HANDLE

Developmental pathway towards autonomy and dexterity in robot in-hand manipulation

Deliverable 24

Parameterizing and creating new actions

Due date of deliverable: **Month 36**
Actual submission date: **Month 36**
Partner responsible: UHAM

Version 1.1

Classification: PU
Grant Agreement Number: 231640
Contract Start Date: 2009-02-02
Duration: 48 Months
Project Coordinator: UPMC
Partners: UPMC, SHADOW, UC3M, FCTUC, KCL, ORU, UHH, CEA, IST
Project website address: www.handle-project.eu

Revision History

Date	Version	Change	Author
2011.09.14	0.1	created	fnh
2011.12.13	0.2	synergies section	fnh
2012.01.11	0.3	action-gist material	cg
2012.01.21	0.4	IST-object set experiments	fnh
2012.01.28	0.5	partner review	
2012.02.01	1.0	revision and final version	fnh
2012.03.12	1.1	postural synergies analysis	alex, fnh

Future Revision

Note: it is planned to **revise and extend** this report (version 1.2) with additional experimental data as soon as the tactile-sensing upgrade on the UPMC hand has been completed and the demonstrator platform is available for experiments.

Contents

List of Figures

List of Tables

1 Abstract

This report summarizes the current research regarding *parameterizing and creating new actions* within the HANDLE project. We describe the methods and algorithms developed in the project (task 3.2) for creating new actions, in particular the parameterization of action descriptions in action-space metrics. Therefore, the report is concerned with the behaviour of the Shadow Dextrous Hand and its motion-primitives, when performing either simple (static) grasps or sequences of manipulation motions.

In this initial step towards the creating of new actions, the modeling and the expected effects of using target object affordances plays only a minor role; see [71] for a review of the work performed so far. The upcoming reports D30 and D31 will provide a much better solution in this regard. For now, we concentrate on the definition of *action space metrics* to measure similarity in motor-terms — the measured joint-angles defining the hand-pose and tactile-sensor readings. However, the data also includes model-specific data like air-pressure for the Shadow C5-type air-muscle-hand, or the motor-currents for the C6-type motor-hand and the prototype of the new CEA hand [76,77,82], where the motor-current is related to joint-torque. As sketched in the project proposal and technical annex, a *clustering in the action space* followed by (where possible, linear) interpolation of the motion-primitives is used, providing the grasp-planning with a suitable initial solution in the action-space while also supporting generalization of object properties.

Outline of this report The remainder of this report is structured as follows. While the database of human grasps recorded and compiled by the project provides a rich source of information [63, 68], the direct transfer to the Shadow Hand has proven more difficult than initially expected. Therefore, the project decided to slightly adjust and enlarge the scope of work-package WP1 and to also record a database of grasps performed directly by the Shadow hand — while tele-operated and supervised by human experimenters to ensure the most human-like hand poses possible. Section 2.1 describes the reason behind this, and presents the setup and results of a first series of grasp experiments. All grasp-classes listed in the *grasps taxonomy* [68] were targeted by those experiments. In order to ensure the most human-like grasps for the set of target objects, human-teleoperation via the Cyberglove was used to control the hand, with careful adjustment of the finger-positions. The experiments were repeated by multiple test persons, allowing us to determine and analyse the "typical" human grasps. The results provide a ground-truth for the later experiments and confirm and extend previous work performed with the Shadow hand [17]. In particular, we demonstrate the ability of the Shadow hand to pick up and grasp a variety of everyday objects and tools.

Section 3 then concentrates on the parameterization of the *basic grasping skills*. It first summarizes the basic grasp-planning algorithms proposed by the project [70],

and then reviews the concepts from [80] regarding basic skills. Obviously, many of the grasps from the overall taxonomy, including all power-grasps, leave little room for finger movements or adjustments once the fingers are closed around the target object. Therefore, the section concentrates on the grasps we consider essential for manipulation and tool use, in particular variants of the pinch-grasp (thumb + index-finger) and the variants of the tripod-grasp. Again, human-teleoperation experiments were recorded on the UHAM C5 hand, and results are analysed and extrapolated to objects of different size and hands of different kinematics and joint-coupling. The object set was suggested and provided by partner IST and will also be used for recording similar experiments on the iCub hand. To confirm our analysis, additional experiments and an update of this report are planned, as soon as the tactile sensing solution [80] has been installed on the Paris C6 demonstrator hand and is available for recording.

The basic grasps studied in the previous sections cover only a small percentage of the overall state-space of a dextrous multi-fingered hand. To overcome the *curse of dimensionality* — grasping with the Shadow C5/C6 hands or the upcoming CEA hand involves the control of 20-DOF for the fingers and hand pose alone, in addition to the 6-DOF inherent to the relative orientation of the hand and a rigid object — a more radical approach is needed. The role of *grasp synergies* is well-known from physiology [20] and forms the topic of section 4. We first summarize the *eigengrasp* concept and algorithms available in the *GraspIt!* simulator [22–24] and the Columbia Grasp database [25]. Using an empirical mapping from the original Santello synergies to the Shadow hand, we then show how the Eigengrasp-planner can also be extended to grasp-planning on the Shadow hand, using arbitrary 3D-objects and also 3D-obstacles.

Unfortunately, neither the GraspIt! Eigengrasp-planner nor the auto-generated grasps available in the Columbia Grasp Database consider object affordances or the context of object manipulation. Instead, the standard goal-function of the Eigengrasp-planner is still grasp-stability, so that most grasps recorded in the CGDB are power-grasps and form-closure grasps with little or no leeway for object manipulation. Apart from isolated attempts [37, 40], no suitable *goal-oriented* grasp quality-measures for *manipulability* are available. Section 4.4 describes our initial attempt at recording grasp-synergies and movement-skills for the Shadow C5 hand. This includes the Principal Component Analysis of grasps already described in the previous section, as well as the modeling of task context. We also propose an algorithm to modify the finger-pose in order to adapt the basic hand-pose to the given object and given finger-kinematics based on tactile feedback. The strategy will be tested and demonstrated on the Shadow C6 hand as soon as possible, proving viability for generalization to the CEA hand design and future multi-fingered hands.

The last section of this report presents a novel approach for analysis and representation of *finger-gaiting* manipulation sequences. As described in [21], human in-hand manipulation can be reduced to a small number of basic movements. We propose to

use a *Gaussian Random Markov Field* network to represent the finger-movements recorded by human-demonstration, and demonstrate the use of the analysis.

The report concludes with a short summary and the list of references.

2 Experiment setup for recording robot motion

The recording of human grasping and manipulation actions has been the first major activity of the project, and by now the HANDLE grasp database contains traces for a large number of experiments [68, 74], ranging from simple pick-and-place tasks to fully dextrous manipulation. Given the unique multi-sensor setup with its integration of absolute finger-tip position (Polhemus), relative hand-pose (Cyberglove), tactile-data (TekScan), and data from instrumented objects (Shadow), the recordings provide a rich source of information for learning from human demonstration. Good progress has been made to annotate the recorded sequences according to the proposed grasp taxonomy and the Laban motion analysis [18], see deliverable [74] (chapter 4) for details. Additionally, approaches towards fully- (or at least semi-) automatic segmentation and annotation of the sequences based on the analysis of the finger-motions and tactile data have been proposed [19, 74] by the project.

Despite of this, the direct transfer of the human grasping-skills to the Shadow Dextrous hand has proven to be more difficult than expected. With 24-DOF overall, and 20-DOF controllable, the Shadow hand matches the human hand in shape, size, and finger kinematics. However, a few key differences remain, and several human-like grasps turned out to be surprisingly difficult to perform with the robot; see section 3.3 below for a few examples and some comments.

One reason for this is obvious regarding the human recordings: *no two human hands are the same*. Of course, the rest position of the experimenters' hands with the data-glove and the sensors on the hand have been documented for our experiment recordings, but despite of this the extraction of the hand kinematics has turned out to be extremely difficult. For several of our human experiments, the Polhemus tracking sensors were mounted on top of the TekScan tactile sensor, this in turn on top of the CyberGlove data-glove, and of course the glove was only worn on the fingers, not glued or fixed to the hand [63, 64]. Even a slight movement of the fingers inside the glove can shift all of the sensors and might require a lengthy recalibration, or a very complex estimation of the sensor-positions from the recorded data itself, e.g. by checking for certain invariants like a known size of the grasped object. Some of the issues related to the mapping from the human hand to the robot have been noted in previous deliverables [69, 78].

Therefore, we decided to try the obvious alternative, and to also *record grasps and manipulation tasks with the Shadow hand* itself. The basic idea is to tele-operate the Shadow hand under full control of the experimenter, and to only record those grasps or manipulation sequences where the experimenter accepts them as "*human-like*". As the kinematics of the Shadow hand is known exactly, this approach completely bypasses any issues of calibration and augments our database with known finger positions and trajectories. Given a 3D-model of the target object, the known kinematics of the hand also allows us to reconstruct the contact points during the whole manipulation sequence.

Now the new problem arises on how to control the robot so that the resulting finger movements may be considered *human-like*. The original plan to provide a tactile-glove solution on the Shadow-hand has failed and had to be abandoned due to problems with the TekScan sensor [67]. Instead, we ended up on a rather straightforward tele-operation setup with the Shadow hand controlled by the CyberGlove sensor. The overall sensor setup is described in the next subsection 2.1, and some details about the software-architecture are given in subsection 2.2. The key to operating the hand under full human control is the mapping from the CyberGlove-II sensors to the Shadow hand; this is described in subsection 2.3.

This section concludes with the demonstration of a few complex grasps from the grasp-taxonomy in subsection 2.5 and a demonstration of in-hand manipulation 2.6 using the proposed glove-calibration system.

2.1 Experiment setup

As explained above, the direct analysis of human grasping motion has turned out to be more difficult than expected, not least of all due to the different hand-shapes of the different experimenters. One way to bypass this problem is to use the Shadow hand itself to perform the grasps, and this approach has been tried out during a recent joint recording campaign in Hamburg.

This subsection first describes the experiment setup currently installed at UHAM, see figure 1 for a photo. The setup consists of the Shadow C5 (air-muscle) Dextrous Hand mounted onto a Mitsubishi PA-106C robot arm, with the CyberGlove-II data-glove used for tele-operation of the hand. Several cameras, the Kinect, and a variety of other additional sensors are available:

- **Shadow C5 air-muscle-hand:** This version of the Shadow Dextrous hand shares the same geometry and kinematics as the C6-type motor hand, but the tendons are actuated by a pair of McKibben-style air-muscles for each controllable joint instead of electrical motors. The muscles are elastic and provide full passive compliance, resulting in good grasp stability for a large variety of static grasp poses, but the actuation is slower (and more noisy) than the electric motors on the C6 hand.

  For the experiments described in this report, we rely on the basic joint position control algorithms provided by the Shadow real-time (RTAI) software. In principle, separate PID-control parameters can be used for the two muscles in each antagonistic pair, allowing for stiffness control, but this has not been used so far. Our recording software connects to the Shadow RTAI drivers and collects the actual joint-angles, target joint-angles for position-control, available tactile-data, and muscle air-pressure for all joints. Joint-torque data is not available directly, but for static grasps it may be estimated indirectly for

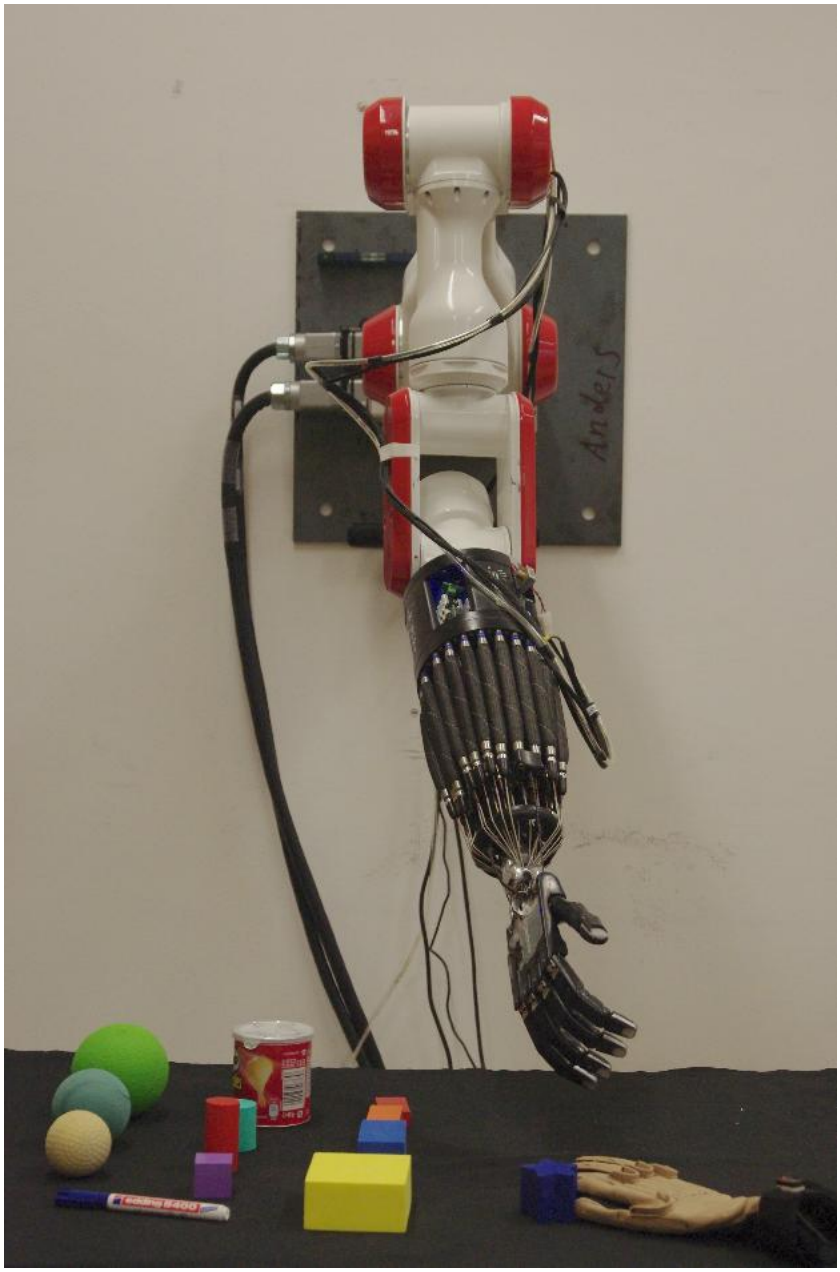


Figure 1: The experiment setup in Hamburg, with the Shadow air-muscle hand mounted on the PA-106C robot arm and tele-operated via the CyberGlove-II. The objects on the table are from the IST object set (compare section 3.1). A Kinect sensor is mounted below the robot base; the multi-camera system is not shown.

static grasps from the air-pressure in the muscles, the corresponding elasticity of the muscle, and the difference between target setpoint and actual joint-position.

Note that recording of the hand state is possible at full speed, with about 100 Hz sample-rate used for the experiments reported here. However, controlling the hand setpoints had to be slowed down to approximately 10 Hz, due to intermittent bugs in the Shadow control-software and firmware. Given the compliance and slow actuation of the air-muscles, this has not been a major obstacle for the experiments.

- **CyberGlove-II:** The Immersion CyberGlove-II data-glove is widely considered the best input-device for recording human hand pose. It provides 22-DOF, including 2-DOF for the wrist and 4-DOF of the thumb.

- **PA10-6C:** The Shadow hand is currently mounted onto a standard Mitsubishi PA10-6C robot arm. The arm has six rotational joints, has a total reach of approximately one meter, and a nominal payload of 10 kg. We bypass the Windows-based control-software provided by Mitsubishi and use the open-source RCCL robot control library [16] for FK/IK and trajectory execution under Linux. The robot is wall-mounted and has been calibrated to the workspace with an accuracy of about 1 mm.

- **PSeye camera system:** The vision system for the experiments consists of a total of three cameras, in a co-linear orientation which can be used as two stereo-camera pairs [90]. The Sony PSeye cameras provide RGB images of 640x480 pixels, at frame-rates of up to 60 fps. The price/performance ratio of these cameras is exceptional, and automatic white-balance and gain-control work well.

- **Kinect:** The *Kinect* sensor from Microsoft/PrimeSense has established itself as the de-facto standard device for the recording of high-quality 3D point-clouds, fused point-clouds and color-images, and high-level feature-detection like skeleting and human pose-detection. Our current setup includes a Kinect mounted onto the wall just below the base of the robot arm, so that the experiment table is within the best range of the Kinect. However, the arm and Shadow hand will obviously occlude the sensor during execution of most grasping task.

Pose reconstruction

The pose of the Shadow-hand in the workspace can be calculated with good accuracy from the recorded joint-angles of the robot arm simply by following the known kinematic chain. The main source of error is the noise and calibration of the wrist joint-angle sensor of the Shadow hand.

No separate sensor system is available in our setup to track the pose of the target object. Instead, we rely on pose-estimation and stereo-reconstruction from the vision systems. Experiments with marker-based tracking using the three-camera system have shown a position accuracy of roughly 1 mm after stereo-reconstruction [90].

A number of additional optional sensors are also available for our experiments:

- Force/torque sensor: We recently acquired an ATi Delta-45 force/torque sensor, to be installed between the PA10-6C arm and the base of the Shadow hand. This will provide force information during object handling and reduce the risk of collision damage. The sensor was not available in time for the experiments reported here.
- Shadow Instrumented Object: One *instrumented soda can* force-sensing object [62] has been provided by Shadow. This has not been used in the experiments reported here.
- Wiimote and Nunchuk: The Nintendo Wii remote controller ("Wiimote") has been selected as the target object for one of the project scenarios. Its internal accelerometer provides pose information and may be used to detect finger contacts and slippage. It will be used as an instrumented object, where the robot hand grasps, re-grasps, and re-orients the object, and where the thumb and fingers find and press the buttons.
- Other sensors: Similar to the Wiimote, the (very small) JoyWarrior accelerometer provides pose estimation. It will be used as an instrumented object for learning in-hand rotation tasks.

2.2 Software

All individual hardware devices from our setup are connected to two computers running Linux, with the first computer running a real-time kernel for controlling the Shadow hand, and the second system controlling the sensors and the robot arm. Most of the sensors use their own proprietary low-level data-format, but the required device drivers are readily available. The RCCL library for controlling the arm requires a special kernel module and connects to the robot via an optical Arcnet interface, but the driver code could be re-used from a previous project.

To simplify the recording of the tele-operation experiments, we developed a special software tool. It connects to all the sensors, either directly or communicating via TCP/IP with a corresponding small server, and provides a graphical user-interface that allows the user to set up, control, and analyse the experiments.

The software architecture of the tool is shown in figure 2, where the basic data-flow is from the sensors on the left to the actuators on the right. The joint-angle data from

the data-glove is first calibrated and mapped to the kinematics of the Shadow hand. This data is then sent via TCP/IP to either the GraspIt! simulator for visualization (top right) or via TCP/IP to the Shadow hand server (middle right), or both. Arm motion commands are also sent via TCP/IP to the arm-server.

Both the hand server and the arm server send their current state periodically back to the tool, where the data is timestamped and can be recorded together with the raw data from the glove and the vision sensors. For best compatibility with the existing human recordings from WP1, we decided to store all experiment data as XML-files in the format as specified in D4 [63]. The data corresponding to an experiment is

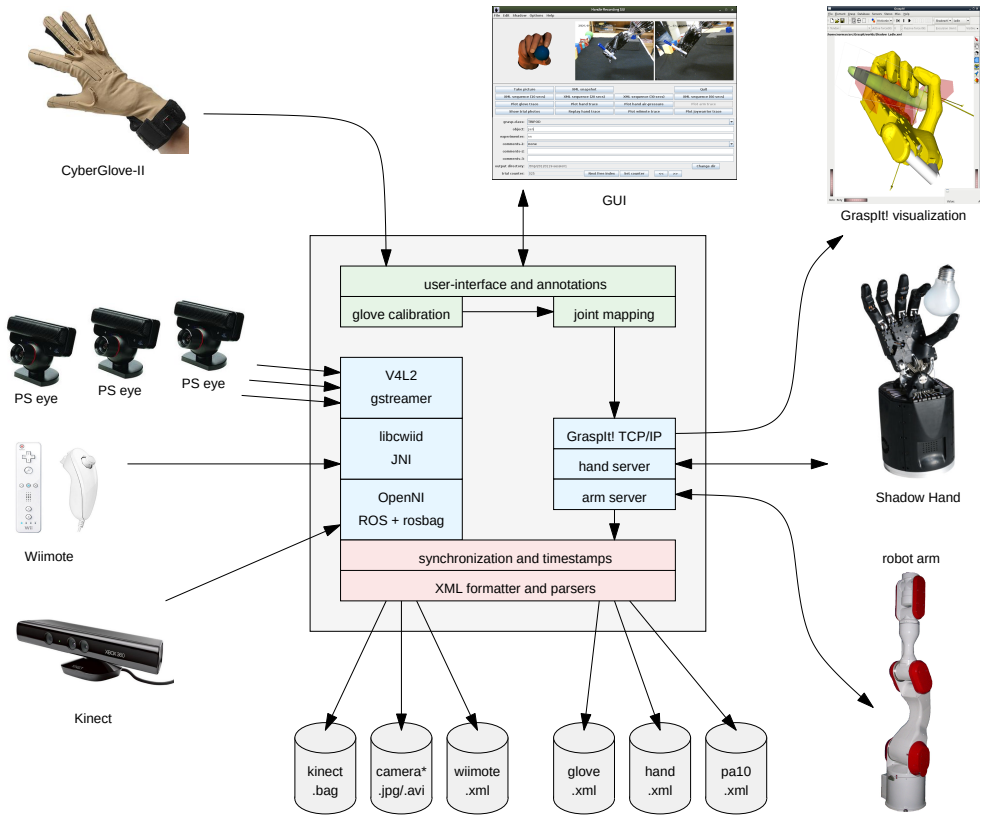


Figure 2: Software architecture of the recording tool. The main user-interface controls access to all functions and allows the experimenter to either save momentaneous snapshots or sequences of sensor-values — including data-glove pose, hand-state, arm-movements, images-and videos, point-clouds, and the data from a variety of other sensors. Incoming finger joint-angles from the data-glove are calibrated, mapped to the Shadow-hand kinematics, and forwarded to both the real Shadow hand and to the GraspIt! simulator for visualization. Recorded data is stored in XML-files compatible with the Handle database format [63, 68] or as ROSbag files.

stored as individual files, one per sensor or device, with a header including device information and all available calibration information, followed by the data section with the time-stamped raw data. Where required, the low-level drivers of the sensors were updated to provide their data in this format. Our Java library for analysis of the human-recordings (*hdbt.jar*) was updated with the corresponding XML formatters and parsers. The tool can be used standalone or from Matlab, and complements the existing tools and parsers from WP1. It is available on the project SVN server.

