State-Action Gist based In-hand Manipulation Learning from Human Demonstration

Dissertation

zur Erlangung des akademischen Grades
Dr. rer. nat
an der Fakultät für Mathematik, Informatik und Naturwissenschaften
der Universität Hamburg

eingereicht beim Fach-Promotionsausschuss Informatik von
Gang Cheng
aus Guangdong, China

June 2013

Gutachter:

Prof. Dr. Jianwei Zhang
Prof. Dr. Bernd Neumann

Hamburg, 12. August 2013 (Tag der Disputation)

Abstract

In-hand manipulation, as a behavior mastered by human, primates and a few kinds of other animals, involves a hand and objects. Transfering these manipulation skills to service-robots is an open and important research topic in the field of robotics. Driven by the hand movements the objects are moved. The hand movements are considered as actions, and we expect the objects to be moved to their destinated states. Therefore, we use "state-action" to model an in-hand manipulation process. For modeling the hand movements, a direct way is to memorize the joint angle variation. However, there are different-sized hands in the world, repeating finger joint angles can produce different manipulation results. Because of that, we propose to use a small number of patterns to summarize the finger motions. In this way we generate the essential information on the actions, and in order to distinguish this idea we name it in-hand manipulation action gist. Correspondingly, with sensors we can capture criteria in terms of the hand, the objects, and the entire environment in the manipulation process, so we use the specific criteria to describe the achievement of the hand movements. Since the criteria are also essential information, we call them state gist.

In the state-action based in-hand manipulation learning framework, everybody can successfully teach the robot. At the beginning of the robot learning, we need persons demonstrating in-hand manipulation movements to the robot. With the state-action gist extracted from multiple devices, e.g., data-gloves, cameras, and tactile sensors, the robot starts to learn the skill itself. Through motor babbling the robot finally masters the in-hand manipulation skill.

In detail, this thesis applies the Gaussian Markov Random Field to extract the action gist from a data-glove. The applied method does not only work for the simple movements such as grasping, but also works for complicated movements such as finger gaiting. Concerning the state gist, this thesis mainly discusses its relationship to the sensors, and gives examples with respect to several typical sensors and several simple state gists. Afterwards, according to the scenarios with multiple demonstrations and the scenarios with periodic hand movements (like screwing), this thesis offers corresponding solutions. Furthermore, regarding the self-learning, this thesis applies the *Particle Swarm Optimization* and the *Line Search with Re-Start* to babbling learn the parameters guided by the corresponding state-action gist. Because babbling learning requires many trials, simulations are taken before the real robot execution. In case the simulated solution is not proper for a real humanoid hand, this thesis proposes a human-interactive mechanism to enhance the real robot learning. In the process of human-robot interaction, the feedbacks are in the form of "compared with the previous trial, this trial is better/worse/equal". With this kind of feedbacks, the robot finds a better solution for the real scenario.

Kurzfassung

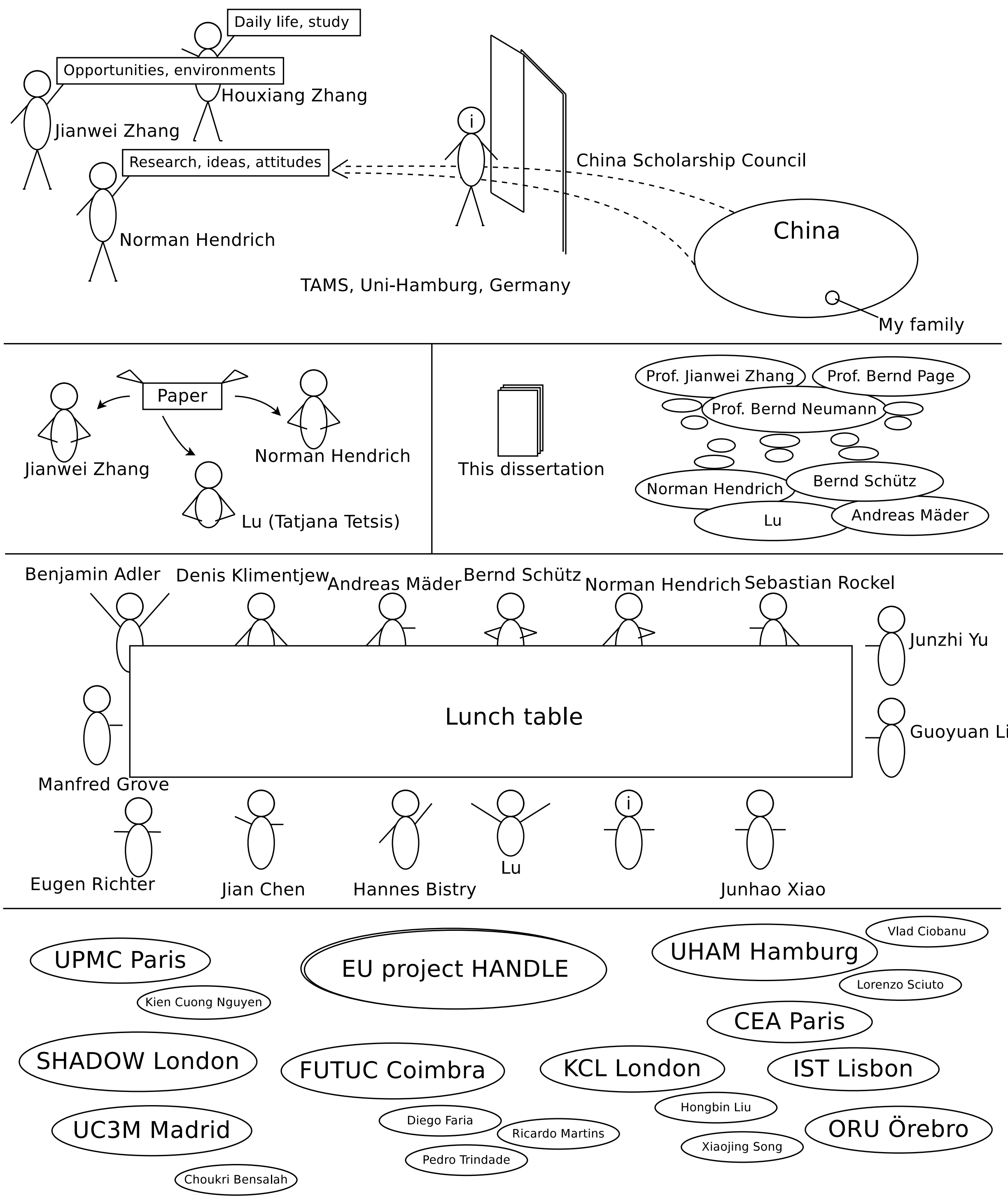
In-Hand-Manipulation bezeichnet das zielgerichtete und kontrollierte Greifen und Bewegen von Objekten mit einer Hand. Neben Menschen verfügen nur Primaten und wenige andere Tiere über diese Fähigkeit, und die Übertragung auf Service-Roboter ist eines der zentralen offenen Probleme der Robotik. Dabei können die Bewegungen der Finger als Aktionen angesehen werden, mit denen ein Zielzustand der manipulierten Objekte angestrebt wird. Es liegt daher nahe, den gesamten Vorgang als Zustands-Aktions-Beziehung zu modellieren. Natürlich kann jede Bewegung der Hand über den zeitlichen Ablauf der Fingerstellungen beschrieben werden, aber es gibt kleine und grosse Hände, die Bewegungen verlaufen nicht immer gleich, usw. Zur Vereinfachung und Abstraktion wird in dieser Arbeit daher zunächst das Konzept der Meta-Bewegungen (*meta-motions*) eingeführt. Auf diese Weise wird die essentielle Information des Bewegungsablaufs der Finger repräsentiert, also der Kern einer Aktion (*action-gist*). Eine ähnlich kompakte Darstellung ist auch für viele Sensordaten möglich, mit denen der Zustand von Hand und Objekt und ggf. der Umgebung während der Bewegung aufgezeichnet wird. Diese Kriterien ergeben die wesentlichen Zustandsänderungen (*state-gist*).

Aktuelle antropomorphe (fünf-Finger) Roboterhände sind rein mechanisch in der Lage, alle Bewegungen der menschlichen Hand nachzubilden. Bisher allerdings basieren fast alle erfolgreichen Anwendungen für Robotermanipulation immer noch auf detaillierter Modellierung und zeitaufwendiger Programmierung durch Experten. In dieser Arbeit wird deshalb ein Framework entwickelt, mit dem der Roboter die wesentlichen Zustands-Aktions-Beziehungen aus Demonstrationen lernen kann. Damit gilt "Jedermann ist ein Lehrer", denn nicht nur Experten können die Manipulationsaufgaben für den Roboter demonstrieren. Die Bewegungen der Testpersonen werden mit verschiedenen Geräten aufgezeichnet, zum Beispiel Datenhandschuh, taktile Sensoren, und Kameras. Daraus werden die wesentlichen Zustands-Aktions-Beziehungen abgeleitet und dienen als Start für den Lernprozess des Systems. Durch *Motor-Babbling* (ein Lernprozess mit zufälligen Bewegungen, angelehnt an das Lernen und Brabbeln von Babies) verfeinert der Roboter dann seinen Bewegungsablauf, bis die Manipulationsaufgabe gelöst wird.

Im ersten Teil dieser Arbeit wird ein *Gaussian Markov Random Field* eingeführt, um die Aktions-Beziehungen (action-gist) aus Datenhandschuh-Sensordaten zu extrahieren. Dann werden die Zustands-Beziehungen (state-gist) beschrieben und mit Beispielen für verschiedene Sensoren erläutert. Für verschiedene Szenarien werden die zugehörigen Lösungen präsentiert, unter anderem für periodische Handbewegungen wie das Aufschrauben einer Flasche. Zum Lernen des Roboters werden die beiden Lernverfahren *Particle Swarm Optimization* und *Line Search with Re-Start* ausgewählt und analysiert. Da das Lernen mittels Motor-Babbling viele Versuche benötigt, kommt eine Simulation der Hand zum Einsatz, bevor die Bewegungen auf dem echten

Roboter getestet werden. Schließlich wird in dieser Arbeit ein neuartiges Verfahren vorgeschlagen (*simple human reward*), bei dem der Mensch als Kritiker die zufälligen Bewegungen des Roboters während der Babbling-Phasen bewertet, um den Lernvorgang in die richtige Richtung zu lenken und zu beschleunigen.

Acknowledgement



Hamburg

Wei Wang
Junlei Yu
Jianhua Zhang
Bo Sun
Hansong Wang
Yifeng Song
Junhao Xiao
Meng Chen
Weiwei Kong
Guoyuan Li
Yuan Li
Junzhi Yu
Junyu Lai
Liwei Zhang
Rong Liu
Jian Chen

Other TAMSers

Wiebke Eggers
Martin Nöske
Sascha Jockel
Dominik Off
Luiza Mici
Mohammed Elmogy
Tim Baier-Löwenstein
Martin Weser
Markus Hüser

Finally...
my readers!

And finally finally...
my love.

Symbols

A few symbols are used with two different meanings in the thesis. Fortunately, these symbols only stay in a small scope. If the symbol is not specified for any chapter, we can understand it with the general concept as listed follows.

d	Distance	General variables
i, j, k, t	Indexes	General variables
N, N_{idx}, L, L_{idx}	Length	General variables
$P(X)$, $P(X = x)$	Probability	General variables
S	Score	General variables
x, y, z, r, v, p, q, s, t	Variables	General variables
w	Weight	General variables
X, X', Y, Y'	Events	General variables
Z	The sum of all probabilities	General variables
a, a_{idx}, α, θ, $\Delta\theta$	Angle	General variables
ρ, $\Delta\rho$	Distance	General variables
f, f_i	Functions	General functions

$\mathbf{A}$	Joint angle variations	Chapter 6
$BioTac_{ij}$	BioTac sensor contact feedback	Chapter 7
$C(\cdot)$	The condition of becoming meta motions	Chapter 3
$d_{\mathbf{m}}$	The dimension of control parameters	Chapter 6, 7
d_{sw}	The size of the sliding window	Chapter 3
f	Focus length	Section 4.2
$\mathbf{F_g}$	A set consisting of finger joint indexes, corresponding to finger g	Chapter 3
$G(\cdot)$	Gaussian distribution function	Chapter 3, 5
$H_{a,r,l}$	The element of the meta motion occurrence histogram	Chapter 5
I_i^j	The single meta motion similarity	Chapter 3
J_{idx}	A joint angle	Chapter 6
L_i	The lower boundary of the searching space	Chapter 7
$\mathbf{M}$	multiple action gists (meta motion sequences), or a long but periodic meta motion sequence	Chapter 5
$\mathbf{m}$	meta motion sequence	Chapter 5, 6
m	meta motion	Chapter 5, 6
N_{frame}	the number of frames to communicate with the robotic hand	Chapter 6
$\text{Neigh}(\cdot)$	A set consisting of a node and its neighbor	Chapter 3
P_i^j	The influence from other nodes of the MRF model	Chapter 3
R_i	Reward for simple human feedback learning	Chapter 7
$Score(\mathbf{m}^s)$	The popularity of the meta motion sequence	Chapter 5
S_a	achievement score of a manipulation trial	Chapter 6
S_b	basic score of a manipulation trial	Chapter 6
S_c	The score from the tactile sensors	Chapter 7
S_t	overall score of a manipulation trial	Chapter 6
$\mathbf{T_j}$	Tactile segmentation solution	Chapter 5
U_i	The upper boundary of the searching space	Chapter 7
$\mathbf{v}$	a set consisting of data-glvoe value variations	Chapter 3
v_i^k	data-glove value variation	Chapter 3
w	Image width	Section 4.2
W	Real world width with camera	Section 4.2
$\mathbf{Z}$	action gist segmentation solution	Chapter 5
ι	Temporal variable	Chapter 5
σ	The variance of a Gaussian distribution	Chapter 3
$\tau_{begin}(\cdot)$, $\tau_{end}(\cdot)$, ...	Fetching the corresponding information from the meta motion	Chapter 5, 6
ω, φ_p, φ_c	parameters of Particle Swarm Optimization	Chapter 6

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Contents

Human beings pass not only biological instincts, but also culture and knowledge to their offspring. Therefore, as a highly intelligent creature, human beings become more and more powerful, to compete and survive in the evolution on the earth. As we all know, knowledge is not born when a life begins. Children have little knowledge foundation, but congenitally have the ability to win the skill and grow up, by means of the multiple sensory organs and the complex neural cognition. It is difficult to imagine how a brain works to process and memorize so much information from the channels of seeing, hearing, smelling, tasting and touching over the entire life. However, we believe that the brain compresses the sensing information, and only necessary, or say key, important information remains in mind (Olofson, 2008). In order to abbreviate the expression of "key/important information" and specify our idea in the thesis, we employ the word "gist" throughout this dissertation.

NOTE: in the Oxford dictionary, "gist" has two meanings: 1. *the substance or general meaning of a speech or text*; 2. *(Law) the real point of an action*. Even though this word is not originally designed for the in-hand manipulation description, *substance or general meaning* and *the real point* exactly hit our desire. In order to distinguish our work, we keep using this word.

This world is very fascinating because of the diversities of the creatures. It endows humans very dexterous bodies, especially the hand for each one, which consists of flexible fingers and holds about 20% bones of a human skeleton. Consequently, there is no doubt that the hand is the most dexterous part of the human body. With the hands more skillful than other animals (Marzke and Marzke, 2000), we manipulate every kind of object, solve many kinds of tasks, and build this world.

The skill of manipulation is a kind of knowledge depending on the property of object, the performance of the manipulator (hand), and the target of the application. Generally, a skill is improving along with age (Crast et al., 2009), and some complicated skills have to be passed from the olders to the youngers. As a kind of knowledge, we should find a way to represent the manipulation process. Besides, as a kind of skill, we should find a way to coordinate the knowledge and the implementation.

In the current information age, we can set the manual analysis free but extract the gist with various sensors, and meanwhile we also aim at integrating the hand manipulation skill with robot. This chapter gives an overview of robot hands and take snapshots of what they can do for the manipulation scenarios. Afterwards the main target of this thesis is pointed out: it concentrates on in-hand manipulation learning from demonstration. Then the main contributions are listed. Finally the entire structure of this dissertation is introduced.

1.1 Overview of robot hands

The definition of *robot hand* is difficult to determine. However, we are certain at least about several facts: firstly, a robot hand is articulated; secondly, each link is more or less rigid even though it may have soft skin; thirdly, it is commonly used for object operation. From the literatures, we can find the described robot hands have from two to five fingers. Two fingered hands are usually discussed in the earlier years, and they are currently used to study some complicated dynamic problems. Three fingered hands can fulfill the requirements of most applications. The hands with more than four fingers are installed on the humanoid robots, or applied for the imitation of human behaviors.

We can list several typical robot hands as Fig. 1.1 to have a straightforward impression. Nevertheless, nowadays there are too many robot hands to show, e.g., Karlsruhe Dextrous Hand II from IPR (Osswald and Woern, 2001), and RoboCasa Hand (Zecca et al., 2006). Besides the fact that some robot hands are born for the market competition, some robot hands are designed and studied because of the specific mechanisms, e.g., single-motor-driven actuation (Chen and Xie, 1999), the underactuated hand (Odhner and Dollar, 2011) and the concentration on compliance (Biagiotti et al., 2003). Therefore, for more information, we can refer to (Mattar, 2013), (Mindtrans, 2013) and other survey resources, or the corresponding specifications.

Furthermore, when we apply a robot hand for manipulation task, besides the basic properties of the hand, e.g., size, weight and the number of the links, we consider the following points:

- **Joint angle range**. This feature determines the available working space of the hand. High values indicate the flexibility of a hand, but according to a humanoid hand, the joint angle ranges should be similar to those human hands.
- **Joint reaction speed**. Usually it is a set of configurable parameters with respect to the hand joints. Quick response promises the expected action is executed on time. Meanwhile low speed but stable movement is required when we just attempt some trials.
- **Torque capability**. To protect the hand joints not being broken, this is a very important factor. With a high torque, we can apply the manipulator to handle a heavier object, or move the finger quicker even if we estimate it will meet a collision.

- **Control accuracy**. For the dexterous manipulation, especially for the in-hand gaiting manipulation, it is no doubt that we have to pursue for the high accuracy in the world coordinate. This world coordinate is for the object, but we configure the angle value in the joint space to achieve the interaction between the hand and the object. Besides the emphasis on the resolution, accuracy is a concept related to the stability, i.e., we set the joint angle to a specific value, the actual error should be limited at a fixed range.
- **Skin design**. Skin is in charge of the contact with object. On the one hand the material should not be so soft, because in this case it is difficult to estimate the exact position of the object in the hand; But on the other hand it should not be so hard, because in this case it is difficult to integrate the hand with contact sensors. Besides, we should pay attention on the texture of the skin (e.g., finger print, palm print), because this is related to the friction. Generally, the texture should have resistance to abrasion, so as to keep the same operating result with repeatedly performing a set of hand movement. When the friction is not satisfied, the hand can wear specific glove to compensate.
- **Sensors**. Human hand has touch feeling via the nervous system. Specifically, we can consider that there are countless arrays of touch sensors mounted on the hand, so we can feel strong or subtle contact anywhere around it. However, installing countless, tiny-sized and reliable contact sensors on a robot hand is a promising but on-going topic. Thus, when we are comparing the contact capabilities between two robot hands, we have many factors to evaluate their performance, such as the contact array size, the scale of a sensing unit, the sensing range, the sensing sensitivity, the sensor life and so on. Besides touch sensor, some robot hand mount other sensors as torque sensors (Liu et al., 2008) and absolute position sensors (Lovchik and Diftler, 1999).
- **Communication speed**. With the increasing number of the hand joints and sensors, the requirement of transmitting the joint commands and sensor feedback becomes an important target of robot hand design. In-time information guarantees that we have more chances to control the robot hand to perform proper reactions. Contrarily, with the delay of information transmission, we have to spend extra work on recovering the real situation. For example, we hold an object and now release it, but the transmitted contact state remains in five seconds ago; in this case the robot may make a wrong plan.

Corresponding to the robot hands we have illustrated, Tab. 1.1 provides us with quick notes on the given performance. From the table we can see that every robot hand has its own characteristic. Besides, we want to emphasize that while the joint angle resolutions of most robot hands are lower than 1 degree, there are some uncertainties for the joint control. According to our experience, the control error of the robot hands (Shadow hands) in our lab cannot be limited below 1 degree. Therefore, on the one hand we can learn the basic functions from the documents, on the other hand we should have the real test on the robot. By this way, we will be clear about what kind of applications can be focused on.

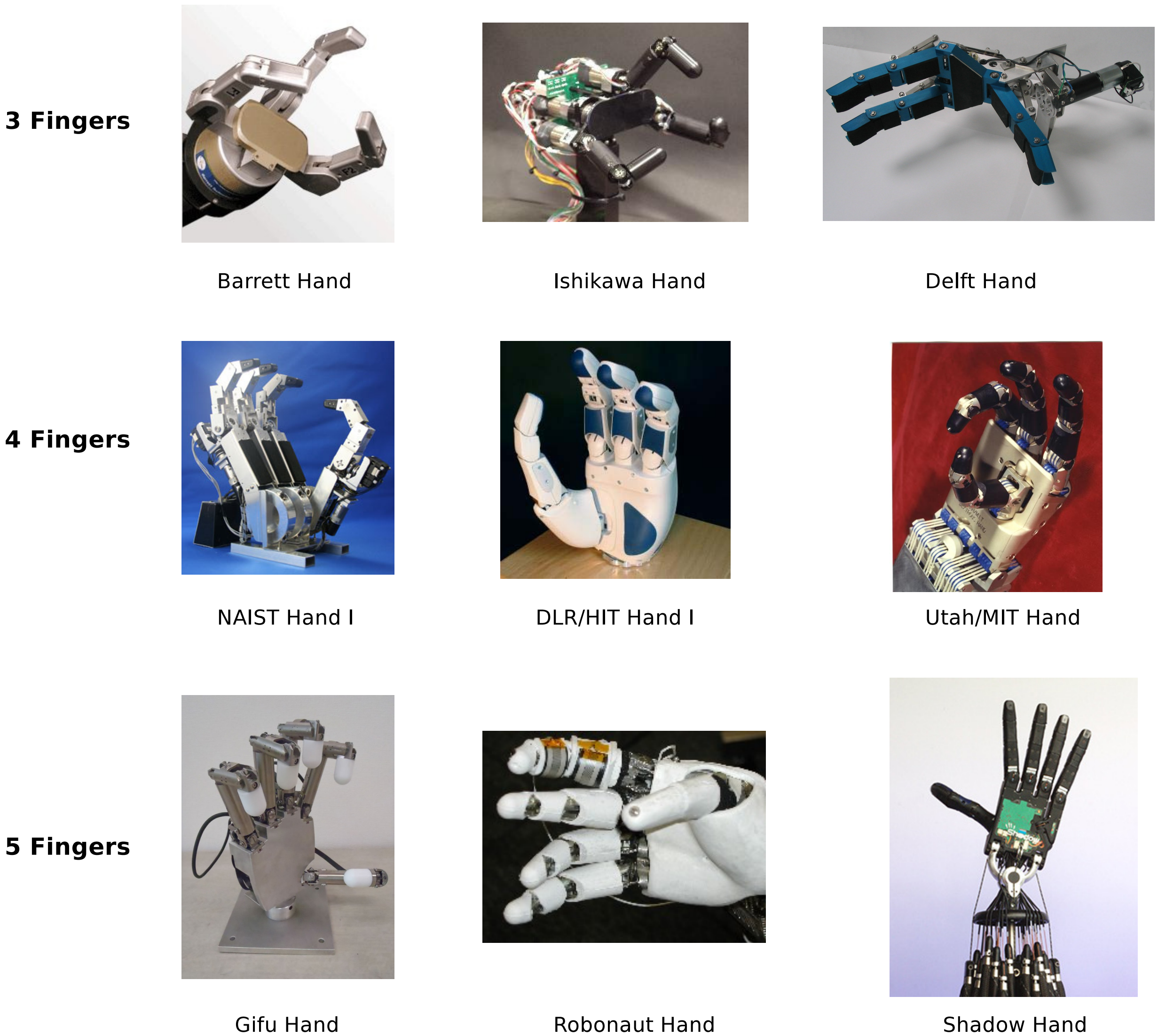


Figure 1.1: Typical robot hands. From the top to the bottom, as well as the left to the right, they are Barrett hand (Barrett, 2013), robot hand from the Ishikawa Lab in the Tokyo University (Ishikawa, 2013), Delft hand (Delft, 2013), NAIST hand I (Ueda et al., 2010), DLR/HIT Hand I (Gao et al., 2003b; Stone et al., 2007), Utah/MIT hand (Narasimhan et al., 1989; Fuentes and Nelson, 1996, 1998), Gifu hand (Kawasaki et al., 2001; Mouri et al., 2011), Robot hand for robo-astronaut (Lovchik and Diftler, 1999) and Shadow hand (ShadowRobot, 2013). So far we can find many robot hands available in the world, from three fingers to five fingers. We do not count two finger grippers here. Principally, three finger hand qualify for most manipulation tasks as long as each finger joint is actuated as human hand joint. However, according to the fact that a normal human has five fingers, humanoid hands have the same number of fingers with respect to the natural evolution. On the way of the humanoid hand development, some products consist of four fingers in order to save the data transmission workload; besides, another reason is because human seldom subjectively and actively use little finger. However, the new versions, e.g., NAIST Hand II and DLR/HIT Hand II, have five fingers. Because of the page limit, they are not illustrated in this figure.

Table 1.1: Performance of the robot hands shown in Fig. 1.1

Robot hand	**Highlight**	**Joint angle range and Joint reaction speed**	**Torque capability**	**Control accuracy**	**Skin design**	**Sensors**	**Communication speed**
Barrett hand	commercial product	different from human fingers BH8-280: 1 sec from fully open to fully closed	1 Nm	$<$ 1 deg	suitable for most objects	tactile sensors BH8-280: 96 active cells	CAN, RS-232, USB
Ishikawa hand	fast speed system sensing and reaction	different from human fingers 0.1 sec for 180 degs	tip joint 0.245 Nm others: 1.71 Nm	$<$ 1 deg	suitable for most objects	tactile: binary response from each fingertip visual: active, stereo, 1 kHz	special design for real-time visual processing
Delft hand	all actuated by a single motor	different from human fingers just a gripper	8 kg payload	N/A	suitable for various objects	with tactile sensors	not clear
NAIST hand	all joints are driven without wires	similar to human fingers speed not clear	not clear	$<$ 1 deg	suitable for various objects	vision-based grip-force sensing	not clear
DLR/HIT hand	sensing its own pose and torque	similar to human fingers maximum 180 degs/sec	2.4 Nm	$<$ 1 deg	suitable for most objects	torque sensors tactile array sensors	serial communication bus, 25 Mbps
Utah/MIT hand	an early dexterous humanoid hand	similar to human fingers speed not clear	30 lbs	$<$ 1 deg	suitable for most objects	torque sensors tactile sensors	not clear
Gifu hand	servomotors are built into the hand parts	similar to human fingers human speed	fingertip output thumb: 8.8 N others: 1.1 N	$<$ 1 deg	suitable for various objects	(optional) six-axis force sensing arrays	control frequency is higher than 15 Hz
Robonaut hand	for space use	similar to human fingers speed not clear	30 lbs	not clear	suitable for most objects	absolute position sensors	not clear
Shadow hand	commercial product air muscles	similar to human fingers half human speed	not clear	$<$ 1 deg	suitable for most objects	Fingertip: pressure or (optional) BioTac muscle hand: air pressure motor hand: tendon, temperature and current	100 Mbps Ethernet CAN bus

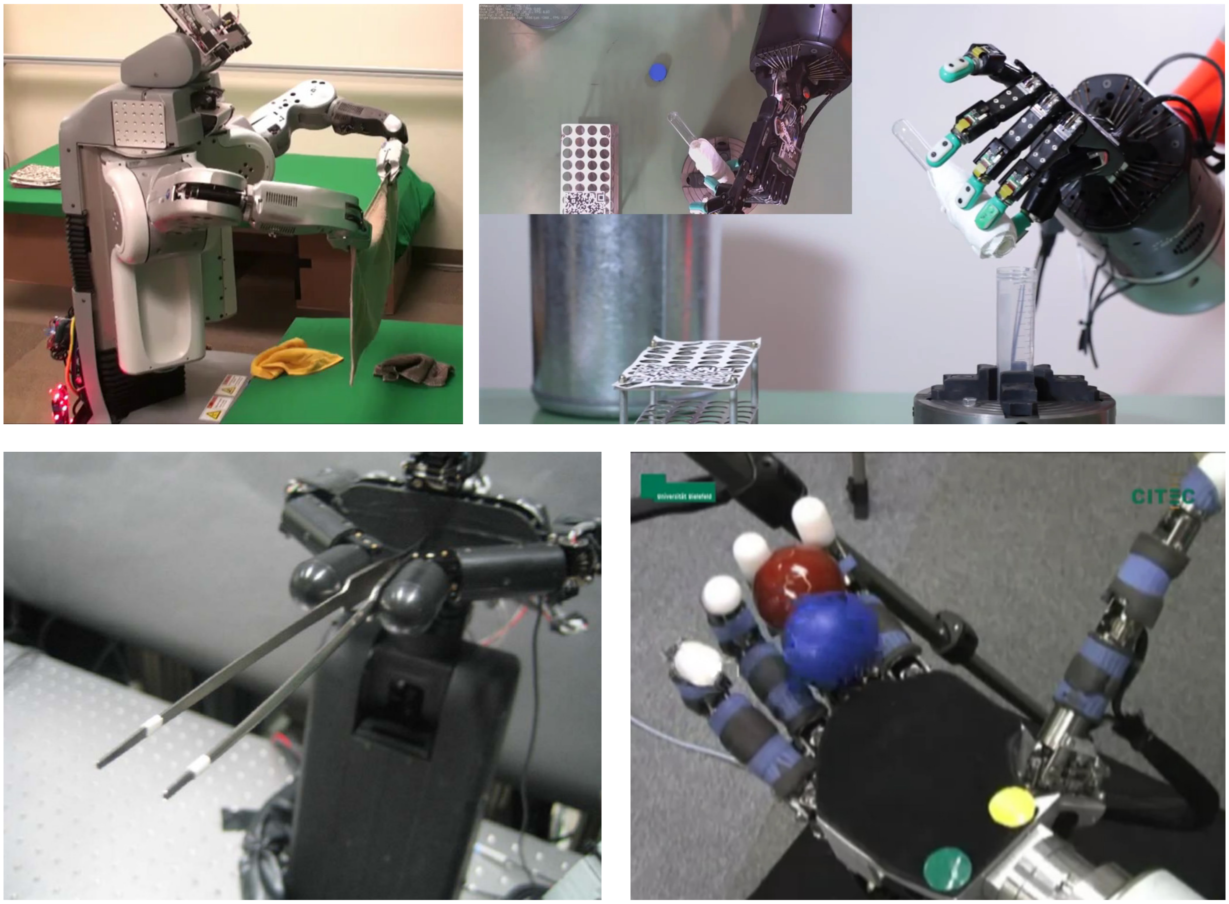


Figure 1.2: Some applications of robot in-hand manipulation. From the top to the bottom and the left to the right, they are: towel folding by PR2 Gripper (Maitin-Shepard et al., 2010); test tube manipulation by a BioTac hand (Uni-Hamburg, 2013); tweezers type tool manipulation by a high-speed three fingered hand (Mizusawa et al., 2008); ball swapping by a Gifu hand (Citecbielefeld, 2010; Moore and Oztop, 2012).

1.2 In-hand manipulation using a robot hand

Once we have a robot hand, we can replay human behavior on the platform. So far there are many scenarios manipulating objects with robot hands. From simple grasping, adjusting the grasping pose meanwhile the object remains in hand (Hasegawa et al., 2003; Sauser et al., 2012), and more interesting applications in Fig. 1.2.

Regarding to the robot hand manipulation, a typical example is the Ishikawa Lab from Tokyo University. They concentrates on their fast-speed three fingered hand design, and through over ten years devotion, they receive fruitful return (Namiki et al., 2000; Ogawa et al., 2002; Namiki et al., 2003; Shimojo et al., 2004; Ogawa et al., 2006; Senoo et al., 2008, 2009; Shimojo et al., 2010). Once they make the dynamical model for a specific manipulation application, they can offer the job to their robot hand and announce their result to the world. So far they have realized Dynamic Catching (Imai et al., 2004), Batting (Senoo et al., 2004, 2006), Dribbling (Shiokata et al., 2005), Regrasping (Furukawa et al., 2006), Pen Spinning (Ishihara et al., 2006), Tweezers type tool manipulation (Mizusawa et al., 2008), Rope knotting (Yamakawa et al., 2007, 2008) and Poker distributing (Yamakawa et al., 2012). From their success, we can find **fast sensing** and **fast reacting** are two important factors for in-hand manipulation.

"In-hand manipulation" is a quite big topic, there are many research branches such as follows:

- **Grasp point: where to (not) touch**. (Bell and Balkcom, 2010) enhanced the classical grasping points theorem in the field of clothes grasping, which is based on the linear programming on convex vertices. Furthermore, (Sahari et al., 2010; Maitin-Shepard et al., 2010) also had interest in how to manipulate the cloth by precise calculation; (Krainin et al., 2010) supposed a way of modeling the manipulated object with Iterative Closest Point to determine where to grasp; (Matsuo et al., 2008) classified different grasp types according to the specific contact regions, as a result, they arranged the positions of tactile sensors to save the resource. For an object with complex appearance, we can use bounding box to simplify the searching space (Huebner et al., 2008). Additionally, we note that the grasp point accept error, that means for some cases the finger has 0.1 mm does not harm anything, such as the Inescapable Configuration Space (ICS) in (Sudsang et al., 2000). Besides determining the grasp points from the 3D information, we can also predict them for 2D image, (Saxena et al., 2008).
- **Grasp quality: which hand states make the manipulation better**. In this field, (Borst et al., 2004) suggested to use task wrench space. Later, (Fu and Pollard, 2006) outlined a linear programming approach for computing a grasp quality metric. The metric includes tendon force constraints and contact constraints and can handle any task described as a polytope in wrench space. Furthermore, (Baier and Zhang, 2006; Baier, 2008) discussed the criterion for grasp evaluation based on reusability. The point is, some hand actions may be possible in mechanical operation, but not similar as the way of what human operate in the daily life. Therefore, their research concentrate on the proper grasping context learning.
- **Force balance: how to make the object stable in hand**. Several researchers are interested in this topic, such as (Woehlke, 1994; Yeap and Trinkle, 1995; Nguyen et al., 2004; Prattichizzo and Trinkle, 2008), proposed how to make a force closed-loop. Besides, other researches such as (Pollard, 2004) concerned about this stability question from observation but not depending on force sensor. (Haschke et al., 2005) proposed the grasping evaluation related to maximal applicable contact force. (Ozawa et al., 2005) discussed a parallel plane condition to maintain a stable grasping. Furthermore, when the object size is similar to the fingertip, the shape of fingertips becomes an inevitable factor to consider. (Arimoto et al., 2010) modeled for 2D grasp and manipulation from Riemannian-geometric standpoint, the curves of fingertip contour are taken into consideration in detail.
- **Synergies: how to use fewer parameters to govern the movement of the multiple joints**. We note that not all researchers like to use the word "synergy", but their target is the same. For example, (Lin et al., 2000) reduced the configuration space of the hand manipulation so as to lower the searching space. Besides, (Vinjamuri et al., 2010) concentrated on determining the morphology of kinematic synergies in rapid hand movements. Furthermore, (Catalano et al., 2012) designed a multi-fingered hand, and the mechanical structure of the hand is directly based on the synergies.

- **Grasping and manipulation mechanism: the natural and theoretical views regarding to the topics**. For example, (Jau, 1995) designed a compliance mechanism for robot hand; (Mason et al., 2011) think there are chances from complex hand structure to simple hand structure, as long as the finger grasping is stable.
- **Motion planning: how to move the fingers**. The general planning techniques are Hierarchical Tasks Networks (HTN) and Markov Decision Process (MDP) (Ghallab, 2004). Besides, considering in-hand manipulation scenarios, moving finger joints will generate trajectories. Therefore, many researchers make contributions corresponding to above key words, such as: (Vass, 2005) proposed a kind of hand-finger trajectory planning algorithm based on simulated annealing and A* search; (Hourtash, 2006) suggested to use the hand configuration space (C-space) and the generalized relative hand-payload pose space to generalize the grasping movement. (Berenson et al., 2008) planned moving trajectory by Rapidly-Exploring Random Trees (RRTs) for grasping manipulation; (Bae et al., 2006) optimized the continuous trajectories planning into a fewer point set planning, the points are key in the manipulator motion control; (Petroff and Goodwine, 2010) applied fuzzy control to a four-fingered manipulator; (Melchior and Simmons, 2010) proposed an idea of trajectory reduction, the work extends Isomap by introducing a neighbor-finding technique suitable for time-series data; As the growing of the skill knowledge, (Yamane et al., 2011) proposed to create a motion database for motion recognition, object state estimation and prediction, and robot motion planning; Last but not least, (Xue et al., 2008) proposed a method to plan the multi-fingered dexterous manipulation. Firstly it computes the contact point trajectories for each finger separately. Afterwards a task-orientated manipulation quality measurement was defined considering the stability during the manipulation on the plane perpendicular to the rotational axis. The manipulation quality is maximized to move the fingers synchronously with same angular velocity.
- **Preshaping: how to move the fingers before we start to handle an object**. This is a key process before grasping, and for the work we can refer to (Smeets and Brenner, 1999; Prats et al., 2007). Different from the previous item **Motion planning**, this process has not interaction between the hand and the object. Therefore, the researchers feel more relaxed on the planning algorithm, but have to consider more challenging objects and scenarios.
- **Inverse kinematics and dynamics: How is the hand pose based on the contact points**. Besides trajectory planning, we can connect the contact points with the hand pose by the inverse calculation. With the pre-determined manipulatior structure, the corresponding manipulator posture can be easily obtained. A reference just like (Han et al., 2000) who emphasized the importance of IK in dexterous manipulation planning and control. Furthermore, manipulation also involves inverse dynamics, such as (Nguyen-Tuong and Peters, 2010).

Additionally, there is another branch, what we are concentrating on: learning.

1.3 In-hand manipulation learning from human demonstration

From the previous section we know that robot hands have been applied to many applications, and the corresponding topics are hotly discussed. However, we notice that usually we have to do modeling and teach the robot how to operate the objects with professional mathematical and physical knowledge. In this case, if we pass the modeling process to the robot, it turns to be a fact that we can set ourselves free. As a result, the robot starts to learn, as from an innocent baby to an experienced adult. We can imagine a following scenario from the future:

A user feels unsatisfied with his humanoid servant because it knows nothing about how to open a bottle of wine. Even though the user knows nothing about how to program the robot, he just has to demonstrate his servant several times the process of opening the bottle. Afterwards, the servant gets the idea, practices itself and finally masters the skill.

Therefore, in order to kick off this topic, we compare this learning process with children learning hand skill from their parents. Firstly, babies use motor babbling to coordinate (or say calibrate) their bodies with the world (Walker and Bass-Ringdahl, 2008; Aronov et al., 2008). Afterwards, as (Poulin-Dubois and Chow, 2009), children create their behavior decision based on their observation beliefs. However, the manipulation behavior performing by the children is not complete reproduction. (Dingwell et al., 2001) designed the experiments for determining if humans adopt model-dependent control strategies when learning a novel motor-skill task. The conclusion is that the observed behavior could not be reproduced by a controller that relied on modulating hand impedance alone with no inverse model. Actually, we will have a concise model in memory and find the explicit movement during the trial. As (Garner et al., 2012) implies, brain encodes sensory input as patterns and stores them for future usage. Thus, we suggest to use a set of patterns to present the manipulation process, instead of to memorize every value trajectory in the demonstrations. Generally, if we insist learning from human demonstration, there are two steps: learning from others and self-learning.

Inspired from the investigation of (Goldstein and Schwade, 2008), we can see the effectiveness of bringing several teachers into the learning process. The teachers absolutely promote the skills of the robot hands to faster the in-hand manipulation learning. Here the “teachers” is not equal to “programmers”, on the specific application they offer the manipulation tips to the robot but spend much less time than the programmers. However, from the view of (Perani et al., 2001), only perception of hand actions in reality maps onto existing action representations, whereas virtual-reality conditions do not access the full motor knowledge available to the central nervous system. Therefore, only relying on a human oral teaching or programming for complex hand movement is not enough, we may miss some key details in the process. The robot should observe what the teachers show, and then learn and practice itself to master the manipulation skill. From fail to success, the robot is experiencing the process called “babbling” (Sjoelander, 2000; Wallace and Whishaw, 2003). In the babbling stage, we are unsure about whether the actions the robot takes will work, but we are certain about the robot is earning necessary information. We can also consider this stage as a kind of reinforcement learning. However, just because the dynamic robot learning process is quite similar to a baby learning, we select the word “babbling” to represent the idea.

Human hand owns a characteristic anatomy (Taylor and Schwarz, 1955). Thus in the initial learning section, we model the hand movement as the correspondent degrees of freedom. From a general to a detailed view, the hand movement can be described as the motions of the whole hand, the fingers and the joints. Besides, (Ansuini et al., 2006) took the reach-to-grasp research, and concluded that the finger joint preshapes and movements are only similar when we deal with the same end-goal object. In this case, in order to perfectly analyze the hand movement, we should straightly focus on the finger level, and extract the necessary information to guide the robot hand movement.

Besides considering the hand, another role in the in-hand manipulation scenario is the object. A property of the object is called affordance. This is a concept involving how people treat an object. For example, a door is usually operated as door opening and door closing, a football is usually kicked, a screwdriver is always for screwing. Dealing with different objects, we need a corresponding manipulating strategy. That is why we are using learning.

Additionally, we need tools to perceive the interaction between the hand and the object. In Chapter 2 we will see various sensing channels on information acquisition. As we have the sensors like the conventional cameras, RGB+D cameras, data-gloves, or tactile sensors, we can extract the posture, the position, the contact and many kinds of state information; Besides, we can extract the hand action information, e.g., from which to which time slot, which finger moves how many degrees. In the field of in-hand manipulation learning, it is necessary to generate a model such as a State-Action Model, so as to guide the manipulation execution.

We note that "state-action" can be understood as "perception-action" or "sensorimotor", but in the area of learning from demonstration, we prefer to use "state-action".

1.4 Motivation and contribution

This thesis mainly aims at proposing a novel but universal solution to fill the State-Action Model. The reason is that so far many researches discuss grasping and hand manipulation consisting of only a few movements. Regarding the complicated in-hand manipulation on the finger-gaiting level, there are few works. Especially, considering the current state of humanoid five-fingered hands, it is time to propose a suitable action model to present the finger movement.

Therefore, what we suggest is a "State-Action Gist based" solution. As the name implies, we need to figure out the gist — key information from the raw sensory input to describe the interaction between the hand and the object. For the state gist, it consists of the corresponding perceptional information such as position, posture and contact variation. For the action gist, it consists of the finger moving directions. All of them are not complicated patterns or values, but they effectively present the in-hand manipulation (see Fig. 1.3).

The overall schedule is shown as Fig. 1.4. It indicates that the core of this thesis is how to generate a state-action gist model and how to apply the model to the five-fingered humanoid hand in-hand manipulation. From our point of view, the contribution of this thesis are:

- **For the action modeling, we propose a set of qualitative patterns extracted from a data-glove to present the finger moving direction**. We name this kind of sequential patterns "action gist". A piece of action gist reflects the key finger movement in a demonstration. Besides applying this technique to record/present each demonstration of in-hand manipulation, we can also employ it for multiple demonstrations or a very long demonstration, to analyze human hand behavior or automatically segment the periodic natural hand movement.

- **Corresponding to the action gist, we propose the concept "state gist" to present the variation of the hand, the object and the hand-object interaction**. We apply some available sensors to record the key information of in-hand manipulation, which involves the immediate and continuous sensory feedback. Paired with action gist, we can describe any in-hand manipulation process as a script. Because there are countless state information in the real world, we enumerate and analyze the popular state features.
- **Based on the State-Action model, we propose an "incremental motor babbling learning" solution to interactively refine the robot hand manipulation skill**. When a humanoid hand carries out an in-hand manipulation task, a concise State-Action model is not enough, i.e., robot hand is driven by explicit joint angle values. Therefore, we design parameter exploring methods based on the state-action gist script.
- **Beyond the simulation of babbling learning, we propose a simple human-robot-interactive mechanism for the real robot in-hand manipulation learning**. The error between the simulation and practice always exists. In this case, we can have the same control method to command the robot hand, but in order to take less time on the real robot we consider bringing in human interaction for learning. However, the attention of a human is limited, the interaction should be simple enough to keep our discretion. As a result, when the robot refers to our feedback, we just need to answer them: compared with last trial, this time is "better", "worse", "equal".

Additionally, we can refer to the in-hand manipulation classification (Elliott and Connolly, 1984; Bullock and Dollar, 2011) to check our achievement in this field. According to a latest in-hand manipulation classification by (Bullock and Dollar, 2011), we test the proposed methods with a robot hand in terms of the scenarios shown in Fig. 1.5. Generally, there are two movements of an object: translation and rotation. (Bullock and Dollar, 2011) considered the contact points and the axis components, so they have such kind of classification. Since the "no motion at contact" cases are indistinctively dependent on finger gaiting movement, the action gist cannot fully play its characteristics, i.e., we just need a few finger motions to move the object to the final state. Therefore, we are interested in the "motion at contact" cases. So far the rotation and the translation along the z-axis are tested.

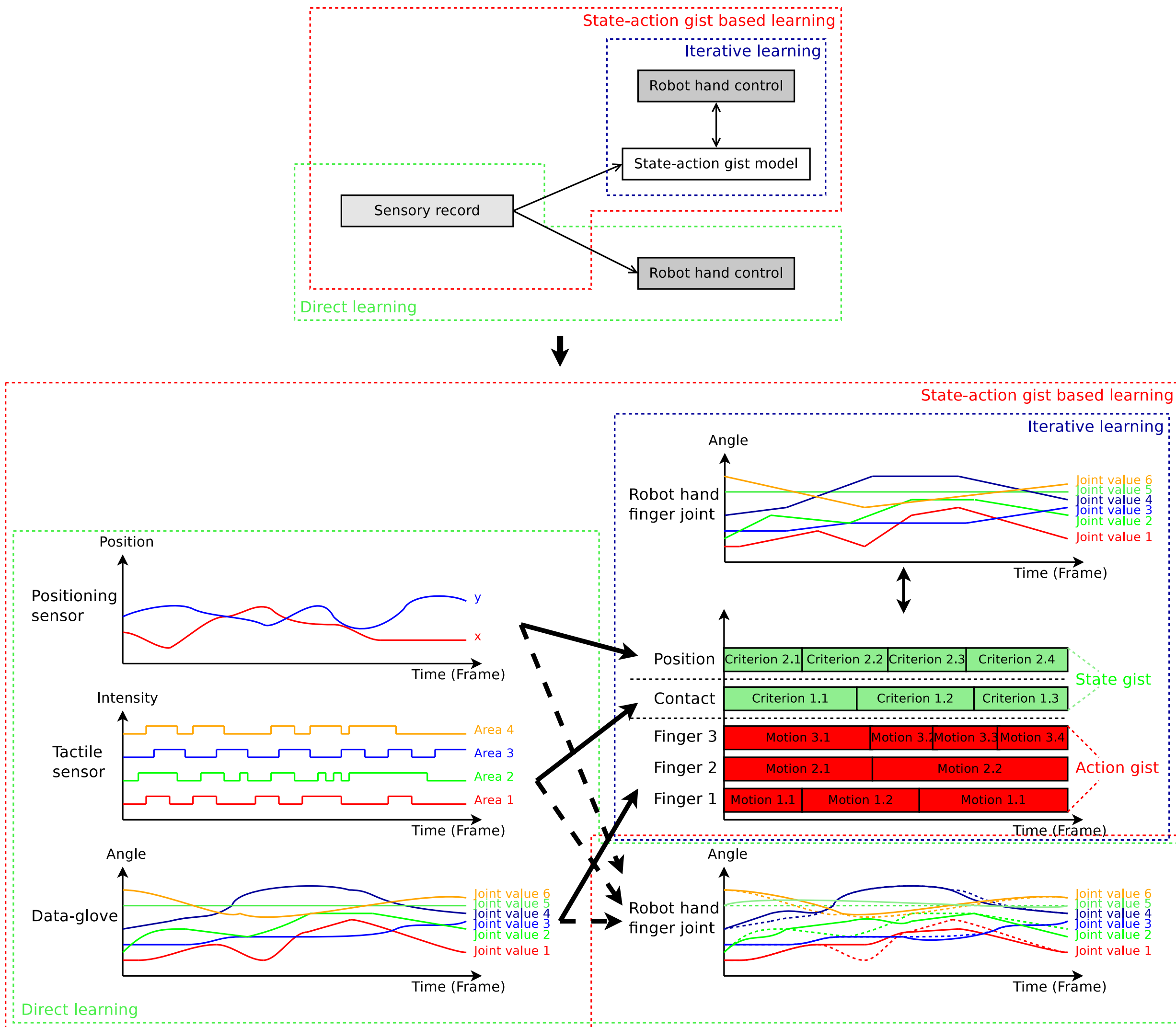


Figure 1.3: Specific research points of the thesis. A robot hand is usually controlled by sending joint values according to the sensory feedback such as positional and tactile information. Even though we have a huge number of values, straightly copying the recorded values to the robot hardly makes the manipulation successful. Therefore, we should learn and plan how to deal with the manipulation task. Nowadays many methods of robot hand manipulation learning are directly concentrating on the value level. When we try **Direct Learning** (green dashes bounded), the joint values are modeled separately. If we apply **State-action gist based learning** (red dashes bounded), we can use fewer memory to remember the in-hand manipulation skill. Besides, action gist reduces the searching space for joint value exploration in the process of **Iterative learning** (blue dashes bounded). In this thesis, **how to model the state-action gist** and **how to make the state-action gist work for in-hand manipulation** are discussed.

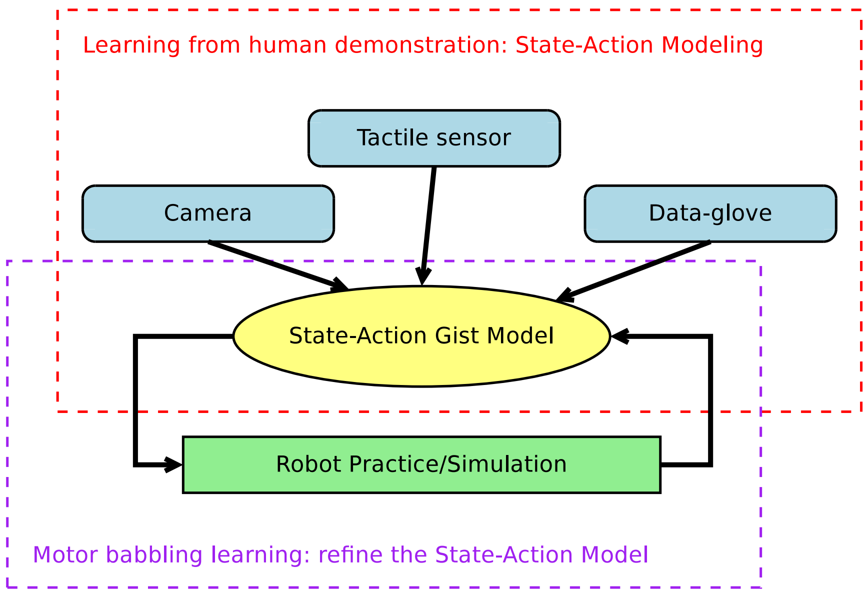


Figure 1.4: Overall framework of this thesis. The core of in-hand manipulation is State-Action Model, so we should know "If we do this action, what state we will get" or "If we observe this state, what action we should take". In other words, we should have a schedule of the in-hand manipulation. What we propose is called "State-Action Gist". This is a concept consisting of key finger moving directions corresponding to the proceeding hand-object interaction. To understand the entire process, we should have sensors, e.g., cameras, data-glove, and tactile sensor, to perceive the relative information of the hand, the object and the contact. By the proposed solution, we get the State-Action Gist Model. Afterwards, we apply the model to robot hand manipulation. Because we are studying learning, the explicit robot hand control commands are also learned iteratively. We can have have simulation environment and real humanoid hands, so we consider the learning solutions for both.

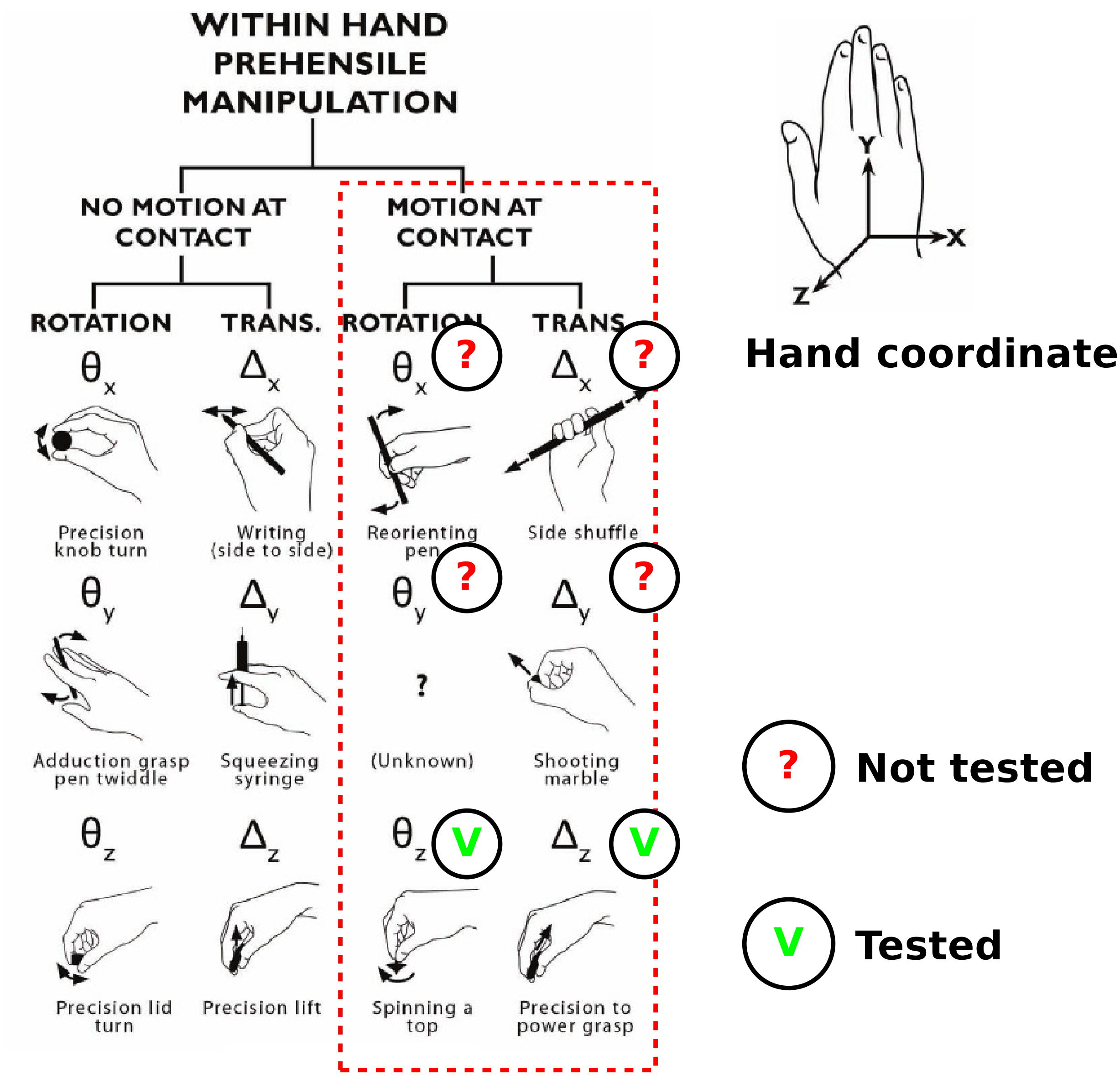


Figure 1.5: A classification of in-hand manipulation proposed by (Bullock and Dollar, 2011). They classified the in-hand manipulation scenarios by contact points, movement directions, and object moving form (translation or rotation). Because the "no motion at contact" cases are indistinctively dependent on finger gaiting movement, we are interested in the "motion at contact" branch. This thesis has proved that state-action gist guided learning at least works for rotation and the translation along the z-axis (as shown in the top-right coordinate).

1.5 Structure

The thesis is organized as follows: Chapter 2 introduces the relevant techniques on the sensory data processing and information generalization, the entire skeleton of learning from demonstration, and the current state and action modeling methods. In Chapter 3, Chapter 4 and Chapter 5, we will see how to model and make use of the action and state gist. Here we model action earlier than state, because we believe that "the action is related to a hand but the state is the world". Thus, We start with solving an easier issue for action gist modeling and then apply similar methods to the state gist modeling. Later in Chapter 6, we propose a incremental reinforcement learning framework based on the state-action model. Especially for the real humanoid hand, we give a learning solution in Chapter 7. Finally Chapter 8 concludes the entire thesis.

