of a single object. A system was developed (questionnaire) showing the two shapes that were demonstrated, and the subject should register in the system his choice (primitive 1 or primitive 2). An incremental function was computed for each primitive to build a histogram distribution. Each subject registered the choice for all possible combinations of the set of defined primitives.

The learning is achieved given the shape primitives (also referred to as quadrics)  $q_i \in Q$ ,  $i = \{1, \dots, n\}$  that compose an object, so that they are labelled when a graspable choice is made, for later computing the distribution of each labelled primitive. The histogram is computed as follows:

$$n = \sum_i^k \mathcal{H}_i, \quad (6.2)$$

where  $\mathcal{H}_i$  is a function that counts the number of observations that fall into each of the disjoint categories  $c$  representing dual-combination of the possible  $q_i$  (e.g., given an object, its components in dual representation is  $q_1 - q_2$  and  $q_i - q_3$ , where  $q_i$  can assume the form of  $q_1$  or  $q_2$ );  $k$  is the total number of categories and  $n$  is the total number of observations. Then the normalization to compute the probability distribution is achieved by:

$$P(c_k) = \frac{n_k}{N}, \quad (6.3)$$

where  $0 \leq c_k \leq 1$  (normalization of each category  $c_k$ , i.e., combination of quadrics);  $0 \leq k \leq K - 1$ , which  $K$  is the total number categories;  $N$  is the number of observations;  $n_k$  is the number of observations for each category;  $P(c_k)$  is the probability of the  $k_{th}$  category.

Figure 6.5 shows some examples of the statistical data represented in histograms. In this figure we can see some histograms of pairs of quadrics and the distribution of preference between both. We can also verify an example of the choice (preference) for the graspable quadrics when given a set of quadrics.

From these observations and statistics we could build a probability table (histogram), representing a learned table, as presented in Figure 6.6. To read this learned table, the axes  $\{x, y\}$  represent the possible pairs of the geometrical shapes (quadrics) that can compose a given object. The probability is assigned to the quadric  $q_i$  in  $x$  axis when it makes a pair with  $q_i$  in  $y$  axis. The color-map varies from 0 to 1 representing the probability (weight) of each quadric  $q_i$ .

Later an inference for an object graspable region can be made, taking into account the quadrics that form the object, as demonstrated next.

### Inference for Object Graspable Region given the Object Shape Primitives

We now address how to perform an inference to find the most probable graspable region on the object given a pair of primitives. The learned table presented in Section 6.2.3 (Figure 6.6) is used as likelihood in a Bayesian inference to update the probability of a graspable region of an object given the combination of quadrics (e.g. given the sequence of the components/quadrics  $q_i$  that compose an object:  $q_1$  and  $q_2$ ,  $q_1$  and  $q_3$ ,  $q_2$  and  $q_3$ , we can identify the graspable region for an initial grasp type (not taking into consideration the task context).

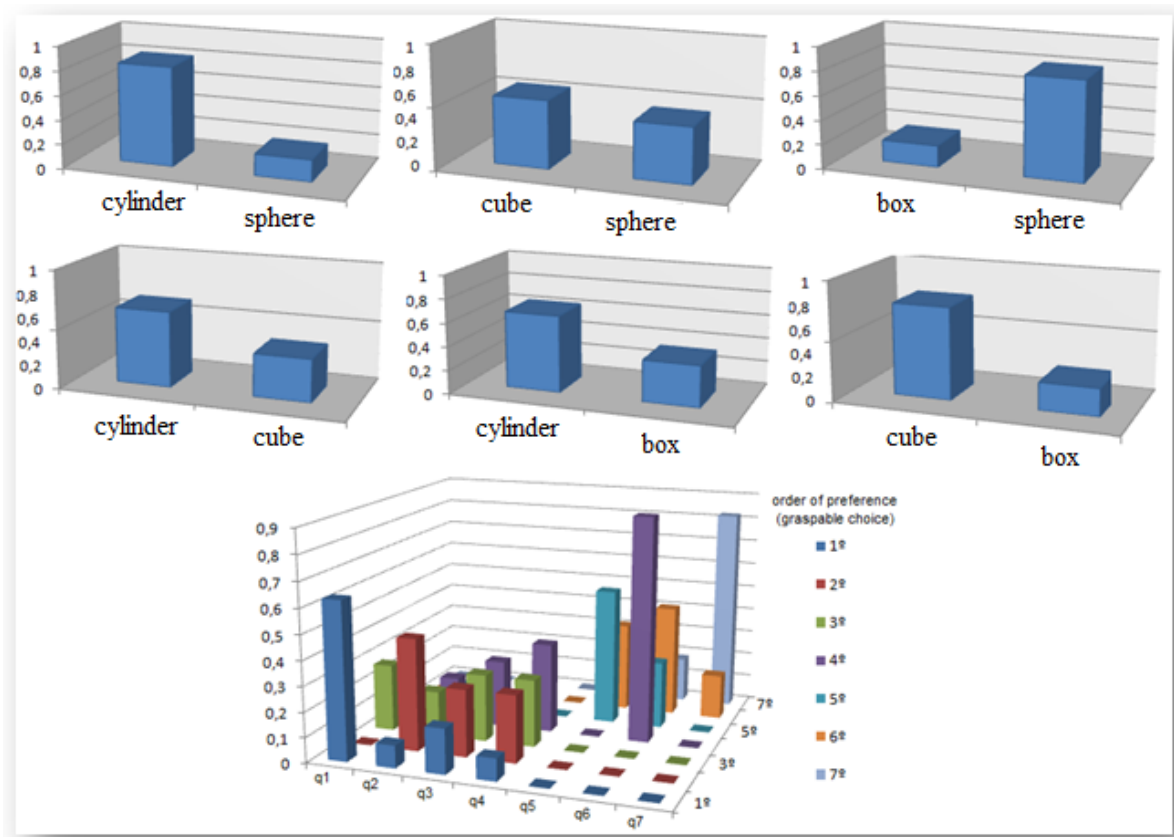


Figure 6.5: Examples of the statistical data acquired during the human grasp demonstrations. The statistics assist to weight the geometrical primitives as preference for grasping when dealing with a specific pair of quadrics.

The model for inference is given by:

$$P(Q = q_i | c_k) = \frac{P(c_k | Q = q_i)P(Q = q_i)}{\sum_j P(c_k | Q = q_j)P(Q = q_j)}, \quad (6.4)$$

where  $P(Q = q_i | c_k)$  is the probability of the object graspable region  $q_i$  given the combination (pair) of quadrics  $c_k$ . Then the classification is achieved according to the maximum a posteriori (MAP) estimate.

#### 6.2.4 Learning Suitable Objects Graspable Regions in Task-oriented Grasps

This learning process is to identify graspable regions on objects that are suitable for grasping given a task context. We are also using this learning process for a more consistent estimate of graspable regions to assist the grasp synthesis when the system is under a task context. The idea is given an object model and task context, using the knowledge learned from human grasp demonstrations, the artificial system can estimate the best region (primitives) to perform a proper grasp for that situation.

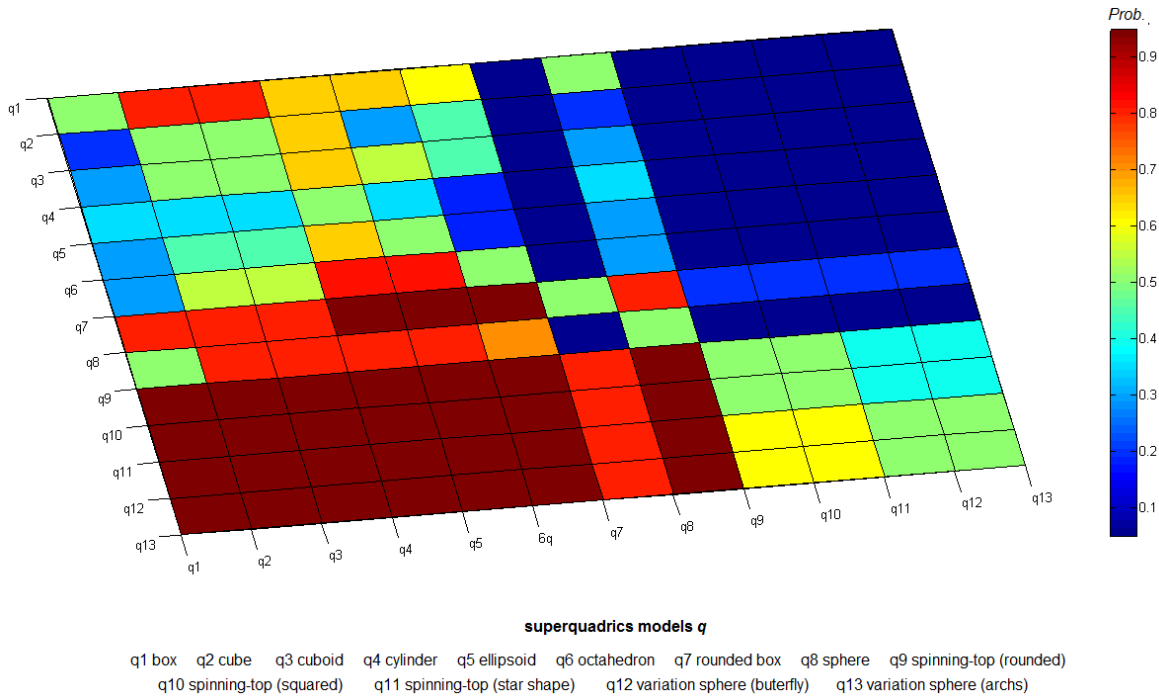


Figure 6.6: Probability Distribution in the Learned Table: Pointing the preference given a pair of quadrics  $q_i$ - $q_j$ ,  $i = \{1, \dots, n\}$ . The axes  $\{x, y\}$  represents all possible pairs of quadrics  $q_i$  that can compose an object. The probability assigned for the pair of quadrics is show through the color-map.

In this specific learning process, we have adapted our previous work with improvements for the grasp detection given the contact points on the object surface to quantify the human graspable choice.

In order to estimate the object region as graspable given a specific context (task-oriented), the combination of human demonstrations of stable grasps and object intrinsic information play an important role in the decision. In the learning process, we have the 3D object model of the object in a volumetric map, so that we can overlay the contact points of stable grasps on the object surface, represented in the grid cells of the object map. It also allows the identification of the grasp type by analysing the contact points locations forming the hand configuration. The probabilistic representation of object shape using 3D map, the grasp types detection using contact points, the overlaid contact points on the object map was presented in the previous Chapters 2 and 5.

For the human grasp demonstrations we are using from our experimental setup (Figure 6.4) the finger 6D poses using a magnetic tracking system, and the tactile forces distributed on the inside of the hand. Thus, with the volumetric information of the object we can overlay the contact points given by human demonstrations on the object surface. The contact points locations are given as 3D positions of the fingers (acquired by the magnetic tracker sensors) when a subject touches the object (i.e. the tactile sensors are active). The contact points locations are easily overlaid on the object

surface (cells in the object map), since we are working in the same frame of reference of the magnetic tracker allowing to map the contact points on the object surface. We consider that the system has previously acquired a 3D model of the object by in-hand exploration or other modality (e.g. vision) in order to have the volumetric model representation. Figures 6.7 and 6.8 show examples of contact points overlaid on the objects surface. The figures present objects grabbed by a human subject with a successful stable grasp during a manipulation task.

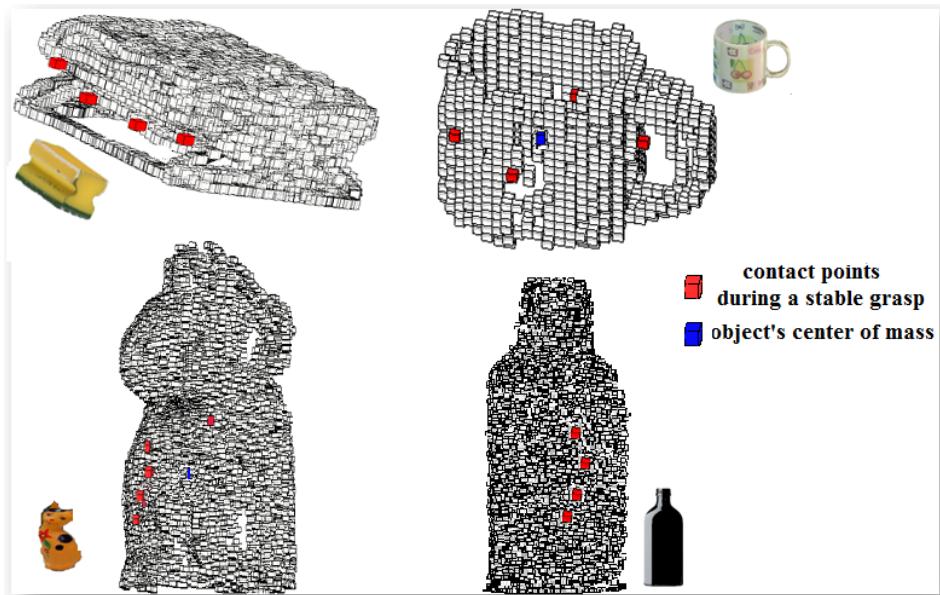


Figure 6.7: Examples of contact points of stable grasps from human demonstration on objects surfaces. The objects are: sponge, mug, wooden cat and bottle.

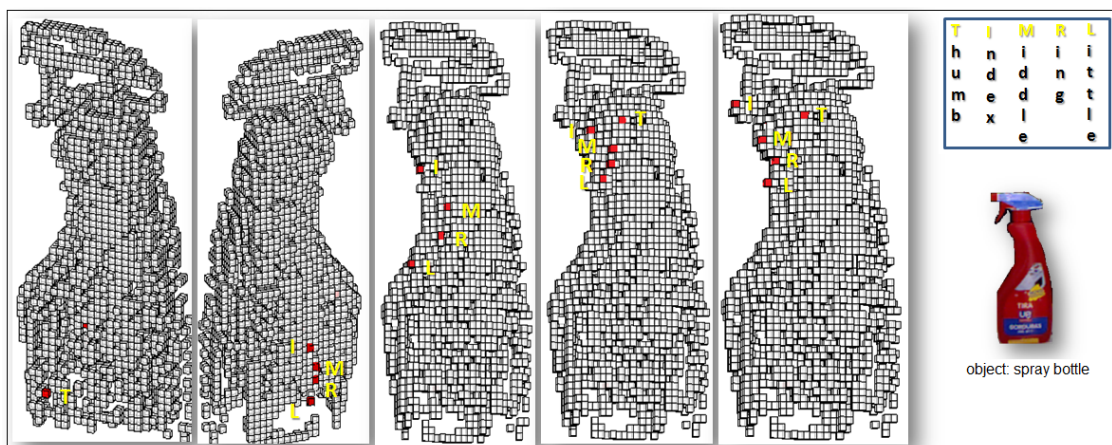


Figure 6.8: Examples of contact points of stable grasps from human demonstration on the object (spray bottle) surface.