Figure 3 shows a screenshot of the user-interface. The top row shows the visualization of the grasp-class for the current grasp, plus images from the stereo cameras. The set of buttons below that is used for updating the camera images, taking a snapshot (single sample) of the current grasp and sensor data, or recording manipulation sequences. The recorded data can be plotted and the manipulation sequence can be played back on the hand immediately. The panel below includes user-controls and text-fields for annotation of the recorded sequences; the user selection is written into the metadata section of the recorded hand XML files.

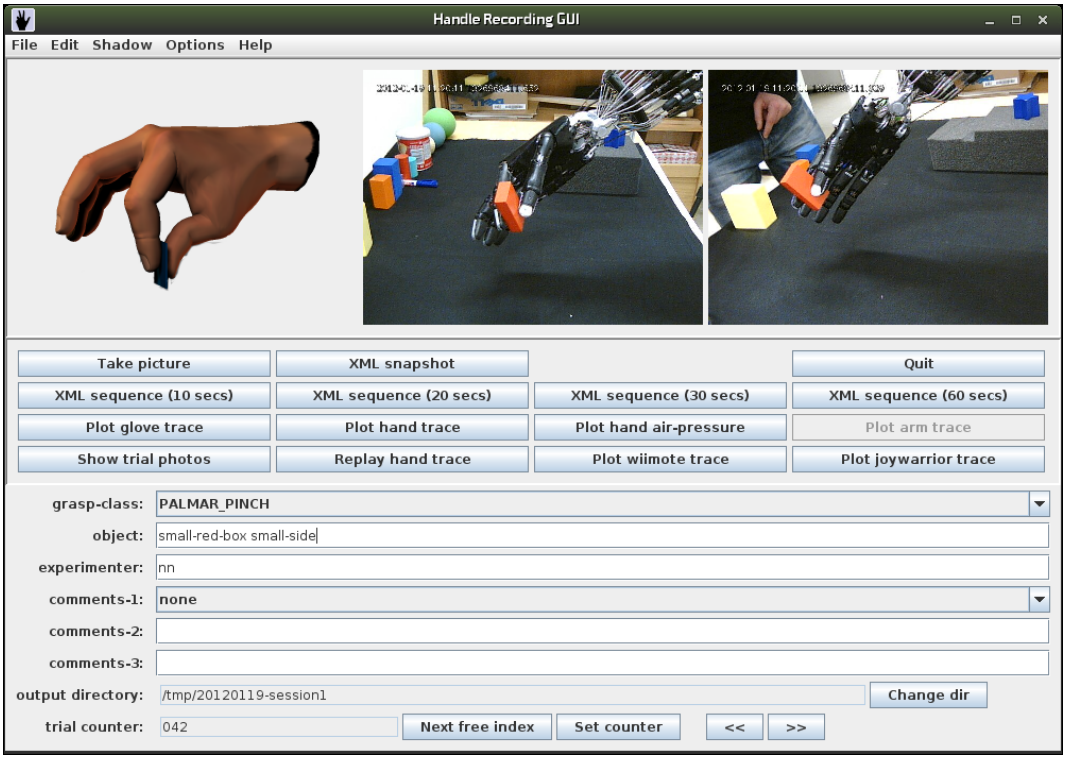


Figure 3: User-interface of the recording tool. The software connects to the Cyberglove and the Shadow hand, as well as additional sensors (e.g. robot-arm, instrumented objects, Wiimote).

2.3 Cyberglove calibration

The data-glove is the obvious sensor for recording the human hand posture. However, as no two human hands are the same, a calibration step is essential for any quantitative analysis. Even for a single test-person, a slightly different fit of the glove will result in differing measured joint angles between different experiment runs.

While an exact calibration might not be necessary for some data-glove applications like virtual reality, the extra accuracy wanted for robotics and tele-operation application has led to several approaches. The most basic algorithm performs a static mapping from the known range of the glove sensors to the joint-angle limits of the robot. A slightly better version records the actual minimum and maximum values of each glove sensor during a quick initial calibration, or updates those minimum and maximum values at runtime, and then maps to the known and constant joint-angles from the robot. This approach is currently used in the ROS stack provided by the Shadow robot company, and has the advantage that each robot joint can be moved through its full range, independent from the experimenter's hand shape. On the other end of the spectrum of available algorithms, very complex mappings with elaborate training sequences and complex algorithms to remove sensor-crosstalk have been proposed.

The approach proposed in this work targets the following goals:

- quick and easy calibration procedure
- calibration should not take more than a minute
- good match for several important hand-poses
- mapping the glove abduction sensors to the Shadow hand
- suitable mapping for the thumb to allow precision-grasping
- intuitive control of the hand after calibration

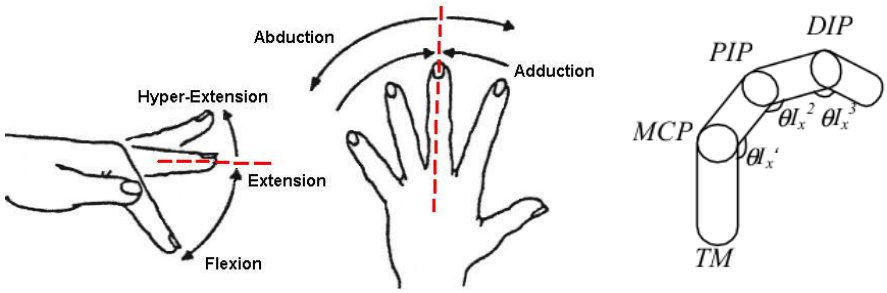


Figure 4: Finger movements: abduction/adduction, flexure/extension (left), Labeling of the finger phalanges and joints: metacarpal, proximal, medial, distal (right).

During calibration, the software developed in this work [89] presents images of a set of carefully selected hand poses to the user, and optionally demonstrates the hand poses on the Shadow hand itself. For each shown or demonstrated hand pose, the user is asked to shape her hand accordingly, and the joint-values from the glove are sampled and stored. Once all hand poses from the calibration set have been recorded, the software uses regression to calculate a mapping from the glove to the robot hand.

Mapping from the CyberGlove to the Shadow Hand

Due to the different sensor locations and sensor resolution, a mapping is required to translate from the CyberGlove sensor values to the corresponding Shadow hand joint angles. The goal here is to provide the user with an *intuitive control* that allows for the re-creation of important grasp poses with minimal effort and good precision.

The *CyberGlove-II* used for the experiments provides a total of 22 sensors, with three flexure sensors per finger, three abduction sensors placed between the fingers, and one palm-flexure sensor. Four sensors measure the thumb position, and two sensors are used for the wrist position (figure 5, left).

The kinematics structure of the *Shadow hand* is based on 24 joints, with four joints (abduction, proximal, medial, distal) per finger, five joints for the thumb, two degrees of freedom for the wrist, and one extra joint near the little finger that mimics palm-flexure movements. Note that the distal finger joints are underactuated from their medial joints, resulting in a total of 20 controllable-DOF (figure 5, right). Due to different designs of the tendon routing, the joint-coupling between the distal and medial phalanges differs between air-muscle (C5) and motor (C6) variants of the Shadow hand.

The mapping designed in this work [89] uses both 1:1 and 2:1 functions. The finger flexure sensors from the CyberGlove are directly translated into the joint-angles of the corresponding joints of the Shadow hand fingers. For other sensors, in particular the finger abduction, hand-flexure, and thumb positions, a 2:1 mapping from CyberGlove values to Shadow hand joints is used.

Software-Architecture and User-Interface

The software consists of several components in a client-server architecture. The main client module, written in Java, provides the user-interface, the calibration and mapping routines, and the hardware interface to the Cyberglove via a helper JNI library [87]. The application can be used in three different modes: standalone for the glove calibration and the recording of grasp experiments, connected to the GraspIt! simulator via a TCP interface, and connected to the real Shadow hand for real-time teleoperation.

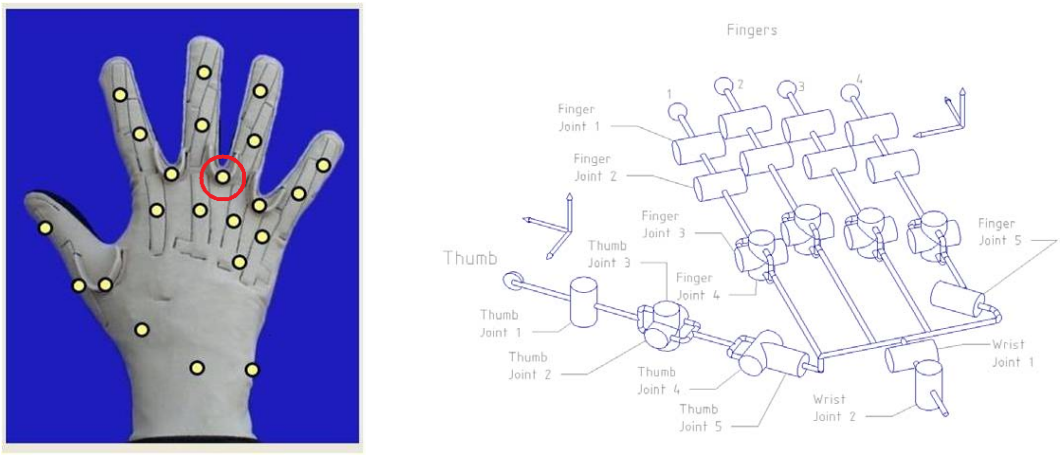


Figure 5: Locations of the 22 sensors on the CyberGlove-II (left), and kinematics structure of the Shadow Dextrous Hand (right).

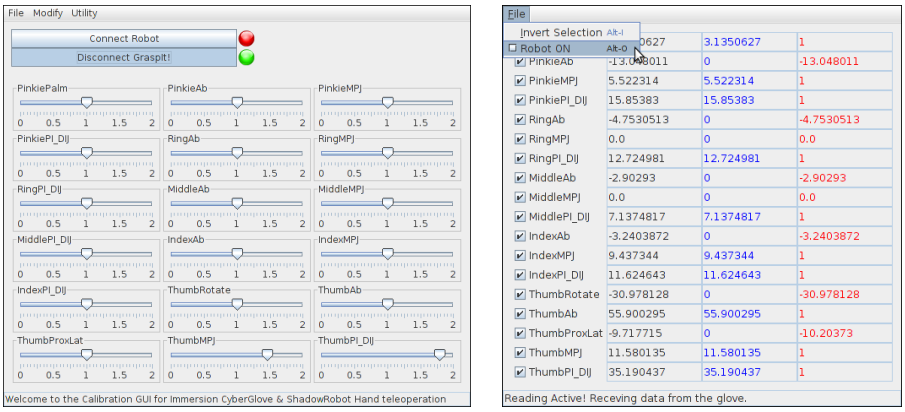


Figure 6: The user-interface of the glove-calibration tool; including the buttons for connecting to GraspIt! and the real robot, and the gain-sliders for online adjustment of the calibration (left). Hand-Server GUI window with menu, and three columns showing the target joint-angles, as well as the minimum and maximum angles for every joint. Joints can be enabled and disabled individually (right).

2.4 Calibration Sequence

The *interactive calibration sequence* uses a set of carefully *selected hand poses*, which are specified as joint-angles in an XML-file (*calibration_set.xml*). The user is then shown a picture of the pose, and given a few seconds to shape his hand accordingly. The resulting finger positions are recorded during an interval of about one second, averaging over multiple samples for suppression of noise and tiny erroneous hand-movements.

Once all poses have been shown, the software writes another XML-file (*calibrated_set.xml*) with the measured CyberGlove joint angles, and then immediately calculates the linear-regression coefficients for the mapping.

Note that the default poses include six different thumb orientations which cover almost the whole workspace of the thumb. Precision-grasps with the middle- and ring-finger also work well after calibration of the thumb in poses (4) and (6) above.

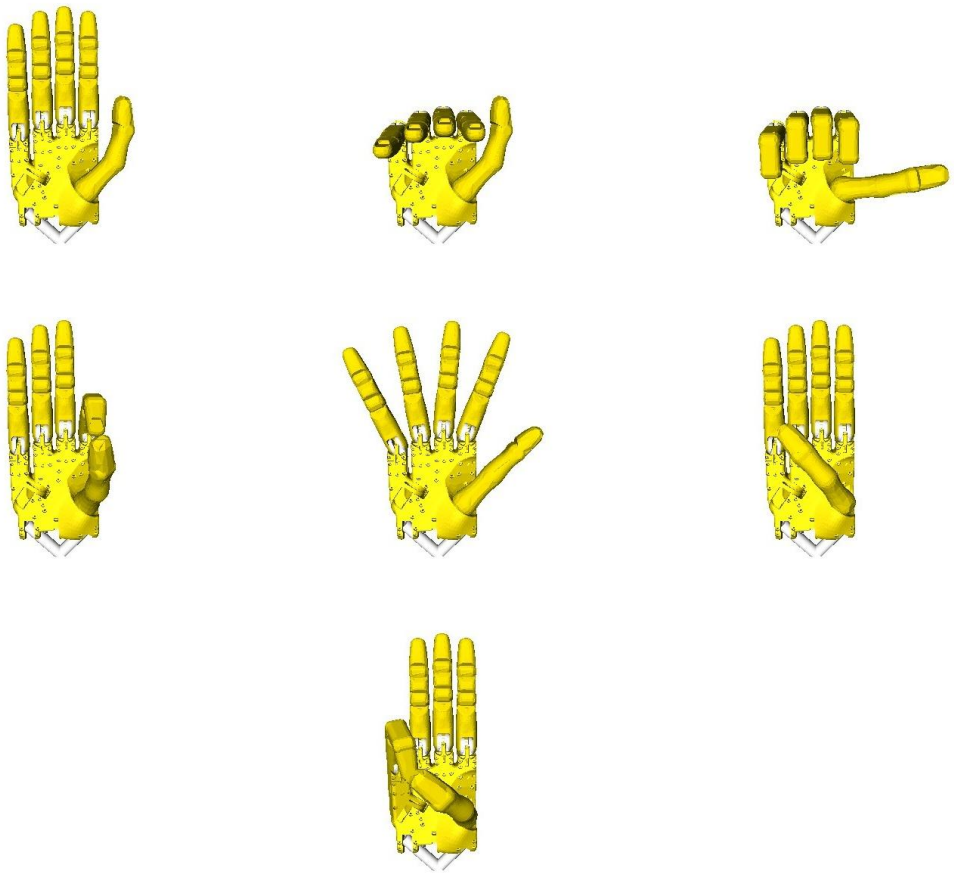


Figure 7: The seven hand poses currently used for the calibration sequence, as visualized by the Shadow hand model in GraspIt!. The set includes stretched and fully flexed fingers, finger abduction, and carefully selected thumb poses. The last pose is used to calibrate the palm-arch sensor and the extreme range of the thumb. The same thumb orientation is used in the first two poses to estimate the finger-flexure crosstalk on the thumb sensors.

2.5 Examples from the Grasp Taxonomy

To verify the viability of the data-glove calibration, we have first tried to perform all grasps listed in the GRASP taxonomy, as adopted by the project [74]. For most of the grasp classes and test objects, the control of the fingers and thumb was straightforward, and stable and robust grasps could be performed with the Shadow hand. However, the fine control of the thumb has room for improvement for some grasps and the experimenter may have to adjust thumb and fingers until reaching a suitable pose for the Shadow hand. The figure below shows a few typical examples.

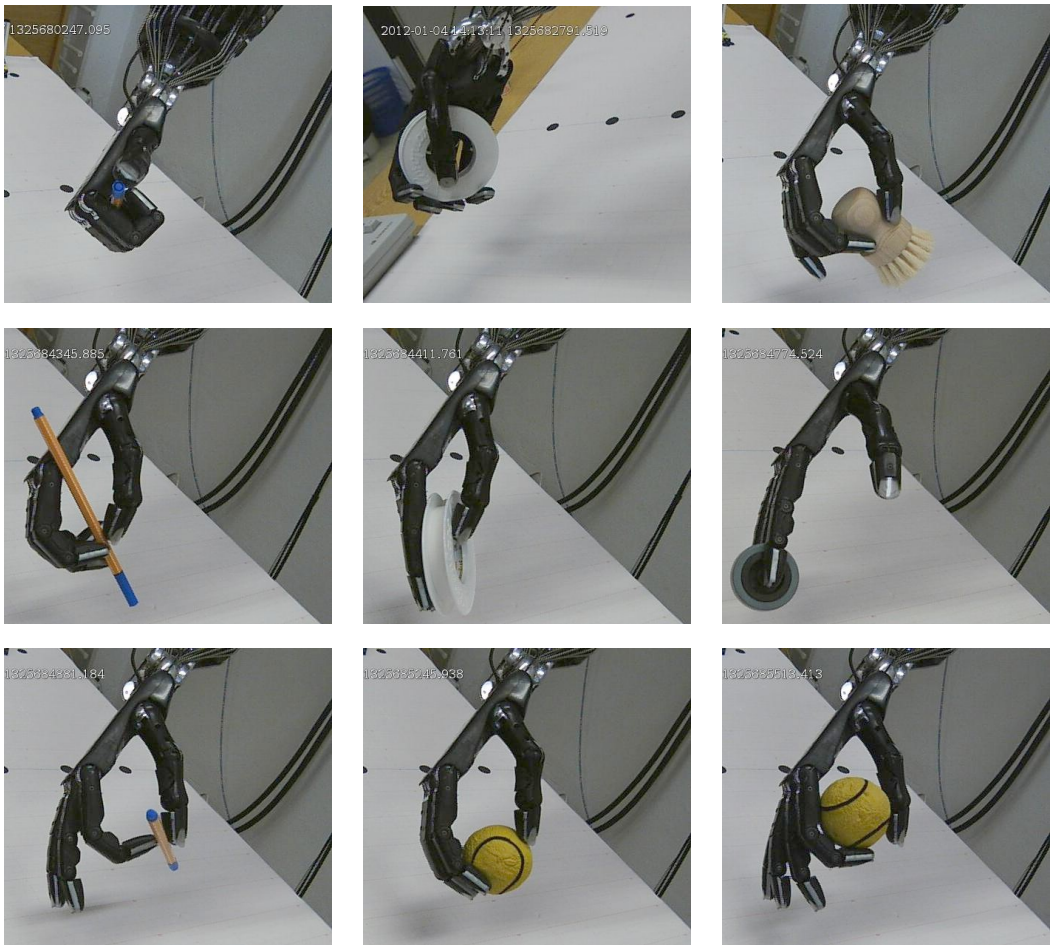


Figure 8: Example grasp-class poses performed under data-glove control: (a) light tool, (b) power disk, (c) small diameter, (d) writing tripod, (e) parallel extension, (f) addiction grip, (g) tip pinch, (h) quadpod, (i) ring.

2.6 In-hand rotation examples

Experience shows that experimenters are able to adapt their thumb movements in order to reach satisfactory Shadow hand shapes. One trick is to shape the thumb so as to approximately reach the target position with the robot hand, and then to twist the glove thumb slightly with the left hand in order to improve the target position. In any case, using the glove is much faster and much more intuitive than using the standard *joint-sliders* user-interface. The following images show a few snapshots from an in-hand rotation applied to a small wooden block from the *toy-sorting box* scenario, and trying to handle *scissors*.

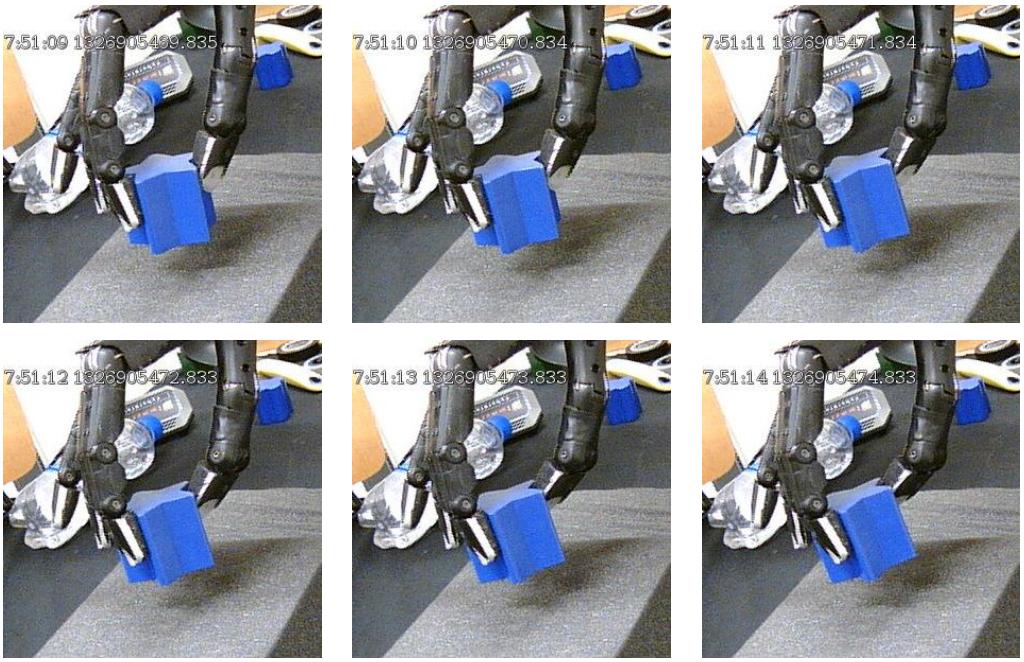


Figure 9: Examples of the tele-operation using the calibrated data-glove: in-hand rotation of a wooden block.

