THÈSE DE DOCTORAT DE L'UNIVERSITÉ PIERRE ET MARIE CURIE

Spécialité

Robotique - Informatique

Présentée par

Mlle. Sahar EL-KHOURY

Pour obtenir le grade de

Docteur de l'Université Pierre et Marie Curie
(Paris VI)

Sujet de la thèse :

Approche Mixte, Analytique et par Apprentissage, pour la Synthèse d'une Prise Naturelle

soutenue le 10 décembre 2008 devant la commission d'examen

V. PERDEREAU	Professeur à Paris VI	Directrice de thèse
A. SAHBANI	Maître de conférences à Paris VI	Co-directeur de thèse
R. ALAMI	Directeur de recherche au LAAS	Rapporteurs
A. MICAELLI	Directeur scientifique au CEA	
G. DREYFUS	Professeur à l'ESPCI	Examinateurs
O. SIGAUD	Professeur à Paris VI	

Acknowledgements

Mes trois années de thèse ont été une expérience enrichissante qui m'a fait grandir sur le plan professionnel ainsi que personnel. Je remercie toutes les personnes qui ont partagé avec moi cette expérience inoubliable.

Je pense tout d'abord à mes deux rapporteurs, Alain Micaelli et Rachid Alami, d'avoir accepté de rapporter à ma thèse dans un délai digne d'être olympique! Je pense aussi à mes deux examinateurs, Olivier Sigaud et Gérard Dreyfus, d'avoir accepté d'être membres de mon jury.

Je remercie ma directrice de thèse, Véronique Perdereau, de m'avoir accueilli dans son équipe de recherche. Anis Sahbani, mon encadrant et co-directeur, merci pour ta disponibilité, tes conseils et ton soutien dans les moments délicats. Tu es pour moi, comme tu aimes bien le dire, mon père dans la recherche.

Cédric Michel et Maurice Milgram, merci de m'avoir donné confiance en moi, pendant ma première année de thèse. Jean-Philippe Saut, je te remercie pour ton aide précieuse en programmation et pour ta compagnie dans les missions aux Portugal et San Diego.

Arash Mokhber, mon collègue de bureau, merci d'avoir été présent pendant les moments les plus joyeux et les plus durs.

Je n'oublierai jamais les moments de pause.... Kévin Bailly, Ammar Mahdaoui, Ilaria Renna, Pablo Negri, Loic Lachèze, Guillaume Walck, Richard Chang, Sio Ieng, Xavier Clady ..... Sans votre présence, et d'ailleurs je l'ai vécu pendant un moment, la vie au labo est trop triste.

Eric, Brahim, Banaf, Corina, Marielle, Farid, vous serez toujours mon groupe préféré du DEA, merci pour vos encouragements et j'attends notre prochain bilan annuel. Aline, Dima, Agnella, Marianne, Diane, Joseph, Carole, Joelle, André, Kathleen, merci d'être ma famille à Paris. Bertha et Alexis, merci pour votre amitié et votre soutien malgré les distances qui nous séparent.

Je tiens finalement à remercier mes parents d'avoir toujours respecté mes choix dans la vie.

Pisa, le 28 août, 2009

Contents

List of Figures

List of Tables

Chapter 1

Introduction

> ” Là où il y a une volonté, il y a un chemin.”
>
> *Général Charles de Gaulle*

If you ask people: what would you use a robot for? Some of them reply spontaneously: ”We do not want a robot!”. For others, the answer is obvious, cleaning the house, ironing, cooking, house holding etc. According to the 2007 World Robotics survey presented by the IFR (International Federation of Robotics) statistical department, up to 2006, about 3.5 million service robots for personnel/domestic use were sold. Such robots are mainly in the areas of domestic vacuum cleaning , pool cleaning and lawn-mowing robots. About 40,000 service robots for professional use were also installed worldwide, i.e the unmanned aerial and ground-based vehicles for military, demolition systems for the construction industry and robot assisted surgery. Thus, it is clear that we are moving towards a world where intelligent robots will be in every house just as televisions and personal computers. This is the rule of supply and demand. But, what will be the implications of such intelligent robots development on the world economy? Will the need for human labor disappear? Will people suffer from unemployment? Or will the quality of life increase? Humans need for lodging affected housing prices and lead to the actual world financial crisis. Will the need for robotics be the next reason for a worldwide economic crisis? All these questions are beyond the scope of this thesis. They are even beyond the scope of our lab and university. May be the robotics society has an answer and may be they should start preparing financially for the robots era.

Regarding the needs and the existing technological level, a lot of work has to be done. For example, robots for handicap assistance is still an active research area and applications such as letting a robot get us a glass of water is still not simple to achieve. To ensure these skills and to interact with a human’s world, robots must be capable of using their hands proficiently. Since we are surrounded by objects specifically adapted to our hands shapes and sizes, robots hands should resemble the most to ours in order to successfully handle such objects. Many companies were interested in constructing robotic hands that satisfy this condition. The Shadow company hand is regarded as the most advanced robotic hand for its ability to perform all 24 movements of the human hand. However, this is not sufficient for a robot to autonomously manipulate objects. Even the forms of the most common objects are infinite. Consequently, a robot will always encounter novel objects and should decide how to use his hand in order to accurately grasp such objects and perform the required task.

Grasp synthesis is the central action of objects manipulation and this study will focus on that phase. Grasp synthesis has to satisfy three main sets of constraints: constraints due to the robotic hand which has a large number of degrees of freedom and its fingers capabilities, constraints due to object geometric features and constraints due to task requirements. Hence, elaborating a grasping approach that meets these constraints is extremely difficult and should address the following questions:

- Where the hand should grasp a novel object?
- Is the chosen grasp stable?
- Is the grasp appropriate to successfully perform a task?

The first logical step in developing a grasping strategy described in the literature was to examine the stability of the grasp. Many analytical methods have addressed this problem and concentrate on the development of stability criteria to compare grasps. These approaches find stable grasps for pick and place operations but are unable to determine a suitable grasp for object manipulation. Thus, the next step was to compute task oriented grasps. For this, the proposed approaches take as input the task to perform and find a grasp suitable for it. Modelling a task is complex and 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 are from different shapes and sizes, generalizing these techniques to novel objects is difficult. Thus, fully autonomous grasping of a previously unknown object remains a challenging problem.

We intend to take this challenge and find appropriate grasps for novel objects. Grasping is not a new field to our research team. A former PhD student, Cedric Michel, has addressed the grasp synthesis problem from a geometrical point of view. He proposed an approach to grasp planning based on the Natural Grasping Axis. This axis is parallel to the palm and surrounded by the thumb facing the other fingers. It characterizes human beings hands shapes when approaching an object. Thus, Michel et al. [MRPD04] uses an analytical method to extract such axis from the 3D object geometry. However, this approach identifies

[MRPD04] C. Michel, C. Rmond, V. Perdereau, and M. Drouin. A robotic grasping planner based on the natural grasping axis. *In Proceedings of the IEEE International Conference on Intelligent Manipulation and Grasping*, 2004.

for one object, several axes and does not select the one that is best adapted to the required task. This induces us to believe that analytical approaches cannot fulfill the challenge by themselves and that they should mix up with knowledge based methods. We propose, in this thesis, an hybrid approach combining empirical and analytical methods.

1.1 Claims of Originality and Thesis Overview

The main contributions of our research are:

- This thesis proposes a novel strategy that associates to each object a handle. This permits to find, for an unknown object, a grasp in accordance with its corresponding task. Identifying objects handles is performed by imitating humans.
- A new sufficient condition for computing force-closure grasps on the obtained handle is also proposed. This method aims at reducing force-closure grasps computation time.

The following chapters will address in details all aspects of the proposed approach. Chapter 2 reviews the current literature for analytical and empirical grasp synthesis strategies for 3D objects. It also shows that these approaches have two main difficulties: task modelling and generalization of learned grasping skills to new objects. Chapter 3 introduces evidence suggesting that a low-level vision, including object segmentation into single parts, provides a useful prelude to object grasping and that a good grasp is the result of the object graspable part identification. Once the object graspable part is selected, chapter 4 proposes a new condition for force-closure grasps computation. Finally, a series of experiments are conducted to test the proposed approach in chapter 5.

Chapter 2

Grasp Synthesis for 3D Objects

> ”Aucune des minutes que nous vivons n’aurait existé sans les millénaires qui l’ont précédée depuis la Création, et aucun de nos battements de coeurs n’aurait été possible s’il n’y avait eu les générations successives des aîeux, avec leurs rencontres, leurs promesses, leurs unions conscarées, ou encore leurs tentations”
>
> Extrait de *Le Rocher de Tanios* de *Amin Maalouf*

Grasp means take hold of or seize firmly. The study of this definition constitutes the ground theory of grasping. Different approaches have been developed to achieve firm grasps. However, when notions such as task requirements are involved, many constraints arise such that finding suitable firm grasps becomes more difficult. When a robotic hand is considered, a grasp has generally to satisfy three main sets of constraints: constraints due to the robotic hand and its fingers capabilities, constraints due to object geometric features and constraints due to task requirements. Determining a set of contacts on the surface of an object that meet these constraints is called grasp synthesis. Grasp synthesis, has been tackled with two different approaches: analytical or empirical. 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 learning algorithms to choose a grasp that depend on the task and on the object’s geometry. Different algorithms have been developed in grasp planning for two-dimensional objects [Liu00, PF95]. However, grasp synthesis for three-dimensional objects is still an active research area. This is mainly due to the complex geometry and high dimensionality of the grasp space. This chapter introduces

[Liu00] Y.H. Liu. Computing n-finger form-closure grasps on polygonal objects. *International Journal of Robotics Research*, 19:(2):149158, 2000.

[PF95] J. Ponce and B. Faverjon. On computing three finger force closure grasp of polygonal objects. *IEEE Transactions on Robotics and Automation*, 11:(6):868881, 1995.

first a description of a grasp along with the properties used to evaluate its suitability. Current literature for analytical and empirical grasp synthesis strategies for 3D objects is then reviewed.

2.1 Description of a Grasp

A grasp is defined as a system in which an object is gripped by the fingers of a robotic or human hand. The configurations of the hand, the object and the contacts on the object have an effect on the properties of the grasp. The foundation for the current research in this area was laid out by Salisbury [SR82] . In his thesis he classified types of contacts between two bodies. He also derived the grip transformation (also known as the grasp map) which relates forces at the contacts points to overall object forces. Finally, he provided an analysis of the desirable kinematic properties of an articulated robotic hand. The research that followed his works developed new methods to synthesize grasps and new measures to analyze the quality of a given grasp. This section summarizes significant grasp attributes. It first examines the contact interface between the fingers and the object. The grip map and the main grasp properties are then presented.

2.1.1 Contact Models

The contact model defines the connectivity at the contact point between a finger and the grasped object. Three common models are: frictionless point contact, point contact with friction (hard finger), and soft finger contact. The frictionless point contact model allows the finger to exert a force on the object only in the direction of the object surface normal at the contact point. The hard finger model adds tangential friction force components at the contact point, limited by a friction coefficient, μ. The set of possible contact forces is defined by a friction cone about the object surface normal. The soft finger model adds the ability to generate a torque about the surface normal, limited by a torsional friction coefficient, γ. Figure 2.1 illustrates each contact model and provides definitions of the wrench bases and force constraints [MLS94]. The contact reference frame is chosen such that the z-axis is collinear with the inward surface normal at the point of contact.

2.1.2 Grasp Wrench Space

Any force acting at a contact point on the object also creates a torque relative to reference point r that can be arbitrary chosen. Often the center of mass is used as that reference point to give it a physical meaning. These force and torque vectors are concatenated to a wrench. A grasp wrench space (GWS) is characterized by the set of wrenches that can be applied to

[SR82] J.K. Salisbury and B. Roth. Kinematic and force analysis of articulated hands. *ASME J. Mech., Transmissions, Automat.,Design*, 105:33–41, 1982.

[MLS94] R.M. Murray, Z. Li, and S.S. Sastry. A mathematical introduction to robotic manipulation. *Orlando, FL: CRC*, 1994.

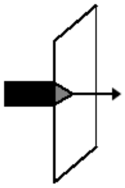
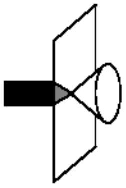
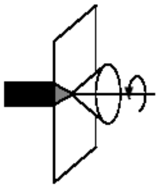
Contact Type	Picture	Wrench Basis	FC
Frictionless point contact		$\begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$	$f_1 \geq 0$
Point contact with friction		$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	$\sqrt{f_1^2 + f_2^2} \leq \mu f_3$ $f_3 \geq 0$
Soft finger		$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	$\sqrt{f_1^2 + f_2^2} \leq \mu f_3$ $f_3 \geq 0$ $\lvert f_4 \rvert \leq \gamma f_3$

Figure 2.1: Common Contact Types [MLS94]

the target object from the contacts of a grasp, given certain limitations on applied forces. The grasp wrench space is bounded by the convex hull of the contact wrenches formed from unit applied forces at the contact of the grasp [FC92]. The length of applied forces is normalized to a unit force as each finger is assumed to apply the same magnitude of force. Note that only the contact model and contact locations on the object are factors in determining the grasp wrench space. The configuration of the hand is not addressed and does not even need to be defined.

2.1.3 The Goal of a Grasping Strategy

The first goal of every grasping strategy is to ensure stability. A grasp is stable if a small disturbance, on the object position or finger force, generates a restoring wrench that tends

[FC92] C. Ferrari and J. Canny. Planning optimal grasps. *In Proceedings of IEEE International Conference on Robotics and Automation*, 1992.

to bring the system back to its original configuration [HK96, BDK98]. Nguyen [Ngu87] introduces an algorithm for constructing stable grasps. Nguyen also proves that all 3D force-closure grasps can be made stable. A grasp is force-closure when the fingers can apply appropriate forces on the object to produce wrenches in any direction. In the literature, this condition may be confused with form-closure. The latter induces complete kinematical restraint of the object and is obtained when the positions of the fingers ensure object immobility. Bicchi [Bic95] describes in detail these conditions.

Obviously, 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. Computing task-oriented grasps is consequently crucial for a grasping strategy.

Finally, because of the variety of objects shapes and sizes, a grasping strategy should always be prepared to grasp new objects.

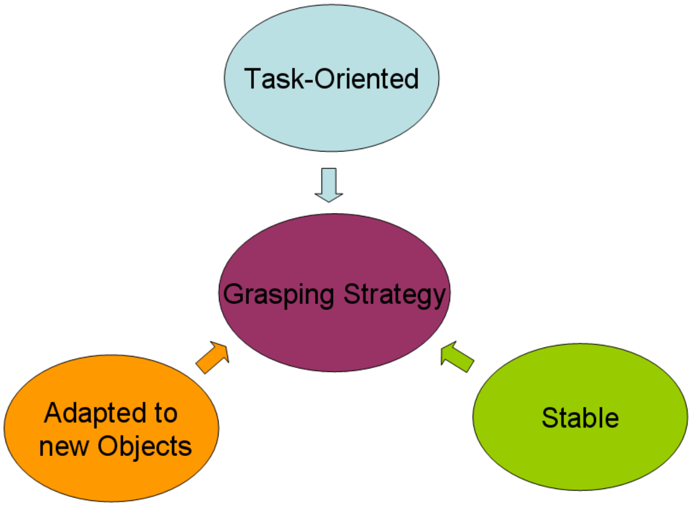


Figure 2.2: The ultimate goal of a grasping strategy.

[HK96] W.S. Howard and V. Kumar. On the stability of grasped objects. *IEEE Transactions on Robotics and Automation*, 12(6):904917, 1996.

[BDK98] H. Bruyninckx, S. Demey, and V. Kumar. Generalized stability of compliant grasps. *In Proceedings of IEEE International Conference on Robotics and Automation*, page 23962402, 1998.

[Ngu87] V.D. Nguyen. Constructing stable grasps in 3d. *In Proceedings of IEEE International Conference on Robotics and Automation*, pages 234–239, 1987.

[Bic95] A. Bicchi. On the closure properties of robotic grasping. *International Journal of Robotics Research*, 14:(4):319–334, 1995.

Thus, a grasping strategy, as shown in (Fig. 2.2), should ensure stability, task compatibility and adaptability to novel objects. In other terms, a grasp synthesis strategy should always have an answer to the following question: where to grasp a novel object in order to accomplish a task? Analytical and empirical approaches answer this question differently.

2.2 Analytical Approaches

Analytical Approaches consider the laws of physics, kinematics and dynamics in determining grasps. The complexity of this computation arises from the number of conditions that must be satisfied for a successful grasp. We previously showed that two main conditions identified in the grasping bibliography are force-closure and task compatibility. The following paragraphs present strategies developed to meet these conditions. The diagram of (Fig. 2.3) summarizes these strategies. A quick look at this diagram shows that many works have been developed to compute force-closure grasps but only few have addressed the problem of computing task oriented ones. This is due to the difficulty of the latter. In the following, we present and discuss some relevant works for generating force-closure and task-oriented grasps.

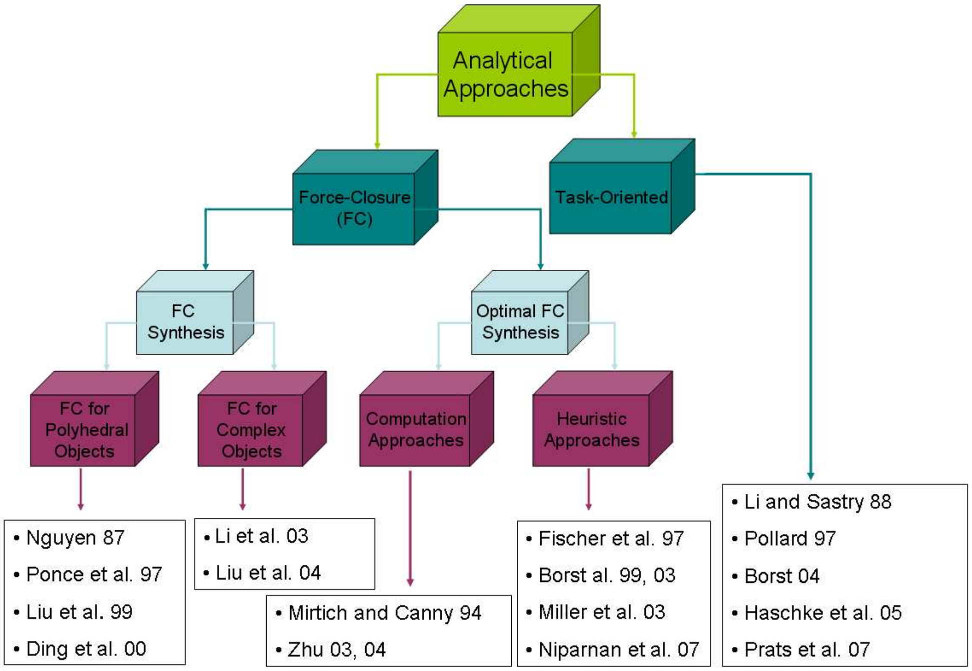


Figure 2.3: Analytical approaches for grasp synthesis of 3D objects.

2.2.1 Force-Closure Grasps

The works in this section present techniques for finding force-closure grasps for 3D objects. For this purpose, two approaches may be considered: (1) analyzing whether a grasp, defined by a set of contacts, is force-closure or not or (2) finding places to put the fingertips, such that the grasp is force-closure. The former considers force-closure necessary and sufficient conditions and will be detailed in chapter 4. The latter is the force-closure grasp synthesis problem and it is the one considered here. Given the quantity of relevant work in this field, we divide them into the following groups: (1) force-closure grasps synthesis for 3D objects and (2) optimal force-closure grasps synthesis according to a quality criterion.

2.2.1.1 Force-Closure Grasps Synthesis for 3D Objects

Depending on the object model, polyhedral or complex, different grasps synthesis strategies have been proposed in the literature. We present first those dealing with polyhedral objects. These objects are composed of a finite number of flat faces. Evidently, each face has a constant normal and the position of a point on a face can be parameterized linearly by two variables. Based on these properties, grasp synthesis approaches dealing with polyhedral objects reduce the force-closure condition to a test of the angles between the faces normals [Ngu87] or use the linear model to derive analytical formulation for grasps characterization [PSBM93, LDW99, DLSX00]. Thus, these approaches do not consider the issue of selecting a grasping facet. An exhaustive search is performed instead. We present briefly some of these methods.

Nguyen [Ngu87] : Nguyen developed efficient algorithms for constructing force-closure grasps on polygons and polyhedra. His analysis concerned frictionless, hard-finger and soft-finger contacts. He divided the force-closure problem into two independent sub-problems: force-direction closure and torque closure. Force-direction condition depends on the angle between the two planes of contacts. Thus, force-direction closure does not depend on the contact points locations on the faces because the normals are constant on these faces. On the other hand, Nguyen proved that, for soft contacts, torque closure condition depends on the line joining contact points location in respect to the corresponding friction cones. This condition was then extended to the hard and frictionless contacts case (Fig. 2.4).

Ponce et al. [PSBM93, PSSM97] : Ponce et al. characterized the force-closure

[Ngu87] V.D. Nguyen. Constructing stable grasps in 3d. *In Proceedings of IEEE International Conference on Robotics and Automation*, pages 234–239, 1987.

[PSBM93] J. Ponce, S. Sullivan, J.D. Boissonnat, and J.P. Merlet. On characterizing and computing three- and four-finger force-closure grasps of polyhedral objects. *In Proceedings of IEEE International Conference on Robotics and Automation*, pages 821–827, 1993.

[LDW99] Y.H. Liu, D. Ding, , and S. Wang. Constructing 3d frictional form-closure grasps of polyhedral objects. *IEEE Transactions on Robotics and Automation*, page 19041909, 1999.

[DLSX00] D. Ding, Y. Liu, Y.T. Shen, and G.L. Xiang. An efficient algorithm for computing a 3d form-closure grasp. *In Proceedings of IEEE International Conference on Robotics and Automation*, page 12231228, 2000.

[PSSM97] J. Ponce, S. Sullivan, A. Sudsang, and J.P. Merlet. On computing four-finger equilibrium and force-closure grasps of polyhedral objects. *International Journal of Robotics Research*, 16:(1):1135, 1997.

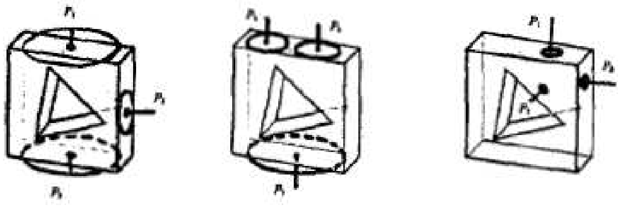


Figure 2.4: Examples of force-closure grasps in 3D with three hard-finger contacts [Ngu87].

grasps of 3D polyhedral objects for hard finger contacts. Based on the property that each point on a plane face can be parameterized linearly with two parameters, the authors formulated necessary linear conditions for three and four-finger force-closure grasps (Fig. 2.5) and implemented them as a set of linear inequalities in the contact positions. Finding all force-closure grasps is thus set as a problem of projecting a polytope onto a linear subspace.

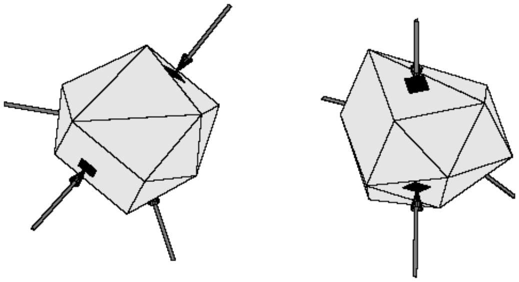


Figure 2.5: Four-finger force-closure grasps of a 18-sided polyhedron [PSSM97].

Liu et al. [LDW99] : Liu et al. discussed the force-closure grasp synthesis problem for n fingers when $n-1$ fingers have fixed positions and the grasp with the $n-1$ fingers is not force-closure. Using the linear parametrization of a point on an object facet, they search locations on that facet for the nth finger that ensure force-closure. Grasp computation is transformed to a problem of computing the convex cone of the primitive contact wrenches of the $n-1$ fingers and then checking the intersection between a circular region and a convex polygon in a $2D$ space. The previous ideas are extended in [DLW00] to compute the locations of $n-k$ fingers given the locations for k fingers that do not generate a force-closure grasp.

Ding et al. [DLSX00] : Ding et al. presented an algorithm to compute the positions for n fingers to form a force-closure grasp from an initial random grasp. The algorithm first arbitrarily chooses a grasp on the given faces of the polyhedral object. If the selected grasp is not form-closure or in other words if the origin O of the wrench space lies outside the primi-

[DLW00] D. Ding, Y. Liu, and S. Wang. Conmputing 3-d optimal form-closure grasps. *In Proceedings of IEEE International Conference on Robotics and Automation*, page 35733578, 2000.

tives wrenches convex hull, the algorithm moves each fingertip position at a fixed step on its corresponding face so that the convex hull moves towards the origin O and consequently the form-closure property is ensured.

► The previous analysis were limited to polyhedral objects such as boxes. They are efficient when the number of faces of the object is low. However, commonly used objects like mugs or bottles are not necessarily polyhedral and can rarely be modelled with a limited number of faces. Hence, when polyhedral grasp synthesis approaches are applied to these objects, they need a huge computation effort to study the combinations of their large number constituting faces. Thus, new techniques are required for force-closure grasps synthesis. Such general approaches are presented in the next paragraph [LLC03, DLW01]. No restrictions are placed on the object model. The latter are modelled with a cloud of 3D points or a triangular mesh.

Li et al. [LLC03] : The authors presented an algorithm for computing three finger force-closure grasps for $2D$ and $3D$ objects. They assume hard-finger contacts. Based on the intersection of the corresponding three friction cones, the authors compute three-finger force-closure grasps of 2D objects. When dealing with 3D objects, the three contact fingers constitute a contact plane. By taking into account this plane, the problem of computing three-finger force-closure grasps for 3D objects is simplified to a $2D$ force-closure problem.

Ding et al. [DLW01] : The authors proposed an algorithm to synthesize force-closure grasps with 7 frictionless contacts. The grasped object is discretized so a large cloud of points p_i as well as their normals n_i is available. Then, a large collection of contact wrenches g_i can be obtained. The algorithm starts with an initial set of seven contacts randomly chosen among the set of points. If the selected grasp is force-closure, the algorithm finishes. Otherwise, the initial contacts are iteratively exchanged with other candidate locations until a force-closure grasp is obtained. The previous heuristic algorithm is extended in [LLD04] for any number of contacts with or without friction.

► Such methods find contact points on a 3D object surface that ensure force-closure. But what about computing good force-closure grasps? For this purpose, different quality criteria were introduced to the grasping literature. In the following, we present some relevant works on computing optimal grasps.

[LLD04] Y.H. Liu, M.L. Lam, and D. Ding. A complete and efficient algorithm for searching 3-d form closure grasps in the discrete domain. *IEEE Transactions on Robotics and Automation*, 20:(5):805816, 2004.

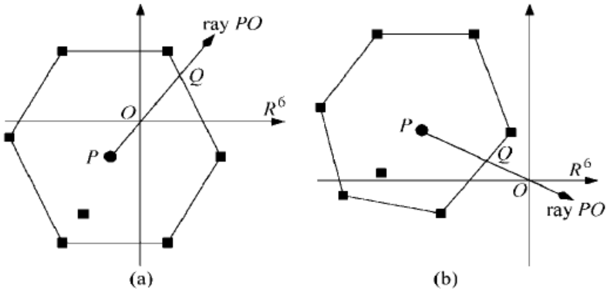


Figure 2.6: Moving the convex hull gradually closer and closer to the origin of the wrench space O through an iterative process. (a) Form-closure grasp. (b) Non form-closure grasp [LXL04].

2.2.1.2 Optimal Force-Closure Grasps on 3D Objects

Optimal force-closure grasps synthesis concerns determining the contact points locations so that the grasp achieves the most desirable performance in resisting external wrench loads. These approaches are tackled between optimizing and heuristical techniques. This paragraph reviews first otpimizing techniques. These techniques compute optimal force-closure grasps by optimizing an objective function according to a pre-defined grasp quality criterion. When objects are modelled with a set of vertices, they search all their combinations to find the optimal grasp [MC94]. On the other hand, for smoothed objects such as ellipsoids, the primitive wrenches of the grasp are also smooth functions of the grasp configuration. Thus, a gradient descent permit in these cases to compute optimal grasps [ZW03, ZD04].

Mirtich et al. [MC94] : Mirtich and Canny developed two optimality criteria and used them to derive optimum two and three finger grasps of 2D objects and optimum three fingers grasps of 3D polyhedra objects. Whether the first or the second criterion is used, the maximum circumscribing or the maximum inscribing equilateral triangle defines the optimum grasp of a 3D object (Fig. 2.7). The optimum grasp points must be vertices of the polyhedron. Thus, if we dispose of a n-vertices polyhedron, testing all triples of vertices of the polyhedron gives an O(n^3) algorithm for finding the optimum three finger grasp of the polyhedron.

Zhu and Wang [ZW03] : Zhu and Wang proposed an algorithm to synthesize grasps for any 3D object with smooth curved surfaces with any number of contacts. The algorithm is based on the concept of the Q distance or Q norm. This norm quantifies the maximum

[MC94] B. Mirtich and J. Canny. Easily computable optimum grasps in 2d and 3d. *In Proceedings of IEEE International Conference on Robotics and Automation*, 1:739–747, 1994.

[ZW03] X. Zhu and J. Wang. Synthesis of force-closure grasps on 3d objects based on the q distance. *IEEE Transactions on Robotics and Automation*, 19:(3), 2003.

[ZD04] X. Zhu and H. Ding. Planning force-closure grasps on 3-d objects. *In Proceedings of IEEE International Conference on Robotics and Automation*, 2004.

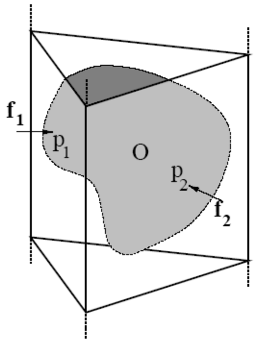


Figure 2.7: A circumscribing prism for object O (point p_3 and force f_3 are hidden on the back face) [MC94].

wrench that can be resisted in a predefined set of directions. Since the Q distance is differentiable on the object surface, its derivative can be calculated and used for the gradient descent minimization in the grasp configuration space to find an optimal grasp.

Zhu and Ding [ZD04] : Zhu and Ding proposed a similar algorithm. The grasp configuration is denoted by u, which specifies the positions of the contact points. Moreover, the authors assume that the object has smooth surface. Thus, the primitive wrenches can be represented as smooth functions of the grasp configuration. f(u) is a function that provides a measure on how far the grasp is from losing the closure property. Thus, a natural way to compute the force-closure grasp is to minimize f(u). The optimization problem can be solved by descent search.

► Searching the grasp solution space for an optimal grasp is a complex problem requiring a large amount of computing time. Fast algorithms are required to integrate grasp planners in on-line planning systems for robots. Hence, heuristic approaches were applied to the grasp synthesis problem. These approaches generate first many grasps candidates, filtered them with a simple heuristic and then choose the best candidate [FH97, BFH03, MKAC03]. However, such approaches suffer from the local minima problem.

Fischer and Hirzinger [FH97] : Fischer and Hirzinger presented a simple heuristic search to synthesize force-closure grasps with three fingers. A coordinate system with arbitrary origin and orientation is generated inside the bounding box of the object. Three rays are generated in predefined directions from the origin of this frame (Fig. 2.8). If all the three rays yield one penetration point, the set of intersection points with the object surface is a grasp candidate. Grasp candidates are then filtered to exclude candidates which can not lead

[ZD04] X. Zhu and H. Ding. Planning force-closure grasps on 3-d objects. *In Proceedings of IEEE International Conference on Robotics and Automation*, 2004.

to feasible grasps. Candidate grasps passing the filters are ordered according to a quality measure, and the algorithm finally chooses the best quality grasp within the initial candidate grasps. This work is extended in [BFH99] to four-fingers grasps.

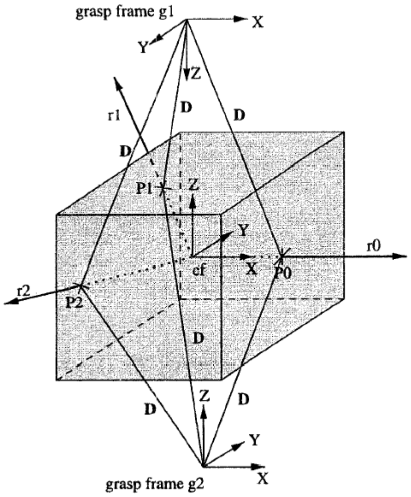


Figure 2.8: Generation of a grasp candidate [FH97].

Borst et al. [BFH03] : The authors generated grasp candidates randomly choosing contacts on the object surface. The candidates are pre-filtered to discard those candidates for which a computed external force f_{ext} breaks the grasp. Among the candidates passing the pre-filtering stage, the force-closure grasp candidates are selected. The quality of these candidates is measured according to the classical quality index presented in [FC92], which quantifies the maximum wrench that the grasp can resist with independence of the wrench direction. The algorithm finally chooses the best quality grasp. It is tested for 3, 4 and 5 fingers, and it is stated that the generated grasps have a quality similar to the expected quality from a human grasp on the same objects.

Miller et al. [MKAC03] : By modelling an object as set of shape primitives such as spheres, cylinders, cones and boxes, the authors defined a set of rules to generate a set of grasp starting positions and pre-grasp shapes that can then be tested on the object model. The best grasp in then determined according to the quality criterion of Ferrari and Canny [FC92].

[FC92] C. Ferrari and J. Canny. Planning optimal grasps. *In Proceedings of IEEE International Conference on Robotics and Automation*, 1992.

► All these approaches have studied stable grasps and developed various stability criteria to find optimal grasps. But what really dictates the choice of a grasp? After examining a variety of human grasps, the authors in [CW86] conclude that the choice of a grasp was dictated by the tasks to be performed with the object. Thus, finding a good stable grasp of an object is only a necessary but not sufficient condition. Therefore, many researchers addressed the problem of computing task-oriented grasps.

2.2.2 Task Compatibility

A good grasp should be task oriented. Few grasping works take the task into account. This is due to the difficulties of modelling a task and providing criteria to compare the suitability of different grasps to the task requirements. Works that addressed task-oriented grasps computation are reviewed in this paragraph.

Li and Sastry [LS88] : Li and Sastry developed a grasp quality measure related to the task to be performed. They showed that the choice of a task oriented grasp should be based on the capability of the grasp to generate wrenches that are relevant to the task. Assuming a knowledge of the task to be executed and of the workpiece geometry (Fig. 2.9), they planned a trajectory of the object before the grasping action in order to model the task by a six-dimensional ellipsoid in the object wrench space. The latter is then fitted to the grasp wrench space. The problem with this approach is how to model the task ellipsoid for a given task, which the authors state to be quite complicated.

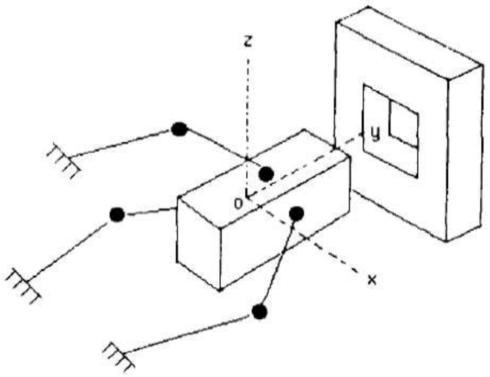


Figure 2.9: Peg-in-hole task [LS88].

[LS88] Z. Li and S.S. Sastry. Task-oriented optimal grasping by multifingered robot hands. *IEEE Journal of Robotics and Automation*, 4:(1), 1988.

Pollard [Pol97] : Nancy Pollard designed a system that found grasps that were within a certain percentage of the quality of a given prototype grasp. A grasp prototype is defined as an example object and a high quality grasp of that object. A task is characterized as the space of wrenches that must be applied to the object by the robot in order to complete the task objective. Assuming that the probability for every wrench direction to occur as a disturbance is equal, the task wrench space is modelled as a unit sphere. The grasp quality measure used is the amount the robot has to squeeze the object in order to be capable of resisting all task wrenches while maintaining the grasp. By accepting the reduced quality, the contact points of the prototype grasp can be grown into contact regions. Pollard's system can be considered one of the more general grasp synthesis tools available, but it has a few difficulties. While the prototypes allow her to greatly reduce the complexity of the search, a system to choose the closest prototype grasp is not given. Thus, the computed grasps are unlikely to be perfect for a given task or object.

Borst et al. [BFH04] : Pollard introduced the Object Wrench Space (OWS) which incorporates the object geometry into the grasp evaluation. The OWS contains any wrench that can be created by disturbance forces acting anywhere on the object surface (Fig. 2.10). Borst et al. combined the idea of the task ellipsoid [LS88] with the concept of the OWS to obtain a new description of the task wrench space (TWS). The latter is the 6D ellipsoid circumscribing the OWS. The quality of a grasp is obtained by comparing the TWS (which is no longer a sphere) with the Grasp wrench space (GWS) of the grasp that is actually evaluated. In other words, for a given TWS, the largest scaling factor is searched to fit it into a GWS.

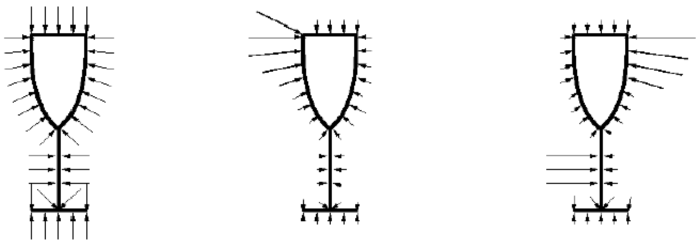


Figure 2.10: Illustration of different force distributions that produce the wrench set of the OWS. Each distribution contributes one single wrench to the OWS set. The length of all force vectors sum to the unit length [BFH04].

Haschke et al. [HSSR05] : The authors proposed a method for computing a task

[Pol97] N.S. Pollard. Parallel algorithms for synthesis of whole-hand grasps. *In Proceedings of IEEE International Conference on Robotics and Automation*, 1997.

[BFH04] Ch. Borst, M. Fischer, and G. Hirzinger. Grasp planning: How to choose a suitable task wrench space. *In Proceedings of IEEE International Conference on Robotics and Automation*, 2004.

[HSSR05] R. Haschke, J.J. Steil, I. Steuwer, and H. Ritter. Task-oriented quality measures for dextrous oriented quality measure. The approach is based on a linear matrix inequality formalism, treating friction cone constraints without the pyramidal approximation. It evaluates the grasp for a given task wrench along a single direction and specifies the largest applicable wrench along this direction (Fig. 2.11). Thus, It allows optimization of the maximal applicable wrench for a given task wrench direction.

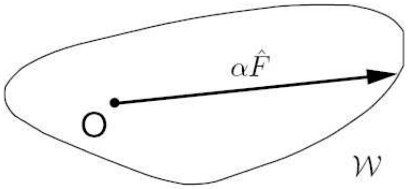


Figure 2.11: The proposed grasp quality measure $\mu(\hat{F}_t)$ maximizes the magnitude α of a given task wrench $\hat{F}_t$ in the grasp wrench space W [HSSR05].

Prats et al. [PSdP07] : Instead of finding a grasp and evaluating its suitability for the desired task, the authors proposed an approach that takes the task into account from the early grasp planning stages using hand-preshapes. They defined four hand preshapes along with an approximation of their grasp wrench space (Fig. 2.12). The hook power preshape is adapted for grasping handles and pushing along a known direction. The hook precision has the same preshape as the hook power one but the contact is made with fingertips. The precision preshape permit forces to be exerted along the two senses of a same direction which enables turning a tap for example. In cylindrical preshape, the fingers enclose the object and make force towards the palm. Thus, to accomplish a task, a robot has to align the appropriate hand's task frame with a target frame that is selected during task planning. The hand preshape and its corresponding target frame are selected according to the task direction and a simplified model of the manipulated object. Objects are modelled as hierarchy of boxes. This algorithm was tested for accomplishing a common task, turning a door handle.

grasping. *Proceedings IEEE International Symposium on Computational Intelligence in Robotics and Automation, CIRA*, pages 689–694, 2005.

[PSdP07] M. Prats, P.J. Sanz, and A.P. del Pobil. Task-oriented grasping using hand preshapes and task frames. *In Proceedings of IEEE International Conference on Robotics and Automation*, 2007.

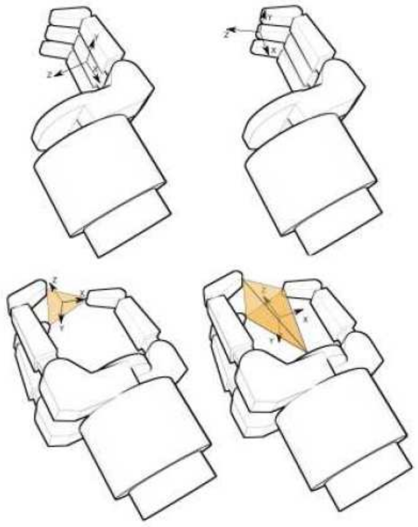


Figure 2.12: Task frames for the hook power (top-left), hook precision (top-right), precision (bottom-left) and cylindrical (bottom-right) preshapes [PSdP07].

► The task wrench space (TWS) models wrenches applied on the grasped object in order to perform a task. Given an object and a task to be executed, Li and Sastry proposed to represent the TWS as a six-dimensional ellipsoid. The latter conforms well the task but its difficult to obtain. The authors were conducted to pre-compute the trajectory followed by the object to accomplish the task. Obviously, this approach is not adapted to new tasks nor to new objects, the whole computation procedure will be repeated. Pollard models the TWS with a six-dimensional unit sphere. Thus, it is assumed that the probability for every wrench direction to occur is equal. This representation has no physical interpretation since wrenches occurring at an object boundary are not uniform. Consequently, the TWS is not uniform as well. Borst approximates the OWS with an ellipsoid in order to model the TWS. This representation takes into account the object geometry and the wrenches it may encounter. But since this representation accounts for different wrenches on the whole object boundary, it does not consider task specific information. Thus, the computed grasp is not the best adapted to a specific task. Haschke optimizes the maximal applicable wrench for a given task wrench direction. However, the paper does not include any information about the corresponding task wrench direction computation. Prats approach is adapted for tasks occurring along a specific direction such as opening a door or a drawer where it is easy to model objects with boxes in order to determine their corresponding target frame. Such approach fails to associate appropriate hand preshapes to more complex tasks.

2.2.3 Discussion on Analytical Approaches

The analytical methods described in the previous sections concentrate on the analysis of a particular grasp or the development of force-closure or task-oriented criteria to compare grasps. The size of the grasp solution space is the most difficult obstacle to overcome in optimizing the grasp. The presented criteria to compute force-closure grasps may yield to optimal stable grasps adapted for pick and place operations (Fig. 2.13). However, physical interaction through manipulation in our daily life, even for simple and common tasks, goes beyond grasping for picking and placing. That's why many researchers addressed the problem of task-oriented grasping. The goal of task-oriented grasp planning is to solve the following problem: given an object and a task, how to grasp the object to efficiently perform the task?

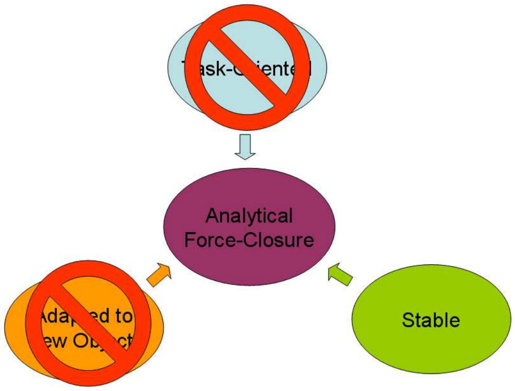


Figure 2.13: Analytical force-closure grasps synthesis ensure stability but not task-compatibility. They are also not adapted to new objects. For each novel object, the complete computation is repeated.

Two main problems are encountered when addressing this issue:
- The difficulty of modelling a task.
- The computational effort to find a grasp suitable for the corresponding task.
Different task-oriented criteria were introduced to the grasping literature and a task-oriented grasp was obtained by generating and evaluating lots of grasps according to these criteria. But all the proposed approaches could not overcome the problem of the task representation and thus are computationally unaffordable (Fig. 2.14). There are also not adapted neither for new tasks nor for new objects. While the selection of task-oriented optimal grasp is very easy for a human hand, it is still a complicated process for a robot hand. Hence, there is a need to a system that takes into account aspects of naturals grasps by imitating humans rather than modelling tasks.

In order to avoid the computational complexity of analytical approaches, empirical techniques were introduced to the grasping problem. By taking a further look at the diagrams of (Fig. 2.3) and (Fig. 2.15), we notice that most recent works are based on empirical approaches. These techniques are detailed in the next paragraph.

