Project Acronym:	GRASP
Project Type:	IP
Project Title:	Emergence of Cognitive Grasping through Introspection, Emulation and Surprise
Contract Number:	215821
Starting Date:	01-03-2008
Ending Date:	28-02-2012

Deliverable Number:	D20
Deliverable Title :	Hybrid visual-tactile-proprioceptive controller and adaptive grasping on multiple platforms
Type:	PU
Authors	V. Kyrki, J. Laaksonen, A. Morales, J. Felip, and D. Kragic
Contributing Partners	LUT, UJI, KTH

Contractual Date of Delivery to the EC:	28-02-2010
Actual Date of Delivery to the EC:	28-02-2010

Contents

Chapter 1

Executive summary

Deliverable D20 presents second year developments within workpackage WP3 "Self-experience of Grasping and Multimodal Grounding". According to the Technical Annex of the project, D20 presents activities connected to Tasks 3.1, 3.2, and 3.3. The objectives of these tasks are defined as

- [**Task 3.1**] **- Control Architecture.** Initially, a hierarchical control architecture will be defined and developed such that it allows relating the concepts of the grasping ontology defined in WP2 to the immediate control. After the architecture has been defined, this task will continue with the definition and development of the general control architecture components, mainly a Cartesian controller and high-level supervisory and visual controllers.
- [**Task 3.2**] **- Multimodal Grounding.** The task aims for the definition and development of a grounding mechanism connecting action primitives and attributes with uncertain sensor information, including modelling of the uncertainties involved. Initially, the modelling of uncertainties of the three sensor types (visual, tactile, proprioceptive) is studied considering the context of the attributes of the grasping ontology. Later, the task will continue by studying the temporal grounding problem as a state estimation problem with uncertain information, as the concepts and therefore the symbol set are defined by the grasping ontology.
- [**Task 3.3**] **- Robust action primitives.** The task aims for the definition and evaluation of adaptive and robust control approaches for individual action primitives. The main focus will be on studying the possible grasp primitives for different hand kinematics (parallel jaw, three-fingered, five fingered) and to identify robust parameterisable primitives through evaluation. Parameterisation of the primitives allows self-experience to be used for improving the performance during future attempts.

The work in this deliverable relates to the following third year milestone:

- [**Milestone 8**] - Implementation of hybrid controllers for on-line adaptive primitive grounding; evaluation in the simulator and on experimental platforms.

The progress in WP3 is presented briefly below, and in more detail in the appendix containing attached scientific publications and reports.

- Attachments A, B, and F present work in connecting tactile sensor measurements to grasp stability, related to Task 3.2. Initial work in the topic was shown in Year 2 deliverable D13, and the work is concluded in the attached publications, which demonstrate that the stability recognition using tactile sensing is feasible, study the usefulness of different feature extraction and machine learning approaches, and evaluate the contribution of different types of knowledge for inferring about the grasp stability in both simulation and on real hardware. A preliminary version of Attachment A was included in the second year deliverable.
- Attachment I presents initial work extending the results of grasp stability recognition to controlling the grasping process. The work is thus connected to both Tasks 3.2 and 3.3. The main idea is

that the whole sensor-based grasping process is considered in a probablistic context. Through this viewpoint, we take into account the inherent uncertainty related to the measurements, the object and the manipulator. Using the uncertain information, we produce a series of grasps leading to a stable grasp.

- Attachment C presents work on the development of visual skills necessary to implement reactive grasping using vision as one of the feedback sensors, related to adaptive primitive development in Task 3.3. In the paper, a new approach is proposed for the visual tracking of a robot hand suitable for observing the interaction between robot and object.
- Adding visual feedback to adaptive control requires the calibration of the camera position with respect to the robot. Attachment G presents a simple yet robust and practical method for optimal estimation of the calibration.
- Attachment D presents improvements on adaptive manipulation primitives. The primitives form a complete set for transporting objects and include both hand and arm control. The work is part of Task 3.3.
- As the primitives (Task 3.3) have been extended to arm motions, one option for learning these is from physical interaction with a human. The work in Attachment E studies which kind of response of a robot is preferable to a human user in the context of the human physically guiding the robot through a motion. In addition, guidance based control primitives are described.
- Attachment H shows an application of the manipulation primitives paradigm (Task 3.3) solving a complex manipulation task. The task consists of emptying a box whose location is barely known, and which contains an undefined number of unknown objects. All the primitives are sensor based and implement a reactive behavior that adapts to the uncertain and changing real environment.

Appendix A

Attached papers

A Janne Laaksonen, Ville Kyrki, and Danica Kragic. Evaluation of feature representation and machine learning methods in grasp stability learning. *IEEE International Conference on Humanoid Robots, Humanoids 2010.*

B Yasemin Bekiroglu, Ville Kyrki, and Danica Kragic. Learning grasp stability with tactile data and HMMs. *IEEE International Symposium on Robot and Human Interactive Communication, ROMAN 2010.*

C Jose J. Sorribes, Mario Prats and Antonio Morales. Visual tracking of a jaw gripper based on articulated 3D models for grasping. *IEEE International Conference on Robotics and Automation, ICRA 2010.*

D Javier Felip and Antonio Morales. UJI humanoid torso manipulation primitives. Research report. June 8, 2010. Universitat Jaume I.

E Marta Lopez Infante and Ville Kyrki. Usability of Force-Based Controllers in Physical Human-Robot Interaction. To be published in *ACM/IEEE International Conference on Human-Robot Interaction, HRI 2011.*

F Yasemin Bekiroglu, Janne Laaksonen, Jimmy Alison Jorgensen, Ville Kyrki, and Danica Kragic. Assessing grasp stability based on learning and haptic data. Manuscript accepted with minor changes to *IEEE Transactions on Robotics.*

G Jarmo Ilonen and Ville Kyrki. Robust robot-camera calibration. Submitted to *International Conference on Advanced Robotics, ICAR 2011.*

H Javier Felip and Antonio Morales. Emptying the box using blind haptic manipulation primitives. Submitted to *IEEE/RSJ Internation Conference on Intelligent Robots and Systems,IROS 2011.*

I Janne Laaksonen and Ville Kyrki. Probabilistic approach to sensor-based grasping. Submitted to *ICRA 2011 Workshop on Manipulation under uncertainty.*

Evaluation of Feature Representation and Machine Learning Methods in Grasp Stability Learning

Janne Laaksonen, Ville Kyrki and Danica Kragic

Abstract— **This paper addresses the problem of sensor-based grasping under uncertainty, specifically, the on-line estimation of grasp stability. We show that machine learning approaches can to some extent detect grasp stability from haptic pressure and finger joint information. Using data from both simulations and two real robotic hands, the paper compares different feature representations and machine learning methods to evaluate their performance in determining the grasp stability. A boosting classifier was found to perform the best of the methods tested.**

I. INTRODUCTION

Grasping a known object in a known environment with a known robotic hand is a tractable problem. But immediately, when some of the facts are unknown, usually true in humanoid robot environments, the problem becomes much more difficult to solve. The problem studied here is how to estimate grasp stability when only haptic information is available. For example, in service robotics the models of objects are usually unknown and must be constructed from e.g. vision. Thus, there is no explicit object model, but the system is learning from haptic images of stable and unstable grasps. We show that it is possible to some extent to recognize when a grasp is stable when given only the haptic pressure and finger joint information.

A number of different sensor modalities can be used to deal with the uncertainty from having an unknown object during grasp. With sensors, we can determine when the object is in contact with the hand, giving additional information besides the kinematic configuration of the hand. Tactile sensors are useful here, as they measure the force or pressure inflicted on the sensor matrix, giving the area of the contact as well as the total force.

To determine the grasp stability, the stability criteria must be linked to the haptic data. This can be done either analytically or through learning. In this paper, we study the use of learning for grasp stability evaluation where a system learns the measure of stability based on a number of examples. Through an experimental study, our aim is to assess the suitability of different feature representations and machine learning methods in the problem of learning grasp stability from haptic input. The focus of the study is to evaluate the grasp stability from a single haptic data instance using both discriminitive and generative classifiers and different feature representations from data-driven dimensionality reduction techniques to application specific feature extraction methods. The approach taken in this paper gives the benefit of detecting whether the grasp is stable or unstable at any instant during grasping knowing neither perfect object information nor the hand kinematics. The approach is also generalizable to any configuration of tactile sensors in the hand which are able to measure pressure level. Both simulated and real data is used to determine the differences and similarities when comparing simulation with real platforms.

The paper is divided into six sections: Section II is a study of related work in the area of the paper, Section III introduces the different features for the classification, Section IV describes the machine learning algorithms used in the experiments and Section V contains the actual performed experiments. Finally Section VI concludes the paper with discussion and future work.

II. RELATED WORK

Grasp stability analysis by analytical means is a well established field. However, to analytically determine the grasp stability, the kinematic configuration of the hand and the contacts between the hand and the object must be perfectly known. Previous studies on this subject are numerous and [1] gives a detailed review. However, the references are useful only in cases when conditions described above are true. When this is the case, it is possible to determine if the grasp is either force or form closure grasp [2], which ensures the stability. Compared to this body of work, we wish to learn the stability from existing data, i.e. the tactile data.

While there is currently little work directly comparable to our work, many have studied the use of tactile and other sensors in a grasping context. Felip and Morales [3] developed a robust grasp primitive, which tries to find a suitable grasp for an unknown object after a few initial grasp attempts. However, only finger force sensors were used in the study.

Apart from using tactile information as a feedback for low level control [4], tactile sensors can be used to detect or identify object properties. Jiméneza et al. [5] use the tactile sensor feedback to determine what kind of a surface the object has, which is then used to determine a suitable grasp for an object. Petrovskaya et al. [6] on the other hand use tactile information to reduce the uncertainty of the object pose, upon an initial contact with the object. In their work, a particle filter is used to estimate object's pose, but the tactile

The research leading to these results has received funding from the European Community's Seventh Framework Programme under grant agreement n° 215821.

J. Laaksonen and V. Kyrki are with Department of Information Technology, Lappeenranta University of Technology, P.O. Box 20, 53851 Lappeenranta, Finland, jalaakso@lut.fi, kyrki@lut.fi; Danica Kragic is with the Centre for Autonomous Systems, Computational Vision and Active Perception Lab, CSC-KTH, 10044 Stockholm, Sweden, dani@kth.se

sensor used to detect contact with the object is not embedded in the gripper performing the grasping.

Object identification has been studied by Schneider et al. [7] and Schöpfer et al. [8] Schneider et al. show that it is possible to identify an object using tactile sensors on a parallel jaw gripper. The approach is very similar to object recognition from images and the object must be grasped several times before accurate recognition rates are achieved. Schöpfer et al. use a tactile sensor pad instead of a gripper or a hand which could be used to grasp the object. [8] is a study on different temporal features which can be used to recognize objects. Similar object recognition systems have been presented in [9], [10].

Preliminary results using the method presented in this paper have been published in [11]. However, this paper takes a significantly broader look into different classifiers and feature reprentations. Learning the grasp stability from examples provides a good ground to cope with the uncertainty in the process generally not studied in the case of analytic approaches.

III. FEATURE REPRESENTATIONS

A haptic data instance, $\mathbf{H} = [\mathbf{t}\ \mathbf{j}]$, consists of the tactile readings, $\mathbf{t}$, and of the grasp joint configuration, $\mathbf{j}$. Depending on the hand used, the dimensionality of both $\mathbf{t}$ and $\mathbf{j}$ changes. In this study, three different platforms are used:

- Simulated Schunk Dextrous Hand (SDH), 3 fingers each with 12x6 tactile elements, $\mathbf{t} \in \mathbb{R}^{216}$, $\mathbf{j} \in \mathbb{R}^7$
- Schunk Dextrous Hand, 3 fingers each with 13x6 tactile elements (Weiss tactile sensors), $\mathbf{t} \in \mathbb{R}^{234}$, $\mathbf{j} \in \mathbb{R}^7$
- Parallel Jaw Gripper, PG70, 2 fingers each with 14x6 tactile elements (Weiss tactile sensors), $\mathbf{t} \in \mathbb{R}^{168}$, $\mathbf{j} \in \mathbb{R}^1$

The dimensionality of $\mathbf{H}$ ranges from $\mathbb{R}^{169}$ to $\mathbb{R}^{241}$ with the listed platforms. The number of features in $\mathbf{H}$ can be considered large and potentially redundant. Thus, an effective method to reduce the dimensionality precedes the subsequent processing. Rest of the section describes the methods that are used to achieve this.

To provide an overview of the effect features have on the classification of the grasp stability, several types of feature representations are studied for training and classification. The features, denoted by $\mathbf{f}$, are derived from the tactile sensor data, $\mathbf{t}$. The features represent a variety of approaches from pure data-driven dimensionality reduction to application specific features. The features are computed from the tactile readings only while the joint configuration is used as is as a part of the haptic features.

A. Principal Component Analysis

Principal component analysis (PCA) is commonly used linear technique for dimensionality reduction. Here, PCA is computed using the covariance of the haptic data, $\mathbf{H}_{1,\ldots,n}$ and the resulting eigenvectors and eigenvalues,

$$C = cov(H_{1,\ldots,n}) \,, \tag{1}$$

$$V^{-1}CV = D \,. \tag{2}$$

Here, V represent the eigenvectors and D the corresponding eigenvalues. We chose the eigenvectors with the largest eigenvalues that combined explain 90% of the data. This results in ~ 60 eigenvectors.

B. Image Moments

Raw image moments are defined as

$$m_{p,q} = \sum_x \sum_y x^p y^q f(x,y) \,. \tag{3}$$

The moments are computed up to order two, that is $(p+q) = o,\ \ o = \{0,1,2\}$, These are related to the total pressure, the mean of the contact area, and the shape of the contact area, indicated by the variance in x- and y-axes. Moments are computed for all tactile sensors individually, thus $\mathbf{f} \in \mathbb{R}^{18}$.

Raw image moments are used in the experiments as normalized image moments did not produce better results. This observation might be due to the fact that, e.g. rotation invariant moments, are not useful for grasp stability learning, as each grasp is unique.

C. Histogram

Histogram representation on the tactile data represents binning of the force affecting each cell of the tactile matrix. This operation also removes all spatial information. Thus, the histogram only considers the distribution of the affecting force. Using 10 histogram bins, $\mathbf{f} \in \mathbb{R}^{10}$.

D. Spatial Partitioning

Spatial partitioning partitions the area of the sensor matrix and sums the affecting force in every cell of the sensor matrix in each of these partitions. In essence, this subsamples the tactile image of each sensor matrix. Partitioning can be thought as opposite to the histogram operation, as partitioning retains the spatial information but loses some information of the force distribution. In the experiments, a 2x2 grid is used to partition the tactile image on each sensor, $\mathbf{f} \in \mathbb{R}^{12}$.

E. Local Binary Pattern

Local binary patterns (LBPs) [12] are used commonly for texture classification but also on face recognition. As its name suggests, local binary pattern codes local changes in a binary code. The local changes are found by thresholding the pixel neighbourhood by the value of the center pixel and checking which pixels are above the threshold. These binary codes are then added to a histogram, which is the final feature representing the original data. Images from all sensors are coalesced into one image and the LBP is applied to this image in the experiments. In the experiments, LBP produces a histogram where $\mathbf{f} \in \mathbb{R}^{59}$.

F. Row and Column Sums

Row and column sums is another form of spatial feature representation, where the colums and rows are summed independent of each other, thus, the resulting dimensionality of the feature representation is the sum of the tactile sensor dimensions, $i + j$, for each sensor,

$$sum_{c_i} = \sum_j t_{ij} \,, \tag{4}$$

$$sum_{r_j} = \sum_i t_{ij} \,, \tag{5}$$

where sum_{c_i} denotes the individual sensor columns and sum_{r_j} denotes the sensor rows.

IV. CLASSIFIERS

From a classification point of view, the problem of classifying grasp stability may be modelled as a classical two-class problem. Thus, the stability is classified as either stable or unstable. This is possible to implement with most of the basic classifiers without extending the theories behind them.

In the work presented here, a number of classifiers have been selected for the experiments. All the classifiers described represent different types of machine learning algorithms that help to understand the underlaying problem in grasp stability classification. In particular, we study both discriminative and generative approaches for classification.

A. Support Vector Machine

As the problem of grasp stability is binary, support vector machine (SVM) classification [13], [14] is suitable for the problem. Thus, here the focus is on the 2-class SVM. SVM is a maximum margin classifier, i.e. the classifier fits the decision boundary so that maximum margin between the classes is achieved. This guarantees that the generalization ability between the classes is not lost during the training of the SVM classifier.

Another feature of the SVM is the ability to use non-linear classifiers instead of the original linear hyper-plane classifier. Non-linearity is achieved using different kernels and in this study radial basis function (RBF) is used as the kernel for SVM:

$$K(x_i, x_j) = e^{-\gamma \|x_i - x_j\|^2}, \quad for \quad \gamma > 0, \tag{6}$$

In addition to the parameter γ, constant C, related to the penalty applied to incorrectly classified training samples [13], needs to be set. The parameters can be found by searching the parameter space to find the optimal values. In this study, as an extension to the basic two-class SVM, probabilistic outputs for SVM by Platt [15] are used to analyze the results given by the SVM. The implementation of the SVM is by Chang and Lin [16].

B. Gaussian Mixture Model Classifier

As the naive Bayes classifier assumes that the data is distributed according to some modelable distribution, it is not optimal in cases where this assumption is not true. The haptic data is distributed according to an unknown distribution, thus it is reasonable to use Gaussian mixture model (GMM) statistical classifier.

While GMM methods assume a Gaussian distribution, GMM uses multiple Gaussian distributions to model the data which enables the methods to model multi-modal and more complex data. The implementation used in the experiments is by Paalanen and Kämäräinen [17].

TABLE I

TABLE OF PARAMETERS FOR FEATURES.

Features	Parameter	Parameter
Raw	-	-
PCA	-	-
Histogram	No. bins: 10	-
LBP	Uniform LBP	Samples: 8,1
Moments	-	-
Partitioning	Grid: 2x2	-
R&C sums	-	-

C. k-Nearest Neighbour

k-nearest neighbour [18] classifier is a very simple algorithm to implement. This classifier requires no training phase, instead during the classification phase, the test samples are compared to all given training samples. The test sample is classified as the class with the most neighboring, i.e. closest, training samples. The k denotes the number neighbouring training samples that are used in the classification phase. *k*-nearest neighbour also has a proven [18] error rate that is no worse than two times the error rate of an optimal classifier when the amount of data approaches infinity.

D. AdaBoost

AdaBoost or adaptive boosting is a meta-algorithm for learning which was developed by Freund and Schapire [19]. Adaboost uses multiple weak classifiers, such as linear hyperplane classifiers, to classify the given training data. AdaBoost has a good generalization ability, however AdaBoost is not effective when outliers are present in the training data.

The AdaBoost-algorithm that is used in this study is based on a decision tree classifier with a variable branching factor. With a branching factor of 1, the tree classifier represents a linear hyperplane classifier. The implementation is by Vezhnevets [20].

V. EXPERIMENTS

The goal of the experiments is to study the effect of the presented features in conjunction with multiple different classifier methods. A number of different datasets with different assumptions are used in the experiments to determine what type of data is suitable for classification.

A. Experimental Setup

The parameters for features and classifiers are shown in tables I and II. The raw data from the tactile sensors is also used as features in addition to the features presented in Section III. The parameters were found by a parameter search across reasonable parameter space. Schunk Dextrous Hand hardware and objects used in the grasping experiments are shown in Figure 1.