Figure 10: Examples of the tele-operation using the calibrated data-glove: scissors.

3 Parameterizing grasping motions

Grasp-planning for multi-fingered hands has been an area of intensive research ever since the first demonstration of the Utah hand [8]. The mathematical theory is well developed by now [3], and several approaches have been proposed [2, 4–6]. An extensive survey of grasp-planning algorithms was included in recent project deliverable D13 [70].

So far, most research still concentrates on the properties of static grasps, without consideration of the grasp-context or manipulation sequences. A typical grasp-planning goal is to find a finger pose and hand/object-contacts so that the stability of the grasp is maximized against disturbances, e.g. external forces applied to the grasped object. This criterion immediately leads to the further classification into form-closure grasps, where the fingers geometrically enclose the object, and force-closure grasps where contact-forces and friction are included in the calculation. The latter requires the estimation of the friction parameters between fingers and objects, and typically involves the calculation of friction-cones and wrenches for every hand/object contact. The concept of *independent contact regions* described in a previous deliverable [69] is one typical representation of this idea.

Not surprisingly, most work has assumed that the hand and fingers as well as the target objects can be modeled as rigid bodies. In this case, the outer surface of the hand and fingers can be specified by detailed 3D-meshes, and the position of the fingers in space can be calculated by following the kinematic chain from the robot-base through all links to the finger-tips. A similar calculation gives a detailed 3D-surface of the target-object in its current pose, and standard algorithms from computer graphics are used to find finger/object contact-points or their collisions. Efficient implementations of those algorithms are available, and several algorithms can also be performed on current graphics hardware, resulting in very good or even real-time performance. This approach is currently used in most robot- and hand-simulators [24, 84, 85].

Unfortunately, while the exact geometry of the robot actuator may be known and modeled with high accuracy, the target objects to be grasped are typically not known in advance, neither are their pose and attributes. This is a major drawback for the methods sketched above, because the search for good (or optimal) contact points has to be performed anew given only a rough estimate of the object size and shape, as provided by the robot sensors. Humans are known to perform tactile exploration of the target scene, and a recent work [91] has indeed coined the term *blind grasping* for tasks involving no pre-knowledge nor visual information about the manipulation object.

The HANDLE planning strategy

Of course, the scientific approach of project HANDLE has been designed to overcome this fundamental problem, and the key ideas were summarized in deliverable D13 [70]. The basic concept relies on and exploits the similarity of the structure and kinematics between the Shadow Dextrous Hand and the human hand. Given a human demonstration of any static grasp, there is a high probability of reproducing this very grasp with the Shadow hand successfully when using a suitable mapping of the human finger pose to the robot hand. The same holds for manipulation tasks, provided that every hand pose used by the human demonstrator can be matched to a corresponding pose of the robot hand.

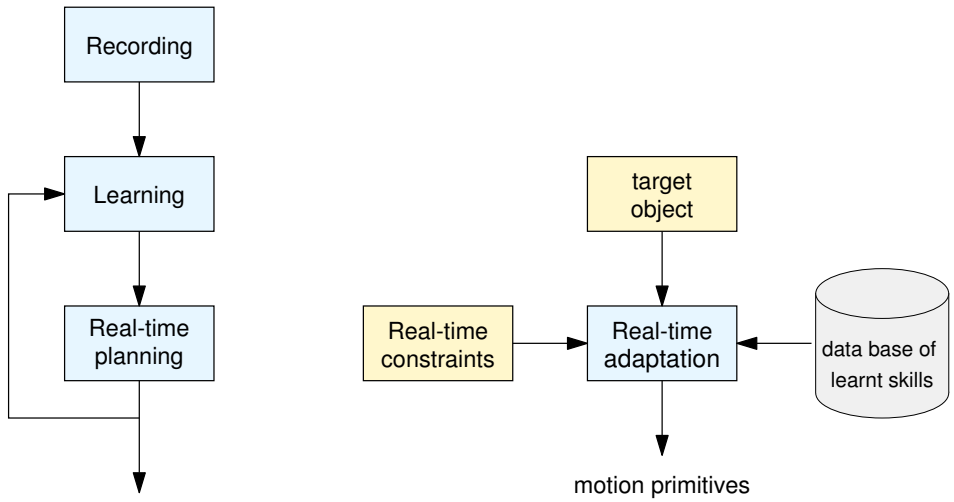


Figure 11: (left) The HANDLE global planning strategy, (right) real-time adaptation of learned skills under task-constraints. Redrawn after [70] (section III.1).

Based on the above assumption, the project proposed the following two-phase approach to grasp planning for anthropomorphic robot hands, see figure 11 and figure 12. The first phase involves the *off-line learning of human-like grasps* for a set of prototype objects, annotated with the grasp-type from the grasp taxonomy, task-context and any additional constraints. For every grasp, all relevant hand and grasp-related data is recorded, including the relative pose of hand and object, the finger joint-angles, tactile-sensor responses and estimated finger forces and torques. Additional hand-specific data (e.g. air-pressure on the Shadow C5 muscle-hand, or motor-torques and -temperatures on the C6 motor-hand) may be also included in the data when considered appropriate. The raw data-sets are preprocessed to reduce noise and eliminate outliers, and are then clustered based on the object and task-attributes. For the analysis presented in this report, we only consider the basic object-shape and object-diameter, but we expect to also model and include more

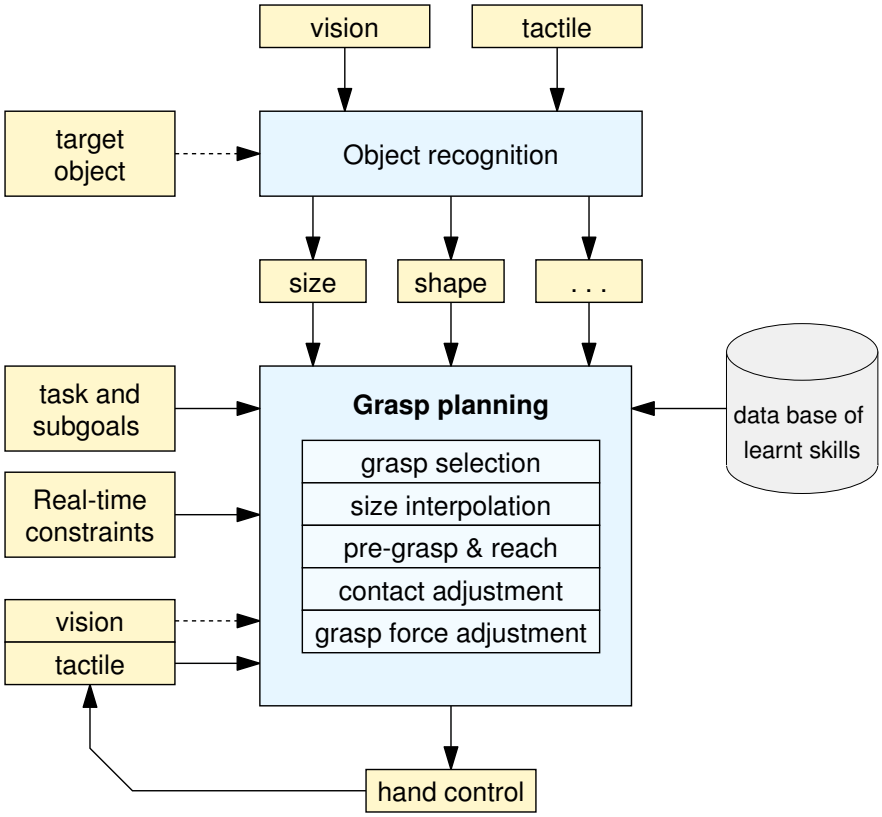


Figure 12: When facing a new task, the robot looks-up a prototype grasp matching the approximate object-type and -size, as well as any additional task-constraints. Initial hand-shape is derived from interpolation of the recorded prototypes, and grasping is performed using tactile-feedback and local finger-position adaptation using a suitable goal-function (e.g. independent contact regions).

complex object affordances in the future. Two typical examples would be to model and memorize the rotation axis of a bottle's screw cap or the offset from the grip to the actual tool-tip of a screwdriver.

When facing a new manipulation task, the robot tries to *recall and adapt a suitable grasp* from the learned human-like grasps. This involves the sequence of steps sketched in figure 12. We assume that the instructions given to the robot provide adequate task-context, and that the sensor information provides at least a rough estimate of the target objects and environment constraints (e.g., obstacles). The robot searches its database for grasps matching the estimated object shape, object size, and task-context and constraints. When several matches are found, the most suitable grasp-class is selected according to the immediate and future task-context, or according to user-specified goal-functions like trying to minimize arm and hand motion to conserve energy.

When a suitable grasp is found, the relative pose of the hand with respect to the (learned) prototype object is available. This is then checked with the robot-arm kinematics and obstacle-maps to execute a suitable reaching motion towards the target object. As soon as the hand is close to the expected object position, the fingers are flexed to reach the grasp configuration corresponding to the learned prototype grasp. In this pre-grasp phase, simple joint-level interpolation will be used to match the actual target object size as estimated by the sensors to the (known) size of the closest prototype objects. The actual grasp is then performed under close feedback from the tactile sensors. Obviously, the robot observes its own actions and will update its database according to the success (or failure) of the executed motion primitives.

It should be noted that the previous step is the essential key to reduce the computational complexity of the grasp-planning. Instead of having to fully search a high-dimensional (say, 30-DOF) space to find contact positions on the target object, the planner restricts itself to grasps known to be human-like. For the Shadow Hand grasping a rigid and static object, the full search space includes the extrinsic 6-DOF of the hand pose in 3D-space with respect to the object, plus the 20-DOF for the controllable finger joint angles, or 24-DOF when also counting the joint-angles of the underactuated (distal) finger joints.

The local search and adaptation of a previously learnt grasp reduces the search space dramatically, especially when also relying on object symmetry (or *key axis*) properties to approximate the approach and initial grasp positions. After the initial grasp, the hand-object contact positions still have to be optimized by a search of the hand state space, but this search is only local and simple gradient-descent algorithms can be expected to perform well without being stuck in local minima far from good global solutions.

3.1 The IST benchmark object set

It is natural to start the search for typical human-grasps with a simplified but typical set of benchmark objects, representing typical object shapes.

For the recordings reported here, we used a set of twelve prototype objects suggested by IST. See figure 13 for a photo of all objects together and refer to table 1 for the relevant object attributes and dimensions. The objects include three basic shapes (sphere, cylinder, box), different object diameters matched to typical human grasping tasks, and different materials (sponge, rubber, wood, metal).

Note that the same object set is used by IST for recording of grasps on the iCub robot hand. This offers the opportunity to compare the performance of the Shadow hand with a total of 24-DOF (20-DOF controllable) and the iCub hand with 20-DOF (9-DOF controllable).

Figure 13: The set of prototype objects suggested by IST. The objects include three basic shapes (sphere, cylinder, box), different object diameters matched to typical human grasping tasks, and different materials (sponge, rubber, wood, metal).

object name and grasp pose	width/mm	height/mm	length/mm	material
big green ball	**86**	**86**	**86**	sponge
medium green ball	**64**	**64**	**64**	rubber
small white ball	**54**	**54**	**54**	sponge
big red cylinder, top	**64**	76	76	metal
big red cylinder, side	64	**76**	**76**	metal
medium green cylinder, top	**38**	38	38	sponge
medium green cylinder, side	38	**38**	**38**	sponge
small red cylinder, top	**59**	27	27	wood
small red cylinder, side	59	**27**	**27**	wood
pen, side	150	**12**	**12**	metal
small purple cube	**30**	**30**	**30**	wood
large blue box, long side	**77**	39	39	sponge
large blue box, short side	77	**39**	**39**	sponge
medium orange box, long side	**60**	30	30	wood
medium orange box, short side	60	**30**	**30**	wood
small red box, long side	**60**	14	29	wood
small red box, short side	60	**14**	29	wood
small red box, medium side	60	14	**29**	wood
large yellow box, short side	80	80	**38**	sponge
large yellow box, long side	**80**	**80**	38	sponge

Table 1: Attributes of the prototype objects from the IST object set.

3.2 Example grasps and parameters

During the experiments, the objects were presented to the test-person in a fixed order (spheres first, then cylinders and boxes), but only for those grasps that were possible given the hand kinematics, or useful given the task. For example, the writing-tripod grasps were only recorded for the cylinders, and the addiction grasps only for the small objects.

For the first round of experiments, example grasps were recorded for eight grasp-classes:

- `LATERAL_TRIPOD`
- `PALMAR_PINCH`
- `TIP_PINCH`
- `LATERAL`
- `WRITING_TRIPOD`
- `ADDICTION_GRIP`
- `PARALLEL_EXTENSION`
- `TRIPOD`

The data shown in the following figures was recorded as soon as the test-person operating the hand announced that the current hand pose was a suitable grasp pose for the given object and task. The following figures present a first analysis of the recorded data, sorted by grasp-classes. A separate plot is shown for each of the fingers and the thumb, with different colors representing the finger-joints and different markers corresponding to the test-persons. While the joint-angle of the distal (underactuated) finger-joint has been recorded, its value only depends on the joint-angle of the medial finger-joint, and as such has been omitted from the figures in order to reduce visual clutter. (For the Shadow C5 muscle-hand, $\theta_{\mathrm{DIP}} \approx \theta_{\mathrm{PIP}} + 10^{\circ}$). The abbreviations are the same as used in the Shadow RTAI control software, namely:

- FF first-finger (index-finger)
- MF middle-finger
- RF ring-finger
- LF little-finger (pinkie)
- TH thumb

The color-scheme used is as follows:

- red FFJ2, MFJ2, RFJ2, LFJ2 (medial joint flexure)
- green FFJ3, MFJ3, RFJ3, LFJ3 (proximal joint flexure)
- blue FFJ4, MFJ4, RFJ4, LFJ4 (proximal joint abduction)
- black LFJ5 (palm-flexure)

Refer to figure 5 on page 14 for the sketch of the Shadow Dextrous Hand kinematics. For the thumb, all five joints are recorded, where THJ1 (red) refers to the distal phalange flexure, THJ2 (green) and THJ3 (blue) define the medial phalange flexure and abduction, while THJ4 (black) and THJ5 (magenta) represent the thumb rotation.

Note that the test-persons performed and requested several trials for some grasps; all raw values recorded are included in the following figures. Where necessary, the data will be sorted and post-processed manually for best significance of the clustering.

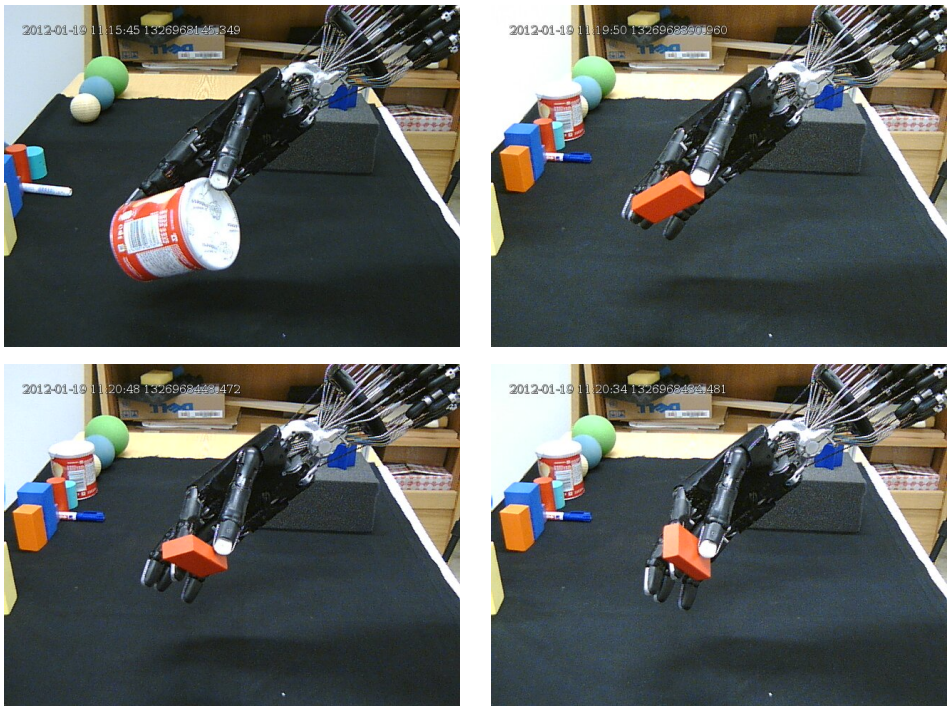


Figure 14: Examples of palmar pinch-grasps on large, medium, and small objects.

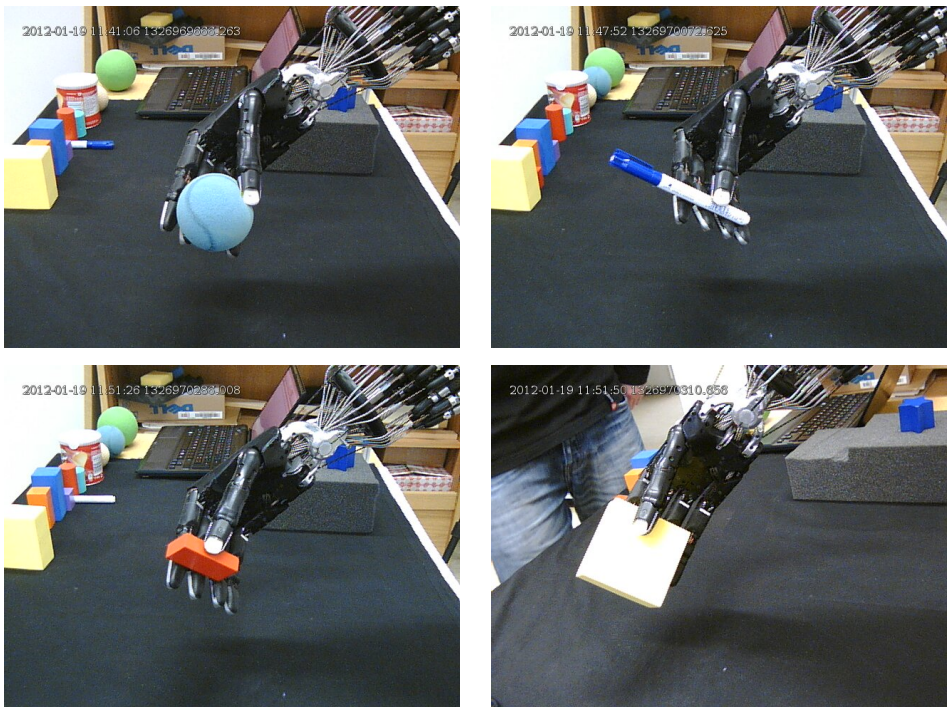


Figure 15: Examples of parallel-extension grasps. Note that the human-like grasps do not care about the location of the tactile sensors.