In this work, we can label manually the grasp type by observing the hand configurations during

the grasp execution, or even in an automatic way, by computing the hand configurations using the contact points on the object surface. To identify a grasp type automatically, we rely on the fingertip 6D pose relative to the wrist as previously explained in Chapter 5.

After some trials of human demonstration on how to grasp an object given the objects models and the context, we could build a probability table to distinguish what kind of grasping is more probable to happen in each specific situation and also the object region that was chosen for the grasping.

Given a set of observations to represent a specific task  $\mathcal{T}$ , for instance, some simple tasks  $\mathcal{T} \in \{\text{pick-up and place; pick-up and lift; pick-up and pour/tilt}\}$ , we have the probability of each grasp type in a specific context represented as  $P(G|\mathcal{T})$ . The probability of each grasp type  $g_i \in G, i = \{1, \dots, n\}$  in a specific context is given by the frequency of observations as expressed below:

$$P(G = g_i) = \frac{o}{N}, \quad (6.5)$$

where  $o$  is the number of occurrences for the specific grasp type  $g_i$  and  $N$  is the total number of possible grasps  $G$ .

To identify the object graspable region, we verify the locations of the contact points on the object surface, and that region where the points are located represents a quadric model  $q_i$  (component of an object). Given a set of observations to represent a task  $\mathcal{T}$ , we have the probability of each object component being the object graspable region  $P(q_i|\mathcal{T})$ . It is computed in a similar way as shown in (6.5) where each component of the object has a probability associated with the graspable region given the context by computing the occurrences based on humans' choices for the object region defined as graspable.

Figure 6.9 shows some statistics computed after the human demonstrations for the chosen object graspable component for a few everyday objects (mug, bottle and wii-mote).

### Inference for Object Graspable Region in Task-oriented Grasps

The object graspable region can be identified applying the Bayes' theorem. Given a task context  $\mathcal{T}$ , to identify the object graspable region between the primitives that compose the object  $\{q_1, q_2, q_3\}$  as explained in Chapter 3, first it is necessary to detect the object components represented by quadrics models  $q_i$ . The probability distributions are obtained from the occurrence statistics acquired during the learning process used to build the likelihood.

Given a context  $\mathcal{T}$ , we can estimate the object graspable part  $q_i$  as follows:



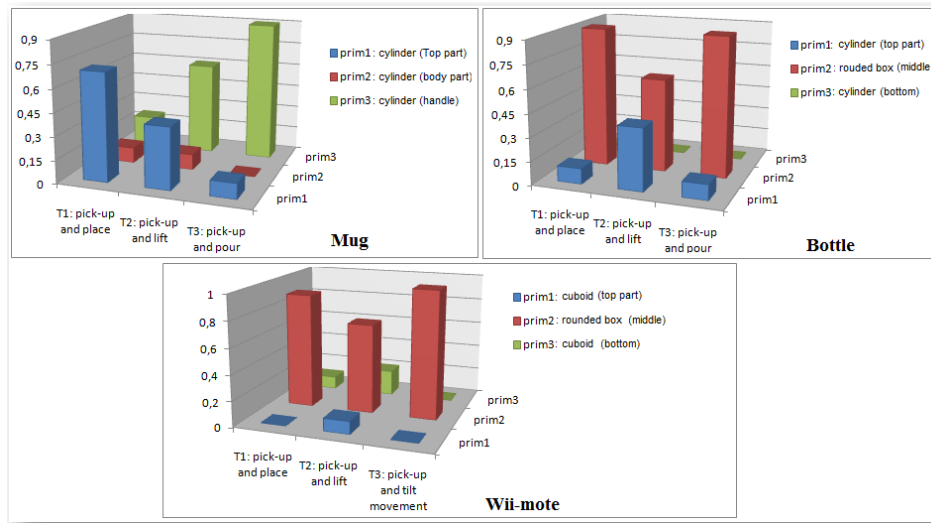


Figure 6.9: Statistics computed from the observations. Three different tasks performed many times by different individuals. By analysing the probability distribution of the chosen primitives to perform the grasp, the object graspable part (given the task context) can be estimated.

$$P(Q = q_i | \mathcal{T}) = \frac{P(\mathcal{T} | Q = q_i)P(Q = q_i)}{\sum_j P(\mathcal{T} | Q = q_j)P(Q = q_j)}, \quad (6.6)$$

where the posterior information  $P(Q = q_i | \mathcal{T})$  is computed for each primitive  $q_i$  of the object in a specific task  $\mathcal{T}$ ; the likelihood  $P(\mathcal{T} | Q = q_i)$  is the learned probability for each primitive of the object given a task context as previously explained. The normalisation factor is the sum of the probability of each object primitive being the graspable region.

A basic example of this application is given during the grasp planning, when a robot needs to execute a task. After detecting the object and its geometrical primitives, the robot can identify the object graspable region for possible suitable grasps, using the learned information from human demonstrations.

After learning a set of objects and task context, when the object is observed again in the same context, the system is able to detect the graspable part as shown in Figure 6.10. The graspable component is chosen according to the maximum a posteriori (MAP) estimate. In these specific examples, we have used just two components due to two reasons: (i) the third component for these specific objects had zero probability or a very low probability assigned to the third component inside the specific context; (ii) these specific objects still have a good representation even with two components. Indeed, just for a better visualisation of the results for these specific objects inside these specific contexts, we have adapted the results showing only the two more expressive components of the object. In a

real situation, we keep the three components of the object, even if one of the components has a zero probability assigned to it.

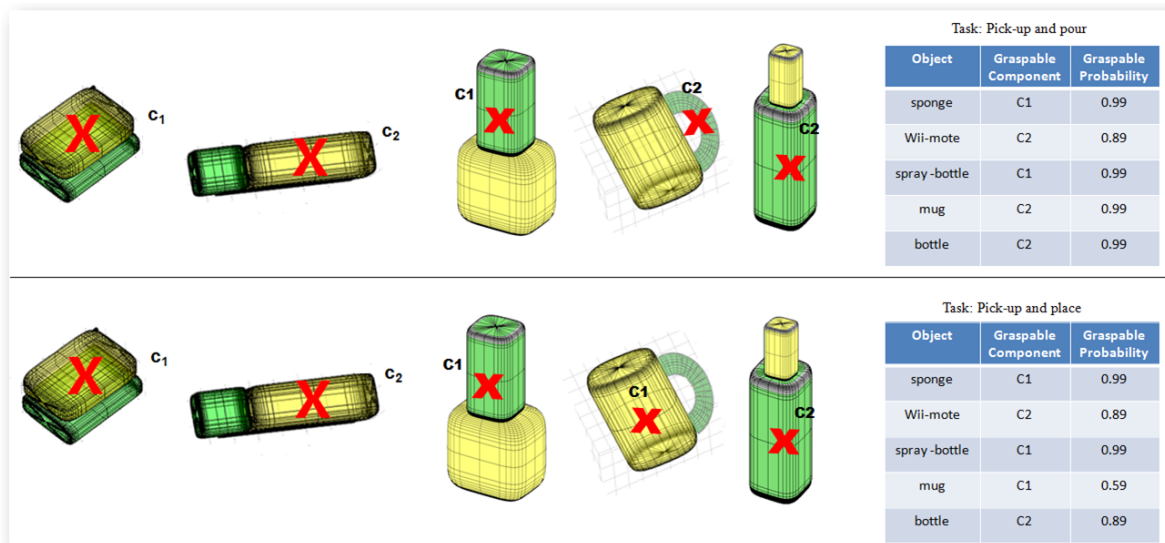


Figure 6.10: Identification of object graspable component for the sponge, *wii-mote*, spray-bottle, mug and bottle. For these trials we have used only two components for each object. Each component has a probability of being graspable, the maximum a posteriori estimate indicates the graspable component in each context.

In case of unknown objects, we have adopted a generalisation process, reusing the prior knowledge for other contexts, for instance, if a unknown object has one primitive in common with a known object, a similar grasp can be attempted. The unknown object falls to a familiar object, i.e. after the object segmentation process (applying the superquadrics model), this object will have known geometrical primitives. Given a task, a Bayesian classification as shown in (6.6) is computed for each object primitive to infer the most probable object primitive for that task.

The feasibility and the quality of the work is somehow dependent of how a given object is represented after the segmentation and how its components are matched to a specific model. This way, the system can generate the hypotheses of regions on objects being graspable, and for each primitive a set of grasp types is associated.

### 6.2.5 Learning Grasping Choice from Human Observations

From human grasp demonstrations we can also observe the grasps types that are assigned to the object regions. This way, we can build a set of possible candidate grasp types to a specific object or for specific geometrical primitives that compose this object.

The learning process is achieved given a dataset  $\mathcal{D}$  with  $N$  labelled examples of grasps types



associated with an object component. Since we have the grasp detection and object components detection, we can learn and associate a set of grasps with each geometrical primitive that can represent an object component.

Through histogram based learning, we quantify the probability table containing a set of grasps  $G$  for each quadric  $q_i \in Q$ . Since we defined a list of possible candidate grasps and the possible geometrical primitives, we have observed some grasps by human demonstrations for the defined geometrical primitives. It means that  $n$  grasps can be mapped to a specific shape, i.e.,  $G = \{g_1, \dots, g_n\} \mapsto q_i$ . Afterwards, a selection of the more probable candidates grasps for each geometrical primitive based on the probability distributions can be made.

The probability table that was built in this learning process can be seen as a 3D histogram, where each quadric  $q_i$  (in axis  $z$ ) has  $n$  grasps  $G$  (in axis  $x$ ) with probabilities associated with each grasp (axis  $y$ ) to be the most probable grasp for each quadric  $q_i$ . In this learning process, for each random variable, the distribution was normalized given the respective occurrences computed in a similar way as previously shown in (6.5).

As mentioned in the previous chapter, due to this learning process adopting histogram techniques, some features might have zero probability, because they never have been observed, i.e., a few grasps will never be applied to some specific shape. Whenever these features with zero probability occur in the classification step, the corresponding hypothesis will receive also a zero probability. Since for the inference, the classifier is continuous, based on a multiplicative update of beliefs, these zeros would lead to a definite out-rule of the hypothesis. To avoid this problem we are using the Laplace Succession Law to produce a minimum probability for non-observed evidences, in the same way as previously explained in Chapter 5, during the grasp movements recognition, but here with different type of variables as presented below:

$$\forall n_i = 0, P(n_i) = \frac{n_i + 1}{N + \chi} \Bigg|_{[n_i=0]} = \frac{1}{N + \chi}, \quad (6.7)$$

where  $P(n_i)$  is the resulting minimum probability that will be assigned to the non-observed grasping ( $n_i = 0$ );  $\chi$  represents the total number of features (i.e., all possible grasps types,  $G = 33$ );  $n_i$  is a specific feature (the non-observed grasps);  $N$  represents the total of occurrences (i.e., sum of all occurrences of features).

The probability table give us the likelihood useful for both estimation  $P(G|Q)$ , probability of

a grasping occurring given a geometrical shape, and  $P(Q|G)$ , probability to estimate a geometrical shape given the set of grasp types. However, in our specific case, we base on the statistical data to associate the most probable grasps with each quadric, using their probabilities as weights to set the preference of the candidate grasps.

Figure 6.11 shows the raw data representing the statistics from human demonstrations where 10 subjects (the majority men), all righted-hand, aged between 22-33 years old, demonstrated for each quadric the most probable grasps (from the grasp list [GRA]) with a minimum of 1 up to 10 grasps. The 10 subjects have demonstrated for the defined superquadrics models of this work, a total of 510 demonstrations (possible grasps). From this data we could verify different statistical information, such as the preferences and the mode of the samples. For instance, the superquadrics models that had more associated grasps, the top-5 are: cylinder, box, cube, cuboid, sphere, respectively. The grasps with more frequency during the demonstrations: g27-quadpod, g13-precision sphere, g1-large diameter, g3-medium wrap, g31-ring.

Figure 6.12 shows the learned table with the probability distribution after a normalization of the statistical data, which is useful for inference. The normalization for the likelihood is achieved by  $P(Q|G = g_i) = \frac{o}{N}$  where  $o$  is the occurrence of a specific grasp  $g_i$  during the human grasp demonstration and  $N$  is the total of demonstrations (all possible grasps) for a specific quadric  $q_i$ .

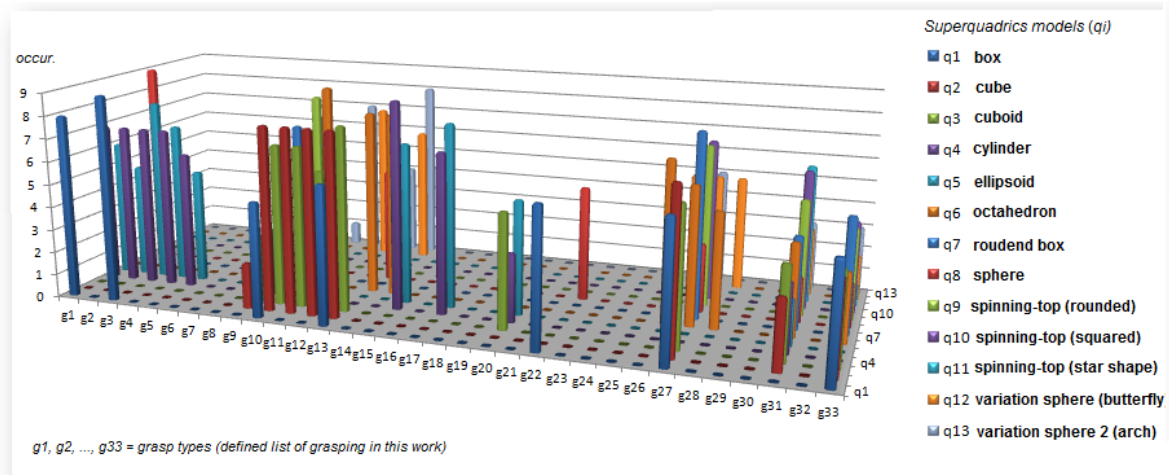


Figure 6.11: Grasp choice given the object quadrics: statistical data acquired by human demonstrations. The demonstrations were chosen from a grasp list [GRA] with 33 grasps types for 13 possible quadrics models  $Q = \{box, cube, cuboid, cylinder, ellipsoid, sphere, octahedron, rounded\ box, rounded\ spinning-top, squared\ spinning-top, star\ spinning-top, variation1-sphere(spherical\ arch), variation2-sphere\ (butterfly\ shape)\}$ .