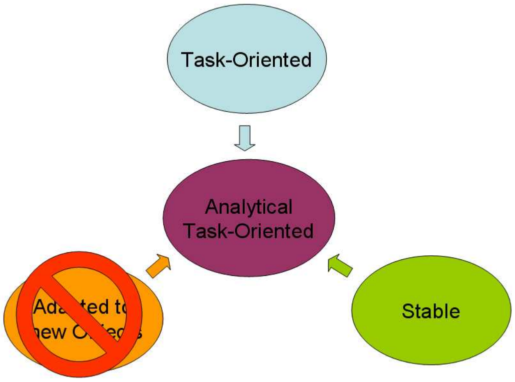


Figure 2.14: Analytical task-oriented grasps synthesis ensure stability and task-compatibility but they are not adapted to new objects and suffers from task-modelling problem.

2.3 Empirical Approaches

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 two main kinds: (1) systems based on the observation of the object to be grasped and (2) systems based on the observation of a human performing the grasp (Fig. 2.15). The former techniques generally learn to associate objects characteristics with a hand preshape, while in the latter, a robot observes a human operator performing a grasp and try then to imitate the same grasp. This technique is called in the literature learning by demonstration approach.

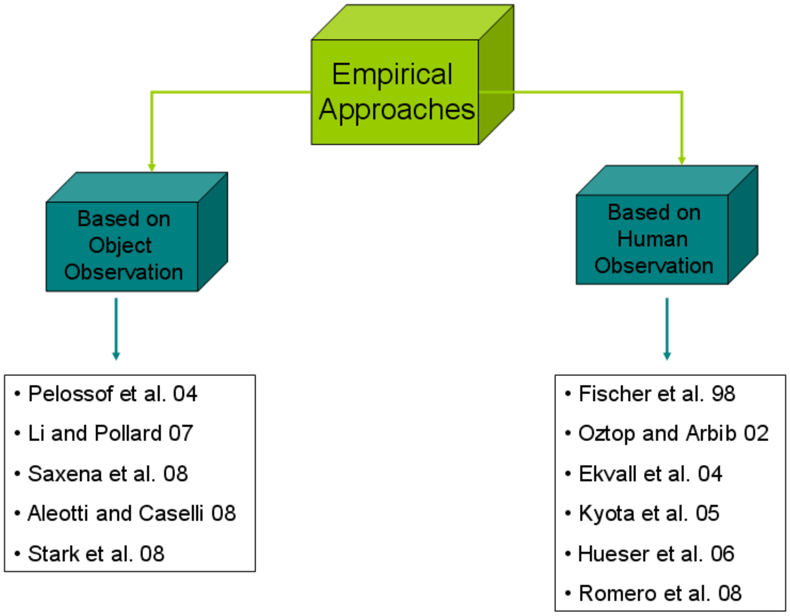


Figure 2.15: Empirical approaches for grasp synthesis of 3D objects.

2.3.1 Systems based on humans observation: Learning By Demonstration

Different Learning-by-Demonstration frameworks, where the robot observes the human performing a task and is afterwards able to perform the task itself were proposed in the literature. One of the problems arising in human based learning settings is the one of measuring human performance. Some researchers use datagloves, map 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. Others use stereoscopy to track the demonstrator's hand performing a grasp [HBZ06] or try to recognize its hand shape from a database of grasp images [RKK08]. Mirror neurones that fire not only when grasping but also when observing an action were also introduced to the grasping problem [OA02]. The following paragraphs discuss these approaches.

Aleotti and Caselli [AC08]: Aleotti and Caselli proposed a method for programming task-oriented grasps by means of user-supplied demonstrations. The procedure is based on the generation of a functional wrench space which is built by demonstration and interactive teaching. The idea is to let an expert user demonstrate a set of task-appropriate example grasps on a given target object, and to generate the associated functional wrench space as the convex union of the single wrenches (Fig. 2.16). The grasp evaluation is obtained by computing a quality metric Q, defined as the largest factor by which the grasp wrench space (GWS) of the grasp to be evaluated can be scaled to fit in the demonstrated functional wrench space (FWS). Functional wrench space Grasp demonstration is performed in virtual reality by exploiting a haptic interface including a dataglove and a motion tracker for sensing the configuration of human hand [AC07].

[FdSH98] M. Fischer, P. Van der Smagt, and G. Hirzinger. Learning techniques in a dataglove based telemanipulation system for the dlr hand. *In Proceedings of IEEE International Conference on Robotics and Automation*, 1998.

[EK04] S. Ekvall and D. Kragic. Interactive grasp learning based on human demonstration. *In Proceedings of IEEE/RSJ International Conference on Robotics and Automation*, 2004.

[KWSN05] F. Kyota, T. Watabe, S. Saito, and M. Nakajima. Detection and evaluation of grasping positions for autonomous agents. *In International Conference on Cyberworlds*, page 453460, 2005.

[AC08] J. Aleotti and S. Caselli. Programming task-oriented grasps by demonstration in virtual reality. *In Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, WS on Grasp and Task Learning by Imitation*, 2008.

[HBZ06] M. Hueser, T. Baier, and J. Zhang. Learning of demonstrated grasping skills by stereoscopic tracking of human hand configuration. *In Proceedings of IEEE International Conference on Robotics and Automation*, 2006.

[RKK08] J. Romero, H. Kjellstrm, and D. Kragic. Human-to-robot mapping of grasps. *In Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, WS on Grasp and Task Learning by Imitation*, 2008.

[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.

[AC07] J. Aleotti and S. Caselli. Robot grasp synthesis from virtual demonstration and topology-preserving environment reconstruction. *In Proceedings of IEEE/RSJ International Conference*

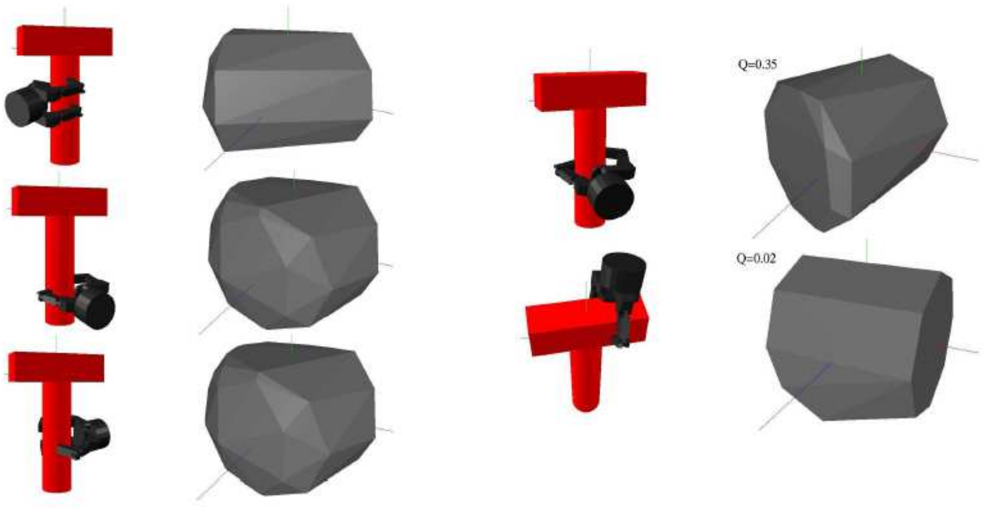


Figure 2.16: The left column shows task oriented demonstrated examples grasps along with the incremental FWS projected onto null torque space. The right column shows evaluation of a good grasp (top row) and a low-quality grasp (bottom row) [AC08].

Fischer et al. [FdSH98] : The authors presented a setup to control a four-finger anthropomorphic robot hand using a dataglove. In order to measure the finger tip positions of an operator wearing a dataglove, the fingertips were marked with round colored pins. A calibrated stereo camera setup was used to track the four color markers in real time, from which the 3D positions of the color markers were computed. During the measurements, the operators wrist was fixed while the fingers were moved. As the kinematics and configuration spaces of a human hand and an artificial robotic hand are generally different, the fingertip positions of the robotic hand cannot correspond exactly to the fingertip positions of the human hand. A good mapping between the human and the artificial fingers positions is required (Fig. 2.17). Thus, to be able to accurately use the dataglove a nonlinear learning calibration using a novel neural network technique was implemented. Based on the dataglove calibration, a mapping for human and artificial hand workspace can be realized enabling an operator to intuitively and easily telemanipulate objects with the artificial hand.

Ekvall et al. [EK04] : A similar framework is proposed in [EK04]. The human and the robot are both standing in front of a table, on which a set of objects are placed. The human demonstrates a task to the robot by moving objects on the table. The robot recognizes which object has been moved. Using magnetic trackers, the location and orientation of the fingers and palm of the human hand are obtained. Then a mapping between human and robot hand permits a robot to learn the different joint values while observing a grasp execution (Fig. 2.18). Based on this data, the authors use Hidden Markov Models (HMM) to

on Intelligent Robots and Systems, 2007.

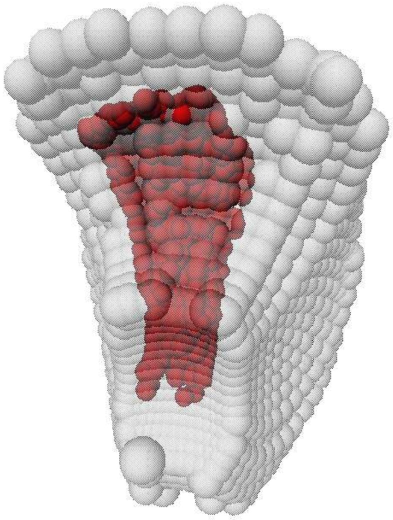


Figure 2.17: Mapping the workspaces of a human finger (dark) to an artificial one [FdSH98].

enable recognition of four different grasp types. Thus, the robot is able to reproduce the task performed by the human. Since objects may not be placed at the same location as during the demonstration, more recently [EK07], the authors addressed the problem of grasp generation and planning when the exact pose of the object is not available. Thus a method for learning and evaluating the grasp approach vector was proposed so that it can be used in the above scenario.

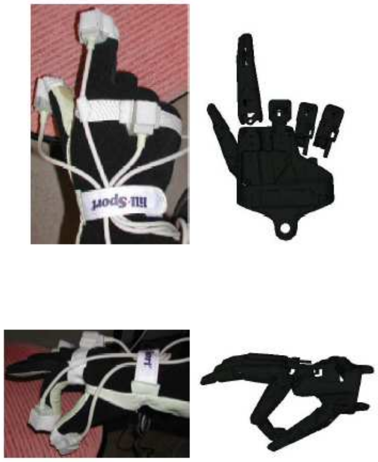


Figure 2.18: An example mapping of a human pointing grasp using the Robonaut Hand [EK04].

► Although magnetic trackers and datagloves deliver exact values of hand joints, it is desirable from a usability point of view that the user demonstrates tasks to the robot as naturally as possible; the use of gloves or other types of sensors may prevent a natural grasp. This motivates the use of systems with visual input.

Hueser et al. [HBZ06] : The authors proposed a vision and audio based approach. The scenario of learning by demonstration consists of a human demonstrator standing opposite the service robot and a table that is placed between them. Several objects are placed on the table. These objects are trained for recognition in advance and offline. The user demonstrates a grasping skill by saying ”start” and reaching out his hand to the object he wants to grasp. Then he grasps the object and says ”stop”. The robot stereoscopically tracks the demonstrator's hand several times to collect sufficient data (Fig. 2.19). However, the accuracy of the visual tracking is limited by the camera's resolution and the quality of the calibration procedure. Additionally, every time a grasp is demonstrated, the user performs it differently. To compensate for these inaccuracies, the measured trajectories are used to train a Self-Organizing-Map (SOM). The SOMs give a spatial description of the collected data and serve as data structures for a reinforcement learning (RL) algorithm which optimizes trajectories for use by the robot. The authors, in [HZ08], applied a second learning stage to the SOM, the Q-Learning algorithm. This stage accounts for changes in the robots environment and makes the learned grasping skill adaptive to new workspace configurations.

Romero et al. [RKK08] : Another vision based Programming by Demonstration (PbD) system is proposed in [RKK08]. The system consists of three main parts: The human grasp classification, the extraction of hand position relative to the grasped object, and finally the compilation of a robot grasp strategy. The hand shape is classified as one of six grasp classes, labelled according to Cutkosky's grasp taxonomy [CW86]. Instead of 3D tracking of the demonstrator hand over time, the input data consists of a single image and the hand shape is classified as one of the six grasps (Fig. 2.20) by finding similar hand shapes in a large database of grasp images. From the database, the hand orientation is also estimated. The recognized grasp is then mapped to one of three predefined Barrett hand grasps. Depending on the type of robot grasp, a precomputed grasp strategy is selected. The strategy is further

[HBZ06] M. Hueser, T. Baier, and J. Zhang. Learning of demonstrated grasping skills by stereoscopic tracking of human hand configuration. *In Proceedings of IEEE International Conference on Robotics and Automation*, 2006.

[RKK08] J. Romero, H. Kjellstrm, and D. Kragic. Human-to-robot mapping of grasps. *In Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, WS on Grasp and Task Learning by Imitation*, 2008.

[CW86] M. Cutkosky and P. Wright. Modeling manufacturing grips and correlations with the design of robotic hands. *In Proceedings of IEEE International Conference on Robotics and Automation*, pages 1533–1539, 1986.

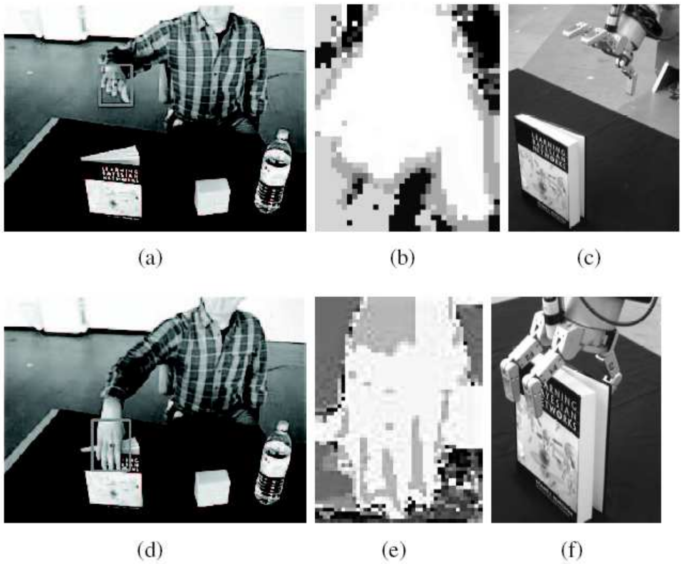
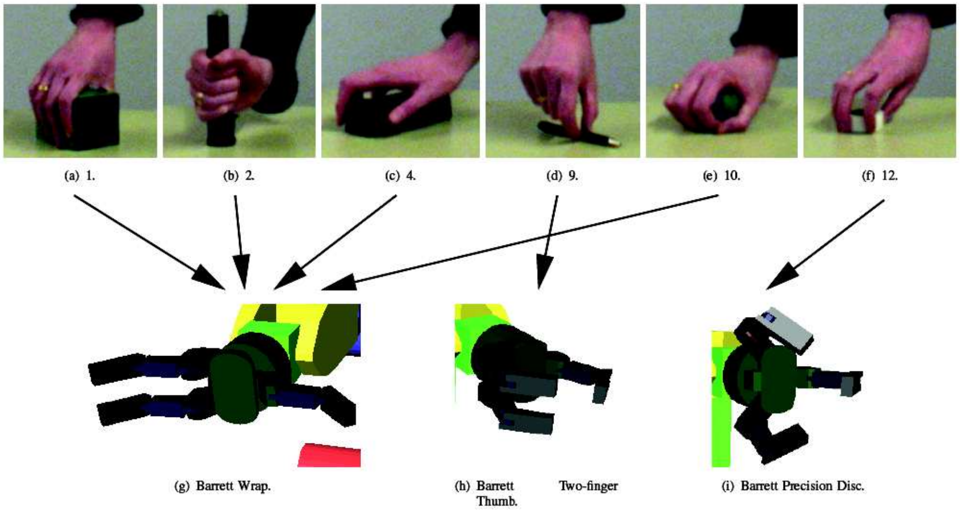


Figure 2.19: Examples of the robot's hand configuration selection: The tracked hand performing different grasps is shown in (a) and (d). The weights computed from the tracking algorithm are shown in (b) and (e). They are used to lead to the Barrett-Hand configurations as shown in (c) and (f) [HBZ06].

parameterized by the orientation of the hand relative to the object.

► These approaches enable objects telemanipulation or grasp type recognition. However, their learning data is based on the hand observation, i.e the joint angles, the hand trajectory or the hand shape. Thus the learning algorithm do not take into consideration the manipulated object properties. Consequently, these methods are not adapted to grasping previously unknown objects. In the remaining of this paragraph, we present two learning approaches that take into account some object features. The first work [OA02] roughly estimates the size and location of the object and relate them to the hand properties. The second [KWSN05] finds cylinder-likeness surfaces on the object and associate these surfaces with different hand shapes.

Oztop and Arbib [OA02] : The authors propose a grasping strategy based on mirror neurones. The latter were identified within a monkey's premotor area F5 and they fire not only when the monkey performs a certain class of actions but also when the monkey observes another monkey (or the experimenter) perform a similar action. It has been argued that these neurons are crucial for understanding of actions by others. In a grasping context, the role of the mirror system may be seen as a generalization from one's own hand to an other's hand. Thus, in a biologically motivated perspective, the authors propose a very detailed



(a) 1. (b) 2. (c) 4. (d) 9. (e) 10. (f) 12.

(g) Barrett Wrap. (h) Barrett Two-finger Thumb. (i) Barrett Precision Disc.

Figure 2.20: The six grasps (numbered according to Cutkosky's grasp taxonomy [CW86]) considered in the classification, and the three grasps for a Barrett hand, with human-robot class mappings ((a,b,c,e)→(g), (d)→(h), (f)→(i)) shown. a) Large Diameter grasp, 1. b) Small Diameter grasp, 2. c) Abducted Thumb grasp, 4. d) Pinch grasp, 9. e) Power Sphere grasp, 10. f) Precision Disc grasp, 12. g) Barrett Wrap. h) Barrett Two-finger Thumb, i) Barrett Precision Disc [RKK08].

model of the functioning of these neurones in grasp learning. They present a hand-object state association schema (Fig. 2.21) that combines the hand related information as well as the object information available. This method is capable of grasp recognition and execution (pinch, precision or power grasp) of simple geometric object models. The only object features used are the object size and location.

Kyota et al. [KWSN05] : Kyota et al. proposed a method for detection and evaluation of grasping positions (Fig. 2.22). Their technique detects appropriate portions to be grasped on the surface of a 3D object and then solves the problem of generating the grasping postures. Thus, points are generated at random locations on the whole surface of the object. At each point, the cylinder-likeness, that is the similarity with the surface of a cylinder, is computed. Then, the detected cylindrical points are evaluated to determine whether they are in a graspable portion or not. Once the graspable portions are identified, candidate hand shapes are generated using a neural network, which is trained using a data glove. Grasps are then evaluated using the standard wrench space stability criterion.

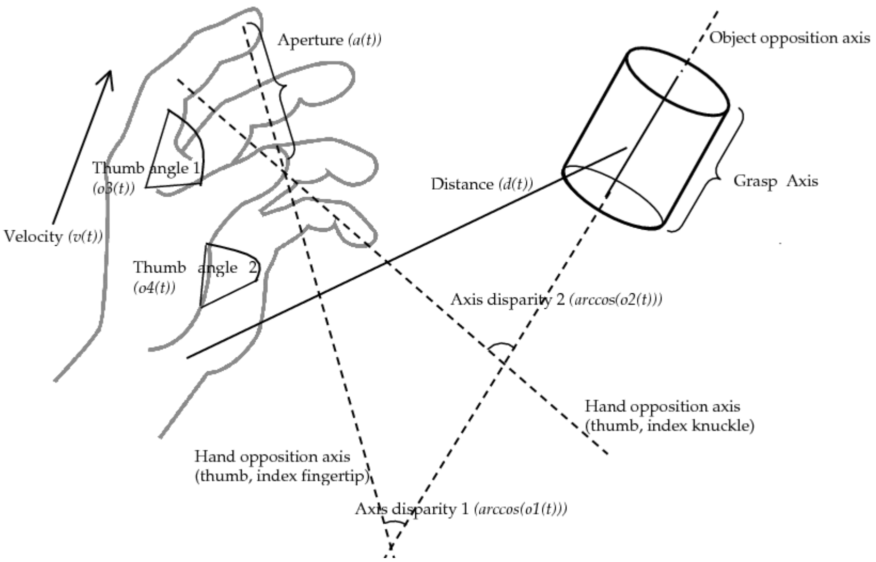


Figure 2.21: The components of hand state F(t)=(d(t), v(t), a(t), $o_1(t)$, $o_2(t)$, $o_3(t)$, $o_4(t)$). Note that some of the components are purely hand configuration parameters (namely v, o_3, o_4, a) while the others relate the hand to the object [OA02]

► Oztop and Arbib's approach can determine the grasp type of simple geometric objects. When facing new objects, it will roughly estimate their sizes and locations in order to identify the corresponding hand parameters and thus the grasp type in order to pick them up. Kyota's method finds different possible grasping regions on the object surface. However, it does not take into account object usage. Thus, these approaches can find stable grasps for pick and place operations but are unable to determine a suitable grasp for object manipulation.

2.3.2 Systems based on the object observation

Grasping strategies based on the object observation analyze its properties and learn to associate them with different grasps. Some approaches associate grasp parameters or hand shapes to objects geometric features in order to find good grasps in terms of stability [PMAT04, LFP07]. Other techniques learn to identify grasping regions in an object image [SDKN08,

[PMAT04] R. Pelossof, A. Miller, P. Allen, and T.Jebara. An svm learning appraoch to robotic grasping. *In Proceedings of IEEE International Conference on Robotics and Automation*, 2004.

[LFP07] Y. Li, J.L. Fu, and N. Pollard. Data-driven grasp synthesis using shape matching and task-based pruning. *IEEE Transactions on Visualization and Computer Graphics*, 13:(4):732–747, 2007.

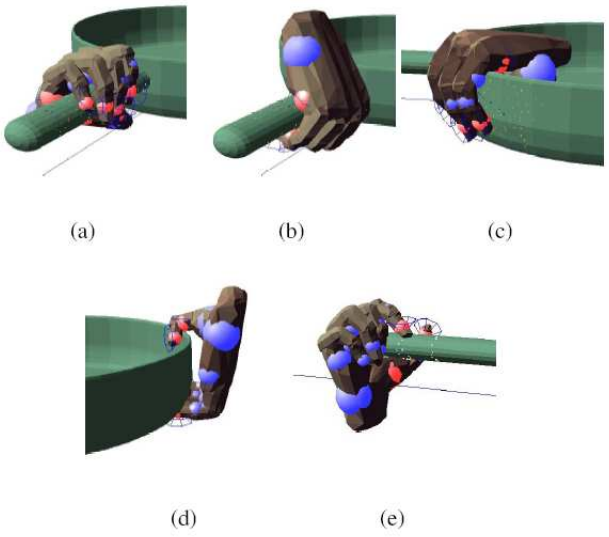


Figure 2.22: Grasping postures for a frying pan [KWSN05].

SLZS08]. These techniques are developed in the following.

Pelossof et al. [PMAT04] : Pelossof et al. used support vector machines to estimate the quality of a grasp given a number of features describing the grasp and the object. The machine learning algorithms used permit to build a regression mapping between object shape, grasp parameters and grasp quality. Once trained, this regression mapping can be used very efficiently to estimate the grasping parameters that obtain highest grasp quality for a new query set of shape parameters. Once more, the authors use simple object representation in their learning algorithm, such as spheres, cylinders etc. (Fig. 2.23). Since the grasp quality metric used, determines the magnitude of the largest worst-case disturbance wrench that can be resisted by a grasp of unit strength [FC92], the optimal grasps computed by the algorithm are good stable grasps adapted for pick and place operations.

Saxena et al. [SDKN08] : A learning approach for robotic grasping of novel objects is presented by Saxena et al. By novel objects, the authors mean ones that are being seen for the first time by the robot. Based on the idea that there are certain visual features that indicate good grasps, and that remain consistent across many different objects (such as coffee mugs handles or long objects such as pens that can be grasped at their mid-point), a learning approach that uses these visual features was proposed to predict good grasping points. The

[SDKN08] A. Saxena, J. Driemeyer, J. Kearns, and A.Y. Ng. Robotic grasping of novel objects using vision. *The International Journal of Robotics Research*, 27(2):157–173, 2008.

[SLZS08] M. Stark, P. Lies, M. Zillich, and B. Schiele. Functional object class detection based on learned affordance cues. *Computer Vision Systems*, pages 435–444, 2008.

[FC92] C. Ferrari and J. Canny. Planning optimal grasps. *In Proceedings of IEEE International Conference on Robotics and Automation*, 1992.

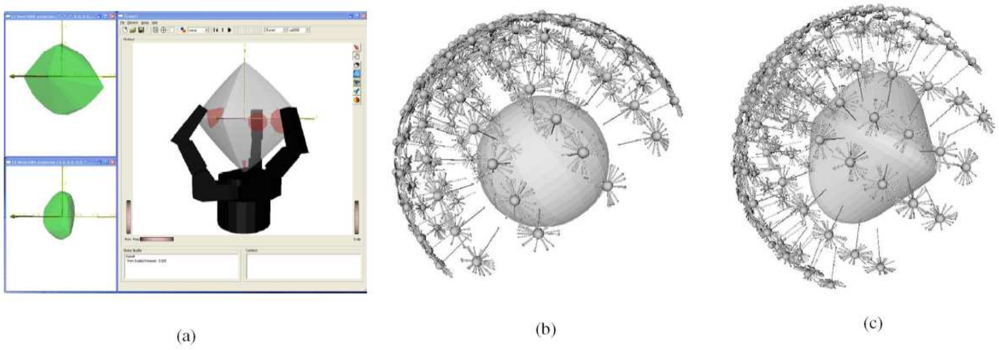


Figure 2.23: The GraspIt! simulator allows to import a robot hand model (here a Barrett hand) and a object model. (a) This image shows one successful grasp of the object. (b) and (c) For each object in the training set, 1600 grasp starting poses are generated and evaluated [PMAT04].

algorithm predicts a point at which to grasp a 3D object as a function of 2D images. A supervised learning is applied to identify images patches that contain grasping points. To do so, a labelled training set of synthetic images of objects labelled with the 2D location of the grasping point in each image is used (Fig. 2.24). The method starts by dividing the image into small rectangular patches, and for each patch compute local image features and predict if it is a projection of a grasping point onto the image plane. The chosen features represent three types of local cues: edges, textures, and color. Thus given two (or more) images of an object, the algorithm identify a few points in each image corresponding to good locations at which to grasp the object. This set of points is then triangulated to obtain a 3D location at which to attempt a grasp.

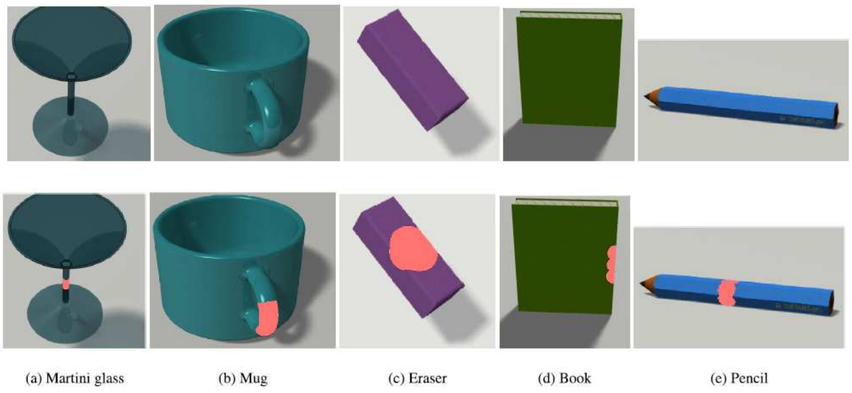

(a) Martini glass (b) Mug (c) Eraser (d) Book (e) Pencil

Figure 2.24: The images (top row) with the corresponding labels (highlighted in the bottom row) of the five object classes used for training. The classes of objects used for training were martini glasses, mugs, whiteboard erasers, books and pencils [SDKN08].

Stark et al. [SLZS08] : In a similar approach, Stark et al. propose a system for the detection of functional object classes, based on a representation of visually distinct hints on object affordances (affordance cues). Objects are classify based on their affordances in two categories: handle-graspable and sidewall-graspable (Fig. 2.25). Thus, the classification itself determines how to grasp the object. First the authors determine the interaction region as the set of object pixels that has been occluded by the human tutor in the course of an interaction. Affordance cues representation is based on geometric features extracted from a local neighborhood around that region. Using the affordance cue representation, an Implicit Shape Model (ISM) serves as the basis for the functional object category detection system in a 2D image. Once they are trained, these systems do not need any external help to perform the grasp. The authors, however, do not explain how the grasping points are obtained.

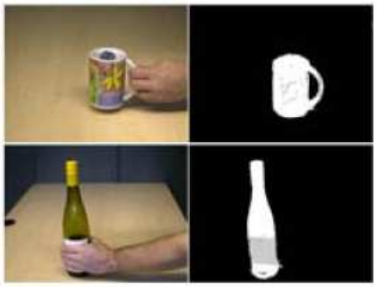


Figure 2.25: Handle-graspable Vs sidewall-graspable objects. Interaction regions are given in grey [SLZS08].

Li and Pollard [LFP07] : When a complete 3D model of the object is available, Li and Pollard treated grasping as a shape matching problem. Based on the idea that many grasps have similar hand shapes, they construct a database of grasp examples. Thus, given a model of a new object to be grasped, shape features of the object are compared to shape features of hand poses in the database in order to identify candidate grasps. These shape features capture information about the relative configurations of contact positions and contact normals in the grasp (Fig. 2.26). After shape matching, a number of grasps is obtained. Some of these grasps may be inappropriate to the task. They may fail to support the object securely or the main power of the grasp may be aligned in the wrong direction for the task. Thus, the authors used a grasp quality that takes into account both the hand and the task requirements to evaluate the computed grasps. By applying such a grasp quality, many grasps are pruned. Even though, the authors stated that the user should select manually the desired grasp from among the possibilities presented by the system because some of the grasps are unintuitive.

[SLZS08] M. Stark, P. Lies, M. Zillich, and B. Schiele. Functional object class detection based on learned affordance cues. *Computer Vision Systems*, pages 435–444, 2008.

[LFP07] Y. Li, J.L. Fu, and N. Pollard. Data-driven grasp synthesis using shape matching and task-based pruning. *IEEE Transactions on Visualization and Computer Graphics*, 13:(4):732–747, 2007.

Thus a fully autonomous system that generates natural grasps should take into account aspects other than ability to apply forces.

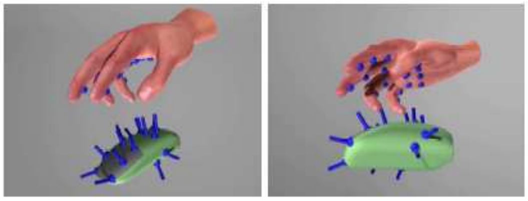


Figure 2.26: Hand pose for the mouse grasp. The figure shows contact points on the hand and object, and contact normals on the object surface. Note that the inside surface of the hand contains a great deal of information about the shape of the mouse. If similar features can be found on a new object, it may be possible to use the same grasp for the new object [LFP07].

► All these approaches learn to use objects features in order to compute a corresponding grasp. Thus, these approaches are capable to generalize to new objects. But what kind of grasps these techniques ensure? Pelossof's startegy can predict the quality of a grasp according to a stability criterion. Saxena's approach find grasping points on mugs handles or on elongated objects mid-points. Such contact points are adapted to some objects in terms of task-compatibility but when this approach encounter elongated objects such as screw-drivers or bottles, it will also identify a grasping region situated at these objects middles. Such grasps are not necessarily adapted to such kinds of objects. Stark's grasping strategy can only distinguish between two objects classes: handle-graspable (adapted for mugs) and side-graspable (adapted for bottles). This method does not take into account the variety of objects shapes and thus the variety of possible grasps. Finally, Li and Pollard strategy determine for one object different grasps and fail to choose the one adapted to the task-requirements. In the following, we discuss in details the limitations of the empirical approaches.

2.3.3 Discussion on Empirical Approaches

The main difficulty of analytical task-oriented approaches was task-modelling. Empirical approaches based on a human demonstration can overcome this difficulty by learning the task. For such approaches, when given an object and a task, the teacher shows how the grasp should be exactly performed. The robot is able afterwards to perform the task for the given object by itself. However, these systems are not fully autonomous when they face a new object or a new task (Fig. 2.27). To ensure the latter ability, rather than trying to reproduce humans grasping gesture, researchers developed systems that focus on objects observation. These approaches learn to find good grasping region in an object image or

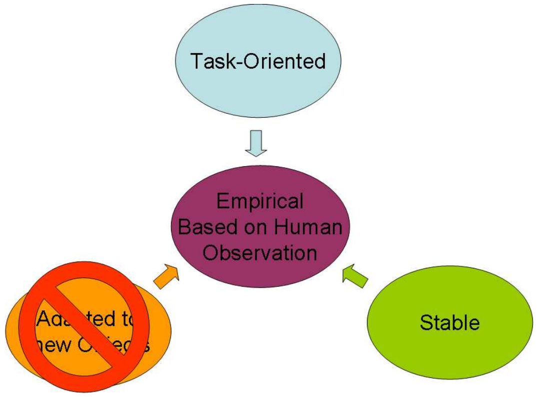


Figure 2.27: Empirical grasp synthesis approaches based on human demonstration ensure stability and task compatibility. However, they are not autonomous when facing new objects.

associate objects local features to different hand shapes. These systems can generalize to new objects but they find either stable grasps or generate for one object different grasps and fail to select automatically the one that best suits the task (Fig. 2.28). This selection

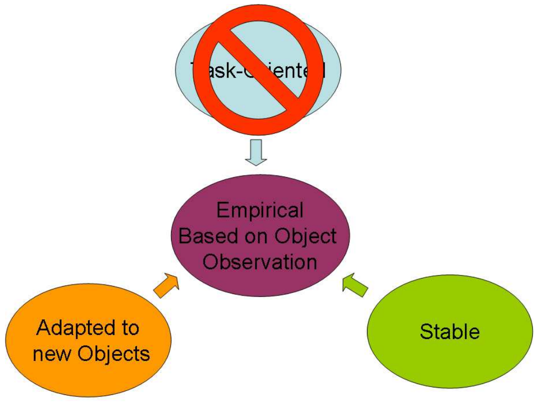


Figure 2.28: Empirical approaches based on objects observation ensure stability and adaptability to new objects. These systems generate a lot of possible grasping positions and fail to select the one that best suits the task.

is done manually or use a task-oriented quality criterion which is complicated to compute (same problem as analytical task-oriented systems). Thus, much research remains to be done to better understand human grasping and to develop algorithms that achieve natural grasps.

2.4 Conclusion

Generally, a grasp has to satisfy three main sets of constraints: constraints due to the robotic hand and its fingers capabilities, constraints due to object geometric features and constraints due to task requirements. Grasp synthesis involves determining a set of contacts on the surface of an object that meet these constraints. We have shown that in the literature, grasp synthesis, has been tackled with two different approaches: analytical or empirical. We have also shown that the ultimate goal of autonomous grasping strategy is to achieve stability and task compatibility when grasping new objects. If we summarize the works presented in the literature, we can conclude that: Force-closure analytical approaches can find stable but not task-oriented grasps (Fig. 2.13). Task-oriented analytical approaches suffers from a major problem: computational complexity when trying to model task requirements (Fig. 2.14). Empirical systems based on the observation of humans overcome task modelling difficulty by imitating humans grasping gesture. However, these systems are not fully autonomous when they face an object completely new (Fig. 2.27). Empirical systems based on objects observation are adapted to new objects but generate a lot of possible grasping positions and fail to select the one that best suits the task (Fig. 2.28). When trying to do this autonomously, they encounter the same problem of analytical task-oriented methods, which is task modelling. Consequently, we are standing in front of a loop!

How to break the loop? What grasping strategy can ensure stability, task compatibility and adaptability to new objects? Obviously, adaptability to new objects is ensured by learning objects characteristics that are relevant to grasping. Stability can be obtained by computing force-closure grasps. But what about the task requirements? On one hand, task modelling is difficult. On the other hand, learning specific task/hand performance is possible but works only on a particular object to perform a particular task. We believe that objects are made specifically in a way to make their grasp easier. Thus, objects shapes contains valuable information about the task they are designated to. Identifying such characteristics in an object may yield to find an appropriate grasp suitable to the required task. Consequently, we propose a grasping strategy that is an hybrid of empirical and analytical approaches. The empirical step will ensure task-compatibility by avoiding the analytical approaches task-modelling complexity. The analytical step permits contact grasp points generation. The following chapter details the proposed method.

Chapter 3

Grasping Objects By Components - Between the Concept and the Technical Approach

> ”Si nous prenons la nature pour guide, nous ne nous égarerons jamais.”
>
> *Cicéron*

The previous chapter shows that in order to ensure adaptability to new objects, a grasping strategy should use learning algorithms. Existing learning algorithms find grasps adapted to pick and place operations and fail to identify the grasp that human choose to pick a novel object and thus that is compatible with the task requirements. Consequently, a strategy that learns to associate a grasp to an unknown object/task is still an unsolved problem. We intend to take this challenge. Grasping a novel object is a task we perform without thinking about. Thus, what are the factors taken into consideration when choosing a specific grasp configuration? What should the algorithm learns in order to pick a new object in the same manner as humans? In other words, what parameters are relevant to new objects grasping? Are these parameters related to the hand characteristics? Are they related to the object features?

We show first, in this chapter, that from the neuroscience point of view, objects features are used differently for recognition and for grasping. In a second time, we show that a low-level part object representation provide a useful preliminary to its grasping. We emphasize then the role of such representation in determining an unknown object good grasp.

3.1 Grasping from the Neuroscience Point of View

Humans are capable of reaching and grasping objects with great dexterity. When we reach out to pick up an unfamiliar object, the opening of our fingers and the orientation of our hand reflect the size of the object and its orientation before we make contact with it [Jea88]. A work with neurological patients 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, GMBR94]. The following paragraphs explain briefly experiments that lead to such results.

3.1.1 Separate pathways for perception and prehension

The study is conducted on two patients, DF and RV, with lesions in different parts of the cerebral visual pathways. DF is a 34-year-old woman who developed visual-form agnosia following anoxia from carbon monoxide poisoning. She's unable to identify or recognize familiar faces, line drawings of common objects or even simple geometric shapes. RV is a 55-year-old woman, who had sustained bilateral lesions of the occipitoparietal cortex. She had no difficulty in discriminating between different objects shapes. Twelve different shapes were used to compare DF's and RV's ability to discriminate between shapes and to use shape information to control grasping. When DF and RV were presented with pairs of these shapes, they showed different discrimination abilities: DF failed to perceive wether two objects had the same or different shapes, RV had little difficulty in making such a discrimination. The opposite pattern was observed when DF and RV were asked to pick up these objects. Even though DF failed to discriminate between these different objects, she had no difficulty in finding stable grasp points on the circumference of these objects. In contrast to DF, RV often chose very unstable grasp points. This suggests that there are two cortical processing streams operating on different coding principles for perception and for action [GMJC91, GMBR94]. Thus, object recognition and object grasping have different degrees of dependance on the object features. In other words, these features are used differently for perception and for action.

These two streams, called ventral and dorsal, were also identified in the macaque monkey by Ungerleider and Mishkin [UM82]. The authors originally proposed that both streams have their origins in the primary visual cortex V but one extends ventrally (from the visual cortex to the inferior temporal cortex IT) and is assumed to subserve object recognition. The other

[Jea88] M. Jeannerod. The neural and behavioral organization of goal directed movements. *Oxford, Clarendon Press*, 1988.

[GMJC91] M.A. Goodale, A.D. Milner, L.S. Jakobson, and D.P Carey. A neurological dissociation between perceiving objects and grasping them. *Nature*, pages 349:154–156, 1991.

[GMBR94] M.A. Goodale, J.P. Meenan, H.H. Blthoff, and C.I. Raciot. Separate neural pathways for the visual analysis of object shape in perception and prehension. *Current Biology*, 4, 1994.

[UM82] L.G. Ungerleider and M. Mishkin. Two cortical visual systems. *MIT Press*, pages 549–585, 1982.

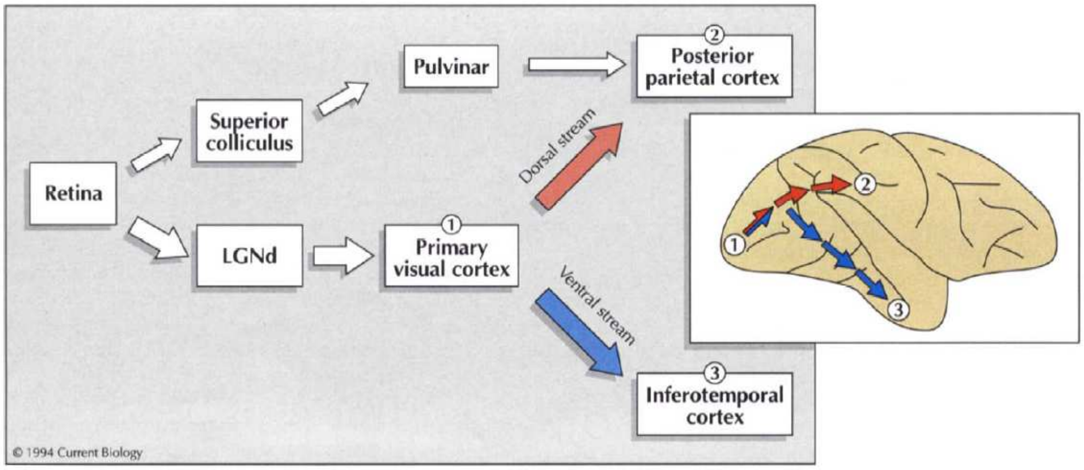


Figure 3.1: Schematic diagram of the main routes whereby retinal inputs reach the dorsal and ventral streams.

extends dorsally (from the visual cortex to posterior parietal cortex PP) and is responsible for localizing objects in visual space (Fig. 3.1).

3.1.2 Knowledge based grasping

Castiello and Jeannerod [CJ91] studied impairment of grasping in a patient (AT) with a lesion of the visual pathway that left the PP, IT, and the pathway $V \rightarrow IT$ relatively intact, but impaired the pathway $V \rightarrow PP$. This patient is, in some sense, the opposite of DF. She can use her hand to pantomime the size of a cylinder, but cannot preshape appropriately when asked to grasp it. When the stimulus used for the grasp was not a cylinder, but rather a familiar object such as a lipstick which is a part of the subject's knowledge, AT showed a relatively adaptive preshape. In other words, the authors showed that previously learned knowledge plays major role in visually guided grasping in humans and in monkeys. This indicates that learning from previous knowledge is relevant for grasping novel objects.

3.1.3 Discussion

The previous two studies suggest that:

- Objects features are coded differently for their recognition and for their grasping.
- Knowledge or learning is relevant to objects grasping.

Thus, a grasping strategy should be able, using a learning algorithm, to grasp objects without recognizing them. It's obvious 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 objects shapes and sizes, predicting every possible object the robot could encounter is impossible. Thus, a robot will

[CJ91] U. Castiello and M. Jeannerod. Measuring time to awareness. *Neuroreport*, 12:787800, 1991.

certainly have to grasp non-identified objects and so are humans. In such situations, what objects features may yield to a good grasp?

3.2 Recognition By Components

Humans can easily distinguish between things they have seen in the past and novel 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].

3.2.1 RBC Theory

According to the Recognition By Components theory of Biederman, or RBC, we are able to recognize objects by separating them into geons, or geometric ions. Geons are composed of different shapes (i.e. cylinders, cones, etc.) that can be assembled in various ways to form an unlimited amount of objects. These geons are derived qualitatively using four attributes of generalized cylinders (Fig. 3.2). Three of the attributes describe characteristics of the cross section: its shape, symmetry and size. The fourth attribute describes the shape of the axis. In the following, we detail Biederman's example of identifying a non-sense object.

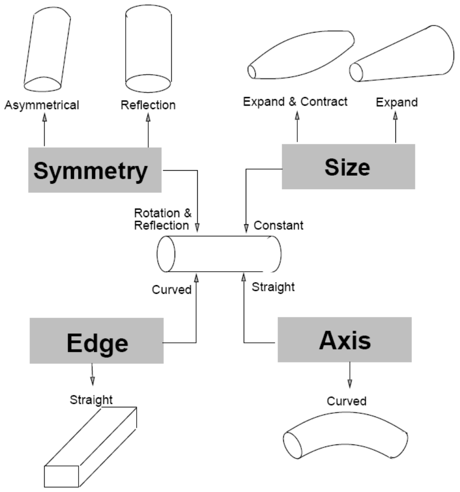


Figure 3.2: Geons are defined by variation in three attributes of the cross-section: (i) curved vs. straight edges; (ii) constant vs. expanded vs. expanded and contracted size; (iii) mirror and rotational symmetry vs. asymmetrical and one of the axis shape (curved vs. straight axis) [Bie87].