The following datasets have been chosen from simulated data, which were generated using simulated SDH hand model in a simulation environment described in [21]:

- D_1, a cylinder, grasps sampled from the side
- D_2, a bottle, grasps sampled from the side
- D_3, a bottle, grasps sampled from the top
- D_4, a cylinder, grasps sampled from a sphere

TABLE II

TABLE OF PARAMETERS FOR CLASSIFIERS.

Classifier	Parameter	Parameter
SVM	C: 0.4	γ: 0.03
GMM	max. clusters: 19	max. error: 0.016
KNN	k: 3	-
AdaBoost	Branch factor: 1	-

- D_5, a bottle, grasps sampled from a sphere

The datasets $D_{1,2,3}$ represent cases where we know the pose of the object with some accuracy, and can plan for a grasp. The datasets $D_{4,5}$ are simulating situations where the position of the object is known to some extent but the orientation is highly uncertain, thus, the grasps are sampled from a sphere around the object. In the simulated data, the grasp stability computation is based on [22], but instead of one convex hull W, two convex hulls, W_f and W_τ are used to separate wrench space with respect to forces and torques, and additional constraints are placed on W_f, so that

$$\alpha(m \cdot \mathbf{g}) \in W_f \, , \alpha = 1.1 \, . \tag{7}$$

This allows the grasp to remain stable even if some additional forces are introduced in addition to the gravity. Datasets generated with real hands are following:

- D_6, a cylinder, grasps sampled from the side, SDH
- D_7, a bottle, grasps sampled from the side, SDH
- D_8, a bottle, grasps sampled from the top, SDH
- D_9, a box, grasps sampled from the side, PG70
- D_{10}, a shampoo bottle, grasps sampled from the side, PG70
- D_{11}, a shampoo bottle, grasps sampled from the top, PG70

Datasets $D_{6,\cdots,11}$ represent cases where an estimate of the object's pose is known, for example, from a vision system. This estimate is commonly noisy and thus we added the noise to the hand pose. The objects in datasets $D_{6,7,8,9}$ are rigid and the objects in datasets D_{10} and D_{11} are non-rigid, i.e. the objects are deformable. The grasp stability in these datasets was determined by grasping an object. In datasets $D_{6,\cdots,8}$ the object was rotated $[-120°, +120°]$ around the approach direction and in datasets $D_{9,\cdots,11}$, the object was lifted and rotated $+90°$ around X and Y axes, where Z axis is the direction of lift. If the object moved independently of the hand, the grasp was unstable, otherwise it was stable.

The method used to evaluate the performance of the classifiers was 10-fold cross validation. The dataset sample size for each of the given datasets are shown in Table III with the maximum classification rate summarized from Tables IV and V. The sample size shown in the table is balanced, so that each dataset has equal amount of stable and unstable grasp samples. All other features were normalized to zero-mean and unit variance, except the raw features which were normalized to range $[0, 1]$. The normalization parameters were obtained from the training set and applied to both training and test sets.

TABLE III

DATASET SAMPLE SIZES AND CLASSIFICATION RATES.

Dataset	Sample size	Max. classification rate
D_1	6400	77.0%
D_2	4906	61.4%
D_3	4446	62.7%
D_4	5302	80.4%
D_5	8990	70.5%
D_6	140	92.1%
D_7	100	92.1%
D_8	50	84.6%
D_9	148	74.6%
D_{10}	148	59.0%
D_{11}	100	64.0%

B. Experimental Results

Result matrix with the described datasets is given in Table IV and Table V. The table shows the classification rate of each dataset with the indicated feature and classifier combination. Each row shows the best classifier in **bold** font and worst in *italic* font. The best and worst classifiers were determined on a 95 % confidence interval using the Agresti-Coull interval which approximates the binomial confidence interval. Multiple classifiers are deemed best if there is no statistically significant difference in the classification performance between them. Some results for GMM are omitted because of the training sample size requirements, thus, results for datasets $D_{6,\ldots,11}$ are not shown.

The results in Tables IV and V show that there is a distinctive performance difference between the datasets. Simulated datasets, D_1 and D_4 perform usually better than the other simulated datasets. This performance gap is caused, at least partially, by the hand configuration, which allows the object to touch other areas of the hand where there are no sensors. This removes some of the important information about the object to be used in determining the grasp stability. Thus, it is important to set up the grasp sequence in a way that allows the sensored part of the hand to grasp the object.

This procedure is evident in the dataset gathered from the real hands, especially sets $D_{6,7,8}$, where the classification performance is above 75 % in some cases. However, the object in the datasets were rigid, which is not the case in sets $D_{10,11}$. These sets show mostly poor performance, indicating that further samples must be used to learn the grasp stability.

The best overall classifier is AdaBoost, which performs the best out of the four classifiers, while SVM is close second. Worst classifier is GMM, partially due to the extensive amount of data needed to train GMM successfully with some of the chosen features. Low amount of data available in datasets $D_{6,\ldots,11}$ makes it difficult to determine within the 0.95 confidence interval the best classifier, but looking at the results, AdaBoost has the highest mean in these cases. SVM has some anomalies, these are suspected to be caused by the parameter and feature combinations, and could be fixed by adjusting the parameters of SVM.

C. Feature Study

To study the effect of the features on the classification rate, tests with a 3-nearest neighbour classifier were conducted on

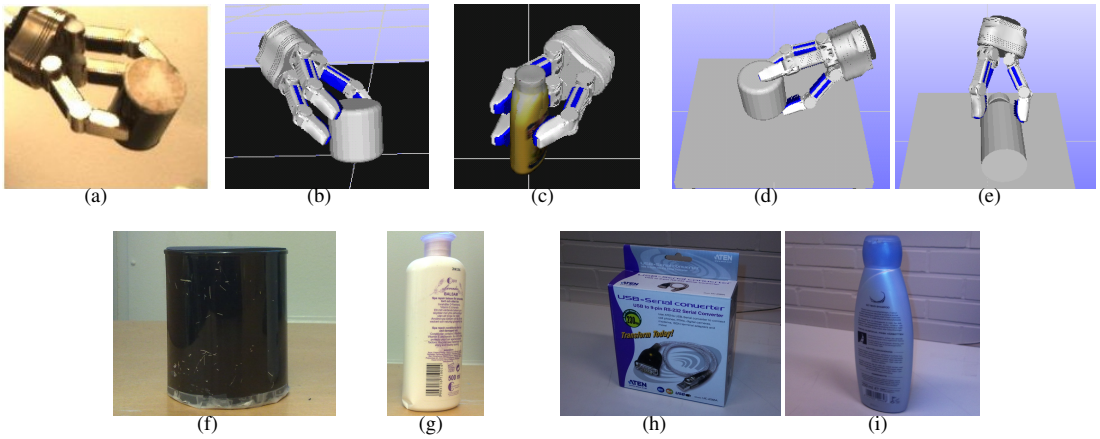


Fig. 1. Hardware and objects used in the datasets: (a) 3-finger SDH; (b) D_1; (c) D_2,D_3; (d) D_4; (e) D_5; (f) D_6; (g) D_7,D_8; (h) D_9; (i) D_{10}, D_{11}.

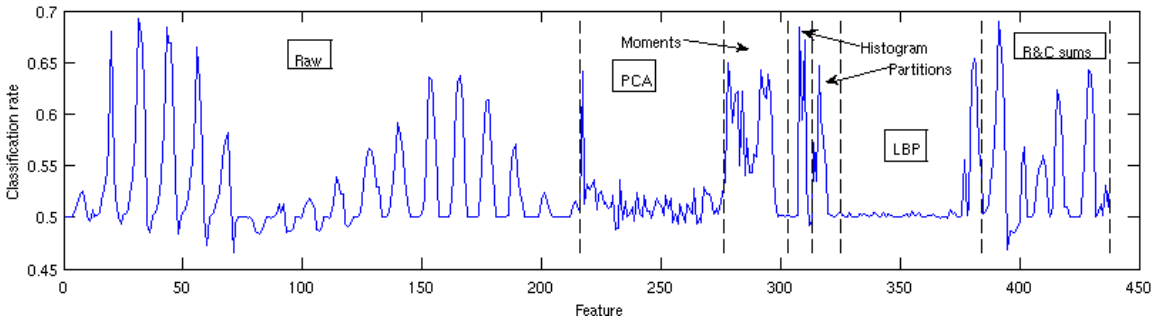


Fig. 2. Classification rates on individual features.

each dimension of all the feature representations described in Section III and also on the raw tactile data. The classification rates are shown in Figure 2 for the dataset D_1.

Classification rates of 0.5 or less in Figure 2 are a sign that the feature used is not particularly useful in learning as it has no correlation with the grasp stability. The figure shows that there are quite many useful features in the set of features that were tested. What is interesting is the raw data as it has multiple spikes which are among the best features for classifying the grasp stability. This indicates that individual cells of the tactile sensors can be used to determine the grasp stability to some extent. Also image moments, histogram and row and column sums seem to have a number of good features to use for classifying. The experiment was also performed on the real data set D_6, for which the results were similar.

VI. CONCLUSIONS AND FUTURE WORK

The focus of the presented work was to investigate how different machine learning methods and feature representations affect the ability to learn and assess the grasp stability from haptic data. Both simulated and real world data was used in an experimental comparison. Experiments indicated that AdaBoost was the best performing classifier, suggesting that boosting approaches would be likely candidates for further studies in the context of grasp stability learning.

The classification performance varied significantly between different data sets. Results of the experiments showed that deformable objects are more difficult to learn with a similar sample size compared to rigid objects. A temporal approach might be useful for deformable objects, as it could extract more information from the grasp, as in [9]. Data also show that if the grasped object has contacts with the hand outside of the tactile matrices, the grasp stability can not be learned effectively. It needs to be noted that perfect classification performance is not necessary, since acceptance threshold can be set such that for example regrasping is triggered in ambiguous cases.

Future work will concentrate on expanding the presented study. Especially the study on deformable objects is interesting as currently there are no grasping simulators that are able to do this, but many household objects have this property. It is also possible to combine data from multiple objects to produce a common classifier for all the objects. Further research on this subject would help to identify the limits of the presented learning approach on completely unknown objects.

REFERENCES

[1] A. Bicchi and V. Kumar, "Robotic grasping and contact: A review," in *ICRA*, 2000, pp. 707–714.

[2] D. Prattichizzo and J. C. Trinkle, "Grasping," in *Springer Handbook of Robotics*, 1st ed., B. Siciliano and O. Khatib, Eds. Berlin, Germany: Springer-Verlag, 2008.

[3] J. Felip and A. Morales, "Robust sensor-based grasp primitive for a three-finger robot hand," in *IEEE/RSJ International. Conference on Intelligent Robots and Systems*, Oct. 2009.

TABLE IV

CLASSIFICATION RATES FOR DATASETS $D_{1,\cdots,11}$.

Data, Feature	SVM	GMM	KNN	AdaBoost
D_1	**75.5%**	-	*73.3%*	**76.7%**
D_2	**59.1%**	-	**56.6%**	**58.1%**
D_3	**60.3%**	-	**60.7%**	**62.1%**
D_4	69.7%	-	*63.1%*	**79.3%**
D_5, Raw	65.7%	-	*58.2%*	**68.9%**
D_6	**82.6%**	-	**90.7%**	**91.4%**
D_7	*22.0%*	-	**84.0%**	**91.0%**
D_8	*49.3%*	-	**84.6%**	**80.4%**
D_9	*54.0%*	-	**71.3%**	**71.1%**
D_{10}	**49.3%**	-	**48.6%**	**46.4%**
D_{11}	**50.0%**	-	**54.0%**	**56.0%**
Mean	66.3%	-	*62.6%*	**69.6%**
D_1	**77.0%**	74.1%	*72.5%*	74.5%
D_2	**59.7%**	56.5%	*56.1%*	57.0%
D_3	**61.3%**	**60.4%**	**59.7%**	**60.4%**
D_4	74.0%	68.7%	*67.5%*	**77.6%**
D_5, PCA	**67.6%**	64.5%	*60.4%*	**67.7%**
D_6	**85.7%**	-	*58.6%*	**90.0%**
D_7	**77.0%**	-	*55.0%*	**69.0%**
D_8	50.0%	-	*47.9%*	**78.6%**
D_9	**73.2%**	-	**65.5%**	**71.7%**
D_{10}	**50.0%**	-	**54.0%**	**54.0%**
D_{11}	**46.0%**	-	**55.0%**	**49.0%**
Mean	**68.5%**	65.4%	*63.3%*	**68.1%**
D_1	**76.5%**	*71.1%*	72.5%	**75.9%**
D_2	**61.1%**	*52.7%*	57.4%	**58.6%**
D_3	**62.7%**	*54.0%*	**60.1%**	**62.3%**
D_4	**80.0%**	*61.2%*	72.3%	**79.7%**
D_5, Moments	**70.5%**	*51.8%*	63.6%	**68.9%**
D_6	**92.1%**	-	**93.6%**	**90.7%**
D_7	**91.0%**	-	**86.0%**	**92.0%**
D_8	*27.1%*	-	**67.1%**	**77.9%**
D_9	**64.6%**	-	**69.5%**	**72.7%**
D_{10}	**44.0%**	-	**50.5%**	**44.7%**
D_{11}	**48.0%**	-	**51.0%**	**64.0%**
Mean	**70.6%**	*58.0%*	65.6%	**69.7%**
D_1	**73.1%**	*64.9%*	65.6%	**73.9%**
D_2	**56.0%**	**55.0%**	*52.2%*	**56.4%**
D_3	**62.0%**	*49.9%*	57.6%	**62.1%**
D_4	**79.0%**	71.9%	*69.8%*	**79.4%**
D_5, Histogram	**67.9%**	**66.8%**	*62.1%*	**68.5%**
D_6	**90.0%**	-	**81.4%**	**90.0%**
D_7	**84.0%**	-	**76.0%**	**82.0%**
D_8	**66.1%**	-	**73.6%**	**72.5%**
D_9	**63.0%**	-	**53.8%**	**57.3%**
D_{10}	**57.6%**	-	**49.2%**	**59.0%**
D_{11}	*38.0%*	-	**62.0%**	**57.0%**
Mean	**68.1%**	62.9%	*62.0%*	**68.7%**

TABLE V

CLASSIFICATION RATES FOR DATASETS $D_{1,\cdots,11}$.

Data, Feature	SVM	GMM	KNN	AdaBoost
D_1	**76.1%**	*65.6%*	71.8%	**75.8%**
D_2	**57.3%**	**55.0%**	**56.2%**	**57.3%**
D_3	**62.6%**	*53.2%*	59.0%	**60.8%**
D_4	**80.4%**	*65.7%*	73.0%	**79.7%**
D_5,Partitions	**69.0%**	*60.9%*	64.7%	**68.9%**
D_6	**91.3%**	-	**91.4%**	**92.1%**
D_7	**91.0%**	-	**89.0%**	**89.0%**
D_8	*36.8%*	-	**81.8%**	**67.5%**
D_9	**63.0%**	-	**60.6%**	**64.4%**
D_{10}	**40.8%**	-	**45.5%**	**48.8%**
D_{11}	**56.0%**	-	**48.0%**	**64.0%**
Mean	**69.6%**	*60.6%*	65.5%	**69.2%**
D_1	**75.0%**	*64.4%*	68.6%	**74.9%**
D_2	**54.8%**	52.0%	*51.4%*	**56.4%**
D_3	**60.9%**	*58.0%*	58.1%	**61.3%**
D_4	75.2%	66.4%	*64.0%*	**79.7%**
D_5, LBP	66.4%	58.7%	*57.8%*	**68.4%**
D_6	**84.3%**	-	**79.3%**	**85.0%**
D_7	*26.0%*	-	**68.0%**	**74.0%**
D_8	**47.1%**	-	**68.2%**	**60.0%**
D_9	**61.1%**	-	**69.2%**	**73.4%**
D_{10}	**50.2%**	-	**45.8%**	**48.3%**
D_{11}	**50.0%**	-	**49.0%**	**51.0%**
Mean	66.8%	*60.1%*	60.3%	**68.7%**
D_1	**76.8%**	*63.8%*	72.1%	**76.5%**
D_2	**61.4%**	58.6%	*57.8%*	58.3%
D_3	**62.7%**	**61.5%**	**61.5%**	**60.7%**
D_4	77.3%	*58.9%*	70.2%	**79.6%**
D_5, R&C Sums	**68.7%**	63.4%	*62.6%*	**68.8%**
D_6	**92.1%**	-	**92.1%**	**91.4%**
D_7	**90.0%**	-	**87.0%**	**91.0%**
D_8	*30.7%*	-	**72.1%**	**68.2%**
D_9	**63.5%**	-	**67.5%**	**74.6%**
D_{10}	**55.1%**	-	**50.3%**	**43.8%**
D_{11}	**54.0%**	-	**52.0%**	**64.0%**
Mean	**69.8%**	*61.6%*	65.1%	**69.5%**

[4] T. Tsuboi and et al., "Adaptive grasping by multi fingered hand with tactile sensor based on robust force and position control," in *IEEE International Conference on Robotics and Automation*, 2008, pp. 264–271.

[5] A. Jiméneza, A. Soembagijob, D. Reynaertsb, H. V. Brusselb, R. Ceresa, and J. Ponsa, "Featureless classification of tactile contacts in a gripper using neural networks," *Sensors and Actuators A: Physical*, vol. 62, no. 1-3, pp. 488–491, 1997.

[6] A. Petrovskaya, O. Khatib, S. Thrun, and A. Y. Ng, "Bayesian estimation for autonomous object manipulation based on tactile sensors," in *ICRA*, 2006, pp. 707–714.

[7] A. Schneider, J. Sturm, C. Stachniss, M. Reisert, H. Burkhardt, and W. Burgard, "Object identification with tactile sensors using bag-of-features," in *In Proc. of the International Conference on Intelligent Robot Systems (IROS)*, 2009.

[8] M. Schöpfer, M. Pardowitz, and H. J. Ritter, "Using entropy for dimension reduction of tactile data," in *14th International Conference on Advanced Robotics*, ser. Proceedings of the ICAR 2009, IEEE. Munich, Germany: IEEE, 22/06/2009 2009.

[9] S. Chitta, M. Piccoli, and J. Sturm, "Tactile object class and internal state recognition for mobile manipulation," in *In Proceedings of the IEEE International Conference on Robotics and Automation*, 2010.

[10] N. Gorges, S. E. Navarro, D. Göger, and H. Wörn, "Haptic object recognition using passive joints and haptic key features," in *In Proceedings of the IEEE International Conference on Robotics and Automation*, 2010.

[11] Y. Bekiroglu, J. Laaksonen, J. A. Jorgensen, and V. Kyrki, "Learning grasp stability based on haptic data," in *Robotics: Science and Systems (RSS 2010) Workshop on Representations for Object Grasping and Manipulation in Single and Dual Arm Tasks*, 2010.

[12] T. Ojala, M. Pietikäinen, and T. Mäenpää, "Multiresolution gray-scale and rotation invariant texture classification with Local Binary Patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 971–987, 2002.

[13] C. Cortes and V. Vapnik, "Support vector networks," *Machine Learning*, vol. 20, pp. 273–297, 1995.

[14] V. Vapnik, *The Nature of Statistical Learning Theory*. Springer-Verlag, 1995.

[15] J. C. Platt, "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods," in *Advances in Large Margin Classifiers*. MIT Press, 1999, pp. 61–74.

[16] C.-C. Chang and C.-J. Lin, *LIBSVM: a library for support vector machines*, 2001, software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.

[17] P. Paalanen and J.-K. Kämäräinen, *GMMBAYES - Bayesian Classifier and Gaussian Mixture Model ToolBox*, 2004, available at http://www2.it.lut.fi/project/gmmbayes/downloads/src/gmmbayestb/.

[18] T. M. Cover and P. E. Hart, "Nearest neighbor pattern classification," *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21–27, 1967.