Chapter 2

Related Work

Contents

Robot learning is quite a hot topic nowadays, and we can find many successful cases in recent researches. A typical example is from Raibert et al. (2008), they brought the great success of “Big Dog” locomotion learning. The reinforcement learning had been applied to robot dog self-learning to develop the skill of both the low level (joints) and the high level (behaviors).

So far many researches are related to hand manipulation learning. However, our scope is limited to **in-hand manipulation** and **learning from human demonstration**. Many research groups pay attention in this field, with groundbreaking work at institutes such as MIT Artificial Intelligence Laboratory (Matsuoka, 1995; Torres-Jara et al., 2006), DARPA (DARPA, 2010), the Robotics Institute in CMU (Pollard, 1994; Pollard and Hodgins, 2002; Pollard, 2004; Chang, 2010; Koonjul et al., 2011), the Centre for Autonomous Systems in KTH (Preisig and Kragic, 2006; Romero et al., 2009b).

We organize this chapter into following sections: Section 2.1 first describes the sensors. Afterwards we take a glance at the entire situation of learning from demonstration in Section 2.2.

The rest parts are Section 2.3 for the state-action modeling and Section 2.4 with respects to our opinions on the related work.

2.1 Sensory input for in-hand manipulation and learning

Humans usually need visual and tactile sensing to complete a manipulation task (Talati et al., 2005). For the skill learning, we can find many kinds of sensors to capture every moment of manipulation demonstration and implementation. Each sensor is a kind of channel, we can use only one, but can also use many sensors as long as we can afford to record the manipulation process. Afterwards, according to the characteristics of the sensor, we should find a proper method to translate the perception into knowledge. In this section, we will go through the related work regarding the sensor types. However, in this chapter we do little comment on them but leave our opinions to the later chapters.

2.1.1 Visual perception

Vision is the most important input of humans. Adding the fact that a digital camera is the most easily available hardware among various sensors, there are much more relevant researches considering cameras rather than other sensors. Once we have a camera, we can localize the object (Weigl and Seitz, 1994), find the grasping position (Hirano et al., 2005), monitor the hand-object interaction (Gavrila, 1999) and execute many other tasks.

Perception from image is a hierachical understanding progress. The image consists of pixels which are not directly oriented to human cognition. A human (and designed computer program) can parse the object from the features, which holds some invariant properties for the same class of objects and gathers the pixels from a specific region in the image. For example, (Ballester, 2003) completed a 2D object recognition tracking and grasp cycle based on the grasp points manipulation. Commonly, a basic technique is image segmentation, to obtain the hand and the object from the background. If we know the color information or other known features, we can employ the features to find what we want in the image (P. Trindade, 2012). Otherwise, we can use attention (or saliency) based methods to actively find the typical foreground to continue our analysis (Bur et al., 2007; Bogdanova et al., 2008; Zhuang et al., 2009). Furthermore, we can actively add features on the sensing target to enhance the perception (Chen et al., 2008). Instead of previous automatic solutions, we can also apply interactive method to extract the key information from an image. For example, (Bai and Sapiro, 2009) implemented a user-guided image segmentation application that can tell the foreground after the user indicates the foreground textures.

Even though we can find several overviews on the vision and in-hand manipulation, such as (Metta and Fitzpatrick, 2003), the researches on visual processing never stop. Therefore, we provide a detailed overview in several sections.

Before we start, we claim that in-hand manipulation is a dynamic process, different from a single image of gesture recognition (e.g., (Jiang et al., 2006)). Gesture recognition was a hot branch in computer vision several years ago, it offers little help to in-hand manipulation learning. According to our investigation, most researches related to the key word “gesture recognition” assume no object and no occlusion; this point contradicts with “hand-object interaction”. Hence, even though the latest works achieve higher precision and more complicated environments, we will skip most of them and only be interested in the relevant publications.

Perception from multiple visual sensors

To perceive the real world, we should get help from multiple visual sensors. Just as each human has two eyes, stereo vision has been widely applied. Examples are outdoor navigation (Agrawal et al., 2007), surface segmentation (Bleyer et al., 2010), human activity monitoring (Fiore et al., 2008), and nearly all fields.

Besides, in order to achieve the optimal observation, the placement of visual sensors is important. According to the specific tasks, we should carefully design the placement to have the best view of objects and motions, and try to avoid the occlusion, e.g., (Bodor et al., 2007).

When we can separate the color of the hand and background, (Donoser and Bischof, 2008) have achieved the real-time 2D hand position tracking based on Maximally Stable Extremal Region (MSER) tracking framework. However, the fact is that to directly detect the hand joints is difficult. It turns to another solution: installing markers to enhance the recognition. We can install the makers on the fingertips and the palm like (Duca et al., 2007) so as to have a real-time hand tracking. Beside that, we can also do like (Ekvall and Kragic, 2005b) to put three prominent markers on our fingertips so as to identify the particular grasping movement. However, by this way we will face another bottleneck: Our hand movement is restricted, we cannot perform very complex manipulation skills.

Multiple visual sensors can be used for grasping point detection. (Saxena et al., 2008) applied vision into grasping point recognition after the supervised learning on synthetic images. The probable grasping points are identified according to the synthetic features generalized from the images.

Additionally, we can also find the applications of grasping posture mapping. (Do and Jain, 2009; Do et al., 2009) applied visual mapping to hand posture memory. As a result, the robot can learn how humans grasp the object. Moreover, for the specific object, the robot will perform the same grasping strategy.

Last but not least, multiple visual sensors can be used in hand-object interaction analysis (Utsumi et al., 1995; Utsumi and Ohya, 1998; Utsumi et al., 2004).

To employ multiple visual sensors brings several issues, they are related to resource, time synchronization and different illumination. Once we select to use multiple cameras, we have to carefully deal with them.

Model-based multiple component tracking

In the discrete view, a trajectory is a set of sequential points. In our scenarios, there are at least two components: the object and the hand. However, the hand is a kind of articulated object. Considering that we are doing in-hand level recording and learning, the component dimension is inevitably increased to more than ten or twenty.

The common framework are Joint Probabilistic Data Association (JPDA) (Hong, 1994; Abbott and Williams, 2009) and Multiple Hypothesis (Bojilov et al., 2002; Chenouard et al., 2009). Nevertheless, the popular algorithm applied to this field is Unscented Kalman Filter (UKF)

(Stenger, 2001; Stenger et al., 2001b,a), Bayesian Filter (Stenger, 2004; Stenger et al., 2006) and Particle Filter (PF) (Hue et al., 2001; Khan et al., 2003; Bray et al., 2004, 2007). However, they focus only on the entire hand, but not the hand components (joints). After tracking, we have to continue the job of component segmentation.

Fortunately, we realize that the topology of human hand is fixed. In another words, we know the model (e.g., generally, a hand model can be treated as a graphical model (Sudderth, 2006, 2008) or a vector model (Chik et al., 2008)). Benefiting from this fact, the tracking results become better (Heap and Hogg, 1996; Herda et al., 2001; Peursum et al., 2010). Similarly, using the object structure also increase the tracking precision (Al-Shaher and Hancock, 2004). Once the hand touches the object, we can infer the position with the help of the know topological information. Moreover, another advantage is to use the known model to conquer the occlusion problem. For example, (Papadourakis and Argyros, 2010) combined object movements and foreground blobs to automatically and dynamically detect the actual object representation; (Grabner et al., 2010) paired the target object with a support object, once the coupled support object is detected, the position of the target object can be identified regardless of it being occluded. The second example is similar to have a set of assistant markers. Actually, textured marker detection and tracking is model-based. Therefore, the general idea is: if we know several key points in noncoplanar positions, and we know their real structural relations, we can estimate the spatial position of the attached object in the real world. Furthermore, **Pose from Orthography and Scaling with ITerations** (POSIT) is such a classical algorithm (DeMenthon and Davis, 1995). For more details, we can refer to Perspective N Point (PnP) problem (Gao and Tang, 2006; Merckel and Nishida, 2007; Faugere et al., 2008; Lepetit et al., 2009). Besides, we note that PnP is the core method for camera calibration (Schweighofer and Pinz, 2006; Bujnak et al., 2008).

Therefore, to complete the hand tracking task, a solution is to employ external features (markers). (Dorner, 1994) used color-coded glove to track the 3D hand joints information. This is an early attempt, but the idea keeps inspiring later researches. (Guskov et al., 2003) used quad color combination to characterize each finger. In this condition, the finger component is traceable. (Li et al., 2004) assigned the markers on the fingertips to track the scenario of hand gripping. Besides this kind of marker, (Grossman et al., 2005) applied several color-ball rings to indicate the finger joint, so as to track the finger movements. In (Maycock et al., 2011), they put 26 markers on the hand joints to advance the hand posture tracking. Instead of "markers for joints", another branch of employing markers is "texture covers the entire hand". (Scholz et al., 2005) designed a series color-coded pattern to enable the 3D shape reconstruction. Later, (Wang, 2008; Wang and Popovic, 2009) applied a special-coded color glove to track the hand movement. The method applies only one camera, and then retrieves the corresponding standardized pose through their hand pose database, and finally via inverse kinematics to improves the matching accuracy.

Even though adding markers is common and effective for tracking, many researchers are pursuing bare hand tracking. The reason is that we will feel strange when we wear a glove or a set of markers in the manipulation process. In this field, (Davis and Shah, 1994) combined finger feature with optical flow to track the hand, and this method can receive a initial result to perceive the 3D hand motion information. Additionally, (Rehg, 1995; Rehg and Kanade, 1995; Shimada et al., 1998; Guan et al., 1999; Wu and Huang, 1999) combined a hand model and inverse kinematics. When the fingertips are detected, the 3D hand posture will be recovered. Nevertheless, without finger detection, IK could not recover the posture. Instead, in (Gumpp et al., 2006), they used depth information to find the edge of a hand. Afterwards, they parsed the structure of the hand, and track the hand movement with Particle Filter. This drawback of this job is that their scope is mainly for 2D gesture application, not for in-hand manipulation.

(Cui and Sun, 2004) kept the hand skeleton, and applied GA-based particle filter to track the hand pixels in order to track the hand. We note that we extract every link information because it is on our demand, otherwise we just need to obtain the necessary information. Therefore, instead of the explicit hand structure, (Stefanov et al., 2005) just extracted the fingertips and the palm because they more concerned about behavior study. (Ionescu et al., 2005) employed a different hand skeleton, which is generated from image. The main aim of this method is used to gesture recognition. Another example is from (Kerdvibulvech and Saito, 2009); They integrated hand model, chamfer distance, adaptive color learning and particle filter to track the guitar playing fingers.

With the development of the techniques, nowadays researches cover more factors. (Gorce et al., 2008) integrated texture (scene + object), shading, lighting to model and track the hand, and meanwhile considered possible occlusions. (Hamer et al., 2009, 2010) did not only consider the hand model, but also the object model. Along with 2.5 dimentional depth information and pairwise Random Markov Model, each segment can be tracked. Another idea roots from (Kjellstroem et al., 2010), they considered human links as cylinders, then applied annealed particle filter with a linear temporal update model and a background difference likelihood. Furthermore, they also applied inferred hand-object contact points to impose kinematic constraints of the human model.

Remarks

We take many paragraphs on our investigation of "vision in in-hand manipulation". One reason mentioned in the beginning is that there are many researchers working on computer vision. Another reason is because we think other sensory processing may more or less borrow the ideas from visual sensors. To introduce visual processing earlier and more detailedly reduces the length of the coming sections concerning other sensors.

A camera can be used of some distance from hand and object, so we are able to freely perform manipulation as long as no external set-up is installed on the hand. Moreover, we are facing many challenges because a camera offers only two dimensional and general information corresponding to the entire environment. If we select to employ this kind of sensor, we should prepare a set of procedures to solve the corresponding problems.

2.1.2 Perception of RGB+D sensors

Compared with the 2D and color information provided by conventional cameras, if possible we may prefer three dimensional information directly acquired by specific sensors. Besides laser range finders, a kind of sensors that integrates both color and depth information has attracted lots of researchers. In recent years it has been given several names: depth camera, ranging camera, flash lidar, time-of-flight (ToF) camera, and RGB-D camera. The popular commercial products are Microsoft Kinect, Asus Xtion, Leap Motion, Swiss Ranger and PMD CamCube. For more specifications and evaluations, we can refer to their official websites and (Litomisky, 2012).

The sensor names depend on their mechanisms. For example, the Swiss Ranger and the PMD CamCube are time-of-flight cameras but the Kinect and the Xtion are not. ToF camera transmits invisible near-infrared light and measures its “time of flight” after it reflects off the objects. This mechanism is similar to sonar. As long as we know the time of the emitted light go and return, we can calculate the distance of the object in front of. The Kinect and the Xtion apply another mechanism, they project some invisible patterns and use their infrared camera to read the projected patterns. By the deformation of the coding patterns, they calculate the distance.

Because of the different measuring mechanisms, they achieve different performance. Generally speaking, ToF cameras achieve higher accuracy, but they are more expensive. Kinect and Xtion are usually used with a distance condition. For very close depth tracking, we have to select Leap Motion.

So far this kind of sensors are applied to the field of object modeling (Foix et al., 2010), SLAM (Sturm et al., 2012), and motion analysis (Zhang et al., 2012; Herbst et al., 2013). We can also find several attempts on the hand manipulation scenarios, such as hand pose estimation (Yao and Fu, 2012) and hand motion tracking (Zhao et al., 2012; Du et al., 2012). Since the resolution of this kind of sensors is lower than conventional cameras, we have not found the applications completely fitting in-hand object manipulation. Moreover, fingertip tracking plus inverse kinematics are the common methods to analyze the hand movement. The solutions are as same as mentioned above in the visual section, the only difference is that the depth points are calculated from several cameras, another set of points are directly from the RGB+D sensors.

2.1.3 Perception of data-glove

A data-glove is a wearable device to capture the bending of fingers. The advantage is that it can sense the entire hand posture from the finger joint angles, and never has occlusion problem (which is a critical issue when we use cameras). Besides designing a data glove ourselves for the specific task (Folgheraiter et al., 2004), a shortcut is to use a commercial product such as Cyber-glove (CybergloveSystems, 2013).

Before usage, usually a calibration step is required for the data-glove (Huenerfauth and Lu, 2009; Sciuto, 2011). This process requires that the user wears the data-glove and performing a set of typical hand postures, so as to create a set of mapping relations between the raw and the calibrated data-glove values. Besides, (Chou et al., 2008) offered their contributions by connecting the calibration with visual data. They put several different color patches on the data-glove joints, and adjusted the joint position with human interaction. In that paper the 3D position accuracy from vision was still a issue, but this idea should be a branch of the data-glove calibration.

For in-hand manipulation learning, data-glove is a convenient tool to capture the hand movement. Some researches prefer to analyze the manipulation process by frames of gestures. For example, (Dimuro and Costa, 2007) applied an interval fuzzy rule-based method for the hand gesture recognition by fuzzily considering the finger joint angles. Other researches are related to the continuous movement learning. For example, (Ekvall and Kragic, 2005a) used a glove-like sensing method to analyze the grasp movement with Hidden Markov Model. In their work, they assumed several preliminary grasping types and applied several specific features to classify the hand movement.

2.1.4 Perception from biological signals

"Learning from human demonstration" indicates that a robot is trying to acquire whatever a human teacher produces for the manipulation purpose. Besides the appearance of the human hand movement perceived by cameras or data-gloves, we can also try to learn what is in the mind of the operator. For this purpose, we need a biological cue named myoelectric signal. This signal is a technical term in electromyography (EMG) and widely studied for prosthesis control (prosthetic devices like artificial limbs). As long as several electrodes are placed on the skin, we can capture the body voltage and analyze it with some classical machine learning algorithms.

Therefore, we can employ this device to monitor the commands transmitted from the brain to the hand and proceed for our aim. According to our investigation, most related works are still on gesture recognition (Zhang et al., 2009) and grasping (Bitzer and Smagt, 2006). For these purposes, we need to arrange the probes around the lower arm, as shown in (Honda et al., 2007). For the in-hand manipulation scale, we have to pay more attention on the finger movement. In this case, we should arrange the electrodes around the hand. For example, (Atzori et al., 2012; Kuzborskij et al., 2012; Atzori et al., 2013) described such recording methods and learn the dexterous hand control with the acquired myoelectric signals. In the future, we will see a bi-directional communication with this kind of sensing channel, i.e., tactile information will also be captured (Kwok, 2013).

Besides the sensing technique of mounting probes on the arm or the hand, we can also read our mind by putting the probes on our head. (Hochberg et al., 2012) realized a system that can mind-control the robotic hand approaching and grasping. First of all, they install a 4 mm by 4 mm, 96-channel microelectrode array on the participant. And then the signals from the brain will be captured and decoded to the control command. As a result, the connected DLR robot hand moves.

Generally speaking, currently this sensing channel is approaching to the level in-hand movement. We will probably find some new progress in coming years.

2.1.5 Tactile sensing

Tactile feedback is an important feedback for manipulation. On the one hand it tells whether the hand contacts the object (Howe and Cutkosky, 1990), on the other hand it indicates that whether the contact is too much or not enough in order to stabilize the manipulation (Rameon et al., 2013). While integrating tactile sensors with the robot hand, we should also have a process to calibrate the sensors. However, since different people have different sensing, (Sauser et al., 2012) investigated the tactile state by human correction for object grasping. In their study, many participants grasp the same object, so we can have a statistical benchmark on how much force should be applied to keep the object in hand. Besides applying tactile sensors to manipulation control, tactile sensors are also applied to object recognition (Klatzky and Lederman, 2008) in order to increase the interaction between the hand and the object. A better recognition of the target object improves the manipulation quality.

Recently, (Tegin and Wikander, 2005) and (Yousef et al., 2011) made reviews of tactile sensing in robot manipulation and especially in dexterous in-hand manipulation. Their investigations are still up-to-date so far. Thus, we are not going to repeat the state-of-the-art in this section

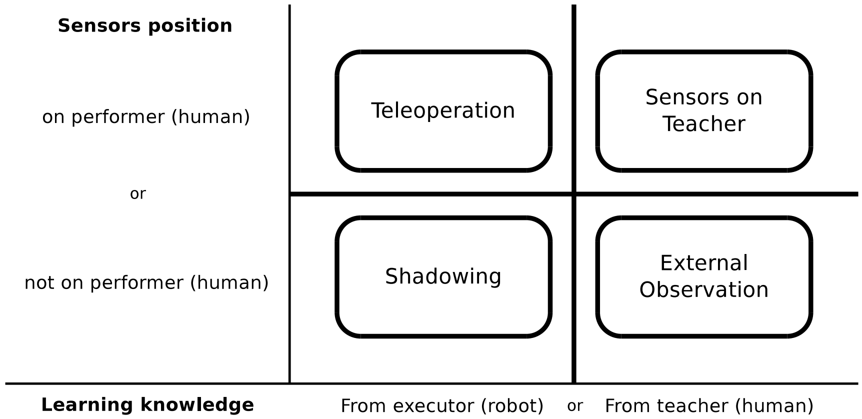


Figure 2.1: Four kinds of LfD approaches. We simplify the comprehensive but complicated expression in (Argall et al., 2009). Generally, four approaches are classified according to the sensors position and knowledge source. As a result, in **Teleoperation**, the human operates the robot platform via the sensors on the human, and the robot records its own behavior and learns; In **Shadowing**, we can understand this approach as "a robot learns from another robot"; In **Sensors on teacher**, the sensors located on the human are used to record the teacher execution, and then the robot can learn the skill from the teacher; In **External observation**, the sensors external to the executing body are used to record the execution, so that the robot can learn from the teacher.

in detail. Generally, depending on the transduction method, the tactile sensors are classified as resistive sensors (Xu et al., 2003), capacitive sensors (Lee et al., 2008), piezoelectric sensors (Steinem and Janshoff, 2007), optical sensors (Hsiao et al., 2009) and organic field-effect transistors (OFETs) as sensors (Darlinski et al., 2005). We can classify the tactile sensors as extrinsic and intrinsic ones. Usually extrinsic sensors adapt many robot hands once it is installed on the hand components. By comparison, intrinsic sensors are fixed in the robot hand, they are not easily updated but more robust for sensing.

2.2 Learning from demonstration

This concept is inherited from Programming by Demonstration (Halbert, 1984). Later, (Argall et al., 2009) gave an abundant and comprehensive overview on Learning from Demonstration (LfD). Generally speaking, in order to realize this mechanism, robot and teachers are the key roles. They transmit the knowledge information in a proper form. Each identity is a system carrying out the plan for the same aim, with the help of a mapping function. The mapping function transfers the knowledge from the teacher to the robot. According to this survey and our further investigation, so far there are four kinds of approaches working for LfD (as Fig. 2.1): **Teleoperation** (Pook and Ballard, 1993; Bitzer and Smagt, 2006), **Shadowing** (Ogino et al., 2005), **Sensors on Teacher** (Calinon and Billard, 2007; Howard et al., 2009a) and **External Observation** (Schaal et al., 2003b; Chang et al., 2008, 2010; Chang, 2010).

Furthermore, with some viewpoints from the in-hand manipulation learning (Turner, 2001), we can find that three approaches are all popular except Shadowing. Anyway, whatever sensors are used or wherever they are installed, the manipulation skill should be finally well presented in an way that a robot understands. As the knowledge involves the sensory feedback and the hand movement, and in the field of learning from demonstration we always name this form as **State-Action Model**. Here, the **State** comprehensively describes the sensory results and the corresponding analysis of the results, meanwhile the **Action** presents the hand movement. The model correlates the states and the actions, and then we know that *at the moment that state s is reached, we can start to execute action a*, or *if we do action a at this moment, we can get state s* (Flanagan et al., 2006).

For manipulation learning, we can try to learn the manipulation trajectory as (Hueser et al., 2006), the contact area as (Baier, 2008), the object-motion map as (Katz et al., 2008), or task order as (Pardowitz and Dillmann, 2007; Pardowitz et al., 2007), but beforehand we should be clear about what model we are applying. Therefore, we introduce the related models in the next section.

2.3 State-action modeling for in-hand manipulation learning

Either state modeling or action modeling consists of many research issues, so each can be a standalone topic. Many researches focus on just one aspect. In this case, we are going to introduce them separately.

2.3.1 State modeling

State consists of many facts along with hand movements. In the modeling process, we need at least select one sensory channel to extract the necessary information.

From the visual channel, we can refer to (Huang et al., 1995). In their opinion, object geometry and hand geometry are the major factors to deal with the grasping problems. For some precise manipulation task, e.g., surgery, the visual criteria should be accurate enough. In this case we need some reliable marker as (Kwartowitz et al., 2009). (Romero et al., 2008a,b, 2009b,a, 2010) clustered human hand grasp appearance, so for the specific object manipulated by robot will be assign as a correct grasp strategy.

If we can learn the entire hand posture with a data-glove, we can also define a series of state transition. (Steffen et al., 2008b,a) considered the dexterous manipulation as a sequence of hand poses. (Vinjamuri et al., 2010) paid their attention on posture decomposition, so as to investigate the hand movement frame by frame on the posture synergies.

From the tactile channel, (Pollard and Hodgins, 2002) supposed that different objects have the same contact area, so they sampled the object surface to find suitable contact regions, and then translated the task from human to robot with another sized object. (Folgheraiter et al., 2004) only considered contact points, but they connect this factor with the grasping types so as to generalize the typical contact strategies. (Saut et al., 2006; Sahbani et al., 2007) proceed the object manipulation based on "from a stable contact state to another stable contact state", the contact point variations are built as a graph. Later in the movement execution, the contact point will tell the robot how to put the fingers, so the stable-grasp (contact point) positions should be very accurate. Similarly, (Kondo et al., 2008; Li et al., 2012) considered manipulation is a flow of touching state transition. In order to figure out which grasping gestures are considered in the manipulation process, (Martins et al., 2010) used the Tekscan tactile sensor for state recognition. Through different contact combinations, the grasping hand configuration is identified. Since many researches just focus on “when-what should be touched”, (Corcoran and Platt, 2010) took a step in “when-what should not be touched”. This is similar to the idea “collision avoidance” but the area is in the sense of touch.

Finally, we can also find many works that involved many sensors. In (Fuentes and Nelson, 1998), they counted position, pose and finger tactile factors. Meanwhile, they used evolution strategy to learn and train their model. For another example we can refer to (Vinayavekhin et al., 2011). It involved the states of the hand, the object and the interaction between them to represent the progress of the manipulation.

No matter what sensors we apply to the manipulation recording, we will finally have a state transition network. Besides generalization, we can also reason about the states. (Ekvall and Kragic, 2005a, 2008) contributed their state reasoning methods for the task learning and planning. In their work, the constraints are identified by the robot itself based on multiple observations and then considered in the planning phase. Additionally, (Vincze et al., 2009) took a attempt to a manipulation perception by vision; for understanding, they made a presumption to the manipulation process, and used reasoning to enhance the process confirmation; for each task the robot has specific steps to execute its actions, so we can easily understand the progress of the task implementation.

2.3.2 Action modeling

Compared with state modeling, the research topics around learning for action modeling are not as widely applied or cited. This is because the actions always connect to the corresponding manipulators. A kind of manipulator just map such kinds of actions. Therefore, action is a concept with hierarchical methodologies. If we generalize the actions like “grasp”, “translate”, and “rotate”, they can be widely applied but we have to do a lot of work on the detailed aspects. However, if we generalize the actions as “x degrees” or “y degrees”, the model is too specific. Anyway, we find some related work in terms of this field besides an overview from (Kulic et al., 2011).

(Fillbrandt et al., 2003) extracted hand posture through pixel-leveled mesh likelihood, and calculated the translating probabilities of each posture pair. Different from other posture transition network mentioned in the previous section, by this work we can know what kind of hand movement is carried out in the manipulation process. Another idea of visual understanding is (Kjellstroem et al., 2008), who claimed that SIFT feature could not present graspable object; instead, based on gradient-oriented histogram, they used two CRF models both in single frame level and image sequence level for action definition. Besides CRF model, Hidden Markov Model (HMM) or Gaussian Mixture Models (GMM) also work for motion primitives generalization (Kulie et al., 2009; Khansari-Zadeh and Billard, 2011).

Mathematical tools are breakthroughs of modeling, and meanwhile another path is the sensor. (Faria et al., 2011b) proposed an approach to represent and recognize a manipulative task from multiple sensing channels. Moreover, if we have some specific constraints, we can also use the data from one-shot demonstration to complete the learning task (Wu and Demiris, 2010).

Above work are around hand level actions. In order to cover most in-hand manipulation application, this thesis is concentrating on the finger level.

2.3.3 State-action modeling

Several researches consider to model the state and the action together. For example, (Faria and Dias, 2009) aimed at identifying several grasp and reach-to-grasp movements by Bayes rule. In their models, the hand trajectories are the actions, the final hand orientation and tactile information are considered as the states. Another example is (Gupta et al., 2009a). It combined the processes of scene, object, action, and object reaction recognition, in order to understand the interaction of human and object. However, the tested scenarios are far away from in-hand manipulation. Furthermore, their action models are not on the finger level, this issue indicates that their model cannot extend to the finger gaiting applications.

2.4 Summary

We can consider in-hand manipulation learning as a modeling and refining process. First of all, we should have a clear view on our state-action model. Based on what we need, we select sensors as to capture every moment of the hand-object interaction. Depending on the characteristics of the sensors, we should find specific analysis methods to extract the manipulation features. In the next step, corresponding to state-action model we extract the necessary information from the acquired data. Finally it turns to the intrinsic parameters in the model, we can employ proper algorithms to complete the last step, e.g. Unsupervised Kernel Regression (UKR, a kind of unsupervised learning technique (Steffen et al., 2008a)), Policy Improvement with Path Integrals (PI2, a kind of reinforcement learning technique (Theodorou et al., 2010)) and many other proper methods.

Even though we investigate many popular relevant researches through the entire process of in-hand manipulation learning, this thesis mainly contributes the state-action modeling part. It indicates that we may apply several methods from others to the fields of sensory data processing and parameter learning, as long as the methods do fit the sensors and robot hands we have, but the state-action framework is different. As the state and the action models mentioned in Section 2.3, we find the current popular frameworks are:

1. **State: Contact points + Action: Inverse kinematics**. We need to know the accurate position of the contact points on the object, so as to plan the hand posture by inverse kinematics. In this framework, how to continuously and accurately locate the contact points are the research issues, meanwhile the hand posture planning for finger collision avoidance is another research topic. If we have no powerful sensors and corresponding analysis algorithms, this solution is not a good idea.
2. **State: Grasp types + Action: Forward kinematics**. Several researches worked for classify the grasping types (Todorov and Ghahramani, 2004; Zheng et al., 2011), because we believe that a simple manipulation can be parsed into several simple grasping postures. Once we have found the correct order, we can plan the hand movement with forward kinematics to handle the object. For complex hand movement, this solution is not enough.

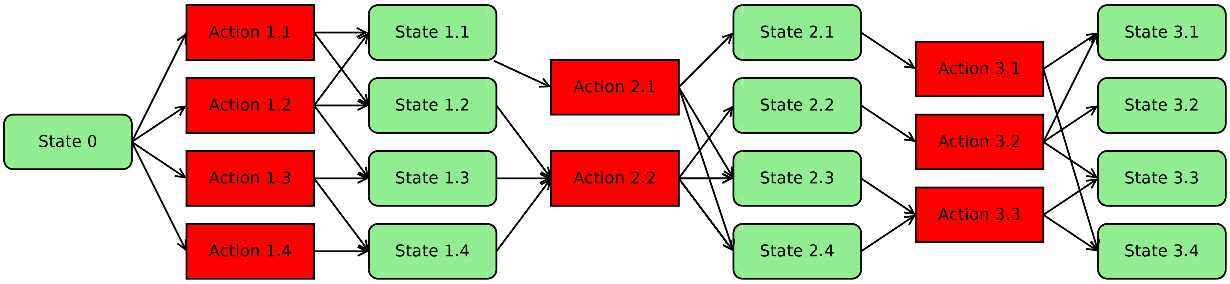


Figure 2.2: A typical State-Action network. From the sensory input, we learn the state-action model as a statistical process. An action results in several ends depending on the uncertainty of the environment. Therefore, we should count the possibilities and generate the in-hand manipulation state-action database. Even though the scale of the database will increase rapidly along with the demonstration accumulation, we store the states and the actions sparsely. In this way, we save the actual memory.

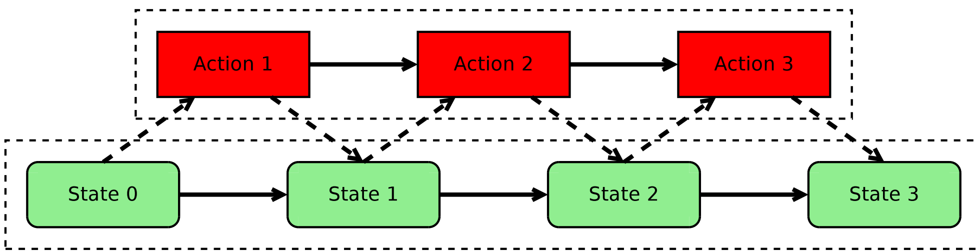


Figure 2.3: A typical State-Action executing process. Even though we model the state-action model like a network, in the real world we only have one route. The robot hand executes the actions, meanwhile we use sensors to acquire the states.

3. **State: Grasp types + Action: Synergy parameters variation**. We can use synergy technique — fewer parameters to govern the finger movement. This is a current hot topic for in-hand manipulation. However, for complex manipulation task, to determine the grasp types is a difficult issue, meanwhile for different manipulation scenarios we should use different set of parameters. This point results in many other issues.
4. **Combination of these or other models**.

Furthermore, we can have the classical state-action modeling and executing mechanism as Fig. 2.2 and Fig. 2.3. Now we are going to propose our ideas to fill the framework.

Chapter 3

Action Gist of In-hand Manipulation

Contents

Different from a *hand gesture*, the *in-hand manipulation action gist* is a concept related to the property of the hand movement. It represents the key hand motions in any given manipulation task and widely adapts to different hands. The manipulation process is generalized as several compact meta motions. On the one hand, this makes it easy to remember, on the other hand it can be translated from one hand to another, just as the knowledge passing from the teacher to the student.

In the mechanism of the human hand, the motions and forces are governed by the neuromuscluar apparatus, refer to (Taylor and Schwarz, 1955). The movement of the hand is continuous, but according to human cognition, it can be classified as finite types of motions in the brain. For example, as the muscles tightening up and relaxing, or the fingers closing and opening. Then in the specific application, the possible solution sequence is recalled and executed. The object in question is touched and released by the fingers and the palm over time. When the touching motion is executed, an interacting force is generated between the object and the hand; meanwhile the hand neuromuscluar system controls the finger forces in a proper state that not only protect the hand itself but also hold the object firmly.

Since action gist will be employed by robot hands, we should find an appropriate platform. In our lab we have a five-finger air muscle hand from the Shadow Robot Company (series C5, see (ShadowRobot, 2013)), very similar to the human hand and well protected against damage even when overforce is applied. With a humanoid hand, a robot can implement much more human-like object manipulation than with a simple gripper. Because of the high degree-of-freedom, a multi-finger robot hand can perform more dexterous skills rather than grasping, holding, or translating the object from one place to another. It can rotate, or shift objects and perform other advanced in-hand movements. These manipulation skills depend on the cooperation of five fingers and the palm, and in the process of in-hand manipulation, the roles are hand and object. The hand plays the role of control, and changing the object state is the aim of the manipulation. Therefore, here the manipulation process is considered as a *State-Action Model* (Ogawara et al., 2002; Pastor et al., 2011; Kjellström et al., 2011), meaning that the whole process is divided into states which are changed through actions. The action is equal to hand movement, and the state is supposed to be the criterion of how the process proceeds. Hand movement can be considered as a continuous hand joint angle variation, with countless angle combinations between each joint pair. The movement leads the manipulation process from one state to another state until the final target of the application is achieved.

The method can be applied in human behavior analysis and in the control of robots with humanoid hands. However, it is unrealistic to map the motion exactly from the demonstrator because of the different hand sizes. It can be imagined that different-sized hands can interact with the object from different distances, obviously resulting in different gaps with the same pose. Actually in developing their hand skills, humans have the ability to learn from others and to practice by themselves. Nobody can memorize the detailed joint angles of their hands, but they can remember the key motions which are related to the moving tendency of each finger. We consider this kind of motions as in-hand manipulation *action gist*.

In this chapter we propose a cognitively feasible in-hand manipulation action gist definition for a robot with an extremely life-like humanoid hand, to enable it to learn in-hand manipulation with a small amount of key information. The action gist is expected to be universal for all in-hand movements regardless of whether it is simple (grasping) or complex (finger-gaiting).

This chapter is organized into several sections: After the following related work, the definition of meta motion is given, which is an element of the in-hand manipulation action gist. Then the modeling process is introduced, and experiments are carried out to discuss the performance of the action gist. The final part is a brief summary.

3.1 Related work

There are multiple ways to generate a manipulation model and represent motions.

One kind of model is to plan the motion in continuous space including the position and speed of each relative component. The major stream is the dynamic movement primitive (DMP) framework introduced by (Ijspeert et al., 2002; Schaal et al., 2003a), in which the movement is recorded and represented with a set of differential equations. The position and speed are controlled in terms of the immediate position and speed feedback. (Pastor et al., 2009) expanded the model into a manipulation control application so that the hand can grasp and place the object in the destined area. To include obstacle avoidance in this job, an extra item is added in the system equation, which causes the form of the framework to change with the task. Different from the separate models to deal with multiple tasks, (Gams and Ude, 2009) applied Locally Weighted Regression to generate the movement, and the manipulating process is divided into several steps by the perceptual input. Rather than generalizing a trajectory in Cartesian or joint angle space, (Gielniak et al., 2010) considered the joint velocity space which enables the robot to accomplish similar tasks. As a result, this method can produce smoother trajectories than others.

The above frameworks consist of models depending on precise perception of spatial manipulator trajectories. However, for muscle control, it tracks the trajectory related to the moving tendency, and not the position. Therefore, DMP does not offer any significant advantages to the target of this paper.

Another stream but a relatively older one is the generalized motor program (GMP), see (Schmidt, 1975; Schmidt and Lee, 1999) where the overall process is guided by invariant features. (Park et al., 2005) extended this model with the symbolic motion structure representation (SMSR) algorithm. The body movement is tracked and segmented according to the joint angles, and then the values are used to plan for a novel similar application. However, the SMSR only extracts the body motion into simple joint angle variations such as increasing, decreasing and stationary. Therefore, it would have difficulties when dealing with the multiple links cooperation application because it does not consider this kind of application very much. Different from simply defining the motion, it is possible to have a related higher semantic model. (Qu and Nussbaum, 2009) applied Fuzzy-Logic Control to execute the motion sequence, and this idea was examined in a 2D five-segment body model by simulation. The above methods suppose that the motion sequence to an application is fixed, but actually humans can have many ways of completing a specific application. What we need are the most effective or common methods of the teacher.

(Mah and Mussa-Ivaldi, 2003) indicated that humans learn motion by way of muscle control, not the position perception. Therefore, to know the posture variation (joint angle) is more important than the absolute posture. Thus the motion tendency-oriented model is more feasible than DMP.

Once the model is decided on, the next problem is how to sense the movement. Many studies concentrate on sensing from the robot, for example, (Pastor et al., 2009; Gams and Ude, 2009; Gielniak et al., 2010). However, for fingers, it is not convenient to directly move the robotic fingers to find the result. Another channel is vision; the components are tracked to complete the motion behavior model. For example, (Riley and Cheng, 2011) employed color patterns on the demonstrator to track the human motion. It is promising to use vision to analyze hand motion, but the visual processing itself is a challenging topic which increases the difficulty of model generation.

A quick way to know the finger movement is using a data-glove, which can sense every finger joint relation in each data frame. Based on this kind of sensing channel, our study intends to generate an action gist model to represent human in-hand manipulation behavior.

3.2 Meta motion definition

To establish a set of hand motions which represents the hand posture transformation in in-hand manipulation, we intend to construct the model as follows:

- It covers as many hand movements as possible.
- Each motion has as few overlaps as possible with other motions.
- The motion involves the relative joint angle variation but no absolute position information.

An exception to the above rules is the idle pose. When the motion remains static for a while, we have to decide whether it is "move, stop and move again" or consider it as moving continuously. Our strategy is to analyze the movement without static motions first, and in the second loop to find the static section following certain rules.

Suppose that the hand has the form of five fingers and one palm, the palm stays still, then the movement is equal to the cooperation of the five fingers. The basic movement of each finger can be classified as open or close, and in terms of the moving direction at the *proximal phalangeal end* related to the palm, every finger has the same motion definition. Specifically, the coordinate origin of the thumb is different from the other four fingers because of its different position on the palm.

Shown in Fig. 3.1, we project the finger motion into two-dimensional space because the finger ends are fixed on the palm. In the *X*-*Y* plane, the finger direction is classified as four directions as the four quadrants in the Cartesian coordinate; altogether with the open, close, and idle motions, each finger has nine types of *meta motions*. To ensure a uniform form of the motion model, the X axis and the Y axis in the moving direction related to the coordinate origin are either parallel or vertical to the palm plane.

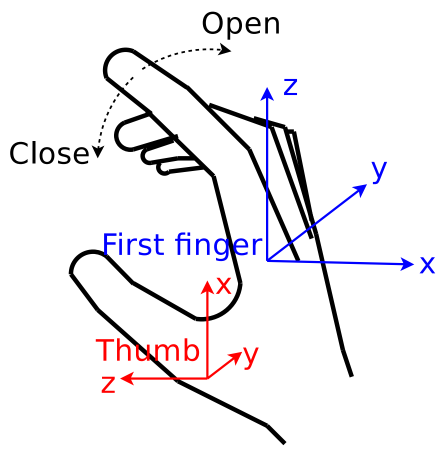
3.3 Action gist from data-glove

Based on the above definition of the meta motions, the *action gist* is defined as the key meta motions between two adjacent states. Guided by the action gist, the object is manipulated from the begin state to the end state.

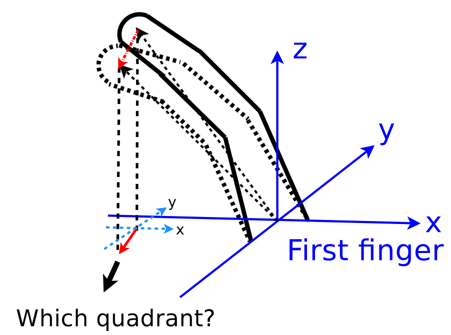
The data-glove is a direct way to perceive the hand movement, as the data is measured by the joint angle value. Therefore, the values from the data-glove become the source for analyzing the hand movement in in-hand manipulation applications.

Corresponding to the degree of freedom, each finger has several joint values from the data-glove. However, usually the distal-intermediate and proximal-intermediate angles increase in close movement, decrease in open movement and the varieties of metacarpal-proximal and abductional angles indicate the moving direction in the *X*-*Y* plane of the finger.

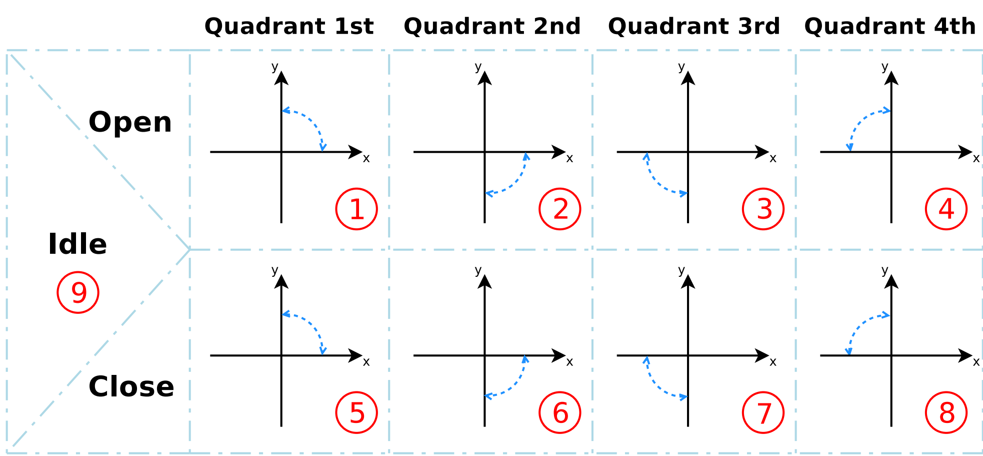
Different from the ideal environment, the acquired data-glove value cannot be directly applied in the analysis. One reason for this is the sensor noise, another one is the issue from the human operator, e.g., a hand tremor in slight operation, a short but unnecessary movement during manipulation, or that at the moment the finger starts to touch the object, the value may be abnormal. From an example shown in Fig. 3.2, we can see the uncertainties and how we are going to deal with them.



(a) Finger coordinates



(b) Motion projection



(c) Motion quadrants

Figure 3.1: Meta motion definition. Five fingers move related to the palm, so we describe their moving directions in the palm-coordinates. The thumb in the red coordinate is different from the other four fingers due to its opposed orientation in the hand. In each finger, two (flex or extend)-joints are modeled as one parameter as open or close, and the abductional angle cooperates with the metacarpal-proximal angle to form a 2D projected direction (blue arc indicates the directional range in the quadrant). The idle motion is set apart and labeled as 9.

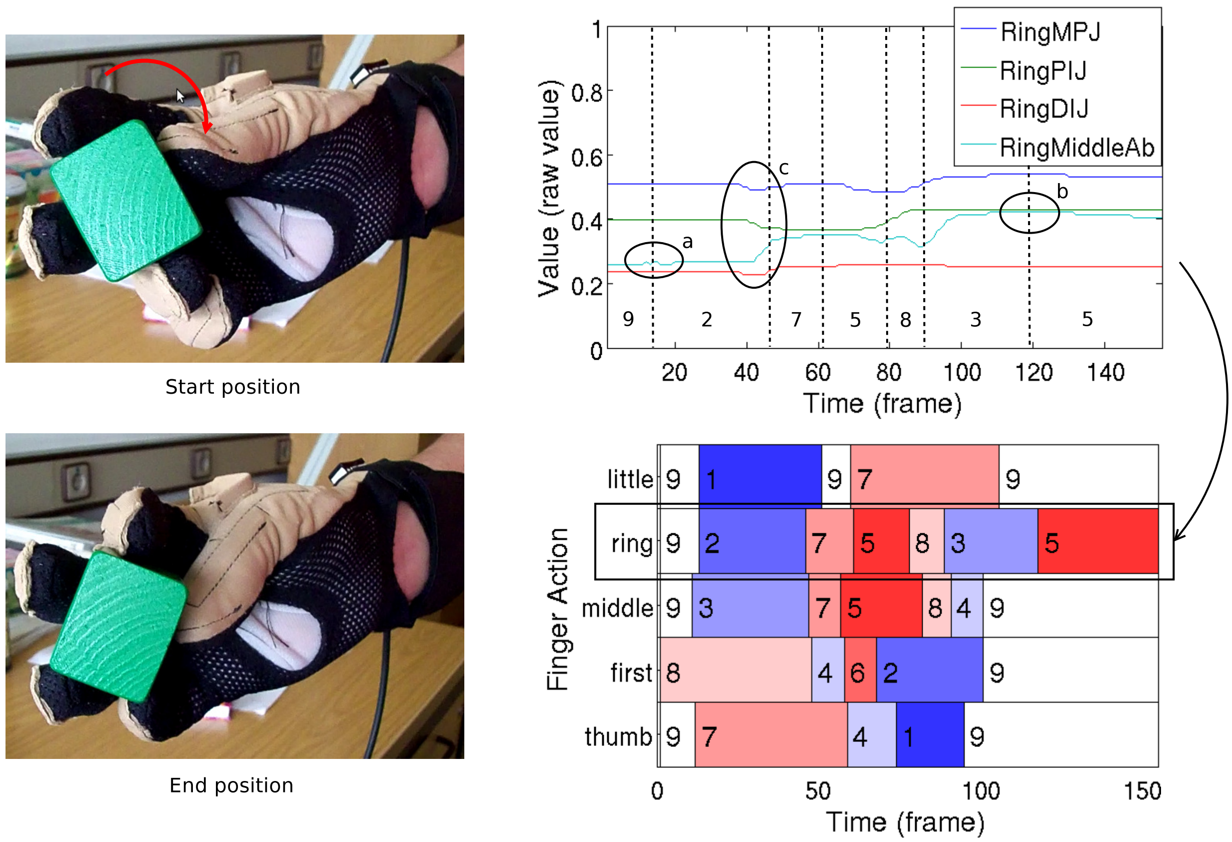


Figure 3.2: Typical data-glove values. We laterally rotate a green cube with four fingers. From the data-glove, we have the joint angle values plotted as the top-right sub-figure and finally extract the action gist chart as the bottom-right sub-figure. In the action gist chart, action gist is composed of the meta motions of five fingers, each meta motion is represented by different color bars with the corresponding type number in Fig. 3.1. The cool colors (blue series) indicate the "opening" motions, and the warm colors (red series) indicate the "closing" motions. The x-axis is a time axis indicating the cyber-glove frame (or sample) number. Here we take the joint values in the ring finger as an example. The joint angles vary over time, we can see some tremors marked as (a). For this kind of signals, we would like to get rid of them. Another practical issue is marked as (b), where the joint angles are flat but we should determine a boundary to separate the different meta motions. Similarly, the angle variation is complex in area (c), but in the end we should give a clear identification. Because of these uncertainties, this chapter proposes a solution to automatically extract the action gist using a Gaussian MRF approach.

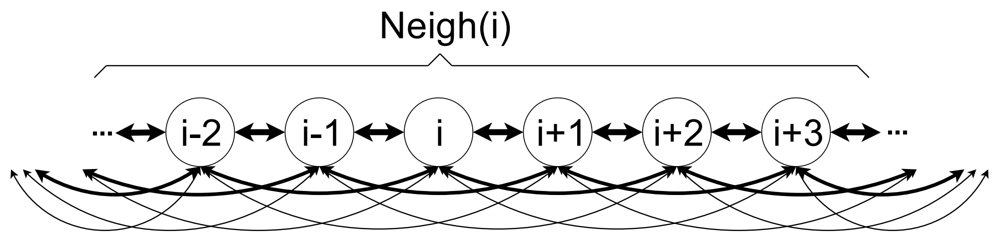


Figure 3.3: Node relationship according to Gaussian MRF. Supposing each data-glove value is a node, then each node is related to other nodes in the neighboring set $\mathrm{Neigh}(\cdot)$. With the impact factor obeying a Gaussian distribution, the line-width indicating the strength of the impact factor, we can see that the neighboring nodes influence with others by distance.

Consequently, a Gaussian Markov Random Field-based algorithm is proposed to extract the action gist of each finger. It can effectively decrease the negative impact from the mentioned issues and provide a concise meta motion sequence.

3.3.1 Gaussian MRF for kinetic meta motion

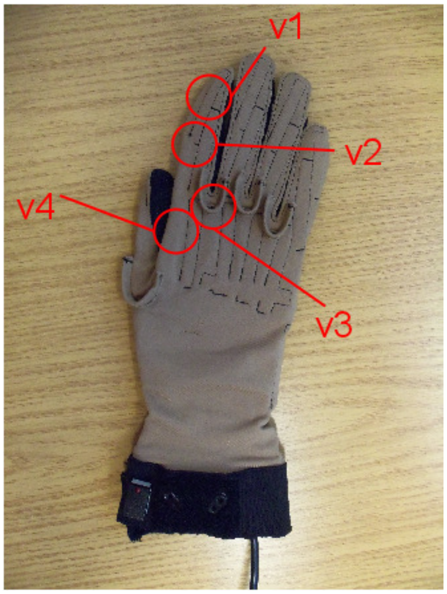
This algorithm considers each value frame sampled from the data-glove as a node, every node can influence the other nodes on which meta motion they belong to. The nearer nodes have the stronger impacts, the criteria are based on the single meta motion similarity and the node distance, the node relationship according to the assumption is illustrated in Fig. 3.3.

The single meta motion similarity of each node can be represented as:

$$I_i^j = \begin{cases} \sum\limits_{k \in \mathbf{F_g}} \left| v_i^k \right| + \varepsilon & , C^j_{k \in \mathbf{F_g}}(v_i^k) = 1 \\ \\ 0 & , C^j_{k \in \mathbf{F_g}}(v_i^k) = 0 \end{cases} \tag{3.1}$$

Here I_i^j is the intensity of node i that is similar to meta motion j, v_i^k is the k th glove value difference (current value minus previous value) in node i. The k th value from the data-glove sensor should belong to one finger $\mathbf{F_g}$, $\varepsilon > 0$ promises the value of intensity is always above 0. This value is not critical but should be a low value, we suggest to fix $\varepsilon = 0.05$ as experience.

Additionally, $C(\mathbf{v})$ is the condition that the finger joint angle difference stay in the range of the corresponding meta motion j. Assuming that there are always four values $v_1, v_2, v_3, v_4 \in \mathbf{v}$ standing for the joint angle variation in five fingers, they are mapped correctly with v_i^k. Commonly, v_1 is for distal-intermediate, v_2 is for proximal-intermediate, v_3 indicates abduction and v_4 is for the metacarpal-proximal angle difference. Specifically, for the thumb values in the data-glove, in order to have a uniform expression, the rotation angle is considered as v_3. Besides, the abduction value v_4 should be adjusted as an identical increasing direction according to the meta motion definition, then the conditions $C(\mathbf{v})$ are listed in the right table of Fig. 3.4. v_1 has a less important effect here because when the object is manipulated, it is easy for the finger tips touching the object to easily create a contra direction with v_2, but v_2 is relatively stable. Whether the finger is open or closed depends mainly on the movement between the proximal and intermediate



meta motion	v_1	v_2	v_3	v_4
1	$\times$ < 0	< 0 $= 0$	$\geqslant 0$	$\geqslant 0$
2	$\times$ < 0	< 0 $= 0$	$\leqslant 0$	$\geqslant 0$
3	$\times$ < 0	< 0 $= 0$	$\leqslant 0$	$\leqslant 0$
4	$\times$ < 0	< 0 $= 0$	$\geqslant 0$	$\leqslant 0$
5	$\times$ > 0	> 0 $= 0$	$\geqslant 0$	$\geqslant 0$
6	$\times$ > 0	> 0 $= 0$	$\leqslant 0$	$\geqslant 0$
7	$\times$ > 0	> 0 $= 0$	$\leqslant 0$	$\leqslant 0$
8	$\times$ > 0	> 0 $= 0$	$\geqslant 0$	$\leqslant 0$

Figure 3.4: An example of the finger joint angle difference in the first finger. v_1 is for distal-intermediate, v_2 is for proximal-intermediate, v_3 indicates abduction between first finger and middle finger, v_4 is for metacarpal-proximal.

joints. In the table, "$\times$" means v_1 can be any value in this condition.

For the data-glove, we have to mention that the abduction angle is not the absolute angle related to the palm. That means v_3 is not working perfectly, but in this study we do not consider it as a critical problem.

When the single similarities of all nodes are calculated, the influence from other nodes can be obtained by:

$$P_i^j = \sum_{t \in \mathbf{Neigh}(i)} I_t^j G(t, i, \sigma) \tag{3.2}$$

where $G(t,i,\sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(t-i)^2}{2\sigma^2}}$ is the typical Gaussian distribution form, $\mathbf{Neigh}(i)$ is the node set near node i (refer to Fig. 3.3). Because the concerned action gist is calculated between each adjacent state pair, the scope of the neighboring nodes is actually set as the entire glove value sequence. Besides, σ is a parameter representing the area one node can primarily impact with, it also means the shortest single motion execution time corresponding to the data-glove sensing speed. Then the likelihood of meta motion j at each node can be compared to find the best meta motion segmentation.

3.3.2 Why MRF, not HMM, MEMM or CRF?

Joint angle values from a data-glove are discretely and sequentially represented. Connecting with current task — identifying meta motion type over the sequences of joint angle values, we easily correlate them as a classical issue: sequence labeling.

In the field of sequence labeling we find many solutions, but the basic ideas are playing around Hidden Markov Model (HMM), Maximum-Entropy Markov Model (MEMM), Conditional Random Field (CRF), and a more general framework presenting these typical models: Markov Random Field (MRF). They can all be described as graphical models, so we can picture their transition graphs as shown in Fig. 3.5. Theoretically, we can enter the data-glove values into every model, and each will return us results as long as we correctly design the corresponding graph model. However, all of these models require the calculation form of the observation joint density:

$$P(X = x) = \prod_{Cluster \in cl(Group)} \phi_{Cluster}(x_{Cluster}) = \frac{1}{Z}\left(\sum_{k} w_k^\top f_k\left(x\left\{k\right\}\right)\right) \tag{3.3}$$

Eq. 3.3 is a common form for MRF. In our case, if we want to know the probability of whether the finger motion should be marked as meta motion x, we should ask a group of data-glove values $x_{Cluster}$ by our design function $\phi_{Cluster}(\cdot)$. Here $cl(Group)$ is a certain set of angle values in the graph $Group$, which is the entire data-glove value set in Fig. 3.3. Besides, the latter part of this equation is a variation of logistic form, the cores are the weights w_k and specific function f_k. Actually, when we use HMM, MEMM or CRF to solve problems, we just need to map $\phi_{Cluster}(x_{Cluster})$ (or w_k and f_k) with a proper form.

We judge the kinetic finger motion by the joint angle variation; but as mentioned in the Section 3.2, the phenomenon “a moment silent in the joint angle variation” frequently happens. In this case, we should not employ HMM or MEMM or other methods based on directed graph. Besides, we should classify the meta motions of each frame by many joint angle values over a set of adjacent frames. This process is quite similar to MRF. Moreover, since we cannot offer benchmark data-glove values because of the measuring uncertainties mentioned at the beginning of this section, we prefer to predefine the relation between each node pair as Gaussian distribution. Therefore, since we are only looking for a good solution to identify the meta motion sequence over the entire data-glove value set, we simply title our method “Gaussian MRF-based”.

3.3.3 Identification of idle meta motion

In addition to the action gist analysis, the idle motion is processed independently from the eight kinetic motions mentioned above. Generally speaking, we can determine the idle motion with many solutions, even the Gaussian MRF-based method can be employed. However, we may get different results in the way of clustering the continuously steady joint values. Theoretically it must be an easy task, but the bottleneck is that we should permit small deviations from fixed values due to the real world sensing.

We suggest several methods, and we can employ them with specific purpose.