[Bie87] I. Biederman. Recognition-by-components: A theory of human image understanding. *Psychological Review*, 94:115–147, 1987.

[MN78] D. Marr and H.K. Nishihara. Representation and recognition of three dimensional shapes. *Proceedings of the Royal Society of London*, Series B. 200:269–294, 1978.

3.2.2 Identifying Unknown Objects

Biederman suggests that segmenting objects for their identification does not depend on our familiarity with these objects. Thus, we conduct the same process for any object, whether it is familiar or unfamiliar. Consider for example the object shown in Figure 3.3. Despite its unfamiliarity, we are able to identify this object by segmenting it into parts at regions of deep concavity. This object resembles a hot dog cart. We can see the large block as the central food storage and cooking area, the rounded part underneath as a wheel, the large arc on the right as a handle, the funnel as an orange juice squeezer and the various vertical pipes as an umbrella supports. This object may not be a good cart, but we can see how it might be related to one. Thus, Biederman concludes that even nonsense objects may be identified by decomposing them into parts. But what about grasping an unfamiliar object? Does its part decomposition emphasize a specific grasp?

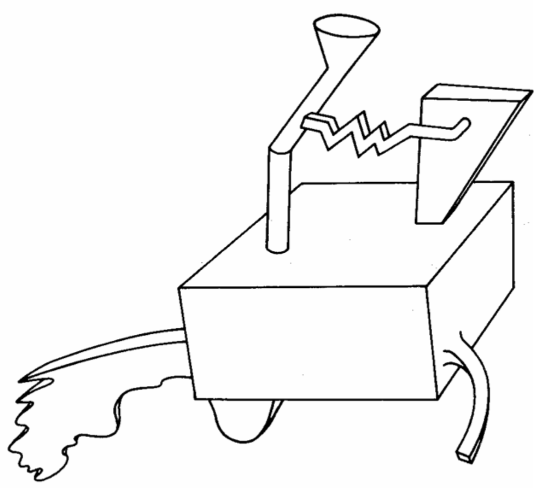


Figure 3.3: Biederman's nonsense object [Bie87].

3.2.3 Discussion

The RBC theory shows that we are able to identify many objects using combinations of a modest number of primitives: the geons. But RBC theory is incomplete in that geons and the relations between them fail to distinguish many real objects. For example, a well known failure of the RBC theory is its inability to distinguish between a pear and an apple. The latter are easily distinguished by humans but lack the corners and edges needed for RBC theory to differentiate them. The reader should note that, from a grasping point of view, one does not need to distinguish a pear from an apple. The two of them are grasped in the same manner or more precisely by the same sub-part. This proves that grasping an object is less-constrained than recognizing it. The following section details the extension of the RBC theory to the Grasping By Components approach.

3.3 Objects Decomposition in the Grasping Literature

Before detailing our Grasping By Components approach, we present briefly some works in the grasping literature using objects decomposition into single parts.

Miller et al. [MKAC03] : The authors model objects as set of shape primitives such as spheres, cylinders, cones and boxes. They define then 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.

Goldfeder et al. [GALP07] : Miller's planner required a manually constructed primitive decomposition of the object. More recently, the authors removed the need for a manual decomposition and introduced a multi-level superquadrics representation.

Lopez-Damian et al. [LDSA05] : The authors 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.

Discussion: Since the grasping problem induces a huge degrees of freedom number, all these 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 that human choose to pick an object and that is consequently adapted to the task requirements.

3.4 Grasping By Components: the Concept

By taking inspiration from the RBC framework, we represent 3D objects as an assembly of volumetric primitives. When considering objects we use for everyday tasks on a part-representation level, we make the following assumptions:

[MKAC03] A.T. Miller, S. Knoop, P.K. Allen, and H.I. Christensen. Automatic grasp planning using shape primitives. *In Proceedings of IEEE International Conference on Robotics and Automation*, 2003.

[GALP07] C. Goldfeder, P.K. Allen, C. Lackner, and R. Pelossof. Grasp planning via decomposition trees. *In Proceedings of IEEE International Conference on Robotics and Automation*, 2007.

[LDSA05] E. Lopez-Damian, D. Sidobre, and R. Alami. Grasp planning for non-convex objects. *36th International Symposium on Robotics, ISR*, 2005.

- Objects are equipped with a part designed specifically to make their grasp easier.

- Objects with similar components are grasped in the same manner.

- The relative sizes of object components is crucial for the graspable part selection.

All these elements conduct us to the elaboration of the ”Grasping By Components” strategy. In the following, we detail each observation.

3.4.1 Every object has a handle

When considering commonly used objects on a part-representation level, we notice that they are equipped with a part that facilitates their grasp. Figure 3.4 shows some familiar objects. The black part indicates the component that humans choose to grasp these objects. Thus, it is also the part that satisfies the task requirements. This part is what we call the object natural grasping component or more simply the object handle. We all agree that the handle of a cup or a mug is its curved part and that the handle of a bottle, a pencil or a spoon is their elongated parts. But what about the handle of an unknown object? Which part of the object is there to facilitate its grasp? If we can determine the handle of an unknown object, we can easily find a grasp of that object adapted to the task it is designated to.

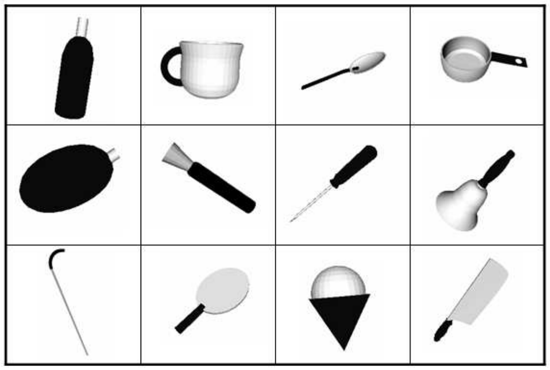


Figure 3.4: Some objects used for everyday tasks. The black part indicates the object handle.

3.4.2 Grasping Similarities

Many objects with similar components are grasped in the same manner. Bags, buckets, mugs and cups are roughly composed of a cylinder and a curved cylinder. Even though the arrangement of these components is different for these objects, they are all grasped by their curved component (Fig. 3.5). Thus, the choice of an object graspable part is influenced by the shape of its constituting single parts. Objects parts orientation is less relevant to that choice.

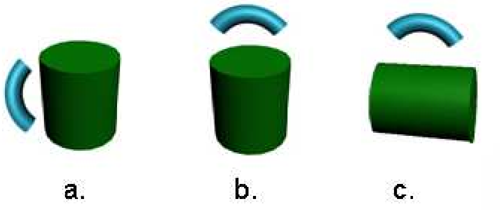


Figure 3.5: The choice of an object graspable part is influenced by the shape of its constituting parts, independently from their orientations, i.e: a) a mug, b) a bucket and c) a bag are all grasped by their curved part.

3.4.3 Handle Vs. Relative Sizes

The relative sizes of object components is crucial for the graspable part selection. The example presented here also shows how objects are designed in a way to make their grasp easier and in accordance with their functions. Let us examine some alcohol glasses shapes and sizes. More precisely, we consider wine, champagne and brandy glasses. Although, all these glasses are composed of three parts: the bowl, the stem and the foot, they are grasped differently (Fig. 3.6). Wine glasses are characterized by their wide bowl which gives the wine the chance to breathe. Champagne flutes are characterized by a narrow bowl on the top. This is designed to keep sparkling wine desirable during its consumption. Wine and champagne glasses are designed to be held by the stem to help prevent the heat from the hand from warming the alcohol. On the other hand, brandy glasses have a short stem. They are designed to be held by the bowl. The wide bowl of the brandy glass accommodates the hand, which warms the brandy for drinking. Thus, the choice of the graspable component is influenced by the objects parts relative sizes.

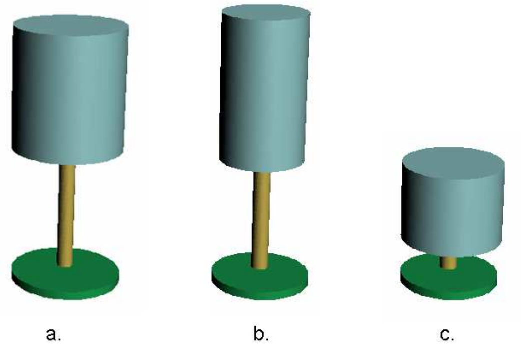


Figure 3.6: Roughly approximation of: a) a wine glass, b) a champagne flute and c) a brandy glass.

In summary, we can say that information about an unknown object parts shapes and sizes may emphasize a specific part for grasping. This leads to the ”Grasping By Components” strategy.

3.4.4 Grasping By Components

The diagram below (Fig. 3.7) illustrates a description of the proposed grasping strategy [KSP07, KS08a]. We have shown previously that by taking inspiration from the recognition by components theory, we represent objects as a set of components in order to identify the graspable one. Hence, objects are first decomposed into single parts. Since information about these parts shapes and sizes are required, a geometrical description of these parts is crucial. This is ensured by the approximation step. Consequently, the learning process disposes of a geometric representation of the object components and uses it to perform an analogue of the human choice of the grasping component. Thus, our approach will learn to imitate humans selection of the object natural graspable part. The different steps of the proposed approach are the following:

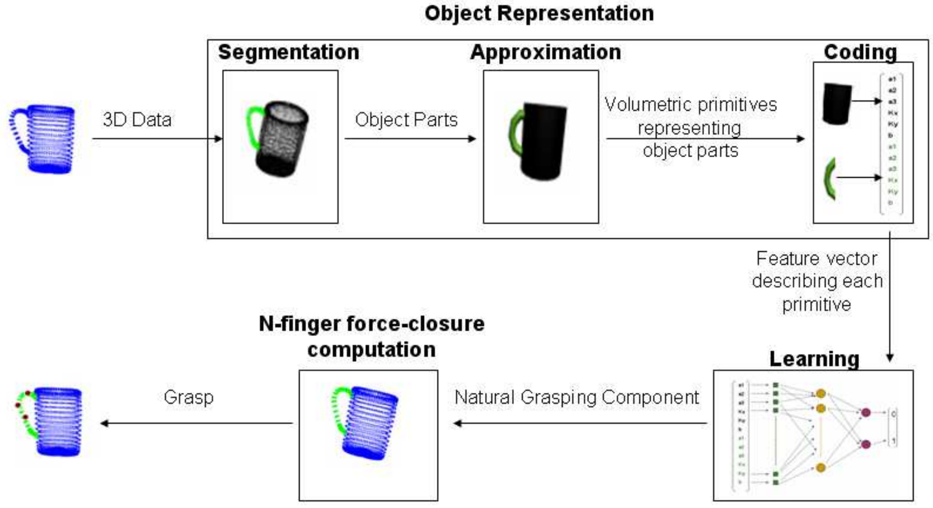

Object Representation
Segmentation
Approximation
Coding
3D Data
Object Parts
Volumetric primitives representing object parts
Feature vector describing each primitive
Learning
N-finger force-closure computation
Natural Grasping Component
Grasp

Figure 3.7: The different steps of the proposed approach.

Part Segmentation: RBC proposes that objects be segmented into parts at surface of deep concavities. The reason for this approach is that it conforms well with human intuition about parts, and does not require a priori knowledge of part shapes even in the case of a nonsense object. We use in our experiments synthetic (CAO models) and real objects (obtained from

[KSP07] S. El Khoury, A. Sahbani, and V. Perdereau. Learning the natural grasping component of an unknown object. *In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2957–2962, 2007.

[KS08a] S. El Khoury and A. Sahbani. Handling objects by their handles. *IEEE/RSJ International Conference on Intelligent Robots and Systems, WS - grasp and task learning by imitation*, pages 58–64, 2008.

a 3D laser scanner or from a 3D reconstruction using a vision system [WD08]) models. With such models, object surface discontinuities can be easily obtained. A common 3D objects segmentation technique is to compute surface features which contrast boundary and non-boundary points and then to decompose the object into parts at boundary points. The key issue here is how to reliably locate the part boundaries?

Shape Approximation: Once the object is decomposed into several parts, the problem of shape approximation can be stated as follows: given a set of data points on a each part, find a model which is the best description of that part. The question to be answered is: Which model has to be used? Biederman used qualitative geons to represent objects parts. Is this representation sufficient for the GBC method? We remind the reader that this step is crucial for providing information about object parts shapes and sizes to the learning algorithm.

Learning: The learning step permits to learn the natural grasping component of an object using information about its sub-parts shapes and sizes. This step is performed by imitating humans choice of objects graspable parts. The difficulty here is to determine the training data. What objects should be selected for training and is the algorithm capable to generalize to novel objects?

Contact Points: Once the handle of an object is identified, a grasp is computed on that part. How the contact points are determined? Is the grasp obtained stable?

In the following paragraphs, we particularly focus on two topics: (1) object modelling that includes object part segmentation and volumetric part models choice and recovery and (2) the Natural Grasping Component (NGC) learning. Computing contact points will be addressed in details in the next chapter.

3.5 Object Modeling

Object modeling is obtained by segmenting a 3D object model into its meaningful parts and by describing the shape of each part. To obtain such a description, we need to know: what are the parts? and what is the model of each part? the former is the issue of object segmentation, while the latter deals with object parts identification. These questions will be addressed in the following two sections (Fig. 3.8).

3.5.1 Object Segmentation

The problem of 3D object segmentation into parts is to decompose the complete object surface into different meaningful regions. The Recognition By Components theory proposes that objects be segmented into parts at deep surface concavities which conforms well with

[WD08] G. Walck and M. Drouin. Reconstruction 3d progressive et rapide. *MajecStic08*, 2008.

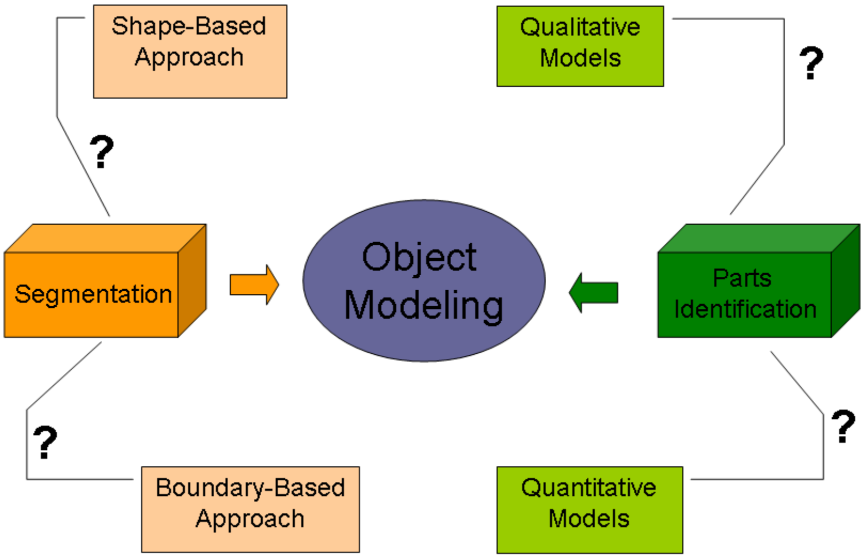


Figure 3.8: Object Modeling considers two main problems: object segmentation and object parts identification.

human intuition about parts. But RBC uses line drawings of objects extracted from intensity images by an edge detection operation to derive their segmentation. Since our input data is a 3D mesh, we need a segmentation scheme that does the same in 3D and that does not require a priori knowledge of part shapes. Within the literature of 3D mesh segmentation, there are two main approaches that satisfy this condition, shape-based and boundary-based approaches (Fig. 3.8).

3.5.1.1 Shape-Based Vs. Boundary-Based Approaches

This paragraph outlines shape and boundary based segmentation approaches. It also shows that boundary-based techniques perform an object decomposition that conforms human visual perception of volumetric parts and thus will be used in our Grasping By Components approach.

Shape-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, S, such as a cylinder, a cuboid and a cone. They generate then a hypothesis of the object representation as an assembly of shapes chosen from S. 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. This kind of approach performs object segmentation into parts directly using part shapes as constraints. 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. In contrast, boundary-based approaches are two-level methods: the first level deals with determining different objects parts while their identification is of the second level concerns.

Boundary-Based Approaches: 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, to our knowledge only three main features 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 expen-

[SLM94] F. Solina, A. Leonardis, and A. Macerl. A direct part-level segmentation of range images using volumetric models. *In Proceedings of IEEE International Conference on Robotics and Automation*, pages 2254–2259, 1994.

[LJS97] A. Leonardis, A. Jaklic, and F. Solina. Superquadrics for segmentation and modeling range data. *IEEE Transactions on Pattern Anal Mach Intell.*, 19:12891295, 1997.

[GB93] A. Gupta and R. Bajcsy. Volumetric segmentation of range images of 3d objects using superquadric models. *CVGIP: Image Understanding*, 58(3):302–326, 1993.

[DPR92] S.J. Dickson, A.P. Pentland, and A. Resenfeld. From volumes to views: An approach to 3d object recognition. *CVGIP: Image Understanding*, 55(2):130–154, 1992.

[CJB03] L. Chevalier, F. Jaillet, and A. Baskurt. Segmentation and superquadric modeling of 3d objects. *Journal of WSCG*, 11(1), 2003.

[WL97] K. Wu and M.D. Levine. 3d part segmentation using simulated electrical charge distributions. *IEEE Transactions On Pattern Analysis and Machine Intelligence*, 19 No. 11:1223–1235, 1997.

[MP77] R.S. Millman and G.D. Parker. Elements of differential geometry. *Prentice-Hall Inc*, 1977.

sive [MW99, PRF02, RB02, RKS00]. Zhang et al. [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 Chen et al. [CG06].

We implemented the latter two approaches. In the following, we present them briefly and discuss their limitations. Before detailing these approaches, we give a description of 3D features used to detect boundaries between different object parts, i.e Gaussian curvature and concaveness estimation.

3.5.1.2 Gaussian curvature and concaveness estimation for a boundary-based segmentation

We previously outlined that boundary-based approaches will be used for objects segmentation. These approaches detect boundaries by estimating Gaussian curvature and concaveness of the object vertices. This section details these 3D features extraction.

Gaussian curvature and Surface behavior: Curvature estimation is a fundamental tool for analyzing and describing a surface's behavior [MP77, Gra93]. The radius of an osculating circle can be used to measure curvature of a line at a given point (Fig. 3.9). For each point p on a surface, there are two principal osculating circles (Fig. 3.10), thus we obtain two principal curvature radii k_1 and k_2. The gaussian curvature at p is defined as $k = 1/k_1k_2$.

To describe the geometric behavior of a surface, one can consider the gaussian curvature. Given a surface S, a point p on S belongs to the following categories [MP77, Gra93]: elliptic if $k > 0$; hyperbolic if $k < 0$; parabolic if $k = 0$ (Fig. 3.11). From this definition, we can distinguish different surface regions using the gaussian curvature. A 3D mesh is composed of distinct components. The points on each individual component have elliptic or parabolic

[MW99] A. Mangan and R. Whitaker. Partitioning 3d surface meshes using watershed segmentation. *IEEE Trans Vis Comput Graph*, 5(4):308321, 1999.

[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. *Computer-Aided Design*, 2002.

[RKS00] C. Rossl, L. Kobbelt, and H.P. Seidel. Extraction of feature lines on triangulated surfaces using morphological operators. *Smart Graphics, AAAI Spring Symposium, Stanford University*, pages 71–75, 2000.

[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. *International Conference on Image Processing*, (3):273276, 2002.

[CG06] L. Chen and N.D. Georganas. An efficient and robust algorithm for 3d mesh segmentation. *Multimedia Tools Appl.*, 29(2):109–125, 2006.

[Gra93] A. Gray. The gaussian and mean curvatures. *Modern Differential Geometry of Curves and surfaces*, pages 279–285, 1993.

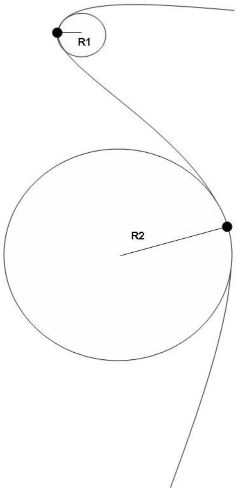


Figure 3.9: Curvature = 1/(curvature radius)

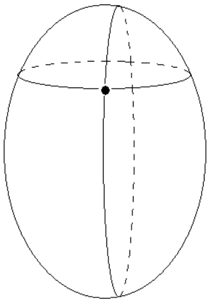


Figure 3.10: Curvature = $1/(k_1 k_2)$

behavior. Therefore, we can divide the surface into disjunctive regions by detecting the boundaries with hyperbolic behavior ($k < 0$). Considering a polygonal surface, curvature estimation can be done using only the information that is given by that surface itself [Gra93]. Hence, given a vertex p on a polygonal mesh, the discrete gaussian curvature $k(p)$ is defined by (Fig. 3.12):

$$k(p) = \frac{3(2\pi - \sum_i^N \theta_i)}{\sum_i^N A_i} \tag{3.1}$$

where :

- N, number of triangles at p;
- θ_i, represents the interior angle of the triangle at p;
- A_i, represents the area of the corresponding triangle.

Therefore, we can easily determine if a vertex is hyperbolic or not by computing $k(p)$.

[Gra93] A. Gray. The gaussian and mean curvatures. *Modern Differential Geometry of Curves and surfaces*, pages 279–285, 1993.

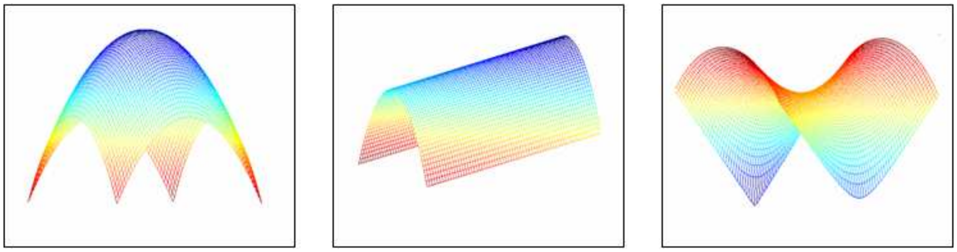


Figure 3.11: The first, the second, and the third patches show elliptic, parabolic and hyperbolic behavior. Thus they have positive, zero and negative Gaussian curvature respectively.

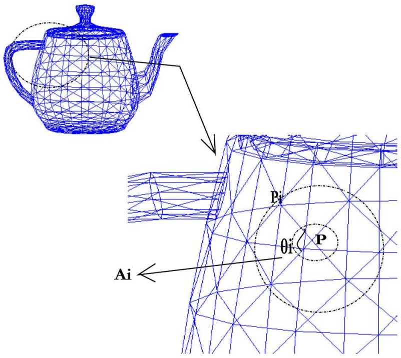


Figure 3.12: Discrete Gaussian curvature computation.

Concaveness estimation: Concave vertices are also pertinent to objects decomposition into meaningful parts. In this section, we will estimate concaveness and convexity of the vertices on a mesh.

In [SZL92], Schroeder et al. defined the average plane H of a vertex p by the normal vector N_p of p and a center point p_c, and the distance d from p to H.

$$N_p = \frac{(\sum_i^N n_i A_i)}{\sum_i^N A_i} \qquad p_c = \frac{(\sum_i^N p_i A_i)}{\sum_i^N A_i} \tag{3.2}$$

$$d = |\overrightarrow{pp_c}.N_p| \tag{3.3}$$

where A_i, p_i and n_i are the area, the vertex and the normal of the adjacent face around p, respectively, and $\overrightarrow{p_c p}$ denotes the vector from p to p_c. The signed distance from p to H is defined as follows:

$$d_s = \overrightarrow{pp_c}.N_p \tag{3.4}$$

Based on the signed distance, the vertex p is defined as convex if $d_s \leq 0$, and as concave if $d_s > 0$. This can be shown in figure 3.13. If a vertex p is convex, the vectors $\overrightarrow{pp_c}$ and N_p are

[SZL92] W. Schroeder, J. Zarge, and W. Lorensen. Decimation of triangle meshes. *Proc. SIGGRAPH, Computer Graphics*, 25(3):6570, 1992.

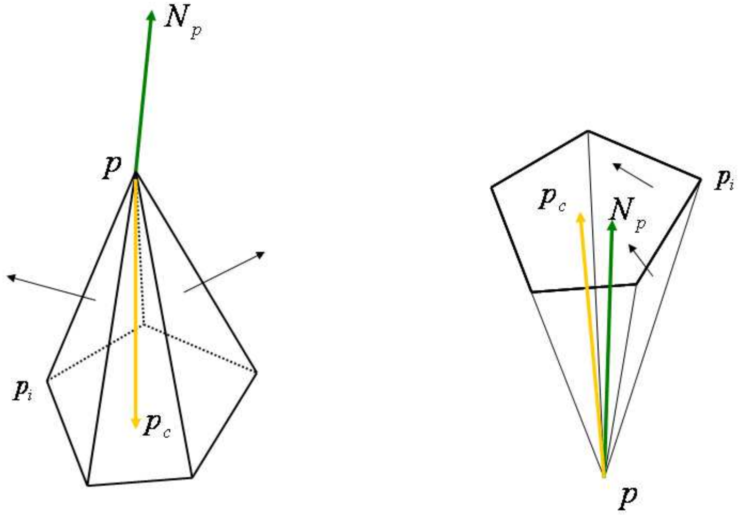


Figure 3.13: Convex vertex to the left vs. concave vertex [CG06].

in roughly opposite directions, whereas if p is concave, these vectors are in roughly the same direction.

3.5.1.3 First Segmentation Algorithm

Based on Gaussian curvature analysis, Zhang et al. [ZPKG02] proposed a simple segmentation algorithm that consists of three major steps: Gaussian curvature estimation, boundary detection and region growing. In the following, we present briefly the approach and issues encountered after its implementation.

The segmentation process: Beginning with a complete 3D object model composed of triangle meshes, a Gaussian curvature is estimated for each vertex of a triangle mesh. A specified threshold is then applied to label vertices as boundary or seed. Vertices of highly negative curvature are labelled as boundaries between two parts while the rest are labelled as seeds belonging to potential object parts. This threshold is determined in a heuristic way depending on object and mesh resolution. After the vertices are labelled, a region-growing operation is performed on each vertex labelled as seed. Region growing is performed as follows. Starting from a seed vertex p, a unique region label is first assigned to the vertex. Second, all the neighbors p_i labelled as seeds initially are then labelled with the same region number as the point p. The same labelling process is performed for each neighbor p_i. This process terminates when the grown region is surrounded by boundary vertices. This is repeated for each seed vertex, but not for a vertex which has been grown and already labelled uniquely. After the seed vertices are assigned new labels, a labelling process is needed for

[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. *International Conference on Image Processing*, (3):273276, 2002.

each boundary vertex. Therefore, given a boundary point x, its mesh neighbors x_i are first sorted in ascending order based on their Euclidean distance to the point x. Next, the vertex x_k is selected from the ordered neighbors where x_k is the first vertex in the list with a region label and not a boundary label. The boundary vertex x is then labelled the same as the vertex x_k. Finally, almost every vertex should now have a region label and thus assigned to different object parts. Remaining isolated vertices are assigned to their neighbors regions.

Discussion: This segmentation method works very well for low resolution objects. But when dealing with high resolution objects models obtained for example from a 3D laser scan, we encounter two major problems with this approach:

- Failure to determine accurately boundaries between different objects parts. When an object model is densely represented with polygonal faces, its surface evolves continuously. Consequently, boundary vertices will have an elliptic behavior rather than an hyperbolic one. Thus, the algorithm will fail to identify these vertices as boundaries.

- Zhang segmentation has a problem dealing with high resolution objects models having concave corners located between different boundaries. Figure 3.14 illustrates the example of a high resolution union of three spheres. The lines in red represents boundaries between these spheres. The latter are also surrounding a concave area. This area has an elliptic behavior and will not be identified by the algorithm as a boundary region. Thus, when the region growing process is activated, this area break the boundaries and the different regions will be merged together. We note that when dealing with low resolution objects, the latter concave area is reduced to one vertex. Zhang has solved the problem of concave corners by merging isolated vertices with their corresponding neighbors. Examples on decomposing objects with different resolutions are presented in the experimental results chapter.

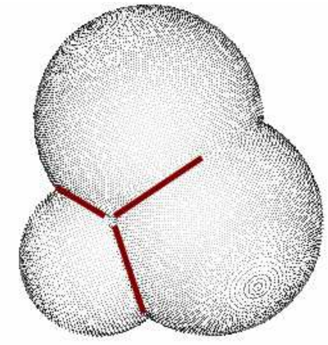


Figure 3.14: Concave area surrounded by three boundaries.

3.5.1.4 Second Segmentation Algorithm

Chen et al. [CG06] segmentation algorithm is based on Gaussian curvature and concaveness estimation. The authors use a watershed segmentation approach that consists of three steps, minima detection, plateau erosion and region merging. In the following, we describe briefly these steps.

The segmentation process: After the Gaussian curvature and the concaveness estimation of each vertex on the 3D triangular mesh of the object model, vertices are labelled as boundary or inner region vertices. A vertex p is a boundary vertex if it has an hyperbolic behavior or is concave. The minima detection step finds then the local minima (the non-boundary regions). The boundary regions are the plateaus. All the local minima are marked with a unique label. Minima detection permits to define the basins which are the core areas of the final regions. The next step is to segment the plateaus. Erosion starts at the boundaries between plateaus and minima. It converts iteratively plateaus to their neighbor basins until all the plateaus vertices are processed. Due to small details or noise on the polygonal surface, the watershed transformation always yields an over-segmented result. Region merging is an essential stage for watershed-based segmentation. Thus small regions are merged with bigger regions. The destination adjacent region to which the small region will merge is determined by the boundary length between the corresponding two regions.

Discussion: Chen segmentation algorithm overcomes Zhang approach weakness. In the following we explain the advantages of this method:

- Chen method uses multi-ring neighborhood in order to compute a 3D object surface features such as the gaussian curvature. Thus, when a model is densely represented with polygonal faces, a multi-ring neighborhood permit to accurately catch their geometric behavior. One should see the multi-ring behavior as a zoom out that gives more general view of the local features behavior. Using multi-ring neighborhood resolves the first drawback of Zhang method that uses only one ring neighborhood.

- Another advantage of this method is that it uses not only gaussian curvature for segmenting objects but also uses concaveness estimation. This is crucial for objects with a concave area surrounded by boundaries (Fig. 3.14). In such cases, the concave area will also be identified as a boundary. This induces a detection of the whole boundaries between different objects parts.

This segmentation approach succeeds in decomposing high resolution 3D laser scanned objects. The corresponding results are presented in the last chapter.

[CG06] L. Chen and N.D. Georganas. An efficient and robust algorithm for 3d mesh segmentation. *Multimedia Tools Appl.*, 29(2):109–125, 2006.

3.5.2 Object Parts Identification

Object segmentation produces a set of parts. The next task is to generate a description for each part. This step is crucial as the learning algorithm needs a compact object description. We represent objects as an assembly of superquadrics. To emphasize our choice, we give first a brief review on part models used in the literature and then we discuss part models recovery.

3.5.2.1 Part Models

Part models are volumetric primitives which describes shapes of object parts. The volumetric primitives developed in previous research can be categorized as qualitative or quantitative.

Qualitative models: Qualitative models provide characteristics which are useful for symbolic object description. Biederman's geons [Bie87] are qualitative part models. They are thirty-six volumetric component shapes described in terms of four qualitative attributes. Researchers have selected different numbers of geons to describe objects. The larger the number of geons, the greater the descriptive power. Obviously, complexity increases with a larger number. Dickinson et al. [DPR92] have defined ten qualitative primitives as part models. Raja and Jain [RJ94] have employed 12 primitives. Ferrie and Levine [FL88] used ellipsoids and cylinders as descriptions of object parts. Since only two shape types were used for the part models, this approach produced only very limited object descriptions. Shapiro et al. [SMHM84] have proposed sticks, plates and blobs as 3D part models. Sticks are long, thin parts that have only one significant dimension. Plates are flat, they have two significant dimensions. Blobs are parts that have all three significant dimensions.

Qualitative models may distinguish between two objects components shapes but fail to convey quantitative information about their relative sizes. In our case, this is a problem since we showed that information about objects parts shapes and sizes are crucial to the graspable part selection. For example, consider the situation where the qualitative shape information is the same for two objects. In this case, quantitative information such as the relative size of the object parts or the specific curvature of the part axis have a great influence on the choice of the object grasping part.

[Bie87] I. Biederman. Recognition-by-components: A theory of human image understanding. *Psychological Review*, 94:115–147, 1987.

[DPR92] S.J. Dickson, A.P. Pentland, and A. Resenfeld. From volumes to views: An approach to 3d object recognition. *CVGIP: Image Understanding*, 55(2):130–154, 1992.

[RJ94] N.S. Raja and A.K. Jain. Obtaining generic parts from range data using a multi-view representation. *CVGIP: Image Understanding*, 60(1):44–64, 1994.

[FL88] F.P. Ferrie and M.D. Levine. Deriving coarse 3d models of objectsn. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 345–353, 1988.

[SMHM84] L.G. Shapiro, J.D. Moriarty, R.M. Haralick, and P.G. Mulgaonkar. Matching three-dimensional objects using a relational paradigm. *Pattern Recognition*, 17(4):385–405, 1984.

Quantitative models: In contrast to qualitative models, quantitative models provide parameters to describe shapes and sizes. Binford [Bin71] proposed generalized cylinders as object part models. Generalized cylinders can be modelled by defining a parametric curve that acts as the axis of the cylinder and then defining a cross section that is swept along the axis. However, generalized cylinders are not unique. There exists a large number of descriptions corresponding to one volumetric shape, depending on how the axis and cross sections are selected. Hyperquadrics [KHGB95] and fourth order polynomials [KCS94] employ parametric equations and thus can be used to describe a large number of volumetric shapes. However, the parameters obtained are not intuitively related to the object shapes. The number of degrees of freedom associated with these two models induces their non uniqueness in describing an object. Pentland [Pen86] has proposed the use of superquadrics. Superquadrics are also defined parametrically. We use superquadrics for objects part identification, for their ability to describe a large variety of solids with only few parameters.

3.5.2.2 Superquadrics for objects parts representation

Superquadrics are a family of geometric solids, which can be interpreted as a generalization of basic quadric surfaces and solids. They have been considered as volumetric primitives for shape representation in computer graphics [Bar81] and computer vision [Pen86]. Indeed, from one hand, they are convenient part-level models that can further be deformed and glued together to model articulated objects. From the other hand, with only few parameters, superquadrics can represent a large variety of standard geometric solids as well as smooth shapes.

Seven superquadrics shapes for objects parts representation: Similarly to Biederman's geons, Wu [WL95] proposed seven geons to describe objects. The choice of the geons shapes was primarily motivated by the art of sculpture. From sculptors point of view, all sculptures are composed of variations of five basic forms: the cube, the sphere, the cone, the pyramid and the cylinder [Zor60]. Another important belief in the world of sculpture is that each form originates either as a straight line or a curve [Zor60]. By generalizing the five

[Bin71] T.O. Binford. Visual perception by computer. *IEEE Conference on Systems and Control*, 1971.

[KHGB95] S. Kumar, S. Han, D. Goldgof, and K. Bowyer. On recovering hyperquadrics from range data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(11):1079–1083, 1995.

[KCS94] D. Keren, D. Cooper, and J. Subrahmonia. Describing complicated objects by implicit polynomials. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(1):38–52, 1994.

[Pen86] A.P. Pentland. Perceptual organization and the representation of natural form. *Artificial Intelligence*, 28:293–331, 1986.

[Bar81] A.H. Barr. Superquadrics and angle-perserving transformations. *IEEE Comput. Graphics Applicat.*, 1:11–23, 1981.

[WL95] Kenong Wu and Martin D. Levine. Segmenting 3d objects into geons. In *ICIAP*, pages 321–334, 1995.

[Zor60] W. Zorach. Zorach explains sculpture: What it means and how it is made. *Tudor Publishing Company*, 1960.

primitive shapes used in sculpture and adding two curved primitives, Wu et al. [WL95] arrive at the following seven shapes: the ellipsoid, the cylinder, the cuboid, the tapered cylinder, the tapered cuboid, the curved cylinder and the curved cuboid (Fig. 3.15).

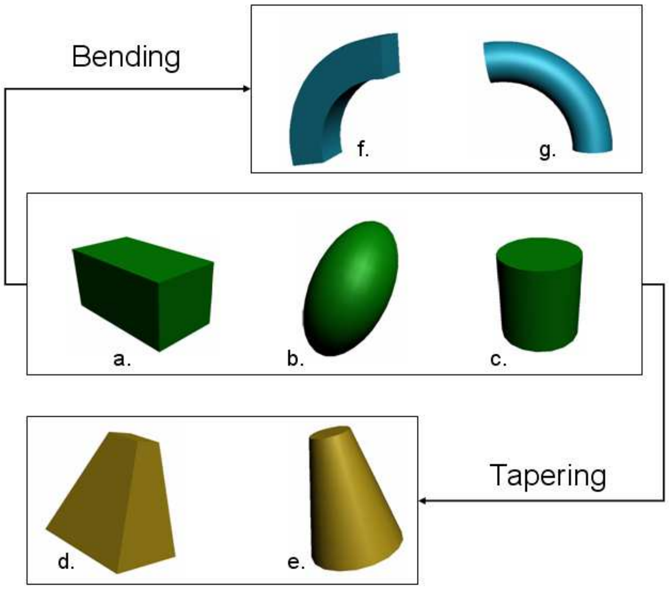


Figure 3.15: Seven superquadrics for objects modeling: a) the cuboid, b) the ellipsoid, c) the cylinder, d) the tapered cuboid, e) the tapered cylinder, f) the bent cuboid and g) the bent cylinder.

Superquadrics formulation: A superquadric surface model is defined by the following equation:

$$f(x,y,z) = \left(\left(\frac{x}{a_1}\right)^{\frac{2}{\epsilon_2}} + \left(\frac{y}{a_2}\right)^{\frac{2}{\epsilon_2}}\right)^{\frac{\epsilon_2}{\epsilon_1}} + \left(\frac{z}{a_3}\right)^{\frac{2}{\epsilon_1}} = 1 \tag{3.5}$$

Where:

- a_1, a_2 and a_3, define the superquadric size;
- ϵ_1 and ϵ_2, determine the shape curvatures that define a smoothly changing family of shapes from rounded to square.

This function determines where a given point $P(x, y, z)$ lies relative to the superquadric surface:

- If $f(x, y, z) = 1$, point P is on the surface of the superquadric;
- If $f(x, y, z) < 1$, the corresponding point lies inside the superquadric;

- If $f(x, y, z) > 1$, the point lies outside the superquadric.

Based on this equation of the superquadric surface solina et al. [SB90] define the following function:

$$F(x, y, z) = f^{\epsilon_1} \tag{3.6}$$

Note that the additional exponent ϵ_1 does not change the superquadric surface itself but is necessary if the function is used for shape recovery with a least squares minimization method. It ensures that, independent of the current value of ϵ_1, points at the superquadric surface have the same value of $F(x)$. Otherwise, when $\epsilon_1 \ll 1$, even very small deviations of a point from the superquadric surface in the z coordinate are greatly amplified. This compact model of superquadrics, defined by only five parameters, can model a large set of building blocks like spheres, cylinders and boxes. When both ϵ_1 and ϵ_2 are 1, the surface vector defines an ellipsoid or, if a_1, a_2, and a_3 are all equal a sphere. When $\epsilon_1 \ll 1$ and $\epsilon_2 = 1$, the superquadric surface is shaped like a cylinder. Boxes are produced when both ϵ_1 and ϵ_2 are $\ll 1$. Modelling capabilities of superquadrics can be enhanced by deforming them in different ways. In order to increase the flexibility of the model (3.5), two deformations are added : tapering and bending [SB90]. Tapering is performed along the z axis and is defined by two parameters k_x and k_y. The bending operation is also applied along the z axis and is defined with the two parameters k and α. k is the curvature parameter and α determines the bending plane. Knowing these two parameters, the bending angle, γ, can be easily computed. Thus, the model (3.5) is modified in order to take into account these deformations. More details on the deformation parameters are provided in (Appendix A). If we take into account these deformations, a superquadric can be modelled by 14 parameters; a_1, a_2, a_3 define the superquadric size; ϵ_1, ϵ_2 are for shape; k_x, k_y for tapering; γ for bending; ϕ, θ, ψ for orientation; and p_x, p_y, p_z for position in space. We will refer to the set of all model parameter values as:

$$\lambda = \{a_1, a_2, a_3 .... a_{15}\} \tag{3.7}$$

Recovery of superquadrics models: Given a set of N 3D surface points, we want to model them with a superquadric. We need to vary the 15 parameters a_j, j = 1, . . . , 15 in (3.7) to get such values for a_j that most of the 3-D points will lay on, or close to the model surface. For this purpose, we use the recovery method explained in [SB90]. Finding the model λ for which the distance from points to the model is minimal is a least-squares

[SB90] F. Solina and R. Bajcsy. Recovery of parametric models from range images: the case of superquadrics with global deformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(2):131–147, 1990.

minimization problem [MNT04]. For each point, the following distance is calculated :

$$d = F - 1 \tag{3.8}$$

This distance is then minimized for the N points:

$$min \sum_{k=1}^{N} d_k^2 \tag{3.9}$$

The minimization is performed with the Levenberg-Marquardt algorithm [MNT04] which consists in a non-linear regression approach.

Discussion: Solina et al [SB90] perform superquadrics recovering using the Levenberg-Marquardt minimization algorithm. A drawback of such an algorithm is that it may converge to a local minimum. To overcome this difficulty, the authors initially estimate the set of model parameters λ. They roughly estimate the 3D data position and orientation by computing the matrix of central moments. They also estimate the size of the initial data with its surrounding box. The model shape is always initialized to a non-deformed ellipsoid. With this initialization, the approximation algorithm performs very well for many objects but we identified the following problems:

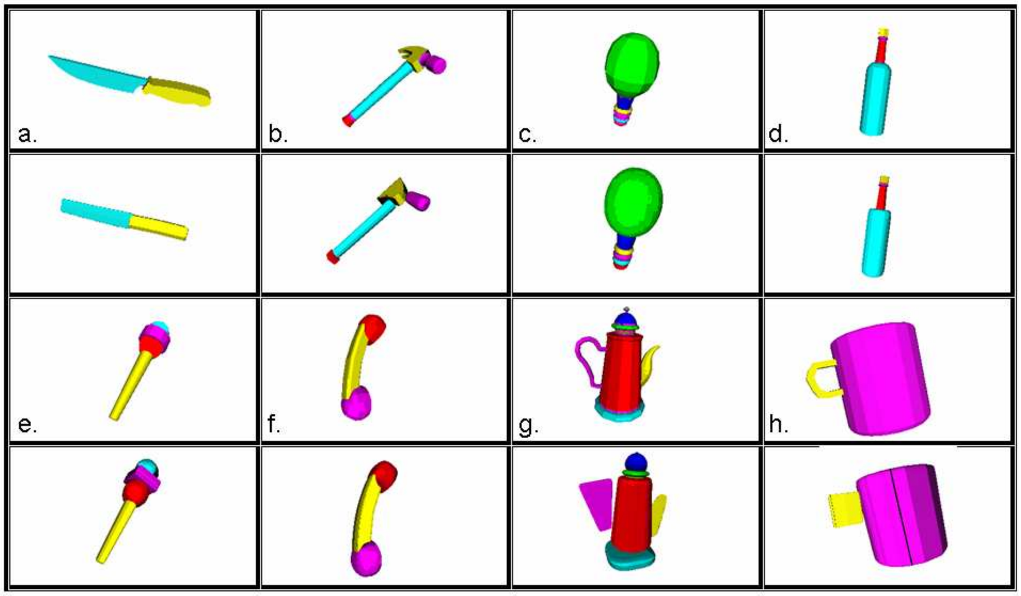


Figure 3.16: Approximation results for some objects models.

- Some roughly rounded objects parts were approximated with squared shaped parts. For example, the magenta and the blue parts respectively for objects (e) and (g) in (Fig. 3.16) are approximated with squared shaped parts.

[MNT04] K. Madsen, H. Nielsen, and O. Tingleff. Methods for non-linear least squares problems. *Technical University of Denmark*, 2004.

- Failure to accurately approximate some strangely shaped curved parts, i.e examples (g) and (h).

The first problem will not have any consequences on our learning algorithm. We remind the reader that the approximation step is needed to convey information about objects sub-parts shapes and sizes to the learning algorithm. The choice of the graspable part of the objects used for the training data is the same whether the corresponding sub-parts are squared or rounded. As for the second problem, we believe that a solution will be to include cavity models. Objects, for which the approximation step does not perform well, weren't taken into account in our experimental tests.