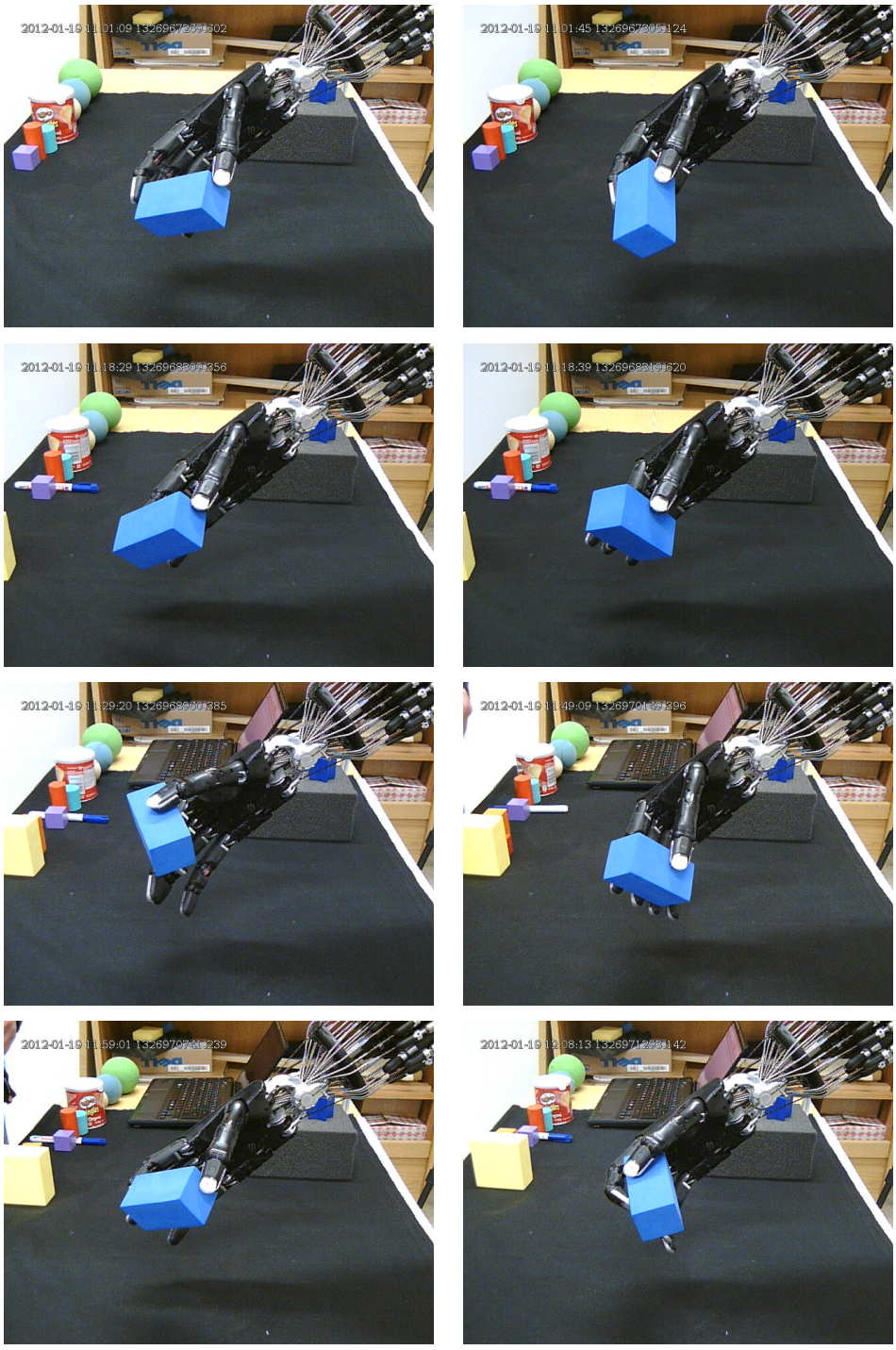


Figure 16: Example grasps performed on the *large blue box* object: (a) tripod, long side, (b) tripod, short side, (c) palmar-pinch, long side, (d) palmar-pinch, short side, (e) lateral, short side, (f) parallel extension, short side, (g) tip-pinch, long side, (h) lateral-tripod

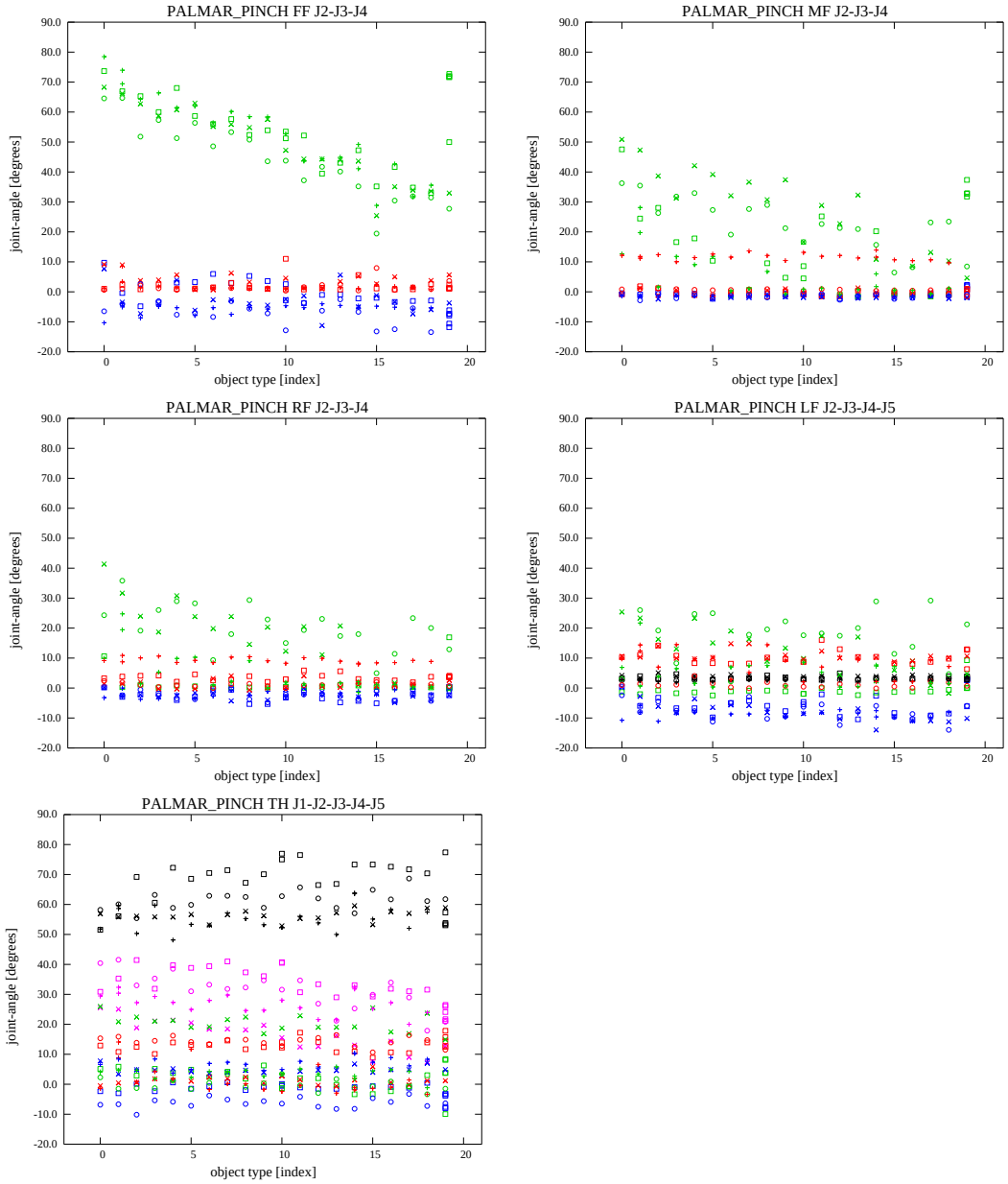


Figure 17: Palmar-pinch grasp for the different objects. The diagrams plot the joint-angles for the first-finger (FF), middle-finger (MF), ring-finger (RF), little-finger (LF) and thumb (TH). The colors encode the joints: medial (red), proximal (green), abduction (blue). Different marker symbols correspond to different experimenters.

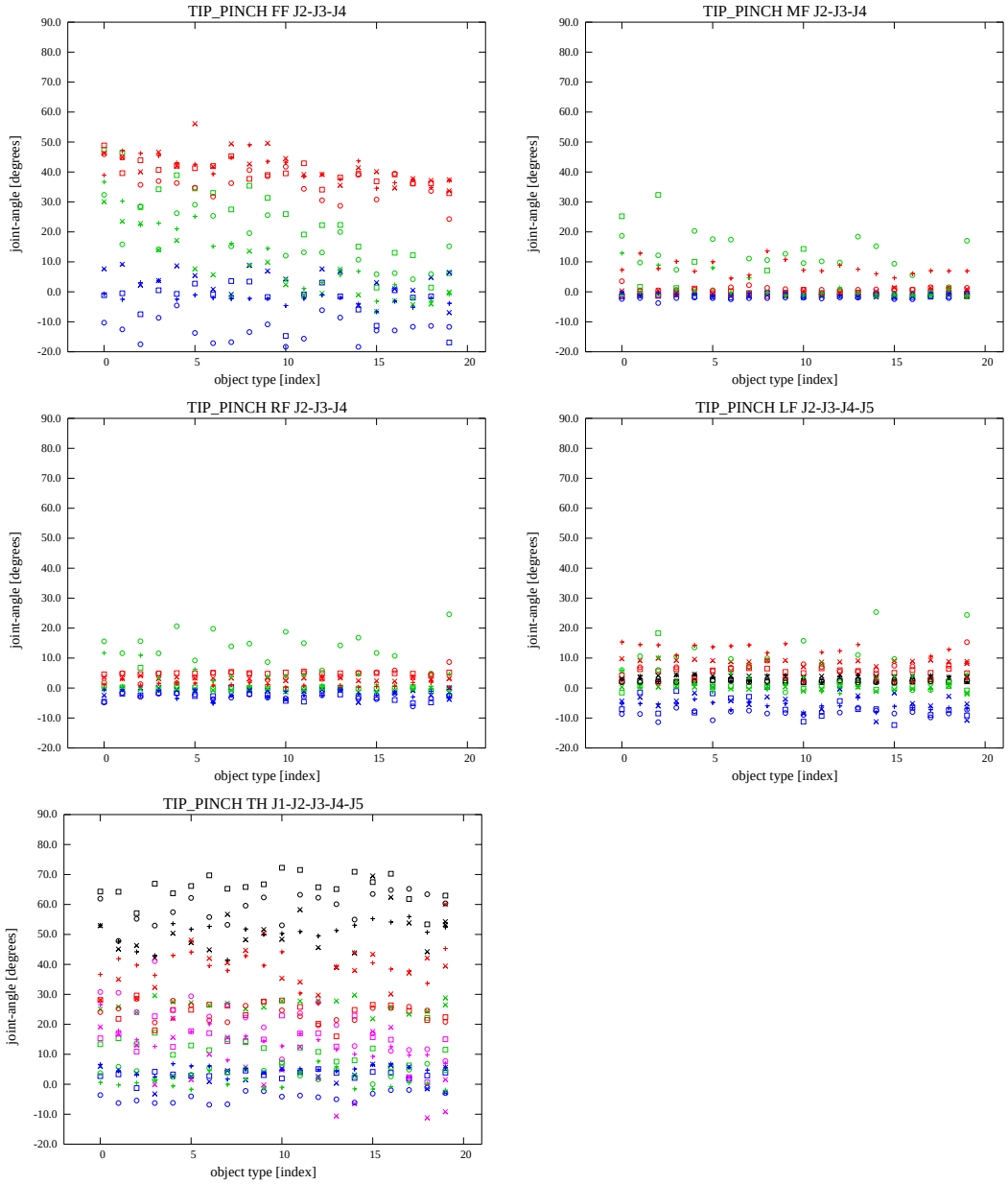


Figure 18: Tip-pinch grasp for the different objects. The diagrams plot the joint-angles for the first-finger (FF), middle-finger (MF), ring-finger (RF), little-finger (LF) and thumb (TH). The colors encode the joints: medial (red), proximal (green), abduction (blue). Different marker symbols correspond to different experimenters.

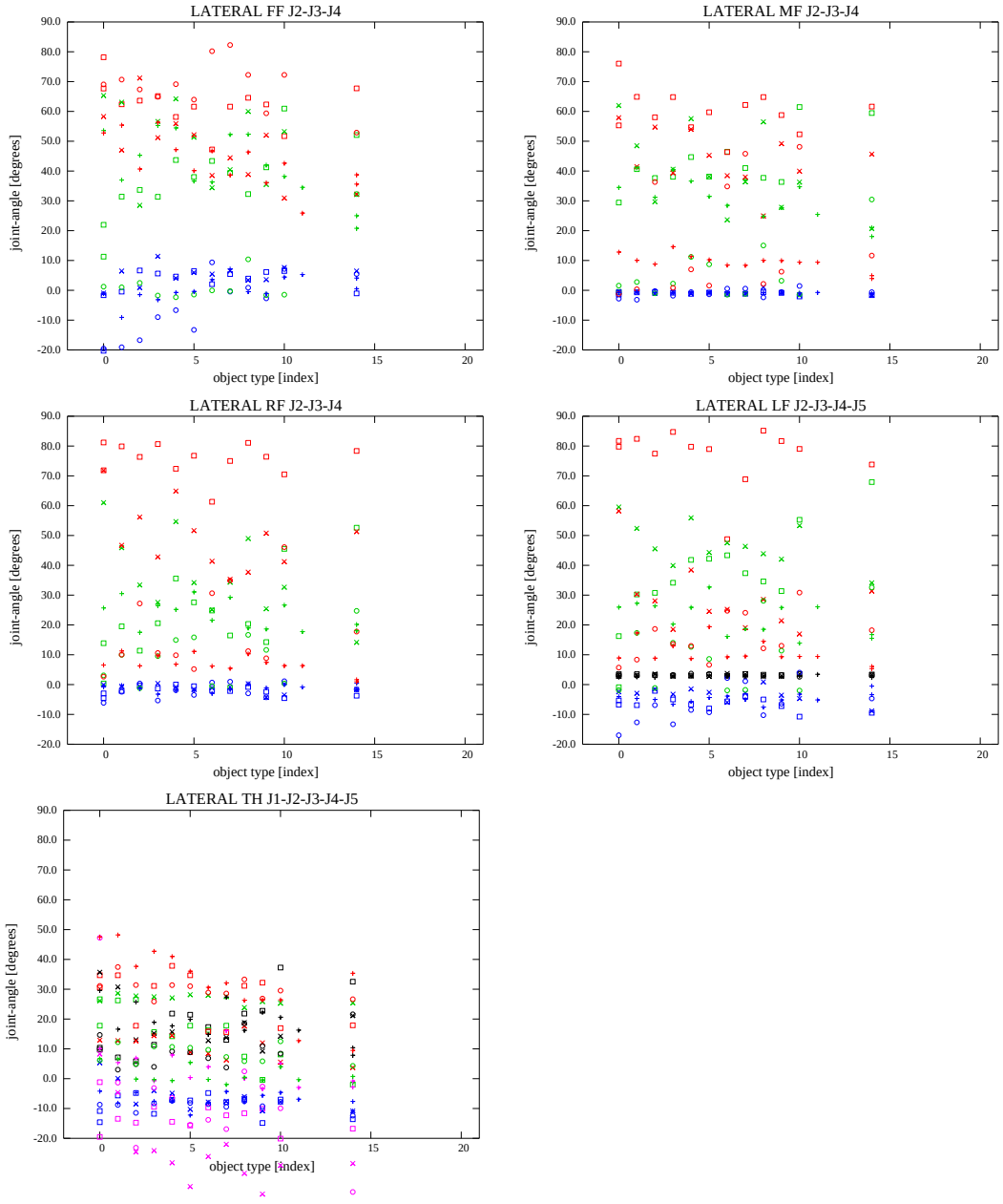


Figure 19: Lateral grasp for the different objects. The diagrams plot the joint-angles for the first-finger (FF), middle-finger (MF), ring-finger (RF), little-finger (LF) and thumb (TH). The colors encode the joints: medial (red), proximal (green), abduction (blue). Different marker symbols correspond to different experimenters.

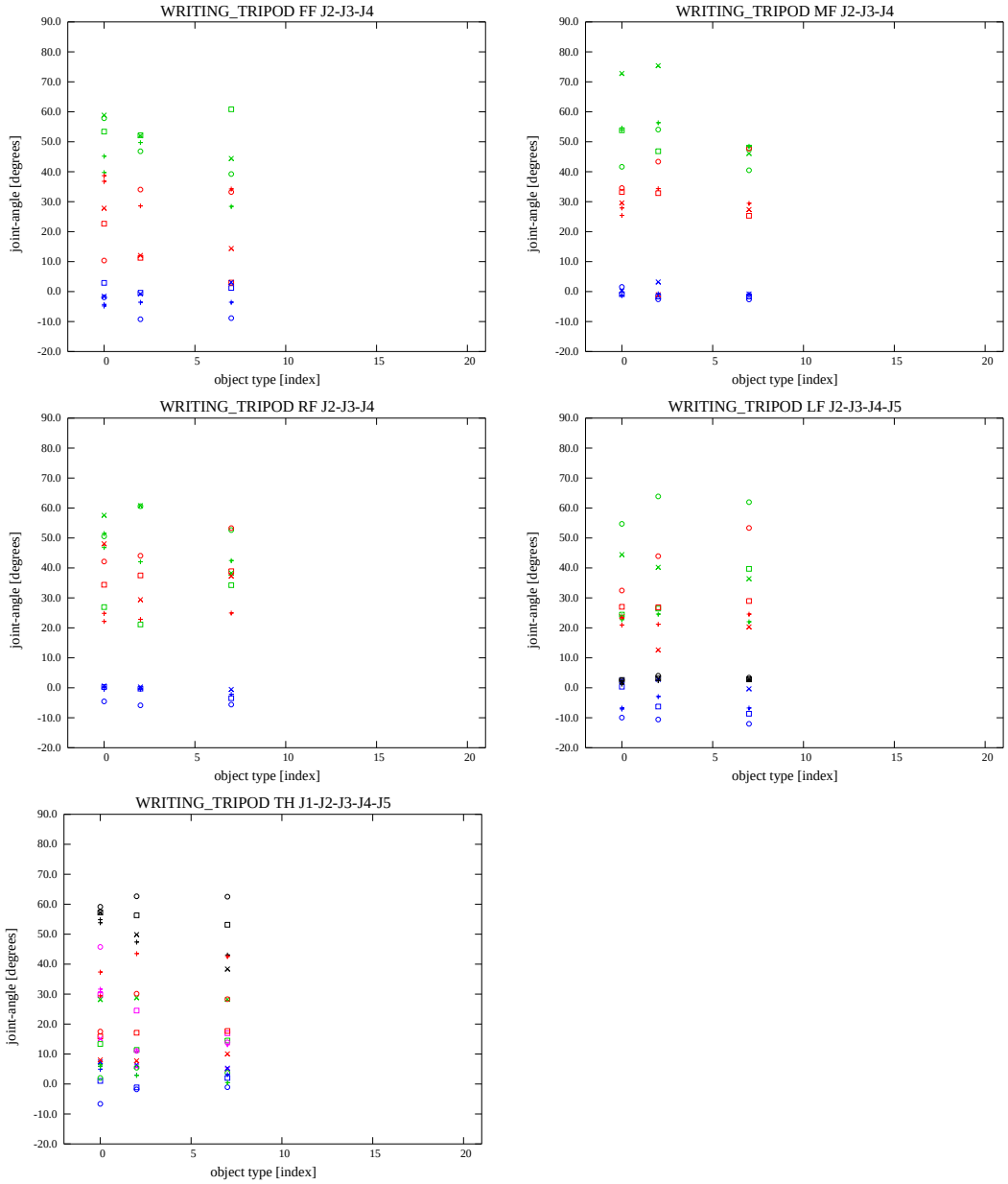


Figure 20: Writing-tripod grasp for the different objects. Note that the grasp is only useful for the cylinders from the object set. The diagrams plot the joint-angles for the first-finger (FF), middle-finger (MF), ring-finger (RF), little-finger (LF) and thumb (TH). The colors encode the joints: medial (red), proximal (green), abduction (blue). Different marker symbols correspond to different experimenters.

3.3 Lessons learned

The following list summarizes some of the insights gained during the tele-operation experiments, including feedback from the experimenters and some issues with the recording software and hardware. Even after glove-calibration, the mapping of the thumb movements was found to be acceptable only in parts of the thumb workspace. A more complex mapping might be required to improve the efficiency of recording a large number of grasps.

- **PST-type tactile sensors:** The current generation of PST-type tactile sensors provided by Shadow worked reliably during the whole recording sessions. This includes some software fixes required for the muscle-hand, which limited sensor sensitivity and stability in the previous version. The sensitivity and dynamics range of the sensors is now matched to typical grasping forces.

  However, due to the construction of the PST sensors, only contacts on the interior part of the fingertip result in a useful signal. For most of the recorded grasps, the human-like grasps include contacts on either the top or the side of the fingertips, and no tactile information is available for those contacts.

- **Lateral grasps:** The lateral- and lateral-tripod grasps were easy to perform, as the glove calibration worked reasonably well for the required thumb positions, and those grasps were recorded very quickly. The grasps were also surprisingly stable, due to the passive compliance of the muscle-hand.

  However, the contact points are on the side of the index finger for the lateral grasp, and on the side of the middle finger for the lateral tripod, where no tactile sensing is available on the muscle-hand. On the motor hand, the forces might be recorded by the tendon strain-gauges for the abduction joints (FFJ4 and MFJ4), but it remains to be checked whether the sensitivity is good enough to detect object contact and perform the grasps autonomously.

  For the lateral-tripod, the experimenters preferred a grasp where the object contact was either on the medial phalange of the middle finger or on the distal (PIP) finger joint. The grasp would also be stable when resting the object on the distal phalange of the middle finger, but humans seem to avoid this due to the extra flexure of the middle finger. For replaying the grasps on the Paris demonstrator hand, we might want to change the grasp pose to rest the object on the distal phalange, in order to use the tactile sensor of the middle finger.

- **Addiction grip:** As mentioned above, addiction grasps were only recorded with the three cylindrical objects from the IST object set. The Cyberglove provides three abduction sensors, and the calibration sequence includes a step to calibrate these. Therefore, it was expected that this grasp would be easy to perform, and in fact the experimenters reached a suitable initial grasp position very quickly.

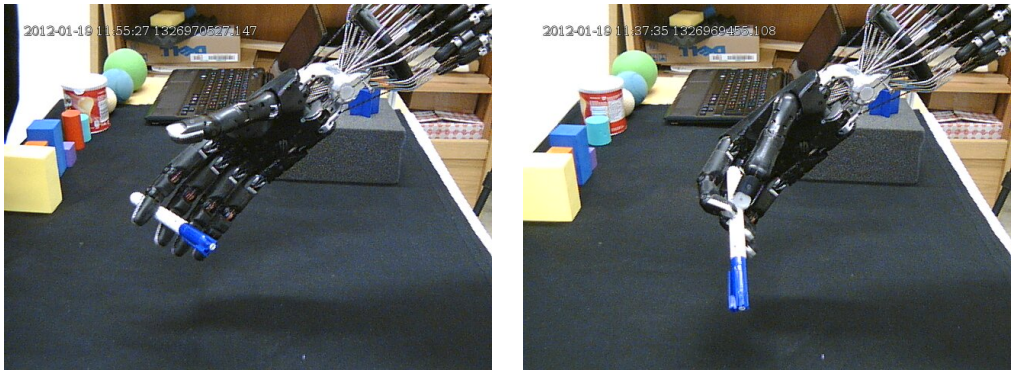


Figure 21: Addiction grip and writing tripod for the *pen* object.

However, it turned out to be very difficult to maintain a stable grasp. Lacking tactile feedback the experimenters had to close the fingers until stabilizing the target object, resulting in fully pressurized air-muscles for the adduction joints (FFJ4 and MFJ4). At the same time, the finger flexure was held mostly constant, and the hand controller had ample time to reach the target position for those joints, with only partially pressurized muscles for the flexure joints. In this situation, the proximal joints (FFJ3 and MFJ3) are less stiff than the adduction joints and tended to give way a bit, resulting in rotation of the object between the fingers and even losing contact in several trials. As the Shadow hand uses the same mechanism for the abduction joint on all fingers, similar results can be expected when performing the addiction grip with the combination of middle and ring finger, or ring and little finger.

It is expected that the addiction grip will be easier to perform on the Shadow C6 hand, even without tactile feedback, because the FFJ3 and MFJ3 ext/flex motions are controlled by a single motor with fixed stiffness.