[19] Y. Freund and R. E. Schapire, "Experiments with a new boosting algorithm," in *Thirteenth Internation Conference on Machine Learning*. Morgan Kaufmann, 1996, pp. 148–156.

[20] A. Vezhnevets, *GML AdaBoost Matlab Toolbox*, 2006, available at http://graphics.cs.msu.ru/ru/science/research/machinelearning/ adaboosttoolbox.

[21] J. A. Jorgensen and H. G. Petersen, "Usage of simulations to plan stable grasping of unknown objects with a 3-fingered schunk hand," in *IROS'08 Workshop on Robot Simulators*, Nice, France, September 2008. [Online]. Available: http://www.robot.uji.es/research/events/iros08/contributions/petersen.pdf

[22] A. T. Miller and P. K. Allen, "Examples of 3d grasp quality computations," in *IEEE International Conference on Robotics and Automation*, 1999, pp. 1240–1246.

Learning grasp stability based on tactile data and HMMs

Yasemin Bekiroglu, Danica Kragic and Ville Kyrki

Abstract— **In this paper, the problem of learning grasp stability in robotic object grasping based on tactile measurements is studied. Although grasp stability modeling and estimation has been studied for a long time, there are few robots today able of demonstrating extensive grasping skills. The main contribution of the work presented here is an investigation of probabilistic modeling for inferring grasp stability based on learning from examples. The main objective is classification of a grasp as stable or unstable before applying further actions on it, e.g. lifting. The problem cannot be solved by visual sensing which is typically used to execute an initial robot hand positioning with respect to the object. The output of the classification system can trigger a regrasping step if an unstable grasp is identified. An off-line learning process is implemented and used for reasoning about grasp stability for a three-fingered robotic hand using Hidden Markov models. To evaluate the proposed method, experiments are performed both in simulation and on a real robot system.**

I. Introduction

For a general purpose service robot, operating in an industrial or a domestic environment, object grasping and manipulation skills are a necessity. Most of the today's robot systems, however, demonstrate only limited object grasping and manipulation capabilities. The classical work in robotic grasping assumes that the object parameters such as pose, shape, weight and material properties are known. If precise knowledge of these is available, grasp stability estimation using analytical approaches is often enough for successful grasp execution. However, in unstructured environments that information is usually uncertain, which presents a challenge for the current systems.

To cope with the uncertainty, one can rely on sensory information for closed loop control [1]. For grasping and manipulation, shape and pose of an object are important inputs to the control loop. However, the accuracy of vision is limited and small errors in object pose can cause failures. These failures are difficult to prevent at the grasp planning stage and need to be taken into account once the contact with the object has been made. Visual servoing approaches [2], [3] can solve these problems only to a certain extent since they commonly need a desired pose with respect to the object to be defined beforehand which is impossible for unknown objects. While the tactile and force sensors can be used to reduce the uncertainty upon contact, a grasp may fail even when all fingers have adequate contact forces. The major issue is that for unknown objects, grasps need to be evaluated from data the robot can extract on-line. Besides the incomplete information about the environment and the objects, there is also a lack of generalizable quality measures for grasp stability assessment under uncertainty.

We present a learning system that infers grasp stability based on tactile sensors. If an unstable grasp is detected, a regrasping step can be initialized before, for example, lifting the object. To achieve a good generalization performance, machine learning approaches typically require large amount of training data. As a solution to the problem of acquiring enough training data, we propose to first simulate the grasping process. Then, we evaluate the feasibility of the approach both on simulated and real data. We have implemented a time-series analysis based on a sequence of tactile measurements with the purpose of investigating the effect of the dynamic process of grasp execution on grasp stability. The results show that the idea of exploiting a learning approach is feasible. The additional contribution of the work is a publicly available database of the experimental sequences [4].

The paper is organized as follows. Related work is reviewed in Section II and the notation summarized in Section III. Then, Section IV introduces the time-series recognition approach using Hidden Markov models. In Section V, the process of generation of the training data is described. Section VI presents the experimental results. Finally, we conclude and discuss directions for future research in Section VII.

II. Related Work

During the last few decades, there has been a significant amount of work reported in robotic object grasping, see [5] for a recent survey. In our previous work, we have integrated vision based object recognition and tactile sensing for closed loop grasp control [1]. Regarding vision based approaches, a number of proposed solutions rely on object recognition and/or shape registration. This commonly requires a database of objects or shapes, as for example in [6], or even of objects combined with grasps, as presented in [7].

The feedback from tactile sensors has been used to maximize the contact surface for removing a book from a bookshelf [8]. In [9], the integration of force, visual and tactile feedback has been proposed for an application of opening a sliding door. The main difference between the above approaches and the work presented here is that we concentrate on using the tactile sensors for assessment of

Y. Bekiroglu and D. Kragic are with the Centre for Autonomous Systems and Computational Vision and Active Perception Lab, School of Computer Science and Communication, KTH, Stockholm, Sweden. V. Kyrki is with the Department of Information Technology, Lappeenranta University of Technology, Finland. yaseminb,danik@csc.kth.se, kyrki@lut.fi

This work was supported by EU through the project CogX, IST-FP6-IP-027657, and GRASP, IST-FP7-IP-215821 and the Swedish Foundation for Strategic Research.

grasp stability. Thus, rather than using the tactile data for control, we reason about the stability before starting to actively manipulate the object.

There have been many examples of grasp planning demonstrated in simulation. Their commonality is the use of a strategy that relies on known object shape and/or pose. Modeling object shape with a number of primitives such as boxes and cylinders [10], or superquadrics [11] reduces the space of grasp hypotheses. The decision about the most suitable grasp is based on grasp quality measures given contact positions. However, these techniques do not deal with uncertainties that may arise in realistic scenarios.

The work of integrating learning with grasping is also related to understanding human grasping strategies. In [12], we have demonstrated how a robot system can learn grasping strategies from human demonstration using a grasp experience database. The human grasp was recognized with the help of a magnetic tracking system and mapped to the kinematics of the robot hand using a predefined lookup-table. More recent work uses vision based grasp recognition in a learning-by-demonstration framework [13]. The recent learning approaches using tactile sensors are focused on either determining the shape properties of objects [14] or object recognition [15], [16].

To our knowledge, the analysis of grasp stability using Hidden Markov models and tactile sensors presented in this paper has not been studied before.

III. Feature Representation

As mentioned, the goal of the paper is to show how grasp stability can be assessed based on temporal sequences of tactile data using Hidden Markov models. The basic idea is to position a hand with respect to an object so that a grasp can be obtained by closing the fingers. A robot hand is equipped with two-dimensional tactile patches at the fingertips. Tactile measurements are recorded from the moment the first contact with the object is obtained and until there is no change in the measurements detected. The whole measurement sequence is denoted by $x_1^i, \ldots, x_{T_i}^i$. For comparison reasons, we will also present results of one-shot classification based only on a single tactile measurements, $x_{T_i}^i$, taken at the end of a grasping sequence. The data is generated both in simulation and on real hardware and will be presented in more detail in Section V. The notation used in this paper is as follows:

- $D = [o_i], i = 1, \ldots, N$ denotes a data set with N observation sequences.
- $o_i = [x_t^i], t = 1, \ldots, T_i$ is an observation sequence.
- $x_t^i = [M_f^{i,t} j_v^{i,t}], f = 1, \ldots, F, v = 1, \ldots, V$ is the observation at time instant t given the i-th sequence; F is the number of tactile sensors and V is the number of joints of the robot hand.
- $M_f^{i,t}$ includes the moment features extracted from the tactile readings $H_f^{i,t}$ on the sensor f at time instant t given the i-th sequence. Details about the extraction of these features are given later in this section.
- $j_v^{i,t}$ is a joint angle at time instant t given the i-th sequence.

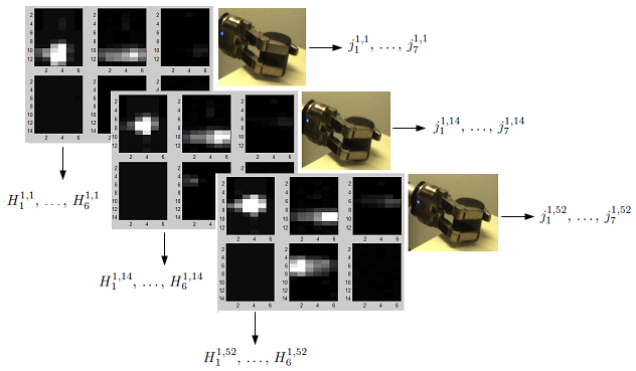


Fig. 1. An example grasping sequence of a cylinder and the corresponding tactile measurements.

The acquired data consists thus of tactile readings $H_f^{i,t}$ and joint angles of the hand $j_v^{i,t}$. In simulation, the data originates from three tactile sensors: one per finger given the Schunk Dextrous Hand (SDH). Each sensor produces 12×6 tactile measurements and there are additionally seven parameters representing the pose of the hand given the joint angles. For the real world data, we used two different robot hands. For the Schunk Dextrous Hand, we store $3 \times (14 \times 6)$ readings on proximal and $3 \times (13 \times 6)$ on distal sensors. The second robot hand is a parallel 2-fingered gripper that is equipped with the same type of tactile sensors and thus delivers $2 \times (14 \times 6)$ readings. Example images from the sensors are shown in Figure 1. The tactile images in the figure represent a stable grasp of a cylinder.

The tactile data is relatively high dimensional and to some extent redundant. Therefore, we start by representing the acquired data as features. Here, we borrow some ideas from image processing and consider the two-dimensional tactile patches as images. We employ image moments as a suitable representation which also reduce the dimensionality. The general parameterization of image moments is given by

$$m_{p,q} = \sum_z \sum_y z^p y^q f(z,y) \tag{1}$$

where p and q represent the order of the moment, z and y represent the horizontal and vertical position on the tactile patch, and $f(z,y)$ the measured contact. We compute moments up to order two, $(p+q) \in \{0,1,2\}$, for each sensor array separately. These then correspond to the total pressure and the distribution of the pressure in the horizontal and vertical direction.

First and second order moments are included in the feature vector according to Equation (1). Two additional features are computed for each tactile sensor: the size of the contact area ($area$) and the center of the contact $(\frac{m_{1,0}}{m_{0,0}}, \frac{m_{0,1}}{m_{0,0}})$. We normalize the zeroth order moment by calculating the average pressure $m_{0,0}/area$. Thus, there are in total nine features for each sensor resulting in an observation $x_t^i \in \mathbb{R}^{9F+V}$.

Normalizing the feature vector is a common step in machine learning methods. In our case, moment features

and finger joint angles are normalized to zero-mean and unit standard deviation. Normalization parameters are calculated from the training data and then used to normalize the testing sequences.

IV. Theoretical framework

This section presents the basics of the Hidden Markov models (HMMs) [17] and their application in our work. We train two HMMs: one that represents stable grasps and one that represents unstable ones. Recognition is then performed using the classical forward procedure: evaluating the likelihood given both models and the final decision is based on maximizing the estimated likelihood.

For the HMM, we use the notation $\lambda = (\pi, A, B)$ where π denotes the initial probability distribution, A is the transition probability matrix

$$A = a_{ij} = P(S_{t+1} = j | S_t = i), i = 1 \dots N, j = 1 \dots N \tag{2}$$

and B defines output (observation) probability distributions

$$b_j(x) = f_{X_t|S_t}(x|j) \tag{3}$$

Here, $X_t = x$ represents a feature vector for any given state $S_t = j$. The structure of an HMM can be ergodic or left-to-right, which determines the structure of A. In the following, we present and evaluate both of these models.

A. Modeling Observations

The estimation of the HMM model parameters is based on the Baum-Welch procedure. The output probability distributions are modeled using Gaussian Mixture Models (GMMs):

$$f_X(x) = \sum_{k=1}^{K} w_k \frac{1}{2\pi^{L/2}\sqrt{|C_k|}} e^{-\frac{1}{2}(x-\mu_k)^T C_k^{-1}(x-\mu_k)} \tag{4}$$

where $\sum_{k=1}^{K} w_k = 1$, μ_k is the mean vector and C_k is the covariance matrix for the k-th mixture component. The unknown parameters $\theta = (w_k, \mu_k, C_k : k = 1...K)$ are estimated from the training sequences $o = (x_1, ...x_T)$.

Initial estimates of the observation densities in (Eq. 4) affect the point of convergence of the reestimation formulas. Depending on the structure of the HMM, we employ different initialization methods for the parameters of the observation densities. The two initialization procedures are denoted by $Init_1$ and $Init_2$:

- $Init_1$: For an ergodic HMM, observations are clustered using k-means. Here, k is equal to the number of states in the HMM and each cluster is modeled with a GMM using standard Expectation Maximization. Initial parameters for the GMMs are found in the standard fashion using the k-means algorithm.
- $Init_2$: For a left-to-right HMM, each observation sequence is divided temporally into equal length subsequences. Then, each GMM is estimated from the collection of corresponding subsequences. Thus, the GMMs represent the temporal evolution of the observations. Initial parameters for the GMM estimation are found identically to $Init_1$.

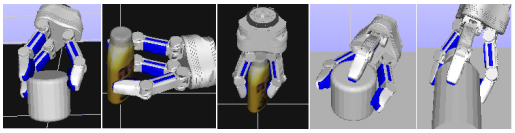


Fig. 2. Example grasps on different objects from five simulated datasets denoted by (D_{S_1}), (D_{S_2}), (D_{S_3}), (D_{S_4}), (D_{S_5}) in the text.

V. Data generation

The data was generated both in simulation environment and using real robotic hands. Both in real and simulated setups, a grasping sequence is recorded from tactile readings and corresponding joint configurations from the first contact with an object is made until a static state is achieved. After placing the hand in front of an object in a fully open position, the fingers are controlled to a closing position with equal velocity. By a static state, we consider a state when the tactile sensors do not report any change or fully closed hand configuration has been reached. The latter can occur only in the case the object was dropped.

The simulated data was generated to investigate two aspects of grasp stability recognition: shape specific and shape independent stability recognition. For the shape specific recognition, the grasping strategies vary for each shape and it is assumed that the system has the knowledge about the shape prior to grasping from, for example, a vision system. The type of grasps generated on objects of known shapes can easily be generated by a grasp planning system.

For the shape independent approach, no knowledge of the object except the approximate position of its center of mass with respect to the hand is considered. Since the knowledge of the object shape is unknown, there will be larger variation in the contact space and therefore more uncertainty in the learning process compared to the shape specific case. The training data for this approach has been generated by sampling the grasps on a unit sphere with the origin in the object center. Example grasps are shown in Figure 2.

For the shape specific approach, simulated datasets D_{S_1}, D_{S_2}, D_{S_3} are generated on a cylindrical object and a bottle. Here, two types of grasps have been applied: a side and a top grasp. D_{S_1} and D_{S_2} include side grasps (for both objects) and D_{S_3} includes top grasps (for the bottle). Simulated datasets D_{S_4}, D_{S_5} are generated on a cylinder and a bottle by applying approach vectors sampled from a sphere around the object and including more than one preshape.

For labeling of the simulated grasp sequences we use a grasp quality measure based on the radius of the largest enclosing ball in the unit grasp wrench space (GWS) constructed as proposed in [18]. Two convex hulls, W_f and W_τ are calculated to separate wrench space with respect to forces and torques. Stable grasps are defined as those for which both quality values are within a threshold which has been set experimentally. The threshold for force is proportional to the weight of the object so that the grasp remains stable even in case of additional forces.

The main purpose of the real world experiments is to demonstrate that the idea of grasp stability recognition is applicable in real-world scenarios. Thus, the experiments aim to serve as a proof-of-concept rather than assessing the exact performance rates in different use cases. We believe that performing real world experiments is important in order to validate the theoretical formalization and modeling.

For the real experiments, we have generated training data according to the shape specific strategy: the object shapes are assumed known and side and top grasps are applied on them. The objects are placed such that they are initially not well centered with respect to the hand to investigate the capability of the learning system to cope with potential uncertainties in the objects' pose. An example real grasp execution is shown in Figure 3.

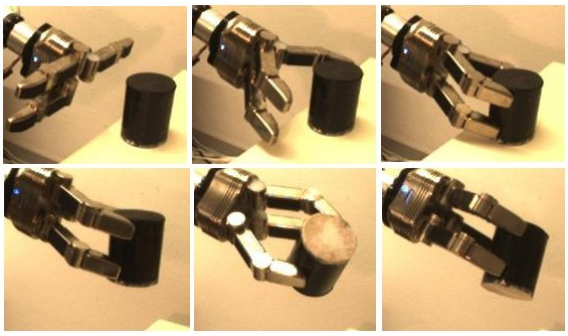


Fig. 3. A few examples from the execution of real experiments.

To generate the stable/unstable label for a grasping sequence, an object is lifted and rotated $[-120°, +120°]$ around the approach direction after a grasp has been applied to it. The grasps where the object is dropped or moved in the hand are labeled as unstable.

Training sequences $D_{R_1^2}$, $D_{R_2^2}$, $D_{R_3^2}$ are obtained by a parallel 2-fingered gripper with a deformable box and a deformable bottle shown in Figure 4. $D_{R_3^2}$ represents top grasps while the other two are side grasps. The rest of real data ($D_{R_1^3}$-$D_{R_6^3}$) are made on more rigid objects. $D_{R_1^3}$, $D_{R_2^3}$, $D_{R_3^3}$ are from the three fingered SDH and include contacts only on distal sensors: $D_{R_1^3}$ represents side grasps of a cylinder, $D_{R_2^3}$ side grasps of a bottle and $D_{R_3^3}$ top grasps of a bottle. $D_{R_4^3}$, $D_{R_5^3}$, $D_{R_6^3}$ are also side grasps for the same three-fingered hand but measurements from all six sensors are included.

Fig. 4. Objects from the real datasets denoted by ($D_{R_1^2}$), ($D_{R_2^2}$, $D_{R_3^2}$), ($D_{R_1^3}$, $D_{R_4^3}$), ($D_{R_5^3}$), ($D_{R_2^3}$, $D_{R_3^3}$), ($D_{R_6^3}$) in the text.

VI. Experimental results

Two HMMs, one for stable grasps and another for unstable ones were trained with the stopping criteria being the convergence threshold 10^{-4} with a 10 iteration limit. Both ergodic and left-to-right HMMs were evaluated independently with different structure parameters. The range of 2–6 for the number of states and 2–5 for the number of components in a mixture were evaluated. Diagonal covariance matrix structure was chosen. By evaluating multiple temporal models we aim at understanding whether the temporal sequence plays part in the understanding of the grasp stability, or if only the final observation is sufficient.

Experiments were performed both on simulated and real data similarly. For simulated data 80% of the samples were used for training and 20% for testing. For the real data 10-fold cross validation was used to evaluate the performance and the best parameters over all folds are presented. The number of stable and unstable samples are equal in each data set and the total number of samples are given in the Table I.

TABLE I
Number of Samples in Datasets

Data sets	Object	Grasp type	Number of samples
D_{S_1}	cylinder	side, 3-fingered	6400
D_{S_2}	bottle	side, 3-fingered	4906
D_{S_3}	bottle	top, 3-fingered	4446
D_{S_4}	cylinder	spherical, 3-fingered	6240
D_{S_5}	bottle	spherical, 3-fingered	2564
$D_{R_1^2}$	box	side, 2-fingered	148
$D_{R_2^2}$	bottle	side, 2-fingered	148
$D_{R_3^2}$	bottle	top, 2-fingered	100
$D_{R_1^3}$	cylinder	side, 3-fingered	140
$D_{R_2^3}$	bottle	side, 3-fingered	100
$D_{R_3^3}$	bottle	top, 3-fingered	50
$D_{R_4^3}$	cylinder	side, 3-fingered	60
$D_{R_5^3}$	cylinder	side, 3-fingered	60
$D_{R_6^3}$	bottle	side, 3-fingered	120

Table II presents the classification rates on simulated data for the ergodic and left-to-right HMMs with the corresponding best parameter values. Ergodic and left-to-right HMMs have comparable results.