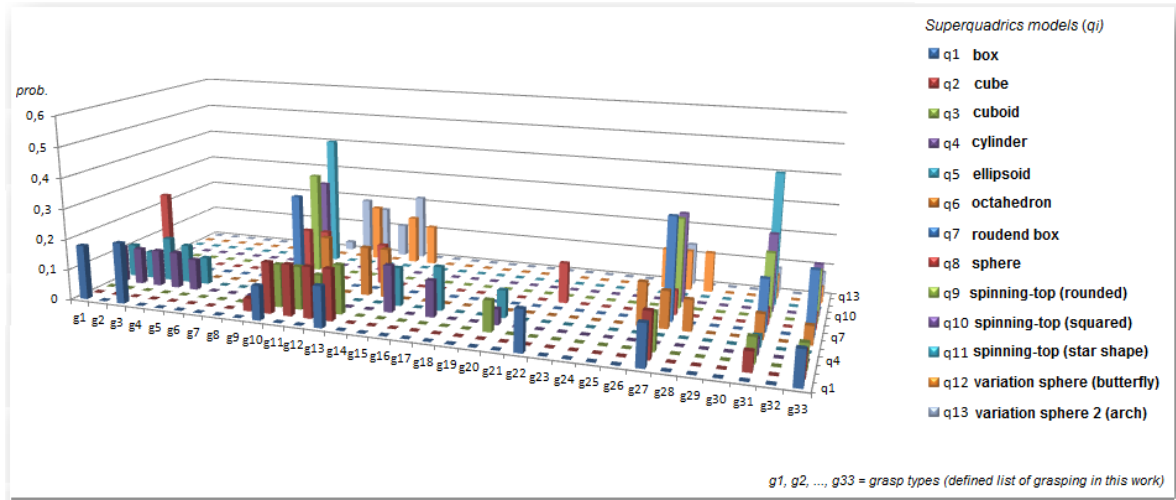


Figure 6.12: Probability Distribution: learned table from the statistics presented in Figure 6.11. Each grasp type has a probability of occurring given a quadric model.

### Inference for Grasping Choice

The inference over the learned table presented in Section 6.2.5 (Figure 6.12) is represented into two possible questions: first  $P(G|Q)$  meaning the probable grasp given one or more quadric model representing an object, and second  $P(Q|G)$  meaning the opposite, the most probable quadric model given a set of grasps. The first and second inference are computed adopting Bayes rule since we know the likelihoods and priors. The Bayesian inferences are demonstrated as follows:

$$P(G = g_i|Q, s) = \frac{P(Q, s|G = g_i)P(G = g_i)}{\sum_j P(Q, s|G = g_j)P(G = g_j)}, \quad (6.8)$$

$$P(Q = q_i|G, s) = \frac{P(G, s|Q = q_i)P(Q = q_i)}{\sum_j P(G, s|Q = q_j)P(Q = q_j)}, \quad (6.9)$$

where  $s$  represents a set of information (temporal), for instance: for (6.8) a set of quadrics to update the probability of the candidate grasps for an object composed of  $n$  quadrics; for (6.9) a set of grasps to update the probability of the possible quadrics.

The prior  $P(G)$  in (6.8) is achieved by another distribution function (probability table) as presented in subsection 6.2.3, considering a weight  $0 < w \leq 1$  based on the pair of quadrics. Indeed, it means that if a specific quadric has a higher weight than other, then the grasp associated with this quadric will have a higher probability. The prior  $P(Q)$  in (6.9) is a uniform distribution.

### 6.2.6 Learning from Object Observations

More information is extracted when dealing with the object model. Through observations of object components, after the detection given a point cloud  $\mathbb{P}$  from a specific sensor, we also learn and build a probability table by analysing some statistics. Adopting the same strategy of histogram based learning as explained before, we have built for each everyday object that we are dealing with in this work, a probability distribution taking into consideration the components of the object. Given a set of quadrics  $Q$  we have the probability distribution of each quadric  $q_i$  being the component of the object (i.e., belonging to an object region such as the top, middle or bottom).

Figure 6.13 shows an example of probability table of a specific object (spray bottle) demonstrating the probability of each quadric  $q_i$  being an object component. The same was done for other objects.

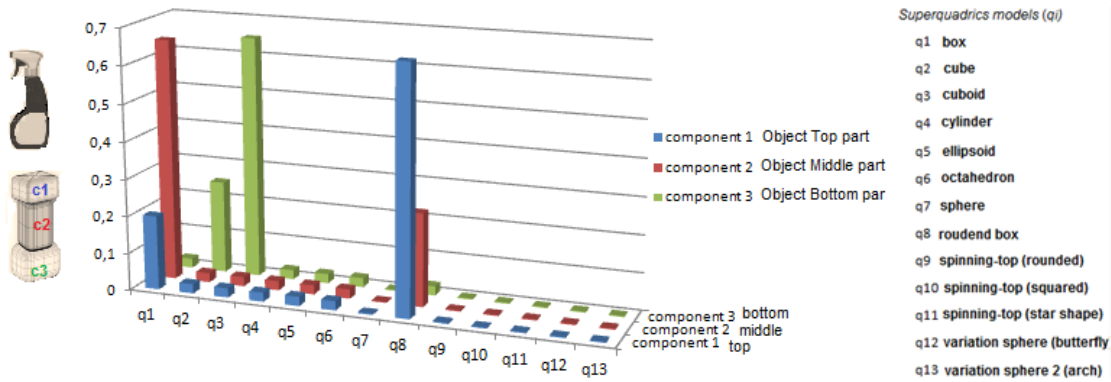


Figure 6.13: Spray Bottle - Probability Distribution of  $q_i$  being considered as an object component  $c_i$ .

Later an inference can be made to identify an object as demonstrated in the next subsection.

#### Inference for Object Identification

The inference to identify an object given the sequence of quadrics  $q_i$  following the order  $\{q_1, q_2, q_3\}$  is computed for all variables (i.e., all possible objects identities) using (6.10), and each one has a likelihood (Figure 6.13) representing the learned components of an everyday object. The identification is computed as follows:

$$P(O|Q, s) = \beta P(Q, s|O)P(O), \quad (6.10)$$

where  $P(O|Q, s)$  is the probability of an object identity given the sequence of quadrics representing each object component;  $\beta$  is the normalization factor,  $\beta = \frac{1}{\sum P(Q, s|O)P(O)}$  meaning the sum of all likeli-

hoods (learned table for all objects) for those 3 object components detected;  $s$  represents the sequence  $\{1, 2, 3\}$  of the detected quadrics.

With the learned information and using the inference for a limited set of objects, we have built a table containing the inference results, so that when the artificial system faces a novel object, it can detect a combination of 3 quadrics to identify the object or at least reasoning that it might be similar to one previously observed. Identifying an object is another alternative to find a graspable region, as well as grasp associated with this object or its components to find the candidate grasps.

### 6.2.7 Storing Learned Data

Since we have a limited number of objects, object components modelled with superquadrics models, task context and grasping types, we can restrict all possibilities of one or more random variables using the inference results and the learning data.

We have built tables storing the learned data, as well as tables with the inference results for the set of possibilities from the learned data. We can use this information to generalise, and apply in other contexts, or in case of grasping or objects, we can use similarities, i.e., find the most similar one to apply in a new context or to an unknown object.

This process was done in order to have sufficient data stored to facilitate and speed up the processes for a real time application during the execution, reducing then the processing time, since we want a system working in a feasible time for our application that can also be incorporated in other context (e.g., for in-hand manipulation tasks with grasp transitions).

## 6.3 Grasp Synthesis

The main purpose of the grasp synthesis is to find feasible grasps, given a 3D object model, and the robot end-effector configuration to maintain a stable grasp during the execution. Therefore, to accomplish this task, we first need to use the object characteristics to find the proper region for grasping, as well as the pose and configuration of the hand relative to the object to approach the object and successfully grasp it. Thus, we have developed an artificial system based on the idea presented in Figure 6.1. The experimental robotic system was a joint effort built within the HANDLE project consortium [HANb], and ROS (Robot Operating System) was used to combine contributions from all the project partners to have a working system. This work is one of the contributions integrated in the project. In the next subsections the modules that comprise the grasp synthesis system architecture are

presented, as well as the details of the system strategy implementation.

### 6.3.1 Using Decomposition Module in the Grasp Synthesis

The decomposition is used for the grasp synthesis objective. As explained in Chapter 3, given an object point cloud (unknown object), we first decompose the object into key components and later, inferences on the learned data to detect suitable regions for grasping and the proper configuration (grasp type using the GRASP taxonomy [GRA]) are made, as explained in Section 6.2.

In this work we have used as a pre-processing step to our system a ROS module named Extract Objects and Table developed by the HANDLE project consortium [HANb]. When the object point cloud is acquired by an RGB-D sensor, by using the ROS algorithms from the PCL (Point Cloud Library) [PCL], we can only extract the object model, ignoring the table and background. The object for manipulation is placed on the table in the sensor range. Algorithms like RANSAC (Random Sample Consensus) [FB81] are used to remove the tabletop, removing the non-interesting regions of the point cloud resulting only the object point cloud or a cluster of objects in case of many objects.

Following the ROS structure, the nodes communicate between each other by publishing messages to topics. A message is a simple data structure including types or arrays similar to the structures defined in C/C++ programming. The nodes were implemented in C++ (OOP - Object-oriented programming) adopting the ROS architecture. The object point cloud is then passed as a message to the Decomposition module. Then in the segmentation step, the object is converted into a new frame of reference (object-centred). The inputs for the second node of the Decomposition module (shape modelling) is a table of object segments (e.g., the segments  $\mathbb{P}_{top}$ ,  $\mathbb{P}_{mid}$ ,  $\mathbb{P}_{bot}$ ) that was published as a message to a specific topic by the first node (segmentation). The output of this second module is a published message into a topic containing the 15 parameters of the superquadrics model, representing the scale in each axis  $\{a_1, a_2, a_3\}$ , two parameters representing the superquadric shape  $\{\epsilon_1, \epsilon_2\}$ , three parameters representing the translation  $\{p_x, p_y, p_z\}$  and three angles representing the rotation  $\{\phi, \theta, \psi\}$  in each axis, as well as the centroid coordinates  $\{c_x, c_y, c_z\}$  and the volume of the quadric  $v_q$ . In the second node, since we achieved the object pose by the computation of the superquadrics models, we can use the extracted information to generate the candidate grasps for each quadric  $q_i$  of the object based on the learned data.

### 6.3.2 Grasp Synthesis Module

This module is in charge of searching for the proper candidate grasp, returning a list of grasp hypothesis given a 3D object model, as well as the correct hand pose to approach the object for grasping. All learned data that was stored are used in this module to assist the grasp generator to make inference over the data, as mentioned in Section 6.2. In our approach we decided to store the learned data and some inference results over some pre-defined situations such as grasp types associated with some geometrical models to gain time, reducing in this way the processing time. Later with the grasp type for a given object, we have to compute the hand pose relative to the object for the grasping execution.

The developed artificial system will always face an inference given the object information, these possible inferences were previously described in Section 6.2, which demands the use of the learned data from human grasp demonstrations for the estimate. The next subsection will present more details on the grasp list generation.

#### Grasps List Generation

When the artificial system receives the inputs coming from the decomposition module, the objective is to then have the candidate grasps for each part of the object.

The superquadrics parameters, the object centroid, object pose and scale (in the metrical superquadric coordinate system) are computed as demonstrated also in [JLS00]. This way, we know the object orientation and the limits of the object (width, height and depth), which allows the system to generate the possible candidate pre-grasp and the grasps near to the object boundary.

A discrete space-state is used as defined in the HANDLE project [HANb] for the frames of references (world for robot platform base, robotic hand and object) as presented in Figure 6.14.

Since we have computed the object pose during the decomposition module, and we know the defined hand state, then the system searches for the learned data in the database to find the possible candidate grasps for the detected superquadrics. Afterwards, for each grasp, the hand pose relative to the object is computed at pre-grasp position and for the selected grasp. We set the pre-grasp position of the hand at 10cm away from the object with neutral state (open hand) as shown in Figure 6.14. The hand pose for each grasp type is computed in the top and side-grasp pose relative to the object pose (i.e., the hand shape of the grasp is preserved, but we have two options for approaching the object, by its top or its side). Usually the top-grasp pose is the chosen one for simple tasks like *pick-up and place*. Some grasps are limited to the side position, such as adducted thumb or medium wrap, used

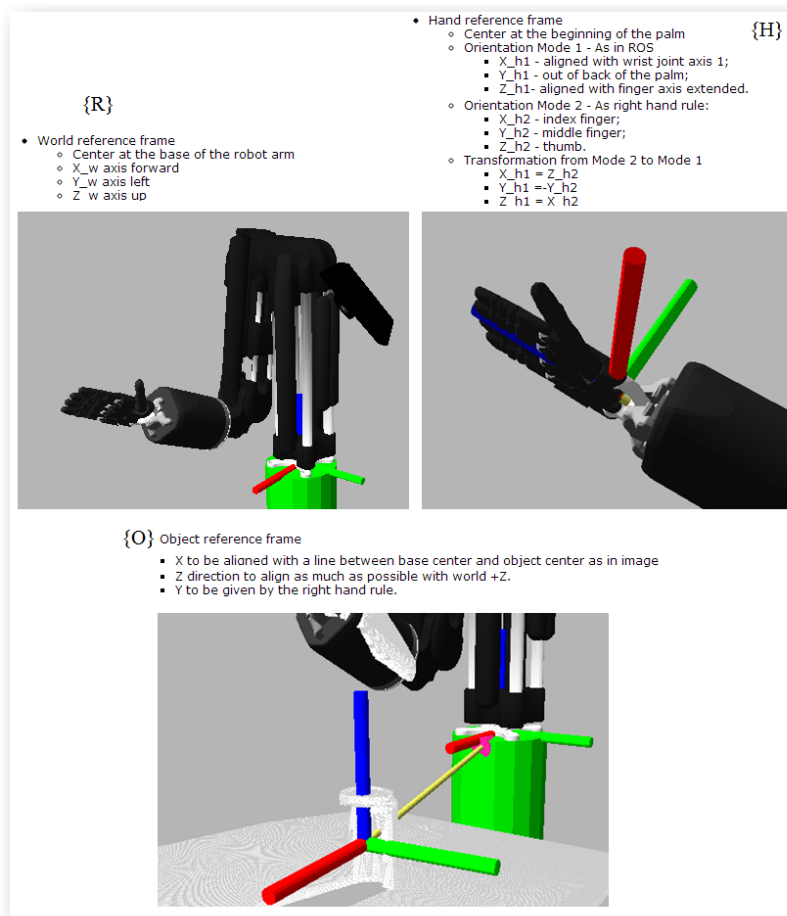


Figure 6.14: Frame of references adopted to generate the grasps pose relative to the object pose. Figure adapted from HANDLE Project Wiki page for the definitions of the ROS modules for the final demonstration of the project [HANb].

for example, to grasp a pipette by its side when it is on a vertical stand.