- **Writing tripod:** The writing tripod grasp turned out to be surprisingly difficult for the pen from the object set, despite the rather large diameter of the pen used (12 mm). The main problem here is the slightly conical form of the fingertips on the Shadow hand, so that a gap remains between neighboring fingers even when the fingers are held fully parallel without abduction. Also the actual tip and the sides of the fingertips provide much less grip than the (rubberized) interior part with the PST sensor.

  The new fingertips on the Paris demonstrator hand use an ellipsoid cover on top of the ATi force/torque sensor, and as such have a different geometry than the previous generations of the fingertips. This difference will have little impact on most of the other grasps, but we plan to re-record the writing-tripod grasps on the Paris hand with pens of different sizes (including small ones) and compare the results.

- **Tripod grasps:** These grasps took the longest to record, due to two problems. Firstly, the default mapping of the thumb motion after the glove-calibration did not work very well, and a manual adjustment of the gain parameters was needed. Secondly, the experimenters tried to achieve an exact opposition between the fingers and the thumb, as this is the most human-like way to perform the grasp. Unfortunately, this is impossible due to the thumb-kinematics of the Shadow hand, where the thumb is oriented sideways towards the fingers for the tripod grasp poses. This again poses a problem for the PST-type tactile sensor, which is not activated on the thumb for the tripod-grasps.

4 Grasp synergies

The development of control algorithms for grasping and manipulation with complex multi-finger robot hands is still in its infancy, and is considered one of the hardest problems in robotics due to both the high-dimensionality of the state-space of the hand itself and the large variety of manipulation tasks. In their first paper on Eigengrasp planning [22], Ciocarlie et. al. noted that "it would be natural to draw inspiration not only from the hardware of the human hand, but also from the software; the way the hand is controlled by the brain". The authors continue to explain that the majority of human hand motions do not make use of highly flexible hand postures as would be possible mechanically, but that most motions indeed lack individuation in finger movements.

This insight is based on the classical and elegant study [20], where Santello et. al. asked several test subjects to shape their hands as if to manipulate imaginary everyday objects, and the hand poses were recorded with a data-glove. The study demonstrated that the fingers were shaped using certain patterns, despite of inter-subject variations. A Principal Component Analysis of the recorded data showed that the two first principal components accounted for more than 80% of the variance, strongly suggesting that the grasp postures used by the humans could be approximated by a 2-dimensional basis instead of the 22-dimensional basis required to describe all 22-DOF typically assigned to the human hand. This fact is also reflected in the classical grasp taxonomies [5], where only a handful of different poses are sufficient to explain the hand motions used by humans for grasping.

In this section we explore the use of grasp-synergies for planning and executing grasps with the Shadow hand. See the earlier project deliverable D13 [70] (chapter V) for a full introduction into the concepts including a review of the relevant literature from neuroscience and robotics. When we started searching for a suitable tool about two years ago, we decided to use the GraspIt! simulator [24] from Columbia University in lack of viable alternatives, and subsection 4.1 summarizes the key features of the simulator. Next, we sketch the key ideas of the Eigengrasp-planner in GraspIt! in subsection 4.2 and then show our initial approximation of Eigengrasps for the Shadow hand. We then demonstrate a few planned grasps for objects of complex shape and the planning of precision grasps.

Finally, subsection 4.4 shows the initial results of our calculation of the synergy matrix for the Shadow hand from the experiments reported in the previous section.

4.1 Grasp planning using the GraspIt! simulator

Given the kinematics of a robot hand and polygon models of the hand, fingers, and grasped objects, it is straightforward to build a simulation of finger and object contacts using basic algorithms developed for 3D computer-graphics. After selection

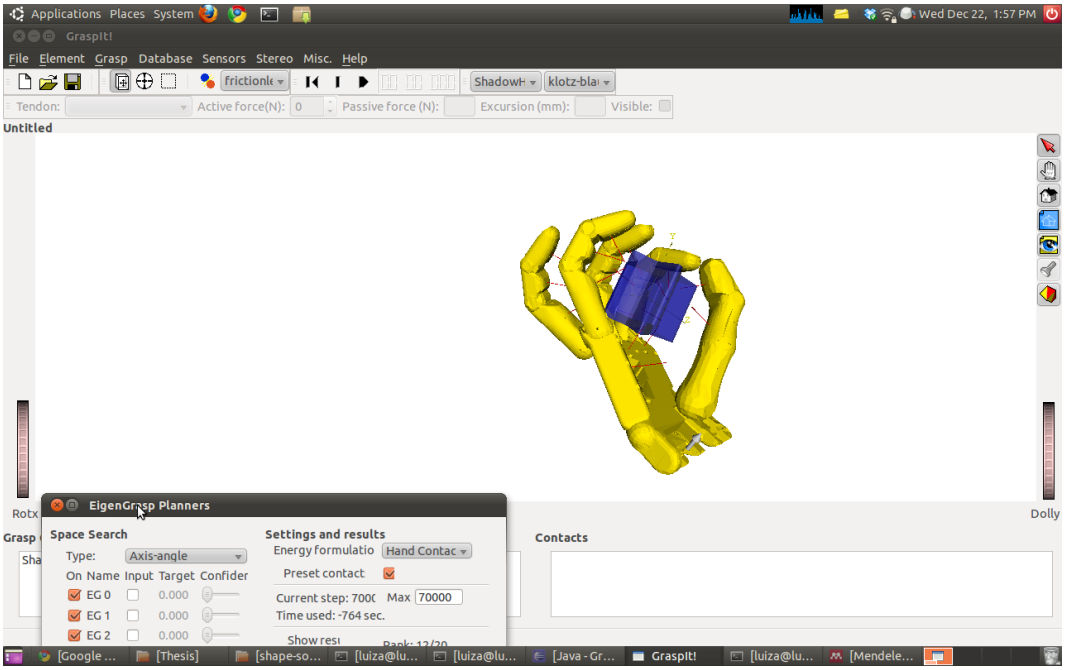


Figure 22: The user-interface of the GraspIt! simulator and the Eigengrasp planner.

of the relative pose of the hand and object, the fingers can be moved until making contact with the object, and the resulting grasp can be analyzed for form-closure. When friction-parameters are included, it becomes possible to also approximate the friction-cone for each contact and to derive the force-closure stability of the configuration. Developed by Andrew Miller at Columbia University, GraspIt! [24, 83] is probably the best-known simulator of this kind.

The software is written in C/C++, is readily available to interested parties, and can be compiled and run on either Windows or Linux. The user-interface provides a 3D-view of the hand and objects, and can visualize contact-points and friction-cones. Optionally, the simulator can be controlled via a built-in small TCP/IP server, which allowed us to integrate it as the visualization component in our recording software (compare 2.2 above). See figure 22 for a screenshot of the GraspIt! user-interface, showing the model of the Shadow hand and the star-like wooden block used as a demonstration object throughout this report.

The simulator is optimized for the simulation of grasps and includes 3D models of several well-known robotic hands, plus a simulation model of a 20-DOF humanoid hand. GraspIt! includes evaluation functions to calculate or estimate grasp-stability and provides several grasp planning algorithms. It also provides support for simulation of underactuated hand mechanisms, which of course is required to model the distal finger joints of the Shadow hand.

However, it should be noted that GraspIt! does not include a full physics-engine, and as such cannot simulate any dynamic behaviour during the grasping. All algorithms

expect that the kinematic chains of the simulated hand start with the palm of the hand, and it is not possible to include a wrist of a full robot system consisting of robot arm and hand.

The software distribution of GraspIt! includes several robot models, among them a simplified Shadow hand. This model corresponds to an older design of the Shadow hand, and is not suitable for simulation of the current C5/C6 (air-muscle/motor) Dextrous Hands. We updated the 3D-model and kinematics, and adjusted the range of the joint-angles to match the limits of the real robot. The sign of the abduction motion for the ring and little finger is still different between the simulation model and the real hand, but this is taken care of in our recording software instead. The figures in this section show different poses of this updated Shadow hand model.

4.2 Eigengrasps for the Shadow hand

For a robot hand with fixed kinematics, any hand pose is fully specified by its joint values, and can be described as a point in a high-dimensional joint space. For d dimensions, the hand-pose p is then given by

$$p = [\theta_1 \theta_2 \dots \theta_d] \in \mathcal{R}^d$$

where θ_i is the current value of the i-th degree of freedom.

Sampling a large set of suitable (e.g. human-like) hand poses and performing the principal component analysis, the resulting set of eigenvectors provide a new basis of the hand joint space. In D13 [70] (section 3.3, equation 1) the set of eigenvectors is called the synergies matrix.

Each eigenvector, also called an *eigengrasp* in [22,23]is a d-dimensional vector and can be thought of as a motion in joint-space. Motion along an eigengrasp direction will typically imply a motion along all degrees of freedom of the hand in joint-space,

$$e_i = [e_{i,1} e_{i,2} \dots e_{i,d}]$$

A linear combination of b eigengrasps with weights $\{a_1, a_2, \dots a_b\}$ specifies a hand posture p as a point in the b-dimensional subspace spanned by the selected eigengrasps,

$$p = \sum_{i=1}^{b} a_i e_i$$

Now the key question is how many eigengrasps are needed to closely approximate a set of target hand poses, the number of *effective degrees of freedom*. The data from the original Santello et. al. paper suggests that keeping just the two most important eigenvectors would be sufficient to approximate about 80% of the grasping poses

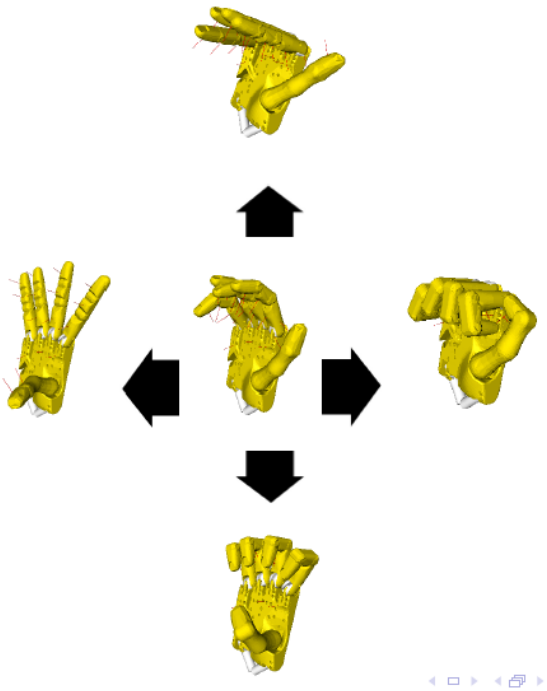


Figure 23: Neutral hand pose and example poses resulting from the first (horizontal) and second (vertical) Eigengrasps.

for everyday objects. Using additional eigengrasps improves the accuracy of the grasps and results in acceptable hand-shapes for all the other objects tested in the study.

The basic idea of grasp-planning based on synergies is to combine a quick search of the reduced subspace spanned by the relevant eigengrasps with a later adjustment phase. This can also be thought of as a hierarchical approach, where the synergies pre-shape the hand with approximate finger positions around the object. The following refinement phase closes the fingers until contact with the object is established, and then adjusts the contact positions and finger-forces to stabilize the object.

This approach has been implemented in the GraspIt! simulator and is called the *Eigengrasp-planner*. Here the grasp planning task is mapped to an optimization problem in the $(6+b)$-dimensional space spanned by the 6-DOF required to specify the relative pose between hand (palm) and the target object plus the number b of eigengrasps included in the search. For best efficiency, $b = 2$ is recommended and demonstrated in [22], but it might be necessary to include additional degrees of freedom to generate specific grasps.

The question now remains on how to establish a suitable set of eigengrasps for the Shadow hand. The results shown in the next two subsections are based on a very simple approach: an empirical mapping of the existing eigengrasp vectors available

in GraspIt! for the model of the 20-DOF humanoid hand. As stated in [22], those eigengrasps are the result of an empirical mapping of the original Santello data in turn. See figure 23 for a visualization of the neutral hand pose and two example poses in each eigengrasp direction.

4.3 Grasping complex objects

As a first test, we have run the Eigengrasp planner on several prototype objects from the Columbia Grasp Database [25], which includes a large collection of 3D-objects. Every object is available in four different sizes, where very large or very small objects have been prescaled additionally to make them graspable.

See figure 24 (upper two rows) for some typical examples. The grasp-planner has no notion of object affordances and will search the state-space at random, until converging on one of the best grasps found earlier during its execution. As will be obvious from the generated grasps, the goal-function used to guide the planning process strongly favors power-grasps, as they typically also involve form-closure and provide a high grasp stability.

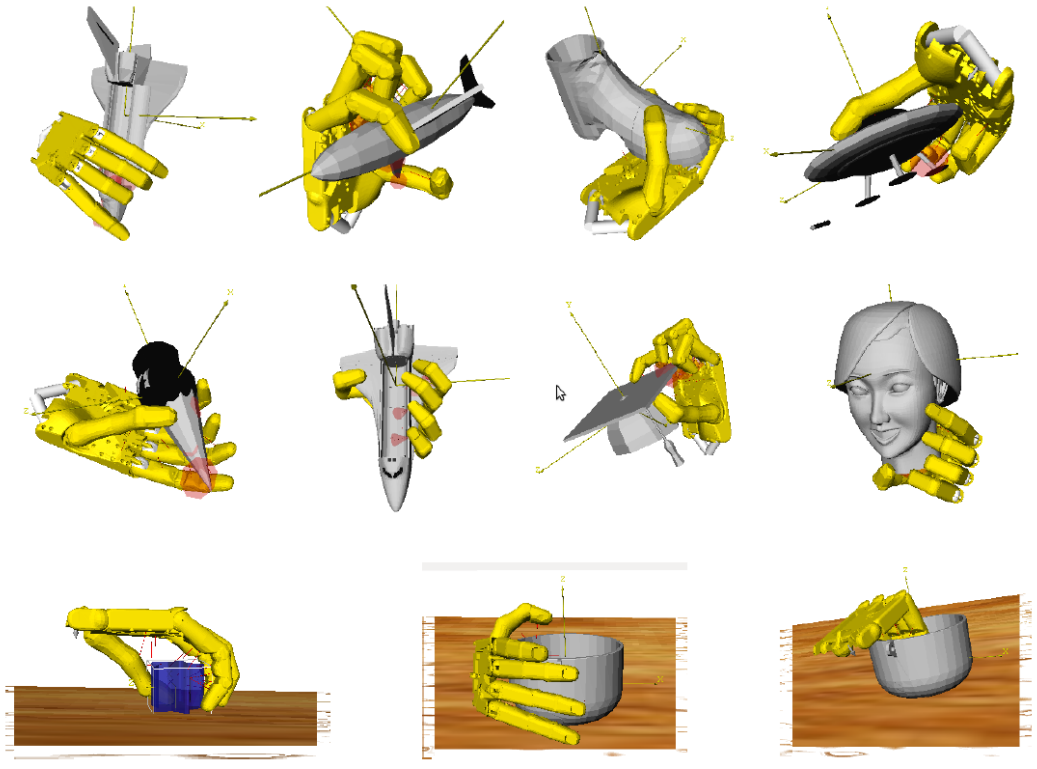


Figure 24: Examples of running the Eigengrasp planner on complex objects selected from the Columbia Grasp Database.

4.4 Precision-grasp experiments

When run with its default parameters, the Eigengrasp-planner is almost certain to converge onto power-grasps, as these maximize the grasp-stability. Manipulation tasks, on the other hand, require precision grasps which allow the object to be re-oriented by the fingers.

To achieve precision grasps with the fingertips only, we used the option to prepare another configuration file that lists the *virtual contact locations* on the hand. This specifies the 3D locations on the hand and fingers that the planner should bring into contact with the object. Where necessary, we added obstacle objects (the table) into the simulation to enforce the planner to converge onto precision grasps. See figure 24 (bottom row) for this approach.

4.5 Synergies derived from the grasp experiments

For the examples presented above, we used the Eigengrasp planner with the empirically mapped set of approximated global synergies from GraspIt!. This works somehow, but the resulting hand poses don't fully exploit the range of motions possible with the Shadow hand. Also note that the original statistical analysis from Santello et. al. targeted static grasps and was based on a single global set of synergies extracted from many different hand poses.

Therefore, we tried to derive human-like synergies matched to the Shadow hand by following the approach described in the previous chapter. It is obvious that separate sets of synergies extracted from hand poses for a specific grasp type will be much better suited to applications in fine manipulation.

In this section, we present first results of our analysis of the experiments reported in chapter 3. The analyzed dataset consisted in recordings of the Shadow Hand joint angles remotely controlled from 4 subjects to execute 8 different grasp types on 20 different object configurations. A total of 438 samples were acquired (not all grasp types can be applied to all objects and some subjects missed or repeated some grasps). These samples were split into data matrices A_i each corresponding to a different grasp type g_i and containing in each row the vector of N joint angles acquired in each trial.

A Principal Component Analysis was performed on each grasp matrix. The hand poses corresponding to the origin of the eigenspace for the eight grasp-classes studied here are shown in figure 25. Those poses represent the mean joint-angles recorded for the specific grasp-class when grasping all objects compatible with the grasp-class.

The amount of variance explained by the first N principal components is presented in figure 26. We can see that most of the variance is concentrated in the first few

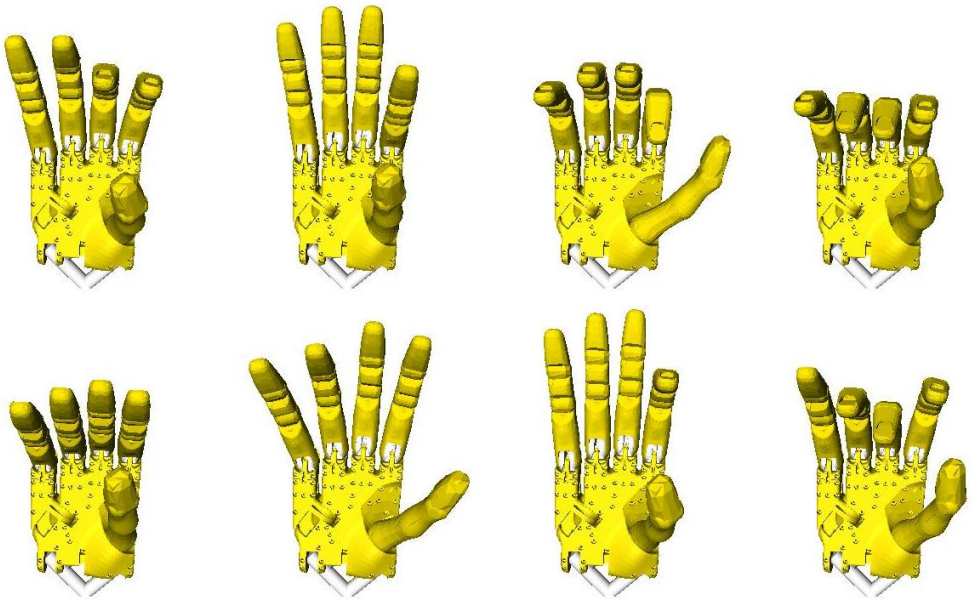


Figure 25: Hand poses corresponding to the origin of the eigenspace (mean-value) for the eight grasp-classes. From left to right and top to bottom: tripod, palmar pinch, lateral, writing tripod, parallel extension, addiction, tip pinch, and lateral tripod.

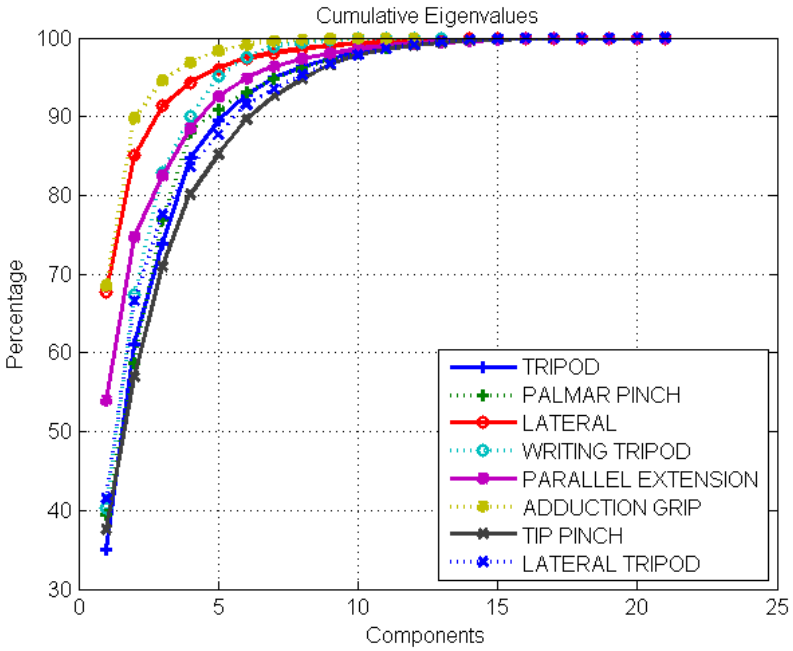


Figure 26: Cumulative eigenvalues of the synergies for each of the eight grasp-types studied. Between two and six eigenvalues are required to reach 90% of variance.