The first method is to find a frequent value as high as desired in the sliding window. To realize it, we should have the single similarities of each node to be similar to Eq. 3.1, but the intensity of meta motion 9 at node i becomes as $I_i^9 = 1$ and the condition becomes $C(\mathbf{v}) = 1 \iff \mathbf{v} = \mathbf{0}$. Thus the idle sections can be determined by the following condition:

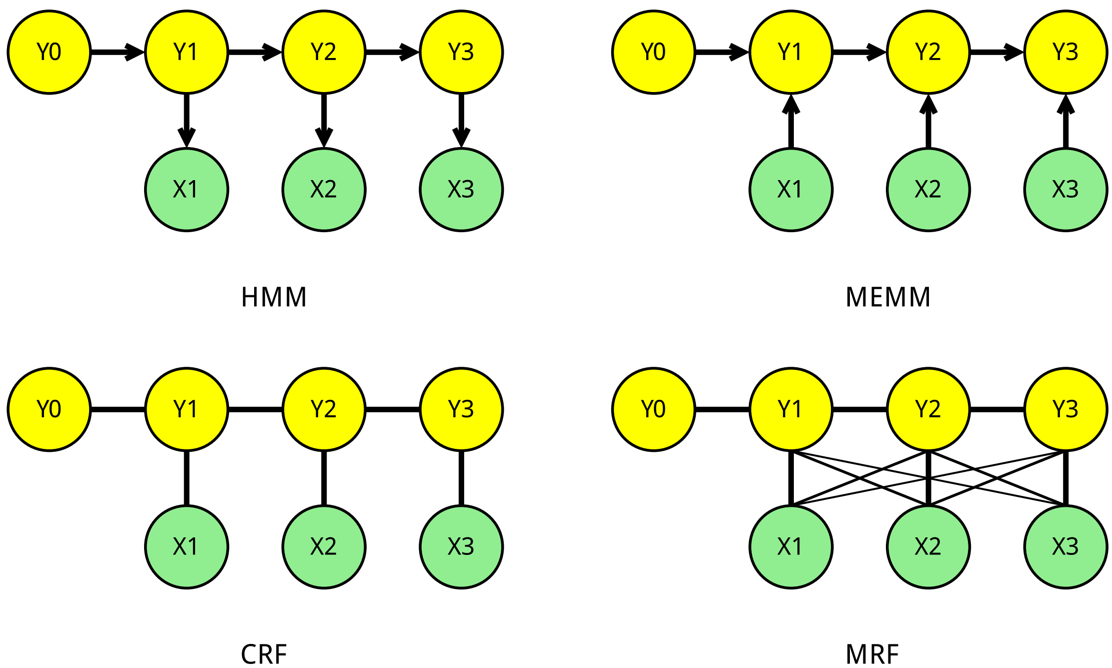
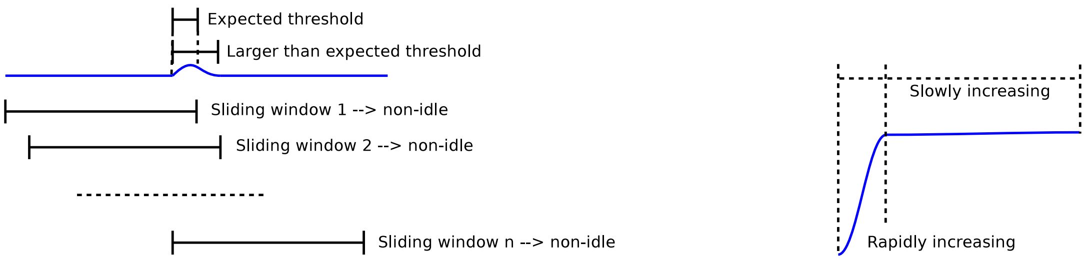


Figure 3.5: Transition graphs of HMM, MEMM, CRF and MRF. Sequence labeling problems are always translated to graphical models. The nodes in a graph are divided into two groups: observations $\mathbf{X}$ and random variables (or say states, output variables, labels) $\mathbf{Y}$. In HMM, we build the relations as transition probabilities $P(Y'|Y)$ and observation probabilities $P(X|Y)$. However, in MEMM, we concern about conditional probabilities $P(Y|X)$ instead of $P(X|Y)$. Thus, HMM and MEMM are directed graphs. When we are not clear about the causalities relevant to the observations and the labels, we should get help from undirected graphs, e.g., CRF. Finally, a more general model is MRF. MRF model actually does not require observations, but for labeling it does have. Since it is a general model, every node in it may have relations with any other nodes. Restricted by the Markov property, node pairs with longer distance have weaker impacts.



(a) Method 1 (b) Method 2

Figure 3.6: Typical defects for the idle motion identification. Assuming that the data-glove values are one-dimension, for method 1 in sub-figure (a), the algorithm cannot avoid to annotate idle to the sliding window including a varying interval larger than the threshold. Although we can find a solution annotating most frames as idle, we have to accept the over-long varying interval. Also, we can find a special case to point out the disadvantage of method 2 in sub-figure (b). Suppose that the value is increasing, but in a long term on the latter part it climbs quite slow and acts like a horizontal line. If we use method 2, it is probably considered as idle motion. However, visually this result seems strange.

$$\sum_{t \in \mathbf{Neigh}(i)} I_t^9 > threshold_1 \cdot d_{sw} \tag{3.4}$$

Thus node i stays idle when the sum of single intensities is larger than the threshold. Here, $\mathbf{Neigh}(i)$ is set to be at the range of d_{sw}, which is the size of the sliding window, then d_{sw} nodes are taken into consideration to find the idle section. In addition, all adjacent idle nodes are merged as an idle section, but if the length of idle sections is shorter than a single motion execution time σ, this section should be considered as not idle. Commonly, we find that $threshold_1 = 0.90$ and $d_{sw} = 20$ fit most cases when the glove fequency is about 20 fps.

The sliding window moves from the beginning to the end of the glove value frames. This process has a drawback as shown in Fig. 3.6 (a): when small waves larger than $threshold_1 \cdot d_{sw}$ block the way of a long and straight values, we have to mark them as two separate idle motions and leave a kinetic meta motion in the middle.

If we expect to have a more tolerant solution, we should consider a second method: assign the minimal length of the idle frames to $L_{idle\,min}$. For each frame (node), we expand its corresponding neighbors and count the non-idle nodes to make a ratio as:

$$r_{idle}^i = \frac{\sum_{t \in \mathbf{Neigh}(i)} I_t^9}{the\ length\ of\ \mathbf{Neigh}(i)} \tag{3.5}$$

Hence we introduce another $threshold_2$ to determine whether the frame should be idle motion. According to our test, $L_{idle\,min} = 15$ and $threshold_2 = 0.1$ are suitable for 20 fps data-glove. However, nothing is perfect, we can also give an example to illustrate its weak point as shown in Fig. 3.6 (b).

Idle motion indicates when the fingers have a break in the process of in-hand manipulation. Nevertheless, for an object operation, our main goal is to identify 8 kinetic meta motions to

Figure 3.7: Rotating a star prism. Thumb, first, middle and ring finger are used to rotate the block. In the stable moment, four fingers pinch the grooves. For rotation, each finger reaches the neighbor groove. This process is defined as one trial, and we will see the hand pose at the beginning state is quite similar to the end state.

describe the features of a manipulation application. We can consider the disqualified kinetic sections as idle when the corresponding joint angle variations are not significant enough.

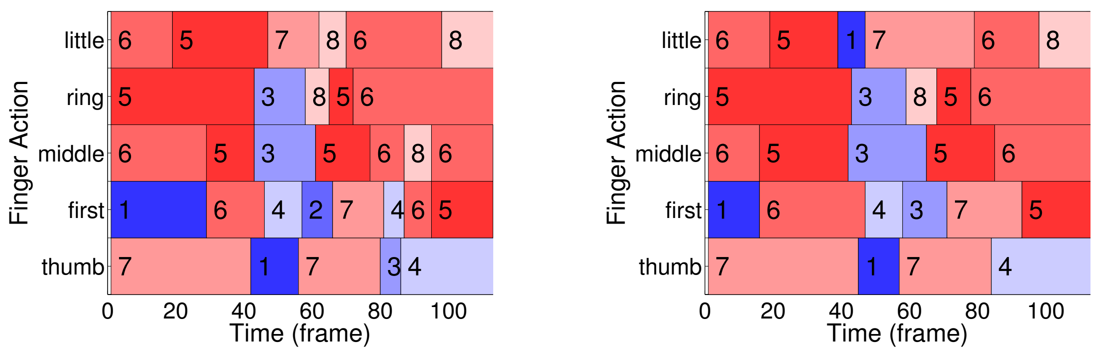
Therefore we have the third solution to simply discover some kinetic but actually idle motions. The principle is: for each interval of the identified kinetic meta motion, check the maximum and minimum joint values. When their difference is lower than $threshold_3$ (usually lower than $0.5°$ degree), we rename it as idle motion. This simple principle is also involved in previous two methods to decrease some misjudgements.

3.4 Experiment

Using the proposed solution from Section 3.3, the action gist can be extracted from the raw data-glove values. Different kinds of objects are used to examine this method, the hand movement ranges from *clearly moving without object* to *complex finger gaiting*. The corresponding action gists are extracted from the glove values automatically, and they all agree with our expectation.

3.4.1 Action gist from different parameters

Among the parameters in the algorithm, σ is the only one depending on the application except from those values determined by experimental experience. As we mentioned in the previous section, this parameter is relevant to movement speed and data-glove framerate. Higher σ merges more short terms, but meanwhile we risk losing critical short meta motions (see Fig. 3.7 and Fig. 3.8). On this point the value selection should be considered carefully. We enumerate all possible values and compare the extraction results for several typical applications, finally we find that the configuration of σ is not so strict. For most cases the extraction results are the same, otherwise the length of meta motion changes slowly with σ variation. Thus it is not necessary to check every possible value (e.g., from 1 to 20 for 20 trials), instead, we can set $\sigma = 5$ to acquire the details, and then set $\sigma = 10$, 20 or even more to get the general context.



(a) $\sigma = 5$ (b) $\sigma = 10$

Figure 3.8: Action gists of star prism rotation corresponding to Fig. 3.7. For different value of σ, we can find commons and differences between the two action gists. Therefore, when we need the detailed motions, we should set σ to a lower value. Meanwhile, a higher value for σ may risk losing tiny but important meta motions.

3.4.2 Action gist from similar objects

In order to make an intuitive example, here we take *bottle-screw-cap unscrewing* as an example. There are four different-sized bottle-screw-caps as shown in Fig. 3.9. A participant rotates the caps by four fingers many times. After several trials we list several typical results through the proposed method as in Fig. 3.10.

There are countless ways to move the fingers to reconfigure the object achieving to the goal state, but for the transfer to a robotic hand one solution is enough to guide the manipulation. Furthermore, by the result of bottle-screw-cap unscrewing we can see: for different-sized caps unscrewing there exist similar action gists. Therefore, we can demonstrate the scenario-specific finger-gaiting movement many times and find the popular action gist. In this case, the common one has stronger adaptability for the robotic hand, which is of different size from the human hand, to complete the manipulative task. Further solutions can be found in Chapter 5.

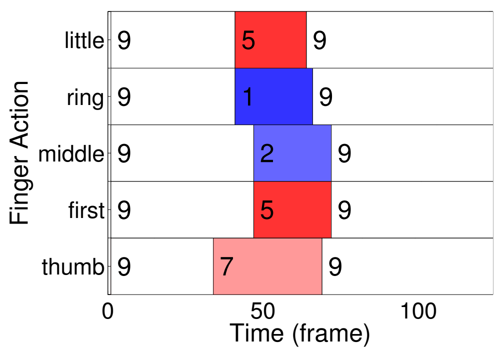
3.4.3 Action gist from reverse movement

Since we have the experience with the bottle-screw-cap screwing, we are now wondering whether the clockwise and the anticlockwise movement have completely opposed action gists. Therefore, we make a try as shown in Fig. 3.11. Commonly, from Fig.3.1, the reverse meta motion pairs are "$1 \leftrightarrow 7$", "$2 \leftrightarrow 8$", "$3 \leftrightarrow 5$" and "$4 \leftrightarrow 6$". For the reverse movement, we should theoretically find a sequence of reverse meta motions. However, the fact is that there are many ways to manipulate an object, we may see the expected result, or many similar results.

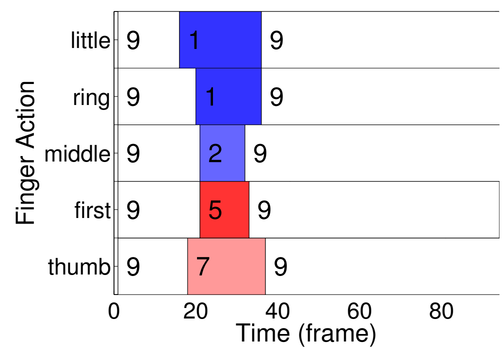
3.4.4 Action gist from the same manipulation skill

We have experienced objects with the same shapes but with different sizes. Now we have interest with the skill itself, e.g., rotation. We have pair of action gist extractions over different shaped

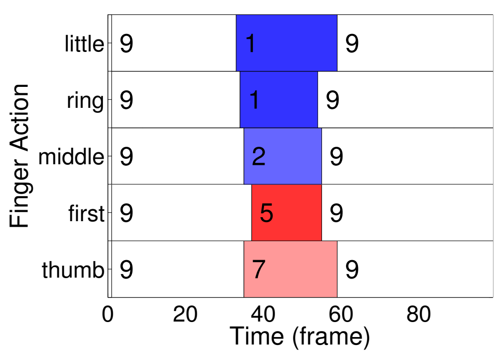
Figure 3.9: Unscrewing different-sized bottle-screw-caps. The thumb, first, middle, and ring finger participate in this scenario. Each screw cap is rotated as around $90°$ anticlockwise, and this process is defined as a trial.



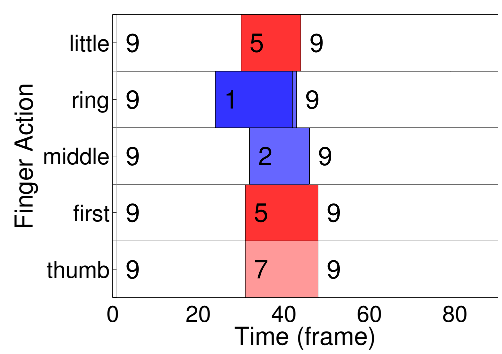
(a) Big



(b) Medium



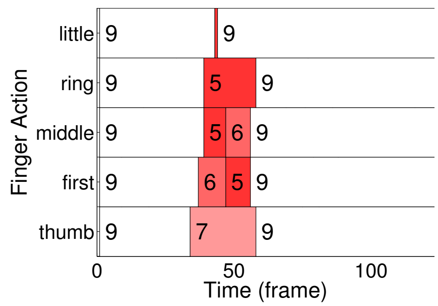
(c) Small



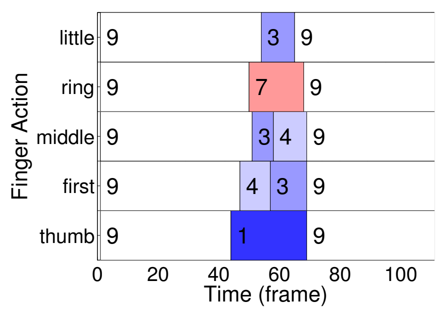
(d) Tiny

Figure 3.10: Action gists of bottle-screw-cap unscrewing. We believe that there are not many finger motions, so we take idle motion into consideration (we use method 2 to maximally identify the idle motions). As a result, we find that the action gists from 4 trials look similar. The common meta motions are motion 7 in the thumb, 5 in the first finger, 2 in the middle finger, and 1 in the ring finger.

(a) Direction

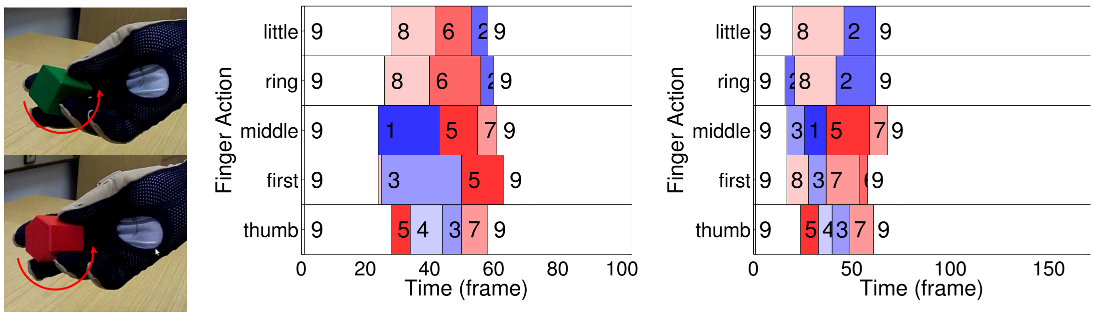


(b) Anticlockwise



(c) Clockwise

Figure 3.11: Bottle screw cap unscrewing clockwise / anticlockwise with three fingers. The thumb, first and middle finger join in the movement. We take many times of screwing separately, and find the best matched results. The movement of the thumb finger match quite well ($1 \leftrightarrow 7$). For the first and middle finger, the meta motions are corresponded but they are not perfect. According to the happening order, “5, 6” should correspond to “4, 3”. Because of this we check the raw data, and find there are several subtle joint value variations disturbing the extraction. We should not get rid of them because they are produced by the demonstrator. Anyway this result does already make sense for a fact: the meta motions in an action correspond to the reverse action.



(a) Prism Rotation (b) Rectangular prism (c) Hexagonal prism

Figure 3.12: Rotation skill for rectangular and hexagonal prisms. We use the thumb, first, middle, and ring fingers to rotate the prisms anticlockwise and extract the correspondent action gists from multiple demonstrations. Among the results we find the most similar action gist as illustrated. Consequently, we can generally believe that the fingers obey some similar rules to rotate an object in the size shown in the photos.

objects as shown in Fig. 3.12. Since we can find similar action gists when we are applying the same manipulation skill on the similar sized objects, we believe that we can use the action gist techniques to explore more interesting result with human hand behaviors.

3.5 Summary

This chapter concentrates on action gist extraction from the demonstration of in-hand manipulation recorded by a data-glove. Different from the manipulator trajectory planning, this model works in a fuzzy way to represent the finger movement. It gives the manipulator a related loosely explored space to implement the task, and from the view of human in-hand manipulation, it is more similar to the mechanism of the human hand.

A discussion is triggered by the number of meta motions. Depending on the partition, it is possible have more angle intervals to more explicitly present the finger moving range. In this case, we may have 16 meta motions, 48 meta motions, etc. However, based on the joint angle variation, we only concern the direction is positive or negative. In this way, 8 meta motions is the exact number. Compared with only “open” and “close”, 8 meta motions offer more criteria to describe the manipulation process. Besides, they are easily identified or memorized by human, so we can interfere and improve the extraction result. Therefore, currently we use such kind of definition.

The model currently is built from the value of a data-glove. One disadvantage is that the human demonstrator wearing the data-glove has a different feeling, executes the movement unnaturally and difficult manipulation applications are hardly handled. Hence to fuse the result from other sensors, such as visual sensing, is another direction for developing the model further. Another drawback is related to the four abduction angles in the data-glove, which are angles between two fingers, not the absolute angle related to the palm. From this point, we do not guarantee that the finger movement perception is always correct. However, according to our long-term experience with the proposed method, the extracted results are reliable. For most cases we compare the extracted action gists and the manual annotations, the meta motion sequences are the same.

In the later chapters, the action gist model is being examined by simulation and real robot tests. When the action gist is mapped back to robotic hand control, it is supposed to work as guidelines because it provides the meta motions of each finger in order. Different-sized hands apply different joint angles to execute the manipulation, but the corresponding meta motions remain the same to indicate the finger moving directions. In every trial we give the robot quantized parameters according to a fixed meta motion sequence, and through iterations the parameters are refined to ensure the correct state transition.

Chapter 4

State Gist of In-hand Manipulation

Contents

Similar to the "gist" of in-hand manipulation actions, the "gist" of states is the key information extracted from raw sensor data. State gist concisely describe a sequence of "what happens" accompanying with action gist. As our plan, we pair the state and action gist for an in-hand manipulation task.

Since state gist is the essential information accompanied with the hand movement, we have two concerned issues: How to collect the information and how to abstract them. For the collection we need sensors, for the modeling we need a concrete model and some mathematical techniques. We have introduced the related work in Chapter. 2. Thus we only refer to our ideas in this chapter.

Generally we concentrate on answering following questions:

- What elements can be considered into state gist?
- How can we acquire the element of state gist by sensors?
- How can we visualize state gist in reality?

Above questions correspond to the coming sections. Hence, we will see "the definition of the element of state gist", "the relations among the sensors, algorithms and state gist", "some experimental results on state gist extraction", and finally a brief summary.

4.1 Meta criterion

In order to make an independent name, we would not like to entitle the basic element of state gist "meta state". Since we use state to check whether the actions work correctly, we think the word "criterion" can properly express our idea. As a result, an in-hand manipulation action gist consists of meta motions, meanwhile its corresponding state gist consists of meta criteria.

Since every manipulation involves a target, i.e., we do have some purpose to move the object, there must be criteria in this physical world. The criteria could be relevant to the position and position of the hand and the object, the interactive contact, or the appearance of the object. Furthermore, we notice that there are more complicated manipulation not counted yet, which depend on other factors away from previous criteria. For example, in the scenario of writing, we manipulate a pen and see characters generally appear on the paper; in the scenario of guitar playing, we hit the chords and hear the sound; in the scenario of keyboard typing, we press the button and find the words on the screen. From these cases we can feel that a manipulation always involves more than one object because of its specific target. We use a pen filled with ink, so when the pen sweeps over a paper we can see the corresponding calligraphy. If the aim of the manipulation is "write my name on the paper with a pen", the criteria should be involved with whether the letters are "my name", whether the letters appear on the paper, and whether the letters are produced by the pen held in the hand. Once all of the criteria meet the target of the manipulation, we can conclude that the trial is successful.

Generally speaking, the criteria of a manipulation task can be any sentence in the form of "*subject is ...*". Here *subject* is an involved object. The entire sentence makes no difference with normal natural language but we will finally enter the description into the robot. We suggest to separately describe each object and parametrically express its current state. For the criteria assignment we prefer comparative relations (e.g., larger, higher) to absolute values (e.g., 23 degrees, 15 cm) unless they must be the final achievement. This is because our job at the moment is extracting the knowledge, and later we will apply the knowledge to a slight different environment with another (robot) hand; An implicit value reduces unnecessary robot control.

For an manipulation task, we merge and organize the meta criteria and meta motions into a schedule as the form similar to the classical state-action model as Fig. 2.3 in Chapter 2. Besides, we are certain about that there are two kinds of meta criteria sampled through the in-hand manipulation process: instantaneous and continuous criteria. The instantaneous criteria record the necessary state at every sampled moment in the current stage, e.g., touched or not, around a certain position, the shape remains the same. Furthermore, the continuous criteria imply the continuous transition of the corresponding state from the previous stage to the current stage, e.g., moving from the left to the right side, clockwise rotating, liquid becomes fewer inside. In order to explain the connection of meta motions and meta criteria better, we give an example as Fig. 4.1. The state-action gist of this example is generated by specific sensors. When we have another set of devices, we may have different discovery.

For an manipulation task, moreover, we suggest to design the meta criteria as simple as possible. The intuitive factor is that simple criteria are time saving, but further we have another two reasons. One reason is related to that we have two steps learning. In each step we sample real-time information and translate them to criteria. Simple but robust criteria promise that we can faster match them and make a correct evaluation on the task. For the complex criteria, we

Figure 4.1: An example of state-action gist. Suppose that we are now trying to open the tab of a can with the thumb, first and middle fingers. After extraction, we have such form of meta motions and meta criteria. We should claim that there are countless methods to open a can, this example is just a case of this in-hand manipulation skill. Depending on the trigger time of the motions, we have four stages, and meanwhile we collect the meta criteria from available sensors. For the meta motions, having the start time is enough. This is because we extract meta motions sequentially; the moment that a motion starts at a finger, indicates the previous motion stops at that finger. For the meta criteria, we mark them at the moment that the next meta motion is triggered (or the end of this manipulation). In this case, the instantaneous and continuous criteria step-by-step present the necessary achievements in the manipulation task. Specifically for this example, we keep the thumb and the middle finger fixing on the can, so the contact states of these two fingers keep "in touched" in all stages. Besides, we use the first finger to open the tab on the can, so we see the first finger approaches to the can in stage 1. Last but not least, for this scenario, our aim is to open the can by moving its tab, so in the stage 3 and stage 4 we find the tab is moved.

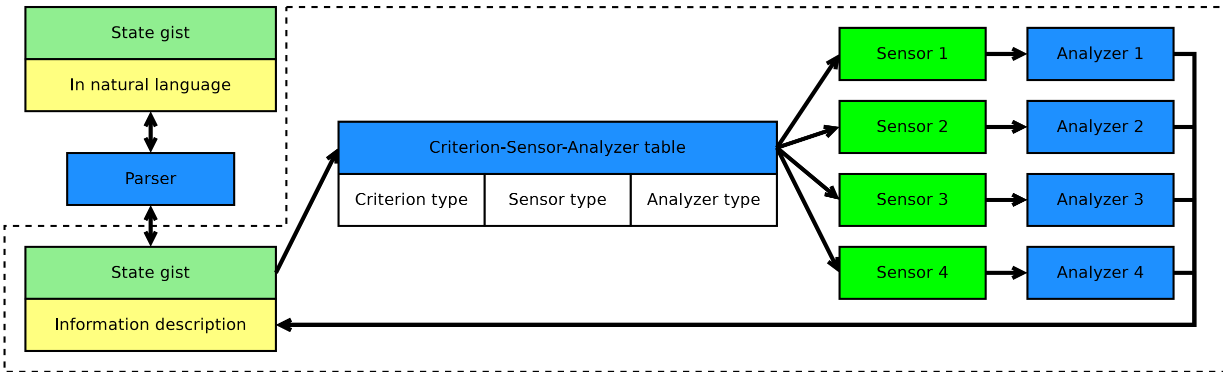


Figure 4.2: Ideal framework of state gist perception. This thesis concentrate on the bounded framework. Different meta criteria require different sensors and analyzing algorithms. Therefore, beforehand we should prepare a looking-up table. Depending on the specific criteria we refer to the sensors and analyzers.

have to spend extra time on designing how to match the corresponding features. The worse is, the complex criteria extracted from the demonstration cannot perfectly match whatever we extract from the practice, so we lose confidence on the manipulation evaluation. We take SIFT feature for an example. For this kind of image features, it has requirement on illumination even though it describes object appearance quite well. Once the illumination quality is quite different between the demonstration and the practice, we cannot use SIFT features for criteria. The other reason is related to robot reaction. As we know, in-hand manipulation is applied to the objects in this dynamic world. For each step, we have a certain time slot to execute the hand motions. When the criteria extraction consume too much time, we will miss the chance to give reaction and have to make another trial. Therefore, unless we have specific target, we prefer simple criteria. So far for the solid object manipulation, we consider position (related position, e.g., near the center of the viewpoint, move from the left to the right side), posture variation (e.g., no change, clockwise rotated along the long axis), and fingertip contact (e.g., touched or not touched) as criteria.

Meta criterion cannot leave perception. Thus in the next section we discuss how to bridge the sensors and state gist extraction.

4.2 Relationships between sensors and state gist

In our ideal plan, we can use our natural language to describe any state gist of a manipulation task. By a parser we informationize the description and combine them with action gist. However, natural language processing is a big research branch in interdisciplinary research fields, so this thesis has not enough concentration on it. Instead, we picture the complete framework as Fig. 4.2 to have an overview and use it to guide our future work. Since gist needs perception but different criteria require different sensors and corresponding extracting algorithms, we should be clear about their details in advance.

Nowadays we can find many sensors but talking about their performance is out of the scope of this thesis. Besides the performance of the sensors, we care about the relations between

Table 4.1: Perception feasibilities of sensors for typical meta criteria

Sensors	**Conventional Camera**	**RGB+D Camera**	**Tactile Sensor**	**Positioning Sensor**	**Inertial Measurement Unit**	**Data Glove**	**Bio-signal Sensor**
Object Position	yes	yes	no	yes	no	no	no
Object Orientation	yes	yes	no	possible	yes	no	no
Object Shape	yes	yes	no	possible	no	no	no
Object Color	yes	yes	no	no	no	no	no
Hand Position	yes	yes	no	yes	no	no	no
Hand Posture	possible	possible	no	yes	yes	yes	possible
Contact	possible	possible	yes	possible	no	no	yes

sensors and criteria. Therefore we make a table as Tab. 4.1 to list the feasibilities of several kinds of sensors for the typical criteria. There are still many sensors or criteria not listed on the table, such as microphone for sound, thermometer for temperature, since the content of the word “in-hand manipulation” widely involves most interactive scenarios in our daily life. As a result, we have to simplify this topic and make some explanation for the given table. We mark “yes” or “no” to indicate whether the corresponding sensor supports the specific criterion extraction. When we mark “possible”, it indicates that originally the sensor is not designed for such kind of criterion perception but by special processing it does fully (or partly) work. For example, even though positioning sensor only offers (x, y, z), if we assign an extra positioning sensor at another fixed area, it is possible to get the object orientation; for the hand posture, because of occlusions we cannot discover all finger links in every case, but we can employ inverse kinematics to recover the entire pose; the contact information is not the job of cameras, but thanks to inference techniques, when we have the spatial information of the hand and the objects we can estimate their relations. From the table we can see that some sensors are able to deal with many criteria perceptions, but meanwhile some dedicate to the specific criteria. Usually, the special-purpose sensors perform better than the all-purpose ones, e.g., straightforward sensing, easier to code processing algorithms, higher precision.

When we apply multiple sensors to in-hand manipulation perception, the belief of criteria becomes an issue. For example, if we know from stereo cameras that the object is moved 4 cm off meanwhile our Kinect tells us the distance is 6 cm, which result shall we trust? The results root from many factors, including sensor performance, processing algorithms, etc. Our solution is to keep both values and label the criterion with its corresponding sensor. Moreover, we suggest to only use one kind of sensor for one criterion, except that the sensor is used as a bridge to connect other dedicated but more accurate sensors. When the result from “bridge sensor” conflicts with the result from “dedicated sensor”, we prefer to trust the result from “dedicated sensor”.

There is no ideal sensor, so the performance of a sensor is a critical factor of meta criterion perception. Firstly, we concern about their precisions.

In-hand manipulation can be considered as typical spatial hand-object interactions. There-

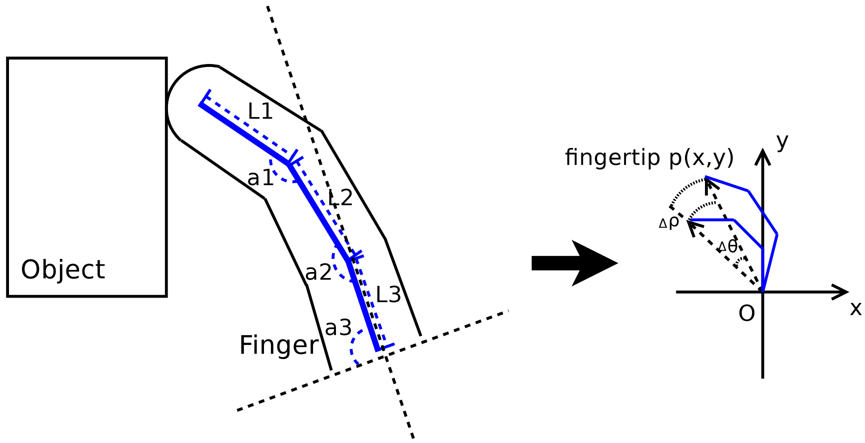


Figure 4.3: A two-dimensional example of typical hand-object interaction. We try to move the object with one finger. Theoretically, the finger has three links, as long as $L1$, $L2$ and $L3$, where the joint angle could be defined as $a1$, $a2$, and $a3$. With the simplest mathematical form, we can imagine that the fingertip is a point in the planar coordinate.

fore, when we decompose everything into particles (e.g., to divide an object into molecules, and then atoms), most criteria are caused by particle deviations, i.e., distance variation. In the in-hand level, the object has to be controlled in the available space where the hand components can reach. Normal hand is about 20 cm, so the sensors should be sensitive enough within this scope. In order to make a better explanation, we prepare a 2D example of typical hand-object interaction as Fig. 4.3. From the figure, we can see that the object is solid, as well as the finger link lengths are fixed. In this case, when we want to move the object, we should vary the finger joints. Since a normal human hand has fixed size, we can parameterize the joint angles to see how the fingertip changes its position. The finger is considered as a three-links chain, so the fingertip displacement obeys the transformation from the base link to the end link. Because we are not studying the error propagation of each joint angle, we can simplify this system into a minimal case: moving a joint, and the fingertip position is moved. Suppose that the current distance between the base and the tip is ρ, and after $\Delta\theta$ turning it varies $\Delta\rho$. Now as the law of cosines, the moving distance from the original to the current fingertip position is:

$$d = \sqrt{\rho^2 + (\rho + \Delta\rho)^2 - 2\rho(\rho + \Delta\rho)\cos\Delta\theta} \tag{4.1}$$

If we are using a data-glove, the error could be joint angle value. Depending on the joint position, ρ has multiple assigned values. We select several typical values for ρ and analyze the distance error caused by joint angle error as Fig. 4.4. Particularly, when we use a byte to store a value ranging from 0 to 90 degrees, the angle resolution can be calculated as $90 \div 256 \approx 0.35$. In this case, we inevitably has 0.35 degree error when the data-glove is inaccurate. Fortunately, when 0.35 degree error happens, retrieved from Fig. 4.4 we find that the error is always less than 0.05 cm. Thus, the presumed angle resolution is acceptable. Besides, We also notice a warning threshold from Fig. 4.4. Since we are dealing with in-hand level application, we would better select a sensor with an angle error in 2 degrees.

Another kind of error disturbing in-hand manipulation criteria is related to the positions in-

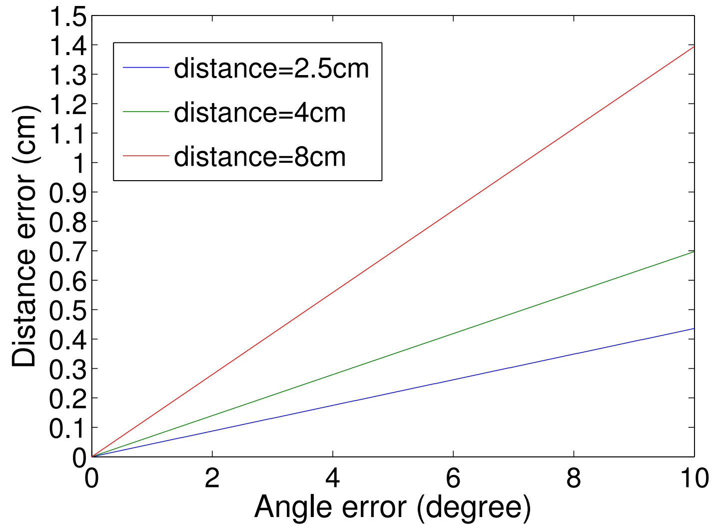


Figure 4.4: Distance error caused by joint angle error. As Eq. 4.1, we calculate the distance error as assigning $\Delta\theta$ from 0 to 10 degrees. For an in-hand manipulation task, we would better keep the distance error in 1 mm. Therefore, the angle error would better keep in 2°. This figure also provides with another kind of hints: the moving distance with respect to the joint angle variation. For example, the blue (lowest) curve indicates the distance from the fingertip to the end, i.e., in Eq. 4.1 $\rho = 2.5$ cm. When it turns 10° the fingertip moves about 0.4 cm.

ferred by conventional or RGB+D cameras. As Fig. 4.5, the three-dimensional world is imaged in a planar chip. Even though we have not yet stepped into multiple cameras or depth sensors to get the third dimensional information, there are already two dimensions of information involved position error. Additionally, if we take the solution of multiple cameras, position error will also interfere depth calculation.

The position error is resulted in the error of image processing algorithms. Usually we count image features starting from pixels. By the color or the illumination of the pixels, processing algorithms group the interested pixels. Because the grouping rules rely on the color values, it is possible that similar-valued and adjacent pixels are collected together but actually they are wrong. As a result, we have to analyze how the position error does matter. A simple method is comparing the imaging chip resolution and the size of the view plane. For example, when we have a camera with a width at $w = 1024$ pixels and concern the manipulation scenario happening as wide as $W = 50$ cm as Fig. 4.5, a pixel error will be $W/w = 50/1024 \approx 0.0488$ cm position error. For a normal in-hand scale error 1 mm, the algorithm error should better be kept in 3 pixels.

There are many sensor errors we have not discussed yet. When the root is spatial issue, we can analyze the specific case similar to above processes: find the relation between the sensor error and spatial error, so as to estimate which sensor error is allowed. For another error species such as color and contact intensity, our solution can be establishing a statistical model and evaluate its stability according to specific benchmarks. For example, we can let the object sit quietly and apply Gaussian model to learn the object color, when the color noise is not acceptable we should

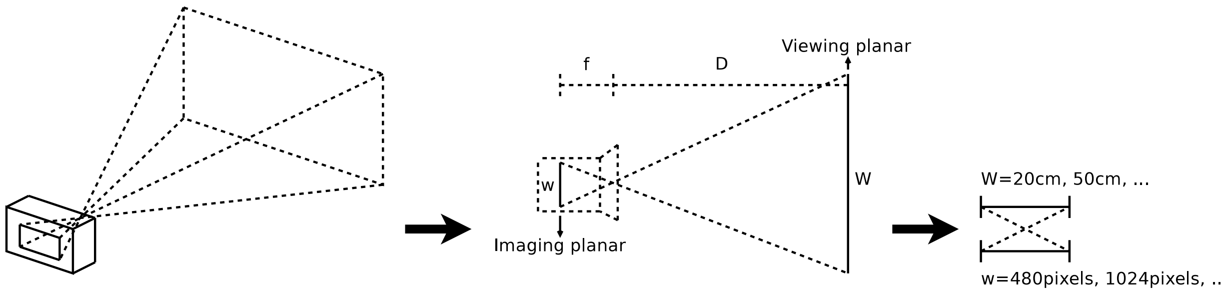


Figure 4.5: Typical camera imaging. By camera lens, the three-dimensional world is scaled down into an planar imaging chip of a camera. We can turn the imaging process into a form of two-dimension, what we concern are the imaging size w, the size of the real view W, the lens focus f, the distance between the view planar and camera D. Obviously, $w/f = D/W$. Furthermore, w is in pixels, so the image in w pixels wide corresponds the view in W cm.

employ another camera; for the tactile sensor, we can put 10 gram, 20 gram, and other standard weights on the sensor skin to decide whether we will use this sensor.

Because sensor error always exists, we prefer to use variation tendencies as features. They are generated from multiple samples, statistically they should be more stable than single sample.

Besides sensor precision, as long as we employ multiple sensors, another point we concern is synchronization. In practice, computing resource is assigned to control sensors, and it is possible to use multiple computers to govern sensors. When the sensor data stream into the computers, usually we will attach timestamps in order to align the data from different channels. As a result, after we extract the corresponding meta criteria or meta motions from the data, we can clearly sort them. On one hand, we should have time synchronization for the applied computing units. This point is a public research topic, so for further reading we can refer to the literatures such as (Sundararaman et al., 2005; PalChaudhuri et al., 2004). On the other hand, we have an issue with respect to sensor frequencies. Generally, a sensor captures corresponding data as a certain frequency. The frequency is adjustable, but commonly it has minimum and maximum, or several optional values. As Fig. 4.6 tells us, different sample rates of the sensors leave uncertainties to align the meta motions and meta criteria. For this issue, we have three solutions. The first one is to trust the nearest sample, and this solution does work when the criteria sensor has higher sample rate than our data-glove. According to the Nyquist–Shannon sampling theorem, a better situation is that the criteria has more than twice sample rate of the data-glove. The second solution is to assume the sample rate of our data-glove is the lowest one, and employ the criteria sensors only if they can achieve an integral number of the lowest sample rate. The third solution is to resample all of the value sequences into new sequences as a standard size. For example, originally a data-glove sensor offers 50 samples and a tactile unit provides with 400 samples, now we interpolate the sequences so as to have 100 data-glove samples and 100 aligned tactile samples. Even tough interpolation or smoothing algorithms may destroy the originality of the data, for the sake of more selection on the sensors, we choose the third solution. Additionally, we sample action gist in the stage of demonstration, so we concern data-glove sample rate. When we are in the stage of practice, a data-glove is unnecessary. At this moment we update the meta criteria with latest sensory information.

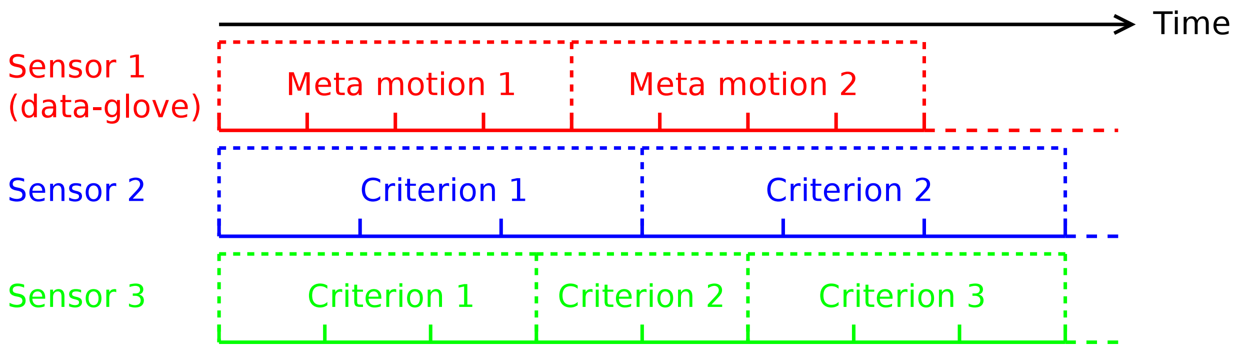


Figure 4.6: Different sensors have different sample rates. Suppose that currently we have three sensors containing a data-glove, and their sample rate is different. As a result, we have 9 samples for Sensor 1 (data-glove), 6 samples for Sensor 2, and 9 samples for Sensor 3. Even though we have a clear boundary of the meta motions and the meta criteria, we are not certain about when a meta motion ends what the meta criteria are because we do not have any sensory sample at the corresponding moment. Therefore, our solution is to determine a standard scale, all data sequence will be remapped to a new sequence as the standard length.

Most content in this section are relevant to technical details. However, we have not yet seen how to extract the state gist from any specific sensor. Therefore, in next section we will demonstrate how to do it.

4.3 Practical results

Since there are multiple sensors taking part in the in-hand manipulation demonstration, we give an overview of them and then extract corresponding state gists. By the way, we have to emphasize again, that state gist could be involved with countless kinds of criteria. What we show in this section are the ones we have experience with.

4.3.1 Experimental configuration

Before we start, we count the involved sensors: webcam (monocular camera), stereo camera, Polhemus sensor (6DOF motion tracking, including three-dimensional position and orientation), data-glove, and Tekscan sensor (external tactile arrays). We can mount them all on the hand as shown in Fig. 4.7. However, according to applications, we usually select several necessary sensors mounted on the hand. This is because too many sensors are burdens for hand movement.

Based on the current setup, we have four scenarios recorded. They are shown as Fig. 4.8. Generally, they are all belong to rotation movement. However, the grasp points are obviously different so the fingers must follow different trajectories to handle the objects.

There are many details about how to process the sensory data but we will skip them and only introduce how to get the state gist.

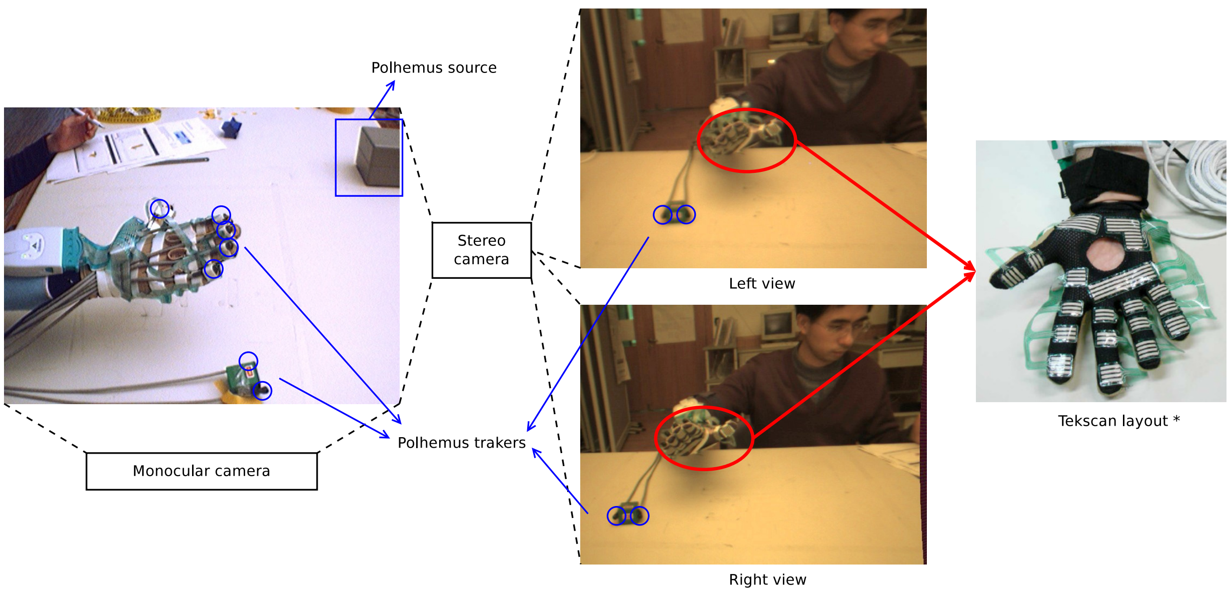


Figure 4.7: A possible experimental setup with multiple sensors. Firstly we wear a Cyberglove, and then we attach Polhemus trackers on the back sides of our fingertips and Tekscan arrays on the front side. Polhemus sensors perceive their positions and orientations by referring the Polhemus source at a certain distance. According to our arrangement as the photos, we are capturing the movement from the hand and the object. In another respect, we note that the photo illustrating the Tekscan layout is borrowed from HANDLE Deliverable 6 (Handle-project, 2009b), but we adopt this layout in the experiments so as to widely capture the contact information on the hand area. Besides, on the lateral side we apply a monocular camera, meanwhile we put a stereo camera opposite the demonstrator. Obviously we find the photo colors are quite different but this phenomenon always happens when we use multiple cameras. In this case, we should try to avoid using color information, or only extract the meta criteria from one camera. Any way, there is a major usage of the cameras that no other sensors can work instead: they provide us with the straightforward impression on the manipulation trial. In this case, we can properly trim the unnecessary data based on the synchronized photos. From the photos we also notice that the entire hand is occupied by the sensors from the fingertips to the wrist. The reality is: when we wear too many sensors on the hand, it is unnatural to perform many kinds manipulation movements with a hand. Therefore, we employ the sensors according to our application. On the other hand, with the technical development, we will have fewer and better sensors.

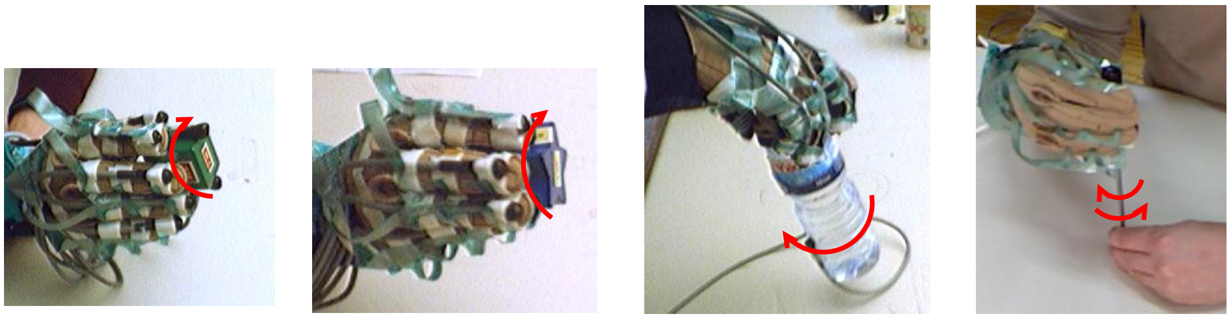


Figure 4.8: Recorded scenarios corresponding to the experimental setup. We have recorded four scenarios, all of them are rotation movement but finger gaitings are different. Specifically, screwing movement does not employ Polhemus sensors.

4.3.2 State gist extraction

Meta criteria from Polhemus sensors

Due to the mechanical reason, we have to analyze the Polhemus data only from the scenario of bottle rotation. In this scenario, five Polhemus sensors are attached on the fingertips and one is fixed near the wrist, as well as one is attached on the bottle. Besides, the frame rate of Polhemus is set to 14 Hz.

From the Polhemus, we "roughly" know the positions and orientations of the fingertips, the wrist and the object. We use the word "roughly", because from the spatial view the Polhemus sensors do not stay exactly where the tracking targets stay. However, they have topological relation and we are probably able to translate their positions.

Based on our "as simple as possible" rule, we extract the related positions information as Fig. 4.9. From the figure we can manually infer a meta criterion related to the position variation: the object keeps a certain distance with the wrist. This criterion can be detected automatically, as long as we input a condition to the analysis program:

$$\sum_i |d_i - d_{i+1}| < threshold \tag{4.2}$$

where d_i is the distance we concern about.

From Fig. 4.9 we notice that the wrist-bottle differences of x,y and z-axis are varying. Because they are just rough estimations between the hand wrist and the object center, the varying direction of the differences can be considered as meta criteria. Similar to the action gist extraction in Chapter 3, we also employ Gaussian Markov Random Field to extract the corresponding criteria. For a better visualization, we take a segmental result as Fig. 4.10.

Because Polhemus sensors also provide us with orientation information, we can explore whether we can find some criteria related to the bottle rotation. Fig. 4.11(a) gives us an criterion that the pitching angle of the sensor located on the bottle is relatively stable (Eq. 4.2 also works here), so the bottle may rotate along its long axis (in Polhemus system it acts as z-axis). If the bottle is rotated along its long axis, we can refer to the variation in the x-axis and the y-axis

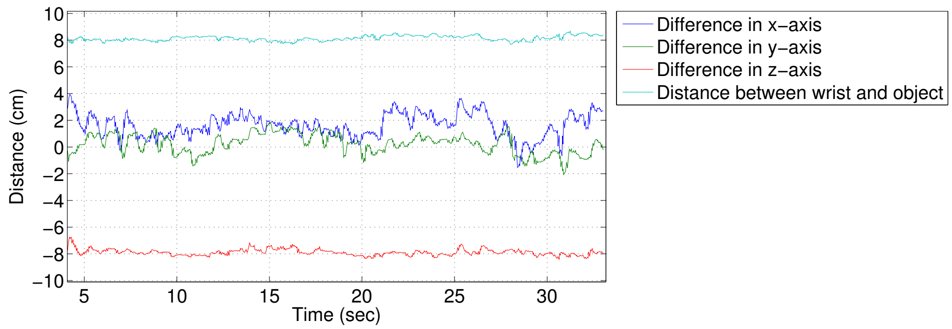


Figure 4.9: The distance between the wrist and the object sensed by Polhemus. Since the wrist and the object are bound with a Polhemus sensors, we know their three-dimensional information and make subtractions in three axes. We can calculate the Euclidean distance between the wrist and the object. The differences in x and y axes look not stable, but we find the z axis and the wrist-object distance are relatively kept with a stable value. Therefore, from the analysis we can generate at least a criterion: in this bottle rotating scenario, the bottle keeps a certain distance with the wrist.

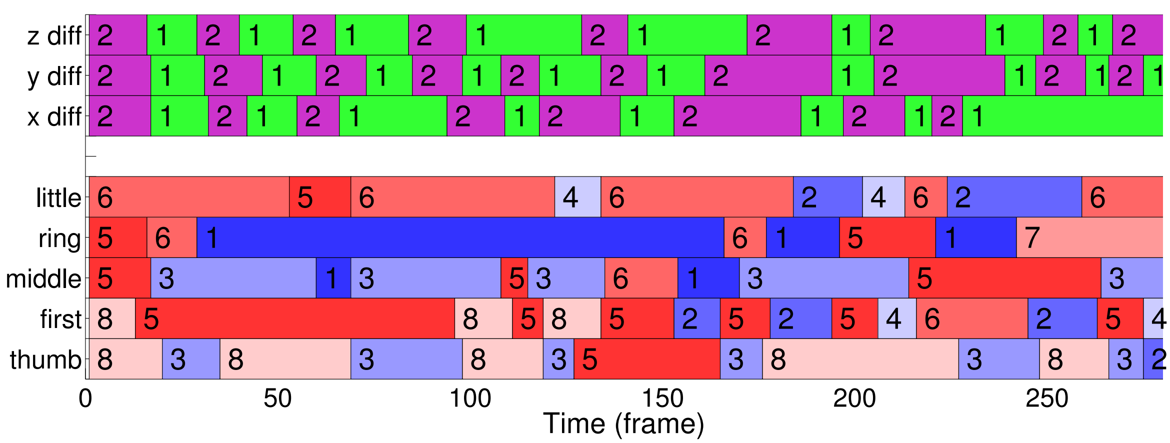
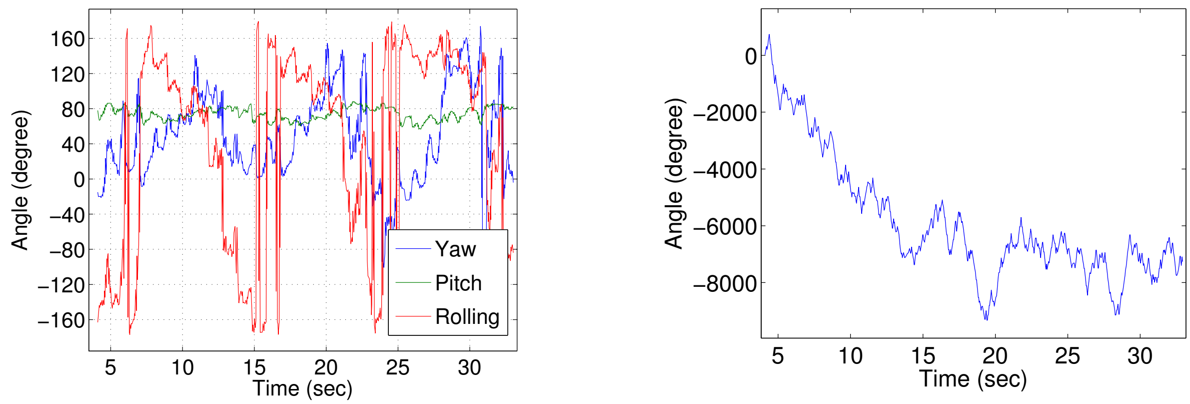


Figure 4.10: The state-action gist of bottle rotation (position criteria from Polhemus sensors). This is a segmental result of Fig. 4.9 because original sequence is too long to show on this page, where the segment starts from 4.79 sec and ends at 15.91 sec. We use pattern “1” indicating that the difference is decreasing meanwhile pattern “2” indicates that the difference is increasing. The data is already synchronized, so we can retrieve the corresponding criteria for the meta motions.



(a) The yaw-pitch-roll angles of the manipulated bottle from the Polhemus sensors. Even though the sensor is not located at the center of the bottle, it keeps a fixed relation with the bottle center. Based on the sensor location, sensor pointing direction and the related stable variation of the pitch angle in this figure, we infer the bottle is somehow rotated along its long axis. Actually, our purpose is to rotate the bottle clockwise, but because the bottle is not on the table, the real trajectory is not smooth.

(b) The accumulated rotation directions of the bottle. From the directional diversion of each frame we can sum up the angles and plot this figure. The variation tendency is trying to tell us it is decreasing, i.e., the bottle is rotated clockwise. However, because the bottle is not rotated on the table, the actual curve is away from the ideal trajectory.

Figure 4.11: Polhemus rotation analysis.

from the Polhemus data. Suppose that the Polhemus sensor positions is represented as (x_i, y_i, z_i), and then position variations will be $(\Delta x_i, \Delta y_i, \Delta z_i) = (x_{i+1} - x_i, y_{i+1} - y_i, z_{i+1} - z_i)$. Now we are concerning the rotation, i.e., angle variation, we can estimate the rotation direction θ_{rot} by this way:

$$\theta_{rot} = \mathrm{atan2}\left(\Delta x_i \Delta y_{i+1} - \Delta y_i \Delta x_{i+1}, \Delta x_i \Delta x_{i+1} + \Delta y_i \Delta y_{i+1}\right) \tag{4.3}$$

where $\mathrm{atan2}$ is an antitangent function, and it can output the angle in $(-\pi, \pi)$ according to the two-dimensional input. As a result, we accumulate the rotated angles as Fig. 4.11(b). In the figure we find the bottle is somehow rotated clockwise as our plan. We can also apply patterns "1" and "2" as Fig.4.10 to extract the corresponding gist, but we skip showing the result because it is a duplicated example.