To generate the hand pose in top and side positions, the object pose and size, the frames of reference of the object and the hand are taken into consideration. The robotic platform consists in several joints and links as seen in Figure 6.14 (Shadow dexterous Hand [Sha]). A proper frame of reference structure was defined and thus the relations between the Shadow dexterous hand and object need to be calculated. The final goal for getting the transformation matrices was to be able to set the correct grasp pose of the hand relative to the object, in two different grasp directions: top  ${}^O\mathbf{T}_T$  and side  ${}^O\mathbf{T}_S$ . The following relations define these side and top grasp poses transformations respectively:

$${}^O\mathbf{T}_S = {}^O\mathbf{T}_H {}^H\mathbf{T}_R {}^R\mathbf{T}_S, \quad (6.11)$$



$${}^O\mathbf{T}_T = {}^O\mathbf{T}_R {}^H\mathbf{T}_R {}^R\mathbf{T}_T, \quad (6.12)$$

where  ${}^O\mathbf{T}_H$  defines the transformation between the frames of reference of the Object  $\{O\}$  and a hand wrist  $\{H\}$ ;  ${}^H\mathbf{T}_R$  defines the transformation between the frames of reference of the hand  $\{H\}$  and the actual frame of reference used in the robotic platform  $\{R\}$  for the hand, as illustrated in Figure 6.14;  ${}^R\mathbf{T}_S$  defines the transformation between the frames of reference of the robotic platform  $\{R\}$  and the side grasp position  $\{S\}$ ;  ${}^R\mathbf{T}_T$  defines the transformation between the frames of reference of the robotic platform  $\{R\}$  and the safe top grasp position  $\{T\}$ .

Below are the matrices that relate to these frames of reference:

$$\begin{aligned} {}^O\mathbf{T}_H &= \begin{bmatrix} & & & 0 \\ & R_{sq} & & 0 \\ & & & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}, & {}^H\mathbf{T}_R &= \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}, \\ {}^R\mathbf{T}_S &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & a_2 \\ 0 & 0 & 1 & -\Delta W_H \\ 1 & 1 & 1 & 1 \end{bmatrix}, & {}^R\mathbf{T}_T &= \begin{bmatrix} 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & a_3 \\ 0 & 0 & 1 & -\Delta W_H \\ 1 & 1 & 1 & 1 \end{bmatrix}, \end{aligned} \quad (6.13)$$

where  $a_2$  and  $a_3$  are the dimensions of the superquadrics that is necessary to set the grasp pose away from the object at a certain distance, and  $\Delta W_H$  is the distance from the wrist to the center of the hand.  $R_{sq}$  represents a rotation matrix based on the  $\{\phi, \theta, \psi\}$  (yaw-pitch-roll) angles extracted from the superquadrics components.

The grasp list is given by the grasps associated with each quadric  $q_i$  that composes the object as previously detailed in Section 6.2. After generating the grasp list and their poses for top and side approach, the artificial system will choose using a rank pool of weighed grasps, based on the learned probabilities. A high weight is assigned to the grasp in case of a success in a specific context, if a failure happens, the grasp will have a lower weight assigned to it in that context. Simulations were done a priori to test specific grasps in different hand poses and different contexts to assist the grasp rank pool to update the weights of each grasp in specific situations and with specific objects. The decision and learning of the rank pool module is based on Gaussian Process Regression implemented

and applied by a partner of the HANDLE project consortium as presented in [VB12].

The grasp execution by the robotic platform is performed after the acquisition of the object point cloud and all processes for the grasp generation. The grasp generator module encloses other modules beyond the scope of this work. In fact our module is one of three that compile to provide suitable grasps into a common rank pool from which a decision is made. The system also has a GUI (Graphical User Interface) to monitor and also interact when necessary to remove candidate grasps or take decisions.

For the robot execution (which is not the focus of this work), the mapping of the chosen grasp to the robotic platform takes into consideration the grasps hand pose, object pose and the robotic hand kinematics. The adopted approach is the one used by the HANDLE project to map the grasps to the robotic hand that has been implemented in the GraspIt! simulator [Mil01] and is called the *Eigengrasp-planner* [CGA07]. The consortium of the HANDLE project explores the use of grasp-synergies and then the *eigengrasps* for the Shadow robotic hand. The basic idea of grasping based on synergies is to combine a quick search of the reduced subspace spanned by the relevant *eigengrasps* with a later adjustment phase as a hierarchical approach where the synergies pre-shape the hand with approximate finger positions around the object. Sampling a large set of suitable (e.g. human-like) hand poses and performing the principal component analysis, the resulting set of eigenvectors provide a new basis of the hand joint space, where the set of eigenvectors is called as the synergies matrix. More details are given in Deliverable 24 (D24) of the HANDLE project [Hen12]. For each discrete grasp type used in this work, a mapping is made using this strategy, allowing later the correct grasp execution by the robotic hand.

Algorithm 4 presents the general idea of the grasp synthesis. The algorithm uses the methods explained in Chapter 3 for the object decomposition, the learned and inference processes explained in Section 6.2, and the grasp pose as explained in Section 6.3, following the respective order: Decomposition (object segmentation and shape modelling) and Grasp List Generation (candidate grasps given the object parameters and their poses relative to the object). The algorithm does not enclose the execution part, where the *eigengrasps* are used to map the discrete grasps to their correct configuration for the robotic hand performing the object grasping, since our goal is the grasp generation.

**Algorithm 4:** Grasp Synthesis Steps Before the Robot Execution

---

```

1 Input: Object Point Cloud  $\mathbb{P}$ 

   // Object Decomposition:
2  $pclouds[\mathbb{P}_{top}, \mathbb{P}_{mid}, \mathbb{P}_{bot}] \leftarrow \text{segment}(\mathbb{P})$ ;
3  $objParamList [a_1, a_2, a_3, e_1, e_2, x, y, z, \phi, \theta, \psi, c_x, c_y, c_z, v_q] \leftarrow \text{getShapeSQ}(pclouds)$ ;
   // Using the stored Learned data for inference given the object shape
   // parameters:
4  $grasps [] \leftarrow \text{getGraspList}(objParamList)$ ;
   // Generating Hand Pose (grasp list) relative to the Object Pose
5  $handPoses [] \leftarrow \text{genGraspPose}(grasps, objParamList)$ 

6 Outputs: Grasp List ( $grasps$ ) and their poses ( $handPoses$ ) relative to the object ( $\mathbb{P}$ )

```

---

## 6.4 Experimental Results

As previously mentioned, the artificial system was implemented under the ROS platform using C++ language to perform the grasp synthesis. The sensor used to acquire the object point cloud during the robot execution is an RGB-D camera. The processing time of the artificial system to run all algorithms described in this work for grasp synthesis (Decomposition and Grasp Generator) takes on average, under 2 seconds before executing the grasp. From all algorithms, the most time consuming is the shape modelling using superquadrics, since it depends on the size of the point cloud to compute the parameters. The learned data and the pre-processing inference enable us to have a fast decision over the object model. The next subsections show the results achieved using our proposed artificial system for grasp synthesis.

### 6.4.1 Simulated Tests

The first stage of tests of our application were performed in an off-line mode. Basically, we have simulated an application that triggers all modules (Decomposition and Grasp Generator), passing as input a point cloud previously acquired from different sensors. This way we could verify the consistence and the outputs of the system for those objects.

Figure 6.15 depicts the segmentation of an everyday object (unknown to the system). First image (top-left) shows the data (raw object point cloud) from the sensor (MS-Kinect) after removing the table-top. The left image shows the segmentation of the object during the decompose module.

Figure 6.16 shows some everyday objects just to exemplify the outputs of each module presenting the candidate grasps and hand pose (in top and side grasp orientation for the chosen grasp type)

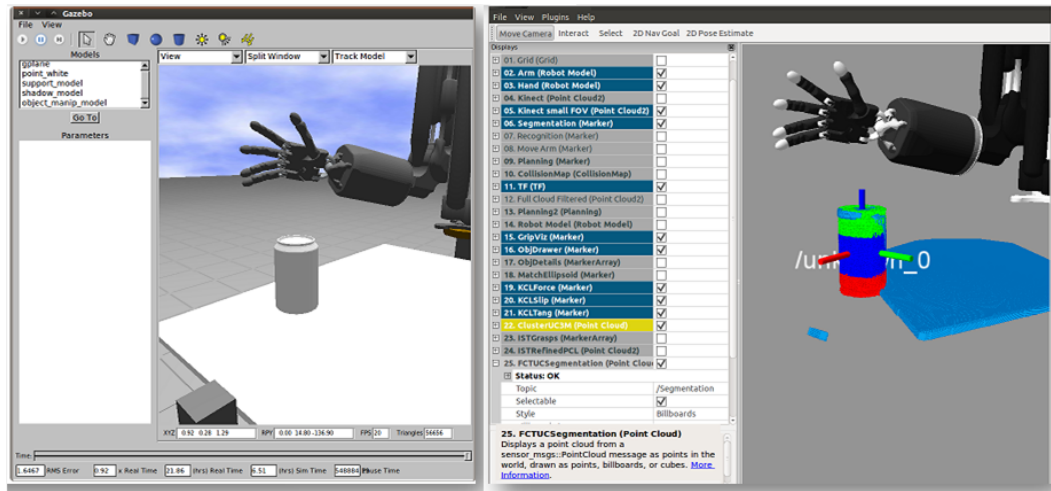


Figure 6.15: Decompose view in a simulator. The images show all steps for the segmentation. Left image represents the raw object data from the Kinect sensor after removing the table-top. The right image shows the segmentation result of the unknown object achieved by our decompose module.

of the object graspable parts. The generated grasp list follows an order of appearance indicating the grasp with highest weight (higher probability to be the selected grasp) down to the lowest one. The figure presents the result of our simulated tests for those objects.

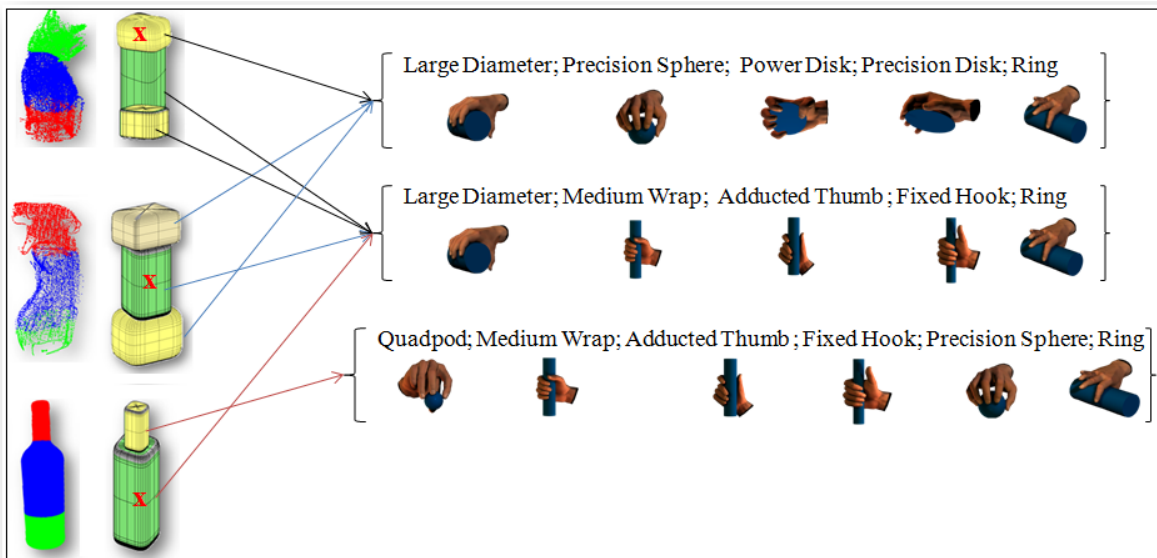


Figure 6.16: Results using the object point clouds to test our modules. The application returned some grasp associated with the geometrical shapes (quadrics  $q_i$ ) of the object. The marked quadrics in red are the object parts with higher probability to be the graspable part, and the grasps associated with this specific part have a higher weight. The order of appearance of the grasp types indicates the most probable grasp for that component.

Some assumptions need to be taken into account before sending the list of candidate grasps to the next module. We have to verify if the grasps are feasible for specific parts of the object. For

instance, if the graspable part is the middle component of the object, then, top-grasp orientation is not allowed for that component if it is in a vertical position, and the same happens to the bottom part. Some grasp configurations for the top and bottom parts of the object may be not proper, due to the size of the segmented part. So, after generating the grasp list, we verify if we need to discard some of the grasps generated.

In general, the results achieved from our simulated tests are suitable for the objects presented to the system.

#### 6.4.2 Tests in the Robotic Platform

The modules explained in this work are used to search for feasible grasps given a 3D object, these are then mapped to the robotic platform using the correct kinematics for the execution. Here, we are not dealing with the planning of the trajectory to approach the object, only the grasp type (and its pose) for the robotic platform. The trajectory planning (reaching movements) are addressed by other modules of the integrated system inside the HANDLE consortium [HANb].

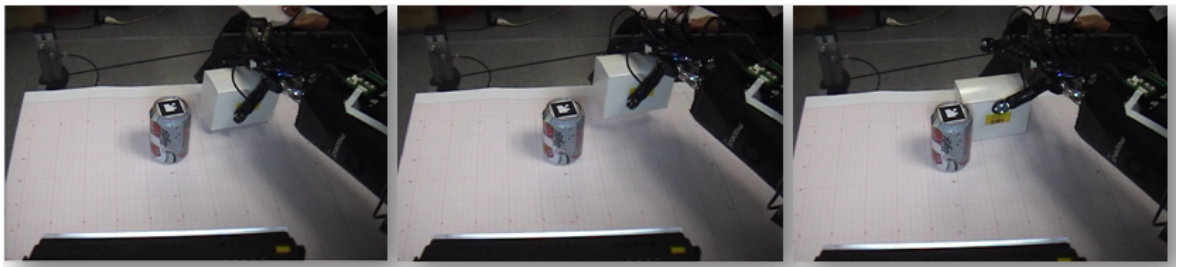


Figure 6.17: Selected grasp (grasp 27 from [GRA]: Quadpod) for the object (box) executed in a robotic platform.

The criteria that the robotic platform uses to chose a specific grasp was defined inside the HANDLE project [HANb] and it is explained as follows:

- If one of the provided grasps is a good starting point to reach the final grasp after in-hand manipulation that is set by the task (grasp transition sequence is easy);
- If the quality of the grasp is higher than those ones provided by other means (other grasp generators);
- If the grasp candidate (hand pose) is suitable for the Inverse Kinematics limits.