To illustrate the difference on performance for different objects, the distributions of logarithms of likelihood ratios are presented for two objects for the same type, ergodic HMM, in Figures 6 and 8. Let L_s be the log likelihood of the stable HMM model and L_u be the log likelihood of the unstable HMM model, then $r = L_s - L_u$ shows the log of the likelihood ratio. Figures 6 and 8 show the histograms of these ratios (r) for stable and unstable samples. Blue bars show the difference for stable samples and red bars are for unstable samples. Figure 6 shows the distributions for the cylinder side grasps, for which the performance was relatively good, while in Figure 8 the distributions are given for the bottle grasps with spherical approach directions, for which the stability was more difficult to recognize. It is

TABLE II

RESULTS ON SIMULATED DATA

	D_{S_1}	D_{S_2}	D_{S_3}	D_{S_4}	D_{S_5}
$Rates_{ERG}$	**0.75**	**0.60**	**0.61**	**0.63**	**0.61**
$StableStates_{ERG}$	5	6	5	6	3
$StableComponents_{ERG}$	4	4	4	4	3
$UnstableStates_{ERG}$	4	5	6	5	2
$UnstableComponents_{ERG}$	4	3	3	5	4
$Rates_{LR}$	**0.75**	**0.60**	**0.61**	**0.65**	**0.62**
$StableStates_{LR}$	6	2	5	6	5
$StableComponents_{LR}$	4	5	4	2	2
$UnstableStates_{LR}$	4	4	5	3	4
$UnstableComponents_{LR}$	5	2	4	3	4
$GMMclassifierRates$	**0.76**	**0.59**	**0.59**	**0.57**	**0.60**
$GMMclusters$	3	4	3	4	3

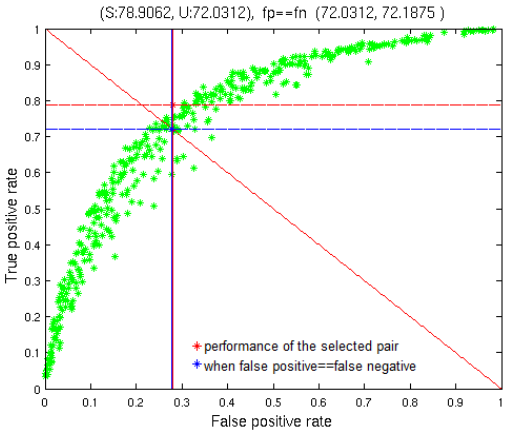


Fig. 5. The ROC for Cylinder side grasps.

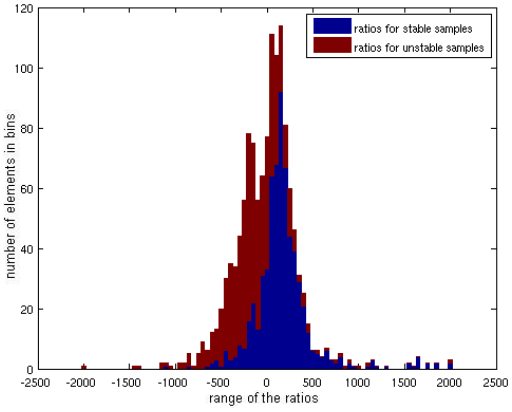


Fig. 6. The distribution of log-likelihood ratios for Cylinder side grasps.

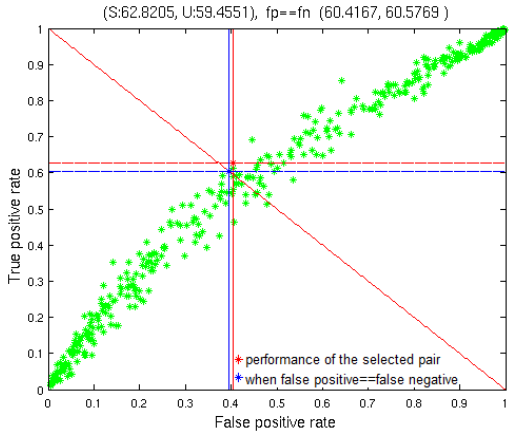


Fig. 7. The ROC for Bottle spherical grasps.

evident in the figures that the stable and unstable grasps differ reasonably.

Figures 5 and 7 with receiver operating characteristic (ROC) curves show how the HMM model parameters are chosen after training with different parameters. Each point in the figures indicates the performance of a trained HMM pair and the red cross indicates the performance of the selected HMM pair. Different HMM models were trained with different number of mixture components and states and finally the best HMM pair was chosen based on the maximum classification rates for stable and unstable grasps. The blue lines cross where the classification performance gives equal number of false positives/negatives and the chosen HMM models give a performance around this point which is the best possible one among the trained models.

From Table III and Table IV, it is evident that the classification rates are reasonable for 2-fingered and 3-fingered grasps with real robots. Table V shows the performance of the HMM system for predicting the stability of the final grasp using the first half of sequences of the sensor readings. The HMMs were trained and tested with the first half of the training sequences.

As shown, the HMM results for the simulated data is similar to the one-shot approach. For the real data, one-shot and HMM results differ in Table V, which may indicate that the process from the beginning to the end of the sequence has additional information that makes the HMM classification rate higher. We note that the real data includes readings from six tactile sensors while the simulated data includes the readings from only three. Therefore, the contacts on the proximal sensors for the real experiments may hold additional information to reason about the stability which needs to be analyzed with more data.

Given the results, it is evident that the idea of using the tactile feedback to evaluate the stability of a grasp is applicable also in a real world scenario.

VII. CONCLUSIONS AND FUTURE WORK

We have proposed the use of tactile sensing for estimating grasp stability using learning from training data. The experimental results show that tactile measurements allow relatively good recognition of grasp stability, and that the ideas studied in simulation are also applicable in real robot systems. The aim of the paper was not a perfect discrimination between successful and unsuccessful grasps but rather a measure of certainty of grasp stability. This also means that the system may reject some stable grasps while

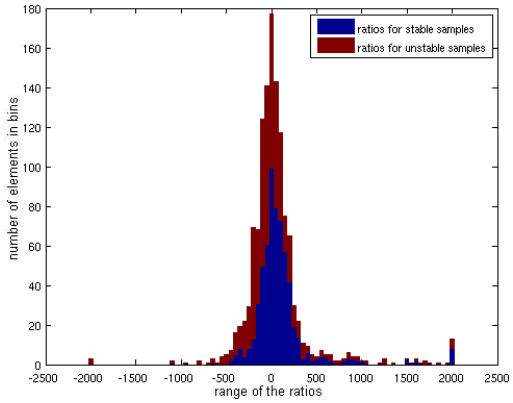


Fig. 8. The distribution of log-likelihood ratios for Bottle spherical grasps.

TABLE III

RESULTS ON REAL DATA WITH A 2 FINGERED GRIPPER

	$D_{R_1^2}$	$D_{R_2^2}$	$D_{R_3^2}$
$Rates_{ERG}$	**0.84**	**0.71**	**0.81**
$S.States_{ERG}$	2	4	6
$S.Components_{ERG}$	3	2	5
$U.States_{ERG}$	2	4	5
$U.Components_{ERG}$	3	2	5
$Rates_{LR}$	**0.85**	**0.70**	**0.73**
$S.States_{LR}$	4	2	4
$S.Components_{LR}$	4	4	3
$U.States_{LR}$	2	3	6
$U.Components_{LR}$	5	5	5

having fewer unstable grasps classified as stable ones. We showed how a one-shot classifier and an HMM classifier perform with different datasets. Experiments showed that using time-series data to evaluate grasp stability appears to be beneficial during dynamic grasp execution.

Future work will be to first perform a more extensive evaluation of the method on more objects with more samples and also include all the sensors in simulation. We also plan to investigate the proposed idea on completely unknown objects by using data that includes multiple objects and then extend the methodology to evaluate part-based grasps.

REFERENCES

[1] J. Tegin, J. Wikander, S. Ekvall, D. Kragic, and B. Iliev, "Demonstration based learning and control for automatic grasping," in *International Conference on Advanced Robotics*, 2007.

[2] D. Kragic and H. I. Christensen, "Cue integration for visual servoing," *IEEE Trans. on Robotics and Automation*, vol. 17(1), pp. 18–27, 2001.

[3] V. Kyrki, D. Kragic, and H. I. Christensen, "New shortest-path approaches to visual servoing," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2004, pp. 349–354.

[4] "Tactile database," http://www.nada.kth.se/∼yaseminb/.

[5] B. Siciliano and O. Khatib, Eds., *Springer Handbook of Robotics*. Springer, 2008.

[6] K. Huebner, K. Welke, M. Przybylski, N. Vahrenkamp, T. Asfour, D. Kragic, and R. Dillmann, "Grasping Known Objects with Humanoid Robots: A Box-based Approach," in *International Conference on Advanced Robotics*, 2009.

[7] C. Goldfeder, M. Ciocarlie, and H. D. P. K. Allen, "The Columbia Grasp Database," in *IEEE International Conference on Robotics and Automation*, 2009, pp. 3343–3349.

TABLE IV

RESULTS ON REAL DATA WITH THE SDH HAND

	$D_{R_1^3}$	$D_{R_2^3}$	$D_{R_3^3}$	$D_{R_4^3}$	$D_{R_5^3}$	$D_{R_6^3}$
$Rates_{ERG}$	**0.98**	**0.99**	**0.97**	**0.90**	**0.97**	**0.90**
$S.States_{ERG}$	4	4	4	3	4	6
$S.Comp._{ERG}$	4	5	2	5	2	4
$U.States_{ERG}$	2	2	2	2	3	4
$U.Comp._{ERG}$	4	3	5	3	2	5
$Rates_{LR}$	**0.99**	**0.98**	**0.96**	**0.93**	**0.98**	**0.93**
$S.States_{LR}$	5	2	6	3	6	3
$S.Comp._{LR}$	5	2	2	5	4	2
$U.States_{LR}$	2	2	2	3	5	6
$U.Comp._{LR}$	2	4	2	2	2	4

TABLE V

RESULTS USING SUBSEQUENCES TO PREDICT THE STABILITY OF THE FINAL GRASP

	D_{S_1}	D_{S_4}	D_{S_5}	$D_{R_5^3}$	$D_{R_6^3}$
$Rates_{LR}$	**0.68**	**0.55**	**0.54**	**0.90**	**0.88**
$S.States_{LR}$	6	3	4	5	4
$S.Components_{LR}$	5	5	3	5	4
$U.States_{LR}$	6	4	4	5	3
$U.Components_{LR}$	5	4	2	2	4
$GMMrates$	**0.68**	**0.57**	**0.55**	**0.79**	**0.78**

[8] A. Morales, M. Prats, P. Sanz, and A. P. Pobil, "An experiment in the use of manipulation primitives and tactile perception for reactive grasping," in *Robotics: Science and Systems (RSS 2007) Workshop on Robot Manipulation: Sensing and Adapting to the Real World*, Atlanta, USA, July 2007.

[9] M. Prats, P. Sanz, and A. del Pobil, "Vision-tactile-force integration and robot physical interaction," in *IEEE International Conference on Robotics and Automation*, Kobe, Japan, 2009, pp. 3975—3980.

[10] D. Kragic, A. Miller, and P. Allen, "Real-time tracking meets online grasp planning," *IEEE International Conference on Robotics and Automation, ICRA'01*, pp. 2460–2465, 2001.

[11] C. Goldfeder, P. K. Allen, C. Lackner, and R. Pelossof, "Grasp Planning Via Decomposition Trees," in *IEEE International Conference on Robotics and Automation*, 2007, pp. 4679–4684.

[12] S. Ekvall and D. Kragic, "Learning and Evaluation of the Approach Vector for Automatic Grasp Generation and Planning," in *IEEE Int. Conf. on Robotics and Automation*, 2007, pp. 4715–4720.

[13] J. Romero, H. Kjellstrom, and D. Kragic, "Markerless human-to-robot grasp mapping based on a single view," in *International Conference on Advanced Robotics*, 2009.

[14] A. Petrovskaya, O. Khatib, S. Thrun, and A. Y. Ng, "Bayesian estimation for autonomous object manipulation based on tactile sensors," in *ICRA*, 2006, pp. 707–714.

[15] M. Schöpfer, M. Pardowitz, and H. J. Ritter, "Using entropy for dimension reduction of tactile data," in *14th International Conference on Advanced Robotics*, 2009.

[16] A. Schneider, J. Sturm, C. Stachniss, M. Reisert, H. Burkhardt, and W. Burgard, "Object identification with tactile sensors using bag-of-features," in *In Proc. of the International Conference on Intelligent Robot Systems (IROS)*, 2009.

[17] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," in *Proceedings of the IEEE*, 1989, pp. 257–286.

[18] C. Ferrari and J. Canny, "Planning optimal grasps," in *IEEE Int. Conf. on Robotics and Automation*, 1992, pp. 2290–2295.

Visual tracking of a jaw gripper based on articulated 3D models for grasping

José J. Sorribes, Mario Prats and Antonio Morales

Abstract— **Robust grasping of objects in uncertainty conditions can be achieved with the visual monitoring of the interaction between the robot hand and the object.**

In this paper we propose a new approach for the visual tracking of a robot hand suitable to observe the interaction between robot and object. It consists on the continuous vision-based recovery of the articular pose of the robot hand. It is based on the principles of virtual visual servoing, which allows to deal with articulated bodies and occlusions.

Its suitability is shown by tracking a parallel jaw gripper under different conditions such as self-occlusion, articulated motion and important changes in the point-of-view. The potential applications range from estimating the robot hand articulated pose under poor hand-eye calibration and joint feedback, until detecting deficient contact configurations, incipient slips, etc.

I. INTRODUCTION

Vision has a key role in robot grasping and manipulation tasks. The process of grasping an object in low-structured scenarios by a robot hand can be roughly divided in three main stages: planning, approaching and execution. In the planning stage, the robot decides how to grasp and approach an object. This approaching is done in the next phase while avoiding obstacles . Finally, in the execution stage the fingers are closed over the object, according to the planned grasp, and first contacts are made. Execution phase can also include corrections with more contacts and releases until a satisfactory grip on the object is reached.

Vision is used at different levels on this process. Firstly, it has been used to locate and identify target objects on the scene. The main approaches differ on whether there exists previous information about the shape or appearance of the object [1], [2], or not [3], [4].

Second, visual input from the objects has been used to plan potential grasp on the target objects and approaching paths to them. Some of these planners assume to have the pose and shape of the object [5], [6]. Other approaches do not rely on the whole shape of the object but use vision to identify and extract specific features that allow the planning of a grasp [7], [8].

And third, visual feedback is also used when the arm tries to reach the object. For an instance, Murphy et al. uses visual techniques to correct the orientation of four-finger hand while approaching an object to allow better contact locations [9], and Namiki et al. uses a fast control schema in combination with tactile feedback to cage an object [10].

Authors are with the Computer Science and Engineering Department, Jaume-I University, 12071 Castellón, Spain. {jsorribe,mprats,morales}@uji.es

Little attention has been put on the use of vision when the robot hand makes the first contacts on the objects. Most closed-loop hand controllers simply rely on contact-based sensors to obtain feedback at this stage [11], [12].

However, there are a number of reasons that motivate the use of vision in the control-loop of grasp execution. First, some arms and hands do not have suitable sensor feedback for providing accurate position information. In other cases, hand-eye calibration is poor and does not allow for open-loop accurate hand-object alignment. In these cases, vision could potentially provide the pose of the objects and the the pose and configuration of the hand. Finally, when contacts occur vision information could be combined with contact sensor modalities to provide richer information about the grasp, such as the contact configuration, object sliding, etc. Such information is of great interest for exploration and learning systems.

In this paper we propose a new approach for the visual tracking of a robot hand based on the continuous vision-based recovery of the articular pose of the hand by means of Virtual Visual Servoing [13]. The main applications of this method in the context of robotic grasping are (i) the capability of tracking the hand/object interaction without the need for special markers [14], (ii) the direct estimation of the hand pose in sensor-less hands, and (iii) the possibility to detect contact points from vision. The two main difficulties that such a tracking solution must overcome are two. In the first place, it must deal with occlusions, both self-occlusions and those produced by the object. In the second place, it must deal with articulated bodies.

Its suitability is shown by tracking a parallel jaw gripper under different conditions such as self-occlusion, camera motion, articulated motion and important changes in the point-of-view.

A. Pose estimation work

The proposed approach is based on the pose tracking of the robot hand. This technique has been mostly used in robot manipulation to track target objects [1], [15], [4], but not robot parts.

Pose estimation techniques can be classified in appearance-based or model-based approaches [16]. Appearance-based methods work by comparing the 2D image of the object with those stored in a database containing previously acquired views from multiple angles. The main advantage of these methods is that they do not need a 3D object model, although a previous process must be performed in order to include a new object in

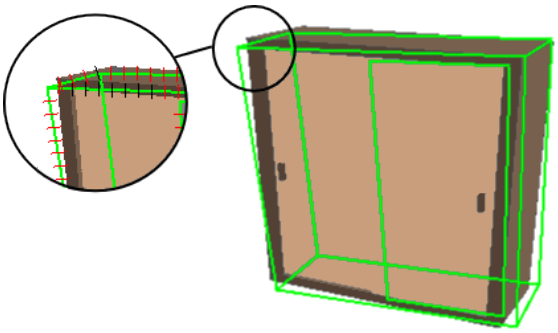


Fig. 1. A feature vector is built from the distances between the projected edges and high-gradient points searched along the edge normals, at the sampling interval. The goal of the non-linear minimization is to reduce all the distances to zero.

the database. Model-based methods obtain better accuracy and robustness, because of the use of model information for anticipating events like object self-occlusions. Some approaches consider a combination of both methods, like [1], where an appearance-based method is used first for getting an initial pose estimation, which is then used as initialization for a model-based algorithm.

Although vision has been widely adopted for detecting and tracking the objects to be manipulated, very few approaches have considered the use of vision for tracking the robot hand.

II. Articulated Virtual Visual Servoing

There are two main methods in the literature for model-based pose estimation and tracking of articulated objects, both based on full-scale non-linear optimization. The first, developed by Drummond and Cipolla [17], is formulated from the Lie algebra point of view, whereas the second, proposed by Comport et. al. [18], [19], is based on the Virtual Visual Servoing (VVS) method [13]. Both methods implement robust estimation techniques and have shown to be very suitable for real-time tracking of common articulated objects in real environments. A comparison between both approaches is reported in [20], where it is shown that both formulations are equivalent, although some differences in performance can appear at run time. In our system, the VVS approach has been implemented [19], [13], mainly for its computational efficiency and because it is based on a solid background theory, i.e. 2D visual servoing, which convergence conditions, stability, robustness, etc. have been widely studied in the visual servoing community [21]. In addition, almost any kind of visual feature can be used and combined with this approach (points, lines, ellipses, etc.), as long as the corresponding interaction matrix can be computed. Different examples of the interaction matrix for the most common features are shown in [22].

A. The concept

The concept of the VVS approach, developed in [13], is to apply visual servoing techniques to a virtual camera, so that a set of object features projected in the virtual image from a model, match with those extracted from the real image. Under this approach, the pose estimation and tracking problem can be seen as equivalent to the problem of 2D visual servoing [18], which has been extensively studied in the visual servoing community [21]. Taking as input an object model, and an initial estimation of the camera pose in object coordinates, denoted as a pose vector, $\mathbf{r}$, the idea is to project a set of 3D features of the object model into a virtual image of the object, taken from the virtual camera position, $\mathbf{r}$. This virtual image is compared with the real one, and a vector of visual features is generated, denoted by $\mathbf{s}(\mathbf{r})$.