Meta criteria from cameras

From Tab. 4.1 we find that cameras work for any criteria. However, meanwhile we should have good enough cameras and own good enough algorithms. For example, in the previous subsection we introduced the criteria brought from Polhemus sensors, and the criteria are possible to extracted from cameras as well; but the point is: except RGB+D camera, conventional planar imaging cameras wait for a process to convert two-dimensional information to three-dimensional information. The error occured in the two-dimensional information processing inevitably spreads

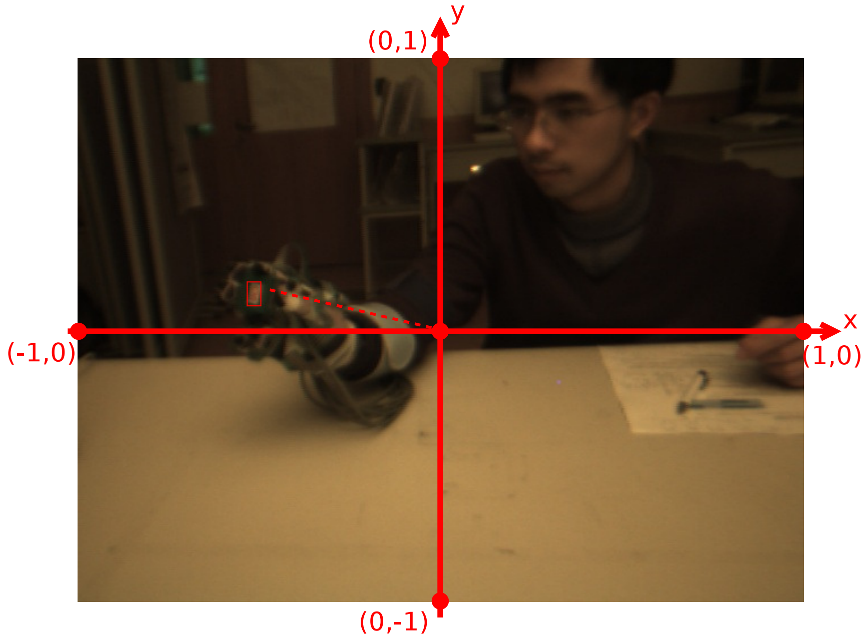


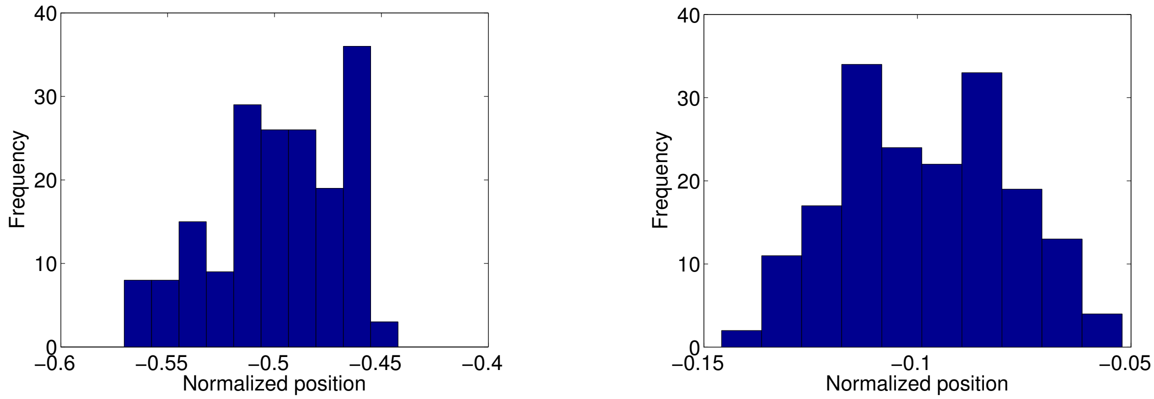
Figure 4.12: Rectangular prism tracking by CAMShift. We mark the tracking result by a red rectangular bounding box. Moreover, we are going to analyze the box center variation in the normalized coordinate.

to the three-dimensional criteria extraction. Therefore, we should seriously consider error diffusion.

Instead of driving further into above research topic, we suggest to extract planar criteria from cameras. Regarding to the difficulties of object detection and tracking in the proposed four scenarios, we take examples based on "rectangular prism rotation". In the scenario, because monocular camera on the lateral side only has slight information with respect to the prism, we are going to analyze the data from the stereo camera. The resolution of the stereo camera is set to 640x480, and the framerate is set to 5 fps. Because we just need planar information, we can refer to the left images. With manually object region assigning, we apply CAMShift algorithm (available in both Matlab and openCV) to track the prism movement. However, we should note that there are many tracking algorithms better than CAMShift, as long as we master them we can update the processing algorithm to get more accurate results. For this example, CAMShift is competent enough.

We plot an interim result as Fig. 4.12. Considering that different cameras have different resolutions, we can normalize the two-dimensional position to have a uniform impression. In this case, when we complete the demonstration, we memorize where the concerning objects were in a proportional form. When we reproduce the scenario in the real practice, we may have another camera but the proportional positions offer us reusable criteria.

We can first look at the position variation through histogram to determine whether we should take further actions. Fig. 4.13 shows us a fact that the rectangular prism does not move so much in the camera view. We can generate a brief criterion: the object moves at most 5% in the images. When we reproduce the scenario, we keep the object in a similar position of the camera as in the



(a) Position histogram in x-axis. (b) Position histogram in y-axis.

Figure 4.13: Position histograms in normalized axes extracted from camera data. After the rectangular prism tracking we have a corresponding trajectory of the bounding box center. We normalize the position information and count the frequencies of the existed positions. As a result we find the variation ranges are under 5% (e.g., in the left histogram, $((-0.45) - (-0.55))\,/2$). In this point we can propose a criterion that the object keeps in a certain position in the camera view.

demonstration. If the robot hand moves the object far from 5% in the camera view, we can stop the robot because it fails to meet the criterion.

Another case is that the tracking object is active in the camera view. For this case, we can analyze the proportional position variation as what we mentioned in the Polhemus section and generate the result as same as Fig. 4.10.

Meta criteria from Tekscan sensors

The TekScan array sensors can achieve 500 Hz sample rate of tactile sensing. However, because our data-glove is usually working with 25 Hz sample rate, we have to filter and down sample the TekScan data. Besides, an important fact is that our robot hand only has tactile sensors on the fingertips. Even though the Tekscan nearly covers the entire side of a hand (Hendrich et al., 2010; Handle-project, 2009a) and we have experience with it (Cheng et al., 2012), this time we just focus on the fingertip sensing.

Since we have used bottle rotation and rectangular prism rotation in previous sections, this time we try to explore the information from star prism rotation. This manipulation involves many finger movements. The thumb, first, middle, and ring finger take part in the manipulation. Along with the finger contacting and releasing, the star is spun in the hand.

The employed preprocessing methods are traditional median filtering and neighboring interpolation. After transforming the contact arrays into "contact" or "not contact" by assigning a threshold, we generate the contact criteria as from Fig. 4.14 to Fig. 4.15. In this section, we keep our position for simplicity and consider binary contact states as meta criteria. Consequently, we generate the paired meta motions and meta criteria. An example can refer to Fig. 4.16. Furthermore, we generate complete contact state-action gist for all four scenarios in Appendix A.

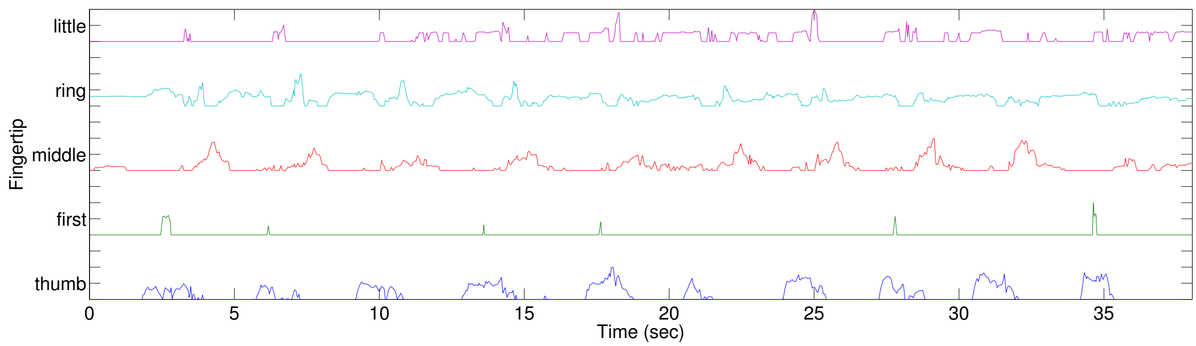


Figure 4.14: Fingertip contact from the TekScan sensors. Each finger link is attached with a TekScan arrays, and each sensor cell offer us a byte to indicate the contact intensity. We sum up the cell values over the arrays on the fingertips and plot this figure. Because in the demonstration the star prism is rotated over 10 indents, we should see 10 peaks in each value channels. However, only thumb values fit our expectation. Based on this criterion, we mainly concern the results related to the thumb finger.

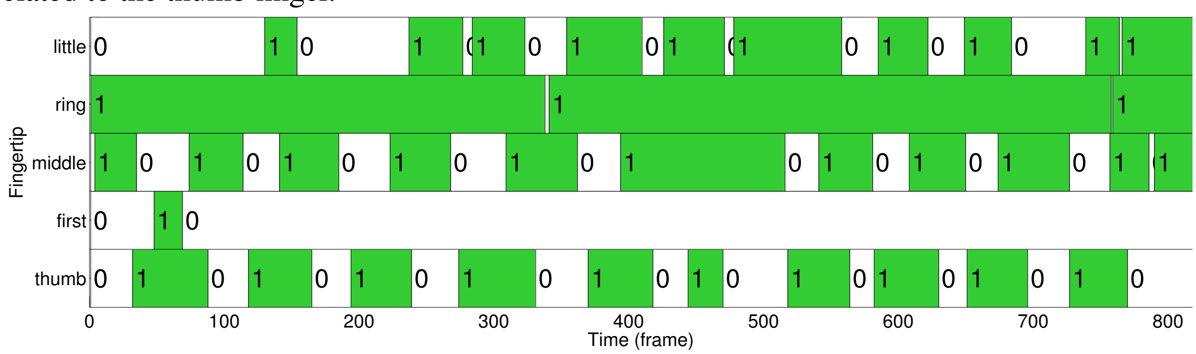


Figure 4.15: Fingertip contact criteria from the TekScan sensors. By several typical filtering algorithm we generate the binary contact criteria corresponding to Fig. 4.14. In this chart, 1 presents there is an contact on the fingertip along the period, meanwhile 0 means no contact. We note that in this figure we transfer the time from second into frame. The reason is: as criteria will be stored as skill knowledge, it is independent to the timing in the demonstrations. Another phenomenon we should point out is: with filtering methods proceed, we take the risk of losing necessary contact information. An obvious example is the contact criteria on the first finger, where only one contact survives. However, if we refuse filtering, we have much more contact fragments to deal with and we cannot generalize rules from the criteria.

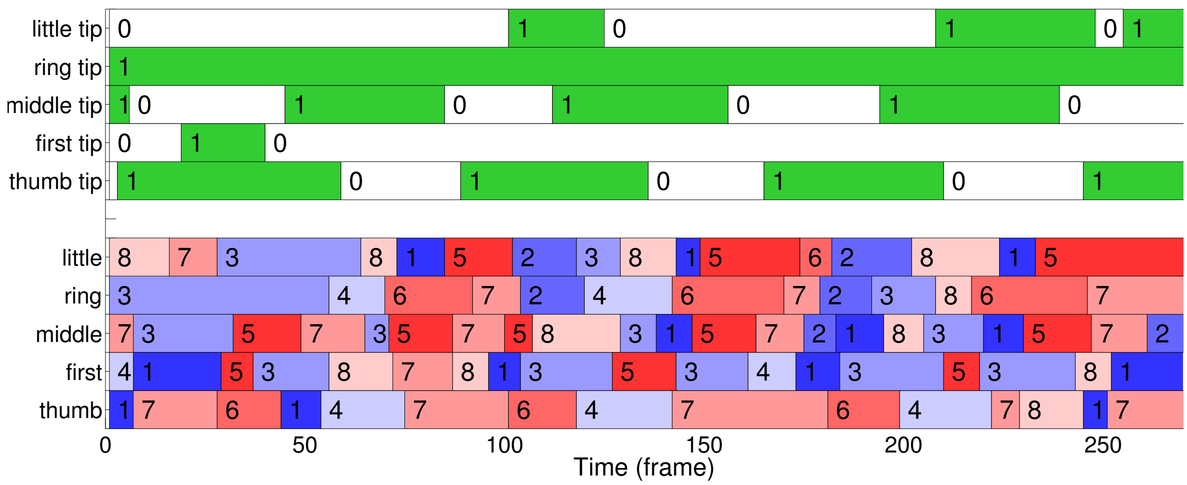


Figure 4.16: State-action gist of star prism rotation (contact criteria from the TekScan sensors). This is a segmental result corresponding to Fig. 4.15, where starts from 30$^{\text{th}}$ frame (c.a. 1.29 sec) and ends at 300$^{\text{th}}$ frame (c.a. 13.58 sec). Since the contact information of the thumb has 10 peaks in Fig. 4.14 as same as the star rotation loops, it brings an interesting fact: when the contact state is “1”, the thumb motions are similar to the form “7-6-4”.

4.4 Summary

In-hand manipulation state gist consists of any criteria companying with finger motions. During robot hand working, the criteria act as checkpoints to assist robot continue its actions. Generally, state-action gist is similar to a brief script to command the robot complete in-hand manipulation. Our ideal state gist is formed by natural language, because in this way we can express our perception freely. However, robots have their own language, whether they understand our idea, we need the techniques of Natural Language Processing. With the technologies developing, we believe in the future our dream will come true, and currently we have to describe the criteria in a way that our robots accept. We prefer simple criteria, because we can easily implant it from demonstration to real test with higher robustness; another reason is that it saves time for robot response, which is important for real-time operation.

State gist extraction relies on sensors, we can apply proper sensors for specific tasks. In this chapter we investigate and summarize the relation between the sensors and criteria. Even though there are still many technical aspects not referred in this chapter, this chapter proposes several kinds of typical analyses for the sensor selection, e.g. spatial error and temporal error. Besides, this chapter discuss the solution of sensor synchronization for meta criteria.

Afterwards, this chapter discusses how to use positioning sensors, cameras and tactile sensors to extract some simple criteria, e.g., position transition, stable criterion, and contact states. We see many sensors are installed on the hand, but they are just examples that we can integrate everything together. Actually we suggest not employ so many sensors on the hand. According to the feeling of the participant, when the sensor number increases the hand gets more difficulty to complete the manipulation movement. For example, normally a hand is as thick as 1.5 cm, when we wear a data-glove the hand becomes 1.7 cm. Further sensors extend the thickness of the hand to 2.0 cm, at this moment it seems that we have a new hand with our body. In this case, we cannot perform the movement as usual. Therefore, sensing the in-hand manipulation away from the hand is a promising topic. Computer vision greatly help in this field but there are many jobs waiting for our future work.

The same as action gist, we could only extract one state gist from one trial of manipulation demonstration. Therefore, in order to find the common operating rule from one manipulation scenario, we should record various demonstration trials and analyze the corresponding state-action gists. The necessary techniques corresponding to this topic is coming in the next chapter.

Chapter 5

Further Techniques of State-Action Gist

Contents

We have had two chapters on action gist and state gist, but remain many technical details waiting for solutions. This chapter considers two typical applications related to state-action gist. One is how to find the common gist from multiple demonstrations. The other is how to segment a long but periodic hand movement into several short segments. Additionally, no matter action gist or state gist, they have the same style presentation in this thesis. Therefore, we believe that just showing the processing methodologies related to action gist is enough to present the core ideas.

The motivation of finding the commonness from a set of demonstrations, originates from a fact: We record many trials corresponding to a specific in-hand manipulation task demonstrated by different persons, and the extracted results are not similar. For some cases, the action gists extracted from the same persons are also different even completely different. We agree that “there are countless way to manipulate an object”, and we pursue a popular (or say common, widely adapted) solution for the scenario. Therefore, in the coming section we will propose a suitable method.

Afterwards, this chapter points to the periodic in-hand movement segmentation. For “periodic in-hand movement”, we can refer to the experimental figures of this chapter (especially

Fig. 5.5) in advance. Generally speaking, periodic in-hand movement repeats several finger motions to operate an object, e.g., rotating a screwdriver or turning pages. The reason of segmentation is that we intend to figure out what finger motions are repeated in one loop. Besides, the methods we adopted are based on the similarity of each candidate segment, i.e. we need the techniques from commonness evaluation.

The structure of this chapter is the following: Section 5.1 directs to in-hand manipulation action gist popularity evaluation in order to find the most popular action gist among multiple trials of a specific task. Because this part is closely bound with action gist, so the related work is already introduced in Chapter 3. Afterwards, Section 5.2 presents the related work of movement segmentation. Then a segmentation algorithm is given in Section 5.3. After that we discuss the possibility of fusing the segmentation result with tactile information in Section 5.4. Section 5.5 illustrates the practical results of the proposed techniques. As the final part, Section 5.6 gives a summary.

5.1 Action gist popularity evaluation from a demonstration set

To manipulate an object, there are countless methods to handle it. This is the reason why action gist is so-called "essential motions" but we find many solutions corresponding to a specific application. Even though this phenomenon happens, it does not mean we cannot find interesting information through a set of extracted action gists. Now our concerned point is to rank the popularity of the obtained action gists from multiple demonstrations regarding to the meta motion types and their appearing order.

Assuming a meta motion sequence is $\mathbf{m} = \{m_1, m_2, m_3, \cdots\}$. Each element m_i involves pieces of information, including finger type, meta motion type, normalized beginning time, normalized end time, and so on. We use $\tau_{begin}(\cdot)$ to extract the begin time of the element, and $\tau_{end}(\cdot)$ for the end time, then for sequence $\mathbf{m}$ the following conditions must be fulfilled:

$$\begin{matrix} \tau_{begin}(m_i) < \tau_{end}(m_i) \\ \tau_{begin}(m_i) \leqslant \tau_{begin}(m_j) \end{matrix} \quad i < j \tag{5.1}$$

The ranking of order sequence depends on the frequency of different action gists with respect to the specific scenario. Thus, our first idea is to employ a simple statistic method, which only counts the frequency of every kind of action gist. However, this simple solution does not make full use of an action gist. As we know, an action gist consists of meta motions. If we find a way to involve every meta motion, that will be a more composite evaluation on the entire demonstration set.

Every action gist from the demonstration set is feasible. No matter how the durations of meta motions are changed, once the motion order is fixed, this motion set is able to manipulate the object to the destined state. For example, no matter we spend 2 seconds or 2 minutes to nip a ball, we succeed as long as we close the fingers to touch it.

Therefore, a meta motion occurrence histogram is applied to describe the statistic feature of the motion order from all demonstrated samples. Firstly, the begin time and the end time of each

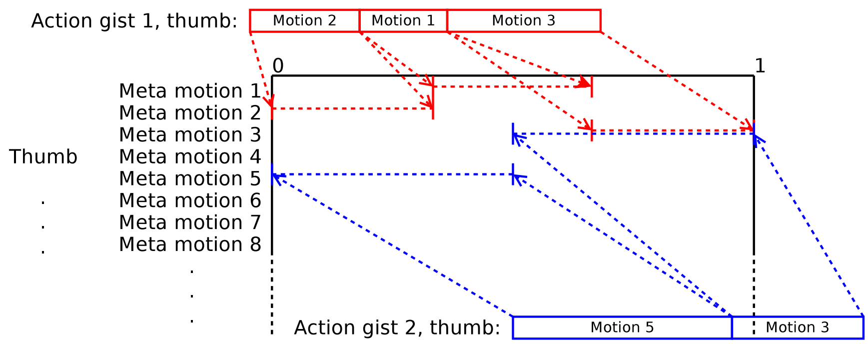


Figure 5.1: Calculating meta motion occurrence histogram. We take the input of the thumb finger for example. As the figure we have a fixed-sized statistical table. Suppose that the first meta motion sequence consists of three meta motions, we scale them into equal sizes (we can see the three red dashes are equal) and sum them up to the corresponding positions of the table. Later the second meta motion sequence comes, and it includes two meta motions. Similarly, we scale it to the table length and update the corresponding cells. After the inputs from all demonstrations, we finally get the meta motion occurrence histogram.

meta motion are normalized in the range between 0 and 1. Secondly, we prepare a fixed-sized statistical table to count the existing meta motions according to their positions. Specifically, we count them with impact factors obeying Gaussian distribution $G(\cdot)$. After counting, the fixed-sized statistical table becomes our meta motion occurrence histogram. The process can be described as Fig. 5.1, and the element of the histogram can be briefly written as a formula as follows:

$$H_{a,r,l} = \sum_{\eta(m_i^s,a,r,l)=1} G(\tau_{pos}(m_i^s), a, \sigma_s) \tag{5.2}$$

where m_i^s is the element from the action gist $\mathbf{m}^s$ in the demonstration set $\mathbf{M}$, $\eta(\cdot) = 1$ if and only if m_i^s belongs to finger r, labeled as meta motion l, and near position a. $\tau_{pos}(m_i^s) \in [0, 1]$ indicates the normalized order position when the meta motion begins. Besides, σ_s is a parameter that controls the impact factor reduction, and it is set to the reciprocal to the length of sequence $\mathbf{m}^s$. Considering the discrete numeric processing, the histogram has a resolution. Hence, the normalized a will finally be scaled as an integer during calculation.

With the meta motion occurrence histogram, the frequent possible meta motion takes a higher value in the corresponding cells. As a result, even if every action gist is independent, the popularity of the action gist (meta motion sequence $\mathbf{m}^s$) can be evaluated as in the following equation:

$$Score(\mathbf{m}^s) = \max_{seg(\mathbf{m}^s)} \sum_{j \in seg(m_i^s)} H_{j,\tau_{finger}(m_i^s),\tau_{label}(m_i^s)} \tag{5.3}$$

where $seg(\mathbf{m}^s)$ can be every kind of segmentation of action gist $\mathbf{m}^s$, it reallocates the normalized begin and end time of the movement as each segment is marked as m_i^s). τ_{finger} indicates the finger type, and τ_{label} indicates the meta motion type. In order to better understand this idea, we

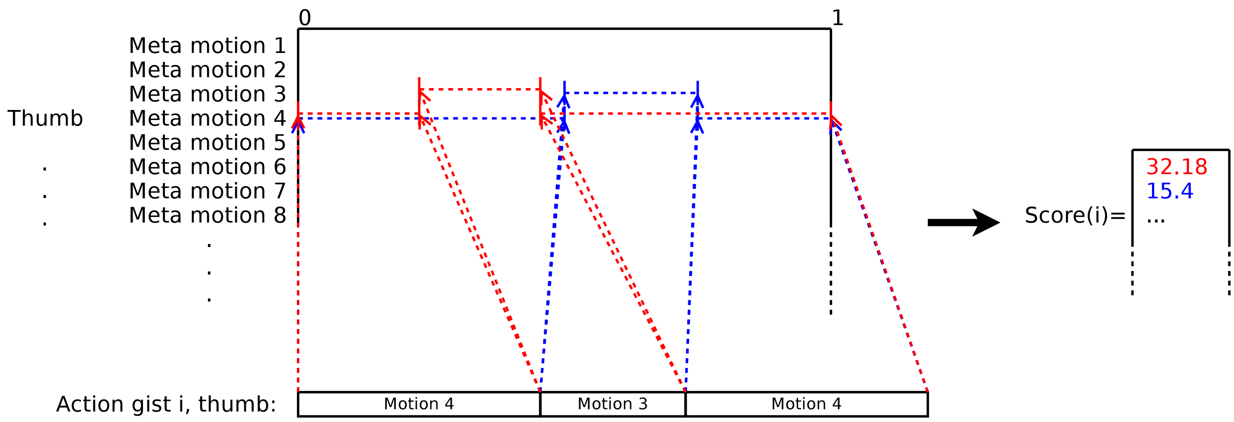


Figure 5.2: Evaluating an action gist by meta motion occurrence histogram. Since we have the statistical information in occurrence histogram, for an action gist we transform its meta motion sizes and sum the values up. For each transformation we have a score, the highest score is the popularity of this action gist.

can refer to Fig. 5.2. Specifically, Eq. 5.3 can be solved by means of Dynamic Programming. Therefore, it is not necessary to enumerate all kinds of transformed segments.

This kind of behavior evaluation method can describe the local similarities of the meta motions, and imply the action gist frequency. As a result, the action gist with the higher score is considered as a more common solution to the specific manipulation task.

5.2 Related work of in-hand movement segmentation

Since the process of in-hand manipulation contains a series of hand movements, it is common to segment the entire process into several smaller parts. In this way, the manipulation process becomes easily understandable so that we can concentrate on abstracting the interesting information in the segments. One kind of segmentation method depends on the hand gesture, which is based on the fact that the whole manipulation process can be understood as the translation of several significant grasping gestures. Several works of research such as (Cobos et al., 2010) have succeeded in reducing the scale of the realistic human hand gestures using principal components analysis (PCA) and discriminant functions. In this case, it is possible to use finite key hand poses to represent the manipulation process and guide the robotic hand in task execution.

It is possible to segment the value sequence according to the joint angle local minima or maxima, and then concentrate on the local extremes to study the periodicity (Valtazanos et al., 2010). However, as in-hand manipulation consists of the synergies of many joints, it is difficult to segment the movement only at this level.

More information can definitely give us more help, such as considering the hand and the object posture together as (Vinayavekhin et al., 2011) to instruct the regrasping movement. This work does not concern the whole hand postures but only the area where the object and the hand interact. Besides taking the object into consideration, another solution is to add sensors and understanding the manipulation in multiple channels (Hendrich et al., 2010; Faria et al., 2011a, 2012). Force sensing is also an important criterion of the manipulation state transition (Kondo et al., 2011). Without a sense of force feedback, humans are unable improve their manipulation skills. (Matsuo et al., 2009) segment the manipulation process as the contact region and the measured force on a specifically designed pencil.

When we consider the in-hand manipulation segmentation as a motion segmentation, we can learn more from similar topics. Basically, motion segmentation methods can be classified as online or offline. The online method can be like (Kulic et al., 2009), it can yield segmentation feedback to improve the robot real-time reaction. But here we are interested in analyzing the human demonstration, so we have enough time to process the acquired data. Thus we are more interested in the offline ideas.

For a time-series motion segmentation, we can consider it as a kind of clustering. Each segment is separated as the local relations of the elements. (Barbič et al., 2004) discussed Principle Component Analysis (PCA), Probabilistic PCA and the Gaussian mixture model in high level motion segmentation through the body joint angle variation. And then (Zhou et al., 2008) continued their work, clustered human motions based on k-means (Jain et al., 1999), and refined the classification by a global minimization algorithm. For the segmentation, key information is extracted as the criterion to divide the time series. Along with the clustering idea, we even can apply a general clustering model for time series data as lately proposed by (Rakthanmanon et al., 2011).

Anyway, we need to define the segmentational feature as the criterion to maintain the segment quality, such as rotation-invariant features (Keogh et al., 2009). We believe that in our in-hand manipulation case, we can find more specific semantic features.

For the hand movement recognition, (Ju and Liu, 2010) imported Empirical Copula to accurately detect the scenario. Besides, (Ju and Liu, 2011) imported a Fuzzy Active curve axis Gaussian Mixture Model (FAcaGMM) to detect the scenario fast. Based on the data-glove value, these methods can analyze what kind of hand movement is performed even if the training set includes only a few samples. However, it is not clear whether both methods can automatically segment the long manipulation process including multiple specific operations.

(Lau et al., 2009) proposed a segmentation method independent from prior knowledge on motion characteristics, and it is very effective in one-dimensional cases. But because of the recursive estimation algorithm they applied, the accuracy runs low with increasing number of joints.

Different from the classical segmentation criteria, our method concentrates mainly on the hand movement itself. It is a kind of semantic analysis based on the similar motions in the periodic manipulation segments. We use the data-glove to generalize the in-hand manipulation action gist with respect to the application, and based on this kind of semantic information we complete the segmentation.

5.3 Periodic in-hand manipulation movement segmentation

In the real world the paradigms of periodic manipulation can be like rotating a key, turning pages, or other movements operating repeatedly. In the process of *Learning from Demonstration*, we can decompose the entire continuous movement into several parts, with each part being a loop.

In this way, we only need to show the robot the complete part dozens of times, and the robot will extract the necessary information for future practice. The problem of this method is that we have to cut the movement by subjective judgment, as the connective information between adjacent movements may be missing. Thus if the demonstrator has the chance to perform periodic movement without interruption, it is a more natural way to acquire knowledge about the skill.

We intend to make use of the techniques of in-hand manipulation action gist to automatically segment the entire manipulation process. Because the meta motion semantically reflects the finger movement, the reduplicative motion patterns can more or less imply the periodic information.

Firstly we give the definition of the segment in a trial of periodic in-hand manipulation movements.

Definition 1 $\mathbf{m}^s$ *is defined as a segment of the periodic in-hand manipulation sequence, and it fits following conditions: It begins from the meta motion* m_i^s *on the finger* $\tau_{finger}(m_i^s)$*, and ends before the begin position of* m_i^{s+1} *on the finger* $\tau_{finger}(m_i^{s+1})$*; it must be* $m_i^s = m_i^{s+1}$ *and* $\tau_{finger}(m_i^s) = \tau_{finger}(m_i^{s+1})$ *except the last segment.*

In this case, every segmentation must begin with the meta motion of equal type. That means we do not need enumerate all possible partition solutions over the long sequence, so we can faster determine the segment boundaries. The assignment as Def.1 may result in mistakes, but the majority should share the same sequence head regarding the statistical point of view. Therefore, as long as we have done enough periodic manipulation demonstration, it is possible to extract the key information in the periodic movement.

Secondly we give the criterion of a good segmentation.

Definition 2 $|\mathbf{m}^{s_1} \bigcap \mathbf{m}^{s_2}|$ *represents the quantity of common same-fingered meta motions at similar positions in both segments* $\mathbf{m}^{s_1}$ *and* $\mathbf{m}^{s_2}$*.*

For a good segmentation, we expect $|\mathbf{m}^{s_1} \bigcap \mathbf{m}^{s_2}|$ to be as high as possible. Here the meta motion occurrence histogram is able to generalize the segment result and provide us with the evaluation of the periodicity of the segmentation.

Definition 3 *To a 3 dimensional Meta Motion Occurrence Histogram* $\mathbf{H}$*, the corresponding Frobenius norm is defined as*

$$\|\mathbf{H}\|_F = \sqrt{\sum_a \sum_r \sum_l |H(a,r,l)|^2} \tag{5.4}$$

Theorem 1 $\max \sum_{s_1} \sum_{s_2} |\mathbf{m}^{s_1} \bigcap \mathbf{m}^{s_2}| \Leftrightarrow \max \|\mathbf{H}\|_F$*,*
i.e., pursuing $\max \sum_{s_1} \sum_{s_2} |\mathbf{m}^{s_1} \bigcap \mathbf{m}^{s_2}|$ *is equal to pursuing* $\max \|\mathbf{H}\|_F$*.*

Proof: According to Eq.5.2,

$$\begin{aligned} & \max \|\mathbf{H}\|_F \\ \Leftrightarrow \quad & \max \textstyle\sum_a \sum_r \sum_l \left(\sum_\eta G\left(\tau_{pos}(m_i^s), a, \sigma_s\right)\right)^2 \end{aligned}$$

Here we can see that, once m_i^s exists, it will contribute to several elements in $\mathbf{H}$. In the meantime, the constraint $\eta(m_i^s, a, r, l) = 1$ controls the meta motion number of the contribution. $m_i^s \in \mathbf{m}^s$, and $\mathbf{m} \subset \mathbf{M}$. According to Def.1, unless m_i^s impacts $H(a,r,l)$ alone, it has to increase the sum of $H(a,r,l)$ with other meta motions.

Obviously, when we neglect the specific a, r and l, we will have a general form, e.g. $\sum_\iota G_\iota$, where ι is the index. Afterwards,

$$
\begin{aligned}
& \left(\sum_\eta G\left(\tau_{pos}(m_i^s), a, \sigma_s\right)\right)^2 \\
\doteq\ & \left(\sum_\iota G_\iota\right)^2 \\
\doteq\ & \left(\sum_i G_i + \sum_j G_j\right)^2 \\
>\ & \left(\sum_i G_i\right)^2 + \left(\sum_j G_j\right)^2
\end{aligned}
$$

Supposed in this case $\mathbf{i}$ meta motions remain a contribution to the specific element of the Histogram, but **j** meta motions work alone. The part $\left(\sum_i G_i\right)^2$ is considered as the new sums of the original element. The other part $\left(\sum_j G_j\right)^2$ is the new sums of where the meta motions jump to. So it means the original form takes a higher sum.

For the cases where $\mathbf{i}$ meta motions remain but $\mathbf{j}$ motions jump to cooperate with another element, there is no rule to judge which form is better. Nevertheless, it is not vital that more than one meta motion fit the constraint $\eta(\cdot) = 1$. All of these cases will be taken into the competition. We will select the winner with the highest sum.

Then we can say the more meta motions join to work together, the higher the total is. On the other side of the theorem,

$$
\begin{aligned}
& \max \sum_{s_1} \sum_{s_2} |\mathbf{m}^{s_1} \bigcap \mathbf{m}^{s_2}| \\
\Leftrightarrow\ & \text{More } \mathbf{m}^s \text{ fits the constraint } \eta(\cdot)
\end{aligned}
$$

Therefore,

$$
\begin{aligned}
& \max \|\mathbf{H}\|_F \\
\Leftrightarrow\ & \max \sum_{s_1} \sum_{s_2} |\mathbf{m}^{s_1} \bigcap \mathbf{m}^{s_2}|
\end{aligned}
$$

So when the segmentation generates a corresponding meta motion sequence set $\{\mathbf{m}^s\}$, we can justify whether this is the best segmentation by examining the corresponding $\|\mathbf{H}\|_F$. Based on this theorem, we propose an algorithm to automatically segment the periodic movement in an in-hand manipulation meta motion sequence as Algorithm 1. The current algorithm is a linear enumerating method to segment the sequence, where each segment begins with the same-labeled meta motion. Later it will be improved as an iterative or head-independent algorithm after we have enough criteria to prove it.

In this way, we can naturally present a repeated in-hand demonstration to the robot. During the process of analysis, the motion sequence is segmented as the proposed algorithm, and evaluated by Eq.5.3. In practical processing, there is the risk of errors or mistaken movements being included in the coherent movement. But as long as the positive data is the majority, we can refer to the evaluation from Eq.5.3 and believe that the segmented movement with a high score is acceptable. The one having a higher score indicates its popularity in the periodic movement, so we can use it to reproduce the manipulation.

5.4 Periodic movement segmentation fusion with tactile sensor

Generally speaking, with more sensors the segmentation will become more accurate. The segmentation process is equal to design a goal function and then to optimize it. However, more sen-

Algorithm 1 Segment the periodic movement of an in-hand manipulation demonstration with the techniques in action gist

Require: The extracted meta motion sequence $\mathbf{M}$

Find the same meta motion l on each finger r, store their starting positions as $\mathbf{P}_{l,r} = \left\{P_{l,r}^1, P_{l,r}^2, P_{l,r}^3, \cdots\right\}$;
$Score_{max} \leftarrow 0$, segmentation solution $\mathbf{Z} \leftarrow \{\}$.
for all $\mathbf{P}_{l,r} \neq \{\}$ **do**
    $\mathbf{m}^s \leftarrow$ The meta motion sequence ranging at the positions of $\left[P_{l,r}^s, P_{l,r}^{s+1} - 1\right]$.
    Demonstration set $\mathbf{M}_{tmp} \leftarrow \{\mathbf{m}^s\}$.
    Calculate the Histogram $\mathbf{H}$ of $\mathbf{M}_{tmp}$.
    if $\|\mathbf{H}\|_F > Score_{max}$ **then**
        $Score_{max} \leftarrow \|\mathbf{H}\|_F$, $\mathbf{Z} \leftarrow \mathbf{P}_{l,r}$.
    end if
end for
    return Z;

sors make the decision complex, we need to weigh and consider balance among many choices. So far, the common form of the fusion obeys the rule like following equation (Hackett and Shah, 1990; Ishikawa and Sasaki, 2002; Mitsantisuk et al., 2012):

$$Partition_{fusion} = \frac{\sum_i w_i Partition_i}{\sum_i w_i} \tag{5.5}$$

where w_i is the weight of the sensory segment component $Partition_i$. Regarding this equation we can imagine that the number of components may increase the uncertainty of segmentation. For example, a sensory channel offers 5 components but meanwhile another sensory channel offers 8 components, and then it is uncertain that how to match the components between different channels. Therefore, we only address how to deal with gloved and tactile data without weighting in this paper.

Because tactile information is an important criterion in hand manipulation, we intend to integrate our segmentation method with tactile perception. Considering Fig. 5.8 from Section 5.5.3 we believe that tactile information also can be refined as periodic criteria. And it is possible to apply the same method to the tactile state sequence as the techniques of manipulation action gist. But according to our current experimental experience, the tactile segmentation is not as reliable as the meta motion segmentation. The reason is related to the different sensitivities of tactile cells, and the complexity of the tactile sensory structure. And we have to point out, that the meta motion is the direction of the finger, but the contact always changes as the finger touches / leaves the object. That means the segments begin and end based on different mechanism, we can not use interpolation between the boundaries of two kinds of segments.

Therefore, we consider the tactile information as a support to the current segmentation method. The workflow is as Algorithm 2.

We consider Algorithm 2 as a kind of compromise between both kinds of segmentation plans. The reason is that the result calculated by action gist techniques based method is always close to the manual segmentation. We had better to keep the scale and the distances of the segments not

Algorithm 2 Segment the periodic movement of an in-hand manipulation demonstration with multiple information

Require: Ranked segmentation solution $\{\mathbf{Z_i}\}$, so $\|\mathbf{H_i}\|_F > \|\mathbf{H_k}\|_F$, $i < k$
Require: Tactile segmentation solution $\{\mathbf{T_j}\}$
$S_m \leftarrow +\infty$;
for $i = 1$ **do**
  for all $\mathbf{T_j}$ **do**
    if $\mathbf{Z_i}$ and $\mathbf{T_j}$ have similar number of the segments **then**
      Sum up the position difference between the nearest segment pair, one in $\mathbf{Z_i}$, and the other in $\mathbf{T_j}$.
      if $S_m >$ the calculated sum **then**
        $p_m \leftarrow j$;
      end if
    end if
  end for
  if $S_m \neq +\infty$ **then**
    goto 16;
  end if
  $i \leftarrow i + 1$
end for
Update $\mathbf{Z_i}$ by $\mathbf{T_{p_m}}$ with the closest segment positions, store it in $\mathbf{Z_{new}}$; **return** $\mathbf{Z_{new}}$.

far from the original solution.

5.5 Practical applications

In this section, we demonstrate the practical results on popularity evaluation, afterwards the glove-value segmentation results with manual segmentation results, and then discuss the possibility to fuse the segmentation from data-glove and tactile sensor.

5.5.1 Popularity evaluation

In order to make an intuitive impression, we take a simple example. In daily life we use a ladle to spoon the soup up (see Fig. 5.4). A subject took part in the task and has nine trials with the thumb, first and middle finger. The complete nine action gists can be found in Appendix B. With statistical processing and popularity calculation, we finally have the ranking list in Tab. 5.3. We can see that the top evaluated action gist is the shortest but the most common one in all trials.

5.5.2 Segmentation for multiple scenarios

We have an integrated system to record the Cyberglove data with synchronized visual data (Handle-project, 2012). By this tool, we are able to compare the segmentation from our pro-

Figure 5.3: Action gist ranking of ladle reconfiguration

Action Gist – m^s	**Rank**
(First, Motion5), (Middle, Motion6), (Thumb, Motion5), (Middle, Motion7)	1
(First, Motion5), (Middle, Motion6), (Middle, Motion7), (Thumb, Motion5)	2
(First, Motion5), (Middle, Motion7), (Thumb, Motion5), (First, Motion8)	3

Figure 5.4: Action gist popularity evaluation of the ladle manipulation. Thumb, first and middle finger participate in this scenario. After nine trials, we have a ranking list.

Table 5.1: Performance of data-glove based segmentation

Scenario	**Repeat**	**Miss**	**Exceed**
screwdriver	4 times	0 times	0 times
	4 times	0 times	0 times
	5 times	0 times	0 times
cover opening	7 times	0 times	0 times
	6 times	0 times	0 times
star rotation	6 times	2 times	0 times
	6 times	0 times	0 times
page turning	5 times	0 times	2 times
	8 times	0 times	0 times
	8 times	0 times	0 times

posed algorithm with manual results. We have 4 scenarios shown in Fig. 5.5 to examine the performance of our method. The demonstrator performs the experiments and repeatedly moves the corresponding object. Each application is demonstrated many times. After that, through maximizing the Frobenius norm of each segmentation, we can have the result like Fig. 5.6.

We have recorded synchronized visual data at the frame rate of 30 fps, and the frame rate of our Cyberglove is set as 15 fps. We spend 10 seconds for each demonstration of performing the periodic movements, and 20 seconds for the page turnings. By comparing with the timestamps of the visual sensor and the Cyberglove, we evaluate the performance of the proposed method as Tab. 5.1 and Fig. 5.7.

In Tab. 5.1, "Repeat" indicates how many times the demonstrator actually performs the motions. "Miss" indicates the number of the segments that the automatic segmentation fails to find. Moreover, "Exceed" counts the extra segments that the automatic segmentation finds but which are not real in the demonstration. In the table, we can see that the automatic segmentation of the star prism rotation and the page turning have mistakes. These two tasks are more complicated than the other two, and without training the demonstrator uses different movements to achieve the manipulation. But regarding that the trial time increases, the demonstrator becomes experienced and then this case will not easily happen.

Fig. 5.7 indicates the errors of the segmentation positions are mostly under 0.2s. Actually

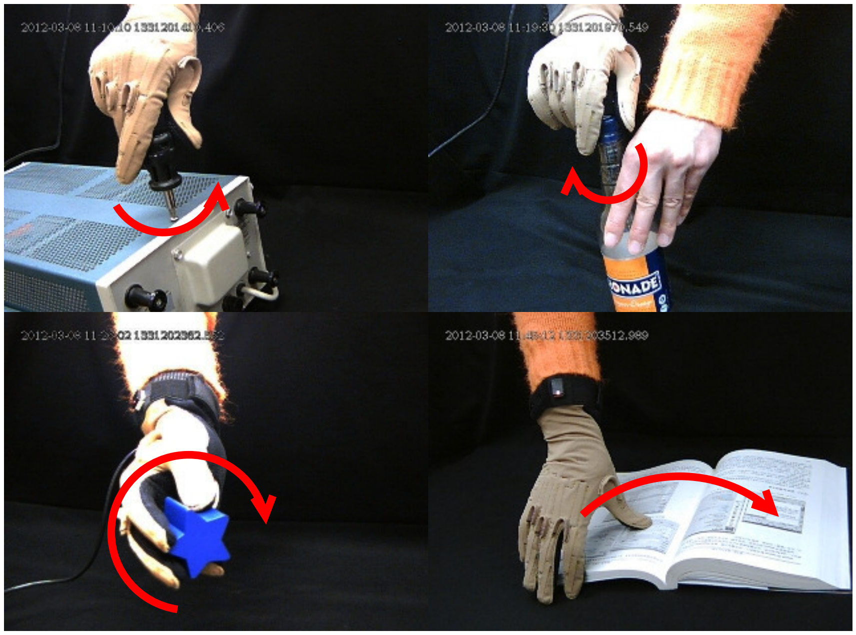


Figure 5.5: Four scenarios of periodic movements. The first one is to use a screwdriver to fix the screw. The second one is to rotate the cover to open the bottle. The third one is to play a star-like toy. The fourth one is to turn the pages of a book. The red arrows indicate the operating directions of the corresponding objects.

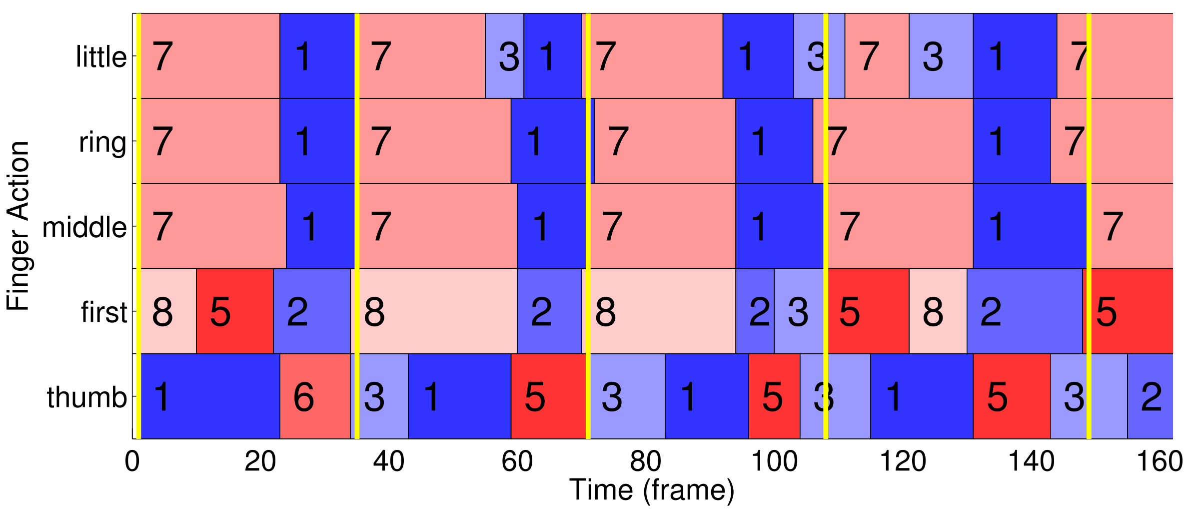


Figure 5.6: A segmentation example with the corresponding meta motion sequence from the screwdriver scenario. The figure shows the meta motions of all five fingers, each type of meta motion is represented by color rectangles with the corresponding type number. Specifically, the closing motions employ warm colors but the opening motions apply cool colors at different saturation levels. The x-axis is a time axis indicating the Cyberglove frame number. The yellow lines segment the entire sequence into several parts. This example is a segmentation by Meta Motion 7 in the middle finger.

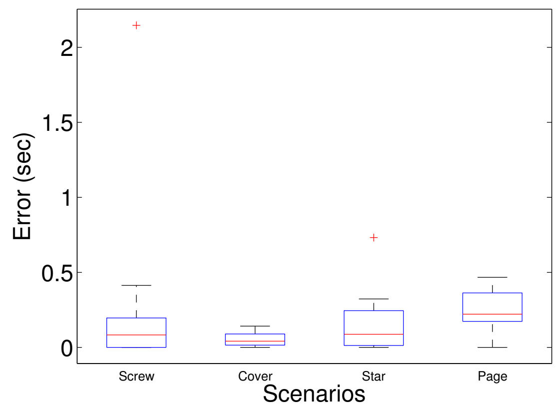


Figure 5.7: The errors of the automatic segmentation. Compared with the manual segmentation, we calculate the related errors measured by second. The blue boxes indicate the main variances of the errors. The black boundaries indicate the minimal and maximal errors of the demonstration, and the red lines in the boxes represent the averages. The red crosses indicate the outliers.

to the normal speed manipulations, with human eyes it is difficult to distinguish the movement difference in this duration level. So we assume the segmentation results are acceptable.

Among the demonstrations mentioned in Tab. 5.1, we have 4 failed segmentations. We investigate the raw data and find two reasons for the failures. One is because the pause between two periodic segmentation is too long, meanwhile the fingers look staying idle but actually slightly move. In this case the meta motion parser finds some unexpected motions which disturb the segmentation. The other reason is that the demonstrator applies multiple methods to carry out the manipulation, then the algorithm can not detect the segmentation correctly.

Therefore, it is helpful that more sensors participate in the manipulation analysis.

5.5.3 Segmentation involving tactile sensors

In this part we aim at analyzing the manipulation skill from multiple sensors. Even though the proposed algorithm in this paper is based on the information processing of the data-glove, we can have the experiment carried out using several devices including a stereo camera, a magnetic tracker, Cyberglove, the Tekscan Grip system (for details of the set up please refer to (Handle-project, 2009a,b), or the applications (Hendrich et al., 2010; Faria et al., 2012, 2011a; Martins et al., 2010)). Too many devices installed on the hand restrain the natural movement of the demonstrator, but it provides a chance to compare the segmentation result.

In Chapter 4 we already see the experimental setup enclosing the hand as Fig. 4.7. The image sequence of the manipulating process, the hand joint angle and the tactile information are available and synchronized in this set up. Besides, the movement of rotating a star prism is a typical periodic in-hand manipulation. We are going to study the block rotation movement by manual, tactile and finger-action-semantic segmentation.

The tactile sensor Tekscan consists of arrays of haptic cells attached to each the finger joint.

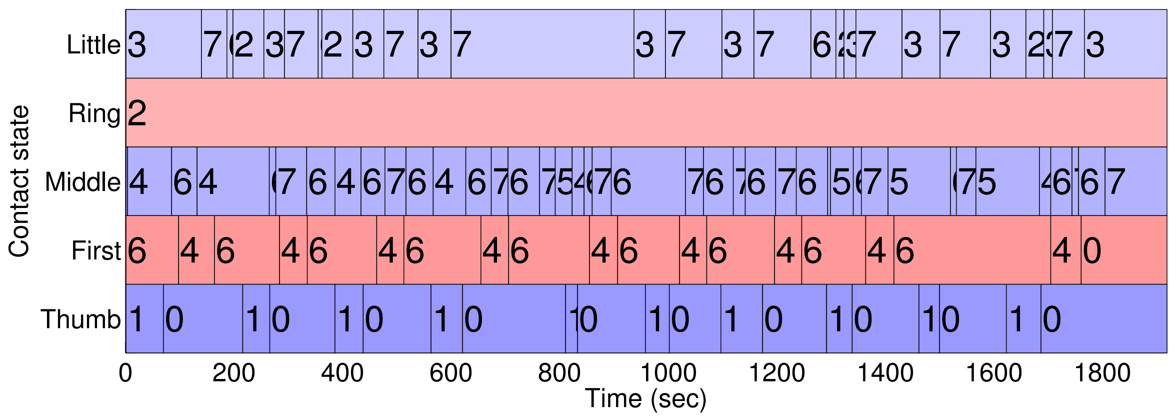


Figure 5.8: A segmentation example based on the tactile information. The scenario is star-like toy rotation. It is assumed that the contact region of each joint has a threshold value indicating whether it is touched. We use the 0 to present the non-touched state, and 1 for the touched state. For each finger we sum up the corresponding joint states, and represent it in decimal digits instead of binary form. For example, the thumb in digit 1 indicates that the distal joint is touched, but the proximal joint is non-touched; The first finger in digit 6 indicates that only the corresponding distal joint is non-touched, but medial and proximal joints are touched. We can count that the thumb spends 10 periods in state 1. Furthermore, in this demonstration, the demonstrator does rotate the block 10 times. However, the demonstrator does move the ring finger with touching and non-touching 10 times, but we can not see that the ring finger has any change in the figure. Therefore we currently only use tactile information as assistance and consider the contact state transition as a future research.

Table 5.2: Star prism rotation performance of fusion based segmentation

Repeat	Glove-based Miss / Exceed	Fusion-based Miss / Exceed
10	0 / 1	0 / 0
10	0 / 1	0 / 0

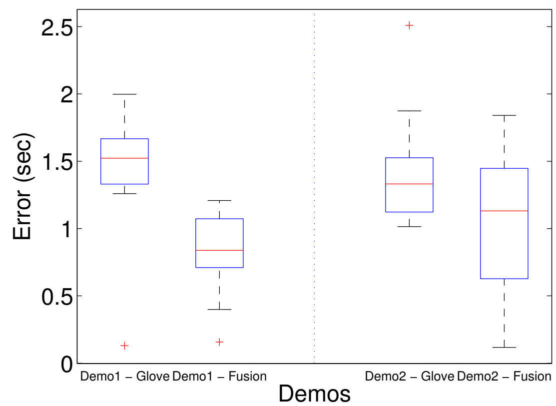


Figure 5.9: The errors of glove-based method and fusion-based method compared with manual segmentation. Because of the experimental set up, and for the demonstrator can not naturally perform the manipulation with wearing too many devices, the errors are higher than Fig. 5.7. But anyway we can see the fusion-based segmentation is better than the single sensor based segmentation.

Each cell contains a value presented by an unsigned byte indicating the intensity of the contact pressure. Basically we can segment the sequence by the different contact area combination. The state definition is similar to (Martins et al., 2010), but we do not need the palm contact information. One reason is that the palm does not participate in the rotation, another reason is that we find too much noise in this application when the parts in and around the palm rub reciprocally. Meanwhile the tactile information involves many factors, even as (Williams et al., 2010) indicates, grip force is affected by the hand posture. Therefore, here we only need to consider whether a force is applied to a specific area.

Therefore, we separate the in-hand manipulation state by the contact force variation with respect to each finger joint. Through summing up the intensity of corresponding cells, smoothing the totals, threshold filtering to separate the high and low value, and a series of post processing, an example of the segmentation is shown in Fig. 5.8. Then we can see as each finger holds a different transition form, there are many possibilities in the entire process.

Anyway, we can apply the segmentation method via the data-glove to understand the entire manipulation sequence. And then integrate the tactile segmentation as Algorithm 2. The results compared with the manual segmentation is shown as Tab. 5.2 and Fig. 5.9. Because the hand wears too many sensing devices, the demonstrator can not perform the manipulation naturally. Therefore the average errors are higher than glove-only method.

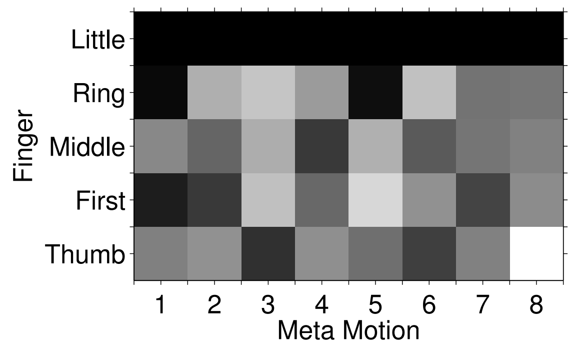


Figure 5.10: The possibilities of the start meta motions in 5.5.3. The block intensity indicates how likely it is for the meta motion according to the finger to become the head of the segment. After several demonstrations of star prism rotation, the meta motion 8 of the thumb wins the highest score. It implies that the demonstrator may have this behavior in the rotation scenario of a similar object.

5.5.4 Popularity of the first meta motion in the segment

To manipulate an object, there are countless finger-gaitings. We can get many meta motion sequences from action gist extraction, and every one will work in practice. However, for each particular manipulation task, we would like to find the common action gist. Because we think if one kind of movement is always performed by humans, it will be more stable than other movements applied in the specific scenario. For many trials from the star prism rotation in Section 5.5.3, we intend to find the popular head of the segment. Thus we sum the Frobenius norm of the Histograms up and evaluate the popularities. The result is shown by Fig. 5.10. The result indicates that when rotating a block with four fingers, the demonstrator always moves the thumb first. This criterion can be considered as a hint to the segmentation by tactile information.

In addition, we give the analysis to other experimental scenarios in Fig. 5.11. We hope the proposed techniques can more or less help us with the behavior understanding.

5.6 Summary

After proposing the popularity evaluation of in-hand manipulation gist, we propose a segmentation method based on maximizing the Frobenius norm of the Meta Motion Occurrence Histogram. The segmentation is a technique of in-hand manipulation action gist, to find the optimized segmentation of periodic hand movements. Different from gesture segmentation, the segment is sharp at the boundary of movement variation. We believe that the proposed method does support the process of *Learning from Demonstration*. Then the robot with a human-like hand allows the demonstrator to teach naturally instead of decomposing the entire operating sequence.

The current method is based on counting the meta motion, it belongs to a kind of semantic

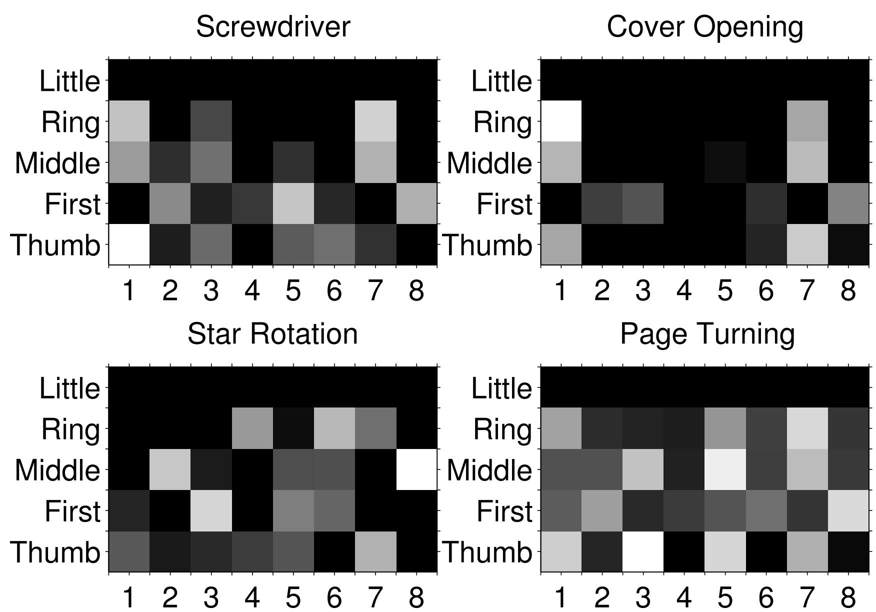


Figure 5.11: The possibilities of the start meta motions in Section 5.5.2. In all scenarios the demonstrator uses his right hand. We can find that the first motion in the "Screwdriver" scenario is using the thumb. Considering with the fact that the screwing demonstrations are anticlockwise, this result is reasonable. And we notice that for the clockwise movement scenario "Cover Opening", the demonstrator likes to move the ring finger first as expected. The "Star Rotation" result is different from Fig. 5.10 because the demonstrator often uses his middle finger to keep the block at the beginning. However, we think the "Page Turning" is the most interesting one in the cases because we find more bright blocks than other scenarios. We preliminarily think that is because the demonstrator wants to use the thumb to fix the page, or other long fingers to touch the margins.

analysis of the in-hand manipulation. The meta motion derives from the joint angles of the fingers; in the process of generation there may be some error. Thus, it is possible to have a more precise result based on the raw data. Anyway, to display with meta motion is more understandable than to display the joint angle values. In this case, humans can more easily interfere in the learning process to improve the cognition of the robot.