In general our modules provided good candidates that assisted the robotic platform in performing successful grasps. However some problems were encountered, such as the robotic hand pose not

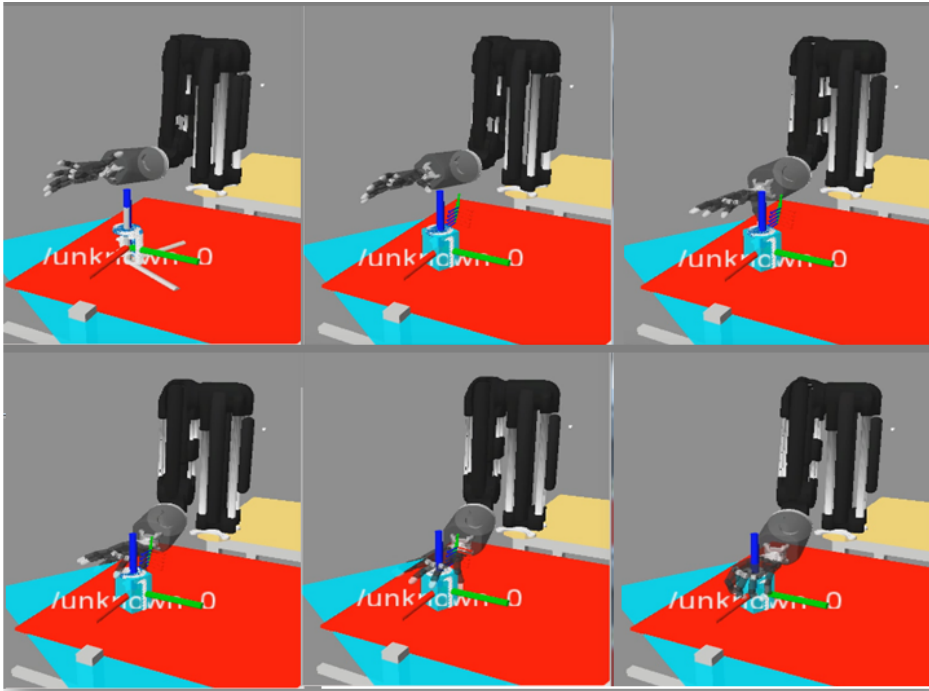


Figure 6.18: Selected grasp executed in a simulator before the execution in the robotic platform.

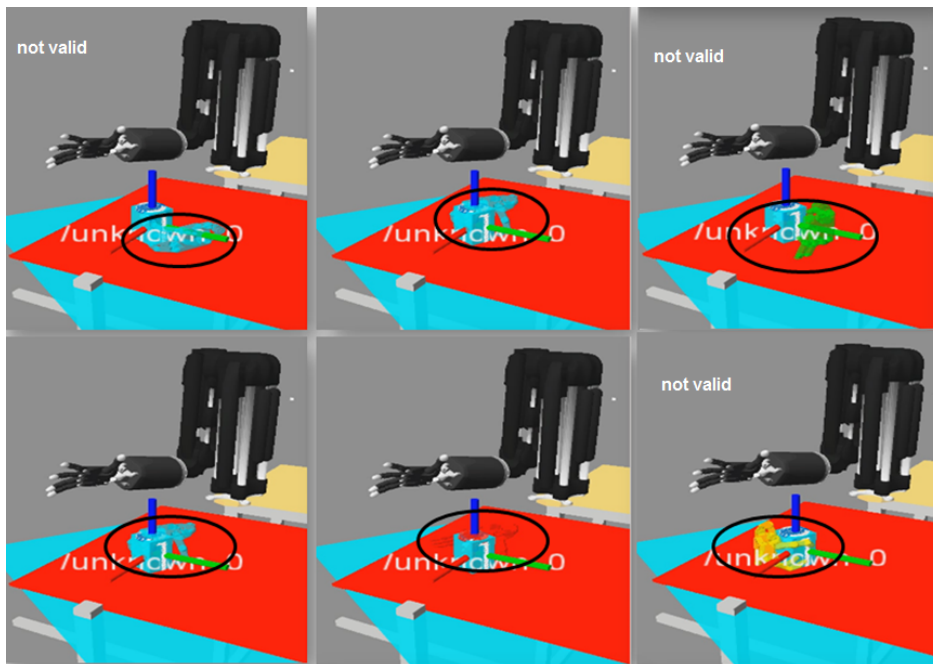


Figure 6.19: Tests in the simulator to verify the valid grasps of the generated list.

being reachable, because it is so close to the table, and finding a correct offset of the grasp to avoid the hand colliding with the object before completing the grasp. More trials will enable a better tuning and also to re-weight the candidate grasps based on the success rate.

Figure 6.17 shows an example of execution when one of the grasp of our modules was chosen as suitable for the object. The sequence shows when the robotic hand touches the object to perform the grasp, then the robotic hand lifts the object and finally releases it.

Figure 6.18 presents the simulation (inside the integrated software developed under HANDLE consortium [HANb]) to chose a specific grasp before the robotic platform execution. In this specific case our grasp was chosen and tested to validate the grasp.

Figure 6.19 shows a sequence of tests using the grasps from our system to indicate which grasps are valid and not valid.

The everyday objects used to test the modules are unknown to the system, so that we are applying the mentioned modules to approximate the object shape into familiar shapes to generate a set of candidate grasps for each known shape. In general, we can affirm that, after some improvements using offsets for hand pose to avoid some restrictions (table limit, hand kinematics), the grasp generator module is a solution to generate valid grasps for everyday objects.

## 6.5 Discussion

The knowledge acquired from both human demonstrations and object's properties allowed the development of an artificial system to respond with grasp synthesis given a novel object to be manipulated by a robotic dexterous hand. The developed artificial system faces a novel object as input, then it is segmented into meaningful parts to later be approximated into known shapes (superquadrics models). Afterwards the artificial system uses previous learned data of how to grasp known objects to be re-used for new objects. The proposed approach limits the amount of grasps for each known object primitive based on learned humans' choice.

The implemented modules generate a list of candidate grasps providing a ranked poll possible grasps. Results show that valid grasps may be generated for everyday objects to be used by a robotic dexterous hand. As future work we intend to implement re-weighting the candidate grasp list based on the success rate of the artificial system. We also intend to improve the grasp generation by re-weighting the grasp candidates after a simulation using the grasp hypothesis for a specific object. Improvements with offsets of hand pose relative to the object pose might be done to avoid some

collision between the robotic hand and table or robotic hand and object before the final grasping (i.e., fingers touching the object before the final hand pose).

The publications related to this chapter's subject are given as follows:

**Journal**

- Diego R. Faria, Ricardo Martins, Jorge Lobo, Jorge Dias. "Knowledge-based Reasoning from Human Grasp Demonstrations for Robot Grasp Synthesis". **Under Review** Robotics and Autonomous Systems, Elsevier, 2013.

**International Conference**

- Diego R. Faria, Ricardo Martins, Jorge Lobo, Jorge Dias. "A Probabilistic Framework to Detect Suitable Grasping Regions on Objects". In 10th IFAC Symposium on Robot Control (SYROCO 2012), Dubrovnik, Croatia, September, 2012.



## Chapter 7

# Overall Conclusions and Future Work



Probabilistic approaches for manipulation of everyday objects were proposed adopting multimodal data from human demonstrations. The learning process adopted is based on the relevant features extracted from the manipulation tasks. This is also useful to find the most probable object regions for successful grasping. The knowledge acquired from the data is useful for autonomous grasping by robotic systems. The outputs of this work can be used in different robotic applications, such as grasp planning and synthesis, imitation learning and etc. The constraints introduced by the presented models can be integrated by the applications during the estimation and synthesis of grasp and movements in different scenarios with everyday objects.

In this study, firstly (Chapter 2) presented how to perceive objects by the kinaesthetic capability, exploring the objects in-hand. Adopting a probabilistic volumetric map (occupancy grid techniques), the object global shape is represented during the exploration. The probabilistic map deals with the sensors uncertainty and real world noise. Bayesian estimation is then used to update each map cell state (i.e., occupied when the fingers pass through that location or empty when the fingers do not explore that region). The object probabilistic volumetric map was proposed to overlay the partially observed volume of the object with data about human visual gaze when initiating a grasp task,

hand-object contact points and tactile forces. During the object representation, the object centroid is computed to define the object frame of reference for object-centric representation. Two ways of in-hand exploration were presented: single hand exploration of static objects and in-hand exploration of non-static objects, when individuals usually use the left hand to assist the other hand for exploration. The results show that it is possible to achieve valid models of the object surface. In this approach, data from different sensors are allowed to improve the object model by means of fusing the multi-modal perception into a single percept. Results of this representation suggest its suitability for grasp planning since a unified model has the relevant observed information on how to grasp the object.

In addition to the object perception given the object volumetric representation, the object is segmented into components enabling the recognition by components (Chapter 3) for selection of object regions that are suitable for grasping. The segmentation of the object into components facilitates the matching by an artificial system observing objects to perform successful grasping taking into account the task context. Moreover, each component of the object is approximated (by fitting each segment of the object) into geometrical primitives given by the parametric modelling of superquadrics. This way, by simplifying the object shape we set the huge amount of candidate grasps into a limited list of possible grasps for that component. In this study, we are assuming that unknown objects, after segmented and associated with geometrical shapes, will have similar or familiar shapes compared to known objects, then allowing possible estimates of candidate grasps for these objects.

Afterwards, in Chapter 4, the capability of perceiving the objects through the hand is explored to have a better perception of the object, recognizing it. For that, the partial volume of the object acquired when an individual grasps the object in different ways is used. A learning stage is employed to associate possible grasps (hand configurations given the contact point on the object surface) with objects. The learning process is achieved by means of GMM/GMR. This application has shown that hand configurations (similar contact points) can be grouped into clusters and later by a regression model, a signature representing the object identity and grasp types associated with the object is achieved. Thus, among everyday objects, a pre-selection of candidate identities is made (i.e., hypothesis generation). Then, a matching between the partial point cloud of the object that is being explored and the full point clouds of objects selected from the database is made. The pre-selection step selects the more probable objects (those ones that can be grasped in the same way), avoiding the matching of the candidate object to all objects stored in the database.

The strategy adopted for in-hand exploration of objects shows that the perception acquired by

---

human hands (haptic: kinaesthetic, cutaneous and thermal) plays an important role when prehensile and manipulation activities acquiring objects' intrinsic and extrinsic information are performed to allow its representation and identification. Results are presented for human manipulation of objects, but the same can be mapped to artificial hands for object identification.

Proposed feature extractions of the multimodal data collected from human grasp demonstrations in manipulation tasks were presented. Based on these features, segmentation of the action phases and trajectory classification were accomplished. From the motion patterns, a generalized probabilistic representation for each type of task was derived. Results show the successful break down of action phases along a trajectory, as well as the suitability of the selected features as descriptors for the probabilistic approach used in task identification. This work with multimodal data also allows for the use of contact regions and tactile force intensities for grasp transitions classification, based on a set of grasp primitives as shown in [MFD10] and [FMLD12a] and can be used to correctly classify the grasp sequences in different tasks. The presented work starts from simple hand trajectories to more complex tasks involving in-hand manipulation of objects, and also shows how to use statistics of human demonstrations of grasps to estimate proper grasps and suitable regions on objects for grasping inside a task context or just for an initial grasp without considering the context. A consistent database with human demonstrations of manipulation tasks (from simple to more complex) was used to test the proposed methods. This study about hand trajectories, task identification and contact points of stable grasps can be used to endow an artificial dexterous hand to perform manipulation tasks.

All research developed in this thesis can be joint into a single framework to be used as knowledge for autonomous robotic grasping as explained in Chapter 1. In this thesis we have joint the knowledge acquired from human grasp demonstrations to design an artificial system for grasp synthesis (Chapter 6), where different strategies demonstrated in this thesis were encompassed. The system relies on the human grasp demonstrations using previous learned data of how to grasp known objects to be re-used for new objects. Therefore, in this study we have contributed to autonomous grasping by applying a probabilistic reasoning on the knowledge acquired from previous observations. Results show that valid grasps may be generated for everyday objects to be used by a robotic dexterous hand. Thus, relevant data was extracted to endow a robotic platform with enough capabilities for autonomous robot grasping.

## Future Works and Novel Concepts

As future work, we propose to extend our framework of grasp synthesis bringing together all knowledge studied in this research into a more complete framework. This extension will include knowledge of hand trajectories of reaching movements and handling tasks, in-hand exploration and object identification, grasp synthesis to be performed by an artificial dexterous hand. The idea exemplified in Chapter 1, Figure 1.1 will be developed for grasping planning and execution, or even to perceive the object by exploration, using the hands and visual sensing to identify and grasp novel objects.

Improvements will be studied for in-hand exploration combined with other modalities such as: using primitives from tactile to improve and update the cell status; improving the visual modality models - adopting stereo vision or using another device, such as RGB-D sensor. Other types of improvements will be investigated for object identification through in-hand exploration. Human studies can reveal different strategies to adapt and interact in different complex environments, so that we intend to base on human demonstrations. Statistical analysis on human data will allow robust inferences to be applied in grasping strategies.

A more thorough study on specific probabilistic reasoning techniques will be investigated to deal with the perception uncertainty when different modalities are adopted. The fact of going through this way is that, the sensory cooperation leads to a more robust and complete estimate of the surrounding environment.

A study for the development of some metrics to evaluate the execution of the approaches presented in this work will be addressed. In the same way, an analysis of the results to identify possible failures verifying the most relevant information (patterns) to accomplish a grasping task, taking into consideration its constraints, will also be carried out with the objective of contributing for autonomous grasping.

# Appendix A

## List of Publications

During the Ph.D. studies, the following papers were submitted/accepted for publication (peer reviewed international conferences and journals):

### Journals

- Diego R. Faria, Ricardo Martins, Jorge Lobo, Jorge Dias. "Extracting Data from Human Manipulation of Objects Towards Improving Autonomous Robotic Grasping". *Robotics and Autonomous Systems*, Elsevier, Volume 60, Issue 3, March 2012, Pages 396-410, 2012.
- Diego R. Faria, Pedro Trindade, Jorge Lobo, Jorge Dias. "Knowledge-based Reasoning from Human Grasp Demonstrations for Robot Grasp Synthesis". **Under Review**: *Robotics and Autonomous Systems*, Elsevier, 2013.