3.6 Learning the Natural Grasping Component

By taking inspiration from the "Recognition By Components" theory, we proposed to describe objects as an assembly of parts and then proceed to the selection of the graspable part in accordance with humans choice. The previous paragraphs detailed an object representation as a set of superquadrics. Thus, the remaining issue is obviously to develop a learning algorithm to select the graspable superquadric. This is difficult since objects are composed of a variable number of parts which are of different shapes and sizes. The question that arise is: What information about the objects is relevant to their grasping? Can this information be reused to grasp novel objects? What could possibly be the training data? Evidently this data should be reused to grasp novel unlearned objects. We already make the observation that many objects with similar components are grasped in the same manner and the relative sizes of object components is relevant for the graspable part selection. Our algorithm is designed to meet these conditions. The following paragraphs will detail object coding, the training data and the learning algorithm used.

3.6.1 Object Coding

We previously showed that the shape and the size of the object constituting parts are pertinent to the choice of its grasping component. Therefore, we are interested in coding these parts characteristics. Objects are represented as an assembly of superquadrics. The latter are completely described by 14 parameters (3.7). But only 8 parameters (e_1, e_2, a_1, a_2, a_3, k_x, k_y and γ) are sufficient to represent the shape and the size of a superquadric. The other 6 parameters encodes the position and orientation of the superquadric. Therefore, a 8xS column vector V, where S is the object part number, represents the whole object. This object representation is invariant to object translation and rotation. For a scale factor invariance, the size parameters of the object components are represented as the ratio of their most important value.

3.6.2 Training Data

The proposed learning algorithm should use object components shapes and sizes in order to select the grasping part. We showed that an object sub-parts assembly is less relevant than their shapes and sizes to the choice of the graspable part. Thus, we can consider multi-part objects grasping as an extension of two-part objects grasping knowledge. This leads to a training data constituted of two-part objects. At this level, two questions arise:

- Which two-part objects choose as training data?
- How can we obtain 3D objects models?

In learning algorithms, a large number of training examples is needed in order to have a good generalization. Collecting real world data is cumbersome. Generating synthetic data is easier and less-time consuming. Therefore, we use synthetic 3D objects models available on Princeton Benchmark [SMKF04] and NTU 3D Model Benchmark [CTSO03] along with labels indicating the grasping component. Since the learning algorithm should perform an analogue of humans choice of the grasping component, different subjects were asked to identify the grasping part of the corresponding objects.

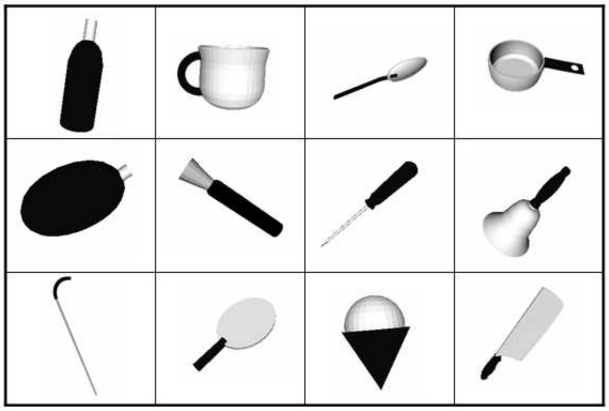


Figure 3.17: Training objects set. The black part indicates the object handle.

As for the choice of the two-parts training objects, supervised learning requires a set of objects that can potentially span the space of two superquadrics assembly. Therefore, the choice of the training objects should effectively sub-sample this space. We mentioned previously that 7 superquadrics are used to model our objects. Thus, the training objects components are chosen to span these 7 superquadrics shapes with different sizes. We use 12 objects for the training set (Fig. 3.17). For more details on spanning the superquadrics shapes and sizes space and the choice of the 12 two-part objects as training data, the reader should refer

[SMKF04] P. Shilane, P. Min, M. Kazhdan, and T. Funkhouser. The princeton shape benchmark. *In Proceedings of Shape Modelling International*, 2004.

[CTSO03] D.Y. Chen, X.P. Tian, Y.T. Shen, and M. Ouhyoung. On visual similarity based 3d model retrieval. *Computer Graphics Forum (EUROGRAPHICS'03)*, 22(3):223–232, 2003.

to Appendix B. Figure (3.18) shows the steps for generating the training data. It shows first the initial object, its decomposition into single parts, the approximation of each part with a superquadric and finally its corresponding grasping part according to humans choice. Additionally, to increase the diversity in our data, once a synthetic model of the object has been created, we vary some properties of the object components such as the size, the bending angle or the tapering parameters without changing the whole appearance of the object. By varying these properties, we generate 72 examples of each object. These examples are divided into training data and testing data.

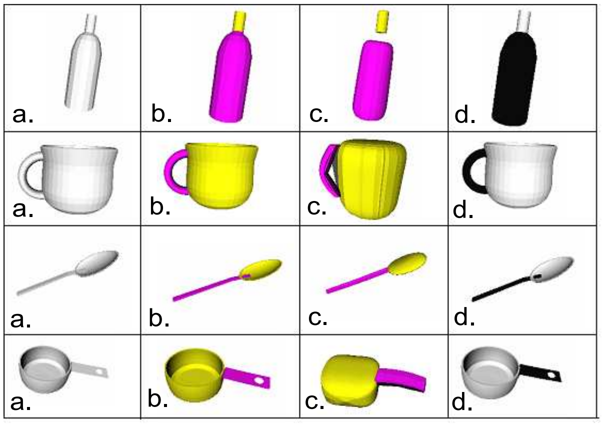


Figure 3.18: Some two-part objects used for generating the training set. (a) shows the initial 3D object. (b) presents its segmentation into single parts. (c) shows the superquadric approximation of each constituting part. (d) shows the natural grasping part.

3.6.3 Learning Algorithm

A multi-layer perceptron, with one hidden layer, is trained with a typical backpropagation learning algorithm [Bis95] in order to select the grasping part of a two-component object. We have shown previously that objects with similar components are grasped in the same manner and that the choice of the graspable part is also influenced by the object components relative sizes. However, this choice is less influenced by the object parts assembly. Thus, our algorithm will use only information about objects parts shapes and sizes in order to identify the graspable one. Eight parameters are sufficient to represent a component shape and size. In the sequel, the first layer has sixteen inputs (training performed on two-parts objects). On the other hand, the output layer represents whether the first or the second component of the object is chosen as grasping part. Thus, the output is a one unit layer. As for the hidden layer, 5 units were chosen empirically for obtaining a score of 99% for the training as well as for the testing data.

For multi-part objects, the decision of the grasping component is taken by considering the object parts two by two. In other words, the algorithm starts by choosing a grasping component between two parts of the object. The chosen part is then compared with another

[Bis95] C.M. Bishop. Neural networks for pattern recognition. *Oxford University Press*, 1995.

component and so on until finding the handle of the multi-part object. Different experiments were conducted to test the ability of the learning algorithm to generalize. These experiments are detailed in chapter 5.

3.7 A Study Relating Object Sub-Part/Task

Recently, a study on demonstrated functional wrench space permits to reinforce our idea of relating the task to an object sub-part. This work was presented in the previous chapter [AC08]. The authors let different users grasp a target object in order to perform a specific task. They compute then a task criterion to evaluate the adaptability of a new grasp to the corresponding task. They tested their algorithm for two different tasks: grasping a ”T” shaped object by its vertical handle and grasping an airplane by its fuselage. New grasps were generated afterwards. Grasps of high quality were those generated on the appropriate objects parts (the vertical handle and the fuselage). Grasps generated on the others objects parts showed a low task criterion. Thus, objects sub-parts and tasks are closely related.

3.8 Conclusion

We took the challenge to associate a grasp to an unknown object/task. Our approach is based on the following observation: commonly used objects are equipped with a part to facilitate their grasp. But, what is the graspable part of an object? We define the handle or the natural grasping component of an object as the part chosen by humans to pick this object with. When humans reach out to grasp an object, It is generally in the aim of accomplishing a task. Thus, the grasp they choose is related to the object task. Consequently, by learning humans choice of the grasping component, the algorithm learns the grasp corresponding to the object/task. That is how we fulfill the challenge. Thus, generating contact points on the appropriate object sub-part is sufficient for ensuring task compatibility. This is the aim of the next chapter.

[AC08] J. Aleotti and S. Caselli. Programming task-oriented grasps by demonstration in virtual reality. *In Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, WS on Grasp and Task Learning by Imitation*, 2008.

Chapter 4

From the Grasping Component to the Grasping Points

> ”La difficulté n’est pas de comprendre les idées nouvelles, mais d’échapper aux idées anciennes.”
>
> *John Maynard Keyne*

At this point, we are able to identify, for an unknown object, its Natural Grasping Component (NGC). The ultimate goal of a grasping strategy is to ensure stability and task compatibility when grasping novel objects. The latter property is obtained by learning the NGC of the object. The former is the issue of this chapter. It addresses calculating fingers positions on the selected graspable part that ensure grasp stability.

Napier differentiates between two basic grasp types: power grasps and precision grasps [Nap56]. Power grasps induce large areas of contact between the fingers, the palm and the object. Thus, this type is chosen when only grasp stability is required. When grasping dexterity is desired, a precision grasp is chosen. In this case, the object is held with the fingertips and consequently the grasp enables object manipulation. Our aim is to successfully use and manipulate objects. Thus, we are interested in computing precision grasps. In order to ensure object immobility in front of external disturbances, these grasps should satisfy one of the following properties: form-closure or force-closure [Bic95]. With the form-closure property, object motion is prevented by the the contact points positioning. With the force-closure one, the forces applied by the fingers ensure object immobility. When the task requires a robust grasp not relying on friction, e.g. objects fixture, form-closure is used. When grasping and manipulation of objects with a low number of frictional contacts is desired, force-closure is

[Nap56] J. Napier. The prehensile movements of the human hand. *Journal of Bone and Joint Surgery*, 38:B(4):902–913, 1956.

[Bic95] A. Bicchi. On the closure properties of robotic grasping. *International Journal of Robotics Research*, 14:(4):319–334, 1995.

employed. For this purpose, we propose a new sufficient condition for N-finger force-closure grasps computation [KS08b]. We detail first some grasping basics necessary to understanding existing works on force-closure grasps generation. This study is followed by our reformulation of force-closure sufficient condition.

4.1 Grasping Basics

The stability of a grasp is characterized by force-closure property. This section includes basic grasping terminologies necessary for force-closure test elaboration.

Definition 1: A **grasp** is a set of contacts.

Definition 2: A **contact** is a location where a finger meets the object surface. Thus, information about contact type and number, and local object surface are required to determine a grasp (Fig. 4.1).

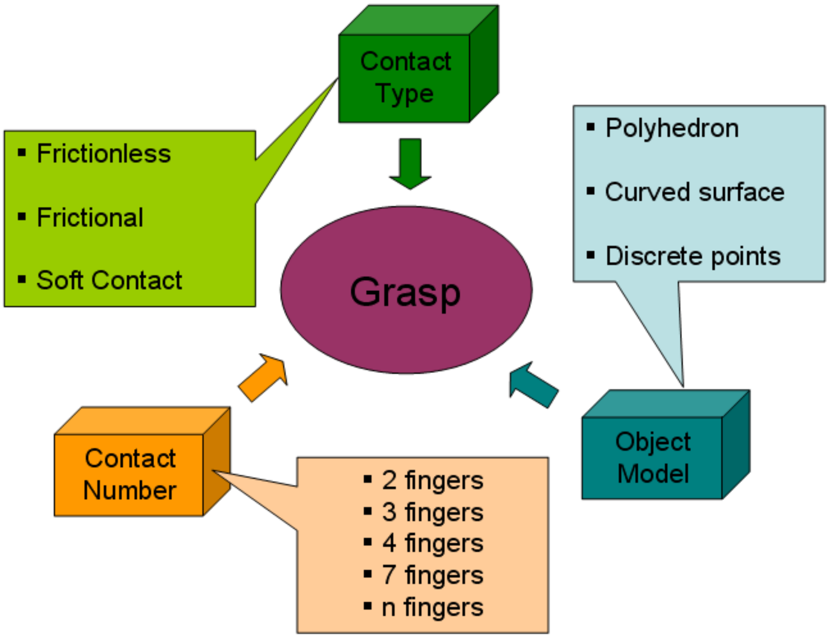


Figure 4.1: Information required to determine a grasp.

Definition 3: A **grasp force** f_i is a force applied by each finger to the object.
In case of a frictionless contact, the grasp force is along the contact normal. Otherwise, the grasp force f_i must satisfy coulomb's law [KKD97], to ensure nonslipping at the contact

[KS08b] S. El Khoury and A. Sahbani. A sufficient condition for computing n-finger force-closure grasps of 3d objects. *IEEE International Conference on Robotics, Automation and Mechatronics*, 2008.

[KKD97] P.R. Kraus, V.I. Kumar, and P. Dupont. Analysis of frictional contact models for dynamic point, (4.1).

$$f_{ix}{}^2 + f_{iy}{}^2 \leq \mu^2 f_{iz}{}^2 \tag{4.1}$$

where $(f_{ix}{}^2, f_{iy}{}^2, f_{iz}{}^2)$ denotes x, y, z components of the grasp force f_i in the object coordinate frame and μ the friction coefficient.

Definition 4: The non linear constraint in (4.1) geometrically defines a cone called **friction cone**.

To simplify the problem, the friction cone is generally linearized by a polyhedral convex cone with m sides.

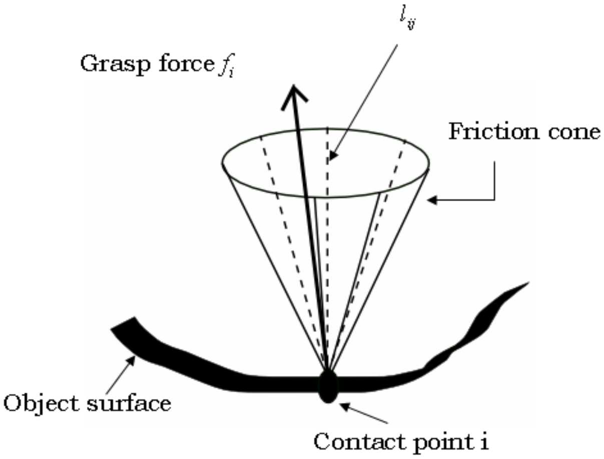


Figure 4.2: The grasp force f_i in a linearized friction cone

Under this approximation, the grasp force can be represented as:

$$f_i = \sum_{i=1}^{m} \lambda_{ij} l_{ij}, \quad \lambda_{ij} \geq 0 \tag{4.2}$$

where l_{ij} represents the j-th edge vector of the polyhedral convex cone. Coefficients λ_{ij} are non negative constants.

Definition 5: A **wrench**, $\underline{w}_i$, is the combination of the force and torque corresponding to the grasp force f_i.

symulation. *In Proceedings of IEEE International Conference on Robotics and Automation,* 1997.

$$\underline{w}_i = \begin{pmatrix} f_i \\ \tau_i \end{pmatrix} = \begin{pmatrix} f_i \\ r_i \times f_i \end{pmatrix} \tag{4.3}$$

where r_i denotes the position vector of the $i-th$ grasp point in the object coordinate frame origined at the center of mass.

Substituting (4.2) into (4.3) provides:

$$\underline{w}_i = \sum_{j=1}^{m} \lambda_{ij} u_{ij} \tag{4.4}$$

where:

$$u_{ij} = \begin{pmatrix} l_{ij} \\ r_i \times l_{ij} \end{pmatrix} \tag{4.5}$$

Vectors u_{ij} are normalized as follows:

$$w_{ij} = \frac{1}{\|l_{ij}\|} u_{ij} \tag{4.6}$$

The term $\|l_{ij}\|$ denotes the L_2 norm of vector l_{ij}. Vectors w_{ij} are called **primitive contact wrenches**. Thus, $N = mn$ is the total number of primitve contact wrenches applied at the object by n fingers.

Definition 6: The **wrench matrix**, W, is a $6 \times nm$ matrix (for 3D objects) where its column vectors are the *primitive contact wrenches.*

$$\mathbf{W} = \begin{pmatrix} l_{11} & \dots & l_{16} & \dots & l_{nm} \\ r_1 \times l_{11} & \dots & r_1 \times l_{16} & \dots & r_n \times l_{nm} \end{pmatrix}$$

Definition 7: According to the definition of Salisbury and Roth [SR82], a grasp is **force-closure** if and only if any external wrench can be balanced by the wrenches at the fingertips.

Proposition 1: Salisbury and Roth has also showed that a necessary and sufficient condition for force-closure is that the primitive contact wrenches resulted by contact forces at the contact points positively span the entire 6-dimensional wrench space. This condition is equivalent to that the origin of the wrench space lies strictly inside the convex hull of the

[SR82] J.K. Salisbury and B. Roth. Kinematic and force analysis of articulated hands. *ASME J. Mech., Transmissions, Automat.,Design*, 105:33–41, 1982.

primitive contact wrenches [MSS87, MLS94, Mon91].

Proof. for a proof, the reader should refer to [SR82]. ■

4.2 Reformulation of the Force-Closure problem

Works on force-closure received a lot of attention during the last two decades. After the pioneering works of Salisbury and Roth [SR82], several force-closure necessary and sufficient conditions were proposed in the literature, but only few concerns 3D objects due to their complicated geometry and high dimension of the grasp space. This section reviews the main related works (Fig. 4.3). Some researchers considered polyhedral 3D objects [Ngu87, PSBM93], others smooth curved surfaces [ZW03] or objects modelled with a set of points [Liu99, LLC03, BSZ08].

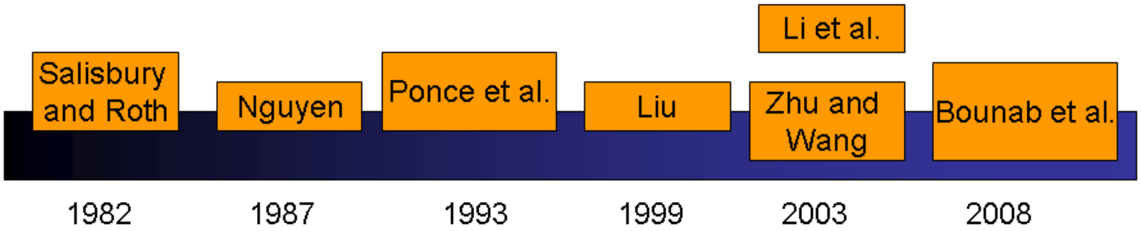


Figure 4.3: Force-Closure grasps conditions for 3D objects.

[MSS87] B. Mishra, J.T. Schwartz, and M. Sharir. On the existence and synthesis of multifinger positive grips. *Algorithmica, Special Issue: Robotics*, 2:541–558, 1987.

[MLS94] R.M. Murray, Z. Li, and S.S. Sastry. A mathematical introduction to robotic manipulation. *Orlando, FL: CRC*, 1994.

[Mon91] D.J. Montana. The condition for contact grasp stability. *In Proceedings of IEEE International Conference on Robotics and Automation*, pages 412–417, 1991.

[Ngu87] V.D. Nguyen. Constructing stable grasps in 3d. *In Proceedings of IEEE International Conference on Robotics and Automation*, pages 234–239, 1987.

[PSBM93] J. Ponce, S. Sullivan, J.D. Boissonnat, and J.P. Merlet. On characterizing and computing three- and four-finger force-closure grasps of polyhedral objects. *In Proceedings of IEEE International Conference on Robotics and Automation*, pages 821–827, 1993.

[ZW03] X. Zhu and J. Wang. Synthesis of force-closure grasps on 3d objects based on the q distance. *IEEE Transactions on Robotics and Automation*, 19:(3), 2003.

[Liu99] Y.H. Liu. Qualitative test and force optimization of 3d frictional form closure grasps using linear programming. *IEEE Transactions on Robotics and Automation*, 15:(1), 1999.

[LLC03] J.W. Li, H. Liu, and H.G. Cai. On computing three-finger force-closure grasps of 2d and 3d objects. *IEEE Transactions on Robotics and Automation*, 19:(1), 2003.

[BSZ08] B. Bounab, D. Sidobre, and A. Zaatri. Central axis approach for computing n-finger force-closure grasps. *In Proceedings of IEEE International Conference on Robotics and Automation*, 2008.

Nugyen [Ngu87] : Nguyen studied force-closure grasps of polyhedral objects and proposed the following necessary and sufficient condition:

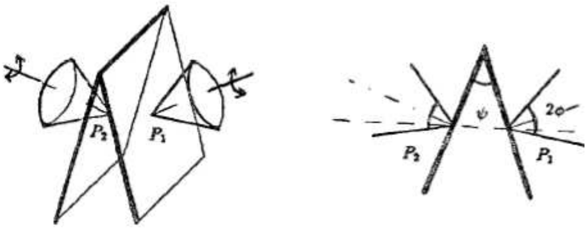


Figure 4.4: Grasp with two soft contact points [Ngu87].

Proposition 2: A grasp with 2 soft-finger contacts is force-closure if and only if the segment P_1P_2, or P_2P_1, joining the two points of contact P_1 and P_2, points strictly into and out of the friction cones respectively at P_1, P_2 (Fig. 4.4).
Another important result proposed by Nguyen is:

Proposition 3: A grasp with at least two distinct soft-finger contact is force-closure if it achieves non-marginal equilibrium.

Definition 8: A n-finger is said to achieve non-marginal equilibrium when there exists a set of forces in the open friction cones at the fingertips such that the sum of the associated wrenches is zero.

Ponce et al. [PSBM93] : Ponce et al. extended the work of Nguyen [Ngu87] to the case of 3 fingers (proposition 4). They also gave a new geometric characterization of force-closure of 3D polyhedral objects with three (propositions 5 and 6) and four (propositions 7 and 8) fingers. Assuming hard-finger contact and coulomb friction, the authors show that:

Proposition 4: In the presence of friction, a sufficient condition for three-dimensional, n-finger force-closure with $n \geq 3$ is non-marginal equilibrium.

Proposition 5: A necessary condition for three points to form a force-closure grasp is that there exists a point in the intersection of the plane formed by the three contact points with the double-sided friction cones at these points (Fig. 4.5).

Proposition 6: A sufficient condition for three points to form a force-closure grasp is that there exists a point in the intersection of the three open internal friction cones with the triangle formed by these contact points (Fig. 4.5).

Proposition 7: A necessary condition for four points to form a force-closure grasp is that there exist four lines in the corresponding double-sided friction cones that intersect in a single point, form two flat pencils having a line in common but lying in different planes, or

[PSBM93] J. Ponce, S. Sullivan, J.D. Boissonnat, and J.P. Merlet. On characterizing and computing three- and four-finger force-closure grasps of polyhedral objects. *In Proceedings of IEEE International Conference on Robotics and Automation*, pages 821–827, 1993.

[Ngu87] V.D. Nguyen. Constructing stable grasps in 3d. *In Proceedings of IEEE International Conference on Robotics and Automation*, pages 234–239, 1987.

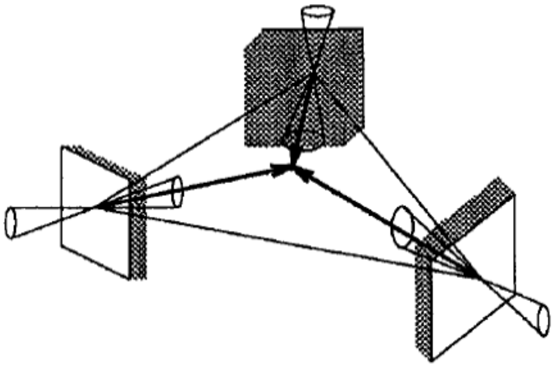


Figure 4.5: Grasping a polyhedron with three frictional fingers [PSBM93].

form a regulus (Fig. 4.6).

Proposition 8: A sufficient condition for four points to form a force-closure grasp is that there exists a point in the intersection of the four open internal friction cones with the tetrahedron formed by these points.

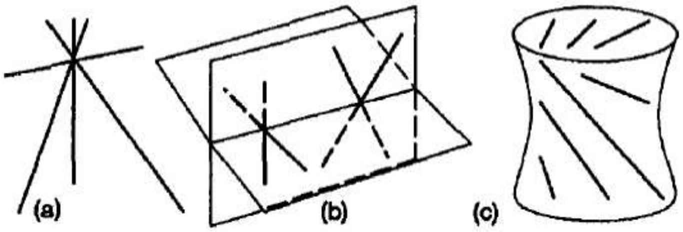


Figure 4.6: Four-finger grasps. (a) four intersecting lines. (b) two flat pencils of lines having a line in common. (c) a regulus [PSBM93].

Zhu and Wang [ZW03] : Zhu and Wang proposed an algorithm to synthesize grasps for any 3D object with smooth curved surfaces and with any number of contacts. The algorithm constitutes a numerical test for the force-closure property based on the concept of the Q distance (or Q norm). The Q norm ($||.||_Q$), or gauge function, is a grasp quality measure defined for a convex compact set $Q \subset R^m$ which contains the origin of the reference system in R^m:

$$g_Q(a \in R^m) = \inf_{a \in \gamma Q, \gamma > 0} \gamma$$

The Q distance is the minimum scale factor required for the set Q to contain a given point a, i.e. it quantifies the maximum wrench that can be resisted in a predefined set of directions,

[ZW03] X. Zhu and J. Wang. Synthesis of force-closure grasps on 3d objects based on the q distance. *IEEE Transactions on Robotics and Automation*, 19:(3), 2003.

given by the set Q. Let $p \in R^m$ and $A \subset R^m$ be a point and a convex polyhedron, respectively; the Q^+ distance is defined as:

$$d_Q^+(p, A) = \min_{a \in A} ||a - p||_Q$$

This distance can be interpreted as the radius of the smallest $||.||_Q$ sphere in contact with $A - \{p\}$.

Proposition 11: A grasp with primitive contact wrench matrix G achieves force-closure only if $d_Q^+(O, CH(G)) = 0$.
However, this is a necessary but not sufficient force-closure condition. This makes necessary the definition of the distance d_Q^- as:

$$d_Q^-(p, A) = -\min_{\partial a \in A} ||a - p||_Q$$

Where ∂a denotes the boundary set of A. The distance $d_Q^-(p, A)$ has a geometric interpretation: it is the radius of the largest sphere contained in $A - \{p\}$. Assuming that the grasped object is piecewise smooth, and each finger is placed at one of the smooth surfaces, it can be stated that the primitive contact wrenches depend on the grasp configuration u, i.e. $G = (g_{11}(u), ..., g_{nm}(u))$. Therefore, if the definition of d_Q^- is used with $p = 0$, $A = CH(G(u))$, the following sufficient and necessary condition is obtained:

Proposition 12: a grasp is force-closure if and only if $d_Q^-(u) < 0$.

► Testing a grasp for force-closure with the previously presented approaches yields solving a system of linear inequalities [Ngu87, PSBM93] or computing the Q distance [ZW03] which are both stated to be computationally complex. The latter requires smooth objects which is not the case of many real objects. The other approaches are only adapted to polyhedral objects composed of a limited number of planar faces, where searching for force-closure grasps induce all faces combinations. Works dealing with more general objects representation are next detailed.

Liu [Liu99] : Liu has developed a qualitative test algorithm of n-finger force-closure grasp. By introducing the polyhedral approximation of the non-linear friction cone, he has reformulated the force-closure condition given by [SR82] as a ray-shooting problem and he solved it by a linear programming method. In detail, Liu first find an interior point P of the convex hull $H(W)$ and then detect the intersection Q of the convex hull with the ray from the interior point P to the origin O of the wrench space. If the distance between points P and O is strictly smaller than that between points P and Q, the origin is an interior point of

[Liu99] Y.H. Liu. Qualitative test and force optimization of 3d frictional form closure grasps using linear programming. *IEEE Transactions on Robotics and Automation*, 15:(1), 1999.

[SR82] J.K. Salisbury and B. Roth. Kinematic and force analysis of articulated hands. *ASME J. Mech., Transmissions, Automat.,Design*, 105:33–41, 1982.

the convex hull so that the grasp is a force-closure grasp. Otherwise, the origin is not inside the convex hull so that the grasp does not form a force-closure (Fig. 4.7).

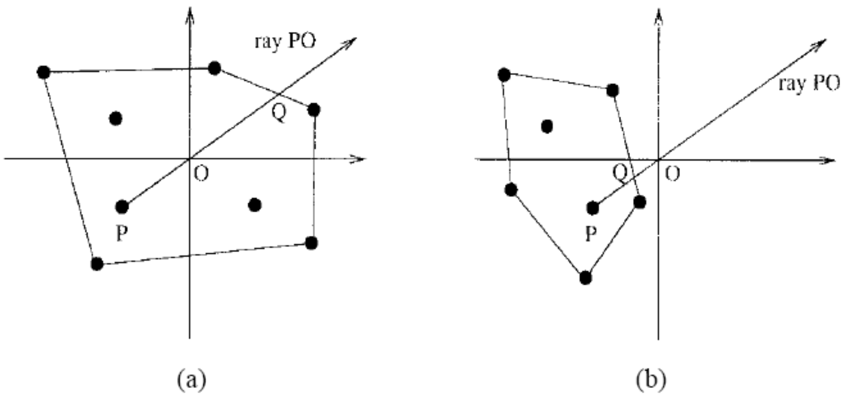


Figure 4.7: A force-closure grasp (a) and a non force-closure grasp (b) [Liu99].

Li et al. [LLC03] : Li et al. extended the work of Ponce et al. [PF95] for polygonal objects and put forward necessary and sufficient conditions for 3-fingered force-closure test of 2D objects. They decompose the problem of three-finger 3D grasps to that in the contact plane (as a planar grasp problem) and that in the direction perpendicular to the plane. Thus the 3D problem is dealt with as a reduced 2D problem. Since three-finger grasp that achieves non-marginal equilibrium also achieves force closure, the authors propose that (Fig. 4.8):

Proposition 9: A necessary and sufficient condition for the existence of three nonzero contact forces, not all of them being parallel, which achieve equilibrium for 2D objects is that there exist three forces in the friction cones at contact points which positively span the plane and whose lines of action intersect at some point.

Proposition 10: A three-finger 3D grasp achieves force closure if and only if: 1) there exist contact plane S and contact unit vectors n_{11}, n_{12}, n_{21}, n_{22}, n_{31} and n_{32} that are the intersection of the three friction cones with S; 2) the contact unit vectors construct a 2D force-closure grasp in S.

Bounab et al. [BSZ08] : Very recently, Bounab et al. developed a new necessary and sufficient condition for n-finger grasps to achieve force-closure. Based on Poinsot's theorem: "Every collection of wrenches applied to a rigid body is equivalent to a force applied along a

[LLC03] J.W. Li, H. Liu, and H.G. Cai. On computing three-finger force-closure grasps of 2d and 3d objects. *IEEE Transactions on Robotics and Automation*, 19:(1), 2003.

[PF95] J. Ponce and B. Faverjon. On computing three finger force closure grasp of polygonal objects. *IEEE Transactions on Robotics and Automation*, 11:(6):868881, 1995.

[BSZ08] B. Bounab, D. Sidobre, and A. Zaatri. Central axis approach for computing n-finger force-closure grasps. *In Proceedings of IEEE International Conference on Robotics and Automation*, 2008.

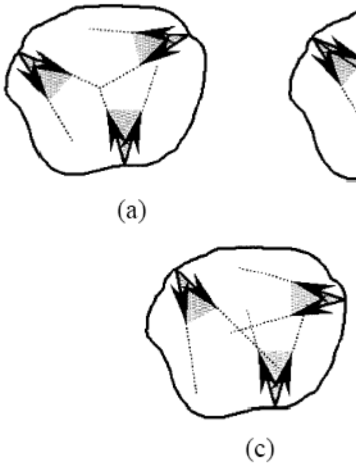


Figure 4.8: Three-finger grasps. (a) Equilibrium but not force-closure grasp. (b) Non-marginal equilibrium and thus, force-closure grasp. (c) Not equilibrium grasps [LLC03].

fixed axis and a torque around the same axis", the authors proposed:

Proposition 13: A 3D (resp. 2D) n-finger grasp is force-closure if and only if w.r.t one arbitrary contact point (e.g c_1):
1) The torque applied by the n fingers positively span R^3 (resp. R^2) at c_1.
2) All grasp wrench central axes at c_1 positively span R^3 (resp. R^2).

► The paper of [BSZ08] does not include any computation time. But since the approach requires solving a system of linear inequalities, we can say that its computation time is similar to the classic convex-hull method. The ray-shooting problem is a typical problem in computational geometry, which is dual to linear programming because a convex hull is dual to a convex polytope [Liu99]. The author stated that the test has a real time efficiency. Similarly, the force-closure test in [LLC03] is adapted for real time applications. However, finding good force-closure grasps with these approaches induce a large computation time.

4.3 A Sufficient Condition for 3D Force-Closure Grasps

We propose a sufficient but not necessary method to compute force-closure grasps of 3D objects. Our approach works with general objects and with any number n of contacts ($n \geq 4$).

4.3.1 Motivation

Generating good force-closure grasps with the previously detailed force-closure necessary and sufficient conditions require considerable computation time. In order to find such grasps, they perform an exhaustive search for the best n-finger force-closure grasp of an object modelled by N points which would take time in the order of $O(N^n)$. Thus, heuristic approaches were proposed to improve performance [BFH03, FH97, NS07]. These approaches are detailed in Chapter 2. They generate many grasp candidates by selecting contacts on the object surface. Then, these grasps are filtered with a necessary but not sufficient force-closure tests. The grasps that pass the filter may or may not be force-closure. In other words the filter reports false positive but not false negative force-closure grasps. The selected grasps are tested afterwards for force-closure. Another way to improve performance, proposed in the literature, is to use a simplified version of the object's geometry consisting only of shape primitives such as spheres, cylinders, cones and boxes. Then, for each shape, define a set of grasping strategies [MKAC03]. This also reduces the number of grasps tested for force-closure.

Our work is an hybrid solution to the force-closure grasp synthesis. The number of grasps tested for force-closure is reduced since we do not consider the object as a whole but we are interested in generating force-closure grasps only on the object natural grasping component. We also propose a new sufficient but not necessary force-closure test. Thus, grasps that pass the filter ensure necessarily force-closure. Our heuristic is original in the sense that it permits simultaneously fast computation and good quality force-closure grasps generation. We believe that the quality of a n-finger grasp depends on the quality of the first $n-1$ fingers locations as well as the placement of the nth finger. Thus, instead of performing an exhaustive search for n-finger force-closure grasps, we introduce a criterion for the generation of these $n-1$ fingers. The next paragraph details the problem statement and justifies the choices we made for objects modelling and contacts type and number.

4.3.2 Problem Statement

Our objective is to find as fast as possible force-closure grasps. In order to determine a grasp, information about contact type and number, and local object surface are required.

An important aspect of the proposed approach is that it is adapted to complex 3D objects. Most works in the literature assume some geometrical model of the objects being grasped [Ngu87, PSBM93, ZW03]. This allows an analytical formulation for characterizing grasps. For example, algorithms assuming that an object should be modelled with a polyhedron may work acceptably when the polyhedron faces number is low (objects presented are seldom composed of no more than 20 faces [BFH03]). Many real world objects are not polyhedral and thus are modelled with a large number of faces. This increases the complexity

[BFH03] Ch. Borst, M. Fischer, and G. Hirzinger. Grasping the dice by dicing the grasp. *In Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2003.

[FH97] M. Fischer and G. Hirzinger. Fast planning of precision grasps for 3d objects. *In Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, page 120126, 1997.

[NS07] N. Niparnan and A. Sudsang. Positive span of force and torque components of four-fingered three-dimensional force-closure grasps. *In Proceedings of IEEE International Conference on Robotics and Automation*, 2007.

[MKAC03] A.T. Miller, S. Knoop, P.K. Allen, and H.I. Christensen. Automatic grasp planning using shape primitives. *In Proceedings of IEEE International Conference on Robotics and Automation*, 2003.

of such algorithms. Our approach assumes no explicit model of the object being grasped. Objects are modelled with a set of points together with their corresponding normals.

As for contact type, one should take into account wether a finger is hard or soft. A hard finger implies that a contact is modelled as a point while a soft finger allows a face contact. Since objects are modelled as a set of points, hard fingers are considered.

Concerning the contacts number, works on grasping have studied the bound required for satisfying grasping properties. This number depends on the type of contact between the fingertips and the object. The earliest contribution in this field can be traced to Reuleaux [Reu63] who showed that in the frictionless case, a minimum of four frictionless contacts are required to achieve force-closure of a planar object. When considering 3D objects, Somoff [Som00] and much later Lakshminarayana [Lak78] showed that seven frictionless point contacts are necessary for force-closure. Mishra, Schwartz and Sharir [MSS87] have shown that six (resp. tweleve) fingers are always sufficient for ensuring force-closure of 2D (resp. 3D) objects without rotational symmetries. Markenscoff et al. [MP89] proved that four contact points and seven contact points are sufficient for force-closure of respectively 2D and 3D non-rotationally objects. An object with rotational symmetry does not have a force-closure grasp with frictionless contacts. They have also shown that in presence of friction, three fingers are sufficient in the 2D case and four fingers are sufficient in the 3D case for any object. These bounds were lowered by one contact each by Mirtich and Canny [MC94] who assumed rounded finger tips to provide continuity to the contact normals around the boundary of the object. For grasping and manipulation of objects, a low number of contacts is required thus we assume frictional contacts. Since we want an approach that works with any object geometry even ones with rotational symmetry, we are interested in generating at least 4-finger force-closure grasps. Consequently, our approach can be stated as follows (Fig. 4.9):

Given a set of N points along with their normals, we have to compute, as fast as possible, n-finger ($n \geq 4$) force-closure grasps assuming hard-finger frictional contacts.

4.3.3 Preliminaries

This section presents definitions, theorems and notations necessary for our force-closure test elaboration. First, we remind that Salisbury and Roth [SR82] has showed that a necessary and sufficient condition for force-closure is that the primitive contact wrenches resulted by contact forces at the contact points positively span the entire wrench space.

[SR82] J.K. Salisbury and B. Roth. Kinematic and force analysis of articulated hands. *ASME J. Mech., Transmissions, Automat.,Design*, 105:33–41, 1982.

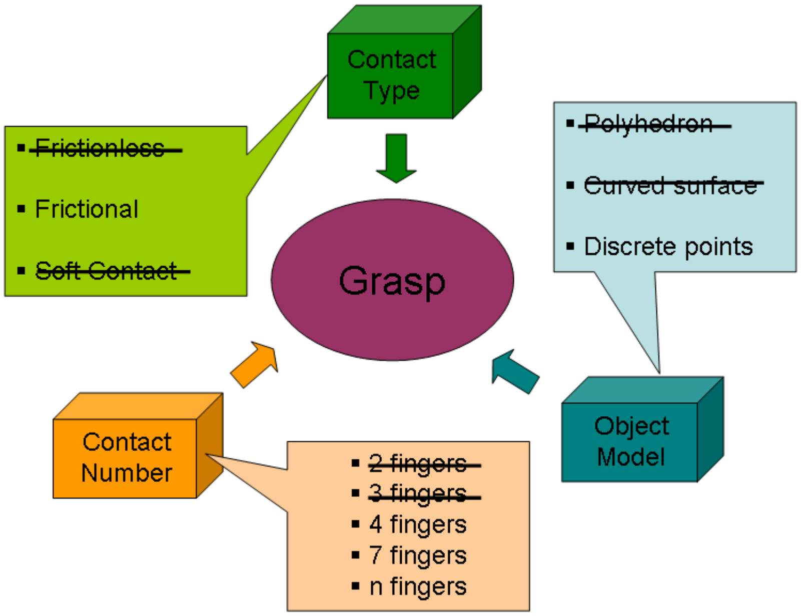


Figure 4.9: Our goal is to compute n-finger ($n \geq 4$) force-closure grasps assuming hard-finger frictional contacts on 3D objects described as a set of discrete points.

▶ **Definition 9:** A set of vectors , $\{v_i\}$ where $i \in I$, positively span a vector space if any vector v in this space can be written as a positive linear combination of v_i, namely :

$$v = \sum_{i \in I} \alpha_i v_i, \quad \alpha_i \geq 0 \tag{4.7}$$

▶ **Proposition 14:** For any n-dimensional Euclidean space E^{n}, $n+1$ vectors are necessary to positively span E^{n}.

Proof. for a proof, the reader should refer to the relative linear algebra results presented by Goldman and Tucker [GT56]. ■

▶ **Lemma 1:** Given a set of $n+1$ vectors, $v_1, v_2, ..., v_{n+1}$, in R^n, such that $v_1, v_2, ..., v_n$ are linearly independent and :

$$v_{n+1} = \sum_{i=1}^{n} \alpha_i v_i, \quad \alpha_i < 0 \tag{4.8}$$

Then each v_i, $i = 1, ..., n+1$, is a unique negative linear combination of the other n vectors [WZG95].

[GT56] A.J. Goldman and A.W. Tucker. Polyhedral convex cones. *Princeton University Press*, 1956.

[WZG95] R. Wagner, Y. Zhuang, and K. Goldberg. Fixturing faceted parts with seven modular struts. *IEEE International Symposium on Assembly and Task Planning*, 1995.

Proof. It is obviously true for v_{n+1}. for any v_i, i=1,...,n, if we solve (C.7) for v_i, we have:

$$v_i = \frac{1}{\alpha_i} v_{n+1} - \sum_{j=1, j \neq i}^{n+1} \frac{\alpha_j}{\alpha_i} v_j \tag{4.9}$$

It is clear by (4.9) that v_i, i=1,...,n, is a unique negative linear combination of the other n vectors. ■

▶ **P**roposition 15: A set of $n+1$ vectors $v_1, v_2, ..., v_{n+1}$ in R^n positively span E^n if and only if v_{n+1} is a unique linear combination of v_i, $i = 1, ..., n$ and all coefficients are strictly negative [WZG95].

Proof. (⇒) Assume that $v_1, v_2, ..., v_{n+1}$ positively span R^n. Then $-v_{n+1}$ can be written as a positive linear combination of v_i:

$$-v_{n+1} = \sum_{i=1}^{n+1} \alpha_i v_i, \quad \alpha_i \geq 0 \tag{4.10}$$

By solving (4.10), they can get the following:

$$v_{n+1} = \sum_{i=1}^{n} \frac{\alpha_i}{(-1 - \alpha_{n+1})} v_i \tag{4.11}$$

Obviously (4.11) shows that v_{n+1} is a negative linear combination of $v_1, v_2, ..., v_n$.

(⇐) Assume that v_{n+1} is a unique linear combination of $v_1, v_2, ..., v_n$ with all coefficients strictly negative, then $v_1, v_2, ..., v_n$ are linearly independent due to uniqueness. Thus, for any vector v in R^n, they can write it as a linear combination of $v_1, v_2, ..., v_{n+1}$:

$$v = \sum_{i=1}^{n+1} \alpha_i v_i \tag{4.12}$$

If all α_i, $i = 1, ..., n+1$, are non-negative, its is done. If not, without loss of generality, assume $\alpha_1 < 0$. By *Lemma(1)*, v_1 can be written as a negative linear combination of $v_2, ..., v_{n+1}$ as follows:

$$v_1 = \sum_{i=2}^{n+1} \beta_i v_i, \quad \beta_i < 0 \tag{4.13}$$

By substituting (4.13) to (4.12), they eliminate the negative term $\alpha_1 v_1$. If there is more than one negative term in (4.12), they simply repeat this elimination for each, until v is a positive linear combination of $v_1, v_2, ..., v_{n+1}$.■

An example of this condition in a 2-dimensional space is shown in (Fig. 4.10) . V_1 and V_2 are non-collinear vectors in R^2, thus form a basis of this space. V_3 can be written as a

[WZG95] R. Wagner, Y. Zhuang, and K. Goldberg. Fixturing faceted parts with seven modular struts. *IEEE International Symposium on Assembly and Task Planning*, 1995.

unique negative linear combination of V_1 and V_2. Therefore, these three vectors positively span E^2.

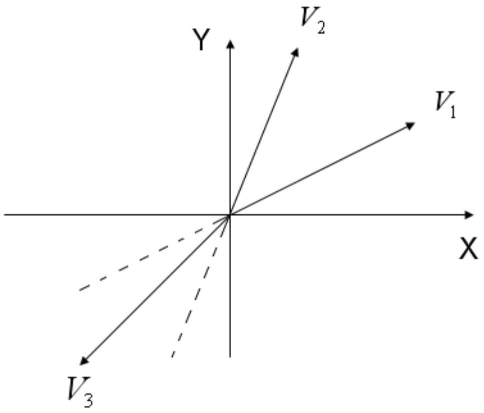


Figure 4.10: Three vectors V_1, V_2 and V_3 that positively span E^2

► All the previous propositions show that in order to ensure force-closure or determine grasp wrenches that positively span the entire 6-dimensional wrench space, one needs to find: (1) primitive wrenches that constitute a 6D basis and (2) a primitive wrench that can be expressed as a negative linear combination of that basis. But, in which case wrenches associated to hard contact points may form a basis of the wrench space? May a representation in the 3D space of 6D wrenches facilitate the problem? Pl*ü*cker coordinates represents 6D wrenches as lines and Grassmann algebra studies the rank of such lines. We use these two studies to prove that wrenches, associated to any three non-aligned contact points of 3D objects, form a basis of the $6D$ wrench space (*proposition* 19). In the following, we present briefly Pl*ü*cker coordinates and the results we use from Grassmann algebra.