Chapter 6

Gist Guided Babbling Learning

Contents

Current robotic hands have become almost as flexible as human hands (Lovchik and Diftler, 1999; Gao et al., 2003a; Stone et al., 2007; Odhner et al., 2013; Mattar, 2013). This applies especially to the Shadow hand, where we can access both real and virtual hands to perform the hand movement supported by the robot operating system ROS. Therefore, in the field of in-hand manipulation it is possible to exactly mimic the hand behavior of humans.

As introduced in Chapter 1, we find many successful cases in object manipulation based on building a kinematic or dynamic model for a specific application. Besides that, another approach to master manipulation skills is to leave the modeling process to the robot itself. In other words, the robot can learn from what a human demonstrates.

We are not going to repeat the introduction of "learning from human demonstration" and "state-action modeling", which are already done in Chapter 2. In this chapter, we aim at applying the gist based state-action model to the in-hand manipulation self-learning. The state-action gist, especially action gist, involves the key hand movements consisting of several predefined basic motions. When the robotic hand performs manipulation tasks, the action gist works as a guideline to instruct the fingers to move step by step. With this kind of input, the search space of the finger gaiting decreases. Nevertheless, in practical implementations the robot will convert the abstract movement (action gist) into explicit joint values, which is where the learning comes into play. In

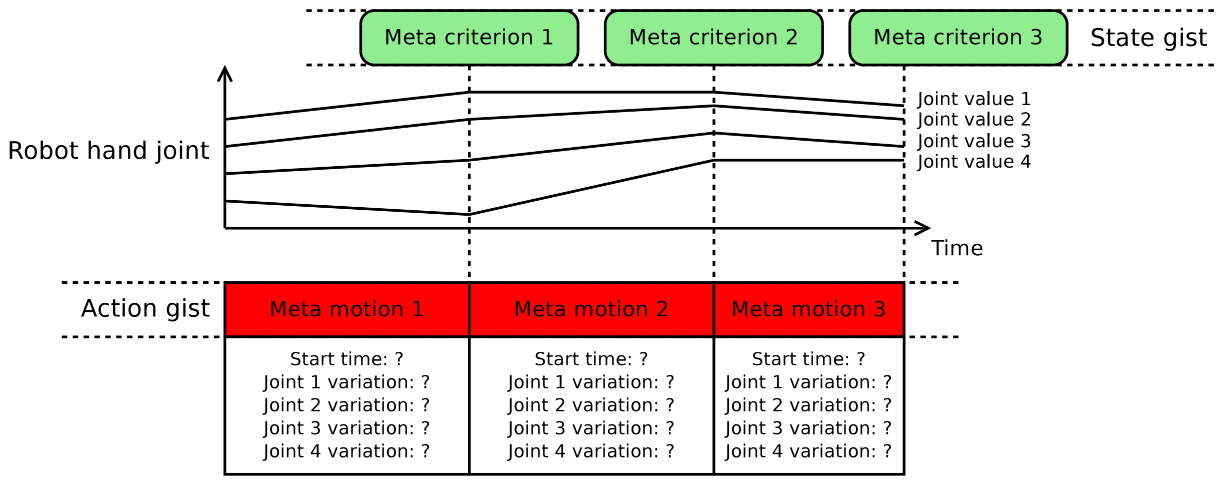


Figure 6.1: The control parameters to be learned. Suppose that an object is grasped in our hand and now we are moving it by the extracted meta motion sequence. Because action gist is learned from another hand (human demonstration), and the action gist only offers joint varying directions; in order to correctly perform the in-hand manipulation skill, the control parameters (when should the motion be triggered, the joint varying angles) are waiting for learning.

each step, the robot fingers move as prescheduled, until the set of control parameters correctly work on the fingers to move the grasped object into the destinated state. Therefore, the control parameter exploration is the key purpose in this chapter, and we find the intuitive description of the control parameters from Fig. 6.1.

The more flexible a robotic hand is, the higher the dimensionality. The Shadow dexterous hand, for example, has 24-DOF overall, of which 20-DOF are controllable due to the J1/J2 coupling of the fingers. For this research we are interested in the in-hand gaiting movements of the fingers, ignoring the wrist motion, resulting in a 18-DOF with the number of meta motions in the specific application. For example, a five-fingered grasping has 18 parameters to learn, and the star prism rotation in Chapter 3 (Fig. 3.7) has nearly 100 parameters to learn (the corresponding calculation method will be given later). We are trying to propose a universal framework for our state-action gist based in-hand manipulation learning, so we consider evolutionary algorithms to tolerate the dimensionality while approximating the best solution.

Here we employ the ideas of swarm optimization to optimize the searching. Compared to all other swarm intelligence algorithms, conceptually, Particle Swarm Optimization (PSO, (Kennedy and Eberhart, 1995; Shi and Eberhart, 1998a)) is the best candidate for finger joint value exploration. Firstly, PSO covers the concepts of particle velocity and position, so we can directly combine the PSO parameters with the joint values; and after the evaluation of one loop of manipulation control using the current parameters, we can adapt the parameters by changing the particle velocity. Secondly, the amount of the particles renders this algorithm likely to be resilient to local minima. This point ensures that we can find a good solution after a sufficient number of iterations. Currently our purpose is to apply state-action gist to directing the finger movement, so it does not matter whether the selected learning algorithm is the best one. However, we do our best to seek the start point of our learning process. For example, we intended to replace PSO by the Cuckoo search (Yang and Deb, 2009), which also has the concept of particle flying and is claimed to outperform PSO according to the recent paper (Civicioglu and Besdok, 2011). Besides, on the way of finding suitable algorithms, we discovered Line Search with Re-Start (LSRS, (Grosan and Abraham, 2009; Grosan et al., 2011)), which operates the searching by iteratively updating the parameter boundaries. According to our test, within a large but acceptable number of agents and iterations, none of these large data exploring methods can definitely converge to the global best solution of a specific manipulation task. However, they all approach the solutions better than the initial method. In the process of implementing and comparison, we notice the highlights of these algorithms: PSO adjusts parameters in a reasonable way; LSRS is good at decreasing the search space for each parameter; Cuckoo search, or other methods similar to the genetic algorithm, can eliminate the weak candidates. Based on these highlights and our practice, we finally integrate state-action gist with PSO and LSRS for our motor babbling learning. The reason of not using those methods which eliminate weak identities, is because after applying them in practice we have not found them to offer any significant improvement.

We have applied action gist to sort the motion order and limit the joint value variation, and state gist to evaluate the achievement of each trial. Meanwhile, the parameters needed to learn involve the joint value variation of each finger as well as the corresponding start time. We treat all of the joint value variations and start times together by translating the value into the proper form for the learning model (PSO and LSRS). In each babbling of the hand, we can find corresponding cost of the learning model; according to the cost new babbling parameters are generated until the final solution is found.

The purpose of this chapter is to propose an in-hand manipulation motor babbling learning framework integrated with the state-action gist and the reinforcement learning methods. We organize it in the following form: the next section introduces the state-of-the-art progress of motor babbling and learning. Afterwards, Section 6.2 illustrates the basic level of our work on how to move the hand and how to adaptively keep the fingers attached to (or away from) the object. Based on low level control, we propose our babbling learning algorithm in Section 6.3. And then Section 6.4 gives the experimental results. Finally Section 6.5 summarizes this chapter.

6.1 Related work

Learning from demonstration has many methods (Argall et al., 2009). Our methodology has two steps, one is let the robot learn the compact key knowledge; the other one, which is the topic of this chapter, is to execute manipulation movement iteratively to convert the short but key knowledge into its own skill. Some researches rely on the position information to learn the manipulator trajectories (Howard et al., 2009b; Ratliff et al., 2009), but currently our method is based on the finger joint angle space. The reason is that, the basic command of robot hand control is in the joint angle space, and the position space will be finally translated to joint angle space (e.g., from fingertip positions to a hand posture); according to Chapter 2 and Chapter 4, position-based methods distract us by sensory information processing; so we start from the joint angle space to see how far we can go. However, usually the exploration in the joint angle space is larger than the position space. Therefore, some parametric simplification methods are proposed in recent years (Vinjamuri et al., 2011; Prieur et al., 2012). We expect our approach adapting to as many in-hand manipulation applications as possible, but we are not certain about whether the simplification can qualify for complex finger-gaiting tasks. Thus, currently we are interested in how to complete our task with the help of motor babbling.

Motor babbling originates from the concept named "body babbling" proposed by Meltzoff and Moore (Meltzoff and Moore, 1997). Under the claim that the imitation is a matching-to-target process, a loop integrated with infant actions, adult actions and proprioceptive feedback was introduced to match Meltzoff and Moore's active inter-modal mapping (AIM) hypothesis. As a key part of the infant action system, body babbling coordinates movements to the organ end states so as to achieve the goal that the adult aims for. Because an infant needs knowledge from adults, prior input is allowed to strengthen the exploration of body babbling. Therefore, the previous framework was extended by (Shon et al., 2007; Demiris and Meltzoff, 2008) to include robots and considered infant motor acts as motor planning. Besides, it explicitly indicated that motor planning consists of three models: forward model (world dynamics), prior model (instructor's policy) and inverse model (action selection). These three components drive the robotic individual taking actions according to its own performance and the instructor constraint. Actually, so far there are two opinions on motor babbling application. One opinion is that motor babbling is a kind of robotic behavior only consisting of random movement (Ognibene et al., 2006; Lopes et al., 2010; Rolf et al., 2010; Bodiroza et al., 2011), the evaluation begins after all trials are done. The other opinion stands for behavior that improves with the iterations of babbling, and (Billing and Hellström, 2010; Billing et al., 2010; Montesano et al., 2007) position motor babbling as "Behavior Coordination". Comparing the above two opinions, we vote for the second one to embody our learning algorithms. Currently we are aiming at in-hand manipulation learning, so exploring finger movement is necessary. Besides, since this search process cannot promise to achieving a better solution everytime, it can be considered as iterated motor babbling learning.

Before the robot hand starts motor babbling, we should prepare a schema to tell the robot what, when and how it can do according to the scenario, as in the cases of (Kjellstroem et al., 2008; Kobayashi and Hosoe, 2009; Gupta et al., 2009b; Kobayashi and Hosoe, 2011). In the process of motor babbling, the robot should exploit its own memory to reorganize prior knowledge to instruct itself (Baxter and Browne, 2011). Even if the robot begins without prior knowledge, it is possible to use forward and inverse models to generate its knowledge database (Demiris and Dearden, 2005).

Furthermore, we can refer to some cases related to manipulation babbling learning: (Saegusa et al., 2008) applied motor babbling in arm posture control combined with visual feedback. (Oztop and Arbib, 2001) applied Motor Babbling in 12 DOF robotic hand grasping. It is based on Hebbian theory (Hebb, 1949); successful grasp parameters are strengthened while the ones that tend to fail to do so are weakened. Furthermore, the hand is controlled through forward kinematics and inverse kinematics. (Caligiore et al., 2008) simulated a babbling reaching and grasping scenario with obstacles, where the control parameters are iteratively updated by a biological constraint — Hebb learning rules. (Ciancio et al., 2011) applied motor babbling in CPG-driven hand grasping simulation. Through adjusting the parameters of different CPG models, hands of varying sizes can learn how to rotate objects.

In this chapter we are realizing five-fingered hand (18 DOF) in-hand manipulation for the Shadow robot hand. Therefore, the degree of freedom and the finger-gaiting complexity are

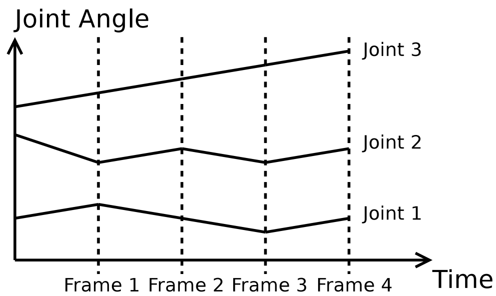


Figure 6.2: Robot hand control by sending joint angle frames. We can send joint angles to the robot hand to reconfigure its posture. Driven by the motors, the fingers gradually approach to their target positions. If we continuously send frames, the fingers move continuously. Furthermore, if we predetermine the joint variations and send the slightly moving joint angles quickly (e.g. 20 Hz), we can see that our robot hand is working smoothly.

much larger than the related work, so we are going to propose a new solution for this new challenge. Another characteristic from this work is that the learning is guided by state-action gist. The learning parameter set is specified by the corresponding action gist, and meanwhile the evaluation process corresponds to state gist. Also, we want to keep the learning time for the real robot as short as possible, because we should protect the real robot. Therefore, we run the learning process in the ROS Gazebo simulation until we have a approximately good result, and then apply the simulated result to the real robot hand to further refine and complete the in-hand manipulation task.

6.2 Robot hand control

The basic unit of an articulated robot hand is the finger joint. Therefore, in low level control, we usually reconfigure the hand posture by sending joint angle information (see Fig. 6.2). We can use the word “frame” to represent the moment of sending one or more joint values to move the finger. Considering the applications of in-hand manipulation, the joint variations are more complicated than pure grasping movements. This means the method of learning finger joint angles frame by frame is not realistic.

Because action gist provides us with the sorted meta motions and each meta motion limits the range of the joint variation, we can tell the robot to perform the manipulating movement in the order predetermined by the meta motions. Supposing that the initial hand and object posture are ready, given the start time, end time, and angle variation corresponding to the finger, it is possible to calculate the exact hand movement as command sending frame by frame. Here, the initial start time, end time and angle variation can be easily found from the data-glove value sequence along with the action gist extraction. Also, we notice that for each finger, the end time of a meta motion is the start time of the successive meta motion, so the end time of each meta motion is redundant. Because the joint values are not calibrated and the sizes of the demonstrator’s and robotic hands are different, we do not expect the initial data to work as soon as it is applied to the target application. Furthermore, we suppose that the order of the meta motion is reliable; as

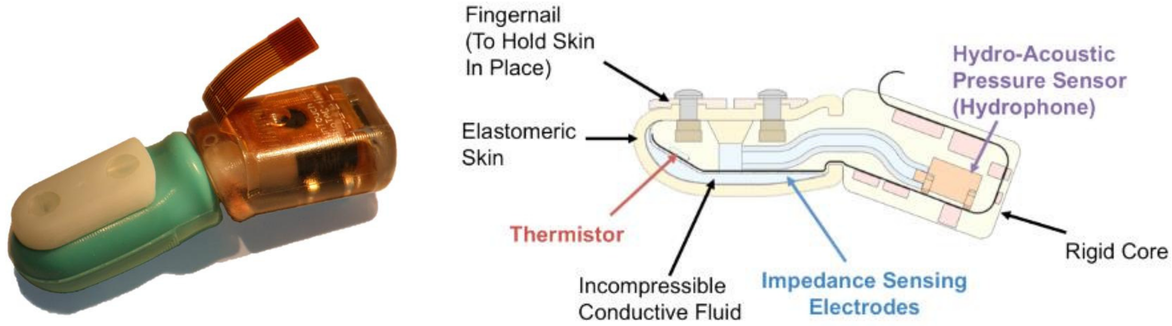


Figure 6.3: BioTac Finger and its internal structure (Fishel et al., 2012). All sensors are protected inside of the finger. We can get static and dynamic pressure, as well as the temperature information from this kind of finger.

a result, after iterations of parameter adjustment, we can find the correct values to complete the manipulation task.

Moreover, the platform is an important factor when we consider in-hand manipulation application. Currently we use BioTac hand. BioTac hand[1] is invented by the cooperation of Shadow Robot Company and Syntouch LLC. It is a five fingered motor hand with 20 actuated degrees of freedoms (DOF). The thumb finger has 5 DOFs, the first, middle and ring finger each has 3 DOFs, and the little finger has 4 DOFs. Specifically, the first two joints of each finger are flex/ext-joints. Each fingertip is a biometrical tactile sensor system. Providing with arrays of force and temperature information, it has a rigid core surrounded by an elastic skin filled with a fluid (Fig. 6.3). This robot hand applies EtherCAT communication protocol, so it is able to keep streaming the sensing, control and feedback data in a high speed.

We control the BioTac hand by sending frames of finger and wrist joint values. When the robot hand receive the joint value commands, the corresponding joints will try to move to the target position as a preconfigured speed. In order to have a smooth hand movement for manipulation, we can keep sending the scheduled sampled joint values with a high frequency.

Hereinafter we describe the details of the mapping between action gist and joint angle frame.

We endow the meta motions $\mathbf{m} = \{m_i\}$ with messages including start time and joint angle variation, and the initial joint angles of the robot hand at that moment. Therefore, the general control flow is presented as Alg. 3. In the procedure we can see several new variables and functions. Because the finger joint angle is sent to the robot hand by frame, first of all, we should normalize the time in order to know whether the working frame is controlled by the meta motion m_i. Therefore we take $T_{nor}(j) = j/N_{frame}$ to normalize the frame time, $\tau_{s\,time}(\cdot)$ and $\tau_{e\,time}(\cdot)$ to obtain the normalized time of meta motion m_i. Second, we should tell the robot hand to update the corresponding finger joint angles with respect to the joint angle variations $\mathbf{A}(\cdot)$. Considering that the joint variations rely on time and motions, there are many possibilities to design this function. Currently we only use the simplest form — a linear model to deal with the joint variation, i.e.,

[1] http://www.syntouchllc.com/Products/ShadowHandKit.php

$$\mathbf{A}\left(m_i, T_{nor}(j)\right) = \frac{T_{nor}(j) - \tau_{s\,time}(m_i)}{\tau_{e\,time}(m_i) - \tau_{s\,time}(m_i)} \times \tau_{jo\,var}(m_i) \tag{6.1}$$

where $\tau_{jo\,var}(m_i)$ involves all relevant joint angle variations of meta motion m_i.

Algorithm 3 General control flow according to provided motion control parameters

Require: The action gist $\mathbf{m}$

Require: The number of frames to communicate with the robotic hand N_{frame}

for $j := 1$ to N_{frame} **do**
  for each $m_i \in \{\tau_{s\,time}(m_i) < T_{nor}(j) < \tau_{e\,time}(m_i)\}$ **do**
    Update the joint angles $\tau_{finger}(m_i)$ as joint variation $\mathbf{A}\left(m_i, T_{nor}(j)\right)$
  end for
end for

In addition, according to the analysis of raw data-glove values, we notice that the linear model is not enough to exactly describe joint angle variation. This simplification is insufficient for some complicated applications, e.g. when a finger is required to move with acceleration. Thus, in the future we will apply higher order or other fitting models to abstract the joint angle variation. Just for the sake of decreasing the dimension of parameters, we aim at solving simpler cases first by assuming that each finger moves at a constant speed in a motion slot.

Because there are tactile sensors mounted on the Shadow hand fingertips, during the in-hand manipulation task we apply a simple technique to correct the improper hand pose in realtime. We can call the technique "adaptive grasping control", which controls the finger joints with respect to the contact states on the fingertips. Firstly, we define several levels for the force intensity, e.g., no contact, just touching, medium level touching, high level touching. Secondly, we adjust each finger as following rules:

- If the destinated contact state is touching but currently the fingertip does not touch the object, the corresponding joint angles increase. In this way we expect that the finger approaches the object.
- If the destinated contact state is touching but currently the intensity level is "high level touching", the corresponding joint angles decrease. In this way we protect the robot hand or the object.
- If the destinated contact state is no contact, but currently the fingertip touches the object, the corresponding joint angles decrease. In this way we expect that the finger leaves the object.

The module "adaptive grasping control" is standalone, we can enable it whenever we want. Besides, this module can fail in some cases. For example, for some touching we should extend our fingers, but this behavior conflicts with the rules we just specify. Anyway, adaptively correcting finger joints helps a lot as we are applying a linear fitting model to the joint angle variation $\mathbf{A}$. Therefore, for a first attempt of the tasks, we always enable this module for babbling learning.

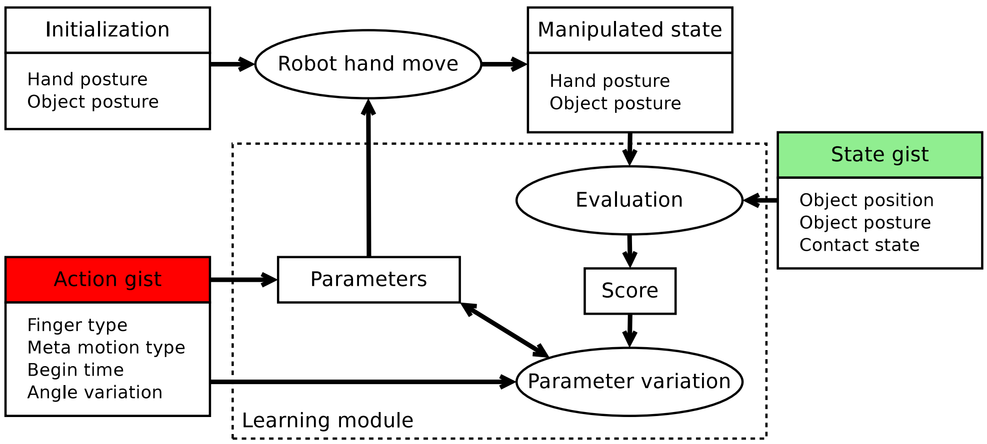


Figure 6.4: Main workflow of action gist based babbling learning. Generally, the meta motion sequence (action gist) consisting of which finger it concerns, when it moves, what the angle variations of corresponding joints are, and which meta motion it belongs to, is considered. The values from the entire meta motion form a particle (parameters to learn), specifically, the begin time and joint angle variation will be slightly changed because there are multiple particles for the algorithm. Then all particles are put into the shadow hand simulation and the object is manipulated. For the result we can estimate the manipulation achievement according to the state gist, and give feedback to the learning module so as to adjust the parameter values. Because action gist constrains the range of each parameter in the particle, the exploration effectivity is improved. Finally we will get an acceptable result after several iterations.

6.3 Action gist based motor babbling learning

Based on action gist, we already have a rough idea of when and how the fingers should move. We now combine the information with PSO / LSRS exploration to refine the control parameters in the simulation and then later in real robot execution. Fig. 6.4 illustrates the framework of our work. Using the Gazebo simulator with a physical engine, we have a testbed to babble the hand movement repeatedly without damaging the real robotic hand. Firstly we should initialize the hand and object, assuming that the object is already in hand. Then the hand is controlled by the command frames, each consisting of all joint variations to complete a trial. In order to avoid useless workload, there are comparisons between several frames to check whether the object is moved as scheduled. Once a state digresses significantly from our expectation, the hand posture and the object pose will be reset for another trial. In order to receive a good learning result, we pay attention to several key issues as detailed in the following subsections.

6.3.1 Joint angle control parameters

As mentioned in Sec.6.2, we apply a set of parameters in relatively compact format indicating joint angle variation, start time, and the corresponding finger. The robotic hand concentrates on learning these parameters from the feedback of executing them. Meanwhile we notice that we should design the parameters in a proper form for the robotic hand, because the prior knowledge

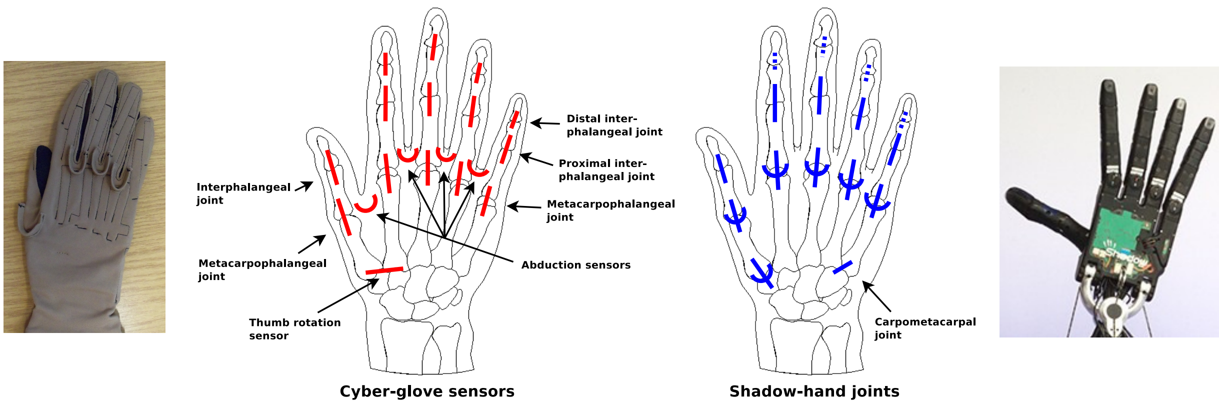


Figure 6.5: The sensor and joint layouts of the Cyberglove and the Shadow C5/C6 hand. As the red tags indicate, the Cyberglove sensors cover all visible joints of the human hand and also support thumb rotation measure. By comparison, we can easily find the difference in the joint structure of the Shadow hand, e.g. each finger has its own abductional degree of freedom, the thumb has 5 DOF and the little finger has a Carpometacarpal joint. Finally, the distal and proximal interphalangeal joints are coupled and can thus be jointly controlled in the Shadow hand.

is extracted from a data-glove. Additionally, we can compute the scale of the parameters.

Joint mapping from the data-glove to the robotic hand

After human demonstration, action gist is extracted from the data-glove data. In order to pass this information to the robot hand, we need to map the joint variation from the data-glove to the corresponding joints on the robot hand. The reasons for this are the following: instead of randomly initializing the control parameters, we configure them similarly to the raw values from the data-glove, which are not calibrated initially. Calibrated data-glove values are not strictly necessary in our case, because action gist is only based on the joint angle variation. However, if the map values are already close to the final solution, this of course provides a better initialization for the learning process. As shown in Fig. 6.5, the Cyberglove measures the abductions between finger pairs, while the abduction angles can be controlled for each finger individually on the Shadow hand. Several other joint angles of the Shadow hand have no direct correspondence with the Cyberglove sensors either. Therefore, according to our experience in generating the action gist, we propose mapping their relations in the following way.

1. For all joints in exactly the same positions on the hand, just keep them.
2. Abductional data-glove joint values are only used to assign the initial value for the robotic hand, thumb-index for the robotic thumb finger, index-middle for the robotic index finger, 0 for middle, ring-middle for ring, and little-ring for little finger.
3. The thumb-index abductional joint controls both Shadow hand thumb abduction joints.
4. The carpometacarpal joint of the little finger is always set to 0.

The organization of the control parameters

Since action gist uses start time, joint angle variation, and the corresponding finger to instruct the in-hand movement, we should arrange them in a proper way and estimate how many parameters are necessary for a specific application. Firstly, we assume the order of meta motions to be correct after action gist extraction, so that we have a fixed order of the parameters. Secondly, depending on the finger, we can determine the number of joints as shown in Fig. 6.6. Therefore, presuming that we have a meta motion sequence $\mathbf{m}$ consisting of L_m motions, and each motion is paired with a **start time**, we can simply get the total dimension as follows:

$$d_{\mathbf{m}} = \sum_i \left(1 + L_{joint}\left(\tau_{finger}\left(m_i\right)\right)\right) \tag{6.2}$$

here $\tau_{finger}\left(\cdot\right)$ indicates the finger of meta motion m_i, and we can get joint number $L_{joint}(\cdot)$ from Fig. 6.6. Obviously, $L_{joint}(\cdot) = \{5, 3, 3, 3, 4\}$ correspond to the thumb, first, middle, ring and little finger. Now we establish a $d_{\mathbf{m}}$-sized vector to match the latter learning algorithm.

Moreover, we can understand the data structure and the dimension calculation with respect to an example. Fig. 6.7 uses a set of photos to describe the process of rotating a screw-cap anticlockwise. Four fingers move the screw-cap, and their joint variations are extracted into meta motions as Fig. 6.8. In this case, we can calculate the parameter dimension as following process:

See Fig. 6.8, the thumb finger has 2 meta motions, so it should have $2\times6 = 12$ corresponding parameters. Similarly the other fingers respectively have $2 \times 4 = 8$, $2 \times 4 = 8$, $3 \times 4 = 8$, and $2 \times 5 = 10$ parameters. Considering that the little finger is not used for rotation, we can say the total dimension for leaving this motion should be

$d_{\mathbf{m}} = 12 + 8 + 8 + 12 = 40$

Therefore, the parameter organization of this application is in the form illustrated as Fig. 6.8. Specifically, the idle motion (meta motion 9) only has one parameter "start time" affecting the manipulation process, other corresponding parameters are dummy and our learning program will skip them automatically. For the sake of easy calculation, we would like to keep the dummy parameters in the total dimension $d_{\mathbf{m}}$.

6.3.2 PSO and LSRS model for babbling learning

Like other evolutionary algorithms, PSO and LSRS have the concept of a group involving many candidates (particles) to approach the goal solution. However, instead of eliminating weak and then generating new candidates, these two algorithms have other mechanisms to improve their candidates. In PSO, all candidates update themselves by moving to new solutions as referring to the current best solution (or the center of a best solution cluster). On the other hand, LSRS updates the boundaries so the candidates are limited to the new solution domains. According to our background, the standard procedures of PSO and LSRS are illustrated as Alg. 4 and Alg. 6. Furthermore, in order to make them work effectively, slightly different versions from the standard PSO and LSRS are proposed as Alg. 5 and Alg. 7.

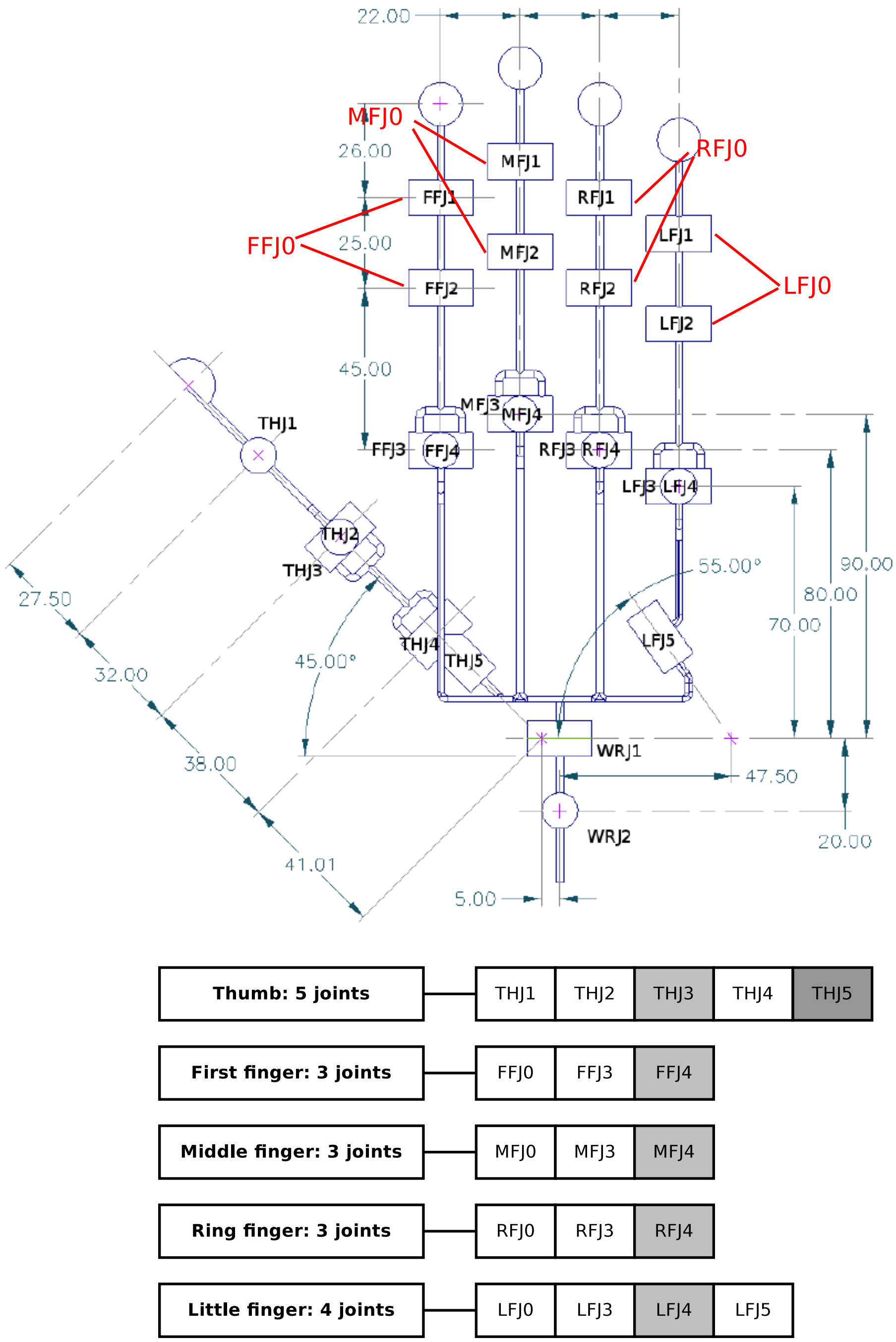


Figure 6.6: The joint map for the corresponding fingers of the Shadow hand. The kinematics diagram of Shadow hand is from (ShadowRobot, 2013). The distal and middle links (J1 and J2) from the first to the little finger are coupled, a single value J0 controls both J1 and J2. The abduction joints are colored in grey, and particularly THJ5 is a rotational joint.

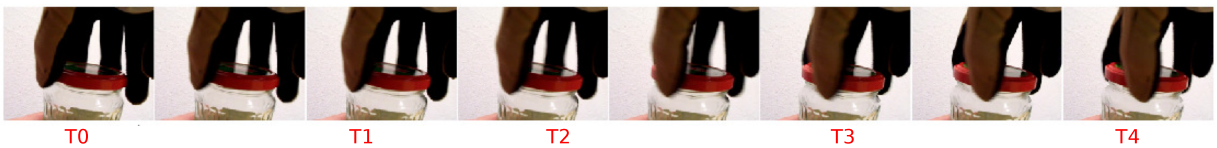


Figure 6.7: Cylinder rotation learning from screw-cap rotation. A demonstrator wearing a data-glove rotates the screw-cap of a bottle. The thumb, first, middle and ring finger are involved in this application. We mark some timestamps for Fig. 6.8.

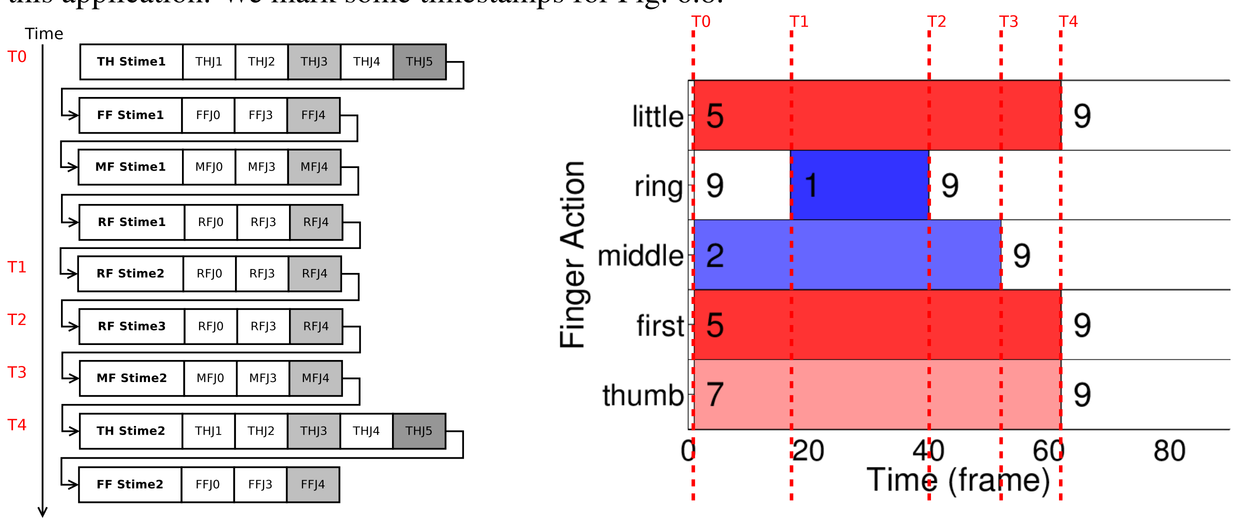


Figure 6.8: The parameter organization according to the action gist of screw-cap rotation in Fig.6.7. Altogether there are 40 parameters for this case. We note that the meta motions are sorted by their start time.

Algorithm 4 Conventional PSO for in-hand motor babbling learning

Require: Maximum of iteration N_{Iter}
Require: Particle number $N_{particle}$
Require: Particle dimension $d_{\mathbf{m}}$

for each particle $i = 1, \ldots, N_{particle}$ **do**
  Initialize the particle's position with a **uniformly distributed** random vector $x_i \sim U(J_{lo} - J_{now}, J_{up} - J_{now})$. Here J_{lo} and J_{up} are the robot joint limitations, J_{now} is the current joint angle.
  Initialize the particle's best known position to its initial position: $p_i \leftarrow x_i$
  Simulate a trial as x_i to get $f(p_i)$
  if $f(p_i) < f(g)$ **then**
    update the swarm's best known position $g \leftarrow p_i$
  end if
  Initialize the velocity: $v_i \sim U(-|J_{up} - J_{lo}|, |J_{up} - J_{lo}|)$
end for
$counter \leftarrow 1$
repeat
  $counter \leftarrow counter + 1$
  for each particle $i = 1, \ldots, N_{particle}$ **do**
    for each dimension $d = 1, \ldots, d_{\mathbf{m}}$ **do**
      Pick random numbers: $r_p, r_c \sim U(0, 1)$
      Update the velocity: $v_{i,d} \leftarrow \omega v_{i,d} + \varphi_p r_p (p_{i,d} - x_{i,d}) + \varphi_c r_c (c_d - x_{i,d})$ ▷ c is the center of a best solution cluster
    end for
    Update the particle's position: $x_i \leftarrow x_i + v_i$
    Simulate a trial as x_i to get new $f(x_i)$
    if $f(x_i) < f(p_i)$ **then**
      Update the particle's best known position: $p_i \leftarrow x_i$
      if $f(p_i) < f(g)$ **then**
        Update the swarm's best known position: $g \leftarrow p_i$
      end if
      Reorganize the best solution cluster c
    end if
  end for
until $count >= N_{Iter}$ or $f(g) = max$
Output the best found solution g

Algorithm 5 Incremental PSO for in-hand motor babbling learning

Require: Maximum of iteration N_{Iter}
Require: Particle number $N_{particle}$
Require: Particle dimension $d_{\mathbf{m}}$
Require: Joint variation limitation A_{lim} ▷ ***1****

for each particle $i = 1, \ldots, N_{particle}$ **do**
  Initialize the particle's position with a **uniformly distributed** random vector $x_i \sim U(b_{lo}, b_{up})$. Here $b_{lo} \leftarrow 0$. Besides, $b_{up} \leftarrow 1$ when x_i is a start time parameter, in other cases b_{up} has a corresponding value from A_{lim}
  Initialize the particle's best known position to its initial position: $p_i \leftarrow x_i$
  Simulate a trial as x_i to get $f(p_i)$ ▷ ***3****
  if $f(p_i) < f(g)$ **then**
    update the swarm's best known position $g \leftarrow p_i$
  end if
  Initialize the velocity: $v_i \sim U(-|b_{up} - b_{lo}|, |b_{up} - b_{lo}|)$
end for
$counter \leftarrow 1$
repeat
  $counter \leftarrow counter + 1$
  for each particle $i = 1, \ldots, N_{particle}$ **do**
    for each dimension $d = 1, \ldots, floor(f(p_i))$ **do** ▷ ***2****
      Pick random numbers: $r_p, r_c \sim U(0, 1)$
      Update the velocity: $v_{i,d} \leftarrow \omega v_{i,d} + \varphi_p r_p (p_{i,d} - x_{i,d}) + \varphi_c r_c (c_d - x_{i,d})$ ▷ c is the center of a best solution cluster
    end for
    Update the particle's position: $x_i \leftarrow x_i + v_i$
    Simulate a trial as x_i to get new $f(x_i)$ ▷ ***3****
    if $f(x_i) < f(p_i)$ **then**
      Update the particle's best known position: $p_i \leftarrow x_i$
      if $f(p_i) < f(g)$ **then**
        Update the swarm's best known position: $g \leftarrow p_i$
      end if
      Reorganize the best solution cluster c
    end if
  end for
until $count >= N_{Iter}$ or $f(g) = max$
Output the best found solution g

Algorithm 6 Conventional LSRS for in-hand motor babbling learning

Require: Maximum of iteration N_{Iter}
Require: Iteration number of Line Search N_{IterLS}
Require: Candidate number $N_{particle}$
Require: Candidate dimension $d_{\mathbf{m}}$
procedure LINE SEARCH
  for $k := 1$ To N_{IterLS} **do**
    for each particle $i = 1, \ldots, N_{particle}$ **do**
      for each dimension $d = 1, \ldots, d_{\mathbf{m}}$ **do**
        Set a new candidate $x_{tmp} \leftarrow x_i - \frac{20}{2^k}$
        Simulate a trial as x_{tmp} to get new $f(x_{tmp})$
        if $f(x_{tmp}) < f(p_i)$ **then**
          $x_i \leftarrow x_{tmp}$
          $f(p_i) \leftarrow f(x_{tmp})$
        end if
      end for
    end for
  end for
end procedure
procedure RE-START
  Find the current best solution as well as the corresponding parameters p_b
  for each dimension $d = 1, \ldots, d_{\mathbf{m}}$ **do**
    Change the d-th component of p_b very slightly, supposing that the original value is $x_{b,d}$, now it becomes $x'_{b,d} = x_{b,d} - \Delta$ and $x''_{b,d} = x''_{b,d} + \Delta$
    Simulate two trials as x', x'' to get $f(x'), f(x'')$
    if $f(x') < f(x'')$ **then**
      Update the d-th component of the upper limitation as $x_{b,d}$
    else if $f(x') > f(x'')$ **then**
      Update the d-th component of the lower limitation as $x_{b,d}$
    end if
  end for
end procedure
Initialize the boundary as Alg. 4
for *loop* $:= 1$ To N_{Iter} **do**
  Randomize the parameters as current boundaries as Alg. 4
  LINE SEARCH
  RE-START
end for

Algorithm 7 Incremental LSRS for in-hand motor babbling learning

Require: Maximum of iteration N_{Iter}
Require: Iteration number of Line Search N_{IterLS}
Require: Candidate number $N_{particle}$
Require: Candidate dimension $d_{\mathbf{m}}$
Require: Joint variation limitation A_{lim} ▷ ***1****
procedure LINE SEARCH
  for $k := 1$ To N_{IterLS} **do**
    for each particle $i = 1, \ldots, N_{particle}$ **do**
      for each dimension $d = 1, \ldots, floor(f(p_i))$ **do** ▷ ***2****
        Set a new candidate $x_{tmp} \leftarrow x_i - \frac{20}{2^k}$
        Simulate a trial as x_{tmp} to get new $f(x_{tmp})$ ▷ ***3****
        if $f(x_{tmp}) < f(p_i)$ **then**
          $x_i \leftarrow x_{tmp}$
          $f(p_i) \leftarrow f(x_{tmp})$
        end if
      end for
    end for
  end for
end procedure
procedure RE-START
  Find the current best solution as well as the corresponding parameters p_b
  for each dimension $d = 1, \ldots, floor(f(p_b))$ **do** ▷ ***2****
    Change the d-th component of p_b very slightly, supposing that the original value is $x_{b,d}$, now it becomes $x'_{b,d} = x_{b,d} - \Delta$ and $x''_{b,d} = x''_{b,d} + \Delta$
    Simulate two trials as x', x'' to get $f(x'), f(x'')$ ▷ ***3****
    if $f(x') < f(x'')$ **then**
      Update the d-th component of the upper limitation as $x_{b,d}$
    else if $f(x') > f(x'')$ **then**
      Update the d-th component of the lower limitation as $x_{b,d}$
    end if
  end for
end procedure
Initialize the boundary as Alg. 5
for *loop* $:= 1$ To N_{Iter} **do**
  Randomize the parameters as current boundaries as Alg. 5
  LINE SEARCH
  RE-START
end for

Action gist limits the exploration space

Action gist decreases the searching space before we start iteration.

We can refer to Fig. 3.1, and find that meta motions determine the joint varying directions. Suppose that the joint limits of the robot hand is $[\mathbf{J}_{lo}, \mathbf{J}_{up}]$, and for current hand pose the joint angles are $\mathbf{J}_{now}$. Originally the searching boundaries should be $[\mathbf{J}_{lo} - \mathbf{J}_{now}, \mathbf{J}_{up} - \mathbf{J}_{now}]$ like Alg. 4. Restricted by the meta motion quadrants (Fig. 3.1), now the searching boundaries have to be $[0, \mathbf{J}_{up} - \mathbf{J}_{now}]$ or $[0, \mathbf{J}_{now} - \mathbf{J}_{lo}]$; so we employ symbol A_{lim} as the upper boundary which is equal to $J_{up} - J_{now}$ or $J_{now} - J_{lo}$ (NOTE: J_{up}, J_{now} and J_{lo} are elements of $\mathbf{J}_{up}$, $\mathbf{J}_{now}$ and $\mathbf{J}_{lo}$). By this way, bi-directional search space is limited to one direction. Considering the dimension of the joint angle control parameters, we can say action gist considerably helps us with decreasing the exploration space at the beginning. We mark this point in Alg. 5 and Alg. 7 as ***1****. However, this exploration space is still very large, so we need more techniques for our application.

Incremental parameter adjustment for parameter exploration

In the previous Section 6.3.1 we introduced the control parameters for our learning modules. As we know, we use the state gist to check the correctness of the motion execution; once the extracted criteria differ from scheduled, we can stop this trial to save time. Meanwhile, we notice that only the control parameters applied from the start criterion to the current criterion have influenced the robotic hand action; the correct functioning of the remaining control parameters cannot be proved. Therefore, even though the original PSO or LSRS will update every parameter in order to make them converge to a better solution, we should not modify any unproved parameters. For example, we have 4 meta motions and 3 meta criteria. Now 3 motions are done, the corresponding 2 criteria are verified. The remaining meta motion should be proved by the last criterion. Since the last criterion is not checked yet, we cannot modify the parameters of the last meta motion.

We mark this point in Alg. 5 and Alg. 7 as ***2****, where function floor(r) returns the nearest integer just below the real number r. The value of $f(x)$ will be explained in the next section 6.3.2. Because the value provides us with the information of how many parameters are involved, we can avoid updating the unused parameters. Only when the current parameters complete the state test and proceed to the next state test, the successive parameters are activated to join the learning iteration. Therefore, the learning progress is considered as an incremental process as shown in Fig. 6.9.

Evaluation function

The evaluation function gives an overall estimation of the learning success after each simulated trial. We connect it with state gist, for the details of state gist we can refer to Chapter 4.

So far the criteria that we have applied are the object position information, object rotation information, and finger tip contact information. For some cases, we just consider the object position, and assume that the object is correctly moved if the object does not fall until after the manipulation movement.

Therefore, we have two evaluation functions. The first one only has one check point at the end of the trial. In this case we just make the evaluation function return how long the object is

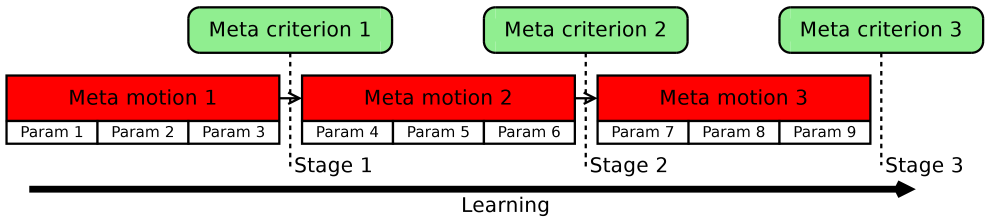


Figure 6.9: Incremental learning with state-action gist. The control parameters can be updated only when the meta criteria are checked. For example, at the beginning we can only learn parameters 1 to 3. When criterion 1 is passed, we start to learn parameters 1 to 6. Once the meta criterion 2 is checked and it fits the expected result, we are able to update parameters 1 to 9. When the meta criterion 3 is achieved, the entire in-hand manipulation task is successful.

stable kept in hand. The second evaluation function is required for Alg. 5 and Alg. 7 as ***3****, it is an incremental function with respect to the number of activated parameters. Before the description of this function, we should declare the mechanism of the score accumulation.

We divide the entire in-hand manipulation process into several parts as the end time of each meta motion (for this point we can also refer to Fig. 6.9). At the end of each part we use several criteria to check whether the motions are correct. As long as the finger motions fully passes the checking of one stage, we can activate the successive control parameters for the next stage. According to Sec.6.3.2, we will set the **basic score** S_b as the number of activated parameters. Here we can employ Fig. 6.9 to explain this point. When stage 1 is passed, $S_b = 3$, and when stage 2 is clear, $S_b = 6$. Finally we achieve stage 3, $S_b = 9$.

If the hand motions cannot pass a specific stage, we can estimate how well the current motion has achieved the goals. During the motion execution, we take samples of the properties we are interested in, e.g., object position, or which finger tips touch the object. Supposing that the sample record is $\mathbf{s}$, while the benchmark for checking is $\mathbf{t}$. We can sum up and rescale these values according to the length L, so as to obtain the **achievement score** S_a:

$$S_a = \sum_{i=1}^{L} q_i / L, \quad q_i = \begin{cases} 1 & \text{if } s_i = t_i \\ 0 & \text{if } s_i \neq t_i \end{cases} \tag{6.3}$$

As a result, we can get an overall score for learning feedback:

$$S_t = S_a + S_b \tag{6.4}$$

Inside PSO, LSRS and beyond

Through Alg. 4 and Alg. 5, we see that the PSO parameters which can be tuned are: the iteration number N_{Iter}, the particle number $N_{particle}$, and the velocity control ω, φ_p and φ_c. Even though we can refer to some literatures on how to set the values (Shi and Eberhart, 1998b; Clerc and Kennedy, 2002; Helwig and Wanka, 2007; Piperagkas et al., 2012), we should consider their experience with current situation. We can separate them into two groups and discuss them individually.

N_{Iter} and $N_{particle}$: Because the particle dimension is a variable depending on the complexity of the manipulation, The iteration number N_{Iter} and the particle number $N_{particle}$ will go along with the dimension of the control parameters $d_{\mathbf{m}}$ in Eq. 6.2. Higher values increase the possibility to find a good solution, but meanwhile take more time. Theoretically, if we want to explore the complete parameter space from the beginning, i.e. for each parameter the algorithm starts searching from the parameter boundary, $N_{particle}$ should be $2^{d_{\mathbf{m}}}$. Since the value is exponentially increasing with the dimension of the control parameters, we usually initialize the particles with random values instead.

ω, φ_p and φ_c: We can find these three PSO parameters in the velocity equation:

$$v_{i,d} \leftarrow \omega v_{i,d} + \varphi_p r_p (p_{i,d} - x_{i,d}) + \varphi_c r_c (c_d - x_{i,d}) \tag{6.5}$$

where r_p and r_c take random values in $[0, 1]$; $p_{i,d}$, $x_{i,d}$ and c_d are respectively the components of the current best solution, current solution and the group solution center (PSO clusters particles into several group so the particles will not be limited in local bests). Generally, ω, φ_p and φ_c control the speed of a particle flying to a better solution. According to the previous listed literatures, we try different configurations as $\omega = 0.4,\ 0.8,\ 1.0,\ 1.2$; φ_p and φ_c are both set to 0.4, 0.8 or 1. However, we do not find any configuration outperform others, neither in the best solutions nor in the convergence time. Currently, we keep them as $\omega = 0.4$, $\varphi_p = 0.4$ and $\varphi_c = 0.4$ for all experiments in order to make the particles flying slowly. We compare this case with people walking: when we walk slowly, we may find surprise on the way. Anyway, this explanation is fully reasonable. Consequently, how to configure the PSO parameters can be a future work.

Besides, from Alg. 6 and Alg. 7 we notice that the LSRS algorithm also has a set of parameters. The iteration number N_{Iter}, $N_{particle}$ and $d_{\mathbf{m}}$ are the same as in the PSO algorithms. For the inner iteration number N_{IterLS} of the Line-Search module, we can set it a number below 6. The reason is: we can refer to the 5th in Alg. 6 or Alg. 7, where a new candidate $x_{tmp} \leftarrow x_i \frac{20}{2^k}$; since the joint angle is at most around $90°$, and the manipulation result will not be as sensitive as $0.1°$, 6 is a proper value. Similarly, Δ can be set to 0.1 for the time variation, but to 1 for joint angle variation.

So far we introduce two modified algorithms in terms of our in-hand manipulation motor babbling learning. They are typical algorithms: PSO focuses on parameter variation as regularities, while LSRS is dedicated to the limiting parameter boundaries. We note that no matter what a new algorithm becomes available, we only need to integrate it with the parameter initialization ***1****, the dimension of parameter updating ***2****, and the executing function ***3****. Afterwards, the new method becomes an effective module for babbling learning.

6.4 Experiment

In this section, we present the babbling learning simulation using the ROS[2] Shadow stack and the Gazebo simulator[3], as well as the practical results. The visual and physical characteristics

[2] http://www.ros.org
[3] http://gazebosim.org/

Table 6.1: Action gist paired state gist definition to rotate a cylinder

Checking moment	Criteria
The ring finger is about to start meta motion 1	The cylinder does not move more than 5cm away from the original position, 4 fingers keep contact to the cylinder
The middle finger completes meta motion 2	The cylinder does not move more than 5cm away from the original position, 4 fingers keep contact to the cylinder, the cylinder is rotated by 90 degrees
The thumb and the first finger complete meta motion 5 and 7	The cylinder does not move more than 5cm away from the original position, 4 fingers keep contact to the cylinder

of the hand and the object in the gazebo world are close to the real ones, as well as noises. Therefore, even we use the same configuration, the results are sometimes different. For each iteration, we initialize the hand in a stable state that grasps and holds the target object by the fingers. Afterwards, the robot hand starts to try the motion sequence by itself, and we can see the object being manipulated.

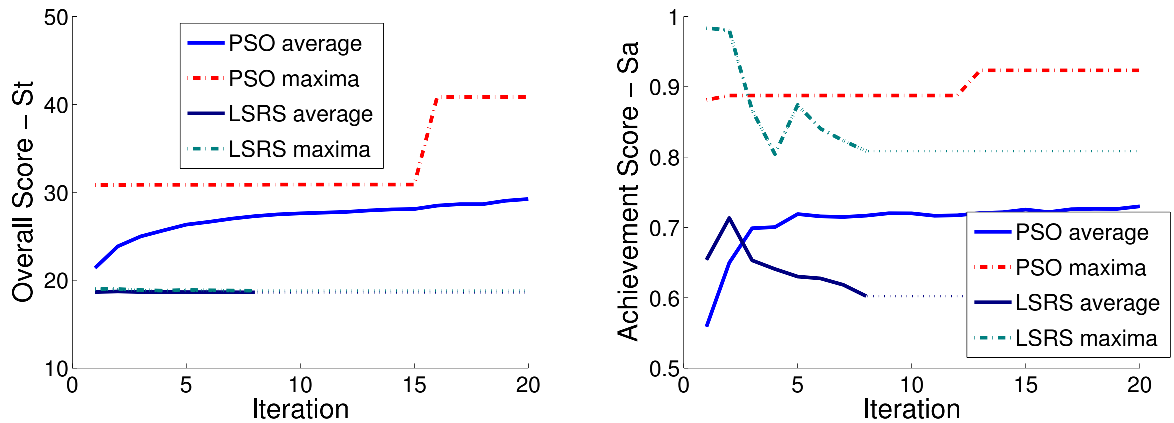
In order to explain the process better, we can refer to some related videos on the website (Cheng, 2013).

6.4.1 Cylinder rotation

Since action gist contains the patterns reflecting joint angle variation, but without relating to the manipulated object, we are going to use the demonstration of screw-cap rotation (as Fig. 6.7) to instruct the rotation of a cylinder. We can find that the thumb, first, middle and ring finger are used in this application; the four fingers tightly grip the screwcap and move. Therefore, when the knowledge is translated into a cylinder rotation, we should also see that four fingers are being used and the cylinder is being rotated in the same direction as the screw-cap in the human demonstration.

For the knowledge preparation, the action gist is automatically extracted from data-glove values, and we selected the popular one among the demonstration set as shown in Fig. 6.8. Meanwhile with state gist we can generate a script to guide the robot learning. Nevertheless, we did not prepare tactile sensing for the demonstration. In order to save time, based on the fact “the involved fingers keep contacting the object” we manually define the state gist as shown in Tab. 6.1. Fortunately, as we suggest to use simple criteria, we can immediately generate the corresponding criteria as listed. In each learning iteration, we record the hand and the object variation, as well as the contact information in the Gazebo world. Using the record we can evaluate the state achievement according to Eq. 6.4.