In our particular implementation, we make use of the point-to-line distance feature, as in [18], although any kind of geometric feature could be used as long as the interaction matrix can be computed. The edges of the object model, projected as lines in the virtual image, are sampled at regular intervals, and a search for a strong gradient is performed in the real image, in a direction perpendicular to the projected line, as shown in Figure 1. For each match, the point-to-line distance is computed and stored in the feature vector. The desired feature vector is given by $\mathbf{s}^* = 0$, which represents the case when all the edges of the object model are projected on strong gradients, and, ideally, the virtual camera position corresponds to the real one. The control law governing the virtual camera motion is given by:

$$\mathbf{v}_r = -\lambda \left(\widehat{\mathbf{D}\mathbf{L}_s}\right)^+ \widehat{\mathbf{D}}(\mathbf{s}(\mathbf{r}) - \mathbf{s}^*) \tag{1}$$

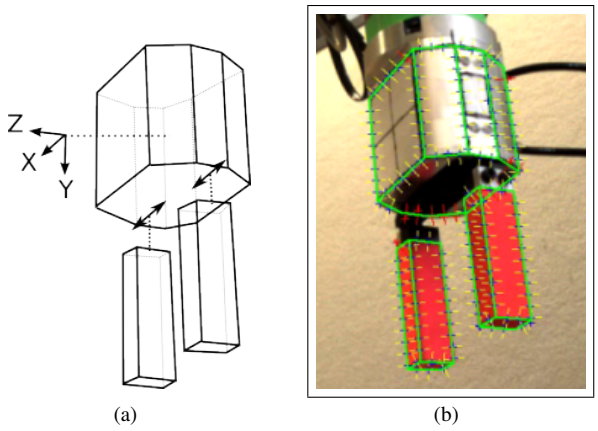
where $\mathbf{v}_r$ is the virtual camera velocity, λ is a control gain, $\widehat{\mathbf{L}_s}$ is the interaction matrix for the point-to-line distance feature, and $\widehat{\mathbf{D}}$ is a diagonal weighting matrix computed by iteratively re-weighted least squares, which is a robust estimator for dealing with outliers [18].

B. Virtual Visual Servoing on articulated objects

Comport et al. presented in [19] an approach for pose estimation and tracking of articulated objects based on the VVS method and the kinematic set concept. In their approach, the articulated pose is estimated directly from the visual observation of the object parts, leading to an efficient method that eliminates the propagation of errors through the kinematic chain. The only condition is that joint parameters must be decoupled in the minimization of the objective function. This can be accomplished by performing the minimization in object joint coordinates instead of in the camera space. Let $\mathbf{s}_1(\mathbf{r}_1)$ and $\mathbf{s}_2(\mathbf{r}_2)$ represent the perceived visual features on both parts of an articulated object composed of two links and one joint, and $\mathbf{s}_1^*$ and $\mathbf{s}_2^*$ be the desired values for those features, with $\widehat{\mathbf{L}_{s1}}$ and $\widehat{\mathbf{L}_{s2}}$ representing the corresponding interaction matrices. Then, the articular pose can be estimated by applying the following image-based control law:

$$\begin{pmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \end{pmatrix} = -\lambda \widehat{\mathbf{A}} \left(\widehat{\mathbf{D}}\widehat{\mathbf{H}}\right)^+ \widehat{\mathbf{D}} \begin{pmatrix} \mathbf{s}_1(\mathbf{r}_1) - \mathbf{s}_1^* \\ \mathbf{s}_2(\mathbf{r}_2) - \mathbf{s}_2^* \end{pmatrix} \tag{2}$$

$$\widehat{\mathbf{H}} = \begin{pmatrix} \widehat{\mathbf{L}_{s1}} & \mathbf{0} \\ \mathbf{0} & \widehat{\mathbf{L}_{s2}} \end{pmatrix} \widehat{\mathbf{A}}$$



(a) (b)

Fig. 2. The 3D model of the hand and its registration into a real view.

$$\widehat{\mathbf{A}} = \begin{pmatrix} \widehat{{}^{C}\mathbf{W}_{O}}\mathbf{S} & \widehat{{}^{C}\mathbf{W}_{O}}\mathbf{S}^{\perp} & \mathbf{0} \\ \widehat{{}^{C}\mathbf{W}_{O}}\mathbf{S} & \mathbf{0} & \widehat{{}^{C}\mathbf{W}_{O}}\mathbf{S}^{\perp} \end{pmatrix}$$

where $\widehat{{}^{C}\mathbf{W}_{O}}$ represents the twist transformation matrix from the camera frame to the object joint frame, and $\mathbf{S}^{\perp}$ is a constraint matrix which depends on the type of joint [19]. Finally, the virtual camera velocities, one for each link, are given by $\mathbf{v}_1$ and $\mathbf{v}_2$.

III. Model-based tracking of a parallel jaw gripper

A. Jaw gripper model

For this particular work, we consider a parallel jaw gripper as the one shown in Figure 2b. It consists of a box-shaped base containing the electronics, and two jaws actuated by a single motor. The hand can receive commands for opening, closing and stopping, and returns feedback only if the jaws are completely opened or closed. However, it does not contain sensors that provide the exact grip aperture, which is one of the reasons that motivate the use of visual information.

We define a 3D model of the gripper composed of the most distinguishable 3D edges, as shown in Figure 2a. The model is composed of three different parts: the base and the pair of jaws. The base and the pincers are kinematically linked through a prismatic joint along the base X axis. This joint is modeled with the holonomic constraint matrix $\mathbf{S}^{\perp} = (1,0,0,0,0,0)^{T}$, and, as the motion of the pair of jaws is coupled and controlled by a single motor, the articulation matrix takes the form of:

$$\widehat{\mathbf{A}} = \begin{pmatrix} \widehat{{}^{C}\mathbf{W}_{O}}\mathbf{S} & \widehat{{}^{C}\mathbf{W}_{O}}\mathbf{S}^{\perp} & \mathbf{0} \\ \widehat{{}^{C}\mathbf{W}_{O}}\mathbf{S} & \widehat{{}^{C}\mathbf{W}_{O}}\mathbf{S}^{\perp} & \widehat{{}^{C}\mathbf{W}_{O}}\mathbf{S}^{\perp} \\ \widehat{{}^{C}\mathbf{W}_{O}}\mathbf{S} & \widehat{{}^{C}\mathbf{W}_{O}}\mathbf{S}^{\perp} & -\widehat{{}^{C}\mathbf{W}_{O}}\mathbf{S}^{\perp} \end{pmatrix}$$

B. Tracking

The hand is tracked by iteratively applying equation 3 adapted to the case of three components and using the previous articulation matrix, i.e.:

$$\begin{pmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \\ \mathbf{v}_3 \end{pmatrix} = -\lambda \widehat{\mathbf{A}} \left(\widehat{\mathbf{D}}\widehat{\mathbf{H}} \right)^{+} \widehat{\mathbf{D}} \begin{pmatrix} \mathbf{s}_1(\mathbf{r}_1) - \mathbf{s}_1^{*} \\ \mathbf{s}_2(\mathbf{r}_2) - \mathbf{s}_2^{*} \\ \mathbf{s}_3(\mathbf{r}_3) - \mathbf{s}_3^{*} \end{pmatrix} \quad (3)$$

$$\widehat{\mathbf{H}} = \begin{pmatrix} \widehat{\mathbf{L}_{s1}} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \widehat{\mathbf{L}_{s2}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \widehat{\mathbf{L}_{s3}} \end{pmatrix} \widehat{\mathbf{A}}$$

The distance feature vector is computed by sampling points in the projected edges at regular intervals and looking for strong gradients in the perpendicular direction. An oriented gradient search is thus performed, as in [18], by applying a gradient convolution mask along a linear path perpendicular to the specific edge on each sampled point. The length of the search line depends on the expected object variation between two consecutive frames. For slow object/camera motion and/or high frame rates, little image variation between two consecutive frames is expected. Therefore, the search line can be of a few pixels around the sampled point, thus increasing the tracker efficiency. On the contrary, for low frame rate and/or high camera/object velocity, the search space should be increased.

Points are sampled only on those edges that belong to visible faces. Face visibility is computed at each iteration by checking the position of the camera with respect to the planes defined by each face normal. Being A, B, C and D the parameters of the plane equation corresponding to a specific face, with the face normal pointing towards outside, the condition for face visibility can be computed as:

$$A \cdot r_x + B \cdot r_y + C \cdot r_z + D > 0$$

where r_x, r_y and r_z are the translational components of the pose vector $\mathbf{r}$ that contains the camera pose with respect to the object.

This approach allows to check face visibility in a very efficient way. However, object self-occlusions cannot be detected. This is not a major problem in our experiments, since outlier rejection via the weighting matrix $\mathbf{D}$ is able to deal with these situations, as long as the occluded edges are only a small part of the object. As future improvements, we would like to deal with self-occlusions via binary space partition trees, like in [18].

C. Camera motion

In grasping situations where the robot hand has to reach for an object, it is important to adopt an active vision approach in which the camera follows the hand motion. This allows to obtain a detailed view of the robot hand at the same time that it is always kept inside the image.

For this purpose, we attach the camera to a pan/tilt unit that allows to point the viewing direction towards any interesting point. The pan/tilt unit is a PTU-46-70 model from Directed Perception Inc., and communicates with the host computer via a RS-232 port. The camera is a standard Firewire camera providing 30 frames per second at the resolution of 640×480. The complete system can be seen in Figure 3.

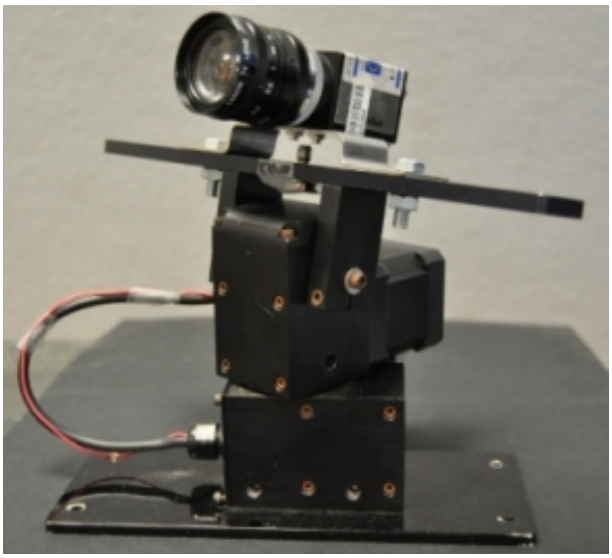


Fig. 3. A pan/tilt unit used for keeping the hand inside the camera field of view.

In order to keep the robotic hand inside the camera field of view, an image-based controller has been implemented. At each iteration the 3D center of the hand model is projected into the image according to the estimated hand pose. Its 2D distance to the center of the image is computed and fed back to a simple proportional controller that sends a pan/tilt velocity that moves the viewing direction towards the hand center.

IV. Results

The hand tracker has been validated with three different experiments that reproduce real situations that will commonly occur during manipulation:

1) Hand rotation that involves self-occlusion and appearance/disappearance of faces.
2) Articulated motion of the pincers.
3) Simultaneous motion of the hand and the robot camera.

In the first of the experiments, the robot end-effector was placed at a fixed position, and the pan/tilt unit in a configuration in which the robot hand was centered in the image. A rotation velocity was set around the hand axis, as shown in Figure 4a. This motion made some faces of the hand appear and disappear, and also generated self-occlusions, specially on the pincers. The hand tracker was able to deal with these situations by dynamically selecting the visible faces. Self-occlusions, not detected by the visibility check described in the previous section, generate wrong matches in the image. However, as long as the number of wrong matches is a small amount compared with those features correctly matched, the robust estimator implemented with the tracker is able to classify them as outliers and reject them.

In the second experiment, both the robot end-effector and the pan/tilt unit were kept at a fixed position. The tracker was initialized and the hand pincers were commanded to open and close repeatedly. The tracker was able to follow this articulated motion even in the presence of self-occlusions, as shown in the top row of Figure 4b.

It is worth mentioning that this capability is specially interesting for this particular robotic hand that does not provide joint feedback. Therefore, vision can be used here in order to provide a direct estimation of the opening distance, and eventually control it to a desired configuration. Another experiment dealt with the case in which the hand joints were manually actuated by a human, as shown in the bottom row of Figure 4b.

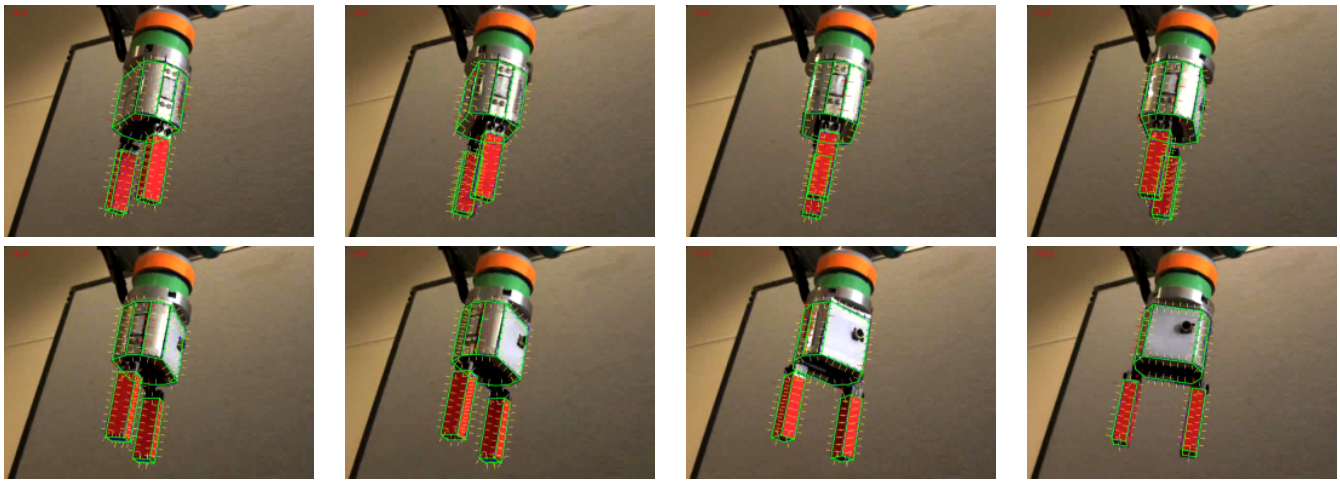
Finally, in the last experiment, a joint velocity was sent to the manipulator elbow, generating both translational and rotational motion of the robot hand. The pan/tilt controller was activated in order to keep the hand inside the camera view, even if part of the hand was outside the image limits, as shown in Figure 4c. The tracker also performed successfully in this situation where both robot and camera motion was performed simultaneously. Finally, it is worth mentioning that the tracker runs at video rate, as it can be observed in the video accompanying this paper.

V. Discussion and future lines

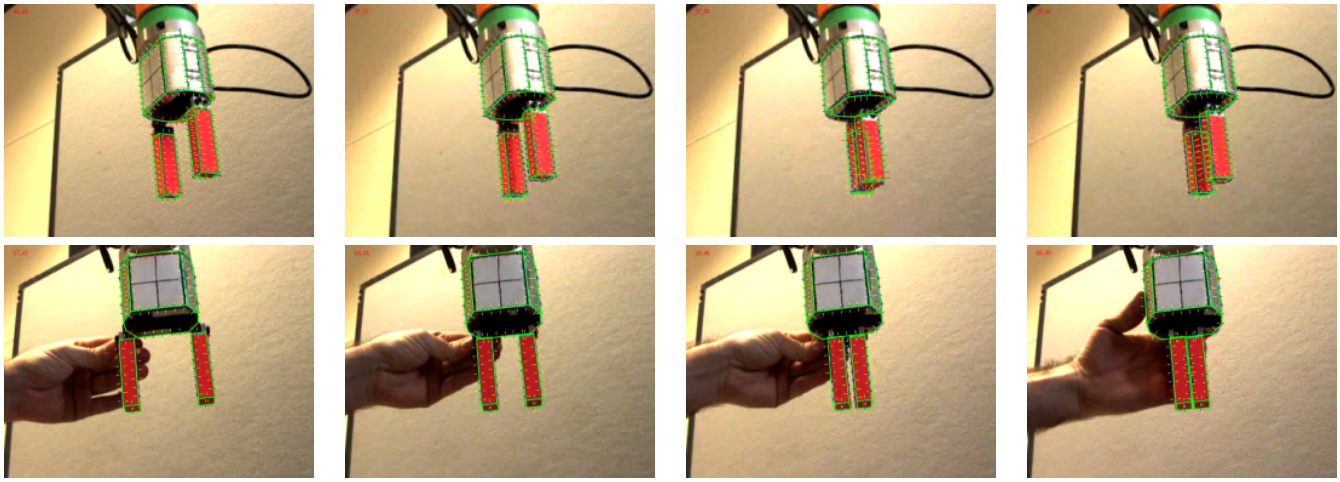
The approach proposed still presents several practical problems that need to be addressed before having a full-working version. The first one is the robustness of point matching. The current implementation requires small changes between two consecutive captured images. This can be accomplished either by a high capture rate or by limiting the hand/camera relative velocity. If the change of the object position between two sequential images is too large, the search distance has to be increased, and the tracker runs more slowly. In this case point tracking problems are frequently experienced. Also, if the movement of the camera or the robot hand is too fast, the tracker may lose the reference. There are several solutions to improve the tracking robustness. On one side is it possible to include forward prediction either using signals coming from arm and hand position controllers or simply using visual cues. A second option is to use a more robust and efficient point matching algorithm. In addition, the use of a kalman filter would also improve considerably this method.

The method has not been yet tested with complex hands. The parallel-jaw gripper that has been used has only one joint that actuates two different parts. Advanced robot hands usually have many more free joints. We have plans for applying this approach into a Barrett Hand, which also lacks position feedback on some joints. In general the method requires a set of strong visual features that can be efficiently tracked.

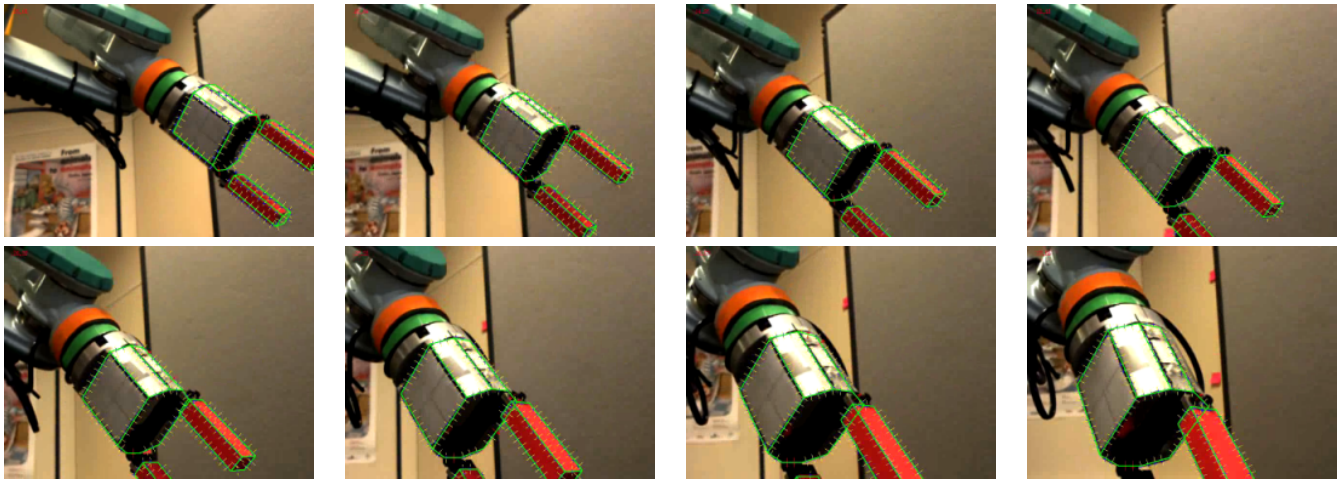
Finally, a crucial aspect of the tracking method is initialization. Currently, a human operator clicks on a camera image to project the initial model of the gripper. However an automatic method needs to be developed and implemented to solve this issue. One possibility is to make use of an initial estimation of the hand position, computed from a coarse hand-eye calibration, in case it is available. If not, an



(a) Occlusions and new faces appearing



(b) Opening and closing the actuator



(c) Hand tracking via head motion

Fig. 4. Results of the model-based hand tracker.

appearance-based pose estimation method can be adopted in order to provide a coarse initialization.