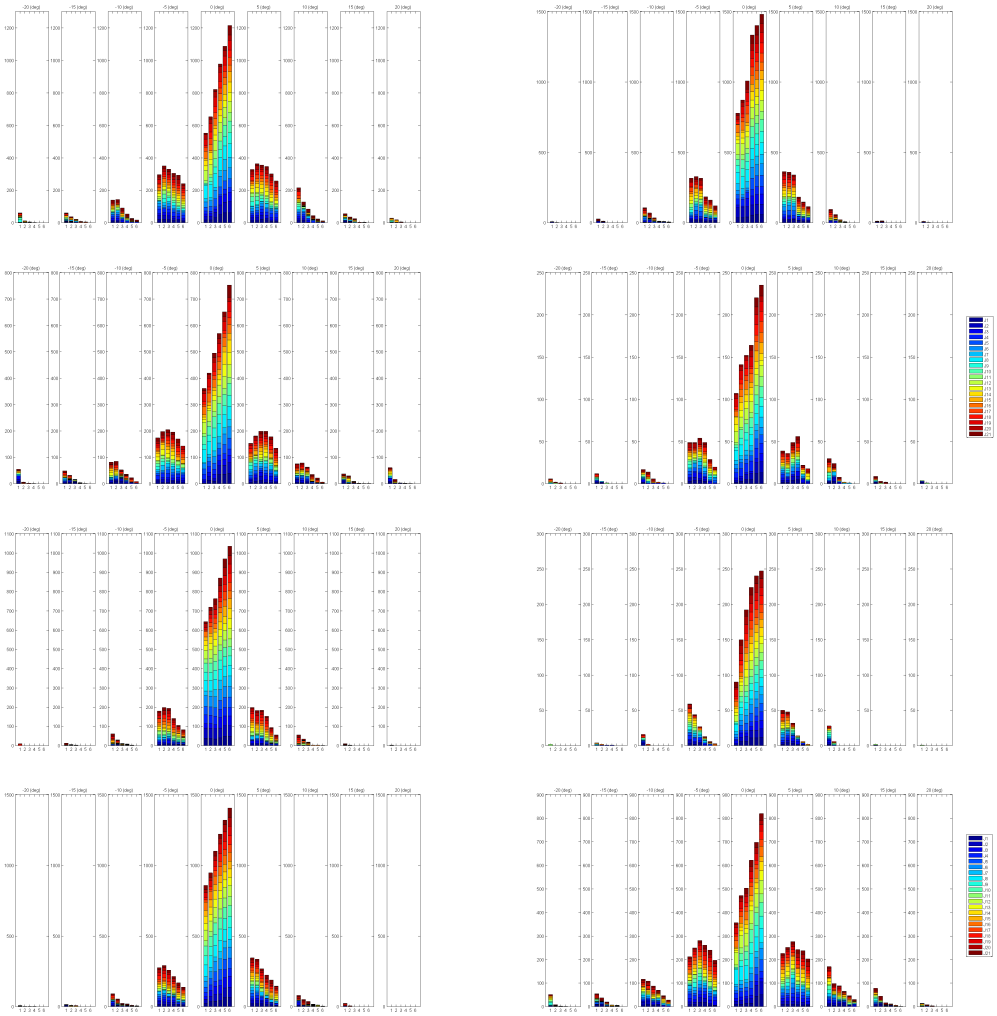


Figure 27: Histogram of pose reconstruction errors. The histograms show the joint-angle errors for pose reconstruction for the eight grasp-classes when using $1..6$ of the most significant principal components. In each diagram the nine big columns correspond to error bins of $\{-20, -15, -10, -5, 0, 5, 10, 15, 20\}$ degrees, and the errors corresponding to the use of $1..6$ principal components are shown as a bar inside the column. From left to right and top to bottom: tripod, palmar pinch, lateral, writing tripod, parallel extension, addiction, tip pinch, and lateral tripod.

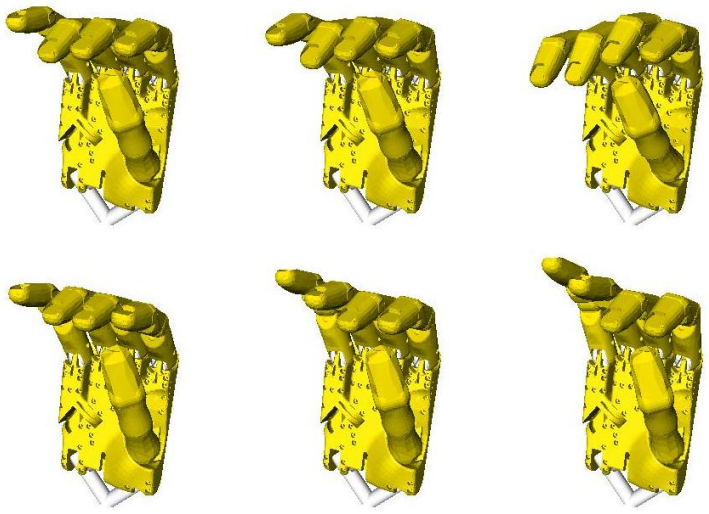


Figure 28: Default hand pose and variation due to the first two principal components for the writing-tripod grasp.

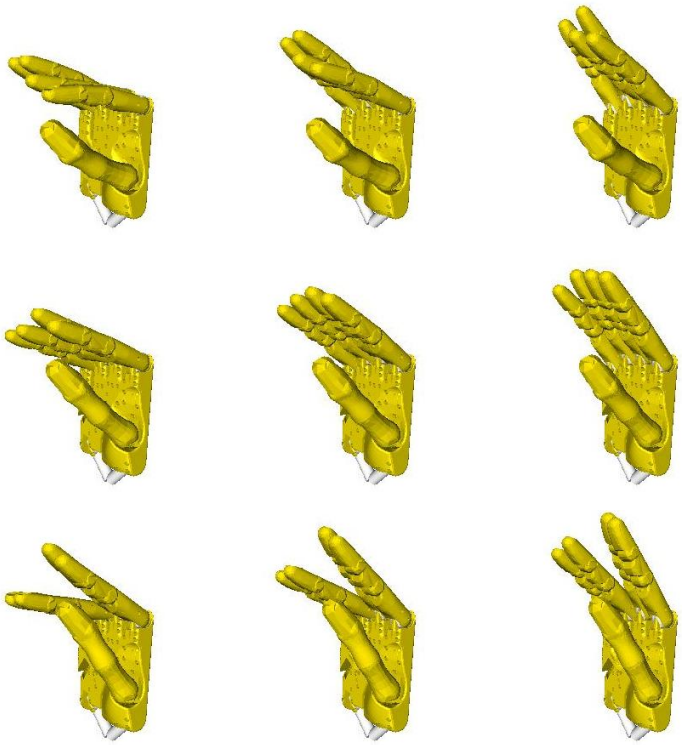


Figure 29: Default hand pose and variation due to the first two principal components for the parallel-extension grasp.

synergies. To explain 90% of the variance, and thus achieve mean square approximation errors below 10%, 6 principal components are enough in general. For the addiction grip type, due to its simplicity, we can achieve the same low approximation errors with only the first two principal components.

For grasp planning, we are interested in the distribution of joint-angle errors after reconstruction when using only the first few principal components. Figure 27 shows the corresponding histograms of joint-angle errors for the eight grasp classes analysed in our study. While the simple addiction grip is reconstructed almost perfectly, the more complex and dextrous grasps are also approximated well. We may observe that in all cases, the use of the 6 principal components concentrates the errors in the ± 5 degree range, with most errors smaller than the position accuracy of the actuators.

See figure 28 for an example of the derived synergies for the *writing-tripod* grasp class. The default hand pose corresponding to the grasp-class is shown in the center, and the outer hand poses correspond to parameterization of one synergy each. Another example is shown in figure 29 for the *parallel-extension* grasp class. The video attachment for this report demonstrates the motions corresponding to the first principal components for the *tip-pinch* grasp synergies.

4.6 Synergies and Object Affordances

Affordances are object properties that suggest ways to use and act upon them. In the manipulation context, affordances constrain the way objects are grasped. Here we have analyzed how the object dimensions along the hand opposition, ventral and lateral directions (see figure 30) are related to each of the synergy dimensions. We have computed the Pearson correlation coefficient between the object dimensions and all the principal components of each trial, at a 0.05 significance level. Strong positive or negative correlation are useful from the point of view of grasp planning since they allow to regress object dimensions to hand pre-shapes.

Results are shown in figure 31. As expected, most of the correlations are with respect to the opposition and ventral directions, which are the most related to possible contacts between hand and object. Parallel Extension shows its biggest correlation with the first, fourth and fifth principal components. Pinches (Palmar and Tip) show large correlations first and second principal component. Tripods (Simple, Lateral) correlate mostly with the third and forth dimensions. The Lateral grip correlates with the fourth and fifth principal components. Addiction grip and writing tripod show correlation with principal components of low energy which may be a result of low signal to noise ratio in these two grasp types—they can only be applied on a reduced set of objects, which limited their variance and number of trials in the recording phase.

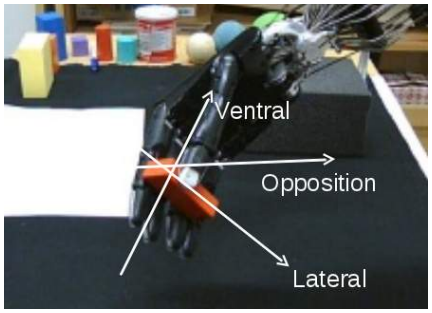


Figure 30: Synergy parameters are analyzed with respect to the object dimensions in the opposition (between contacts), ventral (towards palm) and lateral hand directions.

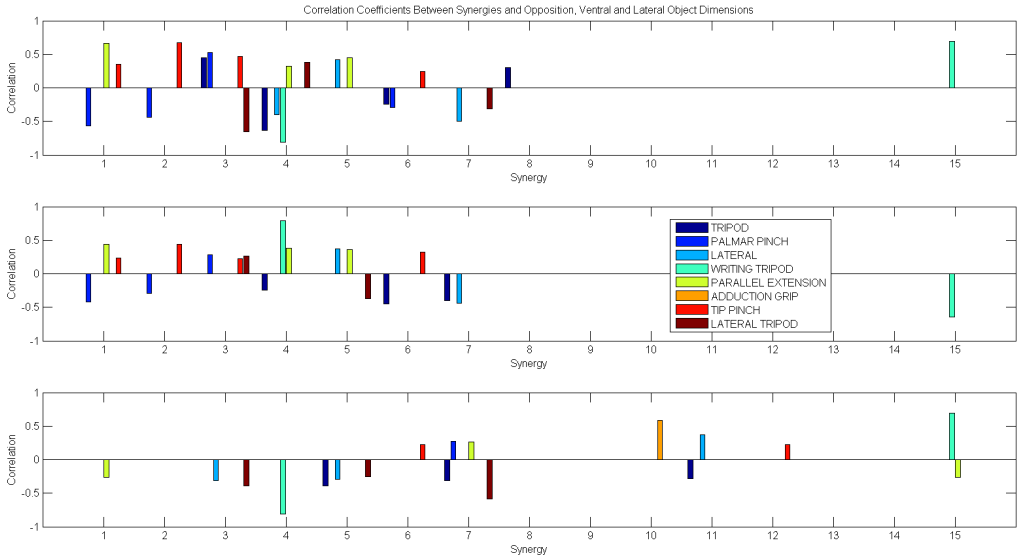


Figure 31: Correlation coefficients at a 0.05 significance level for all grasp types and principal components, with the dimensions of the object in the opposition (top), ventral (middle) and lateral (bottom) directions.

4.7 Ongoing work

The work reported in this section has only been started and will be continued in several research directions. First of all, there remains the integration of the synergies extracted from the robot experiments into the grasp planner, and to check how many eigenvalues are required to achieve stable grasps on a large variety of test objects.

Next, the existing Eigengrasp planner has no notion of object affordances and considers all parts of the target objects instead of exploiting symmetries or concentrating on the "useful" parts of an object — either marked explicitly by annotations in the object models by the user, or derived automatically from features in the geometry or recalled from similar objects.

We expect that the extracted synergies will also form a suitable set of basic skills for manipulation motions, in particular rolling finger contacts and in-hand rotation. However, adjusting the fingers to close on the object requires the new tactile sensors and force estimation algorithms [80], and the corresponding experiments will be performed as soon as the new sensors have been integrated on the Paris demonstrator platform.

5 Parameterizing manipulation sequences

This section proposes a new approach to analyse complex manipulation sequences, including finger gaiting. Our working hypothesis is that the hand movements of different humans are the same or largely similar when performing the same manipulation task, despite differences in hand size and shape. We try to extract this similarity from the sensor-data and call it the *action-gist*.

The method works in several steps. It starts with the extraction of the *meta-motion* by classifying the finger-motions according to carefully selected motion patterns based on finger flexure and abduction. This is done separately for each finger. To remove noise and improve the classification, the meta-motions are then processed by a Gaussian Markov Random Field, and the resulting gist is then presented as a joint histogram for all fingers and the thumb.

Several examples of the method are presented, and we also discuss an approach to invert the method to create and adapt finger motions based on the histograms. Currently, the analysis is based on joint-angle data obtained from CyberGlove recordings, but work is underway to include tactile data in the classification.

5.1 Introduction

The word *gist* means the essential part of an idea or experience. Different from *hand gesture*, the *in-hand manipulation action gist* is a concept with kinetic property. It represents the key hand motions in the manipulation task and widely adapts to different hands. The manipulation process is generalized as several compact motion guidelines. On the one hand, this makes it easy to remember, on the other hand it can be translated from one entity to another, just as the knowledge passing from the teacher to the student.

As we know, in the mechanism of the human hand , the motions and forces are governed by the neuromuscular apparatus, like an overview from [1] described. The movement of the hand is continuous, but according to human cognition, it can be classified as infinite 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 hand components over time. When the touching motion is executed, an interacting force is generated between the object and the hand, and the neuromuscular system keeps the hand in a proper *force applying state* that does not damage the hand itself but still holds the object firmly. Compared with other robotic humanoid hands, our air muscle hand from the Shadow Robot Company is more similar to the human hand and better protected against damage even when overforce is applied.

With a humanoid hand, a robot can implement much more human-like object manipulation than before. Because of the high degree-of-freedom, a multi-finger robot hand can perform more dexterous skills rather than grasping, which only constitutes translating the object from one place to another. It can rotate or shift objects and perform other advanced in-hand movement. 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 the object state is the aim of the manipulation. So here the manipulation process is considered as a *State-Action Model*, 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 joints angle variation, with the 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 is employed by both humans and robots with humanoid hands. However, it is unrealistic to map the motion exactly as from the demonstrator because of the different hand sizes. It can be imagined different-sized hands can interact with the object from different distances. And this can result 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. This is defined as *in-hand manipulation action gist*.

This study aims at proposing a 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 nonetheless key information. The action gist is expected to be universal for all in-hand movements regardless of whether it is simple (easy manipulation) or complex (finger-gaiting). The structure of this study 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 and generalization are introduced, and experiments are carried out to discuss the performance of the algorithms with different parameters. The final part is the conclusion and future work.

5.2 Related Work

There are multiple ways to generate a manipulation model, but the first point is to figure out which kind of model it is.

One class is to plan the motion in continuous space including the position and the speed of each relative component. The major stream is the dynamic movement

primitive (DMP) framework beginning from [46] and [52], in which the movement is recorded and represented with a set of differential equations. The position and the speed is controlled in terms of the immediate position and speed feedback. [49] 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, [43] 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, [44] considered the joint velocity space and enables the robot to accomplish similar tasks. As a result, this method can produce smoother trajectories than others.

Besides DMP, which is in a differential equation form to establish the movement from the beginning to the end, for instance, [45] applied the GMM method, which can also act as a learning core to generalize the moving trajectories.

The above framework consists of models depending on precise dataflow perception. However, for muscle control, it tracks the trajectory related to the moving tendency, not the position. Therefore, DMP does not offer any significant advantages in this field.

Another branch but a relatively older one is the generalized motor program (GMP), see [51] and [53]; here the overall process is guided by invariant features. [48] 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 a novel similar application. So far as the SMSR only extracts the body motion into simple joint angle variations such as increasing, decreasing and stationary, it would have difficulties while dealing with the multiple links cooperation application because it does not consider this kind of application so much. Different from simply defining the motion, it is possible to have a related higher semantic model. [50] 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.

[47] 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.

When the model is decided on, the next problem is how to sense the movement. Many studies concentrate on sensing from the robot, the examples can refer to [49], [43], or [44], but 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, [54] employed a color pattern 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, it can sense every finger joint relation in each data frame. Based on this kind of sensing channel, the study intends to generate an action gist model to represent human in-hand manipulation behavior.

5.3 Meta Motion Definition

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

1. It covers all possible movements of a hand
2. Each motion in the set is unambiguous from other motions
3. The motion involves the relative joint angle variation but no absolute position information

An exception to the above 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.

Supposing 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 phalange end* related to the palm (Fig.32), every finger has the same motion definition. Specifically, the coordinate origin of the thumb is different from the other four fingers because of its diverse position on the palm.

We project the finger motion into 2-dimensional space because the finger ends are fixed on the palm. In the X-Y plane, the finger direction is classified as 4 directions as the 4 quadrants in the Cartesian coordinate; plus with the open, close and the idle period, each finger has 9 types of meta motions, as shown in Fig.33. To ensure that the motion model has a uniform form, the X axis and the Y axis in the moving direction related to the coordinate origin is either parallel or vertical to the palm plane.

Humans can recognize each meta motion easily, but when one teaches another a hand skill, it is not by counting the exact sequence, but only by demonstrating the movement.

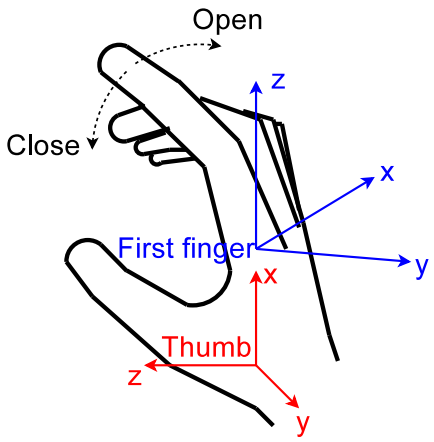


Figure 32: Finger coordinates related to the palm. The thumb in the red coordinate is different from the other four fingers due to its special location in the hand

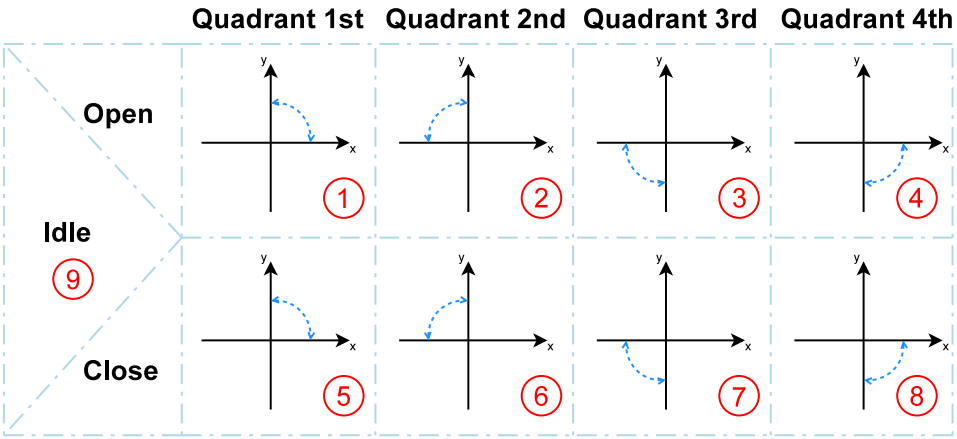


Figure 33: Nine types of meta motion in each finger. Two flex/ext-joints are modeled as one parameter as open or close, and the abduction angle cooperates with the metacarpal-proximal angle to form a 2D projected direction as the finger closes. The idle motion is specifically set apart and labeled as 9.

5.4 Action Gist from Data-glove

Here 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

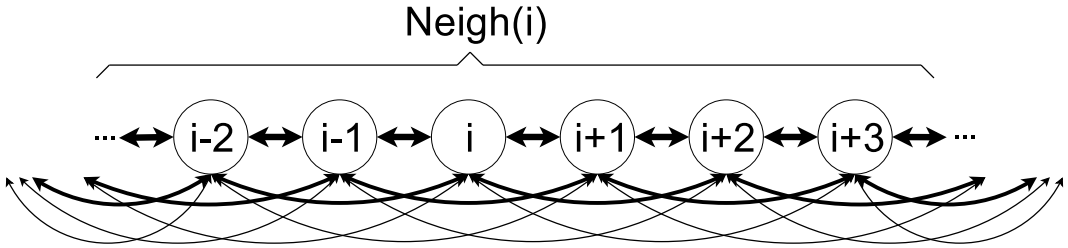


Figure 34: Node relationship according to Gaussian MRF. Supposing each data-glove value is a node, then each node is related to other nodes in the neighboring set $\mathrm{Neigh}(\cdot)$. With the impact factor obeying Gaussian distribution, the linewidth indicating the strength of the impact factor, we can see that the closer nodes sit closer have a stronger impact factor

the data-glove. However, according to the general law, the distal-intermediate and proximal-intermediate angles increase in close movement, decrease in open movement, and the varieties of metacarpal-proximal and abduction 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 noisy points, 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 at the moment the finger starts to touch the object, the value may be abnormal. Therefore 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. This algorithm considers each value frame 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 this assumption is illustrated in Fig.34.

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

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

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. And $C(\mathbf{v})$ 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 (Fig.35), 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 are listed in Tab.2. 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 create a contra direction with v_2, but v_2 is related stably. Whether the finger is open or close mainly depends on the movement between the proximal and intermediate joints. In Tab.2, "$\times$" means we do not need to think about what kind of value v_1 is.

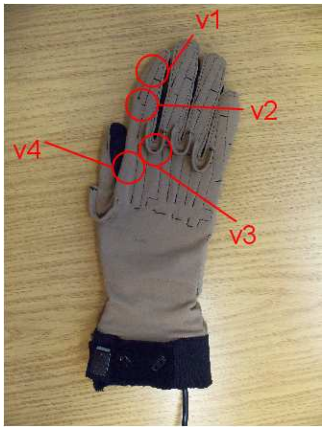


Figure 35: An example of the finger joint angle difference in the first finger. v_1 is for distal-intermediate, v_2 is for proximal-intermediate, v_3 indicates abduction between the first finger and middle finger, v_4 is for metacarpal-proximal

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:

$$P_i^j = \sum_{t \in \mathbf{Neigh}(i)} I_t^j G(t, i, \sigma) \tag{2}$$

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.34). Because the concerned action gist locates between each adjacent state pair, it is actually set as the entire glove value sequence. And σ is a parameter representing the area one node can primarily impact with, it also means the shortest single motion execution time corresponding

Table 2: Meta motion condition

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

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.