Besides the particular parameters of both algorithms are set to what has been mentioned in section 6.3.2, we set the common parameters as $N_{Iter} = 20$, $N_{particle} = 200$. In this case, suppose that one trial is 1 minutes (including hand posture initialization, object posture initialization, manipulation, and evaluation), the total time consuming is $20 \times 200 = 4000$ min-



(a) Overall score

(b) Achievement score

Figure 6.10: The learning rates of rotating a cylinder anticlockwise. According to Eq. 6.4 we have the overall score S_t as the left figure. For PSO, we can see that the quality of the parameters is gradually improving with iterations. For LSRS, the exploration is not so successful. The reason is: all of the parameter boundaries converge to the same values, LSRS learning terminates at the 8th iteration. Besides, according to Eq. 6.3 we have the achievement score S_a as the right figure. Here we find the LSRS curves have a better beginning but soon decrease. It probably indicates that LSRS converges to a wrong parameter domain.

utes, i.e., over 2 days. As mentioned in section 6.3.1, we already know the parameter dimension of this application is $d_{\mathbf{m}} = 40$. Therefore, $\max S_b = 40$, and the value of Eq. 6.4 is $\max S_t = \max S_a + \max S_b = 41$. Because the applied algorithm can not always get the best solution, we can say if a solution is close to 41, it is a good solution.

After many iterations, we achieve a final learning result as in Fig. 6.10(a). Here we note that LSRS fails for misadjusted boundary control. When all the boundaries are limited, the algorithm will terminate. Besides, we understand that LSRS can drive the boundary search to a wrong area, this is an intrinsic issue in the algorithm. Therefore, the common solution for this case is restarting learning, or varying the dead boundaries for new iterations.

We also have interest with the achievement score as Eq. 6.3, hence we inspect them as shown in Fig. 6.10(b). The basic score S_b only reflects how many parameters should be concerned in the next iteration, but achievement score S_a represents the similarity between the current situation and the criteria. Thus, in this figure we find that the average score of PSO is around 0.7, so it indicates that the swarm quality does not improve very much with iteration.

According to this application, we can find a best solution as shown in Fig. 6.11, Fig. 6.12, and Fig. 6.13. From these figures, we assume the babbling learning finds a good result.

Besides this cylinder in this size, we also apply the same action gist to another sized one illustrated as Fig. 6.14. Here we only use PSO, and the particle number remains at 200, but the iteration is increased to 40 since we are unsure of the validity. Finally we have the learning curves as shown in Fig. 6.15(a) and Fig. 6.15(b), We can find the details of the learned best solution from Fig. 6.16, Fig. 6.17, and Fig. 6.18. Since the thinner cylinder is also rotated, we

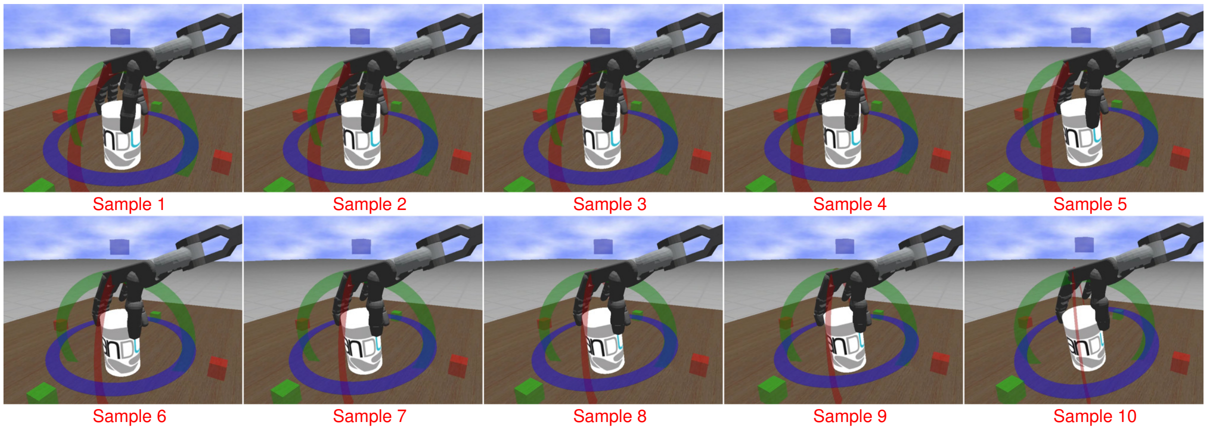


Figure 6.11: Cylinder rotation final result according to the action gist of screw-cap rotation anticlockwise. We sample the simulation with equal time intervals, and we can see that a textured cylinder is manipulated by a shadow hand on a table. We also enable the virtual axes of the cylinder to confirm whether the cylinder is moved. Based on the texture variation and the axes rotation, We can say that the object is rotated as a certain degrees.

can say that the PSO learning process finds a good solution.

Furthermore, we want to test our simulated result in our Shadow hand. However, even though we try to apply the simulation as similar as the real world, the simulated errors still exist. In order to solve this issue, we can manually tune the best simulated solution, or refer to an interactive algorithm mentioned in Chapter 7. Since our BioTac hand has tactile feedback, we can apply adaptive control. As a result, the BioTac hand rotates a tea caddy in the same size of the simulated cylinder. We can refer to Fig. 6.19 to see the photos, and from Fig. 6.20 we can learn the tactile feedback on the robot fingertips.

6.4.2 Star prism rotation

In order to see whether our framework is able to solve more finger-gaiting applications, we test a difficult scenario as shown in Fig. 6.21 (which is the first example in Chapter 3). The corresponding action gist is illustrated in Fig. 6.22. Because all fingers are used, according to Eq. 6.4 in Section 6.3.1, the parameter dimension of this scenario is 137. With the increase of meta motions, the state gist gets longer. In addition to the states that we have applied to the cylinder rotation, we take the contact transition into consideration for this application. For example, the criteria can be not only “the finger keeps (not) touching”, but also “the finger is going to (not) touch”.

The corresponding learning results are shown in Fig. 6.23(a) and Fig. 6.23(b). From the figure we can see the highest score is only around 44. Below the total dimension 137, this means the learning has failed to achieve the goal of the manipulation. According to our experience, we conclude this failure may be caused by the following factors:

1. The number of iterations. In Fig. 6.23(a) we can see there are several better solution

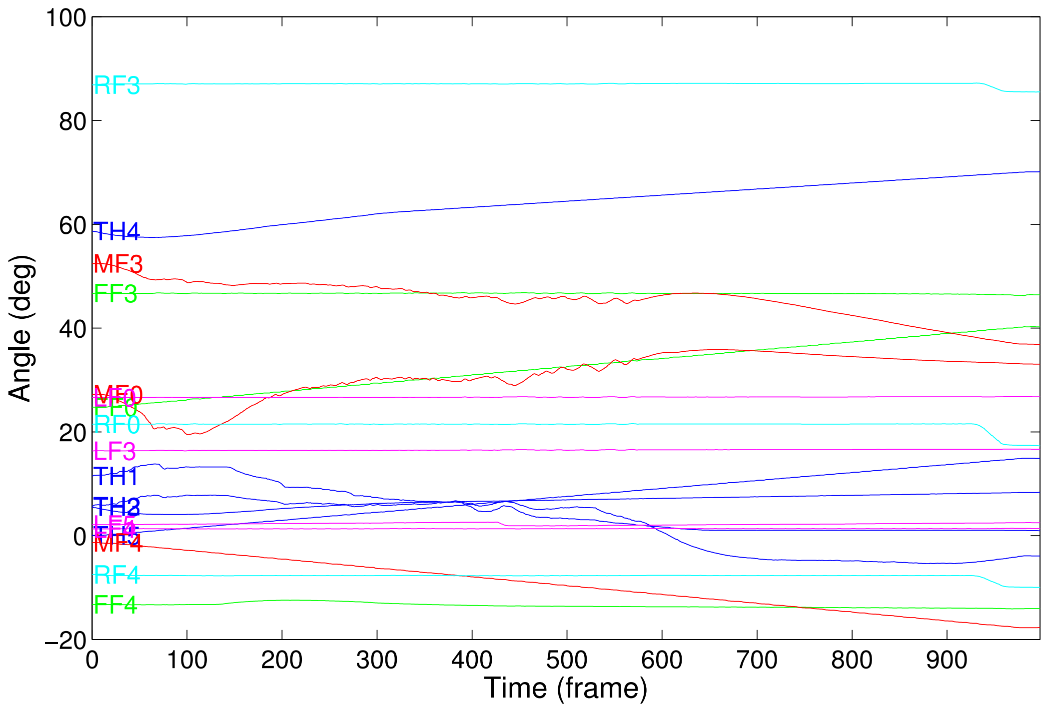


Figure 6.12: Simulated joint angle variation of cylinder rotation. We sample the contact information from the Gazebo world. Because we only concern in-hand movement, 18 finger joints are shown in this figure. According to the fact that we use linear model to control the hand (Eq. 6.1), the variations should be straight linesegments. However, meanwhile we use adaptive control (see Section 6.2), the finger will slowly move to touch/leave the target object depending on the contact. An typical example is between the 500th frame to the 700th frame, **MF0** (in red color) climbs up gradually and makes a curved trajectory.

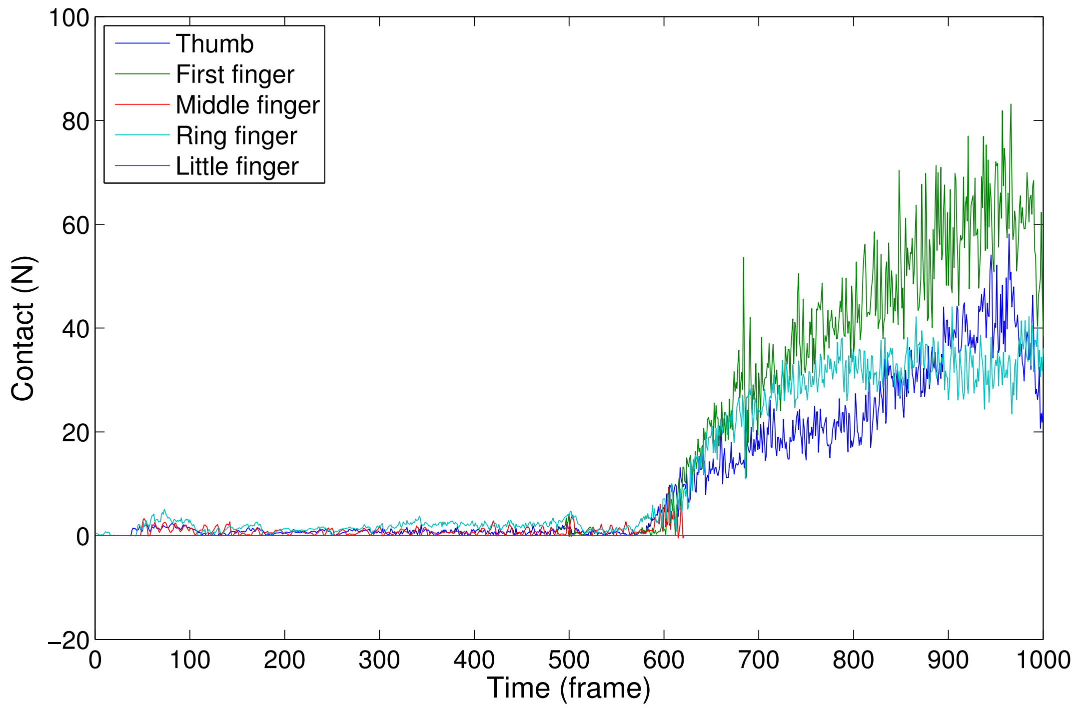


Figure 6.13: Simulated contact information of cylinder rotation. Since our robot only has tactile sensors on the fingertips, we also capture the fingertip contact information in the Gazebo. We can see that after the 600th frame the contact of the thumb, first and ring finger become much larger than the middle finger, meanwhile the little finger does not touch anything as we expect. Theoretically, as long as the contact values are higher than zero, the achievement score S_a (Eq. 6.3) will count them. Force values above 10 N indicate invalid finger-object contacts, and are impossible on the real robot. This is dangerous for real robot control, so when we redesign the state gist, the contact criteria in Tab. 6.1 should be both "higher than a value" and "lower than another value". From this example, we can understand the importance of "simulation before real execution". Besides, since the middle fingertip does not have contact over time, we naturally connect this fact with Fig. 6.10(b), i.e., S_a is close to but not reach 1.

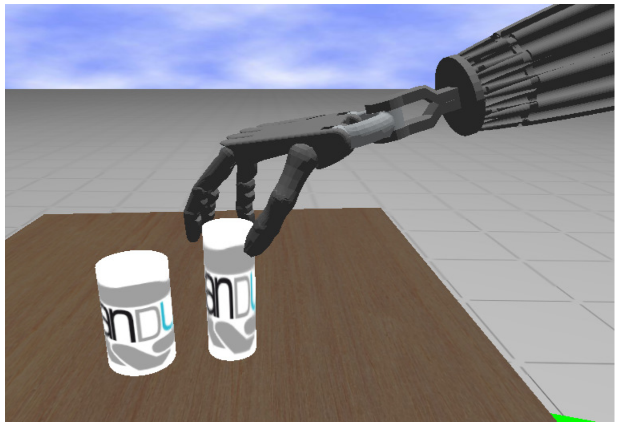
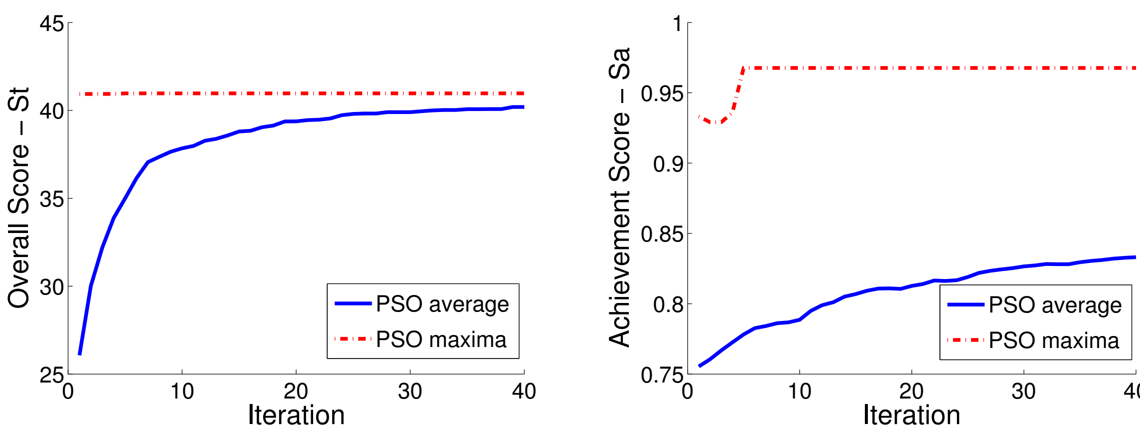


Figure 6.14: Different sized cylinders for rotation according to the scew-cap rotating action gist. Compared with the cylinder in the previous experiment, this cylinder is thinner and taller. Therefore, the initial hand pose is different.



(a) Overall score

(b) Achievement score

Figure 6.15: The learning rates of rotating a thinner cylinder. For the left figure, compared with Fig. 6.10(a), the initial particles gain higher score. As a result, the converge of the particles gets faster than the previous case. Considering the right figure, as mentioned in Fig. 6.15(a), the learning has a good start. Therefore, the average scores are higher than 0.7 in Fig. 6.10(b) of the previous experiment.

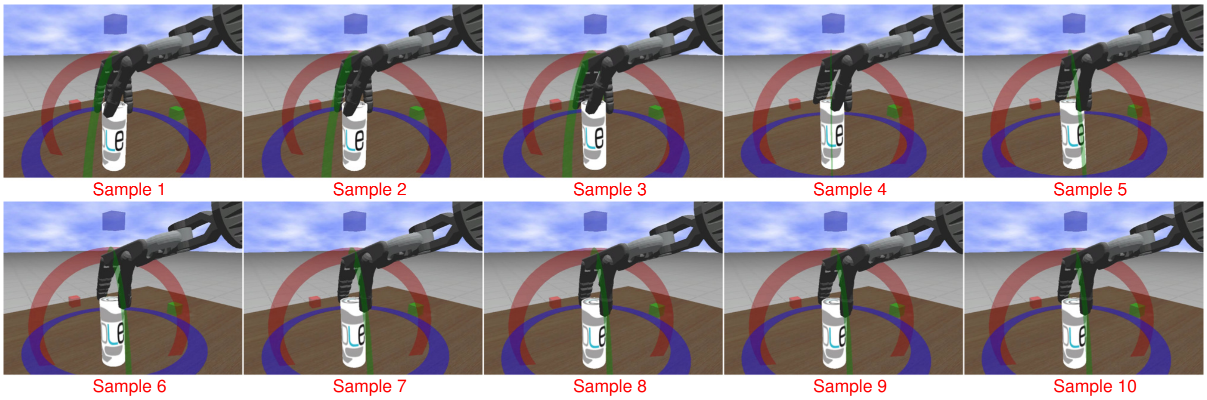


Figure 6.16: A thinner cylinder rotation according to the action gist of screw-cap rotation anti-clockwise. Based on the virtual axes variation, we confirm the cylinder is rotated.

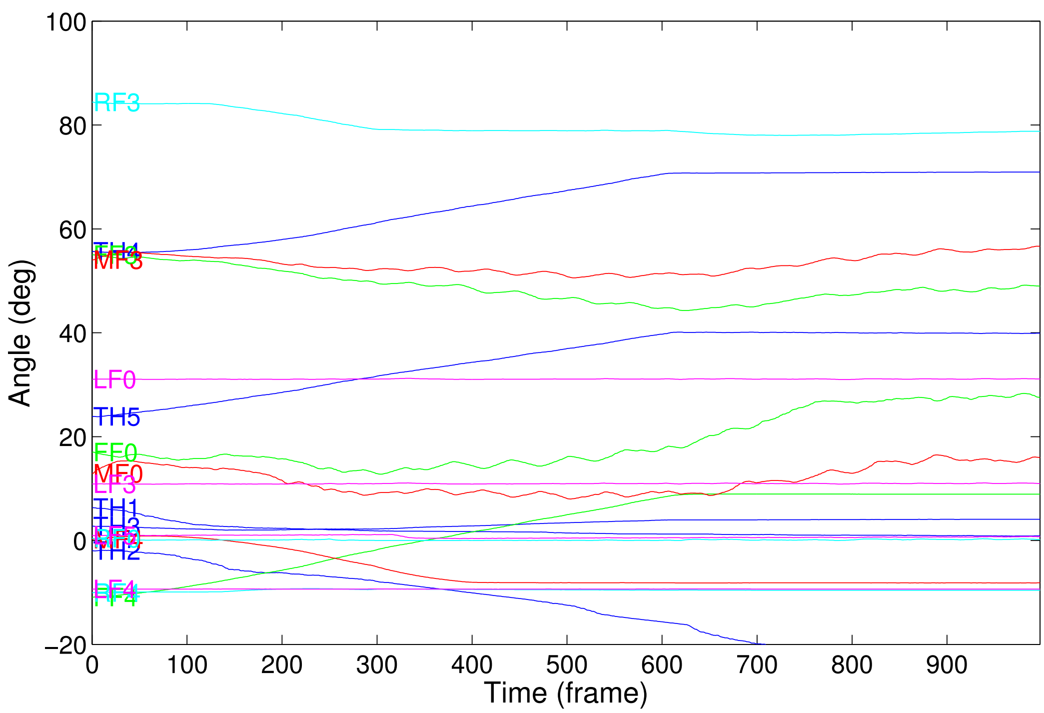


Figure 6.17: Simulated joint angle variation of a thinner cylinder rotation. We can find straight and curved trajectories in the joint variation. Connecting the fact from the other cylinder rotation (Fig. 6.12), we summarize: the fingers move as action gist (straight trajectories) and meanwhile are adaptively controlled with respect to the tactile feedback.

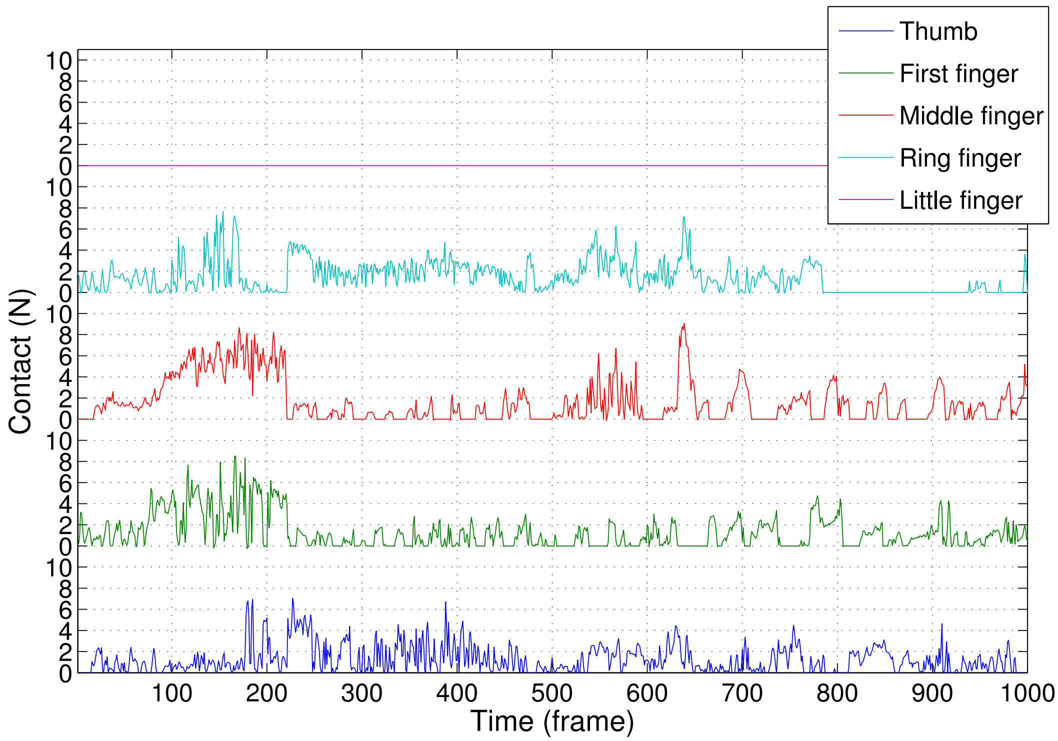


Figure 6.18: Simulated contact information of a thinner cylinder rotation. Since the value range is from 0 to 10 and their trajectories twine together, we have to illustrate them separately. The maximal contact of the thumb, first, middle and ring fingers are lower than 10, and the variations are more stable than Fig. 6.18. Besides, we can find the values are sometimes 0, this phenomenon support Fig. 6.15(a) and Fig. 6.15(b) that the current best solution is not perfect. Anyway, the cylinder is rotated as desired.

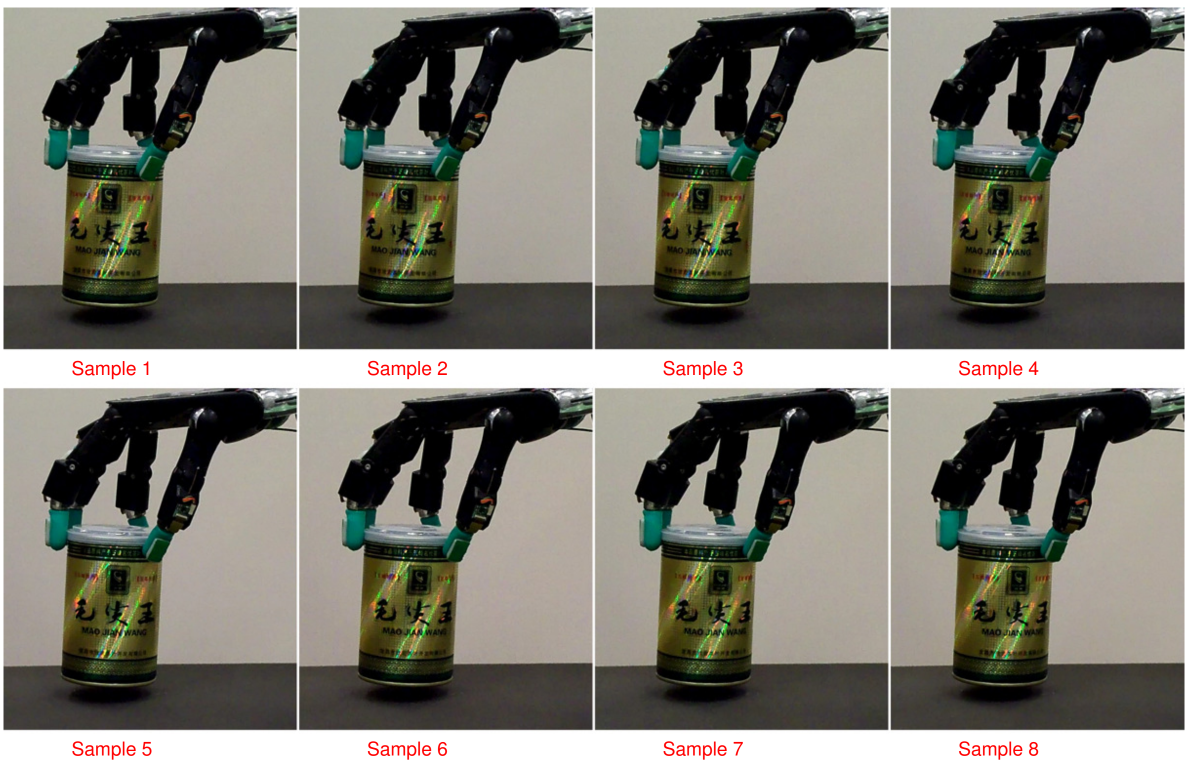


Figure 6.19: The rotation of a tea caddy with a BioTac hand. From the text on the caddy or the hand posture we can find the caddy is slightly rotated. We use the action gist of a screw-cap screwing to guide the rotation. We firstly simulate it with ROS and Gazebo as shown in Fig. 6.11. And then we move the best solution to command the BioTac hand. With the help of adaptive control (Section 6.2), the rotation result seems better than simulation.

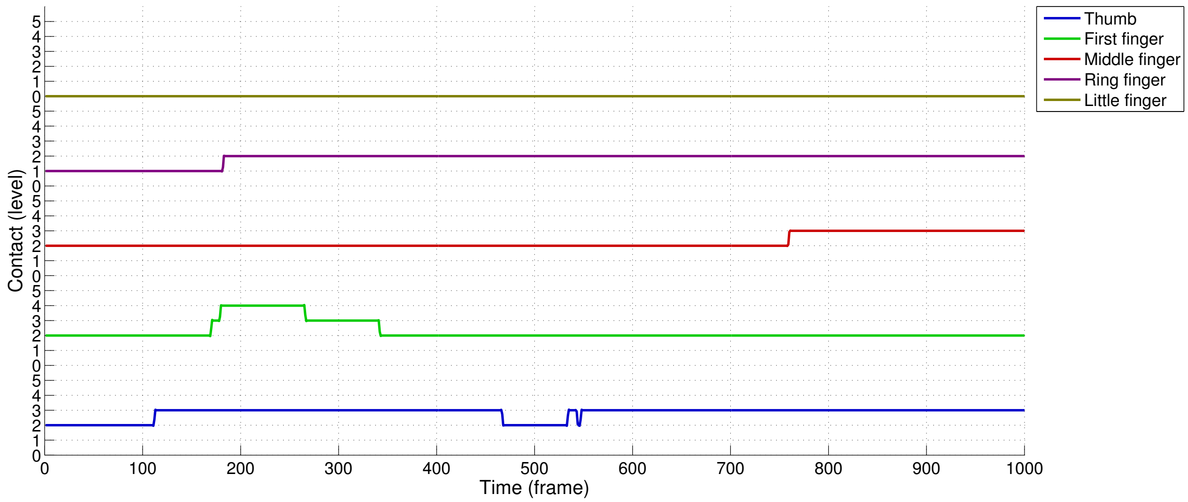


Figure 6.20: The tactile sensing from BioTac hand with respect to a tea caddy rotation. As introduced in Section 6.2, instead of raw value we define several levels of contact for the tactile sensors on the fingertips: from 0 - no contact to 5 - dangerous contact. Besides, each fingers will take action if its contact level is higher than 3. The value variations are more stable than simulated result (Fig. 6.13). According to our understanding, one reason is we adaptively control the fingers by realtime tactile feedback; another reason is that the BioTac hand is elastic, but the simulated hand is a rigid body.

Figure 6.21: Star prism gaiting learning from human demonstration. In this scenario the demonstrator uses the thumb, first, middle, and ring fingers to anticlockwise rotate the posture of the star prism.

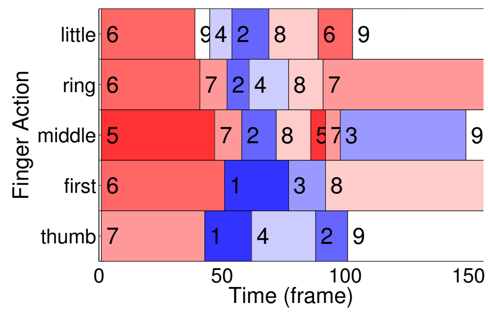


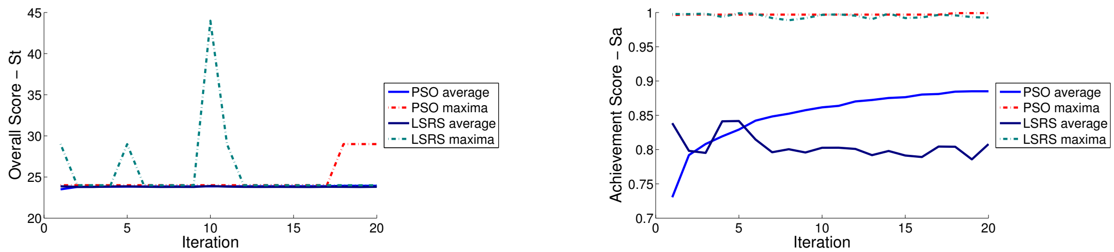
Figure 6.22: Star prism gaiting learning from human demonstration.

standing out from the average. As the characteristic of PSO, later other particles will fly close to the optimum. We believe the result will improve with more iteration, but meanwhile we cannot give a certain answer how many iterations are needed in each case. In this case, we suggest to balance the searching cost (e.g. the number of parameters or iterations) and our final goal.

2. Linear model of the mapping between the action gist and the finger joint control. Even though we use adaptive control to correct some wrong actions in realtime, we agree that this mechanism has a blind area. E.g. from meta motion 1 to 4 of the thumb in Fig. 6.22, it should decrease the first two joint angles to touch the star while the adaptive control mechanism does not. Therefore, we have two solutions, one is updating adaptive control by sensing tactile direction (currently only value), this can be a future work; the other one is to disable this module for this application. We have tried the latter method, but the learning curve increases slower while we cannot see a better solution in acceptable iterations.
3. The types of meta criteria. So far we only take some criteria into consideration. As Eq. 6.3 in Section 6.3.2, if the category of the criteria is not sufficient, the evaluation function cannot give a sensitive feedback to distinguish candidates. Thus, we will later investigate this issue systematically.

6.5 Summary

We proposed an in-hand manipulation babbling learning framework consisting of a priori knowledge (state action gist) and a self-learning mechanism (Particle Swarm Optimization and Line Search with Re-start). With the help of action gist, we decrease the angle joint control parameters as a sequence that includes start time and corresponding finger joint angle variations. The compressed sequence is easy to convert back into joint angle frames, and then we can use simulation to execute and examine the performance of this set of parameters. In addition, the modified optimization algorithms in terms of our application are employed as the cores of babbling learning, to refine the joint angle control parameters to achieve our goal of in-hand manipulation. Since 40 parameters experiment (cylinder rotation) is successful but 137 parameters experiment (star rotation) is failed, we assume that: with current techniques the gist babbling can compete for the



(a) Overall score

(b) Achievement score

Figure 6.23: The learning rates of rotating a star prism. In the left figure PSO only keeps the active parameters at the average of 23. That means for a common iteration in the learning process, only the first 23 parameters are taken into consideration. However, LSRS wins a maximum around 44. The problem is, after iterations LSRS cannot achieve a better score. We guess the reason is LSRS falls into a trap again. From the tendency of PSO (red dashs reach 30 after 17 iterations), we believe that it still has room to promote the score. Regarding the right figure, we notice that the maxima in this application is very close to 1. That means the finger motions quite fit the desired states. However, when the learning is going to proceed, the finger is failed to complete the next task. According to this scenario, the thumb should move around to another corner of the star prism, but actually it completely fails.

in-hand manipulation task with a parameter scale between 40 and 137.

In our points of view, an important advantage of action gist based babbling learning, is that we do not need to study the dynamic process inside the hand and the object. Once assigned the key information of the specific manipulation task, the robot hand can improve itself and we are free. In this case, any people without professional knowledge can teach the robot how to manipulate the object.

From the experimental result we can see that we have much room for extending our research beside the discussion in Section 6.4.2. The first direction is the learning core. So far we have tested PSO and LSRS, because they are typical algorithms from their concepts. In terms of their performances, we think PSO still hold its position of “the best candidate”. There are many more algorithms, if time permits, we will try more of them to make a further decision. The second direction is to try more application. Currently we concentrate on the cylinder-like object rotation and the angular object gaiting, as well as the experiments of Chapter 7. However, we have the foundation of gist extraction and generalization, it is not difficult to record and produce more state action gist according to other applications. And then we can use the proposed self-learning algorithm to find a good solution to manipulate the object in hand. Besides the second research direction, we realize that currently our framework instructed by action gist is not concerned with contact point scheduling (Saut et al., 2006). Further, we skip using inverse kinematics to plan the hand posture in spatial space. Without inverse kinematics the babbling learning may take more time on unnecessary trials, but meanwhile we save time on the research of sensory information processing in our preliminary attempt with state-action gist. In this case, integrating our current work with other available methods is also promising.

Chapter 7

Learning with Simple Human Reward

Contents

In Chapter 6, we have discussed how to babbling learn the in-hand manipulation skill with the help of state action gist. We notice an issue after simulation: even though we try to use a simulated environment as similar as the real world, but the gap between the simulation and this world never disappears. When we replay the learned "best" solution on the real robot hand, the result may not be the best. Instead of manually adjusting the parameters or circling round the simulation and real test (Deisenroth et al., 2011), this chapter proposes another solution: get help from a person, especially the person who was being the demonstrator.

We can evaluate each trial in the simulation. Similarly, as long as we have sensors in the real world, we can collect the sensory data and process them as a basic reward. Once we feel necessary to interfere the automatic evaluation, we can add our opinion to comprehensively enhance the reward. For example, our robot has tactile sensors on the fingertips, but for visual information we do not have sufficiently powerful cameras; at this moment we can interfere the robot self-learning to correct the perception of the robot.

Compared with robot, human is a stronger system with profound knowledge, powerful perception and comprehensive analyzing ability. However, we also understand human is not an accurate system. For example, we can clearly remember what happened 5 seconds ago better than 5 minutes ago; and we can confirm which one is good but we are not good at detailedly

Table 7.1: Features of the related researches on reinforcement learning with human feedback

Research	Num	Form	Calculation	Note
(Pilarski et al., 2011)	1	positive / negative	constant	
(Knox and Stone, 2008, 2009; Knox, 2012)	1	positive / negative	linear model	
(Vien and Ertel, 2012)	1	positive / negative	linear model	For continuous state/action space
(Judah et al., 2010)	20	positive / negative	logarithmic model	
(Goldwasser and Roth, 2011)	10	positive / negative	SVM training	
(Subramanian et al., 2011)	10	positive / negative	linear model	For hierarchical reinforcement learning

explaining why it is good. Therefore, we design the human feedback for each trial of real robot manipulation in a simple way:

1. For each manipulation trial, we can leave the robot to evaluate the reward score alone, or offer a piece of feedback if we are certain about the evaluation.
2. There are 3 pieces of feedbacks: this trial is "better" / "equal" / "worse" than the previous one.
3. According to the feedback, the reward score and the parameter domains are automatically adjusted.

In Chapter 6 we applied Line Search with Re-Start (LSRS) as a learning core algorithm, but it did not outperform Particle Swarm Optimization. However, inspired by the idea of boundary restriction from LSRS, this chapter is going to make use of user feedbacks to approximate the best solution of the real robot.

In order to present our ideas on how to realize this human feedback mechanism, we organize the coming sections as follows: Firstly we compare the proposed reward form with the related works in Section 7.1, and then propose a specific reinforcement learning framework with human reward in Section 7.2. Furthermore, the corresponding experiments are taken as Section 7.3. Finally Section 7.4 concludes this chapter.

7.1 Related work

We investigate the popular frameworks of reinforcement learning with human feedback. Generally, this area differs at the number of human trainers (Num), the form of the feedback (Form), as well as the relation between the human feedback and the entire evaluation (Calculation). We list the features of the related researches in Tab. 7.1. From the table we can see that there are some methods apply multiple trainers to give feedbacks. For the proposed method in this chapter, we employ only one expert at the moment. From the third column we find that most methods use simple feedback mechanism. Compared with their definition, our definition has an extra feedback "equal", this feedback will make the rewards of two trials equal. Since our scope is "original reward" + "human reward", we think the extra feedback "equal" is necessary. Finally, the significant difference of our method is in reward calculation, we will soon refer to this in the next section.

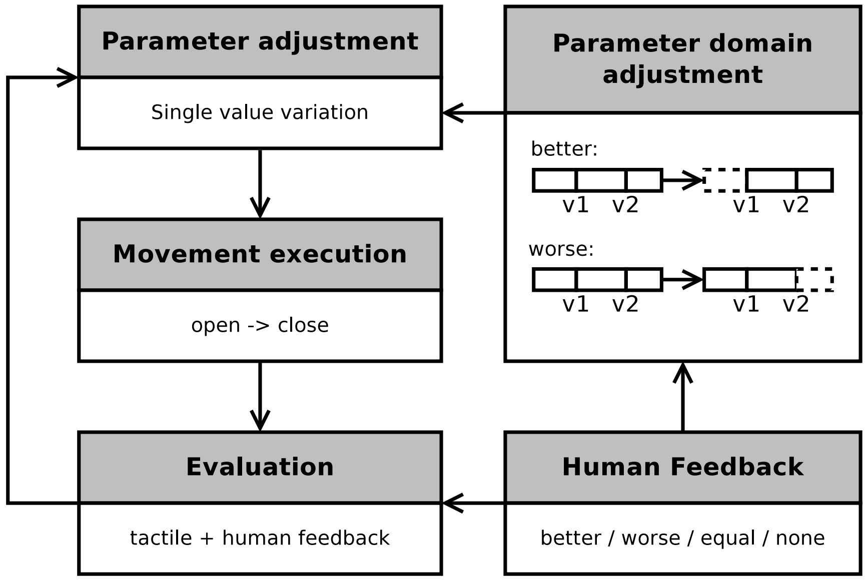


Figure 7.1: The framework of simple human reward based in-hand manipulation learning. The iterative loop is similar to a standard reinforcement learning process. The main difference is the way of dealing with the parameter variation. Besides stimulating the variation by the reward, the human feedback will directly control the parameter space. Therefore, everytime we only vary one parameter to facilitate the human teacher give an intuitive suggestion. In this case, assuming that the previous parameter is v_1, but the current value is v_2; When we say the current trial is better than the previous one, we can trim the parameter domain at the position of v_1; Similarly, we can limit the search area at v_2 if we get a "worse" remark. Because the parameter number is not in a large scale, so the method is applicable.

7.2 Simple human reward for reinforcement learning

We apply reinforcement learning framework to iteratively find a good solution to control the BioTac hand completing the manipulation task. Generally, we combine the parameter exploration with some simple human feedbacks. The human remark will not only impact the reward of evaluation, but also adjust the parameter boundary. In this case, we think one remark is not able to correspond to the relation of once updating many parameters. Thus, we just adjust one parameter for one trial. We show the basic idea in Fig. 7.1, and describe the corresponding modules in the following subsections.

7.2.1 Parameter adjustment

Every time we randomly select one parameter p_i for updating as long as this parameter still has space for searching. For this condition, we need the corresponding boundaries $[L_i, U_i]$; if $U_i - L_i$ is bigger than a certain degrees (e.g. 2 degrees), we can select p_i and update it. Instead of randomly selection, a substitution is to find the parameters significantly higher than the value average. The reason is a big value always has large room to explore.

Anyway, once we select a parameter p_i, we update its value as the corresponding reward R_i:

$$p_i' = p_i + f(R_i) \tag{7.1}$$

Generally, the parameter is set to a higher value if R_i is positive, or a lower value when R_i is negative. Hence we employ

$$f(R_i) = \begin{cases} \mathrm{rnd}(L_i - p_i, 0) & R_i < 0 \\ \mathrm{rnd}(0, U_i - p_i) & R_i > 0 \\ 0 & R_i = 0 \end{cases} \tag{7.2}$$

where $\mathrm{rnd}(\cdot)$ is a randomizing function.

7.2.2 Movement execution

The hand movement control has been described in Chapter 6. For every trial, we manually place the target object on the BioTac hand to have a similar beginning. Afterwards, the hand executes the movement as the parameter configuration. The entire process is not fully automatic, we may find the object initialization is not serious enough. On the technical aspect, we can make some markers on the object, so in each trial the marked position will be align on the specific region of the hand.

7.2.3 Human feedback

A person who is very familiar with the corresponding manipulation skill is beside the trials. After each trial this person can give a piece of feedback such as "better", "worse", or "equal" than the previous trial. The response is not compulsory for every trial, but once the person gives advice, the learning reward and the corresponding parameter adjusted in this trial will be updated.

7.2.4 Evaluation

We record the BioTac fingertip contact information in the movement executing process, and store them as an array $BioTac_{ij} = \{0, 5\}$ (where the value higher than 0 indicates contacted). Supposing that a better trial has longer time that the fingertips are in the contacted state, we simply mark $S_c = \sum_{i,j} BioTac_{ij}$ as the score from the tactile sensors. This score only reflects the contacted state, but we are not clear that whether the finger is touching the object, the other fingers, or other possible objects in the environment. Therefore, we prefer to believe the human feedback, which is the result consisting of sensory fusion and expert knowledge. Presuming that the previous overall score is S, and then the current score will be

$$S' = \begin{cases} S_c & feedback = \text{“}\texttt{better}\text{”}, S_c > S \\ S + \Delta & feedback = \text{“}\texttt{better}\text{”}, S_c \leq S \\ S_c & feedback = \text{“}\texttt{worse}\text{”}, S_c < S \\ S - \Delta & feedback = \text{“}\texttt{worse}\text{”}, S_c \geq S \\ S & feedback = \text{“}\texttt{equal}\text{”} \\ S_c & no\, feedback \end{cases} \tag{7.3}$$

In this way, we create the evaluation integrated with the sensory and human feedback. Besides, we have the corresponding reward

$$R_i = (p'_i - p_i)(S' - S) \tag{7.4}$$

7.2.5 Parameter domain adjustment

Only one parameter is involved with the updating in each trial, and each parameter just corresponds with one finger in the opening and closing movement. Therefore, the human feedback can govern the search space adjustment. For the case that the current trial is better than the previous trial, the new boundary will be

$$[L'_i, U'_i] = \begin{cases} [p_i, U_i] & p'_i > p_i \\ [L_i, p_i] & p'_i < p_i \end{cases} \tag{7.5}$$

Besides, for the case that the current trial is worse than the previous one, the new boundary will be

$$[L'_i, U'_i] = \begin{cases} [p'_i, U_i] & p'_i < p_i \\ [L_i, p'_i] & p'_i > p_i \end{cases} \tag{7.6}$$

Generally, we assume that the feedback "better" indicates that the previous parameter configuration is the worst case, while the "worse" feedback reflects the current parameter is the worst. By this way, we eliminate the unnecessary exploration.

Additionally, we can give a rough estimation that how many iterations required. Currently there are $d_{\mathbf{m}}$ parameters, as long as we give human feedback everytime, and the parameter is always in the center of the search space (e.g. $(L_i+U_i)/2$); We can assume that for each parameter the initial boundary is $[0, 90]$ degrees, but parameter difference under 1 degree make no different manipulation result. In this case, each parameter needs $\mathrm{ceil}(\log_2 90) = 7$ iterations to fix the search range (e.g. $U_i - L_i < 1$), so in total we at most need $7d_{\mathbf{m}}$ times human feedback. Specifically, when we are more sure about the searching boundaries, e.g. only $10°$ wide intervals, and then each parameter needs $\mathrm{ceil}(\log_2 10) = 4$ iterations. As a result, we just need $4d_{\mathbf{m}}$ times feedback in total.

7.3 Experiments

Besides the tea caddy rotation in Chapter 6, we also have other practical tests such as a marker pen spinning in a small range, screwdriver screwing, and triangular prism reorientation. We can refer to their movement in Fig. 7.2, Fig. 7.3, and Fig. 7.4. Besides, we list the corresponding action gist as Fig. 7.5.

In order to complete skill refinement, now we come to the final stage. For each simulated or raw solution, we confirm their validities and directly use them to control the BioTac hand (adaptive control mechanism is enabled as default). As expected, some manipulation skill such as marker triangle rotation and pen spin do not require further adjustment. Besides, we find that

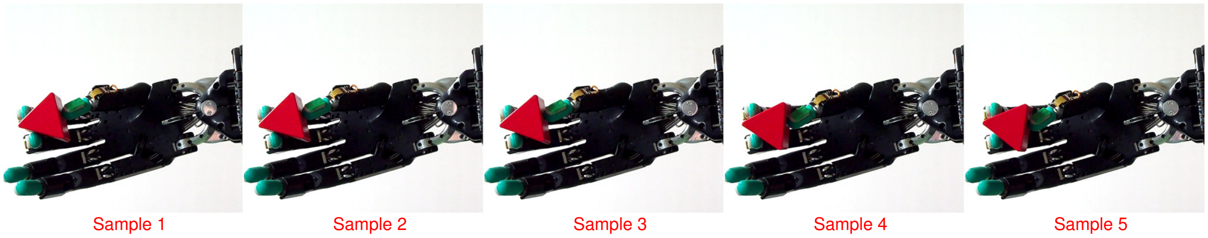


Figure 7.2: Triangle prism rotation by a BioTac hand. The thumb, first and middle fingers participate the movement. Three fingertip occupy three surfaces of the prism.

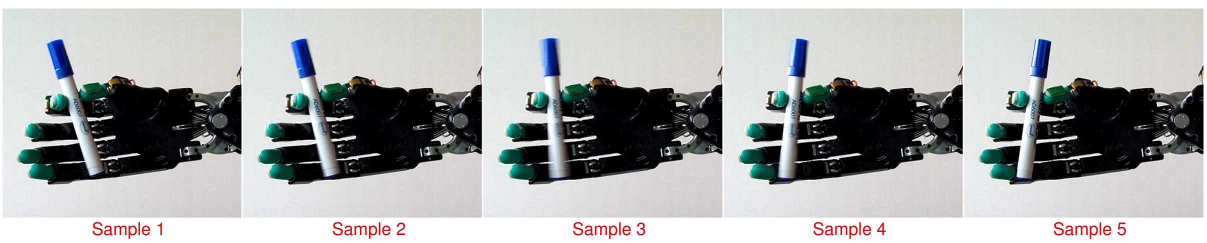


Figure 7.3: Marker pen slightly spin by a BioTac hand. The thumb, first finger participate the movement. When the two fingers change their spacial positions, the gripped marker pen is quickly redirected.

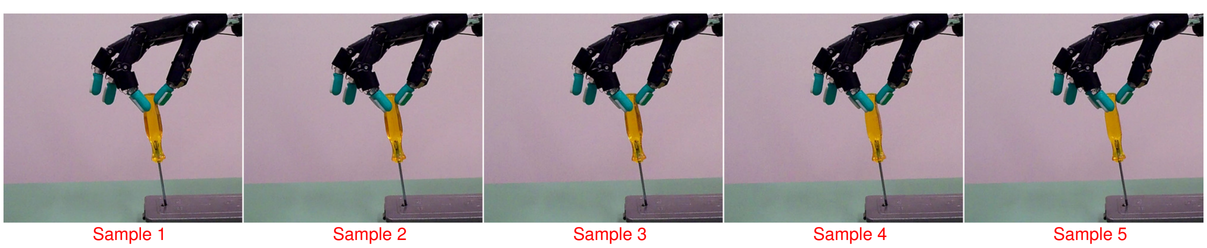
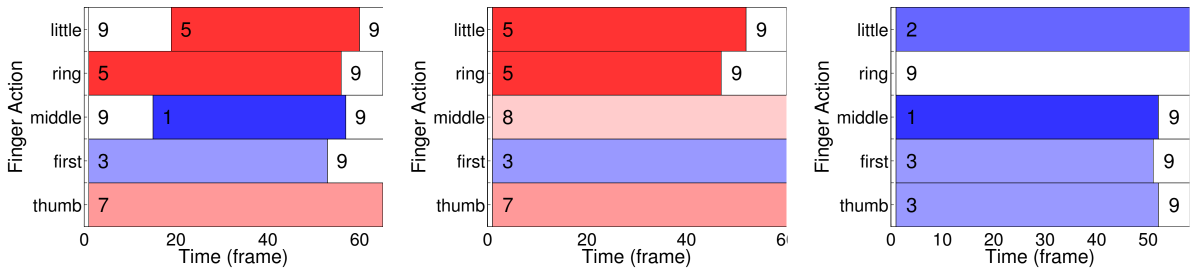


Figure 7.4: Screwdriver manipulation with a BioTac hand. The thumb, first and middle fingers participate the movement. By the finger positions and the screwdriver boundary color, we find the screwdriver is rotated.



(a) Triangle rotation. (b) Marker pen spin. (c) Screwing.

Figure 7.5: Action gist of the triangle rotation, the marker pen spin and the screwdriver manipulation. An interesting discovery is related to the screwdriver operation. For other anticlockwise rotations, the motion, especially the thumb, is usually “7” or in a warm color. However, the screwdriver has a reverse direction. We guess this is because the demonstrator gets used to anticlockwise rotate a screwdriver for installing a screw inside. In this case, he has to extend his finger to push the screwdriver ahead.

the screwdriver is not perfect: according to state gist, the thumb, first and middle fingers should contact the screwdriver, but the fact is the first finger sometimes does not touch the screwdriver.

We illustrate the temporal best solution in Fig 7.6 and Fig. 7.7. Since the first finger does not touch the screwdriver, we can find a place to test our simple human reward mechanism. However, because we cannot perfectly prove the converge of this method, we just list a partial result as Fig. 7.8; from the figure we can understand that the algorithm does learn as what we describe in Section 7.2. Besides, we should point out that if we do not give any feedback to the robot, the robot has to be babbling learn the parameters as previous chapter until the assign iterative loop ends.

7.4 Summary

This chapter proposes a interactive learning method based on simple human feedback. The major differences of this work compared with other works are: the proposed method is based on the boundary restriction by human evaluating “better” or “worse”; once a decision is made by human, the automatically calculated reward will have close relation with the previous reward.

Since there is no experimental comparison with other feedback learning methods, the performance of our method is going to be evaluated. Besides, whether controlling the searching boundary with human feedback will fall into local optimum, remains an open issue. Anyway, the creativity of this simple human reward algorithm is: it make use of the short-term-memory from human. In this case, each experimental trial is not isolated. When the human critics work with our proposed method, the only requirement is patience.

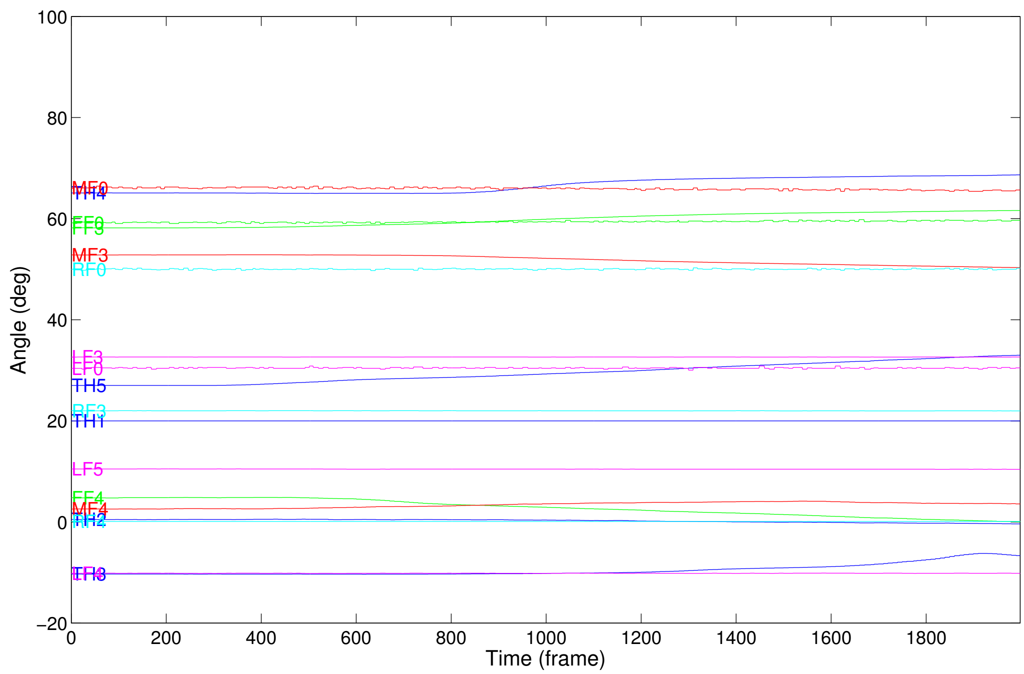


Figure 7.6: Joint angle variations of the screwdriver manipulation. We can see that the value variation is not sharp, this phenomenon corresponds with the video (Fig. 7.4).

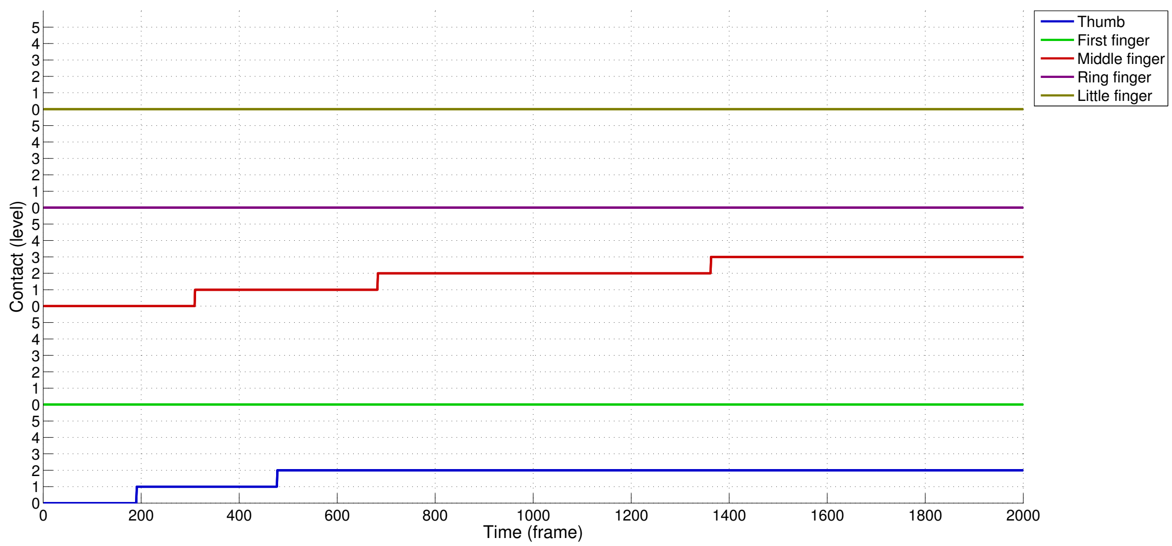


Figure 7.7: BioTac sensor sensing of the screwdriver manipulation. According to the original plan, we expect to see that the thumb, first and middle fingertips are keeping touched. However, the fact is first finger does not touch the screwdriver. Therefore, there is space for promoting the current solution. Moreover, we notice that the screwdriver can be easily rotated by the thumb and middle fingers. Concessively, when we discuss this issue based on the screwdriver functionality, this is not a serious problem.