The approach proposed in this paper deals with a problem that has not been successfully addressed yet in the literature. The results presented in this paper are promising but still some improvements are needed before a robust robot hand tracker is obtained. Such a solution would allow a robust control of the hand while contacting an object, and in the last term allow robust grasping of objects under uncertainty.

VI. Conclusions

Visual tracking of a robot hand offers important potential applications. First is that it allows to estimate the hand configuration when the hand joints do not provide any feedback, or it is inacurate. In addition, it potentially avoids the use of external hand-eye calibration, thus being very suitable in situations where the robot camera position is not accurate, or the kinematics relating the camera and the hand systems is poor. It also allows to detect and correct any hand-object misalignment during and after the grasp execution. This is particularly interesting for the detection of deficient grasps or incipient slip. Finally, the tracking of the robot hand while contacting and object allows the fusion with contact sensor data (force/torque, tactile, pressure), and opens new possibilities for robot control, object exploration and robot learning.

This paper has described an approach for estimating and tracking the articular pose of a robotic hand. This approach is based on the method of virtual visual servoing, and allows to estimate the articular 3D pose of an object from a model and natural object features. It has been tested with a parallel jaw gripper, and under several conditions that are normally present in any manipulation task. Although the method can be improved in many different ways, it already represents a valid solution for simple hands like a parallel jaw gripper. In the future we would like to validate this approach with a more complex hand and during real grasping actions.

VII. acknowledgements

This research was partly supported by the European Commission's Seventh Framework Programme FP7/2007-2013 under grant agreements 215821 (GRASP project), 217077 (EYESHOTS project), and 248497(TRIDENT Project), by Ministerio de Ciencia e Innovación (DPI-2008-06636; and DPI2008-06548-C03-01), and by Fundació Caixa Castelló-Bancaixa (P1-1B2008-51; P1-1A2006-11; and P1-1B2009-50).

References

[1] D. Kragic and H.I. Christensen. Model based techniques for robotic servoing and grasping. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1:299–304, October 2002.

[2] P. Azad, T. Asfour, and R. Dillmann. Combining appearance-based and model-based methods for real-time object recognition and 6D-localization. In *International Conference on Intelligent Robots and Systems (IROS)*, Beijing, China, 2006.

[3] Beata J. Grzyb, Eris Chinellato, Antonio Morales, and Angel P. del Pobil. Robust grasping of 3D objects with stereo vision and tactile feedback. In *International Conference on Climbing and Walking Robots and the Support Technologies for Mobile Machines (CLAWAR)*, pages 851 – 858, Coimbra, Portugal, 2008.

[4] C. Dune, E. Marchand, C. Collewet, and C. Leroux. Active rough shape estimation of unknown objects. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3622–3627, Nice, France, September 2008.

[5] B. Wang, L. Jiang, J.W. Li, H.G. Cai, and H. Liu. Grasping unknown objects based on 3D model reconstruction. In *IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, Monterrey, California, July 2005.

[6] A. Morales, P.J. Sanz, A.P. del Pobil, and A.H. Fagg. Vision-based three-finger grasp synthesis constrained by hand geometry. *Robotics and Autonomous Systems*, 54(6):496–512, June 2006.

[7] Daniel Aarno, Johan Sommerfeld, Danica Kragic, Nicolas Pugeault, Sinan Kalkan, Florentin Wörgötter, Dirk Kraft, and Norbert Krüger. Early reactive grasping with second order 3D feature relations. In *IEEE Conference on Robotics and Automation*, Jeju Island, Korea, 2007.

[8] Ashutosh Saxena, Justin Driemeyer, and Andrew Y. Ng. Robotic grasping of novel objects using vision. *International Journal of Robotics Research*, 27(2):157–173, Feb 2008.

[9] T.G Murphy, D.M. Lyons, and A.J. Hendriks. Stable grasping with a multi-fingered robot hand: A behavior-based approach. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 2, pages 867–874, Yokohama, Japan, July 1993.

[10] A. Namiki, Y. Nakabo, I. Ishii, and M. Ishikawa. High speed grasping using visual and force feedback. In *IEEE International Conference on Robotics and Automation*, Detroit, MI, USA, May 1999.

[11] R. Platt Jr., A. H. Fagg, and R. Gruppen. Nullspace composition of control laws for grasping. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1717–1723, Lausanne, Switzerland, 2002.

[12] H. Kawasaki, T. Mouri, J. Takai, and S Ito. Grasping of unknown object imitating human grasping reflex. In *IFAC world congress*, Barcelona, July 2002.

[13] E. Marchand and F. Chaumette. Virtual visual servoing: a framework for real-time augmented reality. In *EUROGRAPHICS 2002*, volume 21(3), pages 289–298, Saarebrücken, Germany, 2002.

[14] M. Prats, P. Martinet, Angel P. del Pobil, and S. Lee. Robotic execution of everyday tasks by means of external vision/force control. *Intelligent Service Robotics*, 1(3):253–266, 2008.

[15] A. Stemmer, G. Schreiber, K. Arbter, and A. Albu-Schäffer. Robust assembly of complex shaped planar parts using vision and force. In *IEEE International Conference on Multisensor Fusion and Integration*, pages 493–500, Heidelberg, Germany, September 2006.

[16] V. Lepetit and P. Fua. Monocular model-based 3d tracking of rigid objects. *Foundations and Trends in Computer Graphics and Vision*, 1(1):1–89, 2005.

[17] T. Drummond and R. Cipolla. Real-time visual tracking of complex structures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):932–946, 2002.

[18] Andrew I. Comport, Éric Marchand, and François Chaumette. Robust model-based tracking for robot vision. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 692–697, 2004.

[19] Andrew I. Comport, Éric Marchand, and François Chaumette. Object-based visual 3d tracking of articulated objects via kinematic sets. In *Computer Vision and Pattern Recognition Workshop*, volume 1, page 2, Washington, DC, USA, 2004. IEEE Computer Society.

[20] Andrew I. Comport, Danica Kragic, Éric Marchand, and François Chaumette. Robust real-time visual tracking: Comparison, theoretical analysis and performance evaluation. In *IEEE International Conference on Robotics and Automation*, pages 2852–2857, Barcelona, Spain, 2005.

[21] S. Hutchinson, G. Hager, and P. Corke. A tutorial on visual servo control. *IEEE Transactions on Robotics and Automation*, 12(5):651–670, Oct 1996.

[22] B. Espiau, F. Chaumette, and P. Rives. A new approach to visual servoing in robotics. *IEEE Transactions on Robotics and Automation*, 8(3):313–326, Jun 1992.

UJI humanoid torso manipulation primitives

Javier Felip and Antonio Morales*

June 8, 2010

Abstract

This technical report shows the recent advances in manipulation primitives for the UJI humanoid torso: Tombatossals. So far there are four basic primitives implemented: transport, grasp, place, release and one specific: slide. This document is also an update to the previous work done in the robust grasping primitive. Moreover we explain the main ideas for the forthcoming primitives: open/close and push/pull objects.

1 Introduction

Controlling a robot with several degrees of freedom becomes a complex task. Some controllers are developed to deal with this complexity such as cartesian controllers, velocity controllers and so. This controllers help the user by abstracting the robot specific joint configuration and dealing with the inverse kinematics and dynamics but no external sensory data is used.

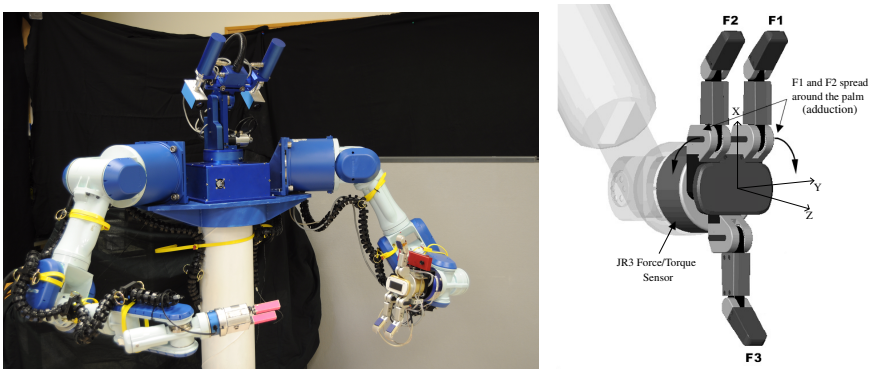
Even with this abstraction the control of robots is not easy. There are many parameters to take into account. But there is another level of abstraction that can be considered,which can help making the robot control much more easy, that is what we call manipulation primitives.

A grasp primitive is a specific controller designed to perform a particular indivisible action such as move, grasp, place, release, etc... This primitives have few parameters that made them able to adapt its behaviour to the desired one. Some parameters are required by the primitive to work but other parameters are just information that can be used to improve the performance of the primitive and the success ratio.

Actions can be described easily using a small set of manipulation primitives. For instance, a pick and place task can be described with five primitives. Move close to the object, grasp it, transport the object near the target position, place it and release. For this example the only required parameters are the position of the object, and the destination position where the object should be placed.

So far, we have already implemented 5 primitives: move/transport, grasp, place, release and slide. This primitives have a general use and with the correct parameters a

*J. Felip and A. Morales are with Robotic Intelligence Laboratory at the Department of Computer Science and Engineering, Universitat Jaume I, 12006 Castellón, Spain {jfelip,morales}@uji.es



(a) Tombatossals.

(b) Detail of the Barrett hand.

Figure 1: Tombatossals: The UJI humanoid torso.

wide scope of tasks can be done. Not only pick and place but pushing buttons, using handles, opening doors and so on. However for this tasks is useful to have more specific primitives. This makes the parameter and primitive selection easier. Moreover it allows to add special features to the primitive controller in order to deal with specific issues for that action as we have already done with the slide primitive.

1.1 System description

Our robotic setup consists of two Mitsubishi PA-10 with 7 d.o.f. (Degrees of freedom), and one pan-tilt-verge head TO40 from robosoft. The left arm is endowed with a three-fingered Barrett Hand and a JR3 sensor mounted on the wrist, between the hand and the end-effector (see Fig.1(a)). The hand has been improved by adding on the palm and fingertips arrays of pressure sensors designed and implemented by Weiss Robotics. The right arm has also a JR3 force-torque sensor and a parallel jaw gripper attached.

The Barrett hand is a 4 d.o.f., three-fingered hand. Each finger has one degree of freedom thus phalanxes are not independent. Fingers F1 and F2 can rotate around the palm and move next to Finger F3 (Thumb) or oppose to it, this d.o.f. is called adduction. The reference frame of the hand and the adduction d.o.f. are depicted in Fig. 1(b). Each finger of the hand has built-in strain-gauge sensor. The JR3 is a 12 d.o.f. sensor that measures force, torque and acceleration in each direction of the space.

1.2 Paper outline

In this paper we will show the details of the four implemented primitives and outline the main ideas for the implementation of some other specific manipulation primitives such as open/close doors, push, pull and so on.

2 Current primitives

2.1 Transport primitive

The aim of this manipulation primitive is to move the arm while it holds an object to the specified target position. It can also be used to move the arm without any object. The only required parameter is the target position.

Sometimes the trajectory to move the arm from the starting point to the target is a straight line without orientation changes, but it also happens that the trajectory generated has orientation changes and curves due to kinematic or planning constraints. Thus it is useful to constraint this primitive in order to prevent unexpected trajectories. This movement can be constrained by some parameters:

- Position limits: List of convex-hull volumes forbidden for the robot's end effector.
- Velocity limits: Cartesian end-effector velocity limits.
- Acceleration limits: Cartesian end-effector acceleration limits.
- Force-torque limits: Wrist force-torque limits.
- Follow torques: Force-torque compliancy mode.

For instance, if the robot grasps a mug full of water and wants to transport it without pouring the liquid, acceleration should be constrained to a low value on all axes and the rotation velocity of the table plane axes should be set to 0 to prevent tilting the mug. Also force-torque limits can be specified to detect collisions, if the force-torque exceeds the limits specified by the constraint, the movement stops immediately. If the target position cannot be reached because it cannot satisfy the specified constraints, the primitive informs about the constraint satisfaction problem. If the 'follow torques' parameter is set to true, the controller will modify its velocity depending on the sensed force and torque.

It is important to highlight that the controller only takes into account the end effector, it is possible to the other parts of the arm to move into the forbidden space.

2.2 Grasp primitive

The main features of the grasp primitive and its parameters are described in [felip09] since the publication of that work, the grasp primitive has been improved with another two correction methods:

- Translation error correction
- Sliding grasp

Moreover the discrete orientation corrections shown in [felip09] have been transformed to control laws that perform the same sensor based corrections with smooth movements.

The updated list of parameters for the robust grasp primitive are:

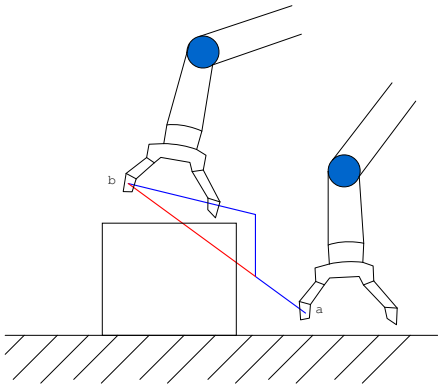


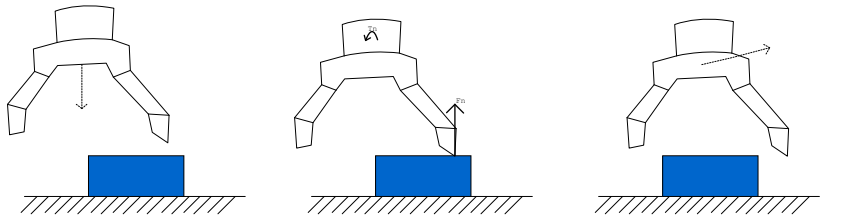
Figure 2: Example of execution of the constrained transport primitive from the starting point (a) to the target point (b). Red line: Standard trajectory. Blue line: Position constrained trajectory.

- Pregrasp size: Initial opening of the hand.
- Grasp preshape: cylindrical, spherical, hook.
- Translation alignment: Tells wether to use the translation correction or not.
- Rotation alignment: Tells wether to use the rotation correction or not.
- Parallel face detection: Tells wether to use the parallel face detection strategy or not.
- Slide grasp: Tells wether to use the slide grasp strategy or not.
- Object centering: Tells wether to use object centering.
- Caging grasp: Tells wether to use a caging grasp instead of a rigid one.

All the parameters are optional, only the starting position of the hand is required for the controller to attempt a grasp. Nevertheless the more parameters the better.

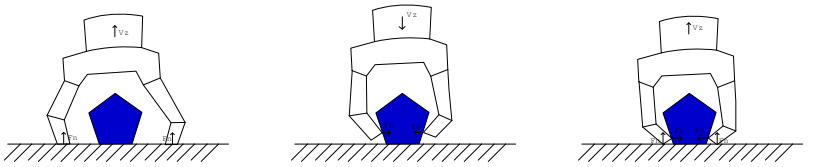
2.2.1 Translation correction

In some situations, the approach vector is not pointing to the center of the object, thus when the grasp primitive approaches to it, the hand collides prematurely with the object. This contact can be felt using the force-torque sensor as a torque force on the wrist. Using that torque, the contact point is determined and a correction is performed to center the object, a sample of this execution is depicted in Fig. 3.



(a) Arm moving towards the object. (b) Contact generates torque in the wrist. (c) Correction movement is performed.

Figure 3: Translation error correction strategy.



(a) The fingers contact the table while closing. Thus the controller sets the velocity to move the hand back. (b) The fingers are closing and the contact with the table is lost. Vz is set forwards. (c) The hand contacts the table again but the object is already grasped.

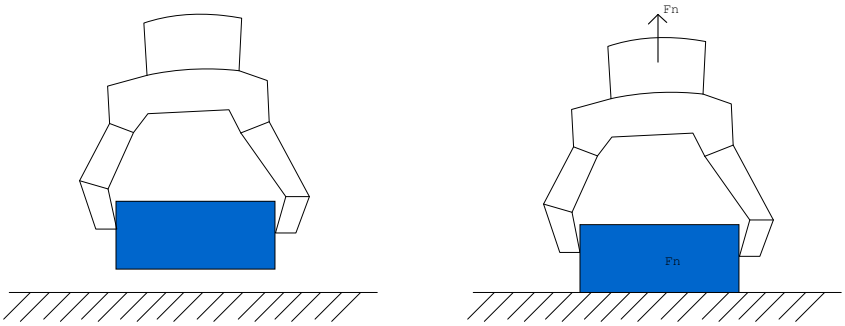
Figure 4: Sliding grasp strategy.

2.2.2 Sliding grasp

The sliding grasp is an alternative strategy for the parallel face detection, already presented in [felip09]. The main idea is simple, the fingers are always closing and the arm moves forward or backward depending on the force sensed along its Z axis (see Fig. 1(b)). When the fingers are no longer able to close, the grasp control ends.

An example of execution is shown in Fig. 4(a). The hand starts closing and when the fingers make contact with the surface, the force they are applying is felt in the wrist Fig. 4(a), thus the arm moves back. The fingers continue closing and cause there is no force felt, the arm moves forward Fig. 4(b). In Fig. 4(c) the fingers are not able to close, the primitive ends successfully.

One drawback of the parallel face detection is that some objects have not parallel faces to grasp but it is possible to apply good grasps on them. Another problem of the parallel surface detection was that it failed for small objects because they have not enough surface to apply two consecutive grasps. Moreover it had also problems when trying to grasp handles or concave objects. The slide grasp solves all of these problems because it tries to keep always hand-object or hand-surface contact.



(a) Arm moving the object towards the surface. (b) Contact is detected by force/torque sensor.

Figure 5: Place primitive.

2.3 Place primitive

The place primitive is the simpler one. Basically the arm moves down until a contact is detected. Then the execution of the primitive ends. To detect the contact, the primitive is monitoring the force sensor. When a force opposing the movement direction is felt it assumes that the object is placed.

This primitive is also configurable by the next two parameters:

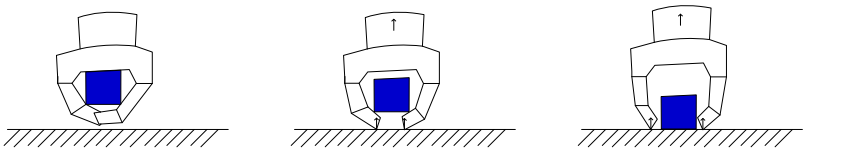
- Target position: Required parameter that determines the direction to place the object.
- Contact threshold: Optional parameter, determines the force needed to detect a contact. Default 8N.