### International Conferences

- Diego R. Faria, Jorge Lobo and Jorge Dias. "Identifying Objects from Hand Configurations during In-hand Exploration". In proceedings of the 2012 IEEE International Conference on Multisensor Fusion and Information Integration (MFI 2012), Hamburg, September, 2012.
- Diego R. Faria, Ricardo Martins, Jorge Lobo, Jorge Dias. "A Probabilistic Framework to Detect Suitable Grasping Regions on Objects". In 10th IFAC Symposium on Robot Control (SYROCO 2012), Dubrovnik, Croatia, September, 2012.
- Ricardo Martins, Diego R. Faria, Jorge Dias. "Representation framework of perceived object softness characteristics for active robotic hand exploration". In Proceedings of 7th ACM/IEEE HRI'2012 - Workshop on Advances in Tactile Sensing and Touch based Human-Robot Interaction, Boston, USA, March 5-8, 2012
- Jafar Hosseini, Diego R. Faria, Jorge Lobo, Jorge Dias. "Probabilistic Classification of Grasping Behaviours using Visuo-haptic Perception". In Proceeding of 3rd Doctoral Conference on computing, Electrical and Industrial Systems (DoCEIS'12), Costa da Caparica, Portugal, 2012. IFIP Advances in Information and Communication Technology, Volume 372/2012, 241-248.
- Diego R. Faria, Ricardo Martins, Jorge Lobo, Jorge Dias. "Manipulative Tasks Identification by Learning and Generalizing Hand Motions". In Proceedings of DoCEIS'11 - 2nd Doctoral Conference on Computing, Electrical and Industrial Systems. Costa da Caparica - Portugal, February, 2011. IFIP Advances in Information and Communication Technology, Volume 349/2011, 173-180.

- Diego R. Faria, Ricardo Martins, Jorge Lobo, Jorge Dias. "Probabilistic Representation of 3D Object Shape by In-Hand Exploration". In Proceedings of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS'10 - Taipei, Taiwan - October 2010.
- Diego R. Faria, Ricardo Martins, Jorge Dias. "Learning Motion Patterns from Multiple Observations along the Actions Phases of Manipulative Tasks". Proceedings of Workshop on Grasping Planning and Task Learning by Imitation: IEEE/RSJ IROS'2010, - Taipei, Taiwan - October 2010.
- Ricardo Martins, Diego R. Faria, Jorge Dias. "Symbolic Level Generalization of In-hand Manipulation Tasks from Human Demonstrations using Tactile Data Information". Proceedings of Workshop on Grasping Planning and Task Learning by Imitation: IEEE/RSJ IROS'2010, - Taipei, Taiwan - October 2010.
- Diego R. Faria, Ricardo Martins, Jorge Dias. "Grasp Exploration for 3D Object Shape Representation using Probabilistic Map". in Proceedings of DoCEIS'10 - Doctoral Conference on Computing, Electrical and Industrial Systems. Costa da Caparica - Portugal, February, 2010. Springer - ISBN: 978-3-642-11627-8.
- Diego R. Faria, Jorge Dias. "3D Hand Trajectory Segmentation by Curvatures and Hand Orientation for Classification through a Probabilistic Approach". In Proceedings of the 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS'09, St. Louis, MO, USA - October 2009.
- Diego R. Faria, Jose Prado, Paulo Drews Jr., Jorge Dias. "Object Shape Retrieval through Grasping Exploration". In the 4th European Conference on Mobile Robots, ECRM'09, Mlini/Dubrovnik, Croatia, September 2009, pp.43-48.
- Diego R. Faria, Hadi Aliakbarpour, Jorge Dias. "Grasping Movements Recognition in 3D Space Using a Bayesian Approach", in Proceedings of ICAR'2009 - The 14th International Conference on Advanced Robotics - Munich, Germany, June 22-26, 2009. Print ISBN: 978-1-4244-4855-5.
- Diego R. Faria, Ricardo Martins, Jorge Dias, "Human reach-to-grasp generalization strategies: a Bayesian approach" - Workshop at Robotics Science and Systems 2009: "Understanding the Human Hand for Advancing Robotic Manipulation" - July 28, 2009 - Dillon Eng Seattle, WA, USA
- Diego R. Faria, Jorge Dias. "Bayesian Techniques for Hand Trajectory Classification", RECPAD 2008 - 14th Portuguese Conference on Pattern Recognition, Coimbra-Portgal, 31st October, 2008.
- Diego R. Faria, Jorge Dias. "Hand Trajectory Segmentation and Classification Using Bayesian Techniques", Workshop on "Grasp and Task Learning by Imitation" 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, Acropolis Convention Center, Nice, France Sept, 22-26, 2008, pp.44-49.
- Carlos Simplicio, Diego R. Faria, Jorge Dias. "3D Photo-realistic talking head for human-robot interaction". VR@P 2007 - 3rd International Conference on Advanced Research in Virtual and Rapid Prototyping, ESTG-Leiria, 24-29th September, 2007, Virtual and Rapid Manufacturing - Bartolo *et. Al.* (eds), 2008 Taylor and Francis Group, London, UK, ISBN 13: 978-0-415-41602-3.

## Appendix B

# Grasp List used in this Study

The grasps used for the object identification by the hand configurations and also for the grasp synthesis system are presented in Figure B.1. Each grasp taxonomy was defined after a study based on human grasps made by consortium within the European GRASP project [GRA].


































1	<a href="#">Large Diameter</a>		11	<a href="#">Power Sphere</a>		21	<a href="#">Tripod Variation</a>		31	<a href="#">Ring</a>	
2	<a href="#">Small Diameter</a>		12	<a href="#">Precision Disk</a>		22	<a href="#">Parallel Extension</a>		32	<a href="#">Ventral</a>	
3	<a href="#">Medium Wrap</a>		13	<a href="#">Precision Sphere</a>		23	<a href="#">Abduction Grip</a>		33	<a href="#">Inferior Pincer</a>	
4	<a href="#">Adducted Thumb</a>		14	<a href="#">Tripod</a>		24	<a href="#">Tip Pinch</a>				
5	<a href="#">Light Tool</a>		15	<a href="#">Fixed Hook</a>		25	<a href="#">Lateral Tripod</a>				
6	<a href="#">Prismatic 4 Finger</a>		16	<a href="#">Lateral</a>		26	<a href="#">Sphere 4 Finger</a>				
7	<a href="#">Prismatic 3 Finger</a>		17	<a href="#">Index Finger Extension</a>		27	<a href="#">Quadpod</a>				
8	<a href="#">Prismatic 2 Finger</a>		18	<a href="#">Extension Type</a>		28	<a href="#">Sphere 3 Finger</a>				
9	<a href="#">Palmar Pinch</a>		19	<a href="#">Distal Type</a>		29	<a href="#">Stick</a>				
10	<a href="#">Power Disk</a>		20	<a href="#">Writing Tripod</a>		30	<a href="#">Palmar</a>				

Figure B.1: Grasps list defined by the European GRASP project. More details and the complete taxonomy is available at [GRA].





## Appendix C

# SCamPol Toolbox for Matlab

A toolbox for Matlab was developed for the calibration of two different sensors to work in the same frame of reference. This toolbox was developed with the purpose of data fusion as explained in Chapter 2. The toolbox is available for download at the author website (<http://www.isr.uc.pt/~diego>). The calibration method supports 3D points from a specific sensor and a stereo camera. Here we are acquiring the 3D points from the Polhemus Liberty 240/8 tracking device and the Videre STH-MDCS3 stereo camera to achieve a transformation to re-project the 3D points from the tracker device into the image plane (stereo camera frame of reference) and vice-versa (3D points from stereo into the tracker frame of reference). For this calibration, we implemented a Matlab toolbox which, given a set of images from the stereo camera and the 3D points from the motion tracking device, it estimates the rigid transformation between the frame of references. The calibration assumes that the sensors are rigidly mounted in relation to each other. Figure C.1 shows a representation of a setup and the relevant frames of reference.

The calibration allows us to transform a 3D point in the local frame of reference of the tracker device into the stereo camera frame of reference. The first step of this calibration is to acquire the intrinsic and extrinsic parameters of the stereo camera, e.g., using the Bouget Camera Calibration toolbox (by Jean-Yves Bouguet, Camera calibration toolbox for Matlab: <http://www.vision.caltech.edu/bouguetj/calibdoc/index.html>).

By using a white tape attached to a sensor of the tracker device, we can recognize this marker in the image to obtain the corresponding 3D point from the stereo depth map, given the stereo camera calibration (see Figure C.2).

As previously explained in Chapter 2, the stereo camera and the tracker frame of references,

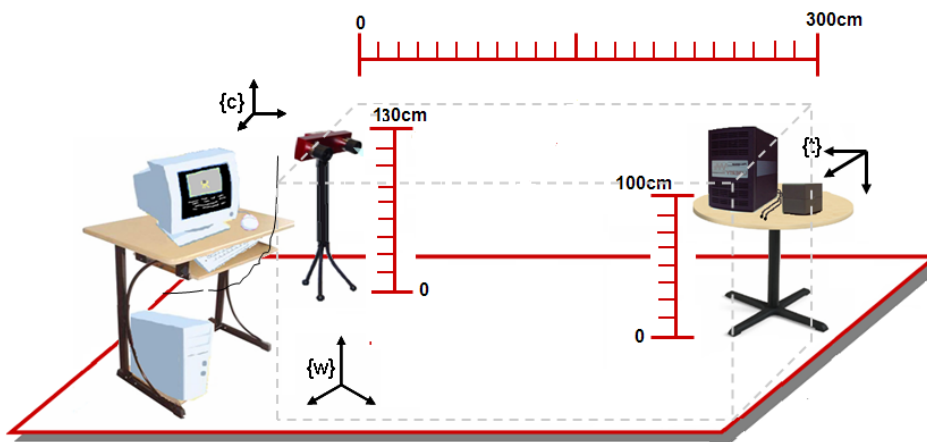


Figure C.1: Representation of an experimental setup with stereo camera and Polhemus rigidly mounted, and also the relevant frame of references.

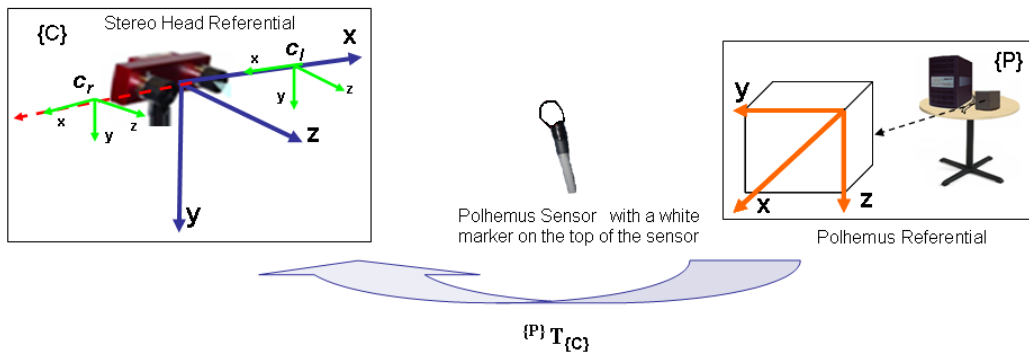


Figure C.2: Calibration strategy: using a white tape on the tracker sensor to acquire both 3D points, in the frame of reference of the tracker sensor, and also from the stereo camera by localizing the white mark in the images.

$\{C\}$  and  $\{P\}$  respectively, are rigid to each other. Collecting two sets of 3D corresponding points in two coordinate references, we can achieve the transformation to map a 3D point from  $\{P\}$  to  $\{C\}$ .

The Main menu of the SCamPol toolbox is presented in the graphic user interface in Figure

C.3.

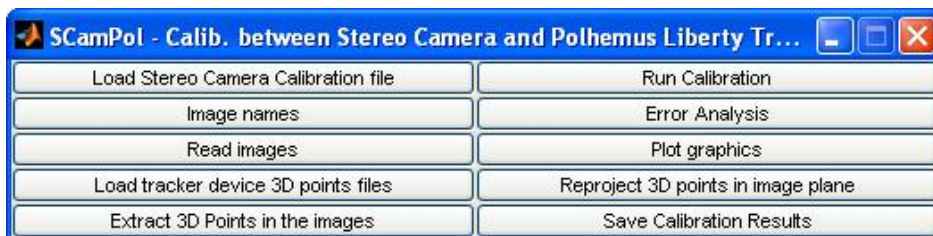


Figure C.3: Graphic User interface: Main menu of the SCamPol toolbox.

To use this toolbox, first it is necessary to have the stereo camera calibration data of the stereo

camera to be possible to compute the 3D points from the images (left and right). During the acquisition, the 3D points from the tracker sensor need to be acquired with the stereo image. For each acquisition two images (left and right) and the 3D point of the tracker is saved. The images have the sensor visible, for later to find the correspondence between the sensor coordinates at the left and right images, allowing to find the 3D location of the sensor by the stereo vision. For each acquisition is recommended that the sensors is placed in different locations in the experimental area to have a variance between the sensor translation and rotation. Note that the 3D data of the sensor need to be collected at same time that the left and right images are acquired. Figure C.4 shows an example of some acquired images with the sensors in different positions.

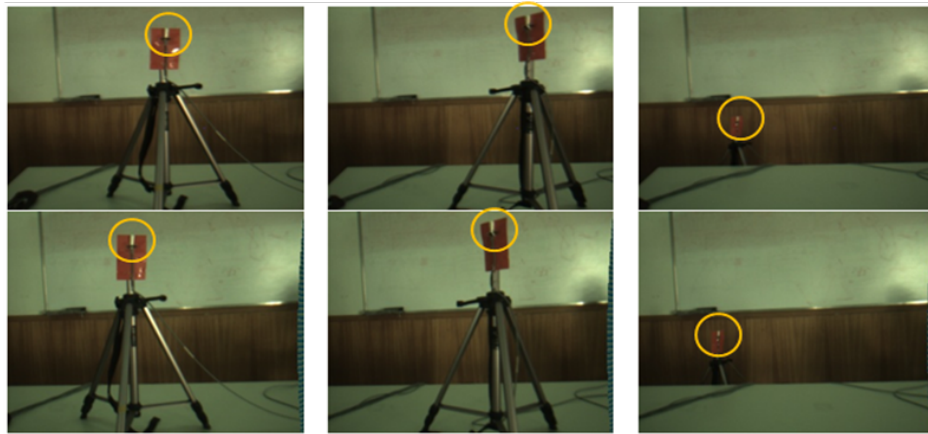


Figure C.4: Example of the images acquisition. The left images from the stereo camera are in the top row and the right images are in the bottom row. The yellow circles show the sensor location in the images.

The options of the SCamPol toolbox are described bellow.

**Load Stereo Camera Calibration File:** This option is used to load the stereo camera calibration file in order to be possible the computation of the frame of references transformation. The stereo camera calibration file is given by the camera calibration toolbox (Bouget) for Matlab.

**Images Names:** This option allows the users to define the base name of the images acquired for calibration. The images have be named with a base name for example, leftimage and rightimage. Afterwards the software assigns a sequential number to the names.

**Read Images:** All the images are loaded in the memory. A window with the loaded left images is opened for the visualization. The same happens with the right images as presented in Figure C.5.