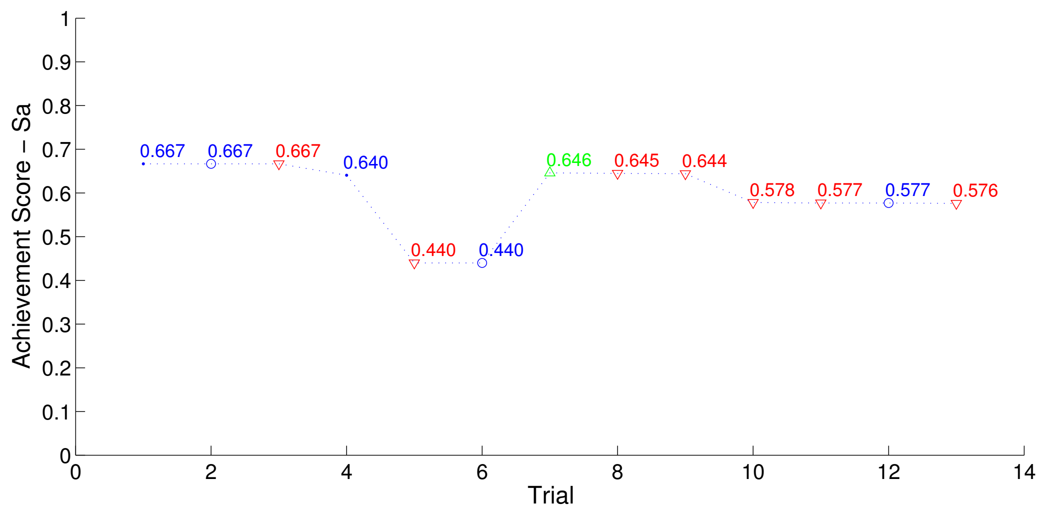


Figure 7.8: A segment of the achievement score variation for the screwdriver manipulation. Considering the parameter dimension (Eq. 6.2 in Chapter 6) and the action gist of screwdriver manipulation (Fig. 7.5(c)), we can have $d_{\mathbf{m}} = 14$. Since all parameters take part in the iteration, it is unnecessary to show the overall score. Instead, we can understand the simple human feedback by the achievement score. We mark that the green triangle (up) indicates the user give a "better" feedback. In this case, the reward must be higher than the previous reward; the red triangle (down) indicates the user give a "worse" feedback, and then the reward decreases; the blue circles appear when the user make an "equal" evaluation. In this segment, all rewards are lower than 0.7 because the first finger has not touched the screwdriver yet. When the iteration ends, the simple human reward method will return the best solution it finds.

Chapter 8

Conclusion and Future Work

Contents

8.1 Summary

We are the creatures in favor of questioning and answering, and in most cases we hope the questions and answers easy to follow. The typical form is somehow like "yes or no", because there are very limited options and the options are not overlapped. Driven by this "easy to follow" target, this thesis concentrates on proposing such kind of in-hand manipulation state-action gist as previous chapters.

The action gist consists of eight kinds of kinetic finger motions and one idle motion. The kinetic motions are differentiated by the directions of joint angle variation, i.e., "left or right", "forward or backward", and "close or open". For a 4 DOF finger, theoretically it should have $2^4 = 16$ kinds of motions. However, when we refer to the human hand anatomy, the distal interphalangeal joint and the proximal interphalangeal joint are coupled for motion execution. As a result, we use "close or open" to describe the variations of these two joints and finally have eight meta motion types.

For hand motion capture we employ a data-glove, which is a straightforward tool to access finger joint angles. Afterwards, for the extraction method we choose Gaussian Markov Random Field, because its structure well fits our expected model. As a result, we have action gist, as a kind new features, to describe in-hand movements. Rather than describing joint values, action gist is similar to describing "yes or no". Besides, it owns an advantage that value-level feature be difficult to achieve: when we are unsatisfied with an extracted result, we can move our own hand to prove the correctness. In this case, every person can teach the robot.

Because we prefer simple, we suggest to apply simple criteria on verifying in-hand manipulation movement. We match the criteria with action gist, so we call them "state gist"; consequently, the basic element of state gist is meta criterion. However, we aware a fact: a hand can be treated as an articulated object, but the concerned criteria are not only of one object. The criteria can be the object being manipulated, can be the object involved in the scenario, e.g., billiards, pen writing. Besides, we have the same motion, and we may receive different feedbacks as criteria, e.g., we press a button, we will see a light on, or a doorbell rings. Furthermore, there are many criteria that we have not considered in this short paragraph. Thus, we cannot summarize all kinds criteria in a countable pattern set, instead, for each kind of criteria we try to find a countable representation. In this thesis the considered criteria are: a fixed position, position varying directions along a specific axis, rotated angle variations along a specific axis, and binary contact states. These criteria are far from the rules govern this world, but we try to show the possibilities of employing countable patterns for presenting the criteria.

The state-action gist can be applied to in-hand manipulation behavior analysis, beside of that it can be used for instructing a humanoid hand to execute manipulation task. The scope of this thesis is **learning** from human demonstration, so we propose a reinforcement learning method based on the state-action gist. On on hand, a humanoid hand is controlled by sending joint angles, and the action gist indicates the joint varying direction of each finger. Consequently, the action gist limits the searching space at the beginning of the parameter exploration. On the other hand, the state gist provides us with criteria to check the achievement of the trials. Since criteria checking is an incremental process (each criterion should be validated one by one), we consider this point and modify the conventional parameter searching algorithms (e.g., the Particle Swarm Optimization and the Line Search with Re-Start in this thesis) to proceed manipulation learning. After modification, only the criteria-involved parameters update themselves, in this way we keep the other parameters safe.

For the sake of protecting the real robot hand, we perform self-learning in the simulation before the practice. However, when we turn to the real robot, we find that the learned control parameters may not work in the real case. Instead of manually correcting the parameters slightly, we obey our scope "**learning from human** demonstration" and propose a learning mechanism with human feedback. The feedback is as simple as the other related researches, i.e. with a positive/negative comment, but a difference is: the human comment is on the comparison between the current trial and the previous trial. Because we are easy to remember what happened in a couple minutes, the mechanism makes use of this point to update parameters.

In general, this thesis solves the in-hand manipulation learning based on state action gist. So far many researches deal with in-hand manipulation problems based on their previous works (e.g. grasping (Romero et al., 2009b, 2010), contact point schedule (Saut et al., 2006; Sahbani et al., 2007)), so they mainly discussed the stability of each grasp stage or how to plan the hand posture transition. However, this thesis directly points to in-hand manipulation and emphasizes hand motions with action gist. Moreover, it not only has more applications than (Romero et al., 2009b, 2010) but also has lower finger positioning requirement than (Saut et al., 2006; Sahbani et al., 2007). Furthermore, with respect to in-hand manipulation state-action model other researches are based on hand level (Gupta et al., 2009a; Faria et al., 2011b; Handle-project, 2011), but this thesis is based on finger level. Because of this point, we can proceed to analyze complicated finger-gaiting movements.

8.2 Limitations

This thesis works with every word of its title. However, when the author recalls the original intentions and summarize the main ideas in the previous section, he sigh with the time flying that there are still many gaps relevant to the topics of this thesis. Hereinafter, some significant flaws are pointed out as follows.

- This thesis discusses several typical in-hand manipulation tasks, but cover not all types. (Elliott and Connolly, 1984; Todorov and Ghahramani, 2004; Zheng et al., 2011; Bullock and Dollar, 2011) proposed several classifications for grasping or manipulation, but this thesis only proceed to the feasibilities of the proposed methods and skip this point.
- This thesis only considers single solid object manipulation. Current researches on the object manipulation gradually extend to various objects, such as articulated objects, and deformable objects (Khalil and Payeur, 2010). Theoretically, regarding to these challenging objects, the action gist is unnecessary for updating, but we should pay more attention on the state gist definition.
- This thesis does not cover experimental comparison with related work. However, current humanoid hands are actually all prototypes, and there are significant differences, which make comparisons very difficult. Even though this thesis is interested in proposing a state-action gist framework as a universal solution of all in-hand manipulation task, and many related work are of grasping but not of in-hand level; we can find some research ideas feasible to some kinds of in-hand manipulation. For example, (Sauser et al., 2012) concentrated on adjusting fingertip positions to maintain grasping force, their work can be applied to object translation in hand. This thesis also considers the experimental scenarios in (Sauser et al., 2012) and the scenario number is more than (Sauser et al., 2012), but it has not considered the stability as (Sauser et al., 2012). Therefore, one of our future work is to study the stability of the babbling learning results.

8.3 Future work

Beside of adding on more content with the current structure of this thesis, there are several new branches waiting for exploiting.

- More planning and faster learning. There are many techniques of manipulation planning introduced in Chapter 1 and Chapter 2 we have not tried yet, such as Inverse Kinematics, Dynamic Movement Primitives, or Policy Improvement using Path Integrals (Pastor et al., 2011; Kulvicius et al., 2012; Jetchev and Toussaint, 2013). However, these techniques usually require position information. So far the parameter learning is based on the joint level, which is the basic level to control the robot. Later when we involve our work with position information we should have better results. For example, with Inverse Kinematics we can calculate the fingertip trajectories so as to estimate whether collisions will happen, and then we can decide whether to protect the robot. Also, we can correspond to the scheduled contact to plan the finger trajectory.

- Sensing without disturbing natural manipulation movement. Usually we interact with objects with bare hands, but when we wear a data-glove or even more sensors it is difficult to perform the in-hand manipulation movements As a result we get unexpected data, there is no doubt that the extracted results are different from the ideal state-action gist. Therefore, we can use some external sensors, e.g. cameras, to track and estimate the interaction between the hand and the object (Song et al., 2013). Besides, we can install the sensor inside of the object, or customize an object for demonstration.
- Manipulation behavior recognition. We can extract the action gist from sensor data, and use Meta Motion Occurrence Histogram to statistically memorize the similar hand movement. The previous work such as (Ogawara et al., 2002; Handle-project, 2011) are relevant to this topics, but they are not interested with finger-level analysis. With action gist, as long as we have enough records with respect to different hand movements, it is possible to use some classification algorithms to identify or cluster the new meta motion sequences.
- Task-state-action research. So far we do not combine our application with any specific task but consider state-action gist as features. Actually a specific task should have its own state-action gist features. For examples, for specific tasks we should contact specific object areas (Baier, 2008; Falco et al., 2011; Aleotti and Caselli, 2012), for specific tasks we should receive specific manipulated results (Mason et al., 2012). On one hand we can try behavior recognition to identify the task, on the other hand we can construct a task-state-action network to find the connection between different tasks.

Appendix A

Contact state-action gist of the experiments

This appendix corresponds to Chapter 4. We take care of four scenarios in Chapter 4, they are: rectangular prism rotation, star prism rotation, water bottle rotation and screwing back and forth. Even though we use multiple sensors and record many trials, the raw data analysis is challenging and we get many unexpected results. For example, the tactile sensors are not sensitive if we do not firmly press our fingertips (the tactile sensing arrays) on the object. This fact makes us difficult to proceed: if we pay attention on pressing the object firmly, we lose the chance to correctly manipulate the object; if we pay attention on naturally performing the movements, we lose the expected contact information. Nevertheless, contact information is very important for robot manipulation. We tried to compromise between the natural movement and the firm contact, and finally list the nicest extractions among all experimental records we have. Before we looking at the figures of state action gist, we use Fig. A.1 to explain how to read the contact state action gist.

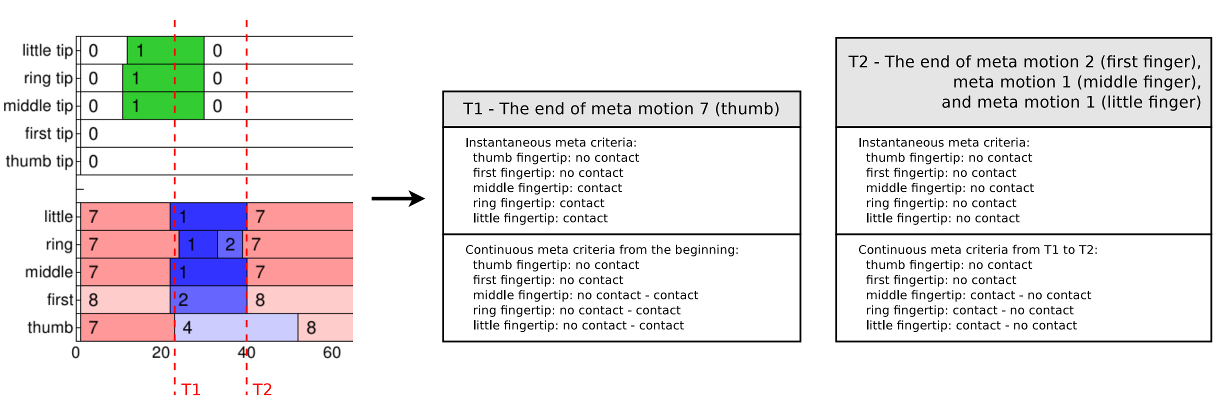


Figure A.1: How to read contact state action gist. We show two examples of reading the meta motions and meta criteria from the gist chart. Usually we read the criteria information at the boundary of two adjacent motions.

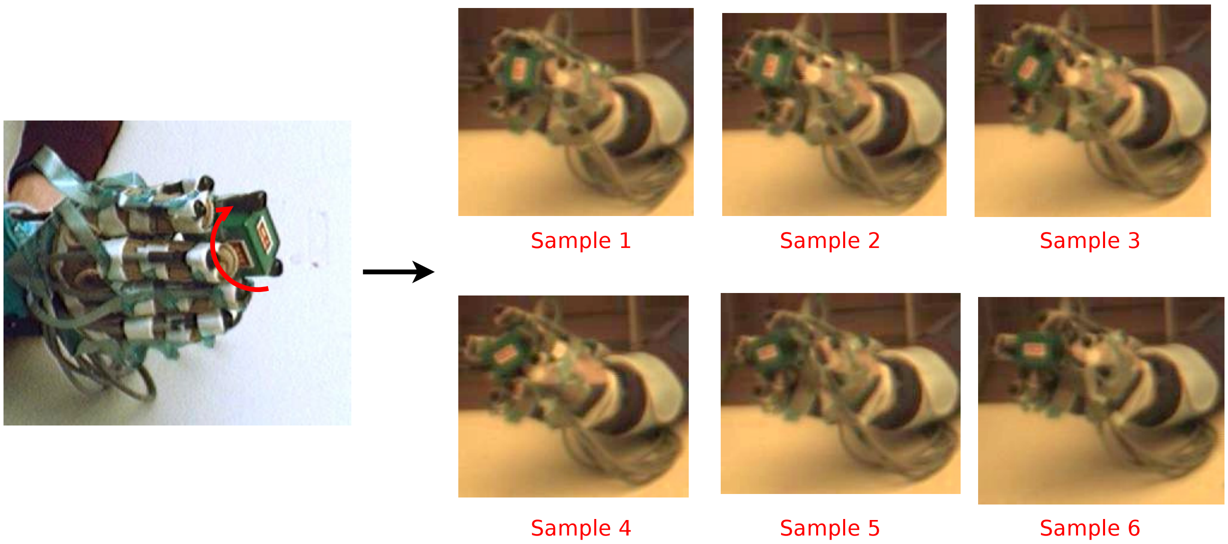


Figure A.2: Rectangular prism (cuboid) rotation. The thumb, first, middle and ring finger participate the manipulation. By releasing and contacting the surfaces of the rectangular prism, it is being span clockwise. The left photo is taken by a monocular camera, and other photos are taken by a stereo camera. Even though the resolution of the monocular camera is higher than the stereo camera, the monocular camera is on the lateral side. Thus, we have to take snapshots with the stereo camera.

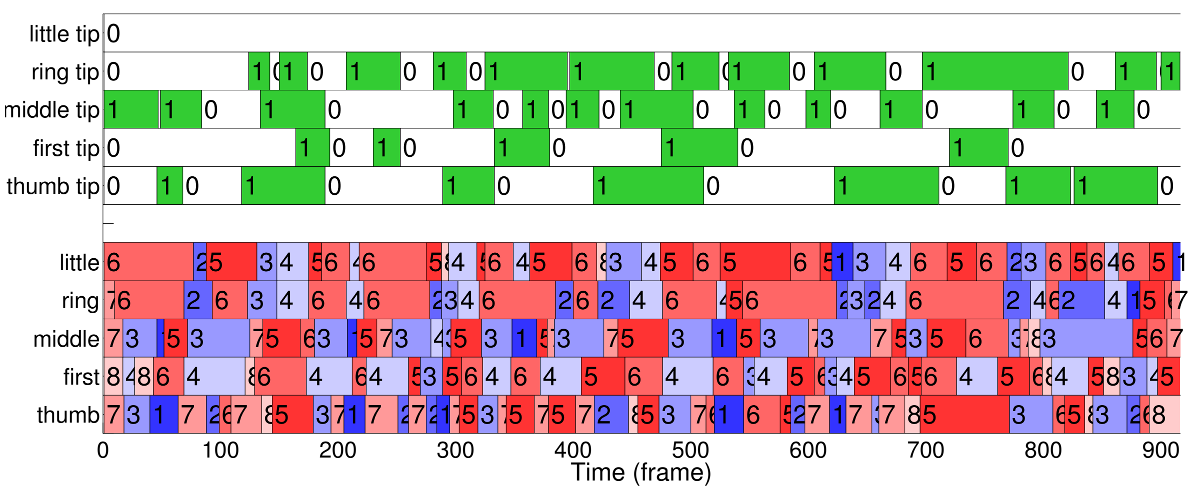


Figure A.3: State-action gist of rectangular prism rotation. By releasing and contacting the cuboid surfaces we rotate it gradually. We repeat the releasing and contacting movements for ten times, but the extraction does not directly reflects this fact. By counting the meta motion 6 of the ring finger, we find that the motion repeats 11 times (including the very short slot after the 900th frame). This fact inspires us that we can design an algorithm to segment this periodic movement, and then we can take deeper analysis. Therefore, Chapter 5 involves how to do the segmentation.

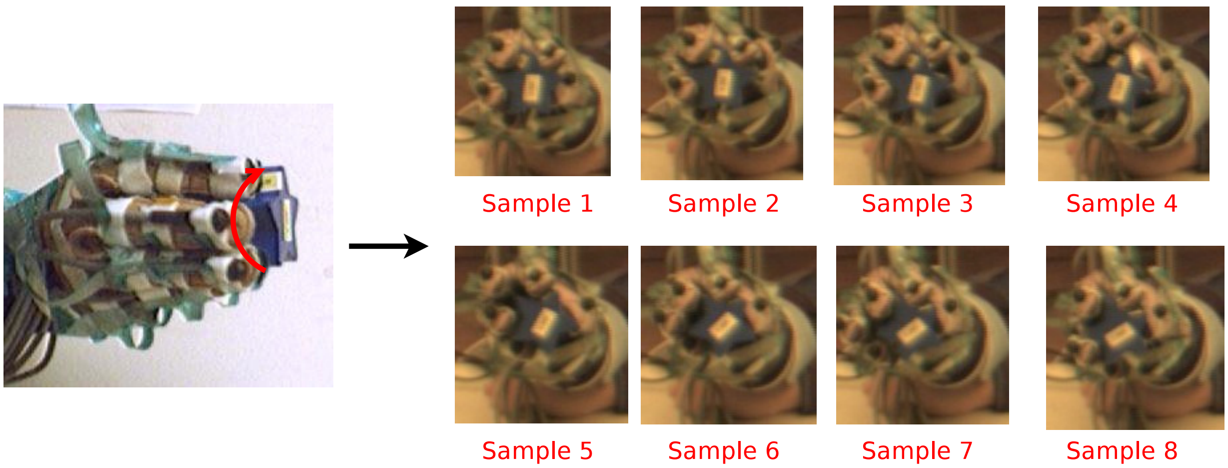


Figure A.4: Star prism rotation. The thumb, first, middle and ring finger participate the manipulation. By releasing and contacting the indents of the star prism, the star is being span clockwise. Since the photos from the stereo camera are too blur to find the finger, we suggest to pay attention to the black dots around the blue star prism. In fact the black dots are Polhemus sensors attached to the fingertips. Consequently, when we figure out the black dots moving direction, we will understand how the fingers move.

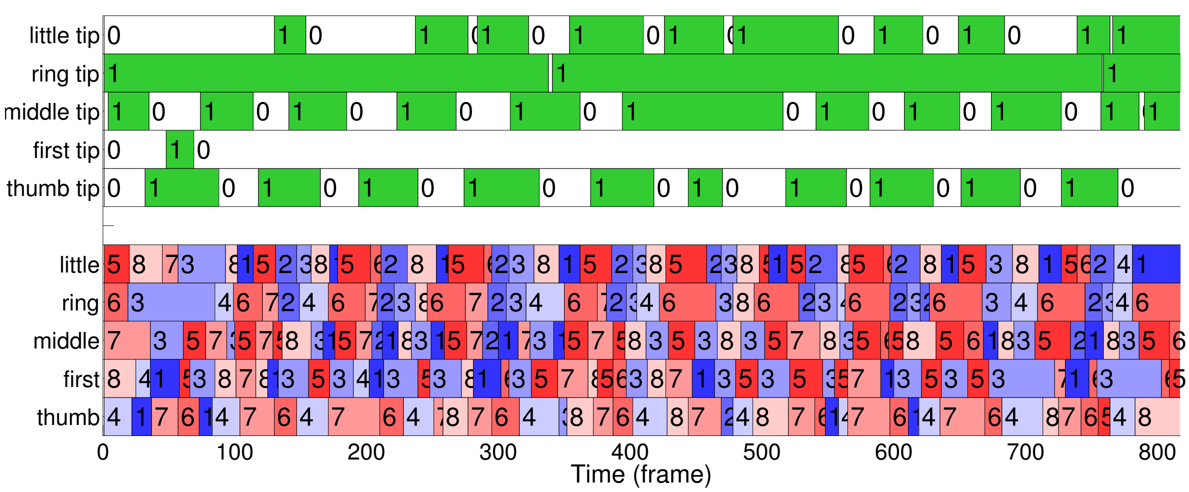


Figure A.5: State-action gist of star prism rotation. The demonstrator uses the thumb, first, middle and ring fingers performing finger gaiting. In this experiment, we touch and release the prism indents for ten times. We can find that the thumb also contact for ten times, as well as the meta motion 7 on the thumb (the exact number of meta motion 7 is 11, but two "7" are very close before the 300th frame). Another fact of this extraction is: the contact of the first fingertip is insensitive.

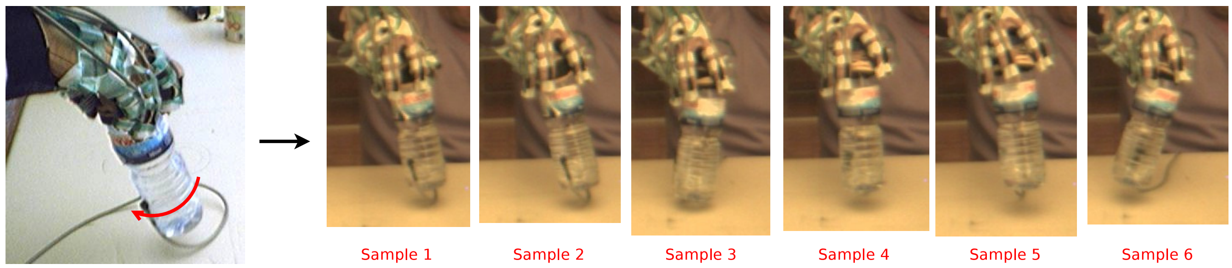


Figure A.6: Snapshots of bottle rotation. The thumb, first, middle and ring finger participate the manipulation. By releasing and contacting the upper part of the bottle, the bottle is gradually rotated clockwise for many times. Because the bottle never touches the table, the trajectory of the bottle center is not similar to a circle.

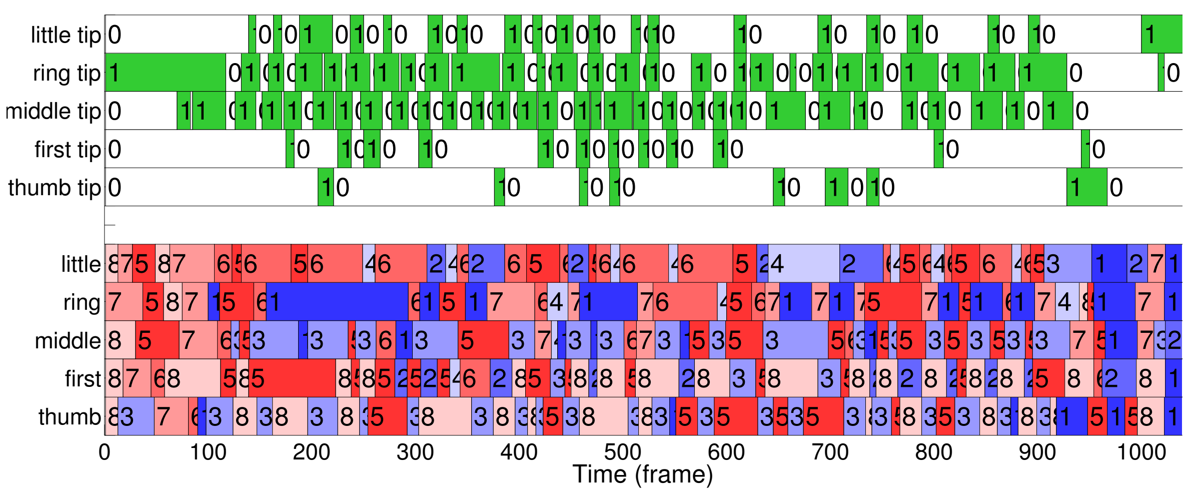


Figure A.7: State-action gist of bottle rotation. We can find many repeated meta motions on each finger. However, the contact states do not match the repeated motions. Here we note a phenomenon which also happens in Fig. A.3, Fig. A.5, and Fig. A.9: Even tough the object looks being touched in the visual sensors, the contact states are all 0 (no contact). In this case the object should fall down on the table, e.g., around the 600$^{\text{th}}$ frame there are "all contacts are zero" frames. Based on the fact that the hand keeps contacts with the object, we find the tactile sensors are insensitive.

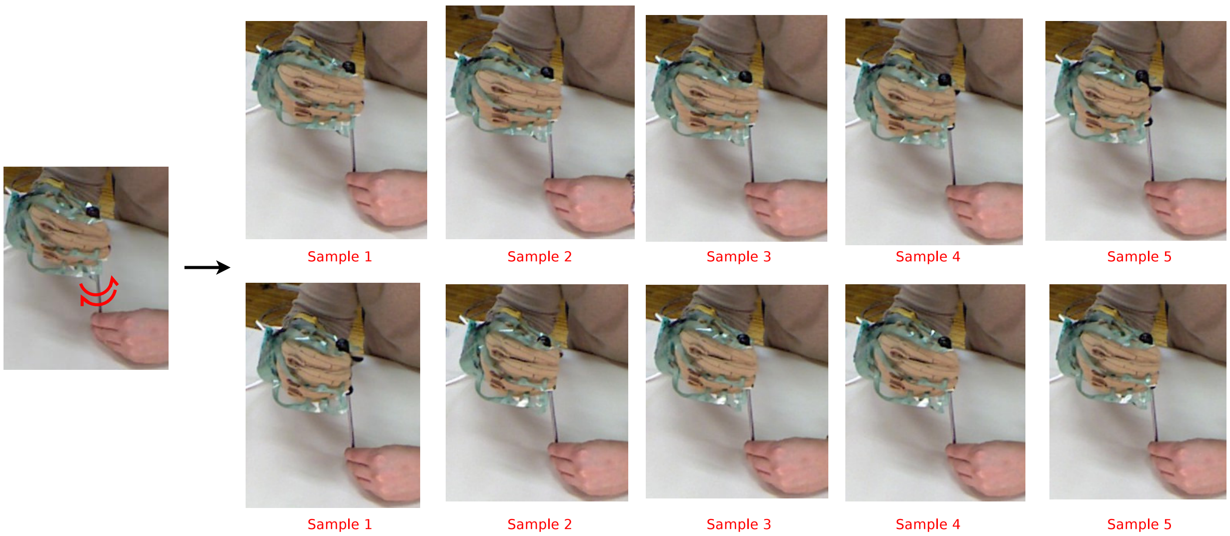


Figure A.8: Movement of screwing. By rubbing the screwdriver with fingers, the screwdriver is span back and forth. From this figure, we know it moves clockwise and then anticlockwise.

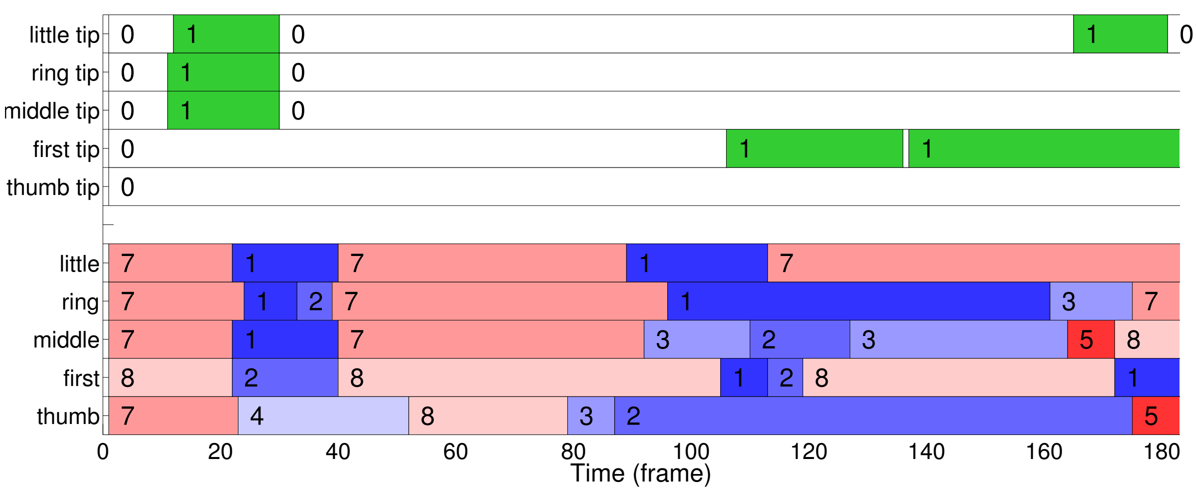


Figure A.9: State-action gist of screwing. We notice that there is no contact information on the thumb, the issue is caused by the contact region between the hand and the screwdriver, i.e., the lateral side of the thumb touches the screwdriver. Even though the contact information is not sensitive enough, we still find a fact: from the tactile information we can infer that when the screwdriver is rotated clockwise, the middle finger is more tight on thee screwdriver; and when the rotation is anticlockwise, the first finger is more tight.

Appendix B

Action gists of ladle manipulation

This appendix corresponds to Chapter 5. We manipulate the ladle in the way shown in Fig. B.1. A participant demonstrates it for nine times. After the extraction from the raw data-glove values, we have the action gists as Fig. B.2. From the illustrated action gists we can see that the ladle is manipulated in similar ways.

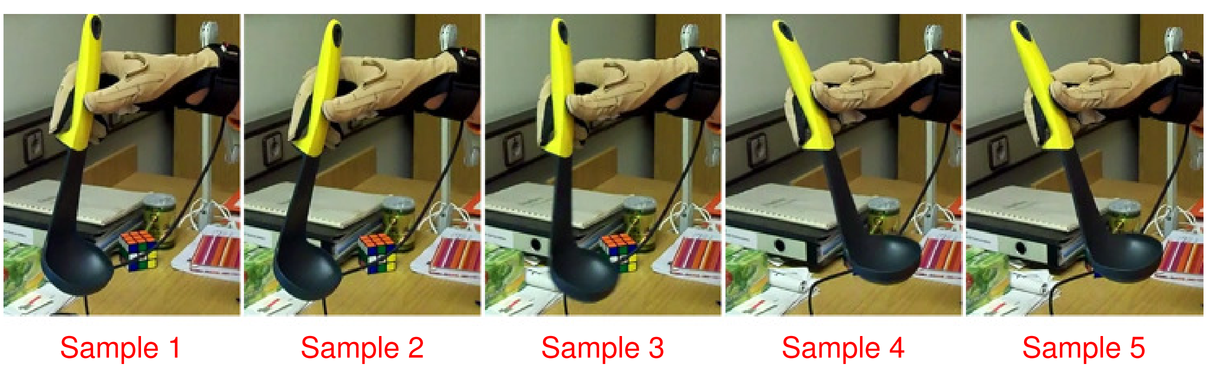
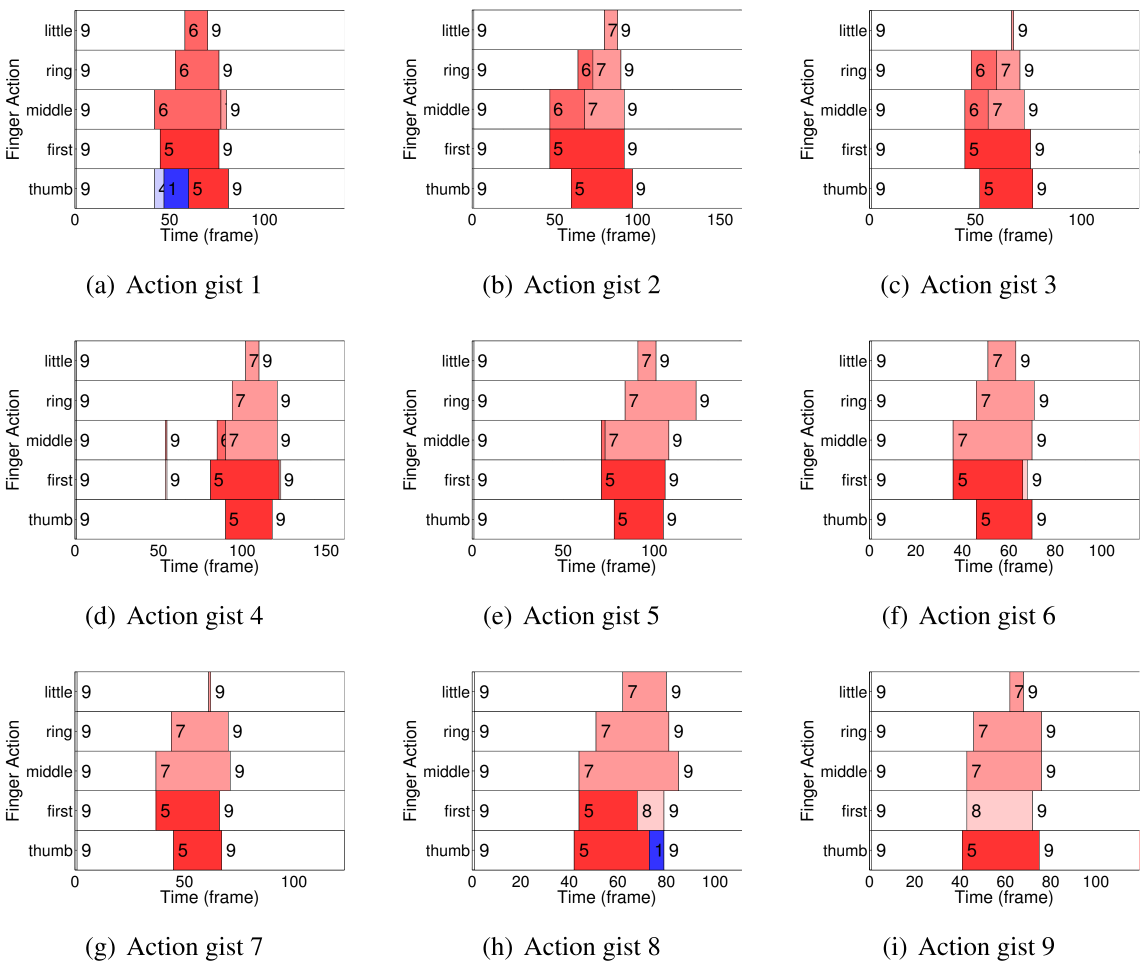


Figure B.1: Snapshots of the ladle manipulation. The ladle manipulation is involved the thumb, first and middle fingers. We note that there are countless ways to manipulate the ladle. However, when we assign the task to a specific person, we always get similar operation sequences.



(a) Action gist 1 (b) Action gist 2 (c) Action gist 3

(d) Action gist 4 (e) Action gist 5 (f) Action gist 6

(g) Action gist 7 (h) Action gist 8 (i) Action gist 9

Figure B.2: Action gists of ladle manipulation. The thumb, first and middle finger are used to spoon the ladle up.

Bibliography

Abbott, Robert G and Williams, Lance R. Multiple target tracking with lazy background subtraction and connected components analysis. *Machine Vision and Applications*, 20(2):93–101, January 2009. (page 19).

Agrawal, Motilal; Konolige, Kurt, and Bolles, Robert C. Localization and mapping for autonomous navigation in outdoor terrains: A stereo vision approach. In *IEEE Workshop in Applied Computer Vision*, 2007. (page 19).

Al-Shaher, Abdullah A and Hancock, Edwin R. Articulated shape mixtures for object recognition. In *The British Machine Vision Conference*, pages 1–10, 2004. (page 20).

Aleotti, Jacopo and Caselli, Stefano. Learning manipulation tasks from human demonstration and 3D shape segmentation. *Advanced Robotics*, 26(16):1863–1884, 2012. (page 126).

Ansuini, Caterina; Santello, Marco; Massaccesi, Stefano, and Castiello, Umberto. Effects of end-goal on hand shaping. *Journal of Neurophysiology*, 95(4):2456–2465, April 2006. (page 10).

Argall, Brenna D.; Chernova, Sonia; Veloso, Manuela, and Browning, Brett. A survey of robot learning from demonstration. *Robot. Auton. Syst.*, 57(5):469–483, 2009. ISSN 0921-8890. (pages 24, 85).

Arimoto, Suguru; Yoshida, Morio; Sekimoto, Masahiro, and Bae, Ji-Hun. A Riemannian-geometric approach for intelligent control and fingertip dsign of multi-fingered hands. *Advanced Robotics*, 24(8):1345–1364, 2010. (page 7).

Aronov, Dmitriy; Andalman, Aaron S, and Fee, Michale S. A specialized forebrain circuit for vocal babbling in the juvenile songbird. *Science*, 320(5876):630–634, May 2008. (page 9).

Atzori, M.; Gijsberts, A.; Heynen, S.; Hager, A.M.; Deriaz, O.; van der Smagt, P.; Castellini, C.; Caputo, B., and Muller, H. Building the Ninapro database: A resource for the biorobotics community. In *IEEE RAS EMBS International Conference on Biomedical Robotics and Biomechatronics*, pages 1258–1265, 2012. (page 23).

Atzori, Manfredo; Baechler, Micheal, and Müller, Henning. Recognition of Hand Movements in a Trans–Radial Amputated Subject by sEMG. In *International Conference On Rehabilitation Robotics*, 2013. (page 23).

Bae, Ji-Hun; Arimoto, Suguru; Yamamoto, Yuuichi; Hashiguchi, Hiroe, and Sekimoto, Masahiro. Reaching to grasp and preshaping of multi-DOFs robotic hand-arm systems using approximate configuration of objects. In *IEEE/RSJ International Conference on Intelligent*

Robots and Systems, pages 1605–1610, 2006. (page 8).

Bai, Xue and Sapiro, Guillermo. Geodesic matting: A framework for fast interactive image and video segmentation and matting. *International Journal of Computer Vision*, 82(2):113–132, 2009. (page 18).

Baier, Tim. *Lernen der Handhabung von Alltagsgegenständen im Kontext eines Service-Roboters*. PhD thesis, Informatik, University of Hamburg, 2008. (pages 7, 25, 126).

Baier, Tim and Zhang, Jianwei. Reusability-based semantics for grasp evaluation in context of service robotics. In *IEEE International Conference on Robotics and Biometics*, pages 703–708, 2006. (page 7).

Ballester, Gabriel Recatala. *Visual determination, tracking and execution of 2D grasps using a behavior-inspired approach*. PhD thesis, The Robotic Intelligence Laboratory, Universitat Jaume I, 2003. (page 18).

Barbič, Jernej; Safonova, Alla; Pan, Jia-Yu; Faloutsos, Christos; Hodgins, Jessica K., and Pollard, Nancy S. Segmenting motion capture data into distinct behaviors. In *Proceedings of Graphics Interface*, pages 185–194, 2004. (page 69).

Barrett, Inc. Barrett hand website. http://www.barrett.com/robot/products-hand.htm, Retrieved June 20, 2013, 2013. (page 4).

Baxter, Paul and Browne, Will. Memory-based cognitive framework: A low-level association approach to cognitive architectures. In Kampis, George; Karsai, István, and Szathmáry, Eörs, editors, *Advances in Artificial Life. Darwin Meets von Neumann*, volume 5777 of *Lecture Notes in Computer Science*, pages 402–409. Springer Berlin / Heidelberg, 2011. (page 86).

Bell, Matthew and Balkcom, Devin. Grasping non-stretchable cloth polygons. *International Journal of Robotics Research*, 29(6):775–784, 2010. (page 7).

Berenson, Dmitry; Kuffner, James, and Choset, Howie. An optimization approach to planning for mobile manipulation. In *IEEE International Conference on Robotics and Automation*, pages 1187–1192, 2008. (page 8).

Biagiotti, L; Lotti, F; Melchiorri, C, and Vassura, G. Mechatronic design of innovative fingers for anthropomorphic robot hands. In *IEEE International Conference on Robotics and Automation*, pages 3187–3192, 2003. (page 2).

Billing, Erik and Hellström, Thomas. A formalism for learning from demonstration. *Paladyn. Journal of Behavioral Robotics*, 1:1–13, 2010. (page 86).

Billing, Erik; Hellström, Thomas, and Janlert, Lars-Erik. Behavior recognition for Learning from Demonstration. In *IEEE International Conference on Robotics and Automation*, pages 866–872, 2010. (page 86).

Bitzer, Sebastian and Smagt, Patrick van der. Learning EMG control of a robotic hand: Towards active prostheses. In *IEEE International Conference on Robotics and Automation*, pages 2819–2823, 2006. (pages 23, 24).

Bleyer, Michael; Rother, Carsten, and Kohli, Pushmeet. Surface stereo with soft segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1570–1577, 2010. (page 19).

Bodiroza, S.; Schillaci, G., and Hafner, V.V. Robot ego-sphere: An approach for saliency detection and attention manipulation in humanoid robots for intuitive interaction. In *IEEE-RAS International Conference on Humanoid Robots*, pages 689–694, 2011. (page 86).

Bodor, Robert; Drenner, Andrew, and Schrater, Paul. Optimal camera placement for automated surveillance tasks. *Journal of Intelligent Robot System*, 50(3):257–295, November 2007. (page 19).

Bogdanova, Iva; Bur, Alexandre, and Huegli, Heinz. The spherical approach to omnidirectional visual attention. In *European Signal Processing Conference*, 2008. (page 18).

Bojilov, Ljudmil V; Alexiev, Kiril M, and Konstantinova, Pavlina D. An accelerated IMM JPDA algorithm for tracking multiple manoeuvring targets in clutter. In *International Conference on Numerical Methods and Applications*, pages 274–282, 2002. (page 19).

Borst, Cb; Fischer, M, and Hirzinger, G. Grasp planning: how to choose a suitable task wrench space. In *IEEE International Conference on Robotics and Automation*, pages 319–325, 2004. (page 7).

Bray, M; Koller-Meier, E, and Van Gool, L. Smart particle filtering for high-dimensional tracking. *Computer Vison and Image Understanding*, 106(1):116–129, 2007. (page 20).

Bray, Matthieu; Koller-Meier, Esther, and Van Gool, Luc. Smart particle filtering for 3D hand tracking. In *IEEE International Conference on Automatic Face and Gesture Recognition*, pages 675–680, 2004. (page 20).

Bujnak, Martin; Kukelova, Zuzana, and Pajdla, Tomas. A general solution to the P4P problem for camera with unknown focal length. In *Computer Vision and Pattern Recognition*, pages 1–8, 2008. (page 20).

Bullock, Ian M and Dollar, Aaron M. Classifying human manipulation behavior. In *IEEE International Conference on Rehabilitation Robotics*, pages 1–6, 2011. (pages 11, 14, 125).

Bur, A; Wurtz, P; Mueri, R M, and Huegli, H. Dynamic visual attention: competitive versus motion priority scheme. In *Proceedings of the ICVS Workshop on Computational Attention and Applications*, 2007. (page 18).

Caligiore, D.; Ferrauto, T.; Parisi, D.; Accornero, N.; Capozza, M., and Baldassarre, G. Using motor babbling and hebb rules for modeling the development of reaching with obstacles and grasping. In *International Conference on Cognitive Systems*, pages 1–8, 2008. (page 86).

Calinon, Sylvain and Billard, Aude. Incremental learning of gestures by imitation in a humanoid robot. In *ACM/IEEE International Conference on Human-robot Interaction*, pages 255–262, 2007. (page 24).

Catalano, M. G.; Grioli, G.; Serio, A.; Farnioli, E.; Piazza, C., and Bicchi, A. Adaptive Synergies for a Humanoid Robot Hand. In *IEEE-RAS International Conference on Humanoid Robots*, pages 7–14, 2012. (page 7).

Chang, Lillian Y. *Pre-grasp interaction as a manipulation strategy for movable objects*. PhD thesis, The Robotics Institute, Carnegie Mellon University, 2010. (pages 17, 24).

Chang, Lillian Y; Zeglin, Garth J, and Pollard, Nancy S. Preparatory object rotation as a human-inspired grasping strategy. In *IEEE-RAS International Conference on Humanoid Robots*, pages

527–534, 2008. (page 24).

Chang, Lillian Y; Klatzky, Roberta L, and Pollard, Nancy S. Selection criteria for preparatory object rotation in manual lifting actions. *Journal of Motor Behavior*, 42(1):11–27, January 2010. (page 24).

Chen, Shengyong; Li, Y F, and Zhang, Jianwei. Vision processing for realtime 3-D data acquisition based on coded structured light. *IEEE Transactions on image processing*, 17(2):167–176, February 2008. (page 18).

Chen, W J and Xie, M. On the design of a novel dexterous hand. In *International Conference on Advanced Robotics*, pages 61–65, 1999. (page 2).

Cheng, Gang. Several videos related to in-hand manipulation learning from human demonstration. http://tams.informatik.uni-hamburg.de/people/cheng/index.php?content=videos, Retrieved June 20, 2013, 2013. (page 100).

Cheng, Gang; Hendrich, Norman, and Zhang, Jianwei. Action gist based automatic segmentation for periodic in-hand manipulation movement learning. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4768–4775, 2012. (page 61).

Chenouard, Nicolas; Bloch, Isabelle, and Olivo-Marin, Jean-Christophe. Multiple hypothesis tracking in cluttered condition. In *IEEE International Conference on Image Processing*, pages 3621–3624, 2009. (page 19).

Chik, Desmond; Trumpf, Jochen, and Schraudolph, Nicol N. Using an adaptive VAR model for motion prediction in 3D hand tracking. In *IEEE Conference on Automatic Face and Gesture Recognition*, pages 1–8, 2008. (page 20).

Chou, Te-Shun; Gadd, Ashley, and Knott, Dave. Hand-Eye: A vision-based approach to data glove calibration. In *Human Interface Technologies*, pages 47–54, 2008. (page 22).

Ciancio, A.L.; Zollo, L.; Guglielmelli, E.; Caligiore, D., and Baldassarre, G. Hierarchical reinforcement learning and central pattern generators for modeling the development of rhythmic manipulation skills. In *IEEE International Conference on Development and Learning*, pages 1–8, 2011. (page 86).

Citecbielefeld, Youtube Video. Ball swapping. http://www.youtube.com/watch?v=2eAWswrWJ9I, Retrieved June 20, 2013, 2010. (page 6).

Civicioglu, Pinar and Besdok, Erkan. A conceptual comparison of the Cuckoo-search, particle swarm optimization, differential evolution and artificial bee colony algorithms. *Artificial Intelligence Review*, pages 1–32, 2011. (page 85).

Clerc, M. and Kennedy, J. The particle swarm - explosion, stability, and convergence in a multidimensional complex space. *IEEE Transactions on Evolutionary Computation*, 6(1):58–73, 2002. (page 98).

Cobos, Salvador; Ferre, Manuel; Ángel Sánchez-Urán, M.; Ortego, Javier, and Aracil, Rafael. Human hand descriptions and gesture recognition for object manipulation. *Computer Methods in Biomechanics and Biomedical Engineering*, 13(3):305–317, 2010. (page 68).

Corcoran, Craig and Platt, Robert Jr. A measurement model for tracking hand-object state during dexterous manipulation. In *IEEE International Conference on Robotics and Automation*, pages

4302–4308, May 2010. (page 26).

Crast, Jessica; Fragaszy, Dorothy; Hayashi, Misato, and Matsuzawa, Tetsuro. Dynamic in-hand movements in adult and young juvenile chimpanzees (Pan troglodytes). *American Journal of Physical Anthropology*, 138(3):274–285, March 2009. (page 2).

Cui, Jinshi and Sun, Zengqi. Visual hand motion capture for guiding a dexterous hand. In *IEEE International Conference on Automatic Face and Gesture Recognition*, pages 729–734, 2004. (page 21).

CybergloveSystems, Inc. Cyberglove website. http://www.cyberglovesystems.com/index.php, Retrieved June 20, 2013, 2013. (page 22).

Darlinski, G.; Bottger, U.; Waser, R.; Klauk, H.; Halik, M.; Zschieschang, U.; Schmid, G., and Dehm, C. Mechanical force sensors using organic thin-film transistors. *J. Appl. Phys*, 97(9): 093708–093708–4, May 2005. (page 24).

DARPA, . Autonomous Robotic Manipulation Software Track (ARM-S) Broad Agency Announcement. Technical report, DARPA, 2010. (page 17).

Davis, James and Shah, Mubarak. Determining 3-D hand motion. In *IEEE Asilomar Conference on Signals, Systems and Computers*, pages 1262–1266, 1994. (page 20).

Deisenroth, Marc Peter; Rasmussen, Carl Edward, and Fox, Dieter. Learning to Control a Low-Cost Manipulator using Data-Efficient Reinforcement Learning. In *Robotics: Science and Systems*, 2011. (page 113).

Delft, Robot. Delft hand website. http://robotics.tudelft.nl/, Retrieved June 20, 2013, 2013. (page 4).

DeMenthon, Daniel F and Davis, Larry S. Model-Based Object Pose in 25 Lines of Code. *International Journal of Computer Vision*, 15(1-2):123–141, 1995. (page 20).

Demiris, Yiannis and Dearden, Anthony. From motor babbling to hierarchical learning by imitation: A robot developmental pathway. In *International Workshop on Epigenetic Robots*, pages 31–37, 2005. (page 86).

Demiris, Yiannis and Meltzoff, Andrew. The robot in the crib: A developmental analysis of imitation skills in infants and robots. *Infant Child Development*, 17(1):43–53, 2008. (page 86).

Dimuro, G P and Costa, A C R. Hand gesture recognition in an interval fuzzy approach. *Sociedad Brasilera TEMA Tend Matematica Aplicada e Computacional*, 8(1):21–31, 2007. (page 22).

Dingwell, Jonathan B; Mah, Christopher D, and Mussa-Ivaldi, Ferdinando A. Manipulating objects with internal degrees of freedom: evidence for model-based control. *Journal of Neurophysiology*, 88(1):222–235, 2001. (page 9).

Do, Martin; Romero, Javier; Kjellstroem, Hedvig; Azad, Pedram; Asfour, Tamim; Kragic, Danica, and Dillmann, Ruediger. Grasp recognition and mapping on humanoid robots. In *IEEE-RAS International Conference on Humanoid Robots*, pages 465–471, 2009. (page 19).

Do, Quoc and Jain, Lakhmi. A biological inspired visual landmark recognition architecture. In *Digital Image Computing: Techniques and Applications*, pages 325–330, 2009. (page 19).

Donoser, Michael and Bischof, Horst. Real time appearance based hand tracking. In *International Conference on Pattern Recognition*, pages 1–4, 2008. (page 19).

Dorner, Brigitte. Chasing the colour glove: Visual hand tracking. Master's thesis, School of Computing Science, 1994. (page 20).

Du, Guanglong; Zhang, Ping; Mai, Jianhua, and Li, Zeling. Markerless Kinect-based hand tracking for robot teleoperation. *International Journal of Advanced Robotic Systems*, 9(36): 1–10, 2012. (page 22).

Duca, Florin; Fredriksson, Jonas, and Fjeld, Morten. Real-time 3D hand interaction: Single webcam low-cost approach. In *IEEE VR Workshop on Trends and issues in Tracking for Virtual Envrionments*, 2007. (page 19).

Ekvall, Staffan and Kragic, Danica. Grasp recognition for programming by demonstration. In *IEEE International Conference on Robotics and Automation*, pages 748–753, 2005a. (pages 22, 26).

Ekvall, Staffan and Kragic, Danica. Integrating object and grasp recognition for dynamic scene interpretation. In *International Conference on Advanced Robotics*, pages 331–336, 2005b. (page 19).

Ekvall, Staffan and Kragic, Danica. Robot learning from demonstration: A task-level planning approach. *International Journal of Advanced Robotic Systems*, 5(3):223–234, 2008. (page 26).

Elliott, J.M. and Connolly, K. A classification of manipulative hand movements. *Developmental Medicine and Child Neurology*, 26(3), June 1984. (pages 11, 125).

Falco, P.; Jakel, R.; Natale, C., and Dillmann, R. Improvement of human hand motion observation by exploiting contact force measurements. In *IEEE-RAS International Conference on Humanoid Robots*, pages 141–146, 2011. (page 126).

Faria, Diego; Martins, Ricardo; Lobo, Jorge, and Dias, Jorge. Manipulative tasks identification by learning and generalizing hand motions. In Camarinha-Matos, Luis, editor, *Technological Innovation for Sustainability*, volume 349 of *IFIP Advances in Information and Communication Technology*, pages 173–180. Springer Boston, 2011a. ISBN 978-3-642-19169-5. (pages 68, 76).

Faria, Diego R and Dias, Jorge. 3D hand trajectory segmentation by curvatures and hand Orientation for classification through a probabilistic approach. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1284–1289, 2009. (page 27).

Faria, Diego R; Martins, Ricardo; Lobo, Jorge, and Dias, Jorge. Manipulative tasks identification by learning and generalizing hand motions. In *International Federation for Information Processing*, pages 173–180, 2011b. (pages 26, 124).

Faria, Diego R.; Martins, Ricardo; Lobo, Jorge, and Dias, Jorge. Extracting data from human manipulation of objects towards improving autonomous robotic grasping. *Robotics and Autonomous Systems*, 60(3):396–410, 2012. (pages 69, 76).

Faugere, Jean-Charles; Moroz, Guillaume; Rouillier, Fabrice, and Din, Mohab Safey EI. Classification of the perspective-three-point problem, discriminant variety and real solving polynomial systems of inequalities. In *International Symposium on Symbolic and Algebraic Computation*, pages 79–86, 2008. (page 20).

Fillbrandt, Holger; Akyol, Suat, and Kraiss, Karl Friedrich. Extraction of 3D hand shape and posture from image sequences for sign language recognition. In *IEEE International Workshop on Analysis and Modeling of Faces and Gesture*, 2003. (page 26).

Fiore, Loren; Fehr, Duc; Bodor, Robot; Drenner, Andrew; Somasundaram, Guruprasad, and Papanikolopoulos, Nikolaos. Multi-camera human activity monitoring. *Journal of Intelligent Robot Systems*, 52(1):5–43, May 2008. (page 19).

Fishel, Jeremy; Lin, Gary, and Loeb, Gerald. *BioTac Product Manual V15*. SynTouch LLC, 2222 S.Figueroa PH2, Los Angeles, CA 90007, 2012. (page 88).

Flanagan, J Randall; Bowman, Miles C, and Johansson, Roland S. Control strategies in object manipulation tasks. *Current Opinion in Neurobiology*, 16(6):1–10, December 2006. (page 25).

Foix, Sergi; Alenya, Guillem; Andrade-Cetto, Juan, and Torras, Carme. Object modeling using a ToF camera under an uncertainty reduction approach. In *IEEE International Conference on Robotics and Automation*, pages 1306–1312, 2010. (page 22).

Folgheraiter, Michele; Baragiola, Ilario, and Gini, Giuseppina. *Teaching grasping to a humanoid hand as a generalization of human grasping data*, chapter Knowledge Exploration in Life Science Informatics, pages 139–150. Springer Verlag, November 2004. (pages 22, 25).

Fu, Jiaxin L and Pollard, Nancy S. On the importance of asymmetries in grasp quality metrics for tendon driven hands. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1068–1075, 2006. (page 7).

Fuentes, Olac and Nelson, Randal. Learning dextrous manipulation skills for multifingered robot hands using the evolution strategy. *Machine Learning*, 31(1-3):223–237, 1998. (pages 4, 26).

Fuentes, Olac and Nelson, Randal C. Experiments on Dextrous manipulation without prior object models. In *IEEE International Symposium on Intelligent Control*, pages 542–547, 1996. (page 4).

Furukawa, Noriatsu; Namiki, Akio; Taku, Senoo, and Ishikawa, Masatoshi. Dynamic regrasping using a high-speed multifingered hand and a high-speed vision system. In *IEEE International Conference on Robotics and Automation*, pages 181–187, 2006. (page 7).

Gams, A. and Ude, A. Generalization of example movements with dynamic systems. In *IEEE-RAS International Conference on Humanoid Robots*, pages 28–33, 2009. (page 31).

Gao, X H; Jin, M H; ; Jiang, L; Xie, Z W; He, P; Yang, L; Liu, Y W; Wei, R; Cai, H G; Liu, H; Butterfass, J; Grebenstein, M; Seitz, N, and Hirzinger, G. The HIT/DLR dexterous hand: Work in progress. In *IEEE International Conference on Robotics and Automation*, pages 3164–3168, 2003a. (page 83).

Gao, X H; Jin, M H; Jiang, L; Xie, Z W; He, P; Yang, L; Liu, Y W; Wei, R; Cai, H G; Liu, H; Butterfass, J; Grebenstein, M; Seitz, N, and Hirzinger, G. The HIT/DLR dexterous hand: Work in progress. In *International Conference on Robotics and Automation*, pages 3164–3168, 2003b. (page 4).