Pl*ü*cker coordinates: Let L be a line in the $3D$ space. Let u be the unit line direction and P a point chosen on L. The direction vector along with its cross product with P are known as Pl*ü*cker coordinates and are denoted by $(u; P \times u)$. These 6 coordinates represent L in $3D$ space [VY10, Cra73]. Consequently a primitive contact wrench, defined as $w_i = (f_i; r_i \times f_i)$ can also be seen as a representation of the line of action L_{fi} of the force f_i applied at the point r_i. The 6 coordinates $(w_{i1}, w_{i2}, ..., w_{i6})$ of w_i are called the Pl*ü*cker coordinates of the line of action of f_i.

[VY10] O. Veblen and J.W. Young. Projective geometry. *the Athenaeum press*, 1910.

[Cra73] H. Crapo. A combinatorial perspective on algebraic geometry. *Colloquio Int. sulle Teorie Combinatorie*, 1973.

The Pl*ü*cker coordinates are homogenous coordinates for a projective space of dimension 5, P^5: the wrenches w_i and λw_i, with $\lambda \neq 0$ both represent the same line L_{fi}. Then every line L_{fi} in the $3D$ space corresponds exactly to one point in P^5. The set of lines form a quadric, called the Grassmannian, defined by $w_1 w_4 + w_2 w_5 + w_3 w_6 = 0$ in this projective space. At this point, we have defined a one-to-one relation between the set of lines in the $3D$ space and points in P^5. The rank of this mapping is 6.

Grassmann algebra : Grassmann studied manifold of lines which rank ranges varies from 0 to 6. The purpose of this study was to find geometric characterization of each variety. We are going to use two main results of this study. For a proof of these results, the reader should refer to [Dan84].

▶ **P**roposition 16: All lines through one point are of rank 3.

▶ **P**roposition 17: When all lines meet one special line, they are of rank 5.

4.3.4 A new sufficient condition for n-finger force-closure grasps

At this point, we showed that a $6D$ contact wrench can be represented by the line of action of its corresponding force. We use this mapping to prove that wrenches associated to three non-aligned contact points are of rank 6. This result induces the formulation of a sufficient condition for n-finger ($n \geq 4$) force-closure grasps.

▶ **P**roposition 18: Wrenches associated to 3 aligned contact points are at most of rank 5.

Proof. A $6D$ contact wrench can be represented by the line of action of its corresponding force. The lines of action of forces applied at a contact point pass through that point. Thus wrenches associated to 3 aligned contact points meet one line, the one joining the 3 contact points. Consequently, from *proposition 17*, these wrenches are at most of rank 5. ■

▶ **P**roposition 19: The 6 lines on the sides of a tetrahedron are independent, and thus form a basis of R^6, (Fig. 4.11).

Proof. To deal with lines in $3D$-space, we need a 4-dimensional linear space. For a basis of this space we can either take a point, O and 3 vectors e_1, e_2, e_3 or 4 points (p_0, p_1, p_2, p_3). We can relate these by:

[Dan84] A. Dandurand. The rigidity of compound spatial grid. *Structural topology*, 10, 1984.

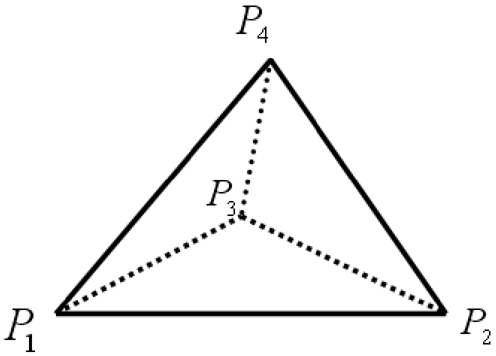


Figure 4.11: The 6 lines on the sides of a tetrahedron are independent.

$$\begin{aligned} p_1 &= O \\ p_2 &= O + e_1 \\ p_3 &= O + e_2 \\ p_4 &= O + e_3 \end{aligned}$$

Any point can be written as a linear combination of these 4 points, for example:

$$\begin{aligned} P_a &= a_1p_1 + a_2p_2 + a_3p_3 + a_4p_4 \\ P_b &= b_1p_1 + b_2p_2 + b_3p_3 + b_4p_4 \end{aligned}$$

where the a_i and b_i are scalars and the sums of the a_i and b_i are unity.

Lines are represented in Grassmannian terms by exterior products of points. Hence from these 4 independent basis points we can construct 6 independent lines which intersect to form a tetrahedron :

$$\begin{aligned} L_1 &= p_1 \wedge p_2 \\ L_2 &= p_1 \wedge p_3 \\ L_3 &= p_1 \wedge p_4 \\ L_4 &= p_2 \wedge p_3 \\ L_5 &= p_2 \wedge p_4 \\ L_6 &= p_3 \wedge p_4 \end{aligned}$$

Any line L is now able to be represented as a linear combination of these 6 basis lines. We can explicitly display this by multiplying out and simplifying the exterior product of two points P_a and P_b on the chosen line:

$$\begin{aligned} L &= P_a \wedge P_b \\ &= (a_1p_1 + a_2p_2 + a_3p_3 + a_4p_4) \wedge (b_1p_1 + b_2p_2 + b_3p_3 + b_4p_4) \quad \blacksquare \end{aligned}$$

▶ **P**roposition 20: Wrenches associated to 3 non-aligned contact points are of rank 6.

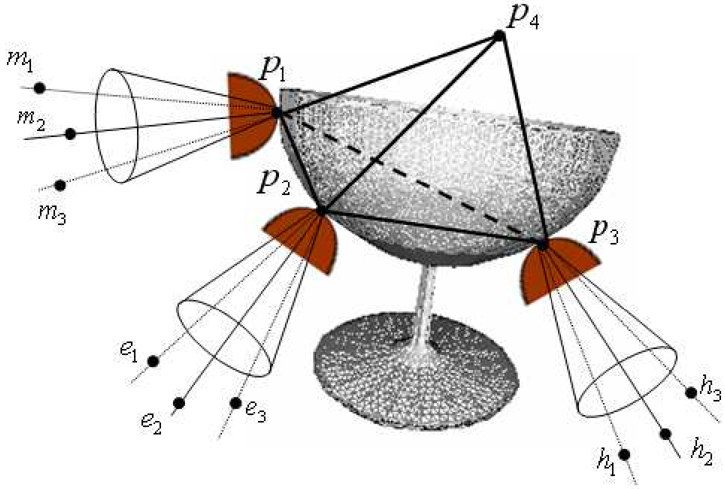


Figure 4.12: The wrenches of rank 3 associated to the frictional contact points $p_1,\ p_2$ and p_3.

Proof. Let $p_1,\ p_2$ *and* p_3 be 3 non-aligned contact points. Consider the friction cone associated to p_1, called CP_1 (Fig. 4.12). Let $\{m_1,\ m_2,\ m_3\}$ be three points chosen on any 3 non-coplanar lines of this cone. The lines $\{l_1 = p_1 \wedge m_1,\ l_2 = p_1 \wedge m_2,\ l_3 = p_1 \wedge m_3\}$ are of rank 3, (from *proposition 16*). Thus any line that passes through p_1 can be expressed as a linear combination of these 3 lines. Similarly, $\{e_1,\ e_2,\ e_3\}$ and $\{h_1,\ h_2,\ h_3\}$, are associated respectively to the friction cones $CP_2,\ CP_3$ at $p_2,\ p_3$. In the same manner, $\{l_4 = p_2 \wedge e_1,\ l_5 = p_2 \wedge e_2,\ l_6 = p_2 \wedge e_3\}$ and $\{l_7 = p_3 \wedge h_1,\ l_8 = p_3 \wedge h_2,\ l_9 = p_3 \wedge h_3\}$ are either of rank 3. Let p_4 be a point non-coplanar with $p_1,\ p_2,\ p_3$, so these 4 points constitute a tetrahedron.

The lines $(p_1 \wedge p_2)$, $(p_1 \wedge p_3)$ and $(p_1 \wedge p_4)$ can be expressed as a linear combination of $\{p_1 \wedge m_1,\ p_1 \wedge m_2,\ p_1 \wedge m_3\}$ since they all pass through p_1, thus:

$$p_1 \wedge p_2 = \sum_{i=1}^{3} \alpha_i (p_1 \wedge m_i) = \sum_{i=1}^{3} \alpha_i l_i \tag{4.14}$$

$$p_1 \wedge p_3 = \sum_{i=1}^{3} \beta_i (p_1 \wedge m_i) = \sum_{i=1}^{3} \beta_i l_i \tag{4.15}$$

$$p_1 \wedge p_4 = \sum_{i=1}^{3} \gamma_i (p_1 \wedge m_i) = \sum_{i=1}^{3} \gamma_i l_i \tag{4.16}$$

In the same manner, the lines $(p_2 \wedge p_3)$ and $(p_2 \wedge p_4)$ can be expressed as a linear combinations of $\{p_2 \wedge e_1,\ p_2 \wedge e_2,\ p_2 \wedge e_3\}$ since they pass through the contact point p_2. Finally the line $(p_3 \wedge p_4)$ passes through p_3 and thus can be expressed as a linear combination of $\{p_3 \wedge h_1,\ p_3 \wedge h_2,\ p_3 \wedge h_3\}$.

$$p_2 \wedge p_3 = \sum_{i=1}^{3} \beta_i^{'} (p_2 \wedge e_i) = \sum_{i=1}^{3} \beta_i^{'} l_{i+3} \tag{4.17}$$

$$p_2 \wedge p_4 = \sum_{i=1}^{3} \gamma_i^{'} (p_2 \wedge e_i) = \sum_{i=1}^{3} \gamma_i^{'} l_{i+3} \tag{4.18}$$

$$p_3 \wedge p_4 = \sum_{i=1}^{3} \gamma_i^{''} (p_3 \wedge h_i) = \sum_{i=1}^{3} \gamma_i^{''} l_{i+6} \tag{4.19}$$

Where $\alpha_i,\ \beta_i,\ \gamma_i,\ \alpha_i^{'},\ \gamma_i^{'},$ and $\gamma_i^{''} \in R$.

Since the lines of the tetrahedron are of rank 6 (from *proposition* 19), they form a basis of R^6:

$$\forall v \in R^6,\ \exists\ \delta_k,\ k = \{1, ...6\},\ \in R\ such\ as$$

$$\begin{aligned} v = {} & \delta_1(p_1 \wedge p_2) + \delta_2(p_1 \wedge p_3) + \delta_3(p_1 \wedge p_4) \\ & + \delta_4(p_2 \wedge p_3) + \delta_5(p_2 \wedge p_4) + \delta_6(p_3 \wedge p_4) \end{aligned} \tag{4.20}$$

Using equations (4.14) to (4.19) in (4.20) gives:

$$\begin{aligned} v = {} & \delta_1(\sum_{i=1}^{3} \alpha_i l_i) + \delta_2(\sum_{i=1}^{3} \beta_i l_i) + \delta_3(\sum_{i=1}^{3} \gamma_i l_i) \\ & + \delta_4(\sum_{i=1}^{3} \beta_i^{'} l_{i+3}) + \delta_5(\sum_{i=1}^{3} \gamma_i^{'} l_{i+3}) + \delta_6(\sum_{i=1}^{3} \gamma_i^{''} l_{i+6}) \end{aligned} \tag{4.21}$$

By regrouping similar terms, we obtain:

$$v = \sum_{i=1}^{3}(\delta_1\alpha_i + \delta_2\beta_i + \delta_3\gamma_i)l_i + \sum_{i=1}^{3}(\delta_4\beta_i^{'} + \delta_5\gamma_i^{'})l_{i+3} + \sum_{i=1}^{3}\delta_6\gamma_i^{''}l_{i+6} \tag{4.22}$$

The equation 4.22 shows that the lines of the tetrahedron can be expressed as a linear combination of the 9 lines l_i. Thus these 9 lines, associated to the 3 friction cones, are also of rank 6. Consequently, a 6-dimensional basis can be extracted from these 9 lines. We remind the reader that the choice of 3 lines among the m sides of each linearized friction cone is due to the fact that these m lines are of rank 3, (from *proposition* 16*)*.■

▶ **P**roposition 21: Assume that the grasp of $n-1$ non-aligned fingers is not force-closure. Suppose that $\{b_i\}_{i=1..k}$ is the k-dimensional (where $k=6$) basis associated to their corresponding contact wrenches. A sufficient condition for a n-finger force-closure grasp is that there exists a contact wrench γ such that:

$$\bullet \quad \gamma\ is\ inside\ the\ linearized\ friction\ cone$$
$$of\ the\ nth\ finger \tag{4.23}$$
$$\bullet \quad \gamma = \sum_{i=1}^{k}\beta_i b_i,\ \beta_i < 0$$
$$\Rightarrow\ \gamma = B\beta \ \ \Rightarrow\ \beta = B^{-1}\gamma \tag{4.24}$$

where $B = [b_1, b_2, ..., b_k]$ is a $k \times k$ matrix and $\beta = [\beta_1, \beta_2, ..., \beta_k]^T$ is a $k \times 1$ strictly negative vector. Thus, a simple multiplication by B^{-1} permits to test if a contact wrench γ, and consequently the location of the *kth* contact point, ensures a force-closure grasp.

Proof. A necessary and sufficient condition for force-closure is that the primitive contact wrenches resulted by contact forces at the contact points positively span the entire k-dimensional wrench space, (from *proposition* 1). A set of $k+1$ vectors in R^k positively span E^k if and only if the $(k+1)th$ vector is a unique linear combination of the other k vectors and all coefficients are strictly negative, (from *proposition 15*). The $k+1$ vectors $\{\gamma, b_1, b_2, .., b_k\}$ satisfy these conditions and thus positively span R^k. ■

P.S: Note that our reformulation of the force-closure problem occurs from proposition 18 till proposition 21. Is it a simple coincidence? or did the force-closure problem really reach maturity with the proposition 21?

The next sections will detail force-closure grasps synthesis based on proposition 21.

4.4 n-Finger Force-Closure Grasps Synthesis

To achieve force-closure, the grasp matrix should positively span the wrench space (*proposition* 1). For this purpose, we propose a two-steps method: (1) we generate randomly locations of $n-1$ fingers and (2) we test for force-closure with a chosen *nth* finger. We showed that wrenches associated to any three non-aligned contact points of 3D objects form a basis of their corresponding wrench space (*proposition* 19). Thus, we can find 6-dimensional basis from the wrenches associated to these $n-1$ contacts. A position of the *nth* finger is located such that an associated contact wrench can be uniquely expressed as a strictly negative linear combination of such basis (*proposition* 21). This approach permits to compute grasp points on the object for the *nth* finger to achieve force-closure grasp with the other $n-1$ fingers.

► **The proposed algorithm**

This paragraph presents the different steps of the algorithm for computing force-closure grasps of a $3D$ object.

```
Require: - 3D points representing the object
         - Linearized friction cone at each point and corresponding wrenches
Ensure:  - n fingers force-closure grasp
 1: L = Rand_Na_Fingers(n-1)
 2: ntry ← 0
 3: r_basis = Find_Basis (L_wrenches)
 4: vertex = Rand_Finger(1)
 5: FC = Force_Closure_Test(vertex, r_basis)
 6: ntry ← ntry+1
 7: if (!FC) and (ntry ≤ nmax) then
 8:    Go to step 4
 9: else
10:    Go to step 1
11: end if
```

Given a representation of an object along with normal directions and a friction coefficient, wrenches associated to each of its vertices are firstly computed. In order to obtain n-finger force-closure grasps, the function *Rand_Na_Fingers* generates randomly, locations of non-aligned $n-1$ fingers on the object surface. A 6-dimensional basis from the wrenches associated to these $n-1$ contacts are determined by *Find_Basis*. An object vertex is then randomly chosen by *Rand_Finger* and tested for ensuring a n-finger force-closure grasp with *Force_Closure_Test*. A position of the *nth* finger ensures force-closure if an associated contact wrench can be uniquely expressed as a strictly negative linear combination of the 6-dimensional basis. We choose the wrench associated to the normal force on the *nth* contact. If the n-finger grasp ensures force-closure the algorithm ends. Otherwise, a parameter *ntry* permits to quantify the number of the *nth* fingers tested for force-closure with the first generated $n-1$ fingers. If this number is lower than a threshold, *nmax*, another object vertex is tested for force-closure. We generate novel $n-1$ fingers locations in the other case. Notice that $nmax/ntry$ is a good estimation of the randomly $n-1$ generated fingers quality. When *ntry* reaches the threshold, the corresponding quality is equal to 1. It means that no n-finger force-closure grasp was found with the chosen $n-1$ fingers. When *ntry* is lower than the threshold, $nmax/ntry$ is more than unity. In other words, a small value of *ntry* means that a force-closure grasp was easily obtained for the generated $n-1$ fingers. These approaches are detailed and discussed in the last chapter of experimental results.

4.5 Optimal n-Finger Force-Closure Grasps Synthesis

Previously, we developed a sufficient condition for force-closure and presented an algorithm that permits the generation of n-fingers locations satisfying that condition. Our objective is to ensure fast robust force-closure grasps generation. In our case, force-closure grasps fast computation and robustness are strongly linked. In order to understand how the two latter are tied together, one should notice that generating a n-finger good grasp will depend on the generation of the first n-1 fingers. A good choice of their locations will induce on one hand robust grasps and on the other hand more locations for the *nth* finger on the object surface guaranteing force-closure and consequently fast computation. Thus, we need to find a criterion that quantifies a good placement of the $n-1$ first fingers. We present in the following different force-closure quality measures proposed in the literature. We propose then a method that generates simultaneously force-closure and good quality grasps.

4.5.1 Existing Force-Closure Quality Criteria

Different grasps quality measures were proposed in the literature. They are generally associated with fingers locations, task requirements or with the robotic hand kinematics. We focus here on those that measure force-closure performance. In other words, we focus on grasps criteria that measure how well a grasp can stand external disturbances. A review on the quality measures proposed in the grasp literature can be found in [SRC06]. These measures do not assume a priori knowledge of disturbance, they assume that an external wrench is uniformly distributed in every direction. We present in the following some of the most used criteria.

Criterion of the largest ball: The most popular quality metric is the one that determines the magnitude of the largest worst-case disturbance wrench that can be resisted by a

[SRC06] R. Suarez, M. Roa, and J. Cornella. Grasp quality measures. *Technical Report, Universitat Politecnica De Catalunya*, 2006.

grasp of unit strength. This measure has been proposed in several forms [KMY92], but it is best described by Ferrari and Canny [FC92]. The quality of a grasp is equal to the distance from the origin to the closest facet of the convex hull of the grasp wrenches. Thus an optimal grasp is the one that maximizes the radius of the largest ball inscribed in the convex hull. Although this measurement is not invariant to the choice of the wrench space origin (since wrenches contain torques depending on the origin choice), it is used in many works such as [BFH03, MC94, MKAC03].

Volume of the ellipsoid in the wrench space: A measure that is invariant to a change in the torque reference system is proposed in [LS88]. The authors map a sphere of unitary radius in the force domain into an ellipsoid in the wrench space. The quality measure proposed considers the contribution of all the contact forces by measuring the volume of this ellipsoid. This measure must be maximized to obtain the best grasp.

Distance between the centroid of the grasp polygon and the center of mass of the object: The effect of the gravitational forces is reduced when the distance between the center of mass of the object and the center of the contact polygon or polyhedron is reduced. This measure is used in several works [PF95, DLW00].

Isotropy index: This criterion looks for a uniform distribution of the contact forces to the total wrench exerted on an object, hence it is called isotropic. It is defined in [KOYS01] as being the fraction of the minimum to the maximum singular values of the grasp matrix G.

[KMY92] D. Kirkpatrick, B. Mishra, and C. Yap. Quantitative steinitzs theorem with applications to multi-fingered grasping. *Discr. Comput. Geom.*, 7:(3):295–318, 1992.

[FC92] C. Ferrari and J. Canny. Planning optimal grasps. *In Proceedings of IEEE International Conference on Robotics and Automation*, 1992.

[BFH03] Ch. Borst, M. Fischer, and G. Hirzinger. Grasping the dice by dicing the grasp. *In Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2003.

[MC94] B. Mirtich and J. Canny. Easily computable optimum grasps in 2d and 3d. *In Proceedings of IEEE International Conference on Robotics and Automation*, 1:739–747, 1994.

[MKAC03] A.T. Miller, S. Knoop, P.K. Allen, and H.I. Christensen. Automatic grasp planning using shape primitives. *In Proceedings of IEEE International Conference on Robotics and Automation*, 2003.

[LS88] Z. Li and S.S. Sastry. Task-oriented optimal grasping by multifingered robot hands. *IEEE Journal of Robotics and Automation*, 4:(1), 1988.

[PF95] J. Ponce and B. Faverjon. On computing three finger force closure grasp of polygonal objects. *IEEE Transactions on Robotics and Automation*, 11:(6):868881, 1995.

[DLW00] D. Ding, Y. Liu, and S. Wang. Conmputing 3-d optimal form-closure grasps. *In Proceedings of IEEE International Conference on Robotics and Automation*, page 35733578, 2000.

[KOYS01] B. Kim, S. Oh, B. Yi, and I.H. Suh. Optimal grasping based on non-dimensionalized performance indices. *In Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 949–956, 2001.

Q **distance**: We previously defined the d_Q^- distance. This distance is also equivalent to largest ball criterion.

Existing force-closure grasps quality criteria permit to select an optimal grasp among a set of stable grasps. Thus, one should generate first several force-closure grasps, compute their corresponding quality to finally choose a good one. For example, the most used criterion that maximizes the radius of the largest ball inscribed in the convex hull is meaningful when the origin is inside the convex-hull of the contact wrenches, thus when the grasp is a force-closure one. We showed that, in our case, computing a good grasp depends on a good placement of the $n-1$ first fingers that are not in force-closure as much as the placement of the nth finger. Consequently, a quality metric should be associated to the locations of these $n-1$ fingers. Hence, a new criterion is needed.

4.5.2 Quality criterion of the $n-1$ fingers locations

This section details the criterion introduced to measure the quality of the first generated $n-1$ fingers in the case of a 2D object and its extension to 3D objects. Force-closure is obtained by choosing a wrench basis associated to the first $n-1$ fingers and then find the nth finger such that an associated contact wrench can be uniquely expressed as a strictly negative linear combination of that wrench basis. In a 2D case, a wrench basis is represented by three points in the 3D space that constitute with the wrench space origin a tetrahedron. A wrench that ensures force-closure grasp is a wrench that can be uniquely expressed as a strictly negative linear combination of the 3D basis. Thus, the larger the tetrahedron, the more choices we have for such a wrench. In the following, this idea is detailed more formally.

4.5.2.1 2D criterion

We prove in the following that, in a 2D case, the largest ball criterion is obtained when the wrenches associated to the two-finger contact points form a regular tetrahedron.

2D grasps wrenches In 2D, a hard finger in contact with an object at a point x exerts a grasp force f with a corresponding torque $\tau = det(x, f)$. Force and torque are combined into a 3D wrench $w = (f, \tau)$. Thus the wrench space is of rank 3.

2D force-closure grasps quality Let A and B be two contact points on the boundary of a planar object. f_{A1}, f_{A2} and f_{B1}, f_{B2} represent their corresponding friction cones boundaries (Fig. 4.13). The wrenches associated to these grasp forces are represented respectively by the four 3D points, w_{A1}, w_{A2}, w_{B1} and w_{B2}. Note that these wrenches can also be associated to any two forces inside the corresponding friction cones.

► **P**roposition 22: Wrenches associated to any two contact points of 2D objects form a 3D basis.

Proof. Immediate when one notes that if we select any two wrenches, there exists a third one that is not a linear combination of the other two. ■

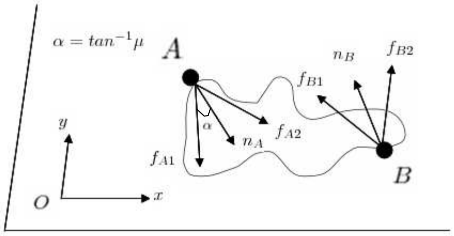


Figure 4.13: f_{A1}, f_{A2} and f_{B1}, f_{B2} represent the $2D$ friction cones boundaries.

Any chosen 3 wrenches from the 4 wrenches associated to the two contact points A and B are of rank 3. Consider for example w_{A1}, w_{A2} and w_{B1}, they form a 3D basis. A sufficient condition for a third finger to ensure 2D force-closure grasp is that its corresponding wrench, w_4, is a strictly negative linear combination of the 3D basis (*proposition 21*). The convex hull of w_{A1}, w_{A2}, w_{B1} and w_4 is a tetrahedron T. Thus one grasp quality corresponds to the largest ball centered at the origin and inscribed in T [FC92]. The best quality is obtained when T is a regular one. Consequently, the best locations of the contact points A and B is obtained when the tetrahedron constituted by normalized w_{A1}, w_{A2}, w_{B1} and the origin O, approximates a regular one.

Tetrahedra quality measure Different tetrahedron quality measures were proposed in the literature especially in the field of mesh optimization. One of the most used quality is $Q = \frac{V}{h^3_{max}}$ [ZBD95], where V is the volume of the tetrahedron and h_{max} is its maximal edge length. Q is maximal when the corresponding tetrahedron is regular. Using this criterion, the quality of the locations of the two contact points A and B is given by:

$$Q(A, B) = \max_{i, i=1..nb} \frac{V_i}{h^3_{i\ max}} \tag{4.25}$$

where nb is the number of tetrahedra constituted by the origin and the wrenches associated to the two contact points. In other words, nb is the number of 3D basis associated to the contact wrenches, (in $2D$, $nb = 4$).

A 2D quality criterion The reader should keep in mind that we are interested in finding a criterion to the locations of the $n-1$ fingers to ensure robust 3D force-closure grasps. Thus,

[FC92] C. Ferrari and J. Canny. Planning optimal grasps. *In Proceedings of IEEE International Conference on Robotics and Automation*, 1992.

[ZBD95] P.D. Zavattieri, G.C. Buscaglia, and E.A. Dari. Finite element mesh optimization in three dimensions. 1995.

in the following, we reformulate (4.25) to be extensible to 3D grasps. As a matter of fact, the volume V of a tetrahedron can be expressed as [RR93]:

$$V = \delta \begin{vmatrix} x_1 & y_1 & z_1 & 1 \\ x_2 & y_2 & z_2 & 1 \\ x_3 & y_3 & z_3 & 1 \\ x_4 & y_4 & z_4 & 1 \end{vmatrix} \tag{4.26}$$

where δ is a constant, it is equal to 1/6. x_j, y_j, z_j are the coordinates of a tetrahedron vertex. Since the origin O is one of the vertices ($i.e$ $x_4 = y_4 = z_4 = 0$), by substituting (4.26) in (4.25) we obtain:

$$\begin{aligned} Q(A,B) &= \max_{i,i=1..nb} \frac{\delta \begin{vmatrix} x_1^i & y_1^i & z_1^i \\ x_2^i & y_2^i & z_2^i \\ x_3^i & y_3^i & z_3^i \end{vmatrix}}{h^3_{i\ max}} \\ &= \max_{i,i=1..nb} \frac{\delta.det(w_1^i, w_2^i, w_3^i)}{h^3_{i\ max}} \end{aligned} \tag{4.27}$$

$Q(A,B)$ is maximal when the corresponding tetrahedron is regular. The volume of a regular tetrahedron is $\sqrt{2}a^3/12$, where a is its edges length. For such a tetrahedron, $h_{max} = a$, thus $Q(A,B)_{max}$ is $\sqrt{2}/12$. A normalized criterion will be:

$$Q(A,B) = \max_{i,i=1..nb} \frac{12.\delta.det(w_1^i, w_2^i, w_3^i)}{\sqrt{2}h^3_{i\ max}} \tag{4.28}$$

In order to show the efficiency of the proposed quality criterion, locations of two contact fingers are randomly generated on a 2D object. Using equation (4.28), the quality of the generated fingers, noted Q_{fg}, is computed. All the 2D object vertices are then tested for force-closure. For all force-closure grasps reported, we calculate the classical grasp quality measure based on the largest ball criterion. The latter is noted Q_{cl}. Figure (4.14) shows the average of Q_{cl} of all force-closure grasps found as a function of Q_{fg} attributed to the fingers, ($mean(Q_{cl}) = f(Q_{fg})$). This figure demonstrates that we are dealing with an increasing function: it means that our criterion and the classical one evolve in the same way. We notice that after a threshold = 0.5, the force-closure quality obtained is above 0.1. In other words, when the tetrahedron constituted with the $3D$ wrench basis and the origin is half-regular, the quality of the force-closure grasps obtained is half-optimal, since the largest ball is of radius $\rho = 0.2041$. This value is computed as follows [RR93]:

$$\rho = 3.\frac{V}{S} = 3.\frac{\sqrt{2}}{12.\sqrt{3}} = 0.2041$$

Where V is the volume of the tetrahedron of unit length and S the sum of its 4 faces surfaces.

[RR93] P. Robert and A. Roux. Influence of the shape of the tetrahedron on the accuracy of the estimate of the current density. *Proceedings of ESA START Conference*, 1993.

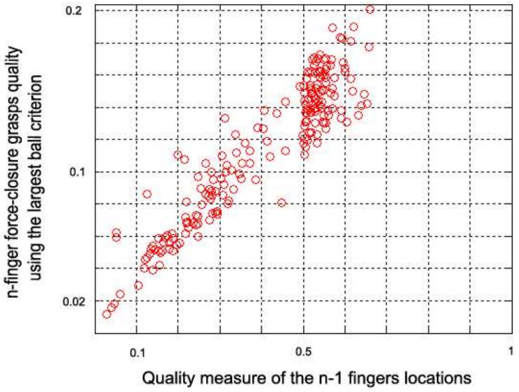


Figure 4.14: The n-finger force-closure grasps quality according to the largest ball criterion as a function of the quality measure attributed to the n-1 fingers locations.

4.5.2.2 A 3D quality criterion

Dealing with $3D$ objects grasps involves $6D$ wrenches. Thus, instead of computing $3D$ tetrahedra volumes, we are conducted to calculate volumes of $6D$ hypertetrahedra. Equation (4.28) introduced in the case of $2D$ grasps could be extended to $3D$ grasps as follows (computation of a 6-volume hypertetrahedron could be viewed as a determinant calculation):

$$Q(C, D, E) = \max_{i, i=1..nb} \frac{\beta.det(w_1^i, w_2^i, w_3^i, w_4^i, w_5^i, w_6^i)}{h_{i\ max}^6} \tag{4.29}$$

Where $\{w_j^i\}_{j=1..6}$ is a $6D$ wrench basis. Note that nb is the number of $6D$ basis chosen among the $(n-1)m$ wrenches, (corresponding to the $n-1$ fingers). This quality measure is used to generate the first $n-1$ contact fingers locations on $3D$ objects and its efficiency is verified in the experimental results.

▶ At this point, we defined a quality criterion for the first generated $n-1$ fingers. The latter is expressed as the volume of their corresponding $6D$ wrench basis. Thus, in this case, a good quality criterion induces a large 6D wrench basis volume yielding more possibilities for placing the nth finger. This, evidently, decreases force-closure grasps computation time and increases their quality. In the following, we present the algorithm taking into account this quality measure for generating robust force-closure grasps:

```
Require: - 3D points representing the object
         - Linearized friction cone at each point and corresponding wrenches
Ensure:  - $n$ fingers force-closure grasp
L = Rand_Na_Fingers(n-1)
ntry ← 0
L_basis = Find_Basis (L_wrenches)
q_L=quality(L_basis)
```

```
5:  if q_L <threshold then
6:      Go to step 1
7:  end if
8:  vertex = Rand_Finger(1)
9:  ntry $\leftarrow$ ntry+1
10: FC = Force_Closure_Test(vertex, r_basis)
11: if (!FC) and (ntry $\leq$ nmax ) then
12:     Go to step 8
13: else
14:     or Go to step 1
15: end if
```

Given a representation of an object along with normal directions and a friction coefficient, wrenches associated to each of its vertices are firstly computed. In order to obtain n-finger force-closure grasps, the function *Rand_Na_Fingers* generates randomly, locations of non-aligned $n-1$ fingers on the object surface. A number *L_basis* of 6-dimensional basis from the wrenches associated to these $n-1$ contacts are determined by *Find_Basis*. The quality of these basis is computed with *quality* function. If the latter is below a threshold, we proceed at the generation of other $n-1$ fingers locations. If the quality of at least one of the *L_basis* is above the specified threshold, an object vertex is then randomly chosen by *Rand_Finger* and tested for ensuring a n-finger force-closure grasp with *Force_Closure_Test*. If the n-finger grasp ensures force-closure the algorithm finishes. Otherwise, *ntry* permits to choose between generating novel $n-1$ fingers locations or testing another object vertex for force-closure with the basis of the same $n-1$ fingers. Note that the choice of the threshold and *nmax* is crucial for the algorithm force-closure grasps computation time. The threshold value cannot be determined analytically. It varies with the object shape and thus is chosen empirically. As for *nmax*, the following chapter discusses in detail its influence on the rapidity of the algorithm.

4.6 Conclusion

This chapter presents an important contribution to existing force-closure grasps computation procedures. It proposes a new sufficient condition for generating n-finger force-closure grasps. Furthermore, contrarily of current works in the literature that generate many grasps and then rank them in order to find the best one, we introduce a quality criterion to the first generated $n-1$ fingers permitting to ensure simultaneously force-closure and good quality grasps according to the classical largest ball criterion. Our quality measure permits also to reduce force-closure grasps computation time. The next chapter details corresponding experimental results.

Chapter 5

Experimental Results

> ”Quoi que vous pensiez ou croyiez pouvoir faire, faites-le. L’action porte en elle la magie, la grâce et le pouvoir.”
>
> *Goethe*

The previous chapters detail our grasping strategy that occurs in two steps. First, an algorithm that predicts grasp as a function of the object’s sub-parts is presented. Starting with a 3D model of the object, a segmentation step decomposes it into single parts. Each part is fitted with a geometric model. A learning step permits then to imitate the human choice of the object’s natural grasping component. In a second time, a new sufficient condition for computing n-finger force-closure grasps on the obtained graspable part is proposed. This chapter aims at testing the proposed approach in a series of experiments. The first experiment quantifies how well learned grasping skills generalize to new objects. Thus, an algorithm is trained to grasp a small set of objects and tested on a much larger set of everyday items. In a second experiment, two aspects of the force-closure sufficient conditions are studied: completeness and rapidity. The efficiency of the proposed method is tested by comparing it to the classical convex-hull one [MKAC03].

5.1 Learning the Natural Grasping Component

We proposed a grasping strategy that describes objects as an assembly of parts and then proceeds to the identification of the handle or the Natural Grasping Component (NGC) in accordance with humans choice. This section aims at testing the ability of our learning algorithm to generalize. For this purpose, different experiments were conducted. We begin

[MKAC03] A.T. Miller, S. Knoop, P.K. Allen, and H.I. Christensen. Automatic grasp planning using shape primitives. *In Proceedings of IEEE International Conference on Robotics and Automation*, 2003.

by validating the learning algorithm model. Two generalization experiments were then performed. The first one tests the algorithm on objects belonging to the same categories as the training data but of different shapes and sizes. The second one considers objects that are completely different from those of the training set. We use for our tests synthetic 3D objects models available on Princeton Benchmark [SMKF04] and NTU 3D Model Benchmark [CTSO03] (Fig. 5.1). We also tested the algorithm on real objects models obtained from a 3D laser scanner and from 3D reconstruction using a vision system. A crucial step of the approach is decomposing objects into parts. Two segmentation algorithms were implemented. Thus, before detailing our learning algorithm results, we compare these two segmentation methods.

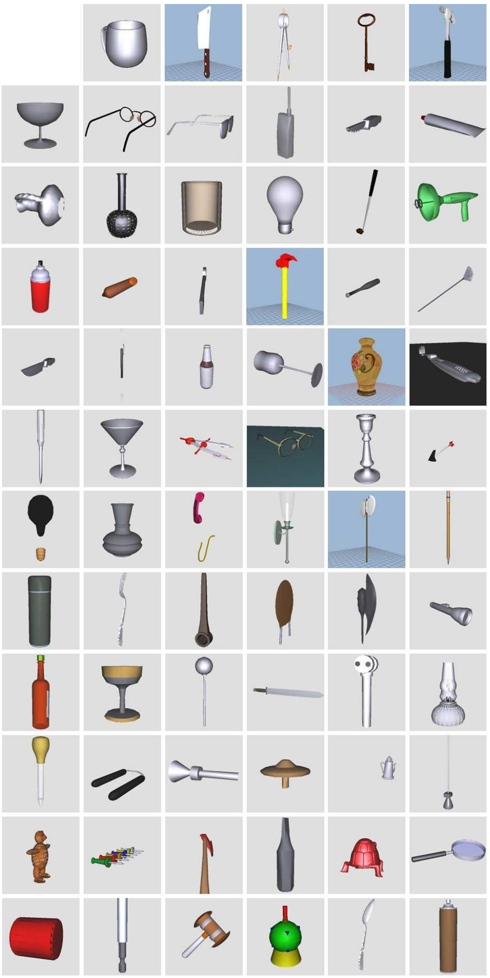


Figure 5.1: Synthetic objects used for testing the algorithm ability to generalize.

[SMKF04] P. Shilane, P. Min, M. Kazhdan, and T. Funkhouser. The princeton shape benchmark. *In Proceedings of Shape Modelling International*, 2004.

[CTSO03] D.Y. Chen, X.P. Tian, Y.T. Shen, and M. Ouhyoung. On visual similarity based 3d model retrieval. *Computer Graphics Forum (EUROGRAPHICS'03)*, 22(3):223–232, 2003.

5.1.1 Objects Segmentation

In order to decompose objects into parts, we implemented two approaches [ZPKG02, CG06]. Zhang's method decomposes objects into parts based on the Gaussian curvature for detecting boundries. In chpater 3, we discussed Zhang's method limitations and showed how Chen's approach can overcome them. We remind the reader of the two limitations identified:

- Problems when dealing with high resolution objects models having concave corners located between different boundaries.

- Failure to determine accurately boundaries between different objects parts when an object model is densely represented with polygonal faces.

Chen's algorithm succeeds to segment high resolution models. In addition to Gaussian curvature, it uses concaveness estimation for detecting boundaries between different sub-parts. Another advantage of the method is that local features are computed using multi-ring neighborhood, contrarily to Zhang's method that uses only one-ring neighborhood. In the following, we perform segmentation on a synthetic 3D model with different resolutions in order to illustrate the previously stated limitations. We finally test the chosen algorithm on real objects obtained from a 3D laser scanner and a vision system.

Limitation due to concave corners: The first example illustrates three spheres union. Their intersection is a concave corner. This area has an elliptic behavior. For a high resolution model, the boundaries between the three spheres will be broken at this region. Figure 5.2 shows Zhang's method segmentation results with different resolutions. The algorithm performs well for low resolution models. It fails to decompose the high resolution one into parts. On the other hand, we tested Chen's method with the latter model. Only one-ring neighborhood was chosen for computing local objects features. Figure 5.3 illustrates the corresponding decomposition. This proves that concaveness estimation is crucial when dealing with concave corners objects.

Limitation due to densely distributed polygons: This experiment aims at studying the ability of the segmentation algorithms to decompose high resolution 3D models. For this purpose, we tested the algorithms on real objects models obtained from a 3D laser scanner. The reflective objects surfaces was treated with a matt spray. A turning table permitted then to acquire 3D objects models. The latter are presented in (Fig. 5.4). Zhang's algorithm fails to decompose these objects into parts. One part was obtained with this method. On the other hand, figure 5.5 illustrates objects decomposition obtained with Chen's segmentation algorithm. One-ring neighborhood is sufficient for computing local features and

[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. *International Conference on Image Processing*, (3):273276, 2002.

[CG06] L. Chen and N.D. Georganas. An efficient and robust algorithm for 3d mesh segmentation. *Multimedia Tools Appl.*, 29(2):109–125, 2006.

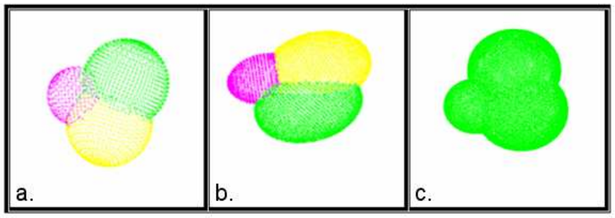


Figure 5.2: Segmentation of three spheres union with different resolutions using Zhang's method: (a) model represented with 2524 points (b) 5091 points and (c) 19020 points.

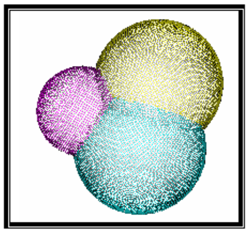


Figure 5.3: Segmentation of the three spheres model represented with 19020 points using Chen's method.

segmenting low resolution objects models, but when dealing with laser scanned objects 6-ring neighborhood was used. The champagne glass is modelled with 137421 vertices, while the mug, the shampoo and bottle are modelled respectively with 183534 and 122307 vertices.

Another example illustrating the advantage of using a multi-ring neighborhood is illustrated in (table 5.1). Notice that the object obtained through a vision system is modelled with 7360 vertices. This corresponds to a low resolution model in comparison to the laser scanned objects. Thus, a 1-ring neighborhood should be sufficient for segmenting such an object. But considering the coarse surface of the reconstructed object, a better decomposition is obtained with a 3-ring neighborhood.

These results show that Chen's algorithm is adapted for segmenting low and high reso-

Table 5.1: Segmenting an object model obtained using vision.

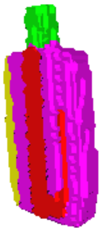
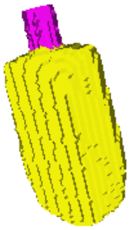
One-ring neighborhood	3-ring neighborhood
	

Figure 5.4: Preparing objects to the scanning procedure.

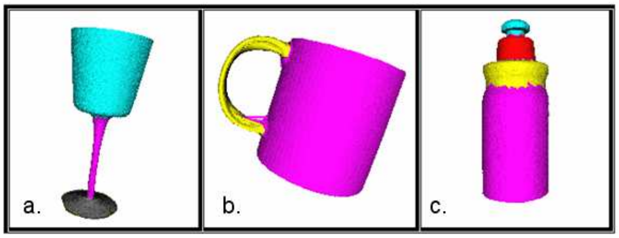


Figure 5.5: Segmentation of real 3D laser scanned objects.

lution models. Thus, It will be used in the following experiments.

5.1.2 Validation of the Learning Algorithm Model

After segmenting objects into parts, each component is fitted with a superquadric. Thus, our learning algorithm uses objects sub-parts sizes and shapes in order to select the grasping component. For this purpose, a multi-layer perceptron with sixteen inputs, one output and one hidden layer is trained with a typical backpropagation learning algorithm. We use 12 two-parts objects for training. In order to increase diversity of our training data, we vary some objects properties such as sub-components sizes, bending angles or tapering parameters without changing the whole appearance of the object. We generate 72 examples for each object. A 10-fold cross validation procedure is then employed to validate the learning algorithm. Thus, our training data is divided randomly into 10 parts. In a first step, the first part is taken apart and used for test data while the 9 remaining parts are used for training data. In a second step, the second part is considered as testing data while the remaining 9 parts are considered as training data. This procedure is repeated ten times. The advantage of this method over repeated random sub-sampling is that all observations are used for both training and validation, and each observation is used for validation exactly once. The 10 results from the folds are then averaged to produce a single score estimation of the training and testing data. We have an average of 99.45% for the training data and of 98.97% for testing data.

5.1.3 First Generalization Test

First, we tested the algorithm on multi-part objects, selected from the database, belonging to the same categories as the training data but of different shapes and sizes. These objects are such as bottles, spoons, knifes, pencils etc. Some of these objects along with their obtained grasping part are shown in (Fig. 5.6). This figure illustrates objects decomposition as well as their graspable part in black. The motivation behind this experiment is that if our algorithm does not work on multi-part objects similar to the training data, then we must conclude that our feature set is not sufficiently discriminative. We use for this test 17 objects. For such objects, the algorithm generalizes very well and was capable of finding each time the handle that human choose to grasp the corresponding object.