**Load 3D points files from the tracker device:** For each loaded image (left and right) will have a corresponding 3D point that will be inside a text file (txt) acquired from motion tracker device

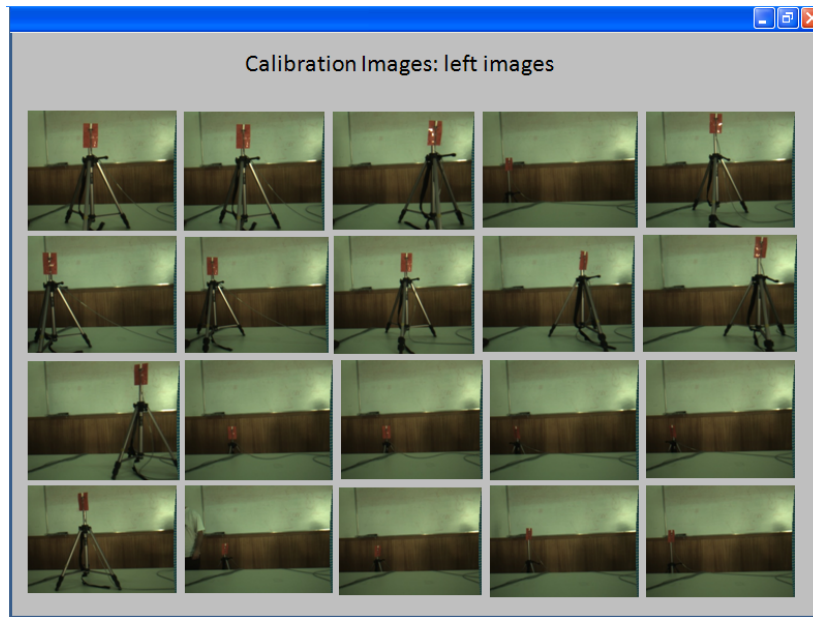


Figure C.5: Window View of the images acquired for calibration after being loaded.

with the  $x, y, z$  coordinate values. Each 3D point in the file is corresponding to each image in the sequence from 1 to  $n$ , that is, the first point in the file corresponds to the first image and so on.

**Extract 3D points from the images:** By selecting this option, the user will be asked to manually mark the motion tracking sensor position in the images (left and right). All image sequences (left and right) will be opened in order to select the region where the tracker sensor is located in the image coordinate. It is not necessary to select a precise location in the image, just the region where the sensor is located, and then the software automatically will find the top of the white mark on the sensor which corresponds its position  $x, y$  coordinates. Figure C.6 shows an example of the selection of the sensor location in the image by a user. After finishing this step, the software have the sensor positions in the images and it is possible compute the 3D position, given the stereo calibration parameters. The next step is to compute the calibration by achieving a transformation between the two frames of reference.

**Run the Calibration:** This option allows the users to compute the calibration between the stereo camera and the motion tracker sensor. After running the calibration step, the software automatically re-projects a 3D point of the tracker sensor on the corresponding image plane (left image).

**Error Analysis:** This option allows the users to evaluate the quality of the calibration result. A table will be shown with the evolution of the errors along the images; it is the average re-projection error values in pixels according to the number of 3D points that were used. The average error of the



Figure C.6: Selection window: selection of the sensor region in the image to find its coordinate in the images (left and right) to allow the computation of the 3D point.

proposed calibration decreases when the method uses a higher number of points. It is possible to consider that for  $N = 20$  points, the calibration method is stable. A graphic of the re-projection errors also will be displayed after choosing this option. Figure C.7 shows a graphic with the number of the points that were used in the calibration, and the value of re-projection errors in the scale of pixel.

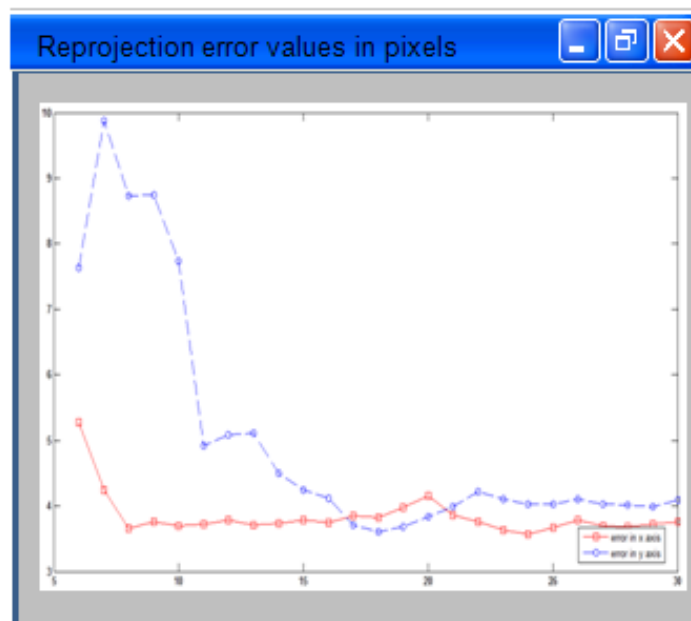


Figure C.7: Graphic generated by the toolbox with the number of points used in the calibration and the value of re-projection errors in the scale of pixels.

**Re-project in the Images Plane:** By selecting this option, it is possible to re-project the 3D points of the Polhemus tracker device in the image plane. It is necessary to select the image for re-projection, the 3D point file of the sensor, and the calibration file (transformation matrix) generated

by the SCamPol toolbox. An example of re-projection is shown in Figure C.8. In this example a person moved the sensor generating the 3D points on the object surface.

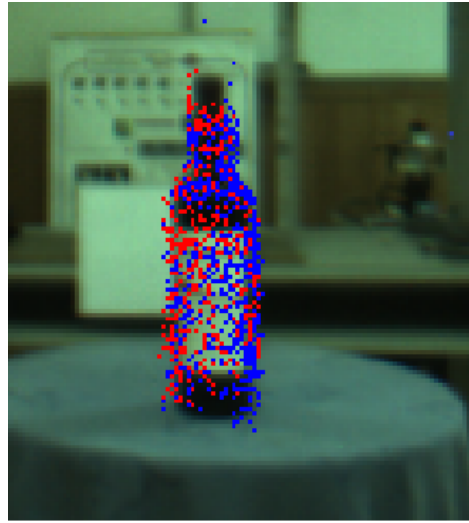


Figure C.8: Re-projection example: 3D points acquired from the tracker sensor in the image plane.

**Plots:** This option allows the users of to generate three possible plots: one for the number of points used in the calibration and the value of re-projection errors in the scale of pixel, another showing the evolution of the rotation matrix, and the last one shows the translation matrix estimated by the calibration according to the number of points used. Figure 49 shows the three possible graphics that can be generated for errors analyzes.

**Save Calibration:** This option start up the process to save the calibration file with the rotation and translation matrices of the homogeneous transformation.

# Bibliography

- [AA88] H. Asada and Y. Asari. The direct teaching of tool manipulation skills via the impedance identification of human motion. In *IEEE ICRA*, 1988.
- [AC08] J. Aleotti and S. Caselli. Programming task-oriented grasps by demonstration in virtual reality. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, WS on Grasp and Task Learning by Imitation*, 2008.
- [AHB87] K. S. Arun, T. S. Huang, and S. D. Blostein. Least-squares fitting of two 3D point sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(5):698–700, 1987.
- [ANP<sup>+</sup>09] H. Aliakbarpour, P. Núñez, J. Prado, K. Khoshhal, and J. Dias. An efficient algorithm for extrinsic calibration between a 3d laser range finder and a stereo camera for surveillance. In *ICAR 2009 - 14th International Conference on Advanced Robotics - Munich, Germany.*, 2009.
- [Bar81] Alan H. Barr. Superquadrics and angle preserving transformations. In *IEEE Computer Graphics and Applications*, 1981.
- [Bie87] I. Biederman. Recognition-by-components: A theory of human image understanding. *Psychological Review*, 94:115–147, 1987.
- [BK10a] Jeannette Bohg and Danica Kragic. Learning grasping points with shape context. *Robotics and Autonomous Systems*, 58(4):362–377, 2010.
- [BK10b] Jeannette Bohg and Danica Kragic. Learning grasping points with shape context. *Robotics and Autonomous Systems*, 58(4):362–377, 2010.
- [BM92] P.J. Besl and H.D. McKay. A method for registration of 3D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14:239–256, 1992.

- [Bou] Jean-Yves Bouguet. Camera calibration toolbox for matlab: <http://www.vision.caltech.edu/bouguetj/calibdoc/index.html>.
- [Bun94] W. L. Buntine. Operations for learning with graphical models. *Journal of Artificial Intelligence Research (AI Access Foundation)*, 2:159–225, 1994.
- [BV07] G. Biegelbauer and M. Vincze. Efficient 3D object detection by fitting superquadrics to range image data for robot’s object manipulation. In *international Conference on Robotics and Automation, ICRA’07*, 2007.
- [BWB<sup>+</sup>07] A. Bierbaum, K. Welke, D. Burger, T. Asfour, and R. Dillmann. A framework for visually guided haptic exploration with five finger hands. In *Proceedings of the Robotics: Science & Systems, Manipulation Workshop - Sensing and Adapting to the Real World*, 2007.
- [Cas05] Umberto Castiello. The neuroscience of grasping. *Nature*, 6:726–736, 2005.
- [CDB10] Francis Colas, Julien Diard, and Pierre Bessière. Common bayesian models for common cognitive issues. *Acta Biotheoretica*, 58(1-2):191–216, 2010.
- [CG06] L. Chen and N.D. Georganas. An efficient and robust algorithm for 3D mesh segmentation. *Multimedia Tools Appl.*, 29(2):109–125, 2006.
- [CGA07] M. Ciocarlie, C. Goldfeder, and P.K. Allen. Dimensionality reduction for hand-independent dexterous robotic grasping. In *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3270–3275, 2007.
- [CGB07] S. Calinon, F. Guenter, and A. Billard. On learning, representing, and generalizing a task in a humanoid robot. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 37(2):286–298, 2007.
- [CGJ96] D. Cohn, Z. Ghahramani, and M. Jordan. Active learning with statistical models. *Artificial Intelligence Research*, 4:129–145, 1996.
- [CJ91] U. Castiello and M. Jeannerod. Measuring time to awareness. *Neuroreport*, 12:787–800, 1991.



- [CJB03] L. Chevalier, F. Jaillet, and A. Baskurt. Segmentation and superquadric modeling of 3D objects. *WSCG*, 11:232–239, 2003.
- [CSPB11] S. Chitta, J. Sturm, M. Piccoli, and W. Burgard. Tactile sensing for mobile manipulation. *IEEE Transactions on Robotics*, 27(3):558–568, June 2011.
- [CT91] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley & Sons, 1991. ISBN-13: 978-0471062592.
- [dGSF06] Charles de Granville, Joshua Southerland, and Andrew H. Fagg. Learning grasp affordances through human demonstration. In *International Conference on Development and Learning*, 2006.
- [DLSX00] D. Ding, Y. Liu, Y.T. Shen, and G.L. Xiang. An efficient algorithm for computing a 3D formclosure grasp. In *In Proceedings of IEEE International Conference on Robotics and Automation*, 2000.
- [DPR92] S.J. Dickson, A.P. Pentland, and A. Resenfeld. From volumes to views: An approach to 3D object recognition. *CVGIP: Image Understanding*, 55:130–154, 1992.
- [EB04] Marc O. Ernst and Heinrich H. Bülthoff. Merging the senses into a robust percept. *Trends in Cogn Science*, 8(4):162–169, 2004.
- [EK04] S. Ekvall and D. Kragic. Interactive grasp learning based on human demonstration. In *IEEE/RSJ International Conference on Robotics and Automation*, 2004.
- [EKS10] Sahar El-Khoury and Anis Sahbani. A new strategy combining empirical and analytical approaches for grasping unknown 3d objects. *robotics and Autonomous Systems*, 58(5):497–507, 2010.
- [EKSP07] S. El-Khoury, A. Sahbani, and V. Perdureau. Learning the natural grasping component of an unknown object. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS07)*. San Diego, CA, USA., pages 2957–2962, 2007.
- [Elf89] A. Elfes. Using occupancy grids for mobile robot perception and navigation. *IEEE Computer*, 22:46–57, 1989.

- [FB81] Martin A. Fischler and Robert C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM.*, 24(6):381–395, 1981.
- [FBJ06] J Randall Flanagan, Miles C. Bowman, and Roland S. Johansson. Control strategies in object manipulation tasks. *Current Opinion in Neurobiology*, 16(6):650 – 659, 2006. Motor systems / Neurobiology of behaviour.
- [FCBD12] J. F. Ferreira, M. Castelo-Branco, and J. Dias. A hierarchical bayesian framework for multimodal active perception. *Adaptive Behavior*, 20:172–190, 2012.
- [FD13] João Filipe Ferreira and Jorge Miranda Dias. *Probabilistic Approaches to Robotic Perception*. Springer Tracts in Advanced Robotics, Vol. 91, 2013.
- [FdSH98] M. Fischer, P. Van der Smagt, and G. Hirzinger. Learning techniques in a dataglove based telemanipulation system for the dlr hand. In *IEEE International Conference on Robotics and Automation*, 1998.
- [FM99] W. Forstner and B. Moonen. A metric for covariance matrices. Technical report, Dpt. of Geodesy and Geoinformatics - University of Stuttgart, 1999.
- [FMLD10] D. R. Faria, R. Martins, J. Lobo, and J. Dias. Probabilistic representation of 3D object shape by in-hand exploration. In *IEEE/RSJ Int. Conf. Intelligent Robots and Systems, Taipei-Taiwan*, 2010.
- [FMLD12a] D. R. Faria, R. Martins, J. Lobo, and J. Dias. Extracting data from human manipulation of objects towards improving autonomous robotic grasping. *Robotics and Autonomous Systems - Special Issue on autonomous grasping*, 60(3):396–410, March 2012.
- [FMLD12b] Diego R. Faria, Ricardo Martins, Jorge Lobo, and Jorge Dias. A probabilistic framework to detect suitable grasping regions on objects. In *10th IFAC Symposium on robot Control (SYROCO'12), Dubrovnik, Croatia, Sept.*, 2012.
- [FPS<sup>+</sup>09] Thomas Feix, Roland Pawlik, Heinz-Bodo Schmedmayer, Javier Romero, and Danica Kragic. A comprehensive grasp taxonomy. In *Robotics, Science and Systems: Workshop on Understanding the Human Hand for Advancing Robotic Manipulation*, 2009.