Gao, Xiao-Shan and Tang, Jianliang. On the probability of the number of solutions for the P4P problem. *Journal of Mathematical Imaging and Vision*, 25(1):79–86, July 2006. (page 20).

Garner, Aleena R; Rowland, David C; Hwang, Sang Youl; Baumgaertel, Karsten; Roth, Bryan L; Kentros, Cliff, and Mayford, Mark. Generation of a Synthetic Memory Trace. *Science*, 335 (6075):1513–1516, March 2012. (page 9).

Gavrila, D M. The visual analysis of human movement: A survey. *Computer Vision and Image Understanding*, 73(1):82–98, January 1999. (page 18).

Ghallab, Malik. An Overview of Planning Technology in Robotics. In *KI*, pages 29–49, 2004. (page 8).

Gielniak, M.J.; Liu, C.K., and Thomaz, A.L. Stylized motion generalization through adaptation of velocity profiles. In *IEEE International Symposium on Robots and Human Interactive Communications*, pages 304–309, Sept. 2010. (page 31).

Goldstein, Michael H and Schwade, Jennifer A. Social feedback to infants' babbling facilitates rapid phonological learning. *Psychological Science*, 19(5):515–523, 2008. (page 9).

Goldwasser, Dan and Roth, Dan. Learning from natural instructions. In *International joint conference on Artificial Intelligence*, pages 1794–1800, 2011. (page 114).

Gorce, Martin de La; Paragios, Nikos, and Fleet, David J. Model-based hand tracking with texture, shading and self-occlusions. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008. (page 21).

Grabner, Helmut; Matas, Jiri; Gool, Luc Van, and Cattin, Philippe. Tracking the invisible: Learning where the object might be. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1285–1292, 2010. (page 20).

Grosan, Crina and Abraham, Ajith. A novel global optimization technique for high dimensional functions. *Int. J. Intell. Syst.*, 24(4):421–440, Apr 2009. ISSN 0884-8173. (page 85).

Grosan, Crina; Abraham, Ajith, and Hassainen, Aboul Ella. A line search approach for high dimensional function optimization. *Telecommunication Systems*, 46:217–243, 2011. ISSN 1018-4864. (page 85).

Grossman, Tovi; Wigdor, Daniel, and Balakrishnan, Ravin. Multi-finger gestural interaction with 3D volumetric displays. In *ACM SIGGRAPH*, pages 61–70, 2005. (page 20).

Guan, H Y; Chua, C S, and Ho, Y K. Hand posture estimation from 2D monocular image. In *International Conference on 3-D Digital Imaging and Modeling*, pages 424–429, 1999. (page 20).

Gumpp, Thomas; Azad, Pedram; Welke, Kai; Oztop, Erhan; Dillmann, Ruediger, and Cheng, Gordon. Unconstrained real-time markerless hand tracking for humanoid interaction. In *IEEE-RAS International Conference on Humanoid Robots*, pages 88–93, 2006. (page 20).

Gupta, Abhinav; Kembhavi, Aniruddha, and Davis, Larry. Observing human-object interactions: Using spatial and funcitional compatibility for recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(10):1775–1789, October 2009a. (pages 27, 124).

Gupta, Abhinav; Kembhavi, Aniruddha, and Davis, Larry. Observing human-object interactions:

Using spatial and funcitional compatibility for recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(10):1775–1789, 2009b. (page 86).

Guskov, Igor; Klibanov, Sergey, and Bryant, Benjamin. Trackable surfaces. In *ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 251–258, 2003. (page 20).

Hackett, J.K. and Shah, M. Multi-sensor fusion: a perspective. In *IEEE International Conference on Robotics and Automation*, pages 1324–1330 vol.2, 1990. (page 72).

Halbert, Daniel Conrad. *Programming by Example*. PhD thesis, Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, 1984. (page 24).

Hamer, Henning; Schindler, Konrad; Koller-Meier, Esther, and Gool, Luc Van. Tracking a hand manipulating an object. In *IEEE International Conference on Computer Vision*, pages 1475–1482, 2009. (page 21).

Hamer, Henning; Gall, Juergen; Weise, Thibaut, and Gool, Luc Van. An object-dependent hand pose prior from sparse training data. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 671–678, 2010. (page 21).

Han, Li; Li, Zexiang; Trinkle, Jeff C; Qin, Zhiqiang, and Jiang, Shilong. The planning and control of robot dextrous manipulation. In *International Conference on Robotics and Automation*, pages 263–269, 2000. (page 8).

Handle-project, . Protocol for the corpus of sensed grasp and handling data. HANDLE project, D4, 2009a. (pages 61, 76).

Handle-project, . Tactile sensing for data-gloves. HANDLE project, D6, 2009b. (pages 56, 76).

Handle-project, . Automatic dataset reduction system for grasp motion aata. HANDLE project, D17, 2011. (pages 124, 126).

Handle-project, . Parameterizing and creating new actions. HANDLE project, D24, 2012. (page 73).

Haschke, R; Steil, J J; Steuwer, I, and Ritter, H. Task-oriented quality measures for dextrous grasping. In *IEEE International Symposium on Computational Intelligence in Robotics and Automation*, pages 689–694, 2005. (page 7).

Hasegawa, Yasuhisa; Ioka, Hayato; Fukuda, Toshio, and Kanada, Kensaku. Dexterous manipulation from pinching to power grasping. In *IEEE International Conference on Robotics and Automation*, pages 785 – 790, 2003. (page 6).

Heap, Tony and Hogg, David. Towards 3D hand tracking using a deformable model. In *International Conference on Automatic Face and Gesture Recognition*, pages 140–145, 1996. (page 20).

Hebb, Donald Olding. *The organization of behavior: a neuropsychological theory*. New York: Wiley & Sons, 1949. (page 86).

Helwig, S. and Wanka, R. Particle Swarm Optimization in high-dimensional bounded search spaces. In *Swarm Intelligence Symposium, 2007. SIS 2007. IEEE*, pages 198–205, 2007. (page 98).

Hendrich, Norman; Klimentjew, Denis, and Zhang, Jianwei. Multi-sensor based segmentation of human manipulation tasks. In *IEEE Conference on Multisensor Fusion and Integration for Intelligent Systems*, pages 223–229, 2010. (pages 61, 68, 76).

Herbst, Evan; Ren, Xiaofeng, and Fox, Dieter. RGB-D Flow: Dense 3-D motion estimation using color and depth. In *IEEE International Conference on Robotics and Automation*, 2013. (page 22).

Herda, Lorna; Fua, Pascal; Plaenkers, Ralf, and Boulic, Ronan. Using skeleton-based tracking to increase the reliability of optical motion capture. *Human Movement Science*, 20(3):313–341, 2001. (page 20).

Hirano, Yutaka; Kitahama, Ken-ichi, and Yoshizawa, Shintaro. Image-based object recognition and dexterous hand/arm motion planning using RRTs for grasping in cluttered scene. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2041–2046, 2005. (page 18).

Hochberg, Leigh R; Bacher, Daniel; Jarosiewicz, Beata; Masse, Nicolas Y; Simeral, John D; Vogel, Joern; Haddadin, Sami; Liu, Jie; Cash, Sydney S; Smagt, Patrick van der, and Donoghue, John P. Reach and grasp by people with tetraplegia using a neurally controlled robotic arm. *Nature*, 485(7398):372–375, May 2012. (page 23).

Honda, Yuichiro; Weber, Stefan, and Lueth, Tim C. Intelligent recognition system for hand gestures. In *International IEEE EMBS Conference on Neural Engineering*, pages 611–614, 2007. (page 23).

Hong, Lang. Two-level JPDA-NN and NN-JPDA Tracking algorithms. In *American Control Conference*, pages 1057–1061, 1994. (page 19).

Hourtash, Arjang. The CX-space: A unified paradigm for grasping using multifingered hands. In *IEEE International Conference on Robotics and Automation*, pages 2117–2123, 2006. (page 8).

Howard, Matthew; Klanke, Stefan; Gienger, Michael; Georick, Christian, and Vijayakumar, Sethu. A novel method for learning policies from variable constraint data. *Auton Robot*, 27 (2):105–121, 2009a. (page 24).

Howard, Matthew; Klanke, Stefan; Gienger, Michael; Goerick, Christian, and Vijayakumar, Sethu. A novel method for learning policies from variable constraint data. *Autonomous Robots*, 27:105–121, 2009b. ISSN 0929-5593. (page 85).

Howe, Robert D and Cutkosky, Mark R. Integrating tactile sensing with control for dextrous manipulation. In *IEEE International Workshop on Intelligent Motion Control*, 1990. (page 23).

Hsiao, Kaijen; Nangeroni, Paul; Huber, Manfred; Saxena, Ashutosh, and Ng, Andrew Y. Reactive grasping using optical proximity sensors. In *IEEE International Conference of Robotics and Automation*, pages 2098–2105, 2009. (page 24).

Huang, Zhiyong; Boulic, Ronan; Thalmann, Nadia Magnenat, and Thalmann, Daniel. *A multisensor approach for grasping and 3D interaction*, pages 235–253. Academic Press Ltd., 1995. (page 25).

Hue, Carine; Cadre, Jean-Pierre Le, and Perez, Patrick. A particle filter to track multiple objects. In *IEEE Workshop on Multi-Object Tracking*, 2001. (page 20).

Huebner, Kai; Ruthotto, Steffen, and Kragic, Danica. Minimum volume bounding box decomposition for shape approximation in robot grasping. In *IEEE International Conference on Robotics and Automation*, pages 1628–1633, 2008. (page 7).

Huenerfauth, Matt and Lu, Pengfei. Calibration Guide for CyberGlove, v4.4. Technical report, The City University of New York, 2009. (page 22).

Hueser, Markus; Baier, Tim, and Zhang, Jianwei. Learning of demonstrated grasping skills by stereoscopic tracking of human hand configuration. In *IEEE International Conference on Robotics and Automation*, pages 2795–2800, 2006. (page 25).

Ijspeert, Auke Jan; Nakanishi, Jun, and Schaal, Stefan. Movement imitation with nonlinear dynamical systems in humanoid robots. In *IEEE International Conference on Robotics and Automation*, pages 1398–1403, 2002. (page 30).

Imai, Yoshiro; Namiki, Akio; Hashimoto, Koichi, and Ishikawa, Masatoshi. Dynamic active catching using a high-speed multifingered hand and a high-speed vision system. In *IEEE International Conference on Robotics and Automation*, pages 1849–1854, 2004. (page 6).

Ionescu, Bogdan; Coquln, Dldler; Lambert, Patrick, and Buzulolu, Vaslle. Dynamic hand gesture recognition using the skeleton of the hand. *EURASIP Journal on Applied Signal Processing*, 13(1):2101–2109, January 2005. (page 21).

Ishihara, Tatsuya; Namiki, Akio; Ishikawa, Masatoshi, and Shimojo, Makoto. Dynamic pen spinning using a high-speed multifingered hand with high-speed tactile sensor. In *IEEE-RAS International Conference on Humanoid Robots*, pages 258–263, 2006. (page 7).

Ishikawa, Lab. Ishikawa Lab website. http://www.k2.t.u-tokyo.ac.jp/index-e.html, Retrieved June 20, 2013, 2013. (page 4).

Ishikawa, M. and Sasaki, N. Gesture recognition based on SOM using multiple sensors. In *International Conference on Neural Information Processing*, pages 1300–1304 vol.3, 2002. (page 72).

Jain, A. K.; Murty, M. N., and Flynn, P. J. Data clustering: a review. *ACM Computing Surveys*, 31:264–323, 1999. (page 69).

Jau, Bruno M. Human-like compliance for dexterous robot hands. In *World Congress on the Theory of Machines and Mechanisms*, 1995. (page 8).

Jetchev, Nikolay and Toussaint, Marc. Fast motion planning from experience: trajectory prediction for speeding up movement generation. *Autonomous Robots*, 34(1-2):111–127, 2013. (page 125).

Jiang, Hyoyoung; Jung, Jin-Woo, and Bien, Zeungnam. A study on vision-based robust hand-posture recognition by learning similarity between hand-posture and structure. In *International Conference on Advances in Natural Computation*, pages 550–559, 2006. (page 18).

Ju, Zhaojie and Liu, Honghai. Recognizing hand grasp and manipulation through empirical copula. *International Journal of Social Robotics*, 2:321–328, 2010. (page 69).

Ju, Zhaojie and Liu, Honghai. Hand motion recognition via fuzzy active curve axis Gaussian mixture models: A comparative study. In *IEEE International Conference on Fuzzy Systems*, pages 699–705, 2011. (page 69).

Judah, Kshitij; Roy, Saikat; Fern, Alan, and Dietterich, Thomas G. Reinforcement learning via practice and critique advice. In *AAAI Conference on Artificial Intelligence*, pages 481–486, 2010. (page 114).

Katz, Dov; Pyuro, Yuri, and Brock, Oliver. Learning to manipulate articulated objects in unstructured environments using a grounded relational representation. In *Robotics: Science and Systems IV*, 2008. (page 25).

Kawasaki, Haruhisa; Shimomura, Hisayuki, and Shimizu, Yuuji. Educational–industrial complex development of an anthropomorphic robot hand 'Gifu hand'. *Advanced Robotics*, 15(3):357–363, 2001. (page 4).

Kennedy, J and Eberhart, R. Particle Swarm Optimization. In *IEEE International Conference on Neural Networks*, pages 1942–1948, 1995. (page 84).

Keogh, Eamonn; Wei, Li; Xi, Xiaopeng; Vlachos, Michail; Lee, Sang-Hee, and Protopapas, Pavlos. Supporting exact indexing of arbitrarily rotated shapes and periodic time series under Euclidean and warping distance measures. *The VLDB Journal*, 18:611–630, 2009. (page 69).

Kerdvibulvech, Chutisant and Saito, Hideo. Model-based hand tracking by chamfer distance and adaptive color learning using particle filter. *EURASIP Journal on Image and Video Processing*, 2009(6):1–10, January 2009. (page 21).

Khalil, Fouad F and Payeur, Pierre. *Dexterous robotic manipulation of deformable objects with multi-sensory feedback - a review*, chapter 28, pages 587–619. InTech, 2010. (page 125).

Khan, Zia; Balch, Tucker, and Dellaert, Frank. An MCMC-based particle filter for tracking multiple interacting targets. In *European Conference on Computer Vision*, pages 279–290, 2003. (page 20).

Khansari-Zadeh, S Mohammad and Billard, Aude. Learning stable nonlinear dynamical systems with gaussian mixture models. *IEEE Transactions on Robotics*, 27(5):943–957, October 2011. (page 26).

Kjellstroem, Hedvig; Romero, Javier; Martinez, David, and Kragic, Danica. Simultaneous visual recognition of manipulation actions and manipulated objects. In *European Conference on Computer Vision*, pages 336–349, 2008. (pages 26, 86).

Kjellstroem, Hedvig; Kragic, Danica, and Black, Michael J. Tracking people interacting with objects. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 747–754, 2010. (page 21).

Kjellström, Hedvig; Romero, Javier, and Kragić, Danica. Visual object-action recognition: Inferring object affordances from human demonstration. *Computer Vision and Image Understanding*, 115(1):81–90, 2011. (page 30).

Klatzky, Roberta and Lederman, Susan J. *Object recognition by touch*, chapter 10, pages 185–207. Lawerence Erlbaum Assoc., 2008. (page 23).

Knox, W. Bradley. *Learning from Human-Generated Reward*. PhD thesis, University of Texas at Austin, 2012. (page 114).

Knox, W. Bradley and Stone, Peter. TAMER: Training an Agent Manually via Evaluative Reinforcement. In *IEEE 7th International Conference on Development and Learning*, pages 292–297, August 2008. (page 114).

Knox, W. Bradley and Stone, Peter. Interactively shaping agents via human reinforcement: the TAMER framework. In *International Conference on Knowledge Capture*, pages 9–16, 2009. (page 114).

Kobayashi, Y. and Hosoe, S. Planning-space shift learning: Variable-space motion planning toward flexible extension of body schema. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3107–3114, 2009. (page 86).

Kobayashi, Yuichi and Hosoe, Shigeyuki. Planning-space shift motion generation: Variable-space motion planning toward flexible extension of body schema. *Journal of Intelligent and Robotic Systems*, 62:467–500, 2011. (page 86).

Kondo, Masahiro; Ueda, Jun, and Ogasawara, Tsukasa. Recognition of in-hand manipulation using contact state transition for multifingered robot hand control. *Robotics and Autonormous System*, 56(1):66–81, January 2008. (page 25).

Kondo, Masahiro; Ueda, Jun, and Ogasawara, Tsukasa. Recognition of in-hand manipulation by observing contact state transition for robot hand control. In Kaneko, Makoto and Nakamura, Yoshihiko, editors, *Robotics Research*, volume 66 of *Springer Tracts in Advanced Robotics*, pages 349–360. Springer Berlin / Heidelberg, 2011. ISBN 978-3-642-14742-5. (page 69).

Koonjul, G.S.; Zeglin, G.J., and Pollard, N.S. Measuring contact points from displacements with a compliant, articulated robot hand. In *IEEE International Conference on Robotics and Automation*, pages 489–495, 2011. (page 17).

Krainin, Michael; Henry, Peter; Ren, Xiaofeng, and Fox, Dieter. Manipulator and object tracking for in-hand 3D object modeling. In *ICRA Mobile Manipulation Workshop*, 2010. (page 7).

Kulic, D.; Takano, W., and Nakamura, Y. Online segmentation and clustering from continuous observation of whole body motions. *IEEE Transactions on Robotics*, 25(5):1158–1166, 2009. (page 69).

Kulic, Dana; Kragic, Danica, and Krüger, Volker. Learning action primitives. In *Visual Analysis of Humans*, pages 333–353. Springer London, 2011. (page 26).

Kulie, Dana; Takano, Wataru, and Nakamura, Yoshihiko. Online segmentation and clustering from continuous observation of whole body motions. *IEEE Transactions on Robotics*, 25(5): 1158–1166, October 2009. (page 26).

Kulvicius, T.; Ning, K.; Tamosiunaite, M., and Worgoetter, F. Joining movement sequences: modified dynamic movement primitives for robotics applications exemplified on handwriting. *IEEE Transactions on Robotics*, 28(1):145–157, 2012. (page 125).

Kuzborskij, I.; Gijsberts, A., and Caputo, B. On the challenge of classifying 52 hand movements from surface electromyography. In *International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 4931–4937, 2012. (page 23).

Kwartowitz, David Morgan; Miga, Michael I; Herrell, S Duke, and Galloway, Robert L. Towards image guided robotic surgery: multi-arm tracking through hybrid localization. *International Journal of Computer Assisted Radiology and Surgery*, 4(3):281–286, 2009. (page 25).

Kwok, Roberta. Neuroprosthetics: Once more, with feeling. *Nature*, 497(7448):176–178, May 2013. (page 23).

Lau, Newman; Wong, Ben, and Chow, Daniel. Motion segmentation method for hybrid characteristic on human motion. *Journal of Biomechanics*, 42(4):436–442, 2009. (page 69).

Lee, H.; Chung, J.; Chang, S., and Yoon, E. Normal and shear force measurement using a flexible polymer tactile sensor with embedded multiple capacitors. *Journal of Microelectromech. Syst.*, 17:934–942, 2008. (page 24).

Lepetit, Vincent; Moreno-Noguer, Francesc, and Fua, Pascal. EPnP: An accurate O(n) Solution to the PnP Problem. *International Journal of Computer Vision*, 81(2):155–166, February 2009. (page 20).

Li, Jiting; Su, Wenkui; Zhang, Yuru, and Guo, Weidong. Vision-based grasp planning system for dexterous hands. In *International Conference on Intelligent Manipulation and Grasping*, pages 555–558, July 2004. (page 20).

Li, Qiang; Meier, M.; Haschke, R.; Ritter, H., and Bolder, B. Object dexterous manipulation in hand based on Finite State Machine. In *International Conference on Mechatronics and Automation*, pages 1185–1190, 2012. (page 25).

Lin, John; Wu, Ying, and Huang, Thomas S. Modeling the constraints of human hand motion. In *Workshop on Human Motion*, pages 121–126, 2000. gesture recognition. (page 7).

Litomisky, Krystof. Consumer RGB-D cameras and their applications. Technical report, University of California, Riverside, 2012. (page 21).

Liu, H.; Wu, K.; Meusel, P.; Seitz, N.; Hirzinger, G.; Jin, M.H.; Liu, Y.W.; Fan, S.W.; Lan, T., and Chen, Z.P. Multisensory five-finger dexterous hand: The DLR/HIT Hand II. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3692–3697, 2008. (page 3).

Lopes, Manuel; Melo, Francisco; Montesano, Luis, and Santos-Victor, José. Abstraction levels for robotic imitation: Overview and computational approaches. In Sigaud, Olivier and Peters, Jan, editors, *From Motor Learning to Interaction Learning in Robots*, volume 264 of *Studies in Computational Intelligence*, pages 313–355. Springer Berlin / Heidelberg, 2010. (page 86).

Lovchik, C S and Diftler, M A. The robonaut hand: A dexterous robot hand for space. In *IEEE International Conference on Robotics and Automation*, pages 907–912, 1999. (pages 3, 4, 83).

Mah, Christopher D. and Mussa-Ivaldi, Ferdinando A. Generalization of object manipulation skills learned without limb motion. *The Journal of Neuroscience*, 23(12):4821–4825, 2003. (page 31).

Maitin-Shepard, Jeremy; Cusumano-Towner, Marco; Lei, Jinna, and Abbeel, Pieter. Cloth grasp point detection based on multiple-view geometric cues with application to robotic towel folding. In *IEEE International Conference on Robotics and Automation*, pages 2308–2315, 2010. (pages 6, 7).

Martins, Ricardo; Faria, Diego R, and Dias, Jorge. Symbolic level generalization of in-hand manipulation tasks from human demonstrations using tactile data information. In *IEEE/RSJ International Conference on Intelligent Robots and Systems Workshop on Grasp Planning and Task Learning by Imitation*, pages 64–70, 2010. (pages 26, 76, 78).

Marzke, Mary W and Marzke, R F. Evolution of the human hand : approaches to acquiring, analysing and interpreting the anatomical evidence. *Journal of Anatomy*, 197(1):121–140, July 2000. (page 1).

Mason, Matthew T; Rodriguez, Alberto; Srinivasa, Siddhartha S, and Vazquez, Andrés S. Autonomous manipulation with a general-purpose simple hand. *Int. J. Rob. Res.*, 31(5):688–703, April 2012. (page 126).

Mason, MatthewT.; Srinivasa, SiddharthaS., and Vazquez, AndresS. Generality and Simple Hands. In *Robotics Research*, volume 70 of *Springer Tracts in Advanced Robotics*, pages 345–361. Springer-Verlag, Berlin/Heidelberg, Germany, 2011. (page 8).

Matsuo, Kazuya; Murakami, Kouji; Hasegawa, Tsutomu, and Kurazume, Ryo. A decision method for the placement of tactile sensors for manipulation task recognition. In *IEEE International Conference on Robotics and Automation*, pages 1641–1646, 2008. (page 7).

Matsuo, Kazuya; Murakami, Kouji; Hasegawa, Tsutomu; Tahara, Kenji, and Kurazume, Ryo. Segmentation method of human manipulation task based on measurement of force imposed by a human hand on a grasped object. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1767–1772, 2009. (page 69).

Matsuoka, Yoky. Embodiment and manipulation learning process for a humanoid hand. Technical report, MIT Artificial Intelligence Laboratory, 1995. (page 17).

Mattar, Ebrahim. A survey of bio-inspired robotics hands implementation: New directions in dexterous manipulation. *Robotics and Autonomous Systems*, 61(5):517–544, 2013. (pages 2, 83).

Maycock, Jonathan; Steffen, Jan; Haschke, Robert, and Ritter, Helge. Robust tracking of human hand postures for robot teaching. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2948–2953, 2011. (page 20).

Melchior, Nik A and Simmons, Reid. Dimensionality reduction for trajectory learning from demonstration. In *IEEE International Conference on Robotics and Automation*, pages 2953–2958, 2010. (page 8).

Meltzoff, Andrew N. and Moore, Keith M. Explaining facial imitation: a theoretical model. *Early Development and Parenting*, 6(3-4):179–192, 1997. (page 86).

Merckel, Loic and Nishida, Toyoaki. Solution of the Perspective-Three-Point Problem: Calculation from video image by using inclinometers attached to the camera. In *International Conference on Industrial Engineering, and Other Applications of Applied Intelligent Systems*, pages 324–333, 2007. (page 20).

Metta, Giorgio and Fitzpatrick, Paul. Better vision through manipulation. *Adaptive Behavior*, 11 (2):109–128, 2003. (page 18).

Mindtrans, Website. A detailed online table for presenting the best five anthropomorphic robotic hands/arms. http://mindtrans.narod.ru/hands/hands.htm, Retrieved June 20, 2013, 2013. (page 2).

Mitsantisuk, C.; Ohishi, K., and Katsura, S. Control of interaction force of twin direct-drive motor system using variable wire rope tension with multisensor integration. *IEEE Transactions on Industrial Electronics*, pages 498–510, 2012. (page 72).

Mizusawa, Satoru; Namiki, Akio, and Ishikawa, Masatoshi. Tweezers type tool manipulation by a multifingered hand using a high-speed visual servoing. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2709–2714, 2008. (pages 6, 7).

Montesano, L.; Lopes, M.; Bernardino, A., and Santos-Victor, J. Affordances, development and imitation. In *IEEE International Conference on Development and Learning*, pages 270–275, 2007. (page 86).

Moore, Brian and Oztop, Erhan. Robotic grasping and manipulation through human visuomotor learning. *Robot. Auton. Syst.*, 60(3):441–451, March 2012. (page 6).

Mouri, Tetsuya; Endo, Takahiro, and Kawasaki, Haruhisa. Review of Gifu Hand and Its Application. *Mechanics Based Design of Structures and Machines*, 39(2):210–228, 2011. (page 4).

Namiki, Akio; Nakabo, Yoshihiro; Ishii, Idaku, and Ishikawa, Masatoshi. 1-ms sensory-motor fusion system. *IEEE/ASME Transactions on Mechatronics*, 5(3):244–252, 2000. (page 6).

Namiki, Akio; Imai, Yoshiro; Ishikawa, Masatoshi, and Kaneko, Makoto. Development of a high-speed multifingered hand system and its application to catching. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2666–2671, 2003. (page 6).

Narasimhan, Sundar; Siegel, David M, and Hollerbach, John M. CONDOR: An architecture for controlling the Utah-MIT dexterous hand. *IEEE Transactions on Robotics and Automation*, 5 (5):616–627, 1989. (page 4).

Nguyen, P T A; Arimoto, S; Nguyen, C H, and Yoshida, M. Dexterous manipulation of an object by means of multi-DOF robotic fingers with hemi-spherical rigid tips. In *International Conference on Hyperbolic Problems: Theory, Numerics, Applications*, pages 49–58, 2004. (page 7).

Nguyen-Tuong, Duy and Peters, Jan. Using model knowledge for learning inverse dynamics. In *IEEE International Conference on Robotics and Automation*, pages 2677–2682, 2010. (page 8).

Odhner, L. U.; Ma, R. R., and Dollar, A. M. Open-Loop Precision Grasping With Underactuated Hands Inspired by a Human Manipulation Strategy. *Automation Science and Engineering, IEEE Transactions on*, PP(99):1–8, 2013. (page 83).

Odhner, Lael U and Dollar, Aaron M. Dexterous manipulation with underactuated elastic hands. In *IEEE International Conference on Robotics and Automation*, pages 5254–5260, 2011. (page 2).

Ogawa, Naoko; Sakaguchi, Yutaka; Namiki, Akio, and Ishikawa, Masatoshi. Adaptive dynamics matching in sensory-motor fusion system and its implementation by reinforcement learning. In *IEEE International Conference on Robotics and Automation, Workshop Learning Skill for*

Enhancing the Capability of Artificial System, 2002. (page 6).

Ogawa, Naoko; Sakaguchi, Yutaka; Namiki, Akio, and Ishikawa, Masatoshi. Adaptive acquisition of dynamics matching in sensory-motor fusion system. *Electronics and Communications*, 89(7):19–30, 2006. (page 6).

Ogawara, K.; Takamatsu, J.; Kimura, H., and Ikeuchi, K. Modeling manipulation interactions by hidden Markov models. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 2, pages 1096–1101 vol.2, 2002. (pages 30, 126).

Ogino, M.; Toichi, H.; Asada, M., and Yoshikawa, Y. Imitation faculty based on a simple visuo-motor mapping towards interaction rule learning with a human partner. In *International Conference on Development and Learning*, pages 148–148, 2005. (page 24).

Ognibene, Dimitri; Rega, Angelo, and Baldassarre, Gianluca. A model of reaching that integrates reinforcement learning and population encoding of postures. In *International Conference on From Animals to Animats: Simulation of Adaptive Behavior*, pages 381–393, 2006. (page 86).

Olofson, Eric Lee. *Infants'processing of action for gist*. PhD thesis, Department of Psychology, University of Oregon, 2008. (page 1).

Osswald, Dirk and Woern, Heinz. Mechanical system and control system of a dexterous robot hand. In *IEEE-RAS Conference on Humanoids Robots*, 2001. (page 2).

Ozawa, Ryuta; Arimoto, Suguru; Nakamura, Shinsuke, and Bae, Ji-Hun. Control of an Object With Parallel Surfaces by a Pair of Finger Robots Without Object Sensing. *IEEE Transactions of Robotics*, 21(5):965–976, October 2005. (page 7).

Oztop, E. and Arbib, M.A. A biologically inspired learning to grasp system. In *International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 857–860, 2001. (page 86).

J. LoboP. Trindade, J. P. Barreto. Hand gesture recognition using colour and depth images enhanced with hand angular pose data. In *IEEE International Conference on Multisensor Fusion and Information Integration*, pages 71–76, 2012. (page 18).

PalChaudhuri, Santashil; Saha, Amit Kumar, and Johnson, David B. Adaptive clock synchronization in sensor networks. In *International Symposium on Information Processing in Sensor Networks*, pages 340–348, 2004. (page 54).

Papadourakis, Vasilis and Argyros, Antonis. Multiple objects tracking in the presence of long-term occlusions. *Computer Vision and Image Understanding*, 114(7):835–846, July 2010. (page 20).

Pardowitz, Michael and Dillmann, Ruediger. Towards life-long learning in household robots: the Piagetian Approach. In *IEEE International Conference on Development and Learning*, pages 88–93, 2007. (page 25).

Pardowitz, Michael; Knoop, Steffen; Dillmann, Ruediger, and Zoellner, Raoul D. Incremental learning of tasks from user demonstrations, past experiences, and vocal comments. *IEEE Transactions on Systems, Man, and Cybernetics*, 37(2):322–332, April 2007. (page 25).

Park, Woojin; Chaffin, Don B.; Martin, Bernard J., and Faraway, Julian J. A computer algorithm for representing spatial-temporal structure of human motion and a motion generalization method. *Journal of Biomechanics*, 38(11):2321–2329, 2005. (page 31).

Pastor, P.; Kalakrishnan, M.; Chitta, S.; Theodorou, E., and Schaal, S. Skill learning and task outcome prediction for manipulation. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 3828–3834, 2011. (pages 30, 125).

Pastor, Peter; Hoffmann, Heiko; Asfour, Tamim, and Schaal, Stefan. Learning and generalization of motor skills by learning from demonstration. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 1293–1298, 2009. (page 31).

Perani, D; Fazio, F; Borghese, N A; Tettamanti, M; Ferrari, S; Decety, J, and Gilardi, M C. Different brain correlates for watching real and virtual hand actions. *Neuro Image*, 14(3): 749–758, 2001. (page 9).

Petroff, Neil and Goodwine, Bill. Nonholonomic and stratified robotic manipulation supplemented with fuzzy control: Theory and experiment. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1202–1208, October 2010. (page 8).

Peursum, Patrick; Venkatesh, Svetha, and West, Geoff. A study on smoothing for particle-filtered 3D human body tracking. *International Journal of Computer Vision*, 87(1-2):53–74, 2010. (page 20).

Pilarski, P.M.; Dawson, M.R.; Degris, T.; Fahimi, F.; Carey, J.P., and Sutton, R.S. Online human training of a myoelectric prosthesis controller via actor-critic reinforcement learning. In *IEEE Int Conf Rehabil Robot*, pages 1–7, 2011. (page 114).

Piperagkas, Grigoris S.; Georgoulas, George; Parsopoulos, Konstantinos E.; Stylios, Chrysostomos D., and Likas, Aristidis C. Integrating particle swarm optimization with reinforcement learning in noisy problems. In *International Conference on Genetic and Evolutionary Computation Conference*, pages 65–72, 2012. (page 98).

Pollard, Nancy S. *Parallel methods for synthesizing whole-hand grasps from generalized prototypes*. PhD thesis, Department of Electrical Engineering and Computer Science, 1994. (page 17).

Pollard, Nancy S. Closure and quality equivalence for efficient synthesis of grasps from exmaples. *International Journal of Robotics Research*, 23(6):595–613, June 2004. (pages 7, 17).

Pollard, Nancy S and Hodgins, Jessica K. Generalizing demonstrated manipulation tasks. In *Workshop on the Algorithmic Foundations of Robotics*, pages 523–540, 2002. (pages 17, 25).

Pook, P.K. and Ballard, D.H. Recognizing teleoperated manipulations. In *IEEE International Conference on Robotics and Automation*, pages 578–585 vol.2, 1993. (page 24).

Poulin-Dubois, Diane and Chow, Virginia. The effect of a looker's past reliability on infants' reasoning about beliefs. *Developmental Psychology*, 45(6):1576–1582, 2009. (page 9).

Prats, Mario; Sanz, Pedro J, and del Pobil, Angel P. Task-oriented grasping using hand preshapes and task frames. In *IEEE International Conference on Robotics and Automation*, pages 1794–1799, 2007. (page 8).

Prattichizzo, Domenico and Trinkle, Jeffrey C. *Grasping*, chapter 28, pages 671–700. Springer, 2008. (page 7).

Preisig, Peter and Kragic, Danica. Robust Statistics for 3D Object Tracking. In *IEEE International Conference on Robotics and Automation*, pages 2403–2408, 2006. (page 17).

Prieur, U.; Perdereau, V., and Bernardino, A. Modeling and planning high-level in-hand manipulation actions from human knowledge and active learning from demonstration. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 1330–1336, 2012. (page 85).

Qu, Xingda and Nussbaum, M.A. Simulating Human Lifting Motions Using Fuzzy-Logic Control. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 39 (1):109–118, 2009. (page 31).

Raibert, Marc; Blankespoor, Kevin; Nelson, Gabriel, and Playter, Rob. BigDog, the rough-terrain quaduped robot. In *World Congress of the International Federation of Automatic Control*, pages 10823–10825, 2008. (page 17).

Rakthanmanon, T.; Keogh, E.J.; Lonardi, S., and Evans, S. Time series epenthesis: Clustering time series streams requires ignoring some data. In *IEEE International Conference on Data Mining*, pages 547–556, 2011. (page 69).

Rameon, Juan Antonio Corrales; Medina, Fernando Torres, and Perdereau, Véronique. Finger readjustment algorithm for object manipulation based on tactile information. *International Journal of Advanced Robotic Systems*, 10(9):1–9, 2013. (page 23).

Ratliff, NathanD.; Silver, David, and Bagnell, J.Andrew. Learning to search: Functional gradient techniques for imitation learning. *Autonomous Robots*, 27:25–53, 2009. (page 85).

Rehg, James M. *Visual analysis of high DOF articulated objects with application to hand tracking*. PhD thesis, School of Computer Science, CMU, 1995. (page 20).

Rehg, James M and Kanade, Takeo. Model-based tracking of self-occluding articulated objects. In *IEEE International Conference on Computer Vision*, pages 612–617, 1995. (page 20).

Riley, M. and Cheng, G. Extracting and generalizing primitive actions from sparse demonstration. In *IEEE-RAS International Conference on Humanoid Robots*, pages 630–635, 2011. (page 31).

Rolf, M.; Steil, J.J., and Gienger, M. Goal babbling permits direct learning of inverse kinematics. *IEEE Transactions on Autonomous Mental Development*, 2(3):216–229, 2010. (page 86).

Romero, Javier; Kjellstroem, Hedvig, and Kragic, Danica. Human-robot mapping of grasps. In *Grasp and Task Learning by imitation workshop, IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2008a. (page 25).

Romero, Javier; Kragic, Danica; Kyrki, Ville, and Argyros, Antonis. Dynamic time warping for binocular hand tracking and reconstruction. In *IEEE International Conference on Robotics and Automation*, pages 2289–2294, 2008b. (page 25).

Romero, Javier; Kjellstroem, Hedvig, and Kragic, Danica. Monocular real-time 3D articulated hand pose estimation. In *IEEE-RAS International Conference on Humanoid Robots*, pages 87–92, 2009a. (page 25).

Romero, Javier; Kjellstroem, Hedvig, and Kragic, Danica. Modeling and evaluation of human-to-robot mapping of grasps. In *International Conference on Advanced Robotics*, pages 1–6, 2009b. (pages 17, 25, 124).

Romero, Javier; Kjellstroem, Hedvig, and Kragic, Danica. Hands in action: Real-time 3D reconstruction of hands in interaction with objects. In *IEEE International Conference on Robotics and Automation*, pages 458–463, 2010. (pages 25, 124).

Saegusa, Ryo; Metta, Giorgio; Sandini, Giulio, and Sakka, Sophie. Active motor babbling for sensorimotor learning. In *IEEE International Conference on Robotics and Biomimetics*, pages 794–799, 2008. (page 86).

Sahari, Khairul Salleh; Seki, Hiroaki; Kamiya, Yoshitsugu, and Hikizu, Masatoshi. Clothes Manipulation by Robot Grippers with Roller Fingertips. *Advanced Robotics*, 24(1-2):139–158, 2010. (page 7).

Sahbani, Anis; Saut, Jean-Philippe, and Perdereau, Veronique. An efficient algorithm for dexterous manipulation planning. In *IEEE International Multi-Conference on Systems, Signals and Devices*, 2007. (pages 25, 124).

Sauser, Eric L; Argall, Brenna D; Metta, Giorgio, and Billard, Aude G. Iterative learning of grasp adaptation through human corrections. *Robotic Autonomous System*, 60(1):55–71, January 2012. (pages 6, 23, 125).

Saut, Jean-Philippe; Sahbani, Anis, and Perdereau, Vieronique. A global approach for dexterous manipulation planning using paths in n-fingers grasp subspace. In *International Conference on Control, Automation, Robotics and Vision*, pages 1–6, 2006. (pages 25, 111, 124).

Saxena, Ashutosh; Driemeyer, Justin, and Ng, Andrew Y. Robotic Grasping of Novel Objects using Vision. *International Journal of Robotics Research*, 27(2):157–173, February 2008. (pages 7, 19).

Schaal, S.; Ijspeert, A., and Billard, A. Computational approaches to motor learning by imitation. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 358(1431):537–547, March 2003a. (page 30).

Schaal, Stefan; Ijspeert, Auke, and Billard, Aude. Computational approaches to motor learning by imitation. *Philosophical Transaction of the Royal Society of London*, 358(1431):537–547, 2003b. (page 24).

Schmidt, R.A. and Lee, T.D. *Motor control and learning: a behavioral emphasis*. Human Kinetics, 1999. ISBN 9780880114844. (page 31).

Schmidt, Richard A. A schema theory of discrete motor skill learning. *Psychological Review*, 82(4):225–260, 1975. (page 31).

Scholz, Volker; Stich, Timo; Keekeisen, Michael; Wacker, Markus, and Magnor, Marcus. Garment motion capture using color-coded patterns. In *ACM SIGGRAPH*, volume 24, pages 439–447, 2005. (page 20).

Schweighofer, Gerald and Pinz, Axel. Robust pose estimation from a planar target. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(12):2024–2030, December 2006. (page 20).

Sciuto, Lorenzo Giovanni. Robotic hand and sensorized glove: A calibration for managing robotic grasp in teleoperation. Master's thesis, University of Siena, 2011. (page 22).

Senoo, Taku; Namiki, Akio, and Ishikawa, Masatoshi. High-speed batting using a multi-jointed manipulator. In *IEEE International Conference on Robotics and Automation*, pages 1191–1196, 2004. (page 6).

Senoo, Taku; Namiki, Akio, and Ishikawa, Masatoshi. Ball control in high-speed batting motion using hybrid trajectory generator. In *IEEE International Conference on Robotics and Automation*, pages 1762–1767, 2006. (page 6).

Senoo, Taku; Namiki, Akio, and Ishikawa, Masatoshi. High-speed throwing motion based on kinetic chain approach. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3206–3211, 2008. (page 6).

Senoo, Taku; Yamakawa, Yuji; Mizusawa, Satoru; Namiki, Akio; Ishikawa, Masatoshi, and Shimojo, Makoto. Skillful manipulation based on high-speed sensory-motor fusion. In *IEEE International Conference on Robotics and Automation*, pages 1611–1612, 2009. (page 6).

ShadowRobot, Company. Shadowhand website. http://www.shadowrobot.com, Retrieved June 20, 2013, 2013. (pages 4, 30, 93).

Shi, Y. and Eberhart, R.C. A modified particle swarm optimizer. In *IEEE International Conference on Evolutionary Computation*, pages 69–73, 1998a. (page 84).

Shi, Yuhui and Eberhart, RussellC. Parameter selection in particle swarm optimization. In *Evolutionary Programming VII*, volume 1447 of *Lecture Notes in Computer Science*, pages 591–600. Springer Berlin Heidelberg, 1998b. (page 98).

Shimada, Nobutaka; Shirai, Yoshiaki; Kuno, Yoshinori, and Miura, Jun. Hand gesture estimation and model refinement using monocular camera - ambiguity limitation by inequality constraints. In *IEEE International Conference on Automatic Face and Gesture Recognition*, pages 268–273, 1998. (page 20).

Shimojo, Makoto; Namiki, Akio; Ishikawa, Masatoshi; Makino, Ryota, and Mabuchi, Kunihiko. A tactile sheet using pressure conductive rubber with electrical-wires stitched method. *IEEE Sensors Journal*, 4(5):589–596, 2004. (page 6).

Shimojo, Makoto; Araki, Takuma; Ming, Aigou, and Ishikawa, Masatoshi. A high-speed mesh of tactile sensors fitting arbitrary surfaces. *IEEE Sensors Journal*, 10(4):822–830, April 2010. (page 6).

Shiokata, Daisuke; Namiki, Akio, and Ishikawa, Masatoshi. Robot dribbling using a high-speed multifingered hand and a high-speed vision system. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3945–3950, 2005. (page 6).

Shon, Aaron P.; J., Storz Joshua; Meltzoff, Andrew N., and Rao, Rajesh P. N. A cognitive model of imitative development in humans and machines. *International Journal of Humanoid Robotics*, 4(2):387–406, 2007. (page 86).

Sjoelander, Sverre. Singing birds, playing cats and babbling babies: Why do they do it? *Biology of Communication and Motor Processes*, 57(2-4):197–204, 2000. (page 9).

Smeets, Jeroen B J and Brenner, Eli. A new view on grasping. *Motor Control*, 3(3):237–271, July 1999. (page 8).

Song, D.; Kyriazis, N.; Oikonomidis, I.; Papazov, C.; Argyros, A.; Burschka, D., and Kragic,

D. Predicting Human Intention in Visual Observations of Hand/Object Interactions. In *IEEE International Conference on Robotics and Automation*, 2013. (page 126).

Stefanov, Nikolay; Galata, Aphrodite, and Hubbold, Roger. Real-time hand tracking with variable-length markov models of behaviour. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 73–80, 2005. (page 21).

Steffen, Jan; Haschke, Robert, and Ritter, Helge. Using manifolds for dextrous hand control. In *RSS 2008 Workshop on Robot Manipulation*, 2008a. (pages 25, 27).

Steffen, Jan; Haschke, Robert, and Ritter, Helge. Towards dextrous manipulation using manipulation manifolds. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2738–2743, 2008b. (page 25).

Steinem, C. and Janshoff, A. *Piezoelectric Sensors*. Springer-Verlag, Berlin/Heidelberg, Germany, 2007. (page 24).

Stenger, B; Mendonca, P R S, and Cipolla, R. Model-based hand tracking using an unscented Kalman Filter. In *British Machine Vision Conference*, pages 63–72, 2001a. (page 20).

Stenger, B; Mendonca, P R S, and Cipolla, R. Model-based 3D tracking of an articulated hand. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages II–310–II–315 vol. 2, 2001b. (page 20).

Stenger, Bjoern. 3D model-based hand tracking. Technical report, Department of Engineering, University of Cambridge, 2001. (page 20).

Stenger, Bjoern; Thayananthan, Arasanathan; Torr, Philip H S, and Cipolla, Roberto. Model-based hand tracking using a hierarchical Bayesian filter. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(9):1372–1384, September 2006. (page 20).

Stenger, Bjoern Dietmar Rafael. *Model-based hand tracking using a hierarchical Bayesian filter*. PhD thesis, St. John's College, 2004. (page 20).

Stone, Peter; Beeson, Patrick; Mericli, Tekin, and Madigan, Ryan. Austin robot technology. Technical report, University of Texas at Austin, 2007. (pages 4, 83).

Sturm, J.; Engelhard, N.; Endres, F.; Burgard, W., and Cremers, D. A Benchmark for the Evaluation of RGB-D SLAM Systems. In *IEEE/RSJ International Conference on Intelligent Robot Systems*, pages 573–580, 2012. (page 22).

Subramanian, Kaushik; Isbell, Charles, and Thomaz, Andrea. Learning options through human interaction. In *Workshop on Agents Learning Interactively from Human Teachers at IJCAI*, 2011. (page 114).

Sudderth, Erik B. *Graphical models for visual object recognition and tracking*. PhD thesis, Department of Electrical Engineering and Computer Science, 2006. (page 20).

Sudderth, Erik B. Describing visual scenes using transformed objects and parts. *International Journal of Computer Vision*, 77(1-3):291–330, May 2008. (page 20).

Sudsang, Attawith; Ponce, Jean, and Srinivasa, Narayan. Grasping and in-hand manipulation: Geometry and algorithms. *Algorithmica*, 26(3-4):466–493, 2000. (page 7).

Sundararaman, Bharath; Buy, Ugo, and Kshemkalyani, Ajay D. Clock synchronization for wireless sensor networks: a survey. *Ad Hoc Networks*, 3(3):281–323, 2005. (page 54).

Talati, Ardesheer; Valero-Cuevas, Francisco J, and Hirsch, Joy. Visual and tactile guidance of dexterous manipulation tasks: An fMRI study. *Perceptual and Motor Skills*, 101(1):317–334, August 2005. (page 18).

Taylor, Craig L and Schwarz, Robert J. The anatomy and mechanics of the human hand. *Artificial Limbs*, 2(2):22–35, 1955. (pages 10, 29).

Tegin, Johan and Wikander, Jan. Tactile sensing in intelligent robotic manipulation – a review. *Industrial Robot: An International Journal*, 32(32):64–70, 2005. (page 23).

Theodorou, Evangelos; Buchli, Jonas, and Schaal, Stefan. Reinforcement learning of motor skills in high dimensions. In *International Conference on Robotics and Automation*, pages 2397–2403, 2010. (page 27).

Todorov, E and Ghahramani, Z. Analysis of the synergies underlying complex hand manipulation. In *Annual International Conference of the IEEE EMBS*, pages 4637–4640, 2004. (pages 27, 125).

Torres-Jara, Eduardo; Vasilescu, Iuliu, and Coral, Raul. A soft touch: Compliant Tactile Sensors for Sensitive Manipulation. Technical report, MIT Artificial Intelligence Laboratory, 2006. (page 17).

Turner, Michael Leo. *Programming dexterous manipulation by demonstration*. PhD thesis, The Department of Mechanical Engineering, Stanford University, 2001. (page 24).

Ueda, Jun; Kondo, Masahiro, and Ogasawara, Tsukasa. The multifingered NAIST hand system for robot in-hand manipulation. *Mechanism and Machine Theory*, 45(2):224–238, 2010. (page 4).

Uni-Hamburg, HYFLAM project. HYFLAM: A Hyper-Flexible Work Cell for Biochemical Laboratory Automation. http://www.echord.info/wikis/website/hyflam, Retrieved June 20, 2013, 2013. (page 6).

Utsumi, Akira and Ohya, Jun. Direct manipulation interface using multiple cameras for hand gesture recognition. In *IEEE International Conference on Multimedia Computing and Systems*, pages 264–267, 1998. (page 19).

Utsumi, Akira; Miyasato, Tsutomu, and Kishino, Fumio. Multi-camera hand pose recognition system using skeleton image. In *IEEE International Workshop on Robot and Human Communication*, pages 219–224, 1995. (page 19).

Utsumi, Akira; Tetsutani, Nobuji, and Igi, Seiji. View-based detection of 3-D interaction between hands and real objects. In *IEEE International Conference on Pattern Recognition*, pages 961–964, 2004. (page 19).

Valtazanos, Aris; Arvind, D. K., and Ramamoorthy, S. Comparative study of segmentation of periodic motion data for mobile gait analysis. In *Wireless Health*, pages 145–154, 2010. (page 68).

Vass, Gabor. *Object manipulation planning for dextrous robot systems*. PhD thesis, Department of Control Engineering and Information Technology, Budapest University of Technology and Economics, 2005. (page 8).

Vien, Ngo Anh and Ertel, Wolfgang. Reinforcement learning combined with human feedback in continuous state and action spaces. In *Development and Learning and Epigenetic Robotics, 2012 IEEE International Conference on*, pages 1–6, 2012. (page 114).

Vinayavekhin, P.; Kudohf, S., and Ikeuchi, K. Towards an automatic robot regrasping movement based on human demonstration using tangle topology. In *2011 IEEE International Conference on Robotics and Automation*, pages 3332–3339, 2011. (pages 26, 68).

Vincze, Markus; Zillich, Michael; Ponweiser, Wolfgang; Hlavac, Vaclav; Matas, Jiri; Obdrzalek, Stepan; Buxton, Hilary; Howell, Jonathan; Sage, Kingsley; Argyros, Antonis; Eberst, Christoph, and Umgeher, Gerald. Integrated vision system for the semantic interpretation of activities where a person handles objects. *Computer Vision and Image Understanding*, 113(6): 682–692, June 2009. (page 26).

Vinjamuri, R; Weber, D; Mao, Z; Collinger, J; Degenhart, A; Kelly, J; Boninger, M; Tyler-Kabara, E, and Wang, W. Toward synergy-based brain-machine interfaces. *IEEE Trans Inf Technol Biomed*, 15, 2011. (page 85).

Vinjamuri, Ramana; Sun, Mingui; Chang, Cheng-Chun; Lee, Heung-No; Sclabassi, Robert J, and Mao, Zhi-Hong. Temporal Postural Synergies of the Hand in Rapid Grasping Tasks. *IEEE Transactions on Information Technology in Biomedicine*, 14(4):986–994, July 2010. (pages 7, 25).

Walker, A Elizabeth and Bass-Ringdahl, Sandie. Babbling complexity and its relationship to speech and language outcomes in children with cochlear implants. *Otology and Neurotology*, 29(2):225–229, February 2008. (page 9).

Wallace, Patricia S and Whishaw, Ian Q. Independent digit movements and precision grip patterns in 1-5-month-old human infants: hand-babbling, including vacuous then self-directed hand and digit movements, precedes targeted reaching. *Neuropsychologia*, 41(14):1912–1918, 2003. (page 9).

Wang, Robert Y. Real-time hand-tracking as a user input device. In *International Conference on Knowledge, Information, and Creativity Support Systems*, pages 178–189, 2008. (page 20).

Wang, Robert Y and Popovic, Jovan. Real-time hand-tracking with a color glove. In *Proceedings of ACM SIGGRAPH*, pages 63:1–63:8, 2009. (page 20).

Weigl, Alexandra and Seitz, Matthias. Vision assisted disassembly using a dexterous hand-arm-system: An example and experimental results. In *IFAC Symposium on Robot Control*, pages 314–322, 1994. (page 18).

Williams, Camille; Shang, Daniel, and Carnahan, Heather. Pressure is a viable controlled output of motor programming for object manipulation tasks. In Kappers, Astrid; van Erp, Jan; Bergmann Tiest, Wouter, and van der Helm, Frans, editors, *Haptics: Generating and Perceiving Tangible Sensations*, volume 6192 of *Lecture Notes in Computer Science*, pages 339–344. Springer Berlin / Heidelberg, 2010. ISBN 978-3-642-14074-7. (page 78).

Woehlke, Günter. Grasp planning & force computation for dextrous object manipulation with multi-finger robot hands. Technical report, International Computer Science Institute of UC Berkeley, 1994. (page 7).

Wu, Yan and Demiris, Yiannis. Towards one shot learning by imitation for humanoid robots. In *IEEE International Conference on Robotics and Automation*, pages 2889–2894, 2010. (page 26).

Wu, Ying and Huang, Thomas S. Capturing articulated human hand motion: A divide-and-conquer approach. In *IEEE International Conference on Computer Vision*, pages 606–611, 1999. (page 20).

Xu, Y.; Jiang, F.; Newbern, S.; Huang, A.; Ho, C., and Tai, Y. Flexible shear-stress sensor skin and its application to unmanned aerial vehicles. *Sens. Actuators A: Phys.*, 105:321–329, August 2003. (page 24).

Xue, Zhixing; Zoellner, J Marius, and Dillmann, Ruediger. Dexterous manipulation planning of objects with surface of revolution. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2703–2708, 2008. (page 8).

Yamakawa, Yuji; Namiki, Akio; Ishikawa, Masatoshi, and Shimojo, Makoto. One-handed knotting of a flexible rope with a high-speed multifingered hand having tactile sensors. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 703–708, 2007. (page 7).

Yamakawa, Yuji; Namiki, Akio; Ishikawa, Masatoshi, and Shimojo, Makoto. Knotting manipulation of a flexible rope by a multifingered hand system based on skill synthesis. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2691–2696, 2008. (page 7).

Yamakawa, Yuji; Namiki, Akio, and Ishikawa, Masatoshi. Card manipulation using a high-speed robot system with high-speed visual feedback. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4762–4767, 2012. (page 7).

Yamane, Katsu; Yamaguchi, Yoshifumi, and Nakamura, Yoshihiko. Human motion database with a binary tree and node transition graphs. *Autonomous Robots*, 30(1):87–98, January 2011. (page 8).

Yang, X. S. and Deb, S. Cuckoo Search via Levy flights. In *Proc. of World Congress on Nature and Biologically Inspired Computing*, pages 210–214, 2009. (page 85).

Yao, Yuan and Fu, Yun. Real-Time Hand Pose Estimation from RGB-D Sensor. In *IEEE International Conference on Multimedia and Expo*, pages 705–710, 2012. (page 22).

Yeap, S L and Trinkle, J C. Dynamic whole-arm dexterous manipulation in the plane. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 405–410, 1995. (page 7).

Yousef, Hanna; Boukallel, Mehdi, and Althoefer, Kaspar. Tactile sensing for dexterous in-hand manipulation in robotics—A review. *Sensors and Actuators A: Physical*, 167(2):171–187, June 2011. (page 23).

Zecca, Massinmilianno; Roccella, Stefano; Cappiello, Giovanni; Ito, Kazuko; Imanishi, Kazutaka; Miwa, Hiroyasu; Carrozza, M Chiara; Dario, Paolo, and Takanishi, Atsue. From the human hand to a humanoid hand: Biologically-inspired approach for the development of RoboCasa Hand #1. In *CISM-IFToMM Symposium on Robot Design, Dynamics and Control*, pages 287–294, 2006. (page 2).

Zhang, L.; Sturm, J.; Cremers, D., and Lee, D. Real-Time Human Motion Tracking using Multiple Depth Cameras. In *IEEE/RSJ International Conference on Intelligent Robot Systems*, pages 2389–2395, 2012. (page 22).

Zhang, Xu; Chen, Xiang; Vuokko, Lantz, and Wang, Kong-qiang. Hand gesture recognition and virtual game control based on 3D accelerometer and EMG sensors. In *International Conference on Intelligent User Interface*, pages 401–406, 2009. (page 23).

Zhao, Wenping; Chai, Jinxiang, and Xu, Ying-Qing. Combining marker-based Mocap and RGB-D camera for acquiring high-fidelity hand motion data. In *ACM SIGGRAPH Symposium on Computer Animation*, pages 33–42, 2012. (page 22).

Zheng, Joshua Z; Rosa, Sara De La, and Dollar, Aaron M. An investigation of grasp type and frequency in daily household and machine shop tasks. In *IEEE International Conference of Robtics and Automation*, pages 4169–4175, 2011. (pages 27, 125).

Zhou, Feng; Torre, F., and Hodgins, J.K. Aligned Cluster Analysis for temporal segmentation of human motion. In *IEEE International Conference on Automatic Face Gesture Recognition*, pages 1–7, 2008. (page 69).

Zhuang, Liansheng; Tang, Ketan; Yu, Nenghai, and Qian, Yangchun. Fast Salient Object Detection Based on Segments. In *International Conference on Measuring Technology and Mechatronics Automation*, volume 1, pages 469–472, 2009. (page 18).