In addition to the action gist analysis, the idle motion is processed independently from the eight kinetic motions mentioned above. The Gaussian MRF based method can also be employed here, but according to the experimental experience, to find a frequent value as high as desired in the sliding window is a better solution. To realize this method, the first step is also to have the single similarities of each node to be similar to Eq.1, but the intensity of meta motion 9 at node i becomes $I_i^9 = 1$ and the condition becomes $C(\mathbf{v}) = 1 \iff \mathbf{v} = \mathbf{0}$. And then the idle sections can be determined by the following condition:

$$\sum_{t \in \mathbf{Neigh}(i)} I_t^9 > threshold \tag{3}$$

So node i stays idle when the sum of single intensities is larger than $threshold$. 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 an idle section is shorter than a single motion execution time σ, this section should be considered as not idle.

With the proposed procedure, the action gist can be extracted from the raw dataglove value.

5.5 Action Gist Generalization from Demonstration Set

To manipulate an object, there are countless methods to handle it. Though the action gist is the abstract from the floating value sequence, many different solutions still can be found according to the same application. In this case, to generalize multiple action gist sequences from human demonstration into a suggestive form is necessary. Generally, in the same application, different persons may behave differently, but the individual has a limited set of primitive hand motions to complete the task. So the concerned points in the gist generalization are to rank the popularity of the action gists obtained from the in-hand manipulation demonstration, including the operating order and the motion duration.

The operating order in a demonstration section is the meta motion permutation of five fingers between two adjacent states. Assuming the order sequence is $\mathbf{m} = \{\overrightarrow{m_1}, \overrightarrow{m_2}, \overrightarrow{m_3}, \cdots\}$, each element indicates the finger, the meta motion type, the normalized beginning, end time and other related information, if $\tau_{begin}(\cdot)$ can extract the begin time of the element and $\tau_{end}(\cdot)$ for the end time, then the following conditions must be fulfilled:

$$\begin{aligned} &\tau_{begin}(\overrightarrow{m_i}) < \tau_{end}(\overrightarrow{m_i}) \\ &\tau_{begin}(\overrightarrow{m_i}) \leqslant \tau_{begin}(\overrightarrow{m_j}) \end{aligned} \quad i < j \tag{4}$$

The ranking of order sequence depends on the frequency of different action gists in the specific manipulation scenario. The simple statistic method that only calculates the frequency of every action gist is a kind of solution, but if the sort can be related to every meta motion according to the motion duration, that will be a more composite evaluation on the entire demonstration set.

Every sample in the demonstration 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.

Therefore a meta motion occurrence histogram is applied to describe the statistic feature of the motion order from all demonstrated samples. The principle is, because the begin time and end time of each meta motion has been normalized as the range of $[0, 1]$, every motion occupying a time-order-related section increases the

corresponding area of the histogram with the impact factor, which is assumed to obey the Gaussian distribution. The process of histogram generation can be written as a formula as follows:

$$H_{a,r,l} = \sum_{\eta(\overrightarrow{m_i^s},a,r,l)=1} G(\psi(\overrightarrow{m_i^s}), a, \sigma_w) \tag{5}$$

where $\overrightarrow{m_i^s}$ is the element from action gist $\mathbf{m}^s$ in the demonstration set $\mathbf{M}$, $\eta(\cdot) = 1$ if and only if $\overrightarrow{m_i^s}$ belongs to finger r, labeled as meta motion l, and position a locates near $\overrightarrow{m_i^s}$ but no other motion on finger r. $\psi(\overrightarrow{m_i^s}) \in [0, 1]$ indicates the normalized order position when the meta motion begins, and σ_w is a parameter that controls the impact factor reduction, it is set as the reciprocal to the length of sequence $\mathbf{m}^s$. Considering the discrete numeric processing, the histogram has a resolution, the normalized a will finally be scaled as an integer form during calculation.

With the meta motion occurrence histogram, the frequent possible meta motion takes a higher value in the duration. As a result, even if every action gist is independent, it can be evaluated as in the following equation:

$$Score(\mathbf{m}^s) = \max_{seg(\mathbf{m}^s)} \sum_{j \in seg(\overrightarrow{m_i^s})} H_{j,r,l} \mid \begin{matrix} \tau_{finger}(\overrightarrow{m_i^s}) = r \\ \tau_{label}(\overrightarrow{m_i^s}) = l \end{matrix} \tag{6}$$

where $seg(\mathbf{m}^s)$ is the duration segmentation to action gist $\mathbf{m}^s$, it reallocates the normalized begin and end time of the movement as each segment is marked as $seg(\overrightarrow{m_i^s})$. τ_{finger} indicates which finger is in the meta motion element $\overrightarrow{m_i^s}$ and τ_{label} indicates the meta motion type.

This kind of behavior evaluation method can describe the local similarities of the meta motions, and has already implied the action gist frequency, so the action gist with the higher score is considered as a more common solution in the specific manipulation scenario.

5.6 Experiment and Discussion

A cyber-glove and several different kinds of objects are used to examine the proposed method. However, because the state recognition has not been designed yet, so far the demonstration is semi-supervised in that the begin and end state is defined manually, and the action gist is extracted from the glove value automatically.

Action Gist Extraction from Single Demonstration

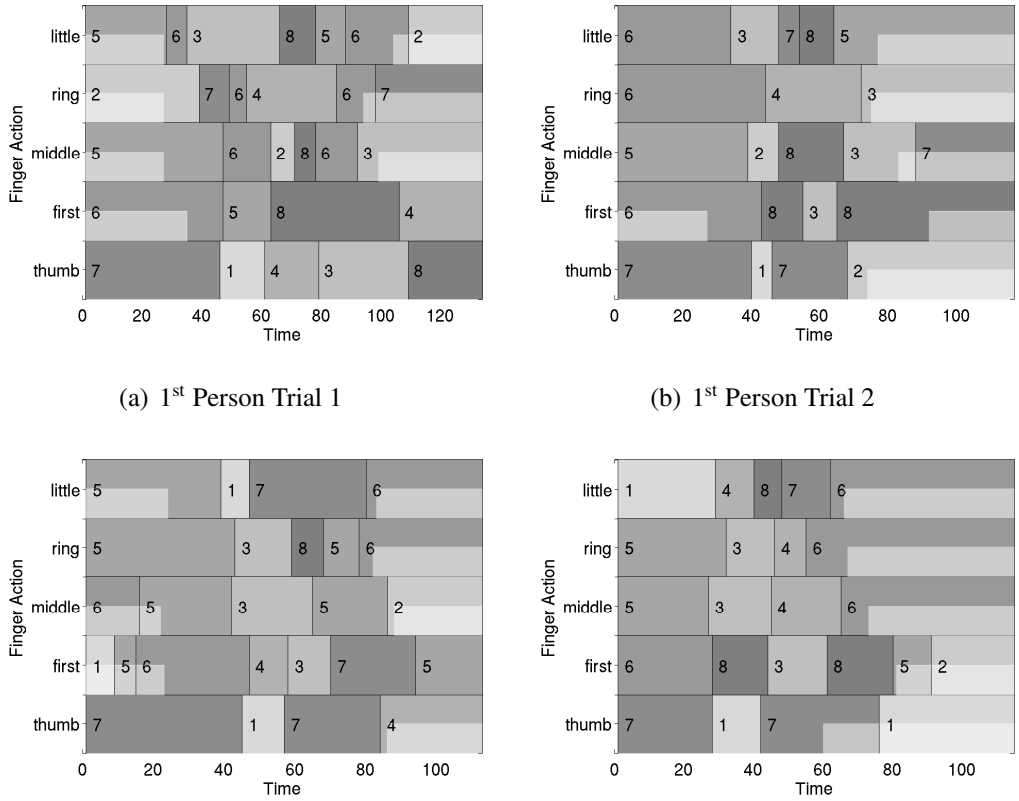
The first attempt is focusing on star-like block rotation. As Fig.36 shows, the block is fixed by four fingers (thumb, first, middle and ring finger) pinching the indentations, and the adjacent states are defined so that the moment that the block looks stable, each finger reaches the neighbor indentation from the previous one

Figure 36: Rotate a star-like block. Thumb, first, middle and ring finger are used to rotate the block about 72 degrees. This process is defined as one trial, the hand pose at the beginning state is quite similar to the end state.

Two subjects take part in this block rotation, each participant rotates the block in 9 trials in the same direction, with a very rough strategy discussed orally. Through the Gaussian MRF based algorithm, idle section identification and the short segment filter, we obtain the action gists. Two trials by two persons are shown in Fig.37. The meta motion is labeled as numbers over time except the possible idle sections are represented as *white transparent medium-high bars*. According to the cyber-glove framerate and the hand speed in this application, the parameters here are $\varepsilon = 0.05$, $\sigma = 10$, $threshold = 0.90$ and $d_{sw} = 20$. If a meta motion segment is shorter than 5 frames, it is filtered.

In the four samples we can find similarities, e.g. the motions in the thumb are likely to be motion 1 after motion 7, the first finger of the 1 person contains motion 8, the 2nd person contains motion 6 and motion 3, the 2nd person uses the middle finger or ring finger as meta motion "5 $\rightarrow$ 3". But generally speaking, because of the limited trial times, compared with similarities, it is easier to find the difference between two trials. Several trials do not have similar gists at all. This is because the application contains many motions between two defined adjacent states, and the participants are not strictly instructed to execute the movement, so we find many possibilities.

We apply Gaussian distribution in the node influence because it can reflect the impact importance by distance, and through parameter σ the impact range can be well represented. It should be set in terms of the motion speed and the cyber-glove performance, Fig.38 is the action gists from the same trials but σ is set as 20. The



(a) 1st Person Trial 1

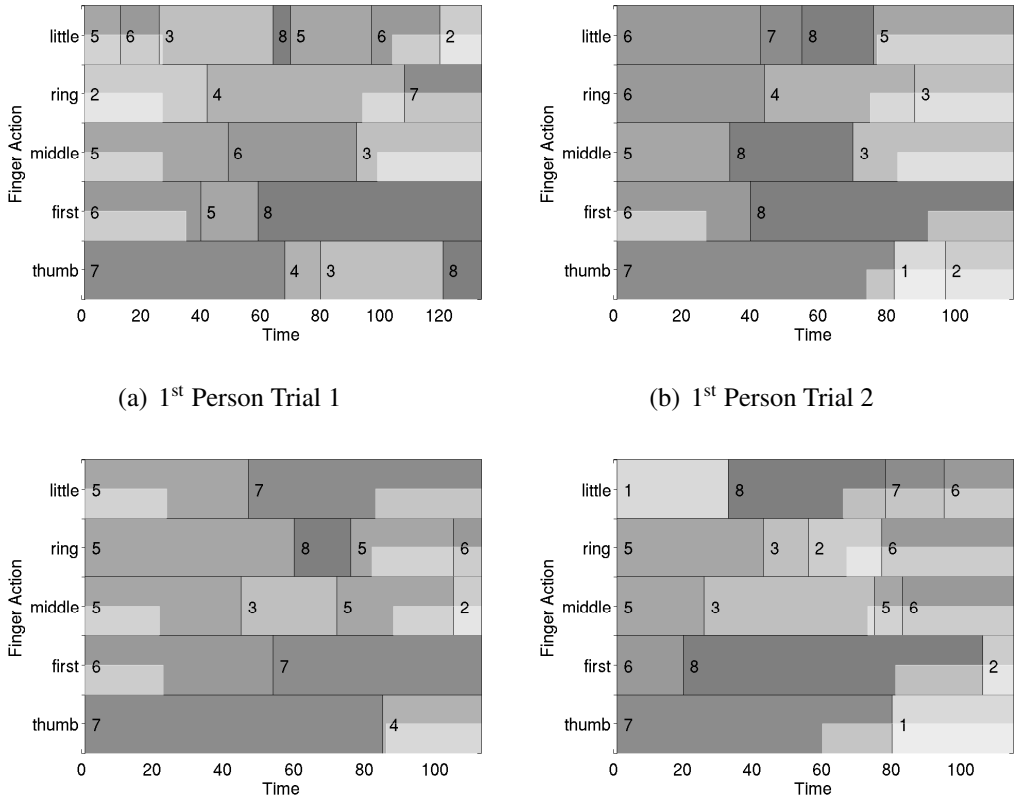
(b) 1st Person Trial 2

(c) 2nd Person Trial 1

(d) 2nd Person Trial 2

Figure 37: Action gists of star-like block rotation. Each subfigure is a repetition like Fig.36. To have the star-like block turn around 72 degrees. Each action gist is composed of the meta motions of five fingers, each meta motion is represented by different gray-level rectangles with the corresponding type number, Specifically for the idle motion, it is represented by the medium-high transparent bars, e.g. in the first finger of 1st Person Trial 1, a half height transparent bar overlaps meta motion 6, or the middle finger of 1st Person Trial 2, an idle section covers meta motion 3 and 7. The x-axis is a time axis indicating the cyber-glove frame number

sequences become clean but more abstract. Take the middle finger value in the 1st trial of the 1st person for example, the result when $\sigma = 20$ merges the meta motion sequence "$6 \rightarrow 2 \rightarrow 8 \rightarrow 6$" when $\sigma = 10$ into motion 6. If we analyze it from the raw value (Fig.39), the result when $\sigma = 20$ is not reasonable, there is apparently a peak around the 70th frame that divides the segments into two meta motions. But theoretically, three meta motions in one finger should be enough to support a block rotation (touch the object surface $\rightarrow$ leave the object surface $\rightarrow$ touch the object surface). So the parameter configuration depends on the usage background, not only one parameter is tried in the application.



(a) 1st Person Trial 1

(b) 1st Person Trial 2

(c) 2nd Person Trial 1

(d) 2nd Person Trial 2

Figure 38: Action gists of star-like block rotation when $\sigma = 20$. Compared with Fig.37, the action gist becomes clean but more abstract

Another key control variable in the Gaussian MRF based algorithm is the single meta motion similarity I_i^j, in Eq.2 it works as the intensity factor of Gaussian distribution. Let I_i^j be the sum of the finger joint value is an experience method in order to emphasize the varying importance of the glove value. We can set it as a fixed value and take a glance at the result, e.g. $I_i^j = 1$, but other parameters stay equal to the origin including $\sigma = 10$. Then we can have a set of result shown in Fig.40, and only a few diversities are found. Fig.41 illustrates the raw value of the first finger in the 2nd trial from the 1st participant, there is a peak around the 50th frame in the glove value, so we had better keep Eq.2 as the original form.

Different from other meta motions, idle motion works as a reference when the finger keeps silent during the kinetic motions, and because to find a frequent value as high as desired in the sliding window is easy to understand, the parameter d_{sw} is not discussed here.

There are many fingers working in block rotation, and then an easier in-hand manipulation scenario follows. On the table we use a ladle to spoon the soup up, so

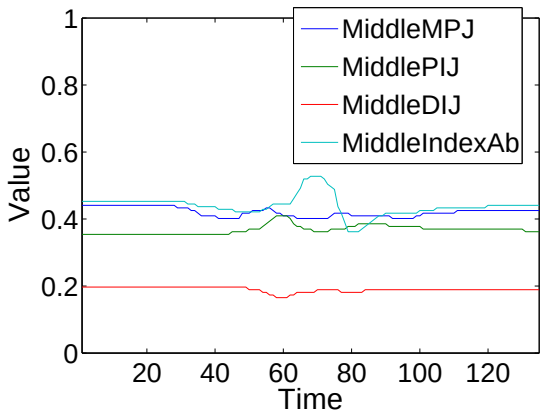


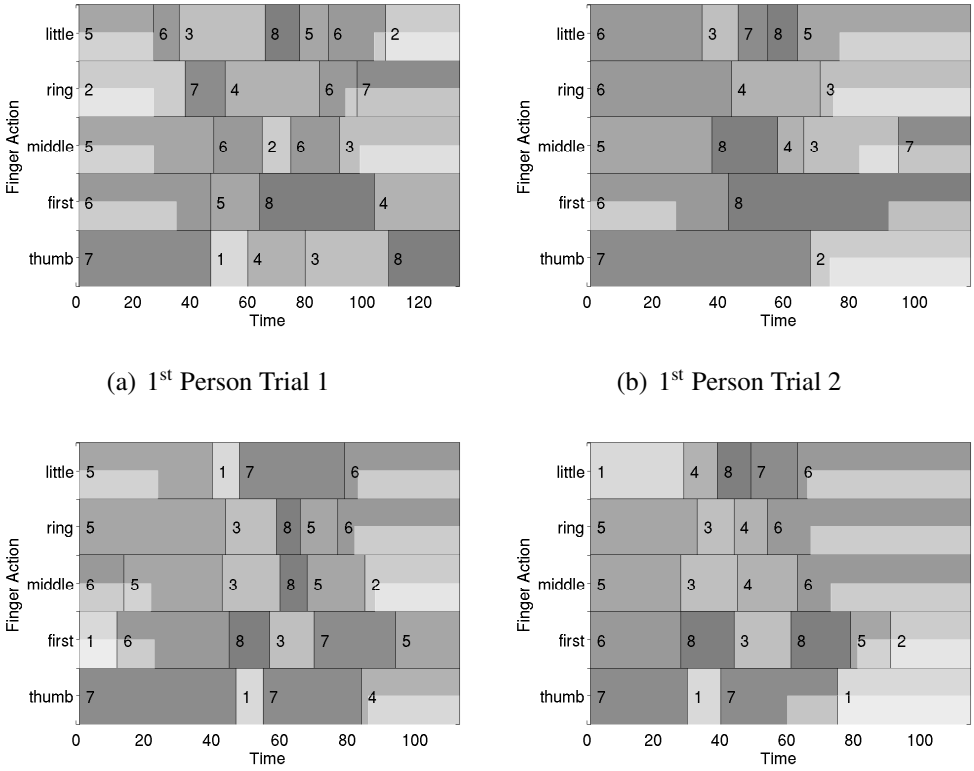
Figure 39: The raw value from the middle finger in the 1st trial of the 1st person. The time axis indicates the frame order number. There is a peak around the 70th frame for the MiddleIndexAb; this means the corresponding meta motion classification when $\sigma = 10$ is more reasonable than $\sigma = 20$

another experiment is related to ladle reconfiguration (Fig.42). One subject took part in the task and has 9 trials in which thumb, first and middle finger are applied. The start state is set as the ladle stomach staying in the low position, and the end state is when the ladle stomach stays in the high position. The action gists are obtained as Fig.43 shows at the parameter set as $\varepsilon = 0.05$, $\sigma = 20$, $threshold = 0.90$ and $d_{sw} = 20$.

The result indicates a probable solution to spoon a ladle up, that is meta motion 5 by thumb, motion $5 \rightarrow 8$ by first finger and motion 7 by middle finger. Theoretically, this result is reasonable, the application requires only a few steps to complete. Besides, it implies that if the two adjacent states are defined as shorter, the human demonstrator is more likely to prepare and implement with an identical manipulation method.

Action Gist Generalization from Demonstration Set

When the scale of the demonstration set becomes large, it is important to know which extracted action gist is popular in the same application by different subjects. Based on the local similarities calculation, we can give an evaluation of the mentioned experiments of star-like block rotation and ladle reconfiguration. According to Eq.5 with a resolution of 100, the generalization of the demonstration set is created, then with Eq.6 the score of each trial in the set can be solved by means of Dynamic Programming. Here the top 3 action gists are listed in the following tables when the parameters $\varepsilon = 0.05$, $\sigma = 20$, $threshold = 0.90$, $d_{sw} = 20$ and σ_w is set as $\frac{MatrixResolution}{ActionGistLength}$ namely $\frac{100}{ActionGistLength}$.



(a) 1st Person Trial 1

(b) 1st Person Trial 2

(c) 2nd Person Trial 1

(d) 2nd Person Trial 2

Figure 40: Action gists of star-like block rotation when $I_i^j = 1$. We can find several differences compared with Fig.37

Tab.3 is for the star-like block rotation, 18 trials from 2 participants are generalized in the meta motion occurrence histogram. There is no equal action gist in 18 trials, but based on the local similarities, each trial has an evaluated score for ranking.

And Tab.4 gives a ranking to the ladle reconfiguration related to 9 trials from 1 participant. 5 action gists are extracted, and the top evaluated action gist is the shortest but the most common one in all trials.

The meta motion occurrence histogram has the ability of position tolerance via normal distribution, and it is independently created for five fingers. Each finger has its own meta motion sequence histogram, so this mechanism is more robust than one bank representing the five fingers. With the promise that the action gist is generalized between two adjacent states, the scale of the meta motions is not large, thus it is possible to find the similarities through several or dozens of manipulation demonstrations. Besides global action gist evaluation, it is possible to find more interesting information from the occurrence histogram in the future.