- [GALP07] C. Goldfeder, P. K. Allen, C. Lackner, and R. Pelossof. Grasp planning via decomposition trees. In *IEEE Int. Conf. on Robotics and Automation*, 2007.
- [GB93] A. Gupta and R. Bajcsy. Volumetric segmentation of range images of 3d objects using superquadric models. *CVGIP: Image Understanding*, 58:302–323, 1993.
- [GC11] Maya R. Gupta and Yihua Chen. Theory and use of the EM algorithm. *Foundations and Trends in Signal Processing*, 4(3):223–296, March 2011.
- [GMB<sup>+</sup>94] M.A. Goodale, J.P. Meenan, H.H. Bulthoff, D. A. Nicolle, K. J. Murphy, and C.I. Racicot. Separate neural pathways for the visual analysis of object shape in perception and prehension. *Current Biology*, 4:–, 1994.
- [GMJC91] M.A. Goodale, A.D. Milner, L.S. Jakobson, and D.P Carey. A neurological dissociation between perceiving objects and grasping them. *Nature*, 349:154–156, 1991.
- [GNGW10] Nicolas Gorges, Stefan Escaida Navarro, Dirk Goger, and Heinz Worn. Haptic object recognition using passive joints and haptic key features. In *IEEE International Conference on Robotics and Automation*, 2010.
- [GRA] GRASP project: Emergence of cognitive grasping through introspection, emulation and surprise. Website: <http://grasp.xief.net/> (Visited September, 2012).
- [HANa] Data Collection Database at ISR-UC, HANDLE project. Website: <http://mrl.isr.uc.pt/experimentaldata/datasets/handle/> (Visited March, 2012).
- [HANb] HANDLE project: Developmental pathway towards autonomy and dexterity in robot in-hand manipulation. website: <http://www.handle-project.eu> (Visited January, 2012).
- [HBZ06] M. Hueser, T. Baier, and J. Zhang. Learning of demonstrated grasping skills by stereoscopic tracking of human hand configuration. In *IEEE International Conference on Robotics and Automation*, 2006.
- [Hen12] Norman Hendrich. Deliverable 24: Parameterizing and creating new actions. In *HANDLE project, D24*, <http://www.handle-project.eu/>, 2012.
- [HK08] K. Hubner and D. Kragic. Selection of robot pre-grasps using box-based shape approximation. In *IEEE Int. Conf. on Intelligent Robots and Systems*, 2008.

- [Jea84] M. Jeannerod. The timing of natural prehension movements. *Journal of Motor Behavior*, 16:235–254, 1984.
- [Jea88] M. Jeannerod. *The neural and behavioral organization of goal directed movements*. Oxford, Clarendon Press., 1988. ISBN-13: 978-0198521969.
- [JLS00] Ales Jaklic, Ales Leonardis, and Franc Solina. *Segmentation and Recovery of Superquadrics*, volume 20 of *Computational imaging and vision*. Kluwer, Dordrecht, 2000. ISBN 0-7923-6601-8.
- [JNR<sup>+</sup>10] P. Drews Jr., P. Nunez, R. Rocha, M. Campos, and J. Dias. Novelty detection and 3d shape retrieval using superquadrics and multi-scale sampling for autonomous mobile robots. In *Proc. of IEEE ICRA'10*, 2010.
- [KDCI10] Robert Krug, Dimitar Nikolaev Dimitrov, Krzysztof Andrzej Charusta, and Boyko Iliev. On the efficient computation of independent contact regions for force closure grasps. In *IROS2010 International Conference on Intelligent Robots and Systems*, pages 586–591, 2010.
- [KFUM09] H. Kawasaki, T. Furukawa, S. Ueki, and T. Mouri. Robot teaching based on motion analysis and hand manipulability for multi-fingered robot. *Journal of Advanced Mechanical Design, Systems, and Manufacturing*, 3:1–12, 2009.
- [KHB<sup>+</sup>10] V. Kruger, D. Herzog, S. Baby, A. Ude, and D. Kragic. Learning actions from observations. *Robotics Automation Magazine, IEEE*, 17(2):30–43, 2010.
- [KI95] S.B. Kang and K. Ikeuchi. Toward automatic robot instruction from perception-temporal segmentation of tasks from human hand motion. *IEEE Transactions on Robotics and Automation*, 11(5):670–681, 1995.
- [KL90] Roberta L. Klatzky and Susan Lederman. Intelligent exploration by the human hand. *Dextrous Robot Hand*. ed.S.T. Venkataraman and T. Iberall Springer-Verlag., -:-, 1990.
- [KLM85] Roberta L. Klatzky, Susan Lederman, and V. Metzger. Identifying objects by touch: An "expert system". *Perception & Psychophysics*, 37(4):299–302, 1985.

- [KP04] D. C. Knill and A. Pouget. The bayesian brain: the role of uncertainty in neural coding and computation. *TRENDS in Neurosciences*, 27:712–719, 2004.
- [KUU08] Masahiro Kondo, Jun Ueda, and Tsukasa Ogasawara. Recognition of in-hand manipulation using contact state transition for multifingered robot hand control. *Robotics and Autonomous Systems*, 56:66–81, January 2008.
- [KWSN05] F. Kyota, T. Watabe, S. Saito, and M. Nakajima. Detection and evaluation of grasping positions for autonomous agents. In *International Conference on Cyberworlds*, pages 453-460, 2005.
- [LDSA05] E. Lopez-Damian, D. Sidobre, and R. Alami. Grasp planning for non-convex objects. In *36th International Symposium on Robotics, ISR.*, 2005.
- [LDW99] Y.H. Liu, D. Ding, and S. Wang. Constructing 3D frictional form-closure grasps of polyhedral objects. *IEEE Transactions on Robotics and Automation*, -:-, 1999.
- [Liu00] Y.H. Liu. Computing n-finger form-closure grasps on polygonal objects. *International Journal of Robotics Research*, 19(2):149–158, 2000.
- [LJS97] A. Leonardis, A. Jaklic, and F. Solina. Superquadrics for segmentation and modeling range data. *IEEE Transactions on Pattern Anal Mach Intell.*, 19:1289–1295, 1997.
- [LK87] Susan Lederman and Roberta L. Klatzky. Hand movements: A window into haptic object recognition. *Cognitive Psychology*, 19:342–368, 1987.
- [LP05] Y. Li and N. Pollard. A shape matching algorithm for synthesizing humanlike enveloping grasps. In *5th IEEE-RAS Int. Conf. on Humanoid Robots*, pages 442–449, 2005.
- [LSN<sup>+</sup>11] Hongbin Liu, Xiaojing Song, Thrishantha Nanayakkara, Kaspar Althoefer, and Lakmal Seneviratne. Friction estimation based object surface classification for intelligent manipulation. In *IEEE International Conference on Robotics and Automation 2011 - Workshop on Autonomous Grasping*, 2011.
- [MFD10] R. Martins, D. R. Faria, and J. Dias. Symbolic level generalization of in-hand manipulation tasks from human demonstrations using tactile data information. In *IEEE/RSJ*

- IROS'2010: Workshop on Grasping Planning and Task Learning by Imitation, Taipei, Taiwan.*, 2010.
- [Mil01] Andrew T. Miller. *GraspIt!: A Versatile Simulator for Robotic Grasping*. PhD thesis, Ph.D. Thesis, Department of Computer Science, Columbia University, June, 2001.
- [MKCA03] Andrew T. Miller, Steffen Knoop, Henrik I. Christensen, and Peter K. Allen. Automatic grasp planning using shape primitives. In *Proceeding of International Conference on Robotics and Automation, ICRA 2003, Sep. 14-19. Tapei, pp. 1824-1829*, 2003.
- [MLBSV08] Luis Montesano, Manuel Lopes, Alexandre Bernardino, and Jose Santos-Victor. Learning object affordances: From sensory motor maps to imitation. *IEEE Transactions on Robotics*, 24(1):15–26, 2008.
- [MN78] D. Marr and H.K. Nishihara. Representation and recognition of three dimensional shapes. In *Proceedings of the Royal Society of London, Series B. 200:269–294*, 1978.
- [Mor88] H. P. Moravec. Sensor fusion in certainty grids for mobile robots. *AI Magazine* 9, 2:61–74, 1988.
- [MP77] R.S. Millman and G.D. Parker. *Elements of differential geometry*. -, 1977. ISBN-13: 978-0132641432.
- [MW99] A. Mangan and R. Whitaker. Partitioning 3D surface meshes using watershed segmentation. *IEEE Trans Vis Comput Graph.*, 5:308–321, 1999.
- [Nap80] John Napier. *Hands*. Princeton University Press, 1980.
- [NDJR<sup>+</sup>09] P. Nunez, P. Drews Jr, R. Rocha, M. Campos, and J. Dias. Novelty detection and 3d shape retrieval based on gaussian mixture models for autonomous surveillance robotics. In *to appear in Proceedings of The 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS'09, St. Louis, MO, USA.*, 2009.
- [NETB01] 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.

- [Ngu87] V.D. Nguyen. Constructing stable grasps in 3d. In *In Proceedings of IEEE International Conference on Robotics and Automation*, pages 234–239, 1987.
- [OA02] E. Oztop and M. A. Arbib. Schema design and implementation of the grasp-related mirror neuron system. *Biological Cybernetics*, 87(2):116–140, 2002.
- [OC01] A. M. Okamura and M. R. Cutkosky. Feature detection for haptic exploration with robotic fingers. *The International Journal of Robotics Research*, 20:925–938, 2001.
- [OCLBC11] C. Oddo, M. Controzzi, C. Cipriani L. Beccai, and M. Carrozza. Roughness encoding for discrimination of surfaces in artificial active-touch. *IEEE Transactions on Robotics*, 27:522–533, 2011.
- [ope] Open source computer vision library (OpenCV). Website: <http://opencv.willowgarage.com/wiki> (Visited January, 2012).
- [OTKH09] K. Ogawara, Y. Tanabe, R. Kurazume, and T. Hasegawa. Detecting repeated motion patterns via dynamic programming using motion density. In *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, pages 1743–1749, May 2009.
- [PCL] Point cloud library (PCL). Website: <http://pointclouds.org> (Visited June, 2012).
- [PF95] J. Ponce and B. Faverjon. On computing three finger force closure grasp of polygonal objects. *IEEE Transactions on Robotics and Automation*, 11:868–881, 1995.
- [PHAS09] Peter Pastor, Heiko Hoffmann, Tamim Asfour, and Stefan Schaal. Learning and generalization of motor skills by learning from demonstration. In *Proceedings of the 2009 IEEE international conference on Robotics and Automation, ICRA'09*, pages 1293–1298, Piscataway, NJ, USA, 2009. IEEE Press.
- [PMAT04] R. Pelosof, A. Miller, P. Allen, and T. Jebara. An svm learning approach to robotic grasping. In *IEEE International Conference on Robotics and Automation*, 2004.
- [Pol] Polhemus Inc. motion tracking system. Website <http://www.polhemus.com> (Visited February, 2009).
- [PRF02] S. Pulla, A. Razdan, and G. Farin. Improved curvature estimation for watershed segmentation of 3-dimensional meshes. *IEEE Transactions Vis Comput Graph.*, -:-, 2002.

- [RB02] A. Razdan and M. Bae. A hybrid approach to feature segmentation of 3-dimensional meshes. In *Computer-Aided Design*, 2002.
- [RD07] J. Rett and J. Dias. Human-robot interface with anticipatory characteristics based on laban movement analysis and bayesian models. In *IEEE 10th International Conference on Rehabilitation Robotics (ICORR)*, 2007.
- [RDC05] R. Rocha, J. Dias, and A. Carvalho. Exploring information theory for vision-based volumetric mapping. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1023- 1028, 2005.
- [RFKK10] Javier Romero, Thomas Feix, Hedvig Kjellstrom, and Danica Kragic. Spatio-temporal modeling of grasping actions. In *International Conference on Intelligent Robots and Systems*, 2010.
- [Ris78] G. Rissanen. Modeling the shortest data description. *Automatica*, 14:465–471, 1978.
- [RKK08] J. Romero, H. Kjellstrm, and D. Kragic. Human-to-robot mapping of grasps. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, WS on Grasp and Task Learning by Imitation*, 2008.
- [RKS00] C. Rossl, L. Kobbelt, and H.P. Seidel. Extraction of feature lines on triangulated surfaces using morphological operators. In *Smart Graphics, AAAI Spring Symposium, Stanford University*, 2000.
- [RS07] Máximo A. Roa and Raúl Suárez. Geometrical approach for grasp synthesis on discretized 3d objects. In *International Conference on Intelligent Robots and Systems*, 2007.
- [RV08] M. Richtsfeld and M. Vincze. Grasping of unknown objects from a table top. In *ECCV Workshop on Vision in Action: Efficient strategies for cognitive agents in complex environments*, 2008.
- [SC78] H. Sakoe and S. Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 1:43–49, 1978.



- [SDKN07] A. Saxena, J. Driemeyer, J. Kearns, and A. Y. Ng. Robotic grasping of novel objects. *Neural Information Processing Systems*, 19:1209–1216, 2007.
- [SE08] D. Säfström and B. B. Edin. Prediction of object contact during grasping. *Exp Brain Res.*, 190(3):265–77, 2008.
- [SEK09] A. Sahbani and S. El-Khoury. A hybrid approach for grasping 3D objects. In *IEEE/RSJ Int. Conf. intelligent robots and Systems, St. Louis, USA.*, 2009.
- [Sha] Shadow robot company. Shadow robot dextrous hand. Website: <http://www.shadowrobot.com> (Visited January, 2010).
- [SLM94] F. Solina, A. Leonardis, and A. Macerl. A direct part-level segmentation of range images using volumetric models. In *IEEE International Conference on Robotics and Automation*, 1994.
- [SMP05] T. Svoboda, D. Martinec, and T. Pajdla. A convenient multi-camera self-calibration for virtual environments. *PRESENCE: Teleoperators and Virtual Environments. MIT Press.*, 14(4):407–422, 2005.
- [SPLS08] M. Stark, M. Zillich P. Lies, and B. Schiele. Functional object class detection based on learned affordance cues. *Computer Vision Systems*, -:435–444, 2008.
- [SRHR04] J. J. Steil, F. Röthling, R. Haschke, and H. Ritter. Situated robot learning for multi-modal instruction and imitation of grasping. *Robotics and Autonomous Systems*, 47:129–141, 2004.
- [Tek] Tekscan Inc. pressure mapping, force measurement and tactile sensors. Website: <http://www.tekscan.com/grip-pressure-measurement> (Visited January, 2010).
- [Thr02] S. Thrun. Robotic mapping: a survey. In *Exploring Artificial Intelligence in the New Millennium, Morgan Kaufmann, San Mateo, CA.*, 2002.
- [VB12] F. Veiga and A. Bernardino. Towards bayesian grasp optimization with wrench space analysis. In *IEEE IROS 2012 Workshop "Beyond Robot Grasping"*, Vilamoura, Portugal, October, 2012.

- [Vid] SRI international's small vision system library SVS. Website: <http://users.rcn.com/mclaughl.dnai/svs.htm> (Visited January, 2010).
- [WL97] K. Wu and M.D. Levine. 3d part segmentation using simulated electrical charge distributions. *IEEE Transactions On Pattern Analysis and Machine Intelligence*, 19:1223–1235, 1997.
- [ZPKG02] Y. Zhang, J.K. Paik, A. Koschan, and D. Gorsich. A simple and efficient algorithm for part decomposition of 3d triangulated models based on curvature analysis. In *International Conference on Image Processing*, (3):273–276., 2002.