2.4 Release primitive

Releasing an object is not as simple as opening the hand. Sometimes if the hand is grasping a handle or is enveloping an object close to the surface (see Fig. 6(a)) the hand cannot be opened because its fingers will collide with the surface. To handle this problem, the release primitive opens the hand slowly while the arm moves back. The movement of the arm is force-controlled and the arm only moves back if there is a contact detected between the opening fingers and the surface. The sequence of movements is shown in Fig. 6.

This primitive has also some optional parameters:

- Move away target: Optional parameter that determines the safe position where the arm should go after releasing.
- Release size: Optional parameter, determines the target hand opening to release. Default: Totally opened.



(a) Hand before opening the fingers.

(b) The hand cannot release the object, the fingers are blocked by the surface. The normal force in each finger propagates to the wrist.

(c) The hand moves back and continues opening the fingers. The object is released successfully.

Figure 6: Release primitive.

- Release handle: Optional parameter, determines wether to use the force-controlled release strategy shown in Fig. 6. Default: false.

2.5 Slide primitive

The aim of this primitive is to push the object from the top and slide it over a surface, see in Fig. 7. Using force control the arm applies the desired force (Fn) to the object, then starts moving towards the target, keeping the applied force constant, Fig. 7(a). This links the arm and object movement allowing to slide the object over the table from the starting to the target position, Fig. 7(b). Only the target position is a required parameter, but the force that the hand has to apply on the object is also configurable.

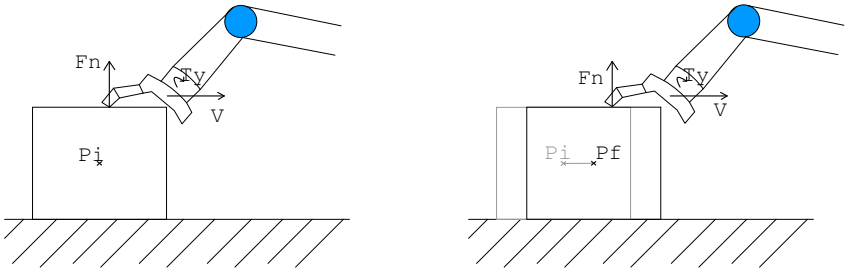
Slide primitive parameters:

- Target position: Target position in homogeneous matrix form (w.r.t World)
- Minimum force: Minimum force needed to slide the object. Default: 3.5N
- Maximum force: Maximum force allowed to slide the object. Default: 6.5N

At some point it may happen that hand-object contact is lost, then the movement towards the target stops and the hand moves down to find the object again. Unfortunately there is no way to distinguish between the object and the surface using force and tactile sensors, another sensorial input, like vision, is needed to supervise the execution of this primitive and decide about success or failure.

3 Future primitives

The current set of primitives are able to deal with the basic movements, such as moving the arm, transporting objects and grasping. It is possible to specify pick and place tasks with these primitives but there are more manipulation tasks that can be done by the robot (i.e. sliding objects, triggering buttons, opening doors, activating handles,



(a) From the starting position with a hook preshape, the arm moves down until it touches the object, then it starts moving towards the target.

(b) The object slides over the table from Pi to Pf. The primitive keeps the applied force stable.

Figure 7: Slide primitive.

etc...). Probably it is possible to use the current primitives to specify this tasks, but implementing specific primitives allows us to focus on the typical problems of each specific action, thus without increasing too much the amount of primitives we can deal more accurately with more manipulation tasks.

The future primitives we have been thinking about are:

- Push/pull:

  This is also a primitive to transport objects without lifting. The aim of this primitive is similar to the slide primitive: To slide the object pushing or pulling from a side using a non-prehensile grasp. For instance, this movement is useful to push objects beyond the arm limits. The tactile sensors provide data about the contacts, it should be enough information to detect the object while pushing (palm sensor) or pulling (fingertips sensors) and adapt the motion to the object movement.

- Open/close:

  Doors, dishwashers, drawers, boxes, etc. There are a lot of objects that can be opened or closed. Usually this task is to pull, push or slide a handle that is linked to the object. Thus its movement is constrained, the arm using tactile and torque data would be able to adapt to its constraints with few information.

These primitives are focused on tasks that change the object position. Triggering a button or a handle is also a manipulation task that could be managed by a manipulation primitive. Plugging-in task is another one that could have a primitive. After grasping the connector the plugging-in task can be modeled using a constrained transport primitive, but to deal with uncertainty in the detected plug position a sensor-based primitive would make the task execution more robust.

Acknowledgment

This technical report describes research carried out at the Robotic Intelligence Laboratory of Universitat Jaume I. The research leading to these results has received funding from the European Community's Seventh Framework Programme FP7/2007-2013 under grant agreement 215821 (GRASP project), and by Fundació Caixa-Castelló (P1-1A2006-11).

References

[felip09] Felip, J. and Morales, A. (2009). Robust sensor-based grasp primitive for a three-finger robot hand. In *IEEE/RSJ International. Conference on Intelligent Robots and Systems*.

Usability of Force-Based Controllers in Physical Human-Robot Interaction

Marta Lopez Infante
Lappeenranta University of Technology
P.O. Box 20, 53851 Lappeenranta, Finland
marta.lopez.infante@gmail.com

Ville Kyrki*
Lappeenranta University of Technology
P.O. Box 20, 53851 Lappeenranta, Finland
ville.kyrki@lut.fi

ABSTRACT

Learning from demonstration is an invaluable skill for a robot acting in a human populated natural environment, allowing the teaching of new skills without tedious and complex manual programming. Physical human-robot interaction, where the human is in a physical contact with the robot, is a promising approach for teaching especially manipulation skills. This paper studies the human side of physical human-robot interaction, in the context of a human physically guiding a robot through the desired set of motions. The paper addresses the question, which kind of response of the robot is preferable for the human user. In addition, different approaches for the guidance are described and relevant technical challenges are discussed. The main finding of the user study is that there is a need for a trade-off between the conflicting goals of naturalness of motion and positioning accuracy.

Categories and Subject Descriptors

I.2.9 [**Artificial Intelligence**]: Robotics

General Terms

Human Factors

Keywords

physical human-robot interaction

1. INTRODUCTION

Robots acting in open-ended, natural environments need a capability to learn and adapt to changes in their environment. Even though some completely autonomous learning approaches have been proposed, their inherent pitfall is often the lack of natural, task-driven feedback. Imitation learning, also known as Programming-by-Demonstration (PbD), seems a promising way of teaching new skills without the need of complex manual programming [2].

Programming by Demonstration (PbD) was born as an alternative to robot programming, solving the major problem of providing a a programming interface for inexperienced robot users. First PbD approaches were focused on task-learning from observation. However, non-intrusive observation, for example by vision, is often insufficient for learning manipulation tasks. To address this, recent efforts have been oriented to imitation of manipulation using more natural means. Physical human-robot interaction (HRI) is based on a physical contact between the human and the robot during their interaction[5].

Physical HRI allows goal-directed imitation, where actions have a specific purpose determined by the human. It allows programming a robot based on the human expertise and knowledge of the task, providing to the robot a demonstration of how to accomplish it. Current state of the art in describing the learned robot motions allows also some generalization, for example, changing the target of the motion[13]. However, little experimental knowledge is available on the human side of physical HRI.

This paper aims to address the question, which qualities are preferable for a human in a physical HRI system. More specifically, the scenario of a human physically guiding a robot arm equipped with a force sensor through a set of motions is studied. The study takes the form of a usability study where the main question is, which type of response of the robot is preferable for a novice human user. The question is not straightforward, as different types of force controllers have been proposed and can be implemented for the task. While the virtual tool approach was proposed already in 1993[11], we are not aware of any works taking the usability viewpoint, which we believe is critical. Moreover, few works if any discuss the implementation details of active control methods.

We begin by introducing in Sec. 2 relevant related work, discussing the theory and practice of physical HRI. Particular emphasis is given to work presenting systems for force-sensor based imitation learning. In Sec. 3 an overview technical description of a force sensor based control is presented. Relevant technical challenges are discussed, particularly the gravity compensation and singularity management, which seem to have got little attention in the literature. Section 4 describes the force control approaches used in the study, which are then demonstrated in Sec. 5. Section 6 describes the design of experiments as well as presenting and discussing the results of the experiments, with the main find-

*Corresponding author.

HRI'11, March 6–9, 2011, Lausanne, Switzerland.

ing that a solution to the problem is a trade-off between the conflicting goals of naturalness of motion and positioning accuracy. Finally, conclusions are drawn in Sec. 7.

2. RELATED WORK

Early work in physical HRI consists mainly of approaches based on the operation in human-populated environments. Some applications in this field are the collaborative work between human and robot for complete industrial tasks[7, 8] and high-force interaction with humans[9]. High-force interaction with human is based on a force controller that regulates the force that the robot should apply to the environment or object to increase the human force applied[9]. In other studies, collaborative work between human and robot for industrial tasks is based on regulating the motion of the robot[7, 8].

The physical HRI scenario considered in this paper is kinesthetic teaching, that is, manually guiding a robot through a desired set of motions. A common approach is to use backdrivable motors and joint encoders, for example, see [10, 1]. Due to inertia, this approach, however, only works well for relatively small robots, even when active gravity compensation is used, as the human needs to move the mechanical components manually. Moreover, the motions tend to become artificial as moving individual motors is easier than executing a coordinated movement [3].

An alternative to backdrivable robots is to use a force-sensor mounted at the end-effector and actively control the motion [4, 6], an approach also considered in this paper. The approach allows to use powerful industrial robots. Ferretti *et al.*[6] developed a system to teach tasks to a robot for applications that require position accuracy, such as spray painting. The authors present the exact dynamic equations of a virtual tool. The concept of virtual tool was earlier introduced in this field by [11]. Nevertheless, in the implementation of the admittance force controller by Ferretti *et al.*, the virtual tool dynamics are simplified, limiting the motion of the robot end-effector just to translations in the Cartesian coordinates. The model of the force controller is based on a gain for the input forces and a damping parameter related with the velocities.

Regarding the application of teaching motions to the robot, Frigola *et al.*[4] implemented a system based on this purpose to correct a robot programmed trajectory on-line by manual guidance.The force controller used in this system regulates the appropriate robot velocities modeled by the Stakes laws in a viscous medium, using a virtual tool. This system is based just on translation motions of the robot end-effector, similar to the work presented above. In contrast to [4, 6], the present paper proposes a force controller that regulates both the position and orientation of the robot end-effector.

To our knowledge, the usability aspect of physical human-robot interaction has not been addressed earlier to this paper. However, Steinfeld *et al.*[14] present an excellent general discussion on metrics for human-robot interaction. Although the paper does not discuss metrics for physical HRI, two types of metrics in the paper are especially relevant here: effectiveness (in the sense of accuracy) and accuracy of mental models (in the sense of naturalness of the motion and sense of being in control). However, there seems to be a further need to study general metrics for physical HRI in more detail. There have been relatively few studies on usability in the robotic field. The available studies are related to programming toolsets[12] or to specific robot applications such as urban research and rescue[16]. The results from those studies can not be directly compared with this study.

3. PHYSICAL HUMAN-ROBOT INTERACTION

This section describes the basic framework for following the physical demonstration of a human on a robot. The framework is based on the following scenario: A human operator grasps the robot end-effector, and the contact force is measured by a force/torque (F/T) sensor located at the wrist of the robot. An example of the interaction is shown in Fig. 1.

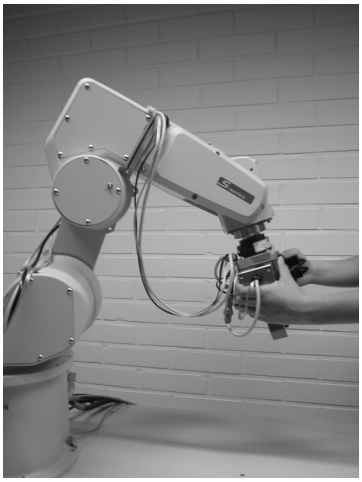


Figure 1: 6 DOF robot

The F/T measurement at the wrist is then converted to a velocity control signal for the velocity driven robot using a controller. The framework can be easily generalized to other control types, such as position controlled robots.

In a programming by demonstration application, the system can run in interaction or reproduction mode. The interaction mode corresponds to the teaching/learning phase of the robot, while the reproduction mode is based on the instantiation of the motion by the robot. In this paper, the focus is on interaction, and the reproduction of the motion will not be considered. The system structure on basic component level is illustrated in Fig. 2.

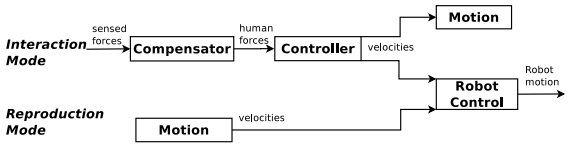


Figure 2: System structure diagram

One of the main technical challenges in the use of a force/torque sensor is that the measurements are not only influenced by the external contact forces, but also by the forces related to the robot hand attached to the sensor, including the gravity and inertial forces. The quantity of especially the gravity force is significant. Thus it needs to be compensated

in order to measure the forces applied by the human. Another challenge present in the system is the handling of the singularities of the robot workspace. These both are next discussed.

3.1 Gravity compensation

Figure 3 shows the location of the end-effector (tool attachment point), robot hand and the force sensor at the wrist. The gravity effect on the F/T sensor depends on the orientation of the sensor in the gravity field. Thus simple compensation schemes based on normalization, such as the one proposed in [6], are not applicable to general motions in six degrees of freedom. Next, we present the compensation approach applicable to the general case, as we are not aware of any other treatment. We begin by the compensation for forces, followed by compensation for torques.

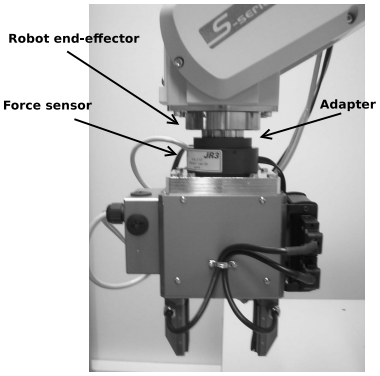


Figure 3: Robot hand structure

Assuming that the velocities in the system are small, allowing us to ignore centrifugal and Coriolis forces, the measured force in the sensor frame ${}^{s}\mathbf{F}_m$ can be written as a sum of the external human force $\mathbf{F}_h$ and gravity $\mathbf{g}$ as

$$ {}^{s}\mathbf{F}_m = {}^{s}\mathbf{F}_h + {}^{s}\mathbf{g}. \tag{1} $$

However, since the robot is controlled in the end-effector frame instead of the sensor frame, the human force used as an input needs to be presented in the end-effector frame. In addition, the gravity can be written as a constant in the world-frame. Thus, (1) can be rewritten using these as

$$ {}^{s}\mathbf{F}_m = {}^{s}\mathbf{R}_{ee}\,{}^{ee}\mathbf{F}_h + {}^{s}\mathbf{R}_w\,{}^{w}\mathbf{g} \tag{2} $$

from which the desired human force can be solved as

$$ {}^{ee}\mathbf{F}_h = {}^{ee}\mathbf{R}_s({}^{s}\mathbf{F}_m - {}^{s}\mathbf{R}_w\,{}^{w}\mathbf{g}). \tag{3} $$

In the above, ${}^{ee}\mathbf{R}_s$ is constant and ${}^{s}\mathbf{R}_w$ can be obtained using forward kinematics.

For the calculation of the contact torque, a similar equation can be computed, where the human torque in the end-effector frame ${}^{ee}\mathbf{T}_h$ can be obtained as a sum of the measured torque ${}^{ee}\mathbf{T}_m$ and the gravity torque of the hand ${}^{ee}\mathbf{T}_g$ as

$$ {}^{ee}\mathbf{T}_h = {}^{ee}\mathbf{T}_m - {}^{ee}\mathbf{T}_g. \tag{4} $$

The measured torque in the end-effector frame ${}^{ee}\mathbf{T}_m$ depends on both the measured force and torque in the sensor frame, and can be written

$$ {}^{ee}\mathbf{T}_m = {}^{ee}\mathbf{R}_s\,{}^{s}\mathbf{T}_m + {}^{ee}\mathbf{p}_s \times ({}^{ee}\mathbf{R}_s\,{}^{s}\mathbf{F}_m) \tag{5} $$

where ${}^{ee}\mathbf{p}_s$ is the position vector of the origin of the sensor frame in the end-effector frame.

The gravity torque of the hand ${}^{ee}\mathbf{T}_g$ can be obtained from the gravity force as

$$ {}^{ee}\mathbf{T}_g = {}^{ee}\mathbf{R}_s\,{}^{s}\mathbf{R}_w\,{}^{w}\mathbf{r} \times {}^{w}\mathbf{g} \tag{6} $$

where $\mathbf{r}$ is the location of the center of mass of the hand in the current pose of the robot in world coordinates.

Therefore, (4) can be rewritten using (5) and (6) as

$$ {}^{ee}\mathbf{T}_h = {}^{ee}\mathbf{R}_s\,{}^{s}\mathbf{T}_m + {}^{ee}\mathbf{p}_s \times ({}^{ee}\mathbf{R}_s\,{}^{s}\mathbf{F}_m) - {}^{ee}\mathbf{R}_s\,{}^{s}\mathbf{R}_w\,{}^{w}\mathbf{r} \times {}^{w}\mathbf{g}. \tag{7} $$

The transformation from the sensor frame to the end-effector frame, ${}^{ee}\mathbf{R}_s$ and ${}^{ee}\mathbf{p}_s$, can be obtained from the specifications of the robot and the F/T sensor. The configuration used in this study is shown in Fig. 3.

3.2 Singularity management

While guiding the robot end-effector, the user may inadvertently move the robot towards a kinematic singularity, where the behavior of the robot is unpredictable for the user. Therefore, the behavior of the robot needs to be managed near the singularities to avoid the user losing the control of the robot.

An option for the management of singularities is the creation of a controller preventing the user to go guide the robot to a singularity. This can be accomplished, for example, by placing a repulsive force near the joint configurations where the singularities occur, forcing the robot motion to recede from the singular configuration area. This approach was implemented for the system presented. In initial experiments, however, it was found that this solution is not optimal for a natural human-robot interaction because the repulsive force will counteract the force applied by the human to the robot, and if the user is not aware of the existence of singularities and the operation near them, the user experience will be poor, because the user is not allowed to move the robot in all six degrees of freedom. Because the study focuses on novice users, in order to avoid the distortion of the results of the user experience study by the singularities, in this study the task area was designed to contain no singularities.

4. FORCE CONTROLLERS

Next, we present the two main approaches of force controllers examined in this paper, a proportional force controller and a virtual-tool based controller. The approaches are based on reviewed literature, but details are presented here for completeness, because these can not be found in the related literature.

4.1 Proportional controller

In the proportional controller, the velocity of motion is linearly dependent on the forces/torques applied by the human in the robot end-effector. The controller block diagram is illustrated in Figure 4. The measured forces and torques after gravity compensation (input to the controller) are passed through a proportional gain block, with gain K. After that, the result is subject to a threshold, t_F for the forces and t_T for the torques, which controls the minimum force causing motion.

Thresholding is necessary because there is uncertainty in the force measurement both due to measurement uncertainty and uncertainty in the gravity compensation. The

Figure 4: Proportional controller basic diagram

choice of the threshold is done based on the norm of the force or torque vector. Therefore, there is one unique threshold value for the force and one unique threshold value for the torque. If the threshold was used independently for each component of force and torque, the result would not be rotationally invariant.