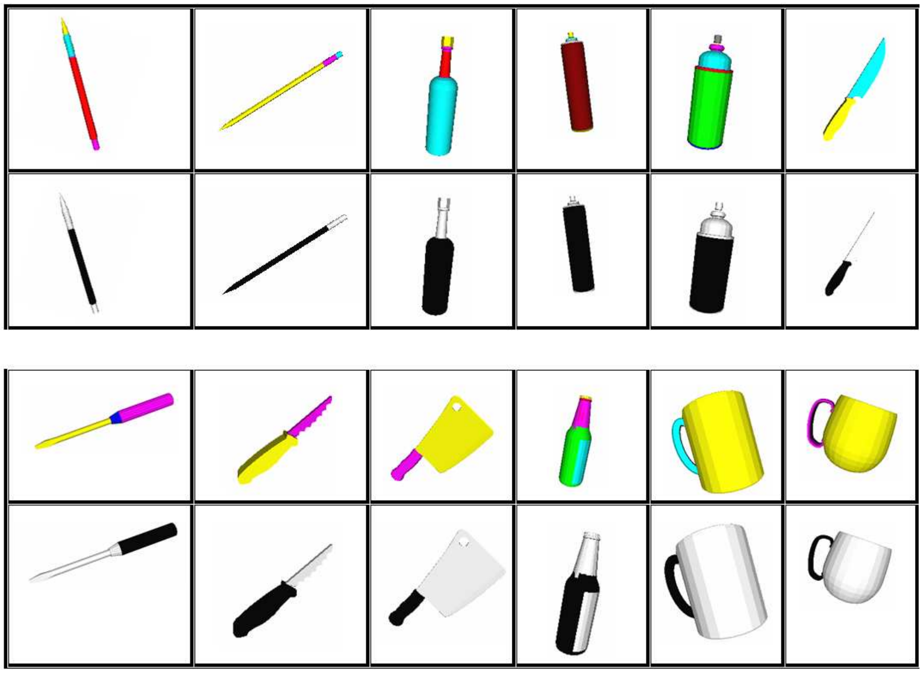


Figure 5.6: Some objects belonging to the same categories as the objects of the training data. The black part indicates the grasping part identified by the algorithm.

5.1.4 Second Generalization Test

In a second time we tested the algorithm on 54 objects that are completely different from those of the training set. This experiment is useful to test the algorithm ability to generalize to completely novel objects. Seven subjects were asked to grasp these objects in order to accomplish a task. We do not specify the task that should be performed. The subjects were supposed to identify objects graspable parts whether they recognize the object or not. Twenty seven objects, *AO* (*Agreed Objects*), were grasped by the same manner. On the other hand, the remaining 27 objects, *CO* (*Confusing Objects*), induced confusion and the seven subjects chose different parts to grasp them. We remind the reader that our aim is to imitate humans choice of the graspable part. The distinction between *AO* and *CO* objects is necessary for measuring our algorithm performance. Their success grasp rate is computed differently.

5.1.4.1 Success Grasp Rate for Agreed Objects

The seven subjects totally agreed on the Agreed Objects handles. Thus, for *AO* objects, whenever the algorithm selects for grasping a part different from the one identified by the seven subjects, it is considered a failure. Some of the *AO* objects are shown in (Fig. 5.7) along with the graspable part identified by the algorithm in black. Objects parts that are marked with a cross are the ones corresponding to humans choice.The system succeeds to find the correct graspable parts for 22 *AO* objects, which corresponds to a successful grasp rate of 81%. This rate shows that features such as sizes and shapes of novel objects subparts are about 81% discriminative to determine the object natural grasping part. An interesting analysis can also be done when computing the grasping rate on objects that the subjects could not identify. This rate will show the ability of the algorithm to imitate humans when grasping unknown objects to them. Figure (5.8) illustrates these objects. Objects b, c, d and

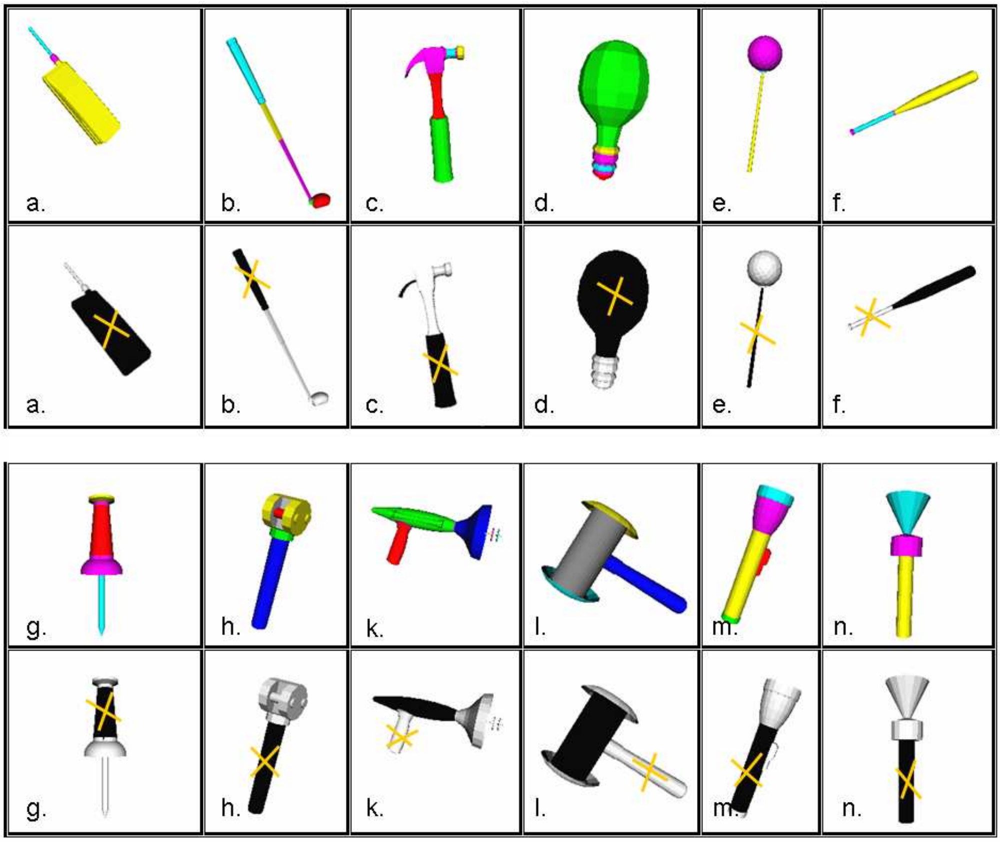


Figure 5.7: Examples of *AO* objects. The black part indicates the corresponding object graspable part identified by the algorithm. The crossed part indicates the one chosen by humans.

Table 5.2: Success Grasp Rate for *AO*.

	AO objects	
	Known	Unknown
Grasp rate computed on the basis of	All people	
Total number	27	
Number	23	4
Number grasped successfully	18	4
Success Grasp Rate	78%	100%

e are *AO* objects. Table (5.2) shows the grasp rate obtained. We have in our database only 4 unknown *AO* objects. This number may seem insufficient for computing a corresponding grasp rate. But, the reader should notice that is hard to find unknown objects. Constructing arbitrary objects may induce unknown objects, however it is contradictory to the basic idea of our grasping strategy, i.e objects are made specifically in a way that makes their grasp easier. Thus, constructing useless objects is not a solution. The algorithm score is of 78% and 100% for known and unknown objects respectively. This result proves that our algorithm completely succeeds in imitating humans choice of novel objects graspable part. In other words, features such as sizes and shapes of an unknown object subparts are discriminative to determine unknown objects natural grasping parts.

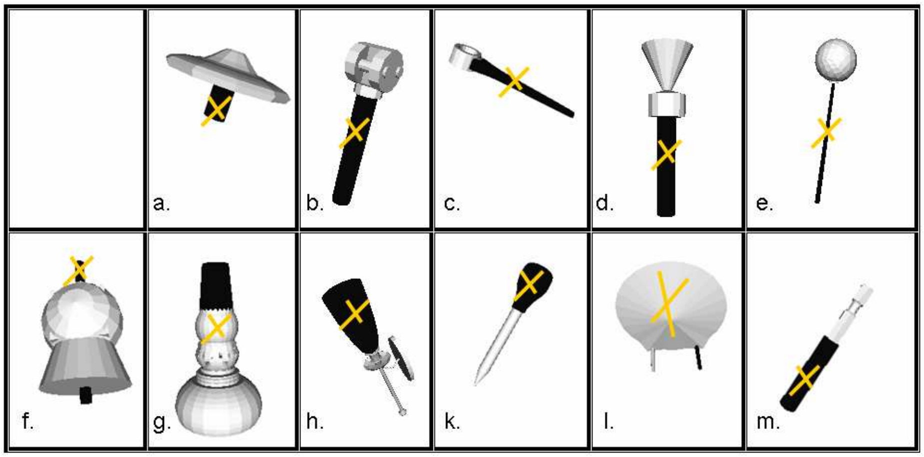


Figure 5.8: Unknown objects grasping. The black parts indicate the ones selected by the algorithm. Objects b, c, d and e are *AO* objects. Objects a, f, g, h, k, l and m are *CO* objects.

5.1.4.2 Success Grasp Rate for Confusing Objects

The seven subjects chose different parts to grasp Confusing Objects. Since humans grasp these objects in various ways, two successful rate may be computed: a successful grasp may be a grasp that identifies the object part chosen by most people, or a successful grasp may be a grasp that identifies a part chosen by at least one person. Otherwise, failure occurs. Figure (5.9) shows some examples of CO objects. The black part indicates the one chosen by the system and the cross-marked part is the one corresponding to most people choice. The algorithm succeeds to find, for 15 CO objects, the part selected by most people. This corresponds to a successful grasp rate of 55%. When considering a grasp rate on the basis of "at least chosen by one person", the algorithm perform well for 23 CO objects which corresponds to a rate of 85%. Table (5.3) shows the grasp rate obtained when dividing CO objects into known and unknown objects. On the basis of "at least one person grasp it this way", the algorithm score is of 100% for unknown objects. These objects are illustrated in (Fig. 5.8).

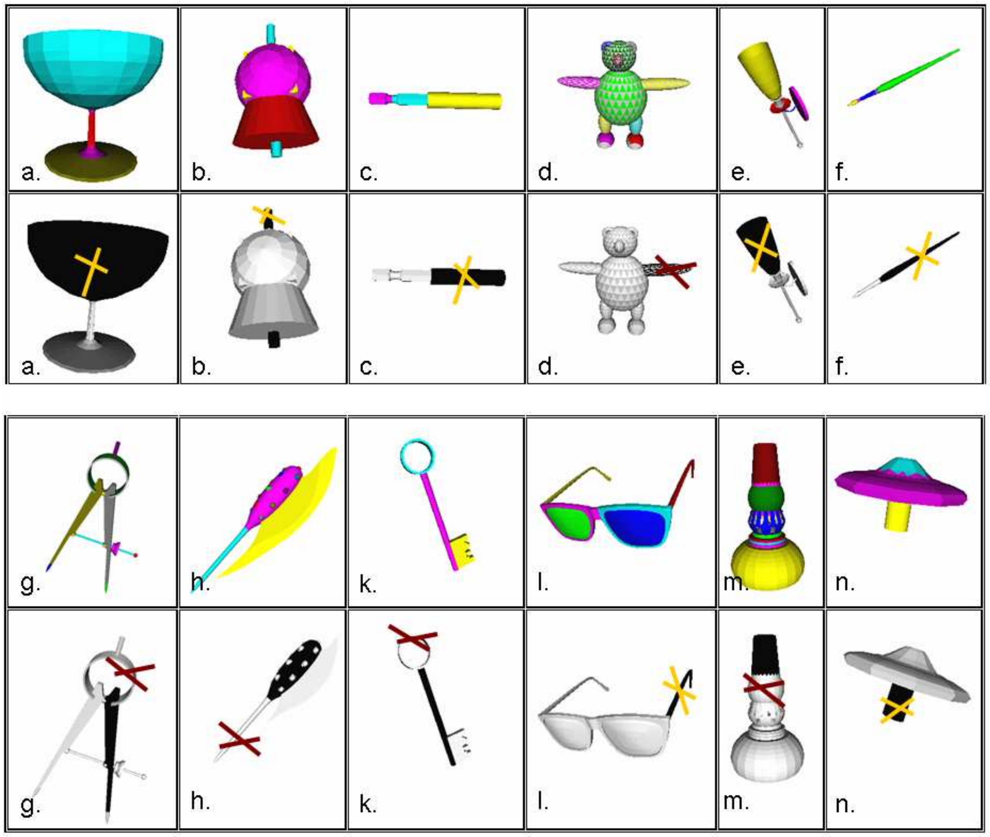


Figure 5.9: Examples of CO objects. The black part indicates the system choice. The cross-marked parts indicate most humans choice.

5.1.5 Discussion and Limitations of the GBC Strategy

The previous two experiments test the ability of the learning algorithm to generalize. The first experiment tests the algorithm on multi-part objects from the same type of those of the training set. The algorithm performs very well on such objects (success rate of 100%). This shows that our feature set is sufficiently discriminative. Thus, the algorithm can be tested

Table 5.3: Success Grasp Rate for *CO* objects.

	CO objects			
	Known		Unknown	
Grasp rate computed on the basis of	MP	ALO	MP	ALO
Total number	27			
Number	20		7	
Number grasped successfully	10	16	5	7
Success Grasp Rate	50%	80%	71%	100%

on objects completely different from those of the training data. These objects were divided into *AO* and *CO* objects whether humans agreed or not on their graspable part. When a success grasp rate was computed on the basis of "at least one person grasp it this way", our algorithm have a success rate of about 80% for known objects and 100% for unknown objects. These results show that Humans use indeed unknown objects subparts shapes and sizes in order to grasp them. However, the proposed algorithm fails to determine the graspable part chosen by most people for objects such as those shown in Figure (5.10). These examples illustrates the two limitations of the algorithm:

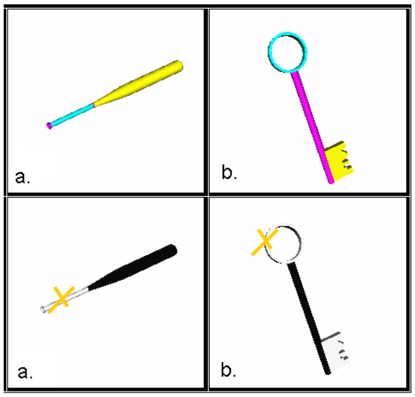


Figure 5.10: Some objects for which the algorithm fails to select the graspable part that most people choose. The black part indicates the one selected by the algorithm. The cross-marked part indicates the part chosen by most people.

- Failing to grasp objects with holey parts such as keys or compasses is due to modelling capabilities limitation. Thus, a solution is to enhance objects modelling by adding toroid shaped superquadrics that in many cases are relevant to the grasping problem.
- Objects with similar components are grasped in the same manner. That was the basic idea of the algorithm. But, although some objects are similar from a geometric point of

view, we grasp them differently. This is the case of the baseball bat. When represented as an assembly of superquadrics, a baseball bat can also be viewed as a screwdriver. This explains the graspable part chosen by the algorithm. To overcome this failure, other features about the object should be included such as its color or material etc.

The limitations described above constitute the two main failure reasons of the proposed approach. Thus, they represent the 20% of objects. The algorithm succeeds to identify the natural grasping component of the other 80% objects as shown previously.

5.2 From the Grasping Component to the Grasping Points

At this point, we are able to identify an unknown object handle. This section aims at computing contact points on the corresponding handle that ensure stability. The force-closure property guarantees stability. It is well known that to achieve force-closure, the grasp matrix should positively span the wrench space (Chapter4:*proposition* 1). Our method generates first, randomly, locations of $n-1$ fingers. We showed that wrenches associated to any three non-aligned contact points of 3D objects form a basis of their corresponding wrench space (Chapter4:*proposition* 19). Thus, we can find 6-dimensional basis from the wrenches associated to these $n-1$ contacts. A position of the nth finger is located such that an associated contact wrench can be uniquely expressed as a strictly negative linear combination of such basis (Chapter4:*proposition* 21). We choose the wrench associated to the normal force on the nth contact. We compare our method with the classical complete method based on the construction of a 6D convex hull [MKAC03]. The latter involves approximating the contact friction cones as a convex sum of a finite number of force vectors around the boundary of the cone, computing the associated object wrench for each force vector, and then finding the convex hull of this set of wrenches. If the origin is contained within this space, the grasp have force-closure. Otherwise, there exists some set of disturbance wrenches that cannot be resisted by the grasp.

We accomplish tests on a sphere model, represented by its 762 vertices and its respective normal directions. Two experiments are performed in order to show the efficiency of the proposed approach. The first test aims at studying the completeness of the approach. The purpose of the second test is to compare the force-closure grasp computation time of our approach to that of the convex hull. Since these two methods require the cone to be linearized and since all lines through one point are of rank 3 (from Chapter4:*proposition* 16), without loss of generality, we use a 3-sided pyramid to represent a linear model of a cone. With the latter model of the friction cone, the maximal number of basis computed from the wrenches associated to the 3 fingers is 84. The friction coefficient is set to 0.5 (corresponds to the

[MKAC03] A.T. Miller, S. Knoop, P.K. Allen, and H.I. Christensen. Automatic grasp planning using shape primitives. *In Proceedings of IEEE International Conference on Robotics and Automation*, 2003.

coefficient between glass and metal). The experiments were run on Pentium Core duo machine with $2GB$ memory and a CPU at 2.13 GHz . The program is implemented in C++.

5.2.1 Completeness Test

The completeness test aims at computing all possible n-finger force-closure grasps on a 3D object with our method and that of the convex-hull. The algorithm below details the steps of the completeness test. The corresponding experimental results are shown in (table 5.4).

```
Require: - 3D points representing the object
         - Linearized friction cone at each point and corresponding wrenches
Ensure:  - Average number of 4 fingers force-closure grasps
$nFC\_new \leftarrow 0$
$nFC\_classic \leftarrow 0$
$nL \leftarrow 0$
$NLoc \leftarrow$ number of all 3 fingers locations non-aligned and not in force-closure
for i from 1 to $NLoc$ do
  L_basis = Find_Basis (L_wrenches)
  q_L=quality(L_basis)
  if q_L <Th then
    $i \leftarrow i + 1$
  end if
  $nL \leftarrow nL + 1$
  for all object vertices do
    if new_Force_Closure (vertex, L_basis) then
      $nFC\_new \leftarrow nFC\_new + 1$
    end if
    if classic_Force_Closure (vertex,L_wrenches) then
      $nFC\_classic \leftarrow nFC\_classic + 1$
    end if
  end for
end for
return $nFC\_new/nL$ and $nFC\_classic/nL$
```

Given a representation of an object along with normal directions and a friction coefficient, wrenches associated to each of its vertices are firstly computed. The completeness test consists in generating all locations of 3 fingers non-aligned and not in force-closure on the sphere. For each 3 fingers locations corresponds 9 wrenches, $L\_wrenches$. All 6-dimensional basis, L_basis, from the $L\_wrenches$ are determined by *Find_Basis*. To ensure a robust force-closure grasp, a quality criterion of the locations of the 3 fingers, q_L is computed by *quality* function. q_L is computed by calculating the quality of each basis according to our

Table 5.4: Completeness test

	Number of Solutions (s)		
	classic	new	n/c
Th0	315	59	18.73%
Th1	349	82	23.5 %
Th2	366	96	26.23 %
Th3	392	118	30.1 %
Th4	419	149	35.56 %

criterion. Thus, *q_L* is the maximum quality of their corresponding basis. If *q_L* is above a threshold Th, good locations for the first three fingers are found and nL is incremented. All object vertices are then tested for a 4-finger force-closure grasp with *new_Force_Closure* and *classic_Force_Closure* (of course with the same first three fingers locations for the two methods). These two functions correspond to our approach and the convex-hull method. A position of the $4th$ finger is located such that its normal contact wrench can be uniquely expressed as a strictly negative linear combination of one of the *L_basis*. If the grasp ensures force-closure, *nFC_new* or *nFC_classic* is incremented according to the approach used. The algorithm returns the average number of 4-fingers force-closure grasps found with the two approaches.

Results: Completeness results for different thresholds $0 = Th0 < Th1 < Th2 < Th3 < Th4$ are shown in (table 5.4). The threshold value varies according to the object geometry and consequently was empirically chosen. Note that Th0 is null and thus presume no constraints on the generation of the first $n-1$ fingers. The latter are thus generated randomly ensuring no minimal quality criterion. We notice that the solutions found by our approach regarding the convex-hull one increases with the threshold. This proves the robustness of the quality criterion proposed. In spite of a good selection of the first three fingers locations, a completeness of only 35% is obtained. Thus, our method reports fault negative results (the method implies no force-closure when it exists). That is due to two main reasons. The first one is the linearization of the friction cone. The second is due to testing only the normal wrench associated to the nth finger for force-closure. The next paragraph will show that even with a low rate of completeness, i.e with $Th0$, our approach will generate a more robust force-closure grasp with at least a quarter computation time needed by the convex-hull method.

5.2.2 Rapidity Test

This paragraph measures the computation time for generating one n-finger force-closure grasp. The algorithm below details the different corresponding steps for generating 4-finger force-closure grasps.

```
Require: - 3D points representing the object
         - Linearized friction cone at each point and corresponding wrenches
Ensure:  - A 4 fingers force-closure grasp
 1: $\alpha \in [0; 1]$
 2: L = Rand_Fingers (3)
 3: L_basis = Find_Basis (L_wrenches)
 4: q_L = quality(L_basis)
 5: if q_L <Th then
 6:     Go to step 2
 7: end if
 8: B_basis = Best_Basis(L_basis)
 9: randomly choose $x \in ]0; 1[$
10: if $x > \alpha$ then
11:     $n \leftarrow 0$
12:     while ($n < Npoints$) and (No Force-Closure found) do
13:         $n \leftarrow n + 1$
14:         vertex=Rand_Finger(1)
15:         Force_Closure (vertex, B_basis or L_wrenches)
16:     end while
17:     if No Force-Closure found then
18:         Go to step 2
19:     end if
20: else
21:     vertex = Rand_Finger(1)
22:     Force_Closure (vertex, B_basis or L_wrenches)
23:     if No Force-Closure found then
24:         Go to step 2
25:     end if
26: end if
```

A 4-finger force-closure grasp may be generated in two different ways. The tuning parameter α will indicate the best way. L stands for the randomly generated locations of 3 fingers non-aligned and not in force-closure on the sphere. Since computing a good grasp depends on a good placement of the first 3 fingers as well as the location of the fourth finger, L must ensure a minimal quality criterion. This criterion computation is detailed in the previous chapter. Once good 3 fingers locations are generated, we compute their corresponding wrench space basis and determine the best quality basis, B_basis. The value x permits to choose between two methods. The first method consists in testing till $Npoints$ vertices for force-closure with B_basis. The second method tests only one randomly generated vertex for force-closure with B_basis. When no force-closure grasp is found, we proceed to generate another locations for the first 3 fingers. The variable x can take values between 0 and 1. When $x > 0.5$, the second method is privileged. Thus, when no force-closure is obtained when testing a vertex for force-closure with the generated $n-1$ vertices, we privilege regenerating new locations for the $n-1$ vertices on testing another vertex with the same first $n-1$ vertices positions. On the contrary, when $x < 0.5$ the first method is privileged. The $Force\_Closure(vertex,\ B\_basis\ or\ L\_wrenches)$ function performs n-finger force closure tests. It takes $B\_basis\ or\ L\_wrenches$ as arguments wether force-closure is tested according to the new or the classic convex-hull method respectively.

Results: Experiments were conducted using different thresholds with the convex-hull and our method. Figure (5.11) shows the evolution of a force-closure grasp computation time with respect to α, for the threshold $Th0$. We remind the reader that Th0 is null and thus presume no constraints on the first n-1 generated fingers locations. This procedure occurs randomly with the only condition of non-collinearity. With this threshold, the best force-closure grasp computation time is obtained for $\alpha = 0.9$, thus when regeneration of fingers locations is privileged (case of $x > 0.9$).

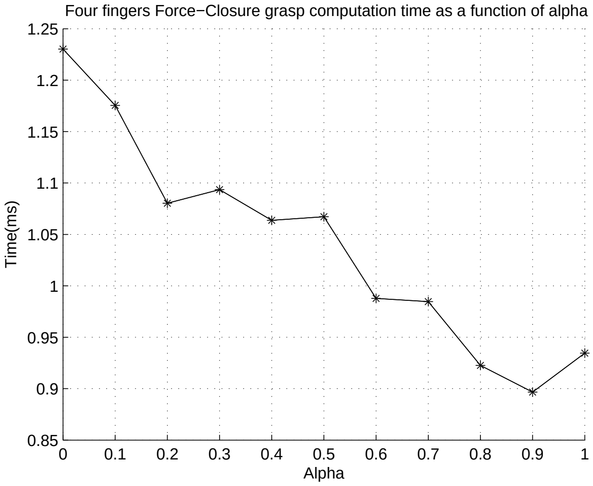


Figure 5.11: 4-finger force-closure grasp computation time using our method with the threshold $Th0$

Figure (5.12) illustrates the influence of the threshold. This graph demonstrates that the better the locations of the first three fingers are chosen, the lower is the computation time of a 4-finger force-closure grasp. The best results are obtained for the threshold $Th4$.

Table (5.5) compares for this threshold and by varying α our method and that of the convex-hull computation time. The *classic* column gives in ms the time for computing one 4-finger force-closure grasp according to the previously described algorithm, with the convex-

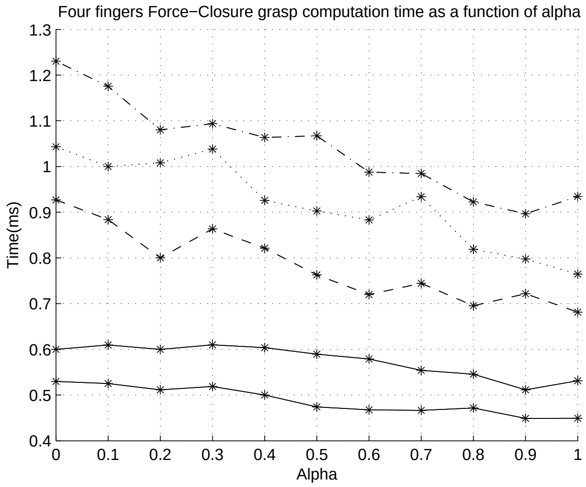


Figure 5.12: 4-finger force-closure grasp computation time using our method with thresholds $Th0 < Th1 < Th2 < Th3 < Th4$

hull method. The *new* column does the same when our approach is used. The last column show the ration of our method computation time to that of the classic method. This ratio varies approximately between 15% and 17.7%. Thus in the worst case, our method is more than five times faster in finding a force-closure grasp than the convex-hull method. In (table 5.6), the same ratio is computed for $\alpha = 1$ and by varying the threshold. In the worst case, when Th0 is considered, this ratio is of 25% approximately, thus our method is four times faster than the convex-hull one. Consequently, when generating the first three fingers randomly without any quality criterion, our method is still faster in computing 4-finger force-closure grasps than the convex-hull method. We notice, that when the threshold increases, the force-closure computation time decreases for both the convex-hull and for our method. This is normal, because a better choice of the first three fingers increases the number of possible 4th finger ensuring 4-finger force-closure grasps.

Table (5.7) shows the quality of the force-closure grasps computed with the two approaches. The quality is obtained according to the largest ball criterion. The quality of the force-closure grasps obtained with our approach is approximately twice better than that of the classical method. Thus, although our method is not complete, it finds good solutions for the force-closure problem.

5.2.3 Discussion

The proposed force-closure condition is sufficient but not necessary. It is based on our conviction that rapidly generating good grasps depends on the locations of the first generated $n-1$ fingers as much as the positioning of the nth finger. Experimental results show that when the $n-1$ fingers are generated randomly, our method is four times faster than the convex-hull one. When the latter are chosen according to our quality criterion, the quality of

Table 5.5: Force-Closure grasp computation time with $Th4$

	Time (ms)		
α	classic	new	n/c
0	3.00382	0.529787	17.64 %
0.1	2.96908	0.525141	17.69 %
0.2	3.00494	0.511493	17.02 %
0.3	3.17757	0.518583	16.32 %
0.4	3.05470	0.499915	16.36 %
0.5	3.07801	0.474132	15.40 %
0.6	3.04441	0.467638	15.36 %
0.7	3.08814	0.466491	15.10 %
0.8	2.80214	0.471555	16.83 %
0.9	2.81084	0.449039	15.97 %
1	2.83721	0.449150	15.83 %

Table 5.6: Force-Closure grasp computation time with $\alpha = 1$

	Time (ms)		
threshold	classic	new	n/c
Th0	3.74062	0.934578	24.98 %
Th1	3.60284	0.764612	21.22 %
Th2	3.43498	0.681242	19.83 %
Th3	3.19753	0.531346	16.62 %
Th4	2.83721	0.44915	15.83 %

Table 5.7: Force-Closure grasp generated quality with $\alpha = 1$

	Mean_Quality		
threshold	classic	new	n/c
Th0	0.0689	0.1431	2.0769
Th1	0.0700	0.1445	2.0643
Th2	0.0731	0.1458	1.9945
Th3	0.0826	0.1545	1.8705
Th4	0.0955	0.1627	1.7037

the obtained grasps is better and our method is six times faster than the convex-hull. This is due to two main reasons:
1) The force-closure test is easy to compute. It is reduced to an inverse matrix calculation.
2) Another main advantage on the convex-hull method is that a construction of a convex-hull is needed whenever a new finger is tested for force-closure with the other $n-1$ fingers. On the other hand, when a basis is associated to the first $n-1$ fingers, it can be used to test all remaining vertices for a n-finger force-closure grasp.

5.3 Grasping By Components

We proposed a grasping strategy that determines in a first place an unknown object handle and is able then to generate, on this component, contact points ensuring the grasp stability. This section shows in a first place examples of contact points computation on the objects graspable parts. It shows then the adaptability of our strategy to the hand kinematics.

5.3.1 From the 3D Model to the Contact Points

This paragraph tests our grasping strategy on different objects models. We first show examples of contact points generation on the graspable parts of synthetic objects (table 5.8). We also run our algorithm on real objects. The 3D model of such objects is acquired either with a 3D laser scanner (table 5.9) or through a vision system (table 5.11). The former is obtained by scanning the object from different view-points using a turntable. The latter is obtained by performing a 3D reconstruction based on 12 images of the object taken from different angles. The reconstruction method used combines the Visual Hull and optimization by Graph-Cuts methods [WD08]. The yellow part indicates each time the grasping component identified by the learning algorithm. The GBC time indicates in seconds the whole process, i.e time required for decomposing objects into single parts, approximating each part by a superquadric, selecting the graspable component and generating a 4-finger force-closure grasp on the selected part. This time increases with the number of vertices constituting the object. The reader may notice that the generated contact points do not always correspond to those that humans choose to grasp the selected objects components. This however is sufficient for ensuring task compatibility [AC08].

5.3.2 Grasping by Taking into account the Hand Kinematics

At this point, our grasping strategy identifies an unknown object handle and generates contact points on it with the only constraint of stability. Dealing with a robotic hand model

[WD08] G. Walck and M. Drouin. Reconstruction 3d progressive et rapide. *MajecStic08*, 2008.

[AC08] J. Aleotti and S. Caselli. Programming task-oriented grasps by demonstration in virtual reality. *In Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, WS on Grasp and Task Learning by Imitation*, 2008.

Table 5.8: Generating 4-finger force-closure grasps for synthetic objects.

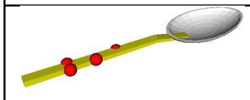
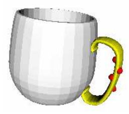
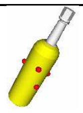
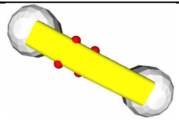
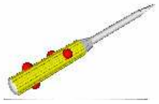
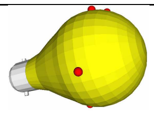
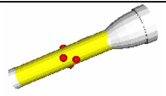
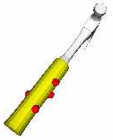
Objects Grasps	Vertices Number	GBC time(s)
	629	2.59
	1725	3.73
	7613	7.82
	6010	7.1
	5363	5.76
	918	2.8
	3394	4.88
	850	4.12

Table 5.9: Generating 4-finger force-closure grasps for laser scanned objects.

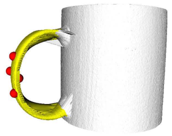
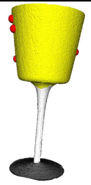
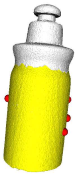
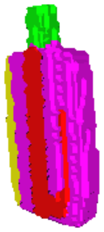
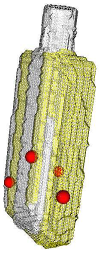
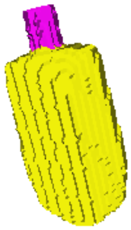
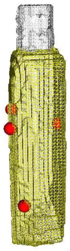
Objects Grasps	Vertices Number	GBC time(s)
	183534	267.63
	137421	173.8
	122307	159.82

Table 5.10: Generating 4-finger force-closure grasps for an object model obtained using vision. The segmentation is performed with one and 3-ring neighborhoods.

Object Segmentation	Object Grasp	Vertices Number	GBC time(s)
		7360	10.45
		7360	8.87

induces additional kinematical and geometrical constraints. Taking these constraints into account results in limiting possible locations for the contact points on the graspable part. The latter should be kinematically feasible for the fingers and they should also avoid collision with the hand, the remaining fingers and the object. Consequently, these contacts should be generated in respect of the accessibility domains of the fingers. Furthermore, a grasp involves several closed kinematic loops between the fingers and the object. Randomly generation of a closed kinematic chain is very difficult. In order to handle these closed kinematic chains and inspired by the thesis of Jean-Philippe Saut [Sau07], we propose to adapt the RLG (Random Loop Generator) algorithm [Cor03] to our grasping strategy. RLG aims at handling closed kinematic loops by dividing them into active and passive parts. The idea of the algorithm is to reduce the closed kinematic chain complexity iteratively until the active part becomes reachable by all passive chain segments simultaneously. In our case, the object is the active part while the fingers constitute the passive parts. A grasp can occur when the object is reachable by all the fingers. The reachable workspace of a kinematic chain is defined as the volume which the end-effectors can reach. RLG approximates such volume with a sphere. Figure 5.13 illustrates an example of the reachable workspace of a finger. It also shows the intersection between this space and the object. Thus, the finger should be placed on this intersection. The placement of the first finger is then taken into account when computing the second finger reachable workspace and so on until the placement of all fingers. We modify our grasping strategy to take these constraints into consideration.

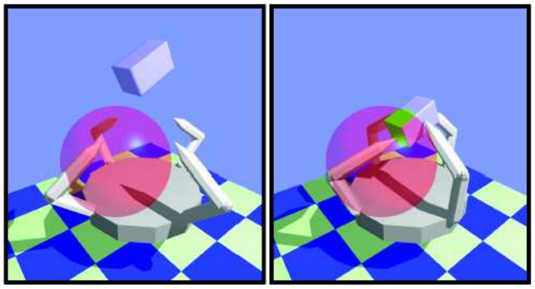


Figure 5.13: The red sphere represents the reachable workspace approximation of a finger. The green part stands for the object surface reachable by the finger. Thus, this finger should be placed on that green part [Sau07].

```
Require: - 3D points representing the object graspable part
         - Linearized friction cone at each point and corresponding wrenches
Ensure:  - n-fingers force-closure grasp for a given object/hand models
Move Hand to Object
$RW_1 \leftarrow Grasp\_RLG(f_1,\text{object})$
```

[Sau07] J.P. Saut. Planification de mouvement pour la manipulation dextre d'objets rigides. *Thése de doctorat de l'Universite Pierre et Marie Curie*, 2007.

[Cor03] J. Cortes. Motion planning algorithms for general closed-chain mechanisms. *Thése de doctorat de l'Institut National Polytechnique de Toulouse*, 2003.

```
3: $CP_1 \leftarrow$ Rand($RW_1$)
4: i $\leftarrow$ 1
5: while $i < n - 1$ do
6:    $RW_i \leftarrow$ Grasp_RLG($f_i$,object, $\sum_{k=1}^{i-1} f_k$)
7:    if $RW_i$ then
8:       $CP_i \leftarrow$ Rand($RW_i$)
9:    else
10:      Go to step 3
11:   end if
12: end while
13: $RW_n \leftarrow$ Grasp_RLG($f_n$,object, $\sum_{k=1}^{n-1} f_k$)
14: FC $\leftarrow$ 0
15: while !FC do
16:    $CP_n \leftarrow$ Force_Closure($\sum_{k=1}^{n-1} CP_k$)
17:    FC $\leftarrow$ $CP_n$ in $RW_n$
18: end while
19: END
```

Since we are interested in computing grasps on the object handle, an obvious first step of our algorithm is to move the hand towards the handle until this object sub-part is reachable by a finger. RW_i stands for the intersection of the reachable workspace of a generated finger f_i and the object. *Grasp_RLG* permits the estimation of RW_i by taking into consideration the object and the $i-1$ fingers positions. A contact location CP_i is then randomly chosen in RW_i. This guarantees that the inverse geometrical model of the finger existence. After placing the $n-1$ fingers, the nth finger location is computed with $Force\_Closure$ in order to ensure the grasp stability. This algorithm can be enhanced by introducing a quality criterion, while generating the first n-1 fingers, that takes into account the hand model. Figure **??** shows several grasps obtained using DLR and Rutgers hands models and GraspIT interface. The latter uses PQP algorithm to detect collisions [Mil01].

[Mil01] A.T. Miller. Graspit!: A versatile simulator for robotic grasping. *PhD thesis, Columbia University*, 2001.

Table 5.11: Generating 4-finger force-closure grasps using DLR hand model in GraspIT interface.

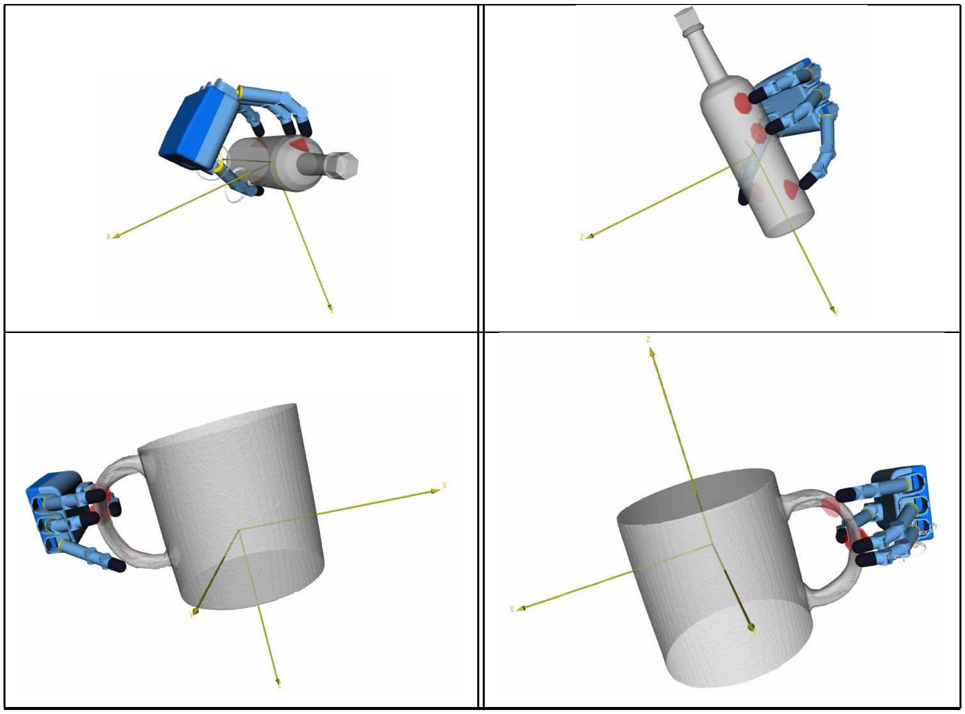
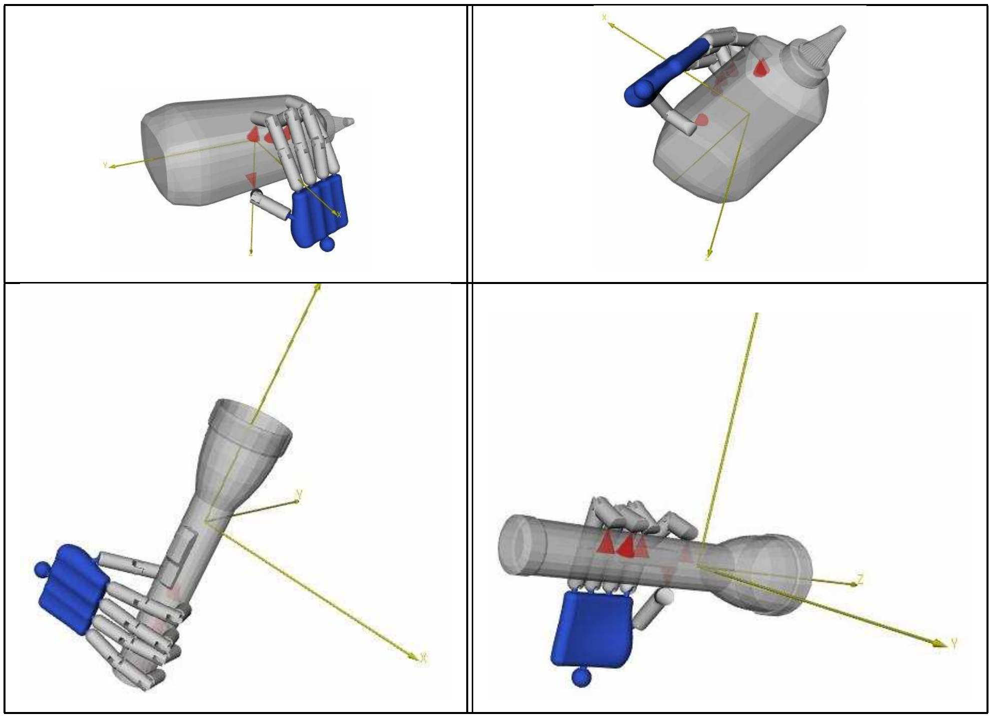


Table 5.12: Generating 5-finger force-closure grasps Rutgers hand model in GraspIT interface.



5.4 Conclusion

Different experiments were detailed in this chapter in order to test the ability of our grasping strategy to predict grasps of unknown objects that conform humans grasping. The first experiment characterizes how well learned grasps generalize to objects that the algorithm has no experience with. Results show that features such as objects sub-parts shapes and sizes are about 80% discriminative to grasping. In other words, an unknown object appropriate grasp can be found only by using information on its constituting components shapes and sizes without any task modelling. Once the graspable part is identified, contact points should be determined. For this purpose, we proposed a new sufficient condition for generating n-finger force-closure grasps. The main advantage of the proposed condition is its fast computation of good grasps on the selected component. Its efficiency was confirmed by comparing it to the classical convex-hull method. Finally, an algorithm for generating contact points on a novel object that takes the hand kinematics into account was proposed.

Chapter 6

Conclusions

This thesis addresses the problem of grasping unknown objects. The latter can be stated as follows: given a previously unknown object, determine a set of contacts on the object surface appropriately in order to ensure stability and to successfully perform a task. Despite the quantity of relevant work in the field, this problem remains challenging and tackles with two main issues:

- Task modelling.
- Generalizing learned grasping skills to new objects.

We overcome these difficulties and propose a strategy that associates a grasp to an unknown object/task by taking inspiration from humans behavior and Biedermann's theory of Recognition By Components. The latter states that humans recognize objects by representing them as an assembly of parts. On a part-representation level, it seems obvious that each object is equipped with a part designed specifically to make its grasp easier, its handle. We define a handle or the natural grasping component of an object as the part chosen by humans to pick this object with. When people reach out to grasp an object, It is in the aim of accomplishing a task. Thus, their grasp is related to the object function. Consequently, identifying an object handle yields a successful grasp in accordance with the object function. Our method learns humans choice of the grasping component based on information such as objects subparts shapes and sizes. Thus, objects are decomposed into single parts and each part is then fitted with a superquadric. The grasp stability is obtained by determining contact points on the object handle verifying force-closure property.

We implemented the proposed approach and tested its ability to generalize on previously unknown objects models whether synthetic or real via laser scanning or 3D reconstruction using a vision system. The experiments show that the algorithm succeeds in imitating humans when grasping unknown objects. A score of about 80% is obtained. Thus, we can conclude that geometric features such as objects sub-parts shapes and sizes, are indeed relevant to objects grasping. The gap of 20% is due to the fact that some objects have similar shapes but different functionalities, hence, they are grasped differently. Once the object graspable part is determined, we process at the generation of grasping contact points on it. For this purpose, a new sufficient force-closure grasp condition is introduced. It aims at reducing good force-closure grasps computation time. It generates locations of n-1 non-aligned fingers according to a quality criterion and finds then locations of the *nth* finger that ensures force-closure. We studied its rapidity and completeness aspects in two experiments. Results showed that, in the worst case, when no quality criterion is used to generate locations of the first $n-1$ fingers, it is four times faster in finding a force-closure grasp than the classical convex-hull method. Finally, we also adapted the algorithm to take into consideration a robotic hand model and corresponding kinematics constraints.

Nowadays, many researchers try to understand humans behavior by programming neuroscience theories in humanoid robots and vice versa [AHFPK00]. Our work is also based on the psychological RBC theory and other neuroscience studies. But, will the high scores obtained validate the corresponding theories? Are primitives such as objects sub-parts shapes and sizes sufficient to identify novel objects grasps? Do we actually understand how people grasp unknown objects? The work presented in this thesis is a first step towards a full understanding of humans behavior and may be our learning algorithm needs to be enhanced to take into account recent/future studies on humans brain.

6.1 Future Works

This work takes into consideration the robotic hand kinematics in order to generate contact points. A complete approach will couple our grasping strategy with a planning algorithm permitting to account for the environment obstacles. Consequently, when a robot cannot reach the object graspable part due to collisions, it may plan another feasible grasp (Fig. 6.1).