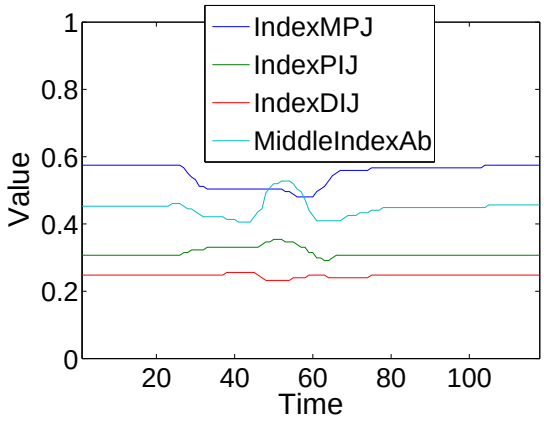
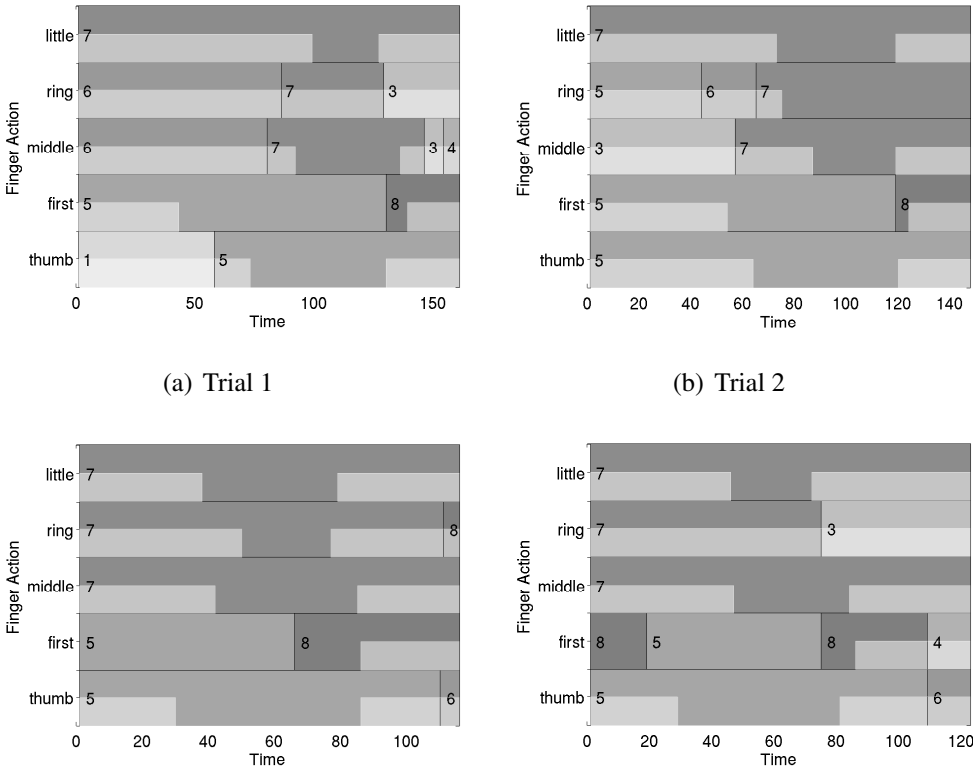


Figure 41: The raw value from the first finger in the 2nd trial of the 1st person. There is a peak around the 50th frame in the MiddleIndexAb, at this point to keep the single meta motion similarity as a sum is better than as a fixed value

Figure 42: In-hand rotation of a ladle. Thumb, first and middle finger participate in this scenario, the ladle is moved to a higher position and this process is defined as a trial

5.7 Conclusion and Future Work

This study concentrates on action modeling and generalization from the demonstration of in-hand manipulation. Different from the manipulator trajectory planning, this model works in a fuzzy way to guide the 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. In



(a) Trial 1

(b) Trial 2

(c) Trial 3

(d) Trial 4

Figure 43: Action gists of ladle reconfiguration. With the consideration of idle motion, we can find the action gists from 4 trials look similar the popular meta motions are motion 5 in thumb, 5 then 8 in first finger and 7 in middle finger

the future the model will be 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. Different-sized hands apply different joint angles to execute the manipulation, but the meta motion is always correct to indicate the finger movement direction. 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. So far the parameter control and the evaluation function in each iteration are being designed; we are confident that the method will work.

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 and executes the movement unnaturally, and difficult manipulation applications are hardly handled. 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; this point makes it impossible to guarantee that the finger movement perception is

Table 3: Action gist ranking of star-like block rotation

Action Gist – $\mathbf{m}^s$	**Rank**
(Thumb,Motion7), (Middle,Motion5), (Ring,Motion6), (First,Motion6), (Thumb,Motion1), (First,Motion5), (Middle,Motion8), (Ring,Motion4), (First,Motion8), (Thumb,Motion4), (Middle,Motion3), (Ring,Motion8)	1
(Thumb,Motion7), (Middle,Motion5), (Ring,Motion6), (First,Motion6), (First,Motion5), (Ring,Motion4), (Thumb,Motion1), (Middle,Motion8), (First,Motion8), (Middle,Motion3), (Thumb,Motion5)	2
(Thumb,Motion7), (First,Motion6), (Middle,Motion5), (Ring,Motion5), (First,Motion5), (First,Motion8), (Middle,Motion3), (Ring,Motion2), (Middle,Motion5)	3

Table 4: Action gist ranking of ladle reconfiguration

Action Gist – $\mathbf{m}^s$	**Rank**
(Thumb,Motion5), (First,Motion8), (Middle,Motion7)	1
(First,Motion5), (Thumb,Motion5), (Middle,Motion7), (First,Motion8)	2
(First,Motion8), (First,Motion5), (Thumb,Motion5), (Middle,Motion7), (First,Motion8)	3

always correct. So to fuse the result from other sensors is another direction for developing the model further.

As introduced at the beginning, the in-hand manipulation process is considered as a *State-Action Model*, this study only discusses the *Action* generation. Another important point is that *State* involves the posture of hand and object, and the contact state. That part is related to visual, haptic and other perceptual channels. This gap is going to be filled in the next step.

6 Summary

The underlying concept of project HANDLE is the recording of human manipulation traces, in order to then derive motion-primitives for manipulation with a multi-finger robot hand. However, despite the unique multi-sensor setup designed for the human recordings, and despite the large number of available experiment trials, the analysis of the traces has proven more complex than anticipated. The direct transfer of human motor-skills extracted from an experiment to the robot requires a difficult match from the human hand to the kinematics of the robot, and no two human hands are the same. On the other hand, traditional grasp planning algorithms for the Shadow hand must consider the extrinsic 6-DOF for the relative pose of hand and target object plus the 24-DOF required to specify all finger joint-positions, and therefore faces all problems related to searching in high-dimensional spaces.

We decided on the straightforward alternative, and simply recorded a new set of grasps under human demonstration, but using the Shadow hand itself to perform the grasps. The hand-shape and kinematics is therefore known exactly, and the finger and arm movements are recorded with high precision. However, the human experimenters are still in full control, and grasps and sequences are only accepted when the experimenter decides that a grasp-pose is "human-like". Section 2 motivated this approach and described the experiment setup and the required data-glove calibration. The recording software generates output files largely compatible with the format used for the human-recordings [63], and the new traces augment the available database.

To extract a number of *basic grasping skills* for the Shadow hand, we then recorded a number of stable grasp poses for a total of eight different grasp-classes from the taxonomy, and using objects of different size and shape. This approach has been detailed in section 3. Clustering the results in joint-space and associating the joint-angles with object shape and size allows us to play back suitable human-like grasp-poses instead of performing a full grasp-planning.

To grasp and handle complex objects, the extraction and parametrization of suitable *grasp-synergies* provides the most promising way to reduce the curse of dimensionality. The current state of this ongoing work has been the topic of section 4. The first step was based on the GraspIt! simulator, which includes hand-synergies for a humanoid 17-DOF hand, based on the original synergies from [20]. An empirical mapping of those synergies to the Shadow hand kinematics allows us to run the Eigengrasp-planner from GraspIt! to generate grasps for arbitrary objects. To achieve better synergies, we have started the PCA on the grasps recorded with the Shadow hand itself. We expect that the synergies extracted from the data will also provide nice motion-primitives for manipulation.

Finally, section 5 has proposed the *action gist* technique, used to define and recognize basic manipulation primitives. A manipulation trace, recorded by the data-glove or other sensors, is first classified locally according to the finger-flexure (idle,

closing, opening) plus finger-abduction. For better robustness against noise and involuntary finger movements, a Gaussian Markov random field approach is then used to filter the data. We presented several examples on complex motion-sequences, also involving finger-gaiting.

6.1 Future work

The work presented in this report is only the first step towards defining suitable motion-primitives for the Shadow hand. While static grasps work very well with the hand, the fine control of manipulation movements needs additional work and would also greatly benefit from better tactile sensing. We plan to record additional grasps and manipulation sequences on the Paris demonstrator hand as soon as the newly designed tactile solution is available for experimentation.

References

[1] C. L. Taylor and R. J. Schwarz. The anatomy and mechanics of the human hand. *Artificial limbs*, 2(2):22–35, 1955.

[2] C. Ferrari and J. Canny; *Planning Optimal Grasps* In Proceedings of the IEEE Int. Conference on Robotics and Automation, pages 2290–2295, Nice, France, 1992.

[3] K.B. Shimoga, *Robot grasp synthesis algorithms: a survey*, International Journal of Robotics Research, vol.15, 230–266, 1996

[4] A. Bicchi, V. Kumar, *Robotic grasping and contact: A review*, IEEE International Conference of Robotics and Automation, 348–353, 2000

[5] M.R. Cutkosky, *On grasp choice, grasp models, and the design of hands for manufacturing tasks*, IEEE Transactions on Robotics and Automation, vol.5, 269–279, 1989

[6] S. Arimoto, *Control Theory of Multi-fingered Hands*, Springer 2008

[7] T. Iberall, *Human prehension and dexterous robot hands*, International Journal of Robotics Research, vol. 16, 285-–299, 1997.

[8] S.C. Jacobsen, J.E. Wood, D.F. Knutti, and K.B. Biggers, *The UTAH/M.I.T. Dextrous Hand: Work in Progress* The International Journal of Robotics Research, Vol.3, 21–50, 1984

[9] M. Kondo, J. Ueda, T. Ogasawara, *Recognition of in-hand manipulation using contact state transition for multifingered robot hand control*, Robotics and Autonomous Systems 56, 66-81, 2008

[10] M. Hüser, T. Baier, J. Zhang; *Learning of demonstrated Grasping Skills by stereoscopic tracking of human hand configuration*, IEEE Intl. Conference on Robotics and Automation, 2795-2800, 2006

[11] H. Kjellström, J. Romero, D. Kragic, *Visual Recognition of Grasps for Human-to-Robot Mapping*, Proc. 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, 3192–3197, 2008

[12] J. Aleotti, S. Caselli, *Grasp Recognition in Virtual Reality for Robot Pregrasp Planning by Demonstration*, IEEE Intl. Conference on Robotics and Automation, 2801-2806, 2006

[13] R.S. Dahiya, G. Metta, M. Valle, G. Sandini, *Tactile Sensing—From Humans to Humanoids*, IEEE Transactions on Robotics, vol. 26, no.1, 1–20, 2010

[14] K. Matsuo, K. Murakami, T. Hasegawa, K. Tahara, R. Kurazume, *Segmentation method of human manipulation task based on measurement of force imposed by a human hand on a grasped object*, Proc. 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, 1767–1772, 2009

[15] Shadow Robot Dextrous Hand, www.shadowrobot.com

[16] John Lloyd and Vincent Hayward, *Multi-RCCL User's Guide*, McGill Research Centre for Intelligent Machines, McGill University, Montreal, Canada, April 1992

[17] F. Röthling, *Real Robot Hand Grasping using Simulation-Based Optimisation of Portable Strategies*, Ph.D Thesis, Technische Fakultät, Universität Bielefeld, 2007

[18] D. Bartenieff and I. Lewis, *Body Movement: Coping with the Environment*, Gordon and Breach Science, New York, 1980

[19] N. Hendrich, D. Klimentjew and J. Zhang, *Multi-sensor segmentation of human manipulation tasks*, Proc. IEEE MFI-2012, Salt-Lake City, 2012

[20] M. Santello, M. Flanders, J. F. Soechting, *Postural hand synergies for tool use*, Journal of Neuroscience, vol. 18 no. 23, pp. 10 105–10 115, 1998.

[21] J.M. Elliott and K.J. Connolly, *A classification of manipulative hand movements*, Developmental Medicine & Child Neurology, 26: 283-296, 1984.

[22] M. Ciocarlie, C. Goldfeder, P.K. Allen, *Dimensionality reduction for hand-independent dexterous robotic grasping*, Proc. 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems, 3270–3275, 2007

[23] M. Ciocarlie and P.K. Allen, *Hand Posture Subspaces for Dexterous Robotic Grasping*, International Journal of Robotics Research, vol. 28, 851–866, 2009

[24] Andrew T. Miller; *GraspIt!: A Versatile Simulator for Robotic Grasping*. Ph.D. Thesis, Department of Computer Science, Columbia University, June 2001.

[25] C. Goldfeder, M. Ciocarlie, H. Dang, P.K. Allen, *The Columbia Grasp Databse*, IEEE Internatioonal Conference on Robotics and Automation, 1710–1716, 2009

[26] F.Röthling, R.Haschke, J.J.Steil, and H.J.Ritter, *Platform Portable Anthropomorphic Grasping with the Bielefeld 20-DOF Shadow and 9-DOF TUM Hand*, Proc. IROS-2007, 2951-2956, San Diego, 2007

[27] M. Ciocarlie, H. Dang, J. Lukos, M. Santello, P. Allen, *Functional Analysis of Finger Contact Locations during Grasping*, Prod. Eurohaptics Conference and Symposium on Haptic Interface for Virtual Environment and Teleoperator Systems, 401–405, 2009

[28] Ch. Borst, M. Fischer and G. Hirzinger; *Calculating Hand Configurations for Precision and Pinch Grasps*. Proceedings of the 2002 IEEE/RSJ International Conference on Intelligent Robots and Systems, Lausanne, Switzerland, 2002.

[29] Ch. Borst, M. Fischer and G. Hirzinger; *Grasp Planning: How to Choose a Suitable Task Wrench Space*. Proceedings of the IEEE Intl. Conference on Robotics and Automation (ICRA), New Orleans, USA, 2004.

[30] M. Schöpfer, H. Ritter, G. Heidemann, *Acquisition and Application of a Tactile Database*, IEEE International Conference on Robotics and Automation, 1517–1522, 2007

[31] D.D. Nguyen, T.C. Phan, J.W. Jeon, *Fingertip Detection with Morphology and Geometric Calculation*, Proc. 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, 1460–1465, 2009

[32] Y. Sun, J.M. Hollerbach, S.A. Mascaro, *Estimation of Fingertip Force Direction With Computer Vision*, IEEE Transactions on Robotics, vo. 25, no.6, 1356–1369, 2009

[33] N. S. Pollard; *Parallel Algorithms for Synthesis of Whole-Hand Grasps*. Proceedings of the IEEE International Conference on Robotics and Automation, Albuquerque, NM, 1997.

[34] Y. Lui and M. Lam; *Searching 3-D Form Closure Grasps in Discrete Domain*. In Proceedings IEEE International Conference on Intelligent Robots and System, Las Vegas, Nevada, October 2003.

[35] L. Han, J. Trinkle, Z. X. Li; *Grasp Analysis as Linear matrix Inequality Problems*. IEEE Transactions on Robotics and Automation, vol. 16, no. 6, pp663-674, 2000.

[36] R. Haschke, J. Steil, I. Steuwer, H. Ritter; *Task-Oriented Quality Measures for Dextrous Grasping*, Proc. IEEE Conference on Computational Intelligence in Robotics and Automation, 2005.

[37] A.T. Miller, S. Knoop, H.I. Christensen, P.K. Allen, *Automatic grasp planning using shape primitives*, IEEE Internation Conference on Robotics and Automation, 1824–1829, 2003

[38] J. Zhang, B. Rössler; *Self-valuing learning and generalization with application in visually guided grasping of complex objects*. Journal of Robotics and Autonomous Systems, 47: 117–127, 2004.

[39] J. Kim, J. Park, Y. Hwang, M. Lee; *Advanced Grasp Planning for Handover Operation Between Human and Robot: Three Handover Methods in Esteem Etiquettes Using Dual Arms and Hands of Home Service Robot*, 2nd Intl. Conference on Autonomous Robots and Agents, 2004

[40] T. Baier, J. Zhang, *Resuability-based Semantics for Grasp Evaluation in Context of Service Robotics*, IEEE International Conference on Robotics and Biomimetics, 2006

[41] T. Baier, J. Zhang, *Learning to Grasp Everyday Objects using Reinforcement-Learning with Automatic Value Cut-Off*, IEEE/RSJ International Conference on Intelligent Robots and Systems, 2007

[42] D. R. Faria and J. Dias, *3D Hand Trajectory Segmentation by Curvatures and Hand Orientation for Classification through a Probabilitistic Approach*, Proc. IEEE/RSJ IROS 2009, 1284-1289

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

[44] M.J. Gielniak, C.K. Liu, and A.L. Thomaz. Stylized motion generalization through adaptation of velocity profiles. In *IEEE International Symposium on Robots and Human Interactive Communications*, pages 304–309, sept. 2010.

[45] Elena Gribovskaya, S. M. Khansari-Zadeh, and Aude Billard. Learning nonlinear multivariate dynamics of motion in robotic manipulators. *International Journal of Robotics Research*, 30(1):80–117, 2010.

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

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

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

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

[50] Xingda Qu and M.A. Nussbaum. Simulating human lifting motions using fuzzy-logic control. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 39(1):109–118, 2009bitemRichardA1975225

[51] Richard A. Schmidt. A schema theory of discrete motor skill learning. *Psychological Review*, 82(4):225–260, 1975.

[52] S. Schaal, A. Ijspeert, and A. Billard. Computational approaches to motor learning by imitation. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 358(1431):537–547, 2003.

[53] R.A. Schmidt and T.D. Lee. *Motor control and learning: a behavioral emphasis*. Human Kinetics, 1999.

[54] M. Riley and G. Cheng. Extracting and generalizing primitive actions from sparse demonstration. In *IEEE-RAS International Conference on Humanoid Robots*, pages 630–635, 2011.

[55] Videre Systems, www.videredesign.com/vision/sth_mdcs3.htm

[56] Polhemus Liberty Electromagnetic Motion Tracking System, www.polhemus.com/?page=Motion_Liberty

[57] Phasespace Impulse optical tracker, www.phasespace.com/productsMain.html

[58] Cyberglove systems, www.cyberglovesystems.com

[59] Tekscan Grip, www.tekscan.com/medical/system-grip.html

[60] HANDLE project, *D3 — Augmented sensing object*, www.handleproject.eu, 2009

[61] Nintendo Corp., Wiimote, www.nintendo.com/wii/what/controllers

[62] HANDLE project, deliverable D3, *Instrumented sensing objects*, www.handleproject.eu, 2009

[63] HANDLE project, deliverable D4, *Protocol for the corpus of sensed grasp and handling data*, www.handleproject.eu, 2009

[64] HANDLE project, deliverable D5, *Sensor system specification and evaluation of different methods for object recognition*, *Protocol for the corpus of sensed grasp and handling data*, www.handleproject.eu, 2009

[65] HANDLE project, *D6 — Tactile sensing for data-gloves*, www.handleproject.eu, 2009

[66] HANDLE project, deliverable D7, *Algorithms for real-time collision avoidance accompanied by a report on existing planning methods*, www.handleproject.eu, 2009

[67] HANDLE project, deliverable D9, *Second robot hardware platform*, www.handleproject.eu, 2010

[68] HANDLE project, deliverable D10, *Annotated cataloge of grasp and force/-motion signatures*, www.handleproject.eu, 2010

[69] HANDLE project, deliverable D12, *Hand-state models of human grasping and manipulation skills*, www.handleproject.eu, 2010

[70] HANDLE project, deliverable D13, *Algorithms for planning the grasping of objects for manipulation and for planning the in-hand manipulation*, www.handleproject.eu, 2011

[71] HANDLE project, deliverable D14, *Improving known actions from motor babbling*, www.handleproject.eu, 2011

[72] HANDLE project, deliverable D15, *Reports on organised workshops with generated documentation*, www.handleproject.eu, 2011

[73] HANDLE project, deliverable D16, *Hand design report*, www.handleproject.eu, 2011

[74] HANDLE project, deliverable D17, *Automatic dataset reduction system for grasp motion data*, www.handleproject.eu, 2011

[75] HANDLE project, deliverable D18, *Visual and tactile perception algorithms for grasping*, www.handleproject.eu, 2011

[76] HANDLE project, deliverable D19, *Embedded electronic design report*, www.handleproject.eu, 2011

[77] HANDLE project, deliverable D20, *Skin design report*, www.handleproject.eu, 2011

[78] HANDLE project, deliverable D21, *Motion primitives for human-like grasping and tool use with a robotic hand*, www.handleproject.eu, 2012

[79] HANDLE project, deliverable D22, *Low-level controllers desgin including hybrid force/position control and visual servoing*, www.handleproject.eu, 2012

[80] HANDLE project, deliverable D23, *Visual and tactile perception system evaluation report*, www.handleproject.eu, 2012

[81] HANDLE project, deliverable D24, *Parameterizing and creating new actions*, www.handleproject.eu, 2012

[82] HANDLE project, deliverable D25, *Complete anthropomorphic hand*, www.handleproject.eu, 2012

[83] Andrew T. Miller, *Graspit!: A versatile simulator for robotic grasping*, IEEE Robotics and Automation Magazine, vol. 11, 110–122, 2004

[84] N. Koenig, and A. Howard, *Design and use paradigms for gazebo, an open-source multi-robot simulator*, Proc. IROS 2004, 2149–2154

[85] Bullet Physics Library, http://bulletphysics.org/, 2006

[86] JBullet physics engine, http://jbullet.advel.cz, 2008

[87] Hanno Scharfe, *Physikbasierter Simulator für Greif- und Manipulationsverfahren mit Mehrfinger-Roboterhänden*, Diploma thesis, University of Hamburg, 2010

[88] H. Scharfe, N. Hendrich, and J. Zhang, *Hybrid physics simulation of multi-fingered hands for dexterous in-hand manipulation*, Proc. ICRA 2012, to appear

[89] Lorenzo Sciuto, *Robotic Hand and Sensorized Glove: A Calibration for Managing Robotic Grasp in Teleoperation*, MSc. thesis, University of Siena, 2011

[90] Eugen Richter, *Hand pose reconstruction using a three-camera stereo vision system*, diploma-thesis, University of Hamburg, 2011

[91] Hao Dang, Jonathan Weisz, and Peter K. Allen, *Blind Grasping: Stable Robotic Grasping Using Tactile Feedback and Hand Kinematics*, Proc. ICRA-2011, 2011

[92] C. Ott, *Cartesian Impedance Control of Redundant and Flexible-Joint Robots*, Springer Tracts in Advanced Robotics 49, Springer 2008