In this thesis, we test our grasping strategy on 3D objects models (CAO, laser scanned and reconstructed using vision system) with different resolutions. Our approach occurs in two steps: (1) acquiring the object model and then (2) determining the grasp. Obtaining a complete 3D model of the object is not simple to achieve. Thus, It may be interesting to pair object reconstruction with grasp generation. A current work with the university of Coimbra consists at recovering objects shapes using probabilistic volumetric maps. These grid-based maps are also known as occupancy grids and are used to represent distributed spatial information, such as occupancy [Thr02]. In order to reconstruct a 3D object, its enveloping workspace is discretized and mapped to different cells with a given resolution. Electromagnetic markers are used to track the motion of a human hand exploring the object surface. Afterwards, each cell has a probabilistic belief about its state (whether belonging to

[AHFPK00] Christopher G. Atkeson, Joshua G. Hale, Shinya Kotosaka Stefan Schaal Tomohiro Shibata Gaurav Tevatia Ales Ude Sethu Vijayakumar Frank Pollick, Marcia Riley, and Mitsuo Kawato. Using humanoid robots to study human behavior. *IEEE Intelligent Systems*, 15(4):46–56, 2000.

[Thr02] S. Thrun. Robotic mapping: a survey. *Exploring Artificial Intelligence in the New Millennium*, 2002.

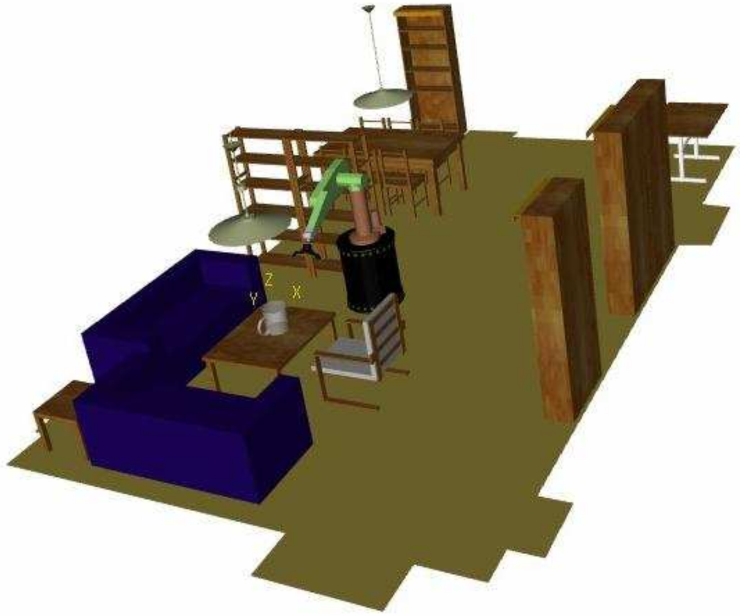


Figure 6.1: Grasping the mug by its handle is impossible in such situation (This is a GraspIT environment [MKAC03]). Another grasp should thus be computed.

the object surface or not). With each exploration of the object, the probabilistic volumetric map is updated. Thus, the map's uncertainty is reduced. Consequently, objects reconstruction occurs incrementally and the idea is to couple our grasping strategy with this gradually increasing representation. Thus, contact points generation and objects reconstruction are looped together until a feasible grasp is obtained. The stop criterion will indicate the quality or the resolution required to accurately grasp and manipulate the corresponding object.

Publications

- **[IEEE Trans.]** S. El Khoury and A. Sahbani, *On Computing Robust N-Finger Force-Closure Grasps of 3D Objects*, Submitted to IEEE Transactions on Robotics and Automation.

- **[ICRA'09]** S. El Khoury and A. Sahbani, *A Sufficient Condition for Robust Force-Closure Grasps Synthesis of 3D Objects*, Submitted to IEEE International Conference on Robotics and automation, Japan, 2009.

- **[IROS'08]** S. El Khoury and A. Sahbani, *Handling Objects by Their Handles*, IEEE/RSJ International Conference on Intelligent Robots and Systems, WS - "grasp and task learning by imitation", pp. 58-64, Nice, France, 2008.

- **[IROS'08]** S. El Khoury and A. Sahbani, *A Sufficient Condition for Force-Closure Grasps Synthesis of 3D Objects*, IEEE/RSJ International Conference on Intelligent Robots and Systems, Poster Session, Nice, France, 2008.

- **[RAM'08]** S. El Khoury and A. Sahbani, *A sufficient condition for computing n-finger force-closure grasps of 3D objects*, IEEE International Conference on Robotics, Automation and Mechatronics, Chengdu, China, 2008.

- **[IROS'07]** S. El Khoury, A. Sahbani and V. Perdereau, *Learning the natural grasping component of an unknown object*, In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 2957-2962, San Diego, CA, USA, 2007.

- **[IROS'07]** J.-P. Saut, A. Sahbani, S. El Khoury and V. Perdereau, *Dexterous manipulation planning using probabilistic roadmaps in continuous grasp subspaces*, In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 2907-2912, San Diego, CA, USA, 2007.

- **[JNRR'07]** S. El Khoury, A. Sahbani, and V. Perdereau *Apprentissage de la partie prhensible d'un objet de forme quelconque*, Journée Nationale de la Recherche en Robotique, Obernai, 2007.

- **[MAJECSTIC'06]** S. El Khoury, A. Sahbani and V. Perdereau *Apprentissage par imitation pour la saisie d'objets de formes quelconques*, MajecSTIC 2006, France, **Prix du meilleur orateur**.

Bibliography

[AC07] J. Aleotti and S. Caselli. Robot grasp synthesis from virtual demonstration and topology-preserving environment reconstruction. *In Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2007.

[AC08] J. Aleotti and S. Caselli. Programming task-oriented grasps by demonstration in virtual reality. *In Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, WS on Grasp and Task Learning by Imitation*, 2008.

[AHFPK00] Christopher G. Atkeson, Joshua G. Hale, Shinya Kotosaka Stefan Schaal Tomohiro Shibata Gaurav Tevatia Ales Ude Sethu Vijayakumar Frank Pollick, Marcia Riley, and Mitsuo Kawato. Using humanoid robots to study human behavior. *IEEE Intelligent Systems*, 15(4):46–56, 2000.

[Bar81] A.H. Barr. Superquadrics and angle-perserving transformations. *IEEE Comput. Graphics Applicat.*, 1:11–23, 1981.

[BDK98] H. Bruyninckx, S. Demey, and V. Kumar. Generalized stability of compliant grasps. *In Proceedings of IEEE International Conference on Robotics and Automation*, page 23962402, 1998.

[BFH99] Ch. Borst, M. Fischer, and G. Hirzinger. A fast and robust grasp planner for arbitrary 3d objects. *In Proceedings of IEEE International Conference on Robotics and Automation*, page 18901896, 1999.

[BFH03] Ch. Borst, M. Fischer, and G. Hirzinger. Grasping the dice by dicing the grasp. *In Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2003.

[BFH04] Ch. Borst, M. Fischer, and G. Hirzinger. Grasp planning: How to choose a suitable task wrench space. *In Proceedings of IEEE International Conference on Robotics and Automation*, 2004.

[Bic95] A. Bicchi. On the closure properties of robotic grasping. *International Journal of Robotics Research*, 14:(4):319–334, 1995.

[Bie87] I. Biederman. Recognition-by-components: A theory of human image understanding. *Psychological Review*, 94:115–147, 1987.

[Bin71] T.O. Binford. Visual perception by computer. *IEEE Conference on Systems and Control*, 1971.

[Bis95] C.M. Bishop. Neural networks for pattern recognition. *Oxford University Press*, 1995.

[BSZ08] B. Bounab, D. Sidobre, and A. Zaatri. Central axis approach for computing n-finger force-closure grasps. *In Proceedings of IEEE International Conference on Robotics and Automation*, 2008.

[CG06] L. Chen and N.D. Georganas. An efficient and robust algorithm for 3d mesh segmentation. *Multimedia Tools Appl.*, 29(2):109–125, 2006.

[CJ91] U. Castiello and M. Jeannerod. Measuring time to awareness. *Neuroreport*, 12:787800, 1991.

[CJB03] L. Chevalier, F. Jaillet, and A. Baskurt. Segmentation and superquadric modeling of 3d objects. *Journal of WSCG*, 11(1), 2003.

[Cor03] J. Cortes. Motion planning algorithms for general closed-chain mechanisms. *Thése de doctorat de l'Institut National Polytechnique de Toulouse*, 2003.

[Cra73] H. Crapo. A combinatorial perspective on algebraic geometry. *Colloquio Int. sulle Teorie Combinatorie*, 1973.

[CTSO03] D.Y. Chen, X.P. Tian, Y.T. Shen, and M. Ouhyoung. On visual similarity based 3d model retrieval. *Computer Graphics Forum (EUROGRAPHICS'03)*, 22(3):223–232, 2003.

[CW86] M. Cutkosky and P. Wright. Modeling manufacturing grips and correlations with the design of robotic hands. *In Proceedings of IEEE International Conference on Robotics and Automation*, pages 1533–1539, 1986.

[Dan84] A. Dandurand. The rigidity of compound spatial grid. *Structural topology*, 10, 1984.

[DLSX00] D. Ding, Y. Liu, Y.T. Shen, and G.L. Xiang. An efficient algorithm for computing a 3d form-closure grasp. *In Proceedings of IEEE International Conference on Robotics and Automation*, page 12231228, 2000.

[DLW00] D. Ding, Y. Liu, and S. Wang. Conmputing 3-d optimal form-closure grasps. *In Proceedings of IEEE International Conference on Robotics and Automation*, page 35733578, 2000.

[DLW01] D. Ding, Y. Liu, and S. Wang. On computing inmobilizing grasps of 3-d curved objects. *Proc. IEEE Int. Symp. on Computational Intelligence in Robotics and Automation*, pages 11–16, 2001.

[DPR92] S.J. Dickson, A.P. Pentland, and A. Resenfeld. From volumes to views: An approach to 3d object recognition. *CVGIP: Image Understanding*, 55(2):130–154, 1992.

[EK04] S. Ekvall and D. Kragic. Interactive grasp learning based on human demonstration. *In Proceedings of IEEE/RSJ International Conference on Robotics and Automation*, 2004.

[EK07] S. Ekvall and D. Kragic. Learning and evaluation of the approach vector for automatic grasp generation and planning. *In Proceedings of IEEE/RSJ International Conference on Robotics and Automation*, page 47154720, 2007.

[FC92] C. Ferrari and J. Canny. Planning optimal grasps. *In Proceedings of IEEE International Conference on Robotics and Automation*, 1992.

[FdSH98] M. Fischer, P. Van der Smagt, and G. Hirzinger. Learning techniques in a dataglove based telemanipulation system for the dlr hand. *In Proceedings of IEEE International Conference on Robotics and Automation*, 1998.

[FH97] M. Fischer and G. Hirzinger. Fast planning of precision grasps for 3d objects. *In Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, page 120126, 1997.

[FL88] F.P. Ferrie and M.D. Levine. Deriving coarse 3d models of objectsn. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 345–353, 1988.

[GALP07] C. Goldfeder, P.K. Allen, C. Lackner, and R. Pelossof. Grasp planning via decomposition trees. *In Proceedings of IEEE International Conference 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(3):302–326, 1993.

[GMBR94] M.A. Goodale, J.P. Meenan, H.H. Blthoff, and C.I. Raciot. 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*, pages 349:154–156, 1991.

[Gra93] A. Gray. The gaussian and mean curvatures. *Modern Differential Geometry of Curves and surfaces*, pages 279–285, 1993.

[GT56] A.J. Goldman and A.W. Tucker. Polyhedral convex cones. *Princeton University Press*, 1956.

[HBZ06] M. Hueser, T. Baier, and J. Zhang. Learning of demonstrated grasping skills by stereoscopic tracking of human hand configuration. *In Proceedings of IEEE International Conference on Robotics and Automation*, 2006.

[HK96] W.S. Howard and V. Kumar. On the stability of grasped objects. *IEEE Transactions on Robotics and Automation*, 12(6):904917, 1996.

[HSSR05] R. Haschke, J.J. Steil, I. Steuwer, and H. Ritter. Task-oriented quality measures for dextrous grasping. *Proceedings IEEE International Symposium on Computational Intelligence in Robotics and Automation, CIRA*, pages 689–694, 2005.

[HZ08] M. Hueser and J. Zhang. Visual and contact-free imitation learning of demonstrated grasping skills with adaptive environment modelling. *In Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, WS on Grasp and Task Learning by Imitation*, 2008.

[Jea88] M. Jeannerod. The neural and behavioral organization of goal directed movements. *Oxford, Clarendon Press*, 1988.

[KCS94] D. Keren, D. Cooper, and J. Subrahmonia. Describing complicated objects by implicit polynomials. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(1):38–52, 1994.

[KHGB95] S. Kumar, S. Han, D. Goldgof, and K. Bowyer. On recovering hyperquadrics from range data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(11):1079–1083, 1995.

[KKD97] P.R. Kraus, V.I. Kumar, and P. Dupont. Analysis of frictional contact models for dynamic symulation. *In Proceedings of IEEE International Conference on Robotics and Automation*, 1997.

[KMY92] D. Kirkpatrick, B. Mishra, and C. Yap. Quantitative steinitzs theorem with applications to multi-fingered grasping. *Discr. Comput. Geom.*, 7:(3):295–318, 1992.

[KOYS01] B. Kim, S. Oh, B. Yi, and I.H. Suh. Optimal grasping based on non-dimensionalized performance indices. *In Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 949–956, 2001.

[KS08a] S. El Khoury and A. Sahbani. Handling objects by their handles. *IEEE/RSJ International Conference on Intelligent Robots and Systems, WS - grasp and task learning by imitation*, pages 58–64, 2008.

[KS08b] S. El Khoury and A. Sahbani. A sufficient condition for computing n-finger force-closure grasps of 3d objects. *IEEE International Conference on Robotics, Automation and Mechatronics*, 2008.

[KSP07] S. El Khoury, A. Sahbani, and V. Perdereau. Learning the natural grasping component of an unknown object. *In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2957–2962, 2007.

[KWSN05] F. Kyota, T. Watabe, S. Saito, and M. Nakajima. Detection and evaluation of grasping positions for autonomous agents. *In International Conference on Cyberworlds*, page 453460, 2005.

[Lak78] K. Lakshminarayana. Mechanics of form-closure. *Technical Report 78-DET-32, ASME*, 1978.

[LDSA05] E. Lopez-Damian, D. Sidobre, and R. Alami. Grasp planning for non-convex objects. *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*, page 19041909, 1999.

[LFP07] Y. Li, J.L. Fu, and N. Pollard. Data-driven grasp synthesis using shape matching and task-based pruning. *IEEE Transactions on Visualization and Computer Graphics*, 13:(4):732–747, 2007.

[Liu99] Y.H. Liu. Qualitative test and force optimization of 3d frictional form closure grasps using linear programming. *IEEE Transactions on Robotics and Automation*, 15:(1), 1999.

[Liu00] Y.H. Liu. Computing n-finger form-closure grasps on polygonal objects. *International Journal of Robotics Research*, 19:(2):149158, 2000.

[LJS97] A. Leonardis, A. Jaklic, and F. Solina. Superquadrics for segmentation and modeling range data. *IEEE Transactions on Pattern Anal Mach Intell.*, 19:12891295, 1997.

[LLC03] J.W. Li, H. Liu, and H.G. Cai. On computing three-finger force-closure grasps of 2d and 3d objects. *IEEE Transactions on Robotics and Automation*, 19:(1), 2003.

[LLD04] Y.H. Liu, M.L. Lam, and D. Ding. A complete and efficient algorithm for searching 3-d form closure grasps in the discrete domain. *IEEE Transactions on Robotics and Automation*, 20:(5):805816, 2004.

[LS88] Z. Li and S.S. Sastry. Task-oriented optimal grasping by multifingered robot hands. *IEEE Journal of Robotics and Automation*, 4:(1), 1988.

[LXL04] G. Liu, J. Xu, and Z. Li. On geometric algorithms for real-time grasping force optimization. *IEEE Transactions on Control Systems Technology*, 12(6):843859, 2004.

[MC94] B. Mirtich and J. Canny. Easily computable optimum grasps in 2d and 3d. *In Proceedings of IEEE International Conference on Robotics and Automation*, 1:739–747, 1994.

[Mil01] A.T. Miller. Graspit!: A versatile simulator for robotic grasping. *PhD thesis, Columbia University*, 2001.

[MKAC03] A.T. Miller, S. Knoop, P.K. Allen, and H.I. Christensen. Automatic grasp planning using shape primitives. *In Proceedings of IEEE International Conference on Robotics and Automation*, 2003.

[MLS94] R.M. Murray, Z. Li, and S.S. Sastry. A mathematical introduction to robotic manipulation. *Orlando, FL: CRC*, 1994.

[MN78] D. Marr and H.K. Nishihara. Representation and recognition of three dimensional shapes. *Proceedings of the Royal Society of London*, Series B. 200:269–294, 1978.

[MNT04] K. Madsen, H. Nielsen, and O. Tingleff. Methods for non-linear least squares problems. *Technical University of Denmark*, 2004.

[Mon91] D.J. Montana. The condition for contact grasp stability. *In Proceedings of IEEE International Conference on Robotics and Automation*, pages 412–417, 1991.

[MP77] R.S. Millman and G.D. Parker. Elements of differential geometry. *Prentice-Hall Inc*, 1977.

[MP89] X. Markenscoff and C.H. Papadimitriou. Optimum grip of a polygon. *International Journal of Robotics Research*, 8:(2):17–29, 1989.

[MRPD04] C. Michel, C. Rmond, V. Perdereau, and M. Drouin. A robotic grasping planner based on the natural grasping axis. *In Proceedings of the IEEE International Conference on Intelligent Manipulation and Grasping*, 2004.

[MSS87] B. Mishra, J.T. Schwartz, and M. Sharir. On the existence and synthesis of multifinger positive grips. *Algorithmica, Special Issue: Robotics*, 2:541–558, 1987.

[MW99] A. Mangan and R. Whitaker. Partitioning 3d surface meshes using watershed segmentation. *IEEE Trans Vis Comput Graph*, 5(4):308321, 1999.

[Nap56] J. Napier. The prehensile movements of the human hand. *Journal of Bone and Joint Surgery*, 38:B(4):902–913, 1956.

[Ngu87] V.D. Nguyen. Constructing stable grasps in 3d. *In Proceedings of IEEE International Conference on Robotics and Automation*, pages 234–239, 1987.

[NS07] N. Niparnan and A. Sudsang. Positive span of force and torque components of four-fingered three-dimensional force-closure grasps. *In Proceedings of IEEE International Conference on Robotics and Automation*, 2007.

[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.

[Pen86] A.P. Pentland. Perceptual organization and the representation of natural form. *Artificial Intelligence*, 28:293–331, 1986.

[PF95] J. Ponce and B. Faverjon. On computing three finger force closure grasp of polygonal objects. *IEEE Transactions on Robotics and Automation*, 11:(6):868881, 1995.

[PMAT04] R. Pelossof, A. Miller, P. Allen, and T.Jebara. An svm learning appraoch to robotic grasping. *In Proceedings of IEEE International Conference on Robotics and Automation*, 2004.

[Pol97] N.S. Pollard. Parallel algorithms for synthesis of whole-hand grasps. *In Proceedings of IEEE International Conference on Robotics and Automation*, 1997.

[PRF02] S. Pulla, A. Razdan, and G. Farin. Improved curvature estimation for watershed segmentation of 3-dimensional meshes. *IEEE Transactions Vis Comput Graph*, 2002.

[PSBM93] J. Ponce, S. Sullivan, J.D. Boissonnat, and J.P. Merlet. On characterizing and computing three- and four-finger force-closure grasps of polyhedral objects. *In Proceedings of IEEE International Conference on Robotics and Automation*, pages 821–827, 1993.

[PSdP07] M. Prats, P.J. Sanz, and A.P. del Pobil. Task-oriented grasping using hand preshapes and task frames. *In Proceedings of IEEE International Conference on Robotics and Automation*, 2007.

[PSSM97] J. Ponce, S. Sullivan, A. Sudsang, and J.P. Merlet. On computing four-finger equilibrium and force-closure grasps of polyhedral objects. *International Journal of Robotics Research*, 16:(1):1135, 1997.

[RB02] A. Razdan and M. Bae. A hybrid approach to feature segmentation of 3-dimensional meshes. *Computer-Aided Design*, 2002.

[Reu63] F. Reuleaux. Kinematics of machinery. *(first published in German), 1875. Reprinted by Dover, New York*, 1963.

[RJ94] N.S. Raja and A.K. Jain. Obtaining generic parts from range data using a multi-view representation. *CVGIP: Image Understanding*, 60(1):44–64, 1994.

[RKK08] J. Romero, H. Kjellstrm, and D. Kragic. Human-to-robot mapping of grasps. *In Proceedings of 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. *Smart Graphics, AAAI Spring Symposium, Stanford University*, pages 71–75, 2000.

[RR93] P. Robert and A. Roux. Influence of the shape of the tetrahedron on the accuracy of the estimate of the current density. *Proceedings of ESA START Conference*, 1993.

[Sau07] J.P. Saut. Planification de mouvement pour la manipulation dextre d'objets rigides. *Thése de doctorat de l'Universite Pierre et Marie Curie*, 2007.

[SB90] F. Solina and R. Bajcsy. Recovery of parametric models from range images: the case of superquadrics with global deformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(2):131–147, 1990.

[SDKN08] A. Saxena, J. Driemeyer, J. Kearns, and A.Y. Ng. Robotic grasping of novel objects using vision. *The International Journal of Robotics Research*, 27(2):157–173, 2008.

[SLM94] F. Solina, A. Leonardis, and A. Macerl. A direct part-level segmentation of range images using volumetric models. *In Proceedings of IEEE International Conference on Robotics and Automation*, pages 2254–2259, 1994.

[SLZS08] M. Stark, P. Lies, M. Zillich, and B. Schiele. Functional object class detection based on learned affordance cues. *Computer Vision Systems*, pages 435–444, 2008.

[SMHM84] L.G. Shapiro, J.D. Moriarty, R.M. Haralick, and P.G. Mulgaonkar. Matching three-dimensional objects using a relational paradigm. *Pattern Recognition*, 17(4):385–405, 1984.

[SMKF04] P. Shilane, P. Min, M. Kazhdan, and T. Funkhouser. The princeton shape benchmark. *In Proceedings of Shape Modelling International*, 2004.

[Som00] P. Somoff. Uber gebiete von schraubengeschwindigkeiten eines starren korpers bie verschiedener zahl von stutz achen. *Zeitschrift fur Mathematik und Physik*, 45:245–306, 1900.

[SR82] J.K. Salisbury and B. Roth. Kinematic and force analysis of articulated hands. *ASME J. Mech., Transmissions, Automat.,Design*, 105:33–41, 1982.

[SRC06] R. Suarez, M. Roa, and J. Cornella. Grasp quality measures. *Technical Report, Universitat Politecnica De Catalunya*, 2006.

[SZL92] W. Schroeder, J. Zarge, and W. Lorensen. Decimation of triangle meshes. *Proc. SIGGRAPH, Computer Graphics*, 25(3):6570, 1992.

[Thr02] S. Thrun. Robotic mapping: a survey. *Exploring Artificial Intelligence in the New Millennium*, 2002.

[UM82] L.G. Ungerleider and M. Mishkin. Two cortical visual systems. *MIT Press*, pages 549–585, 1982.

[VY10] O. Veblen and J.W. Young. Projective geometry. *the Athenaeum press*, 1910.

[WD08] G. Walck and M. Drouin. Reconstruction 3d progressive et rapide. *MajecStic08*, 2008.

[WL95] Kenong Wu and Martin D. Levine. Segmenting 3d objects into geons. In *ICIAP*, pages 321–334, 1995.

[WL97] K. Wu and M.D. Levine. 3d part segmentation using simulated electrical charge distributions. *IEEE Transactions On Pattern Analysis and Machine Intelligence*, 19 No. 11:1223–1235, 1997.

[WZG95] R. Wagner, Y. Zhuang, and K. Goldberg. Fixturing faceted parts with seven modular struts. *IEEE International Symposium on Assembly and Task Planning*, 1995.

[ZBD95] P.D. Zavattieri, G.C. Buscaglia, and E.A. Dari. Finite element mesh optimization in three dimensions. 1995.

[ZD04] X. Zhu and H. Ding. Planning force-closure grasps on 3-d objects. *In Proceedings of IEEE International Conference on Robotics and Automation*, 2004.

[Zor60] W. Zorach. Zorach explains sculpture: What it means and how it is made. *Tudor Publishing Company*, 1960.

[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. *International Conference on Image Processing*, (3):273276, 2002.

[ZW03] X. Zhu and J. Wang. Synthesis of force-closure grasps on 3d objects based on the q distance. *IEEE Transactions on Robotics and Automation*, 19:(3), 2003.

Appendix A

Superquadrics Formulation

A.1 The Implicit Equation for Basic Shapes

A superquadric surface model is defined by the following implicit equation:

$$f(x,y,z) = \left(\left(\frac{x}{a_1} \right)^{\frac{2}{\epsilon_2}} + \left(\frac{y}{a_2} \right)^{\frac{2}{\epsilon_2}} \right)^{\frac{\epsilon_2}{\epsilon_1}} + \left(\frac{z}{a_3} \right)^{\frac{2}{\epsilon_1}} = 1 \tag{A.1}$$

Where:

- $a_1,\ a_2$ and a_3, define the superquadric size;
- ϵ_1 and ϵ_2, determine the shape curvatures that define a smoothly changing family of shapes from rounded to square.

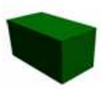
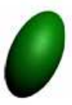
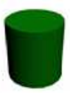
	ε_1	ε_2
	0.1	0.1
	1	1
	0.1	1

Figure A.1: Basic superquadrics.

This compact model of superquadrics, defined by only five parameters, can model a large set of building blocks like spheres, cylinders and boxes (Fig. A.1). When both ϵ_1 and ϵ_2 are 1, the surface vector defines an ellipsoid or, if $a_1,\ a_2$, and a_3 are all equal a sphere. For example, the implicit equation for an ellipsoid ($\epsilon_1 = \epsilon_2 = 1$) is as follows:

$$f(x,y,z) = \left(\frac{x}{a_1}\right)^2 + \left(\frac{y}{a_2}\right)^2 + \left(\frac{z}{a_3}\right)^2 = 1 \tag{A.2}$$

When $\epsilon_1 \ll 1$ and $\epsilon_2 = 1$, the superquadric surface is shaped like a cylinder. Boxes are produced when both ϵ_1 and ϵ_2 are $\ll 1$.

A.2 The Implicit Equation for Tapered Shapes

Two assumptions are made regrading the tapering formulation:

- Tapering deformation is performed along the z axis.
- The tapering rate is linear with z.

The linearity assumption is sometimes violated for real objects. This violation is acceptable since we need only to approximate the shape of the tapered object parts.

Tapering is defined by two parameters k_x and k_y and tapering deformation along z axis is given by:

$$\begin{aligned} X &= f_x(z)x \\ Y &= f_y(z)y \\ Z &= z \end{aligned} \tag{A.3}$$

X, Y, Z are the components of the surface vector X of the deformed superquadric. f_x and f_y are the tapering functions in the x and the y axis of the object centered coordinate system. x, y, z are the components of the original non-deformed surface vector x. The two tapering functions are:

$$\begin{aligned} f_x(z) &= \frac{k_x}{a_3}Z + 1 \\ f_y(z) &= \frac{k_y}{a_3}Z + 1 \end{aligned} \tag{A.4}$$

where:

$$-1 \leq k \leq 1 \tag{A.5}$$

The equation of inverse tapering is given by:

$$\begin{aligned} x &= \frac{X}{\frac{k_x}{a_3}Z + 1} \\ y &= \frac{Y}{\frac{k_y}{a_3}Z + 1} \end{aligned} \tag{A.6}$$

The implicit equation of a tapered shape is written using equations (A.1) and (A.6) as follows:

$$f(x,y,z) = \left(\left(\frac{X}{a_1(\frac{k_x}{a_3}Z+1)} \right)^{\frac{2}{\epsilon_2}} + \left(\frac{Y}{a_2(\frac{k_y}{a_3}Z+1)} \right)^{\frac{2}{\epsilon_2}} \right)^{\frac{\epsilon_2}{\epsilon_1}} + \left(\frac{z}{a_3} \right)^{\frac{2}{\epsilon_1}} = 1 \qquad \text{(A.7)}$$

A.3 The Implicit Equation for Curved Shapes

Bending is defined with the two parameters k and α. k is the curvature parameter and α determines the bending plane. Knowing these two parameters, the bending angle, γ, can be easily computed. the bending is also performed along the z axis and transforms vectors (x,y,z) into vectors (X,Y,Z).

The bending is performed by projecting the x and y components of all points onto the bending plane, performing the bending deformation in that plane, and then projecting the points back to the original plane. The bending plane is defined by coordinate axis z and the angle α. The transformed surface vector is given by:

$$\begin{aligned} X &= x + \cos(\alpha)(R - r) \\ Y &= y + \sin(\alpha)(R - r) \\ Z &= \sin(\gamma)(k^{-1} - r) \end{aligned} \qquad \text{(A.8)}$$

where:

- r is the projection of a point (x, y) on the bending plane;
- γ is the bending angle;
- k is the curvature parameter;
- Bending transforms r into R.

$$\begin{aligned} \gamma &= \arctan \frac{Z}{k^{-1} - R} \\ r &= k^{-1} - \sqrt{Z^2 + (k^{-1} - R)^2} \\ R &= \cos\left(\alpha - arctan \frac{Y}{X} \right) \sqrt{X^2 + Y^2} \end{aligned} \qquad \text{(A.9)}$$

Thus, the implicit equation of a bent shape is written as follows:

$$f(x, y, z) = \left(\left(\frac{X - \cos(\alpha)(R - r)}{a_1} \right)^{\frac{2}{\epsilon_2}} + \left(\frac{Y - \sin(\alpha)(R - r)}{a_2} \right)^{\frac{2}{\epsilon_2}} \right)^{\frac{\epsilon_2}{\epsilon_1}} + \left(\frac{z}{a_3} \right)^{\frac{2}{\epsilon_1}} = 1 \quad \text{(A.10)}$$

Appendix B

Behind the Scenes: the Training Data Choice

Our learning algorithm uses 12 objects as training data. These paragraphs aim at explaining the whole process that conducts to these objects. We remind the reader that our training objects are the result of the assembly of two superquadrics. Therefore, the choice of the training objects should effectively sub-sample the space of two superquadrics assembly. These volumetric primitives are described by 8 parameters each ($\lambda = e_1$, e_2, k_x, k_y, γ, a_1, a_2 and a_3). Thus, sub-sampling superquadrics space induces sub-sampling these parameters.

B.1 Sub-Sampling Superquadrics Shapes and Sizes

In order to have a manageable number of superquadrics shapes, we have chosen 7 representative models that span the space of superellipsoids by choosing ϵ_1 and ϵ_2 to be one of 0.1 or 1 and by adding tapering and bending deformations. The 7 superquadrics shapes are the following: *cylinder, box, ellipsoid, tapered cylinder, curved cylinder, tapered box and curved box*. These shapes are described by the 5 parameters ($\lambda 1 = e_1$, e_2, k_x, k_y and γ). The remaining 3 parameters ($\lambda_2 = a_1$, a_2 and a_3) describe the superquadric size. Thus, to span superquadrics sizes, one needs to vary these parameters. Conisider for example figure (B.1). This figure illustrates three different cylinders, thus λ_1 is the same. On the other hand,

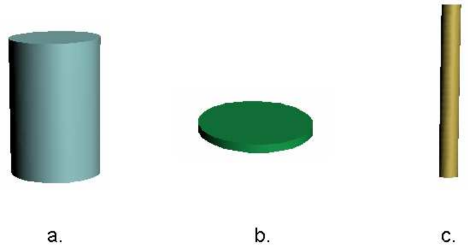


Figure B.1: Variable sizes of a cylinder: (a) cylinder obtained when the three dimensions a_1, a_2 and a_3 are significant, (b) flat cylinder obtained when only two dimensions are significant (a_1 and $a_2 \gg a_3$) and (c) elongated cylinder obtained when only one dimension is significant ($a_3 \gg a_1$ and a_2).

the first cylinder is obtained when the three dimensions a_1, a_2 and a_3 are significant. The second cylinder is a flat one and is obtained when only two dimensions are significant. The third cylinder is an elongated one and it is the case when only one dimension is significant. By varying sizes parameters of the seven previously defined shapes, we obtain 21 (7×3) volumetric primitives that effectively span superquadrics shapes and sizes space. But only 8 of these primitives are relevant to grasping (Fig. B.2). Many volumetric primitives such as elongated cylinders, elongated boxes, elongated ellipsoids can be merged to one primitive since the size information in these cases is more relevant for grasping than the shape. Our training objects will be represented as an assembly of these 8 primitives (Fig. B.2).

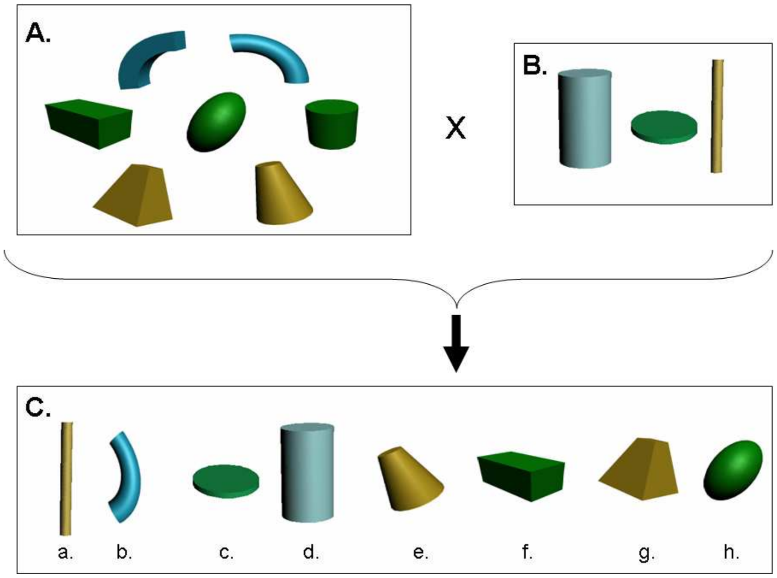


Figure B.2: (A) Illustrates sub-sampling of deformed superquadrics shapes. (B) Shows sub-sampling of a cylinder size space. (B) Presents primitives used for sub-sampling superquadrics shapes and sizes.

B.2 Two Superquadrics Assembly: Theoretical Vs. Real Combinations Number

The training objects are the result of the assembly of two volumetric primitives. These primitives are chosen among the 8 shapes described above (Fig. B.2). Thus the total number of two superquadrics combinations is: $8 + 7 + ... + 1 = 4 \times 9 = 36$. But another aspect should be taken into account when assembling primitives: the relative size. For example, three different combinations of a cylinder and an elongated cylinder are possible (Fig. B.3). The first assembly occurs when the two primitives are of comparable sizes. The second is obtained when a primitive is relatively small in respect to the second one. The inverse case gives the last combination. Consequently, the number of possible combinations is no longer 36, but $3 \times 36 = 108$. This number is not final yet. When assembling two similar primitives, only one case is meaningful when a primitive is smaller than the other. Thus, the theoretical number of possible combinations is: $108 - (8 \times 2) = 92$. In practice, we search for two primitives assembly corresponding to real objects since our learning algorithm imitates humans choice of the natural grasping component. From this point of view, 12 objects were sufficient to span the space of two-superquadrics assembly of real objects.

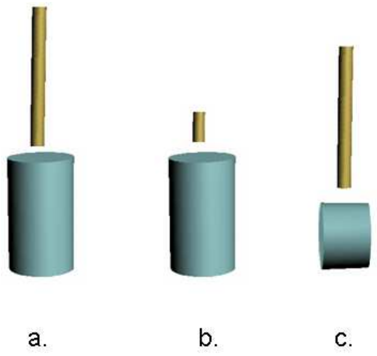


Figure B.3: Three possible combinations of two volumetric primitives: a cylinder and an elongated cylinder. (a) A screwdriver obtained when the corresponding primitives relative sizes are comparable. (b) A bottle obtained when the elongated cylinder is small is respect of the cylinder size. (c) A hammer obtained in the inverse case of (b).

Annexe C

Résumé des travaux

La robotique a comme objectif le développement de robots autonomes. Par autonome, on sous-entend qu'un robot est capable de percevoir son environnement, de raisonner pour accomplir une mission et d'agir sur ce même environnement. La mission à laquelle nous nous intéressons, dans cette thèse, est la saisie d'objets. C'est l'une des fonctions les plus complexes à réaliser par un système robotisé à cause du grand nombre de degrés de liberté qui entrent en jeu. La prise initiale représente une phase capitale de la manipulation d'un objet et conditionne fortement le succès du travail demandé. Par conséquence, la question à poser est la suivante : quelles sont les contraintes que doit satisfaire cette prise initiale afin que la manipulation de l'objet réussisse ? Dans la littérature, nous pouvons identifier trois principales contraintes à respecter par une stratégie de prise. La première est la stabilité ou en d'autres termes que la main tienne fermement l'objet. La deuxième contrainte est la compatibilité avec la tâche ou l'adaptabilité de la prise à la fonction de l'objet. Finalement, comme les objets que nous utilisons sont de formes et de tailles très diverses, une stratégie de prise doit être adaptée à de nouveaux objets.

Plusieurs travaux de recherche se sont intéressés au problème de la génération automatique de prise. Le premier pas dans ce domaine était de trouver pour un objet donné une prise stable. Différentes approches analytiques ont été développées pour cette fin. Vu le grand nombre de degrés de liberté mis en jeu, ces approches nécessitent un temps de calcul considérable et ne garantissent aucune adaptabilité à la tâche. En effet, les chercheurs se sont heurtés à la complexité de la modélisation d'une tâche et étaient même ramenés, dans plusieurs travaux, à calculer à l'avance la trajectoire de l'objet pour être capable de la modéliser. Ensuite, ils devaient trouver la meilleure prise adaptée à la tâche. Ceci nécessite d'une part un calcul complexe. D'une autre part, ces approches restent limitées à un seul objet et à une seule tâche. Pour éviter la complexité des approches analytiques, les approches empiriques ont été introduites au problème de la saisie des objets. Ces approches tentent de résoudre le problème par imitation du comportement humain en utilisant des algorithmes d'apprentissage. Ces méthodes se divisent en deux catégories : les approches basées sur l'apprentissage par imitation du comportement humain et les approches qui s'intéressent plutôt à l'apprentissage des caractéristiques des objets. Pour les premières, le robot observe un opérateur humain en train de saisir un objet et essaie de reproduire le même geste de préhension.

Ces méthodes permettent la télémanipulation des objets ou la reconnaissance du geste de préhension, comme elles observent la phase d'approche. Mais, une fois affrontée à un nouvel objet, ces méthodes n'arrivent pas également à déterminer une prise adaptée à la tâche. Les approches focalisant sur les caractéristiques de l'objet permettent d'associer différentes formes de la main avec des surfaces locales des objets. D'autres méthodes apprennent à identifier, dans une image, des zones de prise. Ces approches trouvent une correspondance entre la forme de l'objet et différentes prises. D'où, elles sont adaptées aux nouveaux objets. Le problème de ces approches est qu'elles trouvent pour un seul objet plusieurs prises. Elles ne permettent pas alors de choisir la prise compatible à la tâche. Pour ceci, ces approches nécessitent l'intervention d'un opérateur humain ou de la modélisation de la tâche. En d'autres termes, trouver pour un nouvel objet une prise qui lui est adaptée en terme de tâche reste un problème ouvert.

Le thème de génération de prise n'est pas nouveau pour notre équipe de recherche. Un ancien thésard de l'équipe, Cédric Michel, a développé une méthode analytique pour la génération de prise. La méthode permet de trouver à partir de l'enveloppe géométrique de l'objet, des axes naturels de préhension. Ces axes caractérisent la forme de la main humaine durant la phase d'approche pour saisir un objet. Cet algorithme trouve pour un seul objet plusieurs axes de préhension mais ne sélectionne pas celui qui convient le mieux à la tâche. Ceci nous a conduit à la conviction que les méthodes analytiques ne sont pas suffisantes pour satisfaire les contraintes nécessaires à la résolution du problème. Ce travail de thèse propose une stratégie de prise combinant à la fois les deux approches : analytique et empirique. A partir de l'observation de la saisie humaine, un réseau de neurones sera entraîné afin d'imiter le choix de la partie préhensible, associée à chaque objet. En second lieu, nous développons une méthode pour calculer les points de contact sur la partie saisissable de l'objet.

C.1 Saisir Par Composantes - Le Concept

Notre intérêt se porte sur la détermination d'une prise adaptée à un objet donné, à partir de sa forme géométrique et des contraintes fixées par la tâche. L'idée de notre approche est issue de l'observation du comportement humain lors de la saisie d'objets. Pour expliquer la faculté de saisie exceptionnelle humaine, nous nous inspirons d'une théorie proposée en neuroscience, intitulée reconnaissance par composantes. Cette théorie suppose que les objets sont découpés mentalement selon leurs parties naturelles et que l'assemblage de ces différentes parties servira de clef pour leur identification. Si l'homme découpe les objets en primitives géométriques simples afin de les reconnaître, pourquoi n'emploie-t-il pas cette décomposition pour les saisir ? Nous avons une forte conviction que plusieurs objets de la vie courante sont munis, à leur fabrication, d'une partie facilitant leur préhension. L'approche développée détermine, pour un objet de forme quelconque, cette partie préhensible. L'objet est alors approché par un ensemble de formes géométriques simples (les superquadriques). A partir de l'observation de la saisie humaine, un réseau de neurones est entraîné afin d'identifier la partie préhensible de l'objet. Les paragraphes suivants détaillent toutes ces étapes. Nous commencons par présenter la théorie de reconnaissance par composantes.

C.1.1 Théorie de Reconnaissance Par Composantes

Cette théorie propose que la reconnaissance des objets repose sur la perception d'éléments géométriques de base à partir desquels on peut construire un objet. Ces éléments sont nommés géons , contraction de geometric ion. Ainsi, un objet quelconque sera segmenté en plusieurs composantes en exploitant ses propriétés locales telle que la concavité. Chaque composante est ensuite identifiée à un géon. L'agencement des différents géons permet alors l'identification des objets. La figure (C.1) illustre un objet non-familier. En decomposant cet objet en parties, nous nous rendons compte qu'il ressemble à un chariot à hot-dog. Le grand compartiment étant l'endroit où sont cuits les saucisses. La partie circulaire en bas pourrait être la roue. La partie courbée sur le côté a la même forme qu'une poignée. Sur le compartiment, nous avons une partie qui ressemble au support d'un parasol et l'autre à un extracteur de jus. Par conséquence, même les objets non-familiers peuvent être identifiés en les découpant en composantes. Mais comment pourrait-on saisir des objets non-familiers ou quelconques ?

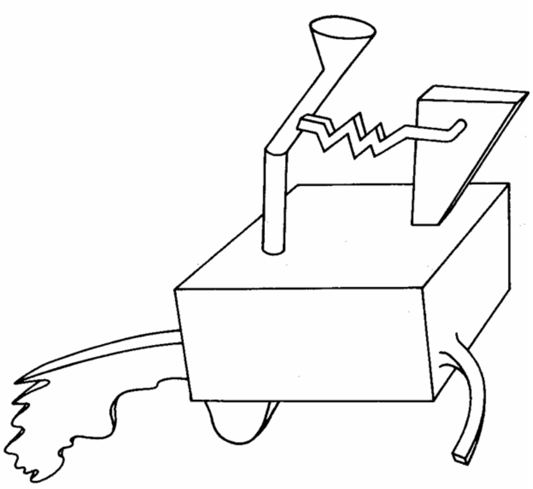


Fig. C.1 – L'objet non-familier de Biederman [Bie87].

C.1.2 Saisie Par Composantes

Le paragraphe précédent montre que la représentation des objets par un assemblage de formes géométriques simples permet leur identification. En effet, cette représentation par composantes nous incite à remarquer que nous saisissons de la même manière plusieurs objets de la vie courante constitués des sous-parties similaires. Considérons par exemple les trois objets de la figure (C.2). Ces trois formes approximent respectivement une tasse, un sceau et un sac. Ces derniers sont constitués de deux composantes : un cylindre et un cylindre courbé. Malgré l'arrangement différent de ces deux parties, les trois objets sont saisis de la même manière, par leur partie courbée.

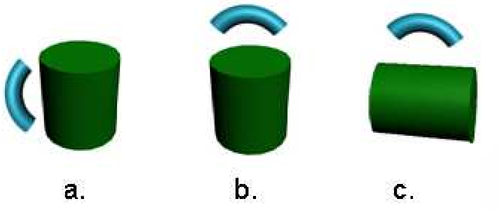


Fig. C.2 – Le choix de la partie préhensible dépend de la forme des sous-parties de l'objet. Ce choix est moins influencé par la disposition relative de ces composantes, i.e : a) une tasse, b) un sceau et c) un sac sont tous saisis par leur partie courbée.

Un autre aspect important dans le choix de la partie préhensible est la taille relative des différentes composantes. Considérons l'exemple des trois verres suivants (Fig. C.3) : un verre à vin, une flûte à Champagne et un verre à cognac. Ces trois verres sont constitués des sous-parties similaires : le calice, la jambe ou la tige et le pied. Malgré la similitude de leurs composantes, ces verres sont saisies de manières différentes. Nous remarquons qu'un verre à vin réclame un calice large permettant aux arômes profonds de librement s'épanouir. Les flûtes à champagne sont caractérisées par un calice étroit et allongé mettant le mieux en valeur l'effervescence de ce dernier. Une coupe trop ouverte laisse échapper les bulles ainsi que leur arôme. Ces deux verres ont une longue tige permettant de les tenir sans poser les doigts sur le calice. Non seulement ceci évite de tâcher la coupe de graisse et d'amoindrir la netteté du verre ou du cristal, tenir le verre par la tige est aussi essentiel pour préserver la bonne température du vin. Le contact avec la main peut réchauffer rapidement un vin. D'une autre part, le verre à cognac se distingue par sa jambe courte. La main se positionne ainsi sous le calice du verre pour réchauffer naturellement l'alcool. Cet exemple montre à la fois l'influence de la taille relative des composantes sur le choix de la partie préhensible et que les objets sont fabriqués d'une manière facilitant leur prise.

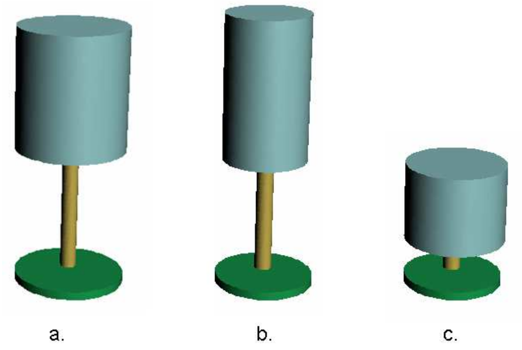


Fig. C.3 – Une représentation approximative de : a) un verre à vin, b) une flûte à Champagne et c) un verre à cognac.

L'approche proposée utilisera alors les informations sur la forme et la taille des différentes composantes consritutants l'objet afin de déterminer, en imitant le comportement humain, la partie préhensible qui lui correspond.

C.2 Saisir Par Composantes - L'Approche Proposée

Nous proposons une nouvelle stratégie de prise permettant de trouver, pour un objet quelconque, une prise qui lui est adaptée en terme de fonctionalité. L'idée à la base de notre algorithme est la suivante : les objets de la vie courante sont fabriqués de manière à faciliter leur prise. Les tasses ont des parties courbées pour les saisir : leurs anses. Les bouteilles ont une partie allongée facilitant leur préhension. Quelle serait alors la partie préhensible d'un objet quelconque ? La réponse à cette question nous conduira vers la détermination d'une prise de l'objet compatible avec la tâche à laquelle il est destiné. Notons que cette approche permet d'éviter la complexité de la modélisation de la tâche. Notre approche est illustrée dans la figure (C.4).

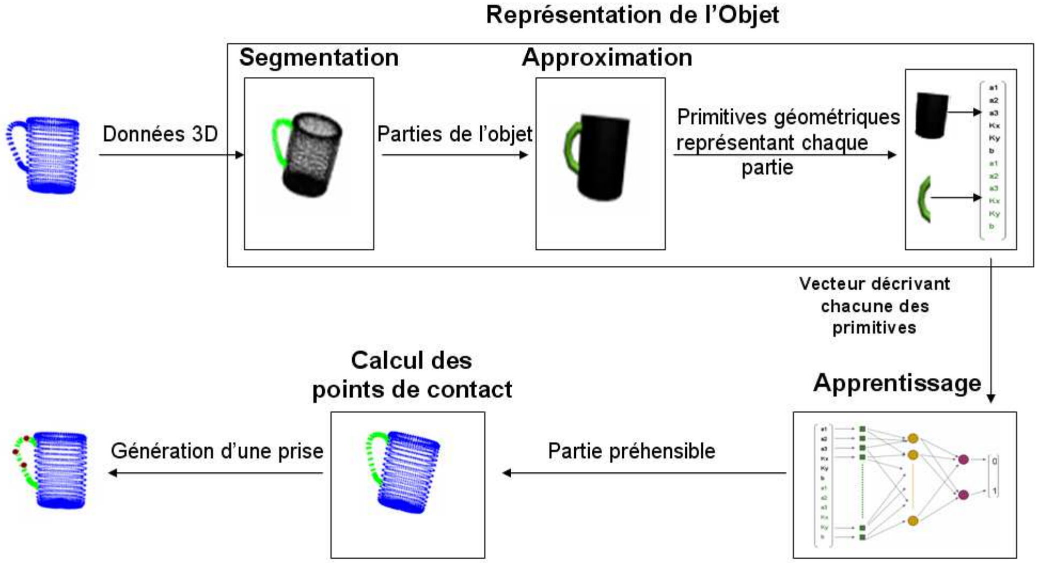

Représentation de l'Objet
Segmentation
Approximation
Données 3D
Parties de l'objet
Primitives géométriques représentant chaque partie
Vecteur décrivant chacune des primitives
Apprentissage
Calcul des points de contact
Partie préhensible
Génération d'une prise

FIG. C.4 – Les différentes étapes de l'approche proposée.

Les différentes étapes sont les suivantes :

Segmentation : Nous nous inspirons de la théorie de reconnaissance par composantes pour décomposer les objets en parties. Le but est d'obtenir une représentation intuitive de l'objet en terme de composantes, conforme à celle de l'être humain. La segmentation s'intéresse alors au problème suivant : comment déterminer les frontières entre les différentes composantes d'un objet quelconque ?

Approximation : Cette étape consiste à approximer un nuage de points par une forme géométrique simple. Mais quelles primitives géométriques choisir ?

Apprentissage : L'étape d'apprentissage permet l'imitation du choix de la partie préhensible de l'être humain. La difficulté sera de construire la base d'apprenissage correspondante.

Les points de contact : Une prise est générée en calculant des points de contact sur la partie préhensible obtenue. Cette prise est-elle stable ?

Toutes ces étapes seront détaillées dans les sections suivantes.

C.3 Saisir Par Composantes - La Technique

Cette section détaille les différentes étapes permettant de déterminer, pour un objet quelconque, sa partie préhensible. Nous présentons d'abord la segmentation en parties. Nous détaillons ensuite l'approximation de chacune des parties par une forme géométrique simple. L'apprentissage nous permet enfin d'identifier la partie saisissable.

C.3.1 Segmentation

la segmentation d'un ensemble de points 3D en parties significatives se ramène à un problème d'étiquetage des points, de telle sorte que ceux appartenant à une même région reçoivent une même étiquette. La méthode utilisée pour la segmentation se base sur un algorithme en 3 étapes [CG06]. Nous commençons par la détection des forntières entre les différentes régions, puis d'associer les frontières aux régions voisines pour ensuite fusionner les régions petites ou non-significatives obtenues. Nous avons testé cet algorithme sur des objets synthétiques, des objets réels dont les modèles sont obtenus à travers un scanner 3D et des objets obtenus par reconstruction 3D. La figure C.5 montre la décomposition des objets réels scannés par un laser 3D.

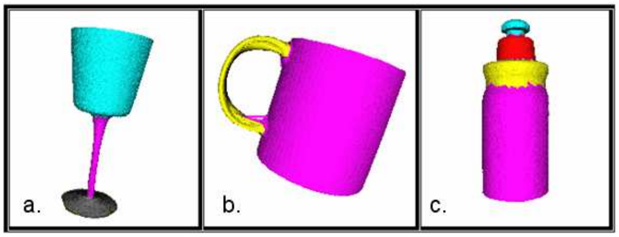


FIG. C.5 – Segmentation des objets réels scannés par un laser 3D.

[CG06] L. Chen and N.D. Georganas. An efficient and robust algorithm for 3d mesh segmentation. *Multimedia Tools Appl.*, 29(2) :109–125, 2006.

C.3.2 Approximation

Nous avons montré que des informations telles que la forme et la taille des sous-parties constituants l'objet sont pertinentes pour la sélection de la partie préhensible. Ainsi, nous étions ramenés à approximer chaque partie obtenue suite à la décomposition par une forme géométrique simple. Nous choisissons comme primitives les superquadriques grâce à leur compacité en terme de nombre de paramètres. Comme les quadriques, les superquadriques sont divisées en quatre classes : les superellipsodes, les superhyperbolodes à une nappe, les superhyperbolodes à deux nappes et les supertorodes. La première classe est la plus utilisée en informatique graphique car elle permet de modéliser, le plus naturellement, des objets tridimensionnels. Nous utilisons 7 formes de superellipsodes pour modéliser nos objets (Fig. C.6). Ainsi, il s'agit d'approximer ce nuage de points par l'une des sept superellipsoides prédéfnies.

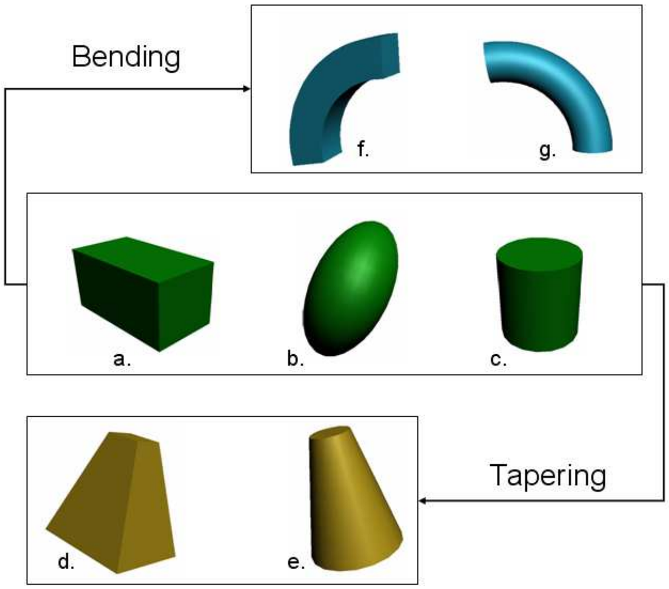


Fig. C.6 – Sept superquadriques pour la modélisation des objets : a) cube, b) ellipsoide, c) cylindre, d) cube étiré, e) cylindre étiré, f) cube courbé et g) cylindre courbé.

En d'autres termes, il s'agit de déterminer les valeurs des paramètres du modèle de la superellipsoide qui approxime le mieux notre nuage de points. L'estimation des paramètres du modèle nécessite la minimisation de la distance entre le nuage de points 3D et les modèles des superellipsoides [SLM94]. Cette distance est une fonction non-linéaire. Nous utilisons l'algorithme déterministe de Levenberg-Marquardt qui est une méthode d'optimisation au sens des moindres carrés.

[SLM94] F. Solina, A. Leonardis, and A. Macerl. A direct part-level segmentation of range images using volumetric models. *In Proceedings of IEEE International Conference on Robotics and Automation*, pages 2254–2259, 1994.

C.3.3 Apprentissage

Nos objets sont modélisés par un ensemble de superquadriques. Cette étape de l'algorithme s'intéresse au choix de la partie préhensible en se basant sur la taille et forme des sous-parties de l'objet. Ce choix doit être conforme à celui de l'être humain. Un réseau de neurones de type perceptron avec une couche cachée est entrainé sur une base d'apprentissage constituée de 12 objets (Fig. C.7). Ces objets sont formés chacun de deux parties. Ces derniers correspondent aux objets réels représentants différentes combinaisons des 7 superquadriques. Or, l'apprentissage nécessite un grand nombre de données. Pour ceci, nous faisons varier des paramètres tels que la taille, la courbure ou l'étirement de facon à conserver la forme générale de l'objet. Nous obtenons ainsi 72 variantes pour chaque objet. La validité du modèle du réseau de neurones est obtenue suite à un 10-fold cross-validation. La capacité du réseau à généraliser est ensuite testé sur plusieurs nouveaux objets.

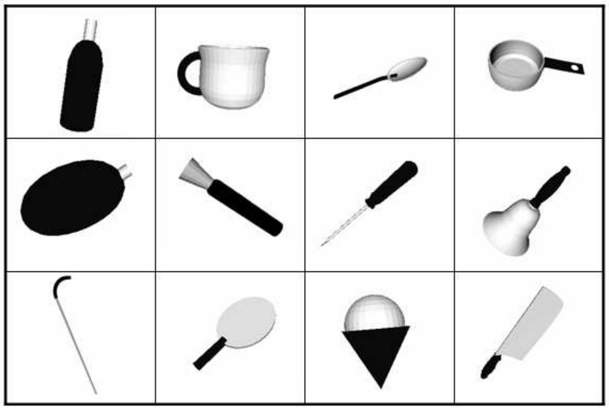


Fig. C.7 – Objets utilisés pour l'apprentissage. La partie en noire correspond à la partie saisissable de l'objet.

C.3.4 Résultats Expérimentaux

Deux expériences ont été effectuées afin de tester la capacité de l'algorithme à généraliser. La première consiste à tester l'algorithme sur des objets formés de plusieurs parties et qui appartiennent aux mêmes catégories de la base d'apprentissage. Ces objets sont tels que des bouteilles, couteaux, tasses, cuillères, etc. Ainsi, si notre algorithme n'arrive pas à identifier la partie saisissable correspondante à chacun de ces objets, nous pouvons conclure que les primitives choisies pour notre algorithme d'apprentissage ne sont pas pertinentes. Pour ces objets, notre algorithme généralise bien et réussi à sélectionner la partie choisie par les êtres humains pour saisir ces objets. En un second lieu, nous effectuons des tests sur des objets nouveaux. Nous obtenons, pour ces objets, un taux de réussite de 80%. Par conséquence, des primitives telles que la forme et la taille relative des différentes sous-parties de l'objet sont à 80% pertinentes pour le choix de la partie préhensible.

C.4 De la Partie Préhensible aux Points de Contact

Dans cette étape, nous disposons de la partie préhensible de l'objet et nous nous intéressons à l'identification des points de contacts sur cette partie garantissant la stabilité de l'objet. Pour cette fin, nous proposons une nouvelle approche pour la génération de prises dites force-closure. Cette propriété est définie par la capacité des forces appliquées par les doigts sur l'objet aux points de contact d'équilibrer tout torseur extérieur. Plusieurs algorithmes ont été développés dans la littérature afin de tester si une prise donnée vérifie cette propriété. Plusieurs mesures de qualité de prise ont aussi été introduites. Pour générer une bonne prise, les approches existantes génèrent en premier lieu plusieurs prises force-closure, calculent ensuite leur qualité et les classent en fonction des résultats obtenus. Ceci nécessite un temps de calcul considérable. Nous proposons une méthode permettant simultanément la génération d'une prise vérifiant à la fois la contrainte de force-closure et ayant une bonne qualité. Nous avons une forte conviction que la qualité d'une prise à n doigts dépend sur la position des $n-1$ premiers doigts autant que de la position du nième doigt. Pour ceci, nous introduisons un critère de qualité pour la génération des positions des n-1 premiers doigts. Nous proposons ensuite une nouvelle condition rapide permettant de choisir le nième point de telle manière que la prise constituée des n doigts vérifie la propriété de force-closure. Nous introduisons en premier lieu quelques définitions et théorèmes nécessaires à l'établissement de notre nouvelle condition.

C.4.1 Définition de la Matrice de Prise

Ce paragraphe présente les notations et définitions nécessaires pour l'élaboration de la matrice de prise. Il donne aussi une définition et une condition nécessaire et suffisante pour qu'une prise respecte la propriété de force-closure.

▶ **D**éfinition 1 : Une **prise** est un ensemble de contacts.

▶ **D**éfinition 2 : Un **contact** caractérise la position d'un doigt sur la surface de l'objet. Pour déterminer une prise, des informations sur le type et le nombre de contact sont nécessaires.

▶ **D**éfinition 3 : Une **force de contact** f_i est la force exercée par chacun des doigts sur l'objet.

Dans le cas d'un contact sans frottement, cette force est exercée selon la normale à l'objet au point de contact. Dans le cas contraire, cette force est contrainte à se situer à l'intérieur d'un cône de frottement centré autour de la normale interne à la surface de l'objet au point de contact (C.1), afin d'éviter le glissement des doigts sur la surface de l'objet.

$$f_{ix}{}^2 + f_{iy}{}^2 \leq \mu^2 f_{iz}{}^2 \tag{C.1}$$

Avec $({f_{ix}}^2, {f_{iy}}^2, {f_{iz}}^2)$ les composantes de f_i dans le repère x, y, z associé à l'objet. Le coefficient de frottement est noté par μ.

Définition 4 : Les contraintes non-linéaires de (C.1) définissent géométriquement un **cône de frottement**.

Pour linéariser le problème, le cône de frottement est généralement echantilloné en m segments.

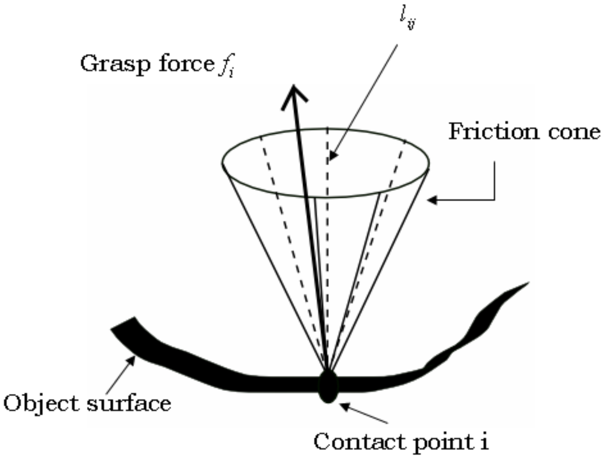


FIG. C.8 – La force de contact f_i sur un cône de frottement echantillonné.

Avec cette approximation, la force de contact est représentée par :

$$f_i = \sum_{i=1}^{m} \lambda_{ij} l_{ij}, \quad \lambda_{ij} \geq 0 \tag{C.2}$$

Avec l_{ij} représentant le j-ième segment du cône de frottement échantillonné. Les λ_{ij} sont des constantes positives.

▶ **D**éfinition 5 : Un **torseur**, $\underline{w}_i$, est la combinaison du vecteur force f_i et de son moment.

$$\underline{w}_i = \begin{pmatrix} f_i \\ \tau_i \end{pmatrix} = \begin{pmatrix} f_i \\ r_i \times f_i \end{pmatrix} \tag{C.3}$$

Avec r_i la position du ième point de contact dans un repère lié au centre de masse de l'objet.

En remplaçant (C.2) dans (C.3), nous aurons :

$$\underline{w}_i = \sum_{j=1}^{m} \lambda_{ij} u_{ij} \tag{C.4}$$

avec :

$$u_{ij} = \begin{pmatrix} l_{ij} \\ r_i \times l_{ij} \end{pmatrix} \tag{C.5}$$

Les vecteurs u_{ij} sont normalisés de la façon suivante :

$$w_{ij} = \frac{1}{\|l_{ij}\|} u_{ij}$$

Le terme $\|l_{ij}\|$ correspond à la norme L_2 du vecteur l_{ij}. Les vecteurs w_{ij} sont les **torseurs primitifs de contact**. Par conséquence, $N = mn$ est le nombre total de torseurs primitifs de contact appliqués par les n doigts.

▶ **D**éfinition 6 : La **matrice de prise**, W, est de dimension $6 \times nm$ pour les objets 3D. Ses colonnes sont *les torseurs primitifs de contact.*

$$\mathbf{W} = \begin{pmatrix} l_{11} & \dots & l_{16} & \dots & l_{nm} \\ r_1 \times l_{11} & \dots & r_1 \times l_{16} & \dots & r_n \times l_{nm} \end{pmatrix}$$

▶ **D**éfinition 7 : Selon la définition de Salisbury et Roth [SR82], une prise satisfait la propriété de **force-closure** si et seulement si tout torseur extérieur peut être équilibré par les forces appliquées par les doigts sur l'objet aux points de contact.

▶ **P**roposition 1 : Salisbury et Roth montrent qu'une condition nécessaire et suffisante pour satisfaire la propriété de force-closure est que les torseurs primitifs de contact génèrent positivement l'espace 6D des torseurs. Cette condition est équivalente à ce que l'origine de l'espace des torseurs soit strictement inclus dans l'enveloppe convexe des torseurs primitifs de contact [MSS87, MLS94, Mon91].

Dém. Pour une démonstration détaillée, le lecteur pourra se reporter à [SR82]. ■

[SR82] J.K. Salisbury and B. Roth. Kinematic and force analysis of articulated hands. *ASME J. Mech., Transmissions, Automat.,Design*, 105 :33–41, 1982.

[MSS87] B. Mishra, J.T. Schwartz, and M. Sharir. On the existence and synthesis of multifinger positive grips. *Algorithmica, Special Issue : Robotics*, 2 :541–558, 1987.

[MLS94] R.M. Murray, Z. Li, and S.S. Sastry. A mathematical introduction to robotic manipulation. *Orlando, FL : CRC*, 1994.

[Mon91] D.J. Montana. The condition for contact grasp stability. *In Proceedings of IEEE International Conference on Robotics and Automation*, pages 412–417, 1991.

C.4.2 Notions Mathématiques

Ce paragraphe introduit des notions mathématiques nécessaires pour la reformulation de la condition de force-closure.

▶ **D**éfinition 9 : Un ensemble de vecteurs, $\{v_i\}$ avec $i \in I$, génèrent positivement un espace vectoriel si et seulement si, v peut s'écrire comme une combinaison linéaire positive de v_i :

$$v = \sum_{i \in I} \alpha_i v_i, \quad \alpha_i \geq 0 \tag{C.6}$$

▶ **P**roposition 2 : Pour un espace Euclidien de n-dimension, $\mathrm{E^n}$, $n+1$ vecteurs sont nécessaires pour générer positivement $\mathrm{E^n}$.

<u>*Dém.*</u> Pour une démostration, le lecteur pourra se reporter aux résultats de l'algèbre linéarire présentés par Goldman et Tucker [GT56]. ■

▶ **L**emma 1 : Etant donné $n+1$ vecteurs, $v_1, v_2, ..., v_{n+1}$, dans R^n, tels que $v_1, v_2, ..., v_n$ sont linéairement indépendants et :

$$v_{n+1} = \sum_{i=1}^{n} \alpha_i v_i, \quad \alpha_i < 0 \tag{C.7}$$

Nous pouvons dire que v_i, $i = 1, ..., n+1$, est une combinaiosn linéaire négative unique des autres n vecteurs [WZG95].

▶ **P**roposition 3 : Un ensemble de $n+1$ vecteurs $v_1, v_2, ..., v_{n+1}$ dans R^n génèrent positivement E^n si et seulement si v_{n+1} est une combinaison linéaire unique de v_i, $i = 1, ..., n$ et que tous les coefficients sont strictement négatifs.

<u>*Dém.*</u> Pour une démonstration, le lecteur pourra se reporter à [WZG95].■

Un exemple de cette condition en 2D est présenté dans (Fig. C.9). V_1 et V_2 sont deux vecteurs non-collinéaires dans R^2. V_3 peut s'écrire comme une combinaison linéaire négative unique de V_1 et V_2. Donc, ces trois vecteurs génèrent positivement E^2.

[GT56] A.J. Goldman and A.W. Tucker. Polyhedral convex cones. *Princeton University Press*, 1956.

[WZG95] R. Wagner, Y. Zhuang, and K. Goldberg. Fixturing faceted parts with seven modular struts. *IEEE International Symposium on Assembly and Task Planning*, 1995.

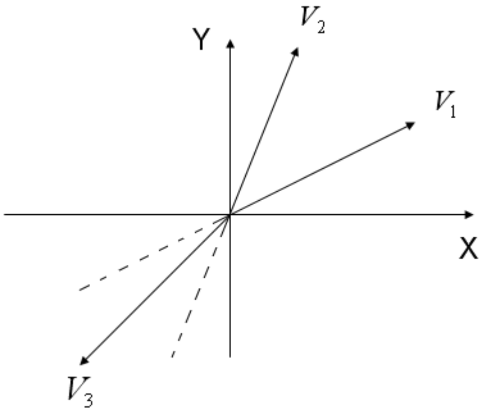


FIG. C.9 – Trois vecteurs V_1, V_2 et V_3 qui génèrent positivement E^2

► Les propositions précédentes montrent que pour générer une prise force-closure ou en d'autres termes pour trouver une prise dont les torseurs primitifs génèrent positivement l'espace 6D des torseurs, nous avons besoin de trouver : (1) des torseurs primitifs de contact qui constituent une base de l'espace 6D et (2) un torseur primitif qui s'exprime comme une combinaison linéaire négative de cette base. Mais, dans quel cas les torseurs associés aux points de contact constituent une base de l'espace 6D ? Pourrions-nous représenter un torseur de dimension 6 dans un espace de dimension 3 ? Les coordonnées de Plücker représentent les torseurs 6D par des droites en 3D et l'algèbre de Grassmann étudient le rang de ces droites. Nous utilisons ces deux études pour montrer que les torseurs associés à 3 points de contact non-alignés constituent une base de l'espace 6D (*proposition* 8). Dans ce qui suit, nous présentons brièvement les coordonnées de Plücker ainsi que quelques résultats utilisés de l'algèbre de Grassmann.

Les coordonnées de Plücker : Soit L une droite dans l'espace $3D$. Soit u son vecteur directeur et P un point choisi sur L. Le vecteur directeur associé à son produit vectoriel avec P définissent les coordonnées de Plücker qui sont notées par $(u; P \times u)$. Ces 6 coordonnées représentent L dans l'espace $3D$ [VY10, Cra73]. Par conséquence, un torseur primitif de contact, défini par $w_i = (f_i; r_i \times f_i)$ peut être représenté par la droite d'action L_{fi} de la force f_i appliqué au point r_i. Les 6 coordonnées $(w_{i1}, w_{i2}, ..., w_{i6})$ de w_i sont applelées coordonnées de Plücker de la droite d'action de f_i.

Les coordonnées de Plücker sont des coordonnées homogènes de l'espace projectif de dimen-

[VY10] O. Veblen and J.W. Young. Projective geometry. *the Athenaeum press*, 1910.

[Cra73] H. Crapo. A combinatorial perspective on algebraic geometry. *Colloquio Int. sulle Teorie Combinatorie*, 1973.

sion 5, P^5 : les torseurs w_i et λw_i, avec $\lambda \neq 0$ représentent la même droite L_{fi}. Donc, à chaque droite L_{fi} de l'espace $3D$ correspond exactement un point dans P^5. L'ensemble des droites forment une quadrique, appelée Grassmannienne, définie par $w_1w_4+w_2w_5+w_3w_6=0$ dans cet espace projectif. Ainsi, nous définissons une relation entre l'ensemble des droites de l'espace 3D et les points de P^5. Cette relation est de rang 6.

L'algèbre de Grassmann : Grassmann a étudié le rang des droites dans l'espace dont le rang varie de 0 à 6. Il a caractérisé géométriquement chaque variété. Nous utilisons deux de ses résultats. Pour une démostration détaillée, le lecteur pourra se reporter à [Dan84].

▶ **P**roposition 4 : Toutes les droites passant par un même point sont de rang 3.

▶ **P**roposition 5 : Toutes les droites qui intersectent une même droite sont de rang 5.

C.4.3 Une nouvelle condition suffisante pour obtenir une prise force-closure à n doigts

Nous avons montré qu'un torseur 6D peut être représenté par la droite d'action de la force qui lui correspond. Nous utilisons cette représentation pour démontrer que les torseurs associés à 3 points de contact non-alignés sont de rang 6. Ce résultat induit la formulation d'une condition suffisante pour obtenir une prise force-closure à n-doigts ($n \geq 4$).

▶ **P**roposition 6 : Les torseurs associés à 3 points de contact alignés sont au maximum de rang 5.

Dém. Pour une démonstration, le lecteur doit se reporter au chapitre 4. ■

▶ **P**roposition 7 : Les 6 droites constituant un tétraèdre sont indépendantes et forment alors une base de R^6 (Fig. C.10).

Dém. Pour une démonstration, le lecteur doit se reporter au chapite 4. ■

▶ **P**roposition 8 : Les torseurs associés à 3 points de contact non-alignés sont de rang 6.

Dém. Pour une démonstration, le lecteur doit se reporter au chapite 4. ■

▶ **P**roposition 9 : Supposons que la prise effectuée avec $n-1$ contacts non-alignés n'est pas force-closure. Supposons que $\{b_i\}_{i=1..k}$, avec $k=6$, est une base 6D associée aux torseurs de contact. Une condition suffisante pour qu'une prise à n-doigts soit force-closure est qu'il

[Dan84] A. Dandurand. The rigidity of compound spatial grid. *Structural topology*, 10, 1984.

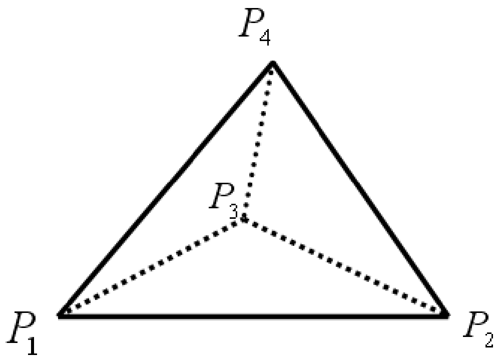


FIG. C.10 – Les 6 droites constituant un tétraèdre sont indépendantes.

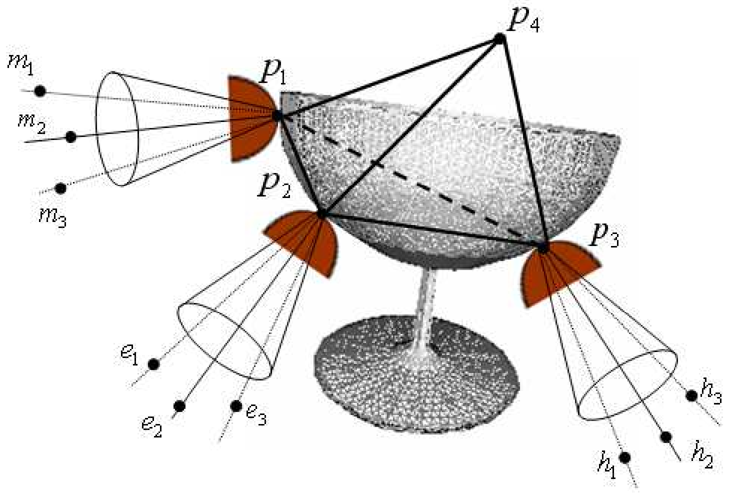


FIG. C.11 – Ls torseurs de rang 3 associés aux points de contact $p_1,\ p_2$ et p_3.

existe un torseur de contact γ tel que :

- $\gamma\ \in\ cone\ de\ frottement\ du\ nieme\ doigt$ (C.8)
- $\gamma = \sum_{i=1}^{k} \beta_i b_i,\ \beta_i < 0$

$$\Rightarrow\ \gamma = B\beta\ \ \Rightarrow\ \beta = B^{-1}\gamma \tag{C.9}$$

Avec $B = [b_1, b_2, ..., b_k]$ est une matrice de dimension $k \times k$ et $\beta = [\beta_1, \beta_2, ..., \beta_k]^T$ est un vecteur de dimension $k \times 1$ strictement negatif. D'où, une simple multiplication par B^{-1} permet de tester si un torseur de contact γ, et par conséquence la position du nième contact garantisse la propriété de force-closure.

Dém. Pour une démonstration, le lecteur doit se reporter au chapite 4. ■

en se basant sur la proposition 9, les paragraphes suivants détaillent la synthèse d'une prise force-closure à n-doigts.

C.4.4 Synthèse de prise Force-Closure à n-Doigts

Pour obtenir une prise force-closure, la matrice de prise doit générer positivement l'espace des torseurs (*proposition* 1). Nous proposons une méthode de deux étapes satisfaisant cette condition : (1) nous générons aléatoirement des positions pour les $n-1$ premiers doigts et (2) nous choisissons ensuite le nième doigt garantissant la propriété de force-closure. Nous avons démontré que les torseurs associés à 3 points de contact non-alignés constituent une base de l'espace 6D des torseurs (*proposition* 8). D'où, une position du nième doigt est choisie de telle manière qu'un torseur de contact qui lui correspond puisse être exprimé comme une combinaison linéaire négative de la base définie (*proposition* 9). Cette approche permet de générer sur l'objet 3D, n points de contact vérifiant la propriété de force-closure.

C.4.5 Synthèse d'une bonne prise Force-Closure à n-Doigts

Pour l'instant, nous avons développé une méthode permettant la génération d'une prise force-closure à n-doigts. Notre objectif est de calculer une bonne prise d'une façon rapide. Dans notre cas, la rapidité et la qualité de la prise sont fortement liés. Pour comprendre le lien entre les deux, il suffit de remarquer que la qualité d'une prise à n-doigts dépend de la position des $n-1$ doigts. Un bon choix de ces derniers induit d'une part une bonne qualité de prise et d'une autre part une plus grande possibilité pour le choix du nième doigt. Cette dernière condition rend plus rapide la sélection de la position du nième doigt. Nous avons donc besoin d'associer un critère de qulité à la génération des positions des $n-1$ premiers doigts.

C.4.5.1 Motivation

Les critères de qualité existant dans la littérature permettent de sélectionner la meilleure prise parmi plusieurs. Ces approches trouvent en premier lieu plusieurs prises force-closure, calculent ensuite leurs qualités pour choisir enfin la meilleure. Par exemple, le critère classique le plus utilisé dans la littérature est celui maximisant la sphère centré à l'origine et contenue dans l'enveloppe convexe. Ce critère n'est défini que si la prise est force-closure. Dans notre cas, générer une prise force-closure dépend de la qualité des $n-1$ doigts, qui, eux, ne sont pas en force-closure. D'où le besoin de définir une nouvelle qualité de prise associée aux positions des $n-1$ points de contact.

C.4.5.2 Critère de qualité associé aux positions des $n-1$ premiers doigts

Nous rappelons que pour trouver une prise à n doigts, nous sélectionnons une base 6D associée aux torseurs des n-1 doigts et trouvons ensuite la position correspondante du nième doigt. Le critère de qualité, sur les positions des n-1 doigts, proposé mesure le volume de la base qui leur est associé. Ce volume correspond à un hypertétraèdre dans l'espace 6D. Plus ce volume est élevé, plus nous avons des possibilités pour le choix du nième doigt, dont un torseur doit s'exprimer comme une combinaison linéaire négative de cette base. Ce critère est testé en 2D et ensuite en 3D. Les résultats expérimentaux montrent que ce critère varie dans le même sens que le critère classique de boule maximale. Ceci prouve le lien entre le choix des n-1 doigts et la qualité de prise à n doigts. Nous présentons en ce qui suit l'algorithme pour générer une prise à n doigts utilisant le critère de qualité proposé :

```
Require: - Ensemble de points 3D représentant l'objet
         - Le cône de frottement discrétisé en chaque point de contact ainsi que les torseurs
           correspondant
Ensure:  - prise force-closure à $n$ doigts
 1: L = Rand_Na_Fingers(n-1)
 2: ntry $\leftarrow$ 0
 3: L_basis = Find_Basis (L_wrenches)
 4: q_L=quality(L_basis)
 5: if q_L <threshold then
 6:    Go to step 1
 7: end if
 8: vertex = Rand_Finger(1)
 9: ntry $\leftarrow$ ntry+1
10: FC = Force_Closure_Test(vertex, r_basis)
11: if ( !FC) and (ntry $\leq$ nmax ) then
12:    Go to step 8
13: else
14:    or Go to step 1
15: end if
```

Etant donné une représentation de l'objet en termes de points 3D, leurs normales correspondantes ainsi que le coefficient de frottement, les torseurs associés à chacun des sommets de l'objet sont calculés. Pour obtenir une prise force-closure à n doigts , la fonction *Rand_Na_Fingers* génère, aléatoirement, sur la surface de l'objet, des positions non-alignées des n-1 premiers doigts. *L_basis* correspondent aux bases 6D associées aux torseurs de ces n-1 points de contact et qui sont déterminées par *Find_Basis*. La fonction, *quality*, calcule la qualité de ces bases. Si cette qualité est en-dessous d'un certain seuil, nous procédons à la génération de nouvelles positions pour les n-1 doigts. Si la qualité d'au moins une de ces bases *L_basis* est supérieure au seuil, un point 3D de l'objet est sélectionné aléatoirement, par *Rand_Finger*, et ensuite testé avec notre condition de force-closure *Force_Closure_Test*. Dans le cas où la prise à n doigts obtenue vérifie la propriété de force-closure, l'algorithme se termine. Dans le cas contraire, *ntry* permet de choisir entre la génération de nouvelles positions des n-1 doigts ou le choix d'un autre nième point de contact. Le choix du seuil ainsi que de *nmax* influe sur le temps de calcul de l'algorithme. La valeur du seuil ne peut pas être calculé analytiquement. Son choix est effectué d'une façon empirique. En ce qui concerne *nmax*, plusieurs expériences détaillent son influence sur le temps de calcul.

C.4.6 Résultats Expérimentaux

Deux expériences ont été effectuées afin de tester la complétude et la rapidité de notre algorithme et ceci en choisissant les positions des n-1 premiers doigts selon des qualités distinctes. Les expériences montrent que des taux de complétude de 18% et de 35% sont obtenus pour une génération des n-1 points aléatoire et avec un critère de qualité respectivement. Ceci est dû au fait que notre condition est suffisante mais pas nécessaire. En revanche, concernant la détermination d'une prise force-closure à n doigts, notre méthode est 4 fois plus rapide que la méthode du convex-hull et les prises trouvées sont deux fois meilleure selon le critère classique de la boule.

C.4.7 Modification de l'algorithme et prise en compte des contraintes géométriques de la main

Jusqu'à maintenant, nous avons supposé que le modèle de la main peut garantir la prise obtenue. Nous tenons compte, dans ce paragraphe, des contraintes géométriques de celle-ci. En d'autres termes, nous ne pouvons plus générer aléatoirement les positions des points de contact sur toute la surface de l'objet. Celles-ci doivent respecter les contraintes géométriques des doigts, d'où la notion d'accessibilité. Ainsi, seulement une partie de l'objet est atteignable par un doigt. La position du doigt correspondant sera alors généré sur cette zone de l'objet. Or, la saisie d'un objet par une main robotisée est un problème à plusieurs chaines cinématiques fermées. La probabilité de générer aléatoirement une chaine cinématique fermée tend vers zéro. Nous nous inspirons, pour ceci, de la thèse de Jean-Philippe Saut [Sau07] et proposons l'utilisation d'une version adaptée de RLG (Random Loop Generator [Cor03]. Cet algorithme permet la génération des configurations respectant les fermetures de chaines. Il découpent ces chaines en chaine active et chaine passive. La chaine active doit être accessible à la chaine passive. Ensuite, le modèle géométrique inverse est appliqué pour le calcul des paramètres de la chaine passive. Dans notre cas, l'objet constitue la cahine active et les doigts correspondent à la chaine passive. Ainsi, l'espace accessible par chacun des doigts est approximé par une sphère. Les collisions entre les doigts et doigts/objet sont également gérées par l'algorithme PQP. L'algorithme permettant de générer une prise force-closure devient le suivant :

Require: - L'ensemble des points 3D points représentant l'objet
- Le cône de frottement discrétisé en chaque point de contact ainsi que les torseurs correspondants

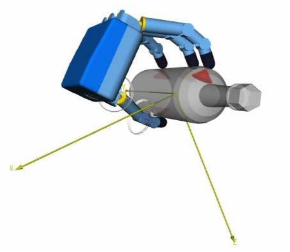
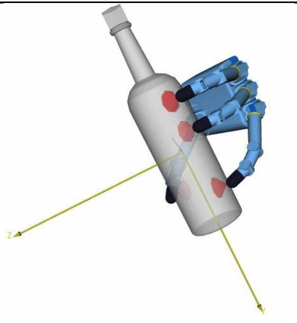
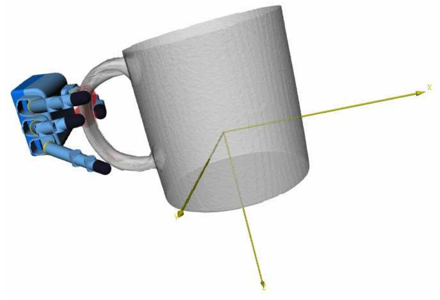
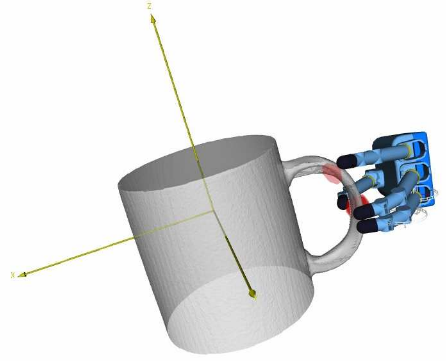
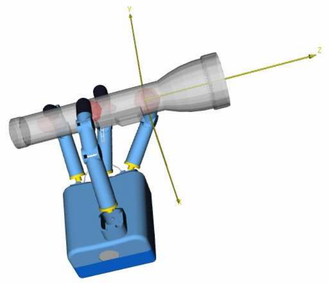
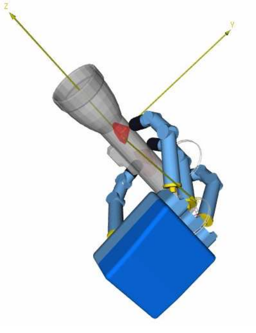
[Sau07] J.P. Saut. Planification de mouvement pour la manipulation dextre d'objets rigides. *Thése de doctorat de l'Universite Pierre et Marie Curie*, 2007.

[Cor03] J. Cortes. Motion planning algorithms for general closed-chain mechanisms. *Thése de doctorat de l'Institut National Polytechnique de Toulouse*, 2003.

```
Ensure: - Prise force-closure à n doigts pour objet/main donnés
 1: Move Hand to Object
 2: $RW_1 \leftarrow Grasp\_RLG(f_1,$object$)$
 3: $CP_1 \leftarrow$ Rand$(RW_1)$
 4: i ← 1
 5: while $i < n - 1$ do
 6:    $RW_i \leftarrow$ Grasp_RLG$(f_i,$object$, \sum_{k=1}^{i-1} f_k)$
 7:    if $RW_i$ then
 8:       $CP_i \leftarrow$ Rand$(RW_i)$
 9:    else
10:       Go to step 3
11:    end if
12: end while
13: $RW_n \leftarrow$ Grasp_RLG$(f_n,$object$, \sum_{k=1}^{n-1} f_k)$
14: FC ← 0
15: while !FC do
16:    $CP_n \leftarrow$ Force_Closure$(\sum_{k=1}^{n-1} CP_k)$
17:    FC ← $CP_n$ in $RW_n$
18: end while
19: END
```

Nous nous intéressons à trouver une prise sur la partie préhensible de l'objet. Une première étape de l'algorithme proposé est alors d'approcher la main de l'objet afin que la partie correspondante soit atteignable par les doigts de la main. RW_i caractérise l'intersection entre le domaine d'accessibilité d'un doigt f_i et de l'objet. *Grasp_RLG* permet l'estimation de RW_i en considérant la position de l'objet et celles des $i - 1$ points de contact. Un point de contact CP_i est ensuite généré aléatoirement dans RW_i. Ceci garantit l'existence du modèle géométrique inverse du doigt. Après le placement des $n-1$ doigts, la position du nième doigt est calculée par *Force_Closure* afin d'assurer la stabilité de la prise. Cet algorithme pourrait être amélioré en introduisant une mesure de qualité sur la génération des n-1 premiers doigts prenant en compte le modèle de la main. Figure C.1 montre plusieurs prises obtenues en utilisant le modèle de la main du DLR et l'interface de GraspIT.

Tab. C.1 – Génération de prises force-closure à 4 doigts utilisant la main du DLR et l'interface GraspIT.

C.5 Conclusion

Cette thèse propose une nouvelle stratégie, qui, issue de l'observation du comportement humain, trouve une prise adaptée à la tâche de l'objet. En se basant sur l'idée que plusieurs objets de la vie courante sont munis, à leur fabrication, d'une partie facilitant leur préhension, l'approche développée détermine, pour un objet de forme quelconque, cette partie préhensible. L'objet est alors représenté par un ensemble de formes géométriques simples. Un réseau de neurones est ensuite entraîné afin d'identifier la partie saisissable de l'objet conforme au choix de l'être humain. Après l'identification de la partie préhensible de l'objet, nous nous sommes intéressés à l'identification des points de contacts sur cette partie garantissant la stabilité de celui-ci. En se basant sur l'algèbre de Grassmann, nous proposons une nouvelle condition suffisante pour la génération de ces prises dites force-closure. Afin de valider l'approche proposée, nous l'avons testée sur des objets CAO ainsi que sur des objets réels scannés par un laser 3D ou obtenus par reconstruction 3D.