The relationship between the input forces and torques and the output translational velocities v_t and rotational velocities v_R is thus given by

$$\mathbf{v_t} = \begin{cases} K(\|\mathbf{F}\| - t_F \frac{\mathbf{F}}{\|\mathbf{F}\|}) & if\ \|\mathbf{F}\| \geq t_F \\ 0 & if\ \|\mathbf{F}\| < t_F \end{cases}. \quad (8)$$

$$\mathbf{v_R} = \begin{cases} K(\|\mathbf{T}\| - t_T \frac{\mathbf{T}}{\|\mathbf{T}\|}) & if\ \|\mathbf{T}\| \geq t_T \\ 0 & if\ \|\mathbf{T}\| < t_T \end{cases}. \quad (9)$$

The motion produced by this controller has specific characteristics due to the absence of feedback and integration. Especially, the output velocities change abruptly when the input forces are changing rapidly, and thus the velocity can be discontinuous, which is physically impossible. However, the controller is valid in a physical system, as the system itself acts as an integrator. As a positive characteristic of the approach, it is likely that the proportional approach will produce motion which is highly controllable by the human operator due to the immediate relationship between applied force and velocity. Nevertheless, the motion is not smooth and the control strategy may produce stops in the robot motion during the human-robot interaction, depending on the values of gain and threshold. Another possible inconvenience related to the approach is the need to maintain constant force in order to produce motion at constant velocity.

4.2 Virtual-tool controller

The aim of the virtual-tool controller is to provide motion in a physically familiar fashion to the user. The dynamics of the robot motion are described using a virtual tool. Similar to the literature, we model the robot end-effector is modeled as a virtual point of a chosen mass at the robot end-effector center of mass in a free space environment. The real mass of the robot end-effector and hand needs not be considered because the gravity of the hand only influences the compensation of the forces. Thus, the acceleration is directly proportional to the compensated forces. In addition, a friction/damping term is included to provide environment resistance and deceleration in the absence of measured forces. The approach is then equivalent to the proposed approach using a spherical particle in a viscous medium as the virtual tool.

A basic diagram block of the virtual-tool controller is shown in Figure 5.

The compensated forces/torques measured by the F/T sensor, F/T, pass first through a gain block and a threshold block. The same block is used individually for each component axis of force and torque. In the first block, the gain K_1 is applied to the input forces/torques. The threshold is used

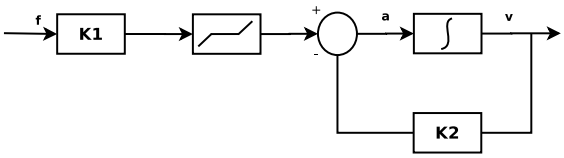


Figure 5: Virtual tool controller basic diagram

individually for each component to allow solving the resultant velocities using an ordinary differential equation. After thresholding, the negative damping is applied in a summation block, to obtain the acceleration values a_t and a_R. The damping gain is represented in the gain block K_2. It presents the feedback, where the velocities calculated in the time interval before, v_t and v_R, are used for the calculation of the new ones.

To summarize, the translational acceleration a_t and the rotational acceleration a_R of the motion depend on the current forces/torques and previous velocities as follows:

$$a_t \equiv \dot{v}_t = \begin{cases} K_1(F \pm t_F) - K_2 v_t & if\ |F| > t_F \\ -K_2 v_t & if\ |F| < t_F \end{cases} \quad (10)$$

$$a_R \equiv \dot{v}_R = \begin{cases} K_1(T \pm t_T) - K_2 v_R & if\ |T| > t_T \\ -K_2 v_R & if\ |T| < t_T \end{cases}. \quad (11)$$

These equations of the acceleration of the motion are ordinary differential equations. To retrieve the commanded velocities for the robot, the equation is solved numerically using the Runge-Kutta method of order 4.

The virtual-tool controller is expected to provide a smooth and natural motion with the use of less force for maintaining the motion with the same velocities compared to the proportional controller. These characteristics are due to the physically valid virtual tool model. The parameter values of the parameters will affect the characteristics considerably.

The gain and threshold applied to the force influence the smoothness of the motion, since they regulate the acceleration. A large gain causes highly sensitive motion, but the choice is also influenced by the fact that a too large gain can cause vibrations due to the latencies of the system. The damping parameter represented by the gain K_2 imposes a resistance value for the motion in the environment. The choice of the damping value will affect the smoothness of the motion and time required for the robot motion to stop after the human releases the robot end-effector. A high value for the gain K_2 involves a high impedance for the environment. Thus the user needs to apply more force and the robot motion will stop almost instantly when the user releases the robot end-effector. On the other hand, for a low value of the gain K_2 the impedance for the environment will be small, requiring less force from the user and more time for the motion to stop. This can cause a loss of positioning accuracy for the robot. The choice of parameters will be considered experimentally in Sec. 6.

5. SYSTEM DEMONSTRATION

This section demonstrates the functionality of a practical physical human-robot interaction system. The aim is to both introduce the setup of the experiments and to illustrate the system in operation.

The robot system used in the experiments is MELFA RV-3SB, a 6 DOF arm, equipped with a Schunk PG-70 parallel gripper. In the wrist of the robot, between the end-effector and the hand, a 6 DOF force/torque sensor by JR3 is attached. The system is shown in Figs. 1 and 3.

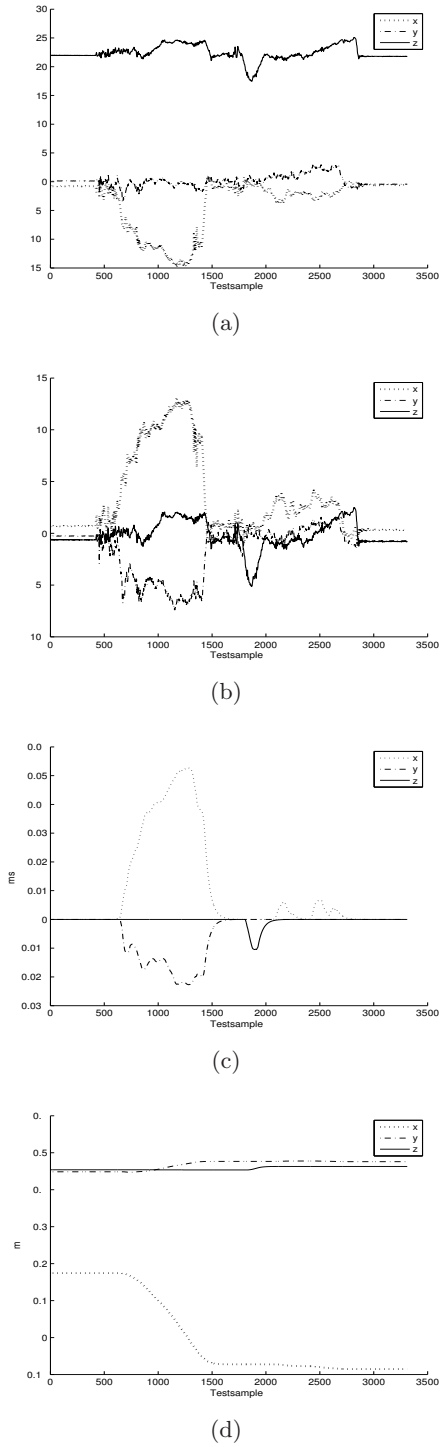
The demonstrated scenario is as follows: The task is to move the robot end-effector along the x-axis of the coordinate system. In the final position, a rotation of the end-effector of 90 degrees around the z-axis should performed. Even though the desired motion in the task description given to the human is along coordinate axes, the motion of the robot is not limited to these axes, but full 6 degree-of-freedom motion is available and the movement of the robot is based on manually guiding the robot hand by the user, as shown in Fig. 1. The virtual tool controller is used in the demonstration.

The task is composed of two phases for illustrative purposes. First, the use of linear forces are studied for translation, and second, torques for the rotation of the end-effector in the second phase of the experiment. However, as already mentioned, all degrees of freedom are allowed to move during the whole experiment.

A demonstration using the above task description was performed by a competent human operator. The measured forces and torques are shown in Figs. 6a and 7a, respectively. The effect of gravity compensation is shown in Figs. 6b and 7b, which show the forces and torques after gravity compensation, thus giving the estimates for the external human induced forces. It can be seen that there are small residual errors after gravity compensation, even after careful calibration of the compensation model. These are due to residual errors in the sensor measurements in different orientations.

After obtaining the human forces, they are transformed into velocities using the virtual tool controller. The commanded velocities are shown in Figs. 6c and 7c. It can be seen that only forces/torques exceeding the threshold values cause acceleration.

The resulting trajectory of the robot is shown in Figs. 6d and 7d, where the position and angles of rotation are presented respectively. The position is represented in Cartesian coordinates while the rotation is shown as the components of axis-angle representation. The graphs show that while the original force measurements are moderately noisy, the system trajectory is smooth.



(a)

(b)

(c)

(d)

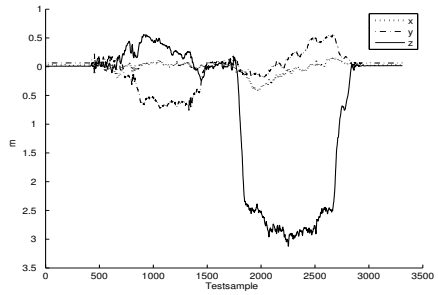
Figure 6: Forces: (a) Measured forces; (b) Human forces; (c) Velocities; (d) Position trajectory

6. EXPERIMENTS

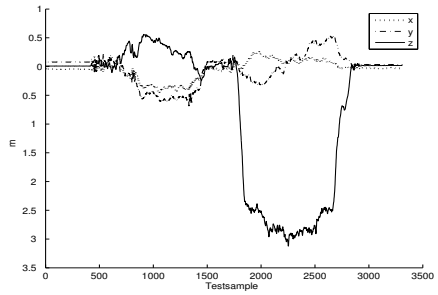
This section presents experiments aiming to address our primary research question, which qualities are preferable for a human in a physical HRI system. The approach taken is based on usability testing using a group of test subjects. Three characteristics were inspected: naturalness of the motion, sense of control of the robot and the positioning accuracy, henceforth denoted naturalness, control, and accuracy. These characteristics were deemed important due to their connection to both ease of use (accuracy of naturally available mental models) and effectiveness.

6.1 Experimental design

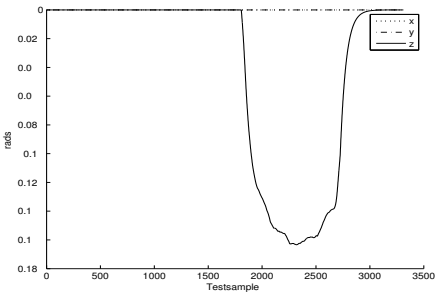
The group of test subjects included 20 participants. An announcement for volunteers was posted at a university notice board and all volunteers responding were accepted as participants. The test subjects included both students and university staff of both sexes, with age range from 20 to



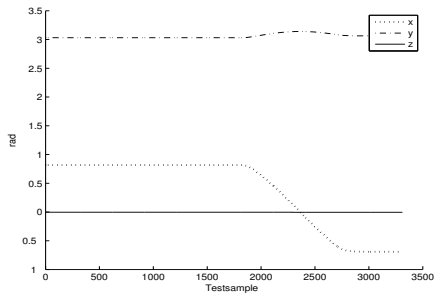
(a)



(b)



(c)



(d)

Figure 7: Torques: (a) Measured torques; (b) Human torques; (c) Velocities; (d) Rotation angles

53. The test subjects did not have previous experience with robotics.

The task performed was designed to correspond to a typical use case for a table top robot manipulator. More precisely, the task was to move the robot hand first to a given position of an object and then to a given final target position, simulating the grasping and moving of an object. The positions were indicated to the users visually. The length of the required trajectory was approximately 1 meter.

The users were instructed about the task and the studied characteristics before the beginning of the experiment. Each user was asked to complete the task with four different controllers described below. The order of controllers was randomized and they were presented to the user as controllers A, B, C, and D. Previously to the realization of the task, the user was allowed to interact with the robot for 30 seconds with each controller to familiarize with the controller. A written questionnaire was provided to the participants where the characteristics (naturalness, control, accuracy) were to be ranked from 1 (poor) to 5 (excellent). In addition, the users were asked which characteristic is most important for usability. Finally, the users were asked to rank the four control approaches. General comments from the users about the interaction were allowed during the execution of the experiment. To validity of the results for differences in scores between each two approaches was studied using a paired t-test with the p-value of 0.05.

Four different controllers were inspected, the proportional controller introduced in Sec. 4.1 and three variants of the virtual tool controller. The variants of virtual tool controller vary by the value of damping parameter K_2, which affects their characteristics. In the following, the approaches are denoted Proportional, Low_gain, High_gain and Variable_gain. For Proportional controller, the gain was set to obtain motion with very little vibration. In Low_gain approach, the parameter K_2 had a low value, which produces lower environment resistance while making the robot feel less stable. Respectively, the higher value of K_2 for the High_gain approach produces higher environment resistance while increasing the sense of control of the robot. Finally, Variable_gain approach used two different values for the parameter depending on the sensed human force. A higher value was used for low forces and a lower value for high forces. The basic idea is that this would allow the human continue producing the robot motion without effort (low resistance of the environment) and, on the other hand, the motion is rapidly stopped when the human releases the robot or decreases the force used to guide the robot (high resistance of the environment). The idea is somewhat similar to [15] where virtual tool parameters are adjusted over time for a collaborative positioning task.

6.2 Results

According to users' comments, naturalness of motion depends on the ease of use of the system. Table 1 shows the mean score and its deviation for naturalness.

The results shown in the table indicate that the virtual tool controllers are more natural than the proportional controller (with the significance level $p<0.01$). Although the Variable_gain approach has the largest mean score, there is no statistically significant difference between the three virtual tool controllers. According to the user comments, more effort (force) was necessary to produce motion with the Pro-

Table 1: Naturalness.

Approach	Mean	Standard Deviation
Proportional	2.15	0.81
Low_gain	3.55	1.00
High_gain	3.50	0.95
Variable_gain	3.65	1.31

portional controller, which was one of the reasons for lower naturalness. Additionally, we can hypothesize that the virtual tool approach feels more natural because it is equivalent to moving an object in a liquid, as mentioned earlier, and the users are likely to be familiar with the associated sensory percepts.

Feeling the sense of being in control of the robot can be defined as good response from the robot to the forces applied by the user to produce the desired motion. This requires both smooth motion and also the fact that the user feedback can easily change the speed or direction of motion. Table 2 shows the mean scores and their standard deviations for the four approaches. High_gain approach exhibits

Table 2: Control over the robot.

Approach	Mean	Standard Deviation
Proportional	2.95	1.15
Low_gain	3.60	0.88
High_gain	3.75	0.97
Variable_gain	3.25	1.47

the best sense of control, followed by Low_gain approach, however without statistically significant difference. Compared to Variable_gain approach, they are not statistically significantly superior with the available experimental data. However, compared to Proportional approach, High_gain approach is statistically significantly superior ($p \approx 0.04$). The superiority is likely mostly due to the smooth motion produced. The scores of Variable_gain approach have a large variance, showing a disagreement among the test subjects. Users who used lower forces to control the robot felt more in control as opposed to users who used higher forces resulting in fast motions. Thus, it seems that the approach would require more training and/or learning by the users compared to the other virtual tool approaches. The low scores of Proportional approach are mostly explained by the existence of discontinuities of velocity and vibrations induced by them.

The positioning accuracy of the motion was also determined by the written questionnaire, and is thus subject to user interpretation. Table 3 shows the results of this characteristic obtained for each approach.

Table 3: Positioning accuracy.

Approach	Mean	Standard Deviation
Proportional	3.30	1.26
Low_gain	3.25	0.91
High_gain	3.85	0.93
Variable_gain	3.20	1.44

High_gain approach has the largest scores on average. Even though differences appear, they are not statistically different when compared to Proportional and Variable_gain approaches. Nevertheless, the superiority of High_gain approach seems to indicate that high damping is useful for the positioning accuracy. Similar to the previous characteristic, Variable_gain approach exhibits large variations in scores in the test group. The reason for the differences is identical to the case in the sense of being in control discussed above. Thus, the conclusion that the approach needs more training holds also here.

The user ranking for the most important characteristic for the usability shows no significant differences between the characteristics. This, with supporting user comments, demonstrates that all three characteristics are important for usability. However, this ranking also shows the users priority for the overall best approach. While there was no clear winner as the overall best approach, Variable_gain and Low_gain approaches were deemed best by users who valued naturalness the most, Proportional approach by the users who valued positioning accuracy, and High_gain approach by the users who tried to balance the three characteristics. These results highlight the need of user-based measures in assessing the usability, in addition to physically grounded measures such as required time and accuracy.

In conclusion, the choice of an approach should depend on the application. As a summary, the virtual tool approach with high damping produces a robot motion with a good naturalness, a good sense of control positioning accuracy. With lower damping, the naturalness of the approach increases further, while decreasing the positioning accuracy. Variable damping seems to induce the most natural motion but is inferior in the sense of control and positioning accuracy, and seems to require more training for the user. Furthermore, virtual tool approaches seem to outperform the proportional velocity control due to their more intuitive user interface, which is likely to be more familiar to the users in their everyday life. In a general application, using the virtual tool approach with large enough damping seems to provide a good overall solution.

7. CONCLUSION

The goal of this paper was to study the qualities preferable for a human in a physical HRI system, where a robot arm is controlled using a wrist-mounted force sensor. Controllers with different characteristics were first described, and the significance of the different characteristics was examined through a user study.

The results indicate that the virtual tool approach for controlling the motion is advantageous through its correspondence to motion of physical systems, which makes it a natural way of control for the users. In other words, even the novice users recognize the mental model used in the control from their everyday life. Thus, the approach is likely to be successful in general service robotic applications even with non-expert users. The experiments also supported that novice users can learn to perform effective physical HRI with the current state-of-the-art technology.

It was also found that the choice of a physical HRI controller should depend on the target application, because a trade-off is necessary between the conflicting goals of naturalness of motion and positioning accuracy. Thus, it seems that a single controller is not likely to succeed in providing ultimate ease of use together with absolute accuracy.

Finally, the study supports the fact that in addition to well-defined quantitative measures such as effectiveness (time required) and absolute positioning accuracy, user experience is greatly affected by several other factors, which need to be measured using questionnaire type studies. It remains a topic of further study to collect these factors for physical HRI. When service robots are introduced to everyday environments, these factors will be essential in providing products the users are comfortable with and willing to use.

8. ACKNOWLEDGMENTS

The research leading to these results has received funding from the European Community's Seventh Framework Programme under grant agreement n° 215821.

9. REFERENCES

[1] H. Ben Amor, E. Berger, D. Vogt, and B. Jung. Kinesthetic bootstrapping: Teaching motor skills to humanoid robots through physical interaction. In *KI 2009: Advances in Artificial Intelligence, 32nd Annual German Conference on AI*, Paderborn, Germany, September 2009.

[2] A. Billard, S. Calinon, R. Dillmann, and S. Schaal. Robot programming by demonstration. In B. Siciliano and O. Khatib, editors, *Springer Handbook of Robotics*, chapter 59. Springer, 2008.

[3] S. Calinon and A. Billard. Statistical learning by imitation of competing constraints in joint space and task space. *Advanced Robotics*, 23(15):2059–2076, 2009.

[4] A. Casals, M. Frigola, J. Poyatos, and J. Amat. Force and contact based control for human robot interaction. In *11th International Conference on Advanced Robotics*, pages 1514–1519, Coimbra, Portugal, June–July 2003.

[5] A. De Santis, B. Siciliano, A. De Luca, and A. Bicchi. An atlas of physical human-robot interaction. *Mechanism and Machine Theory*, 43(3):253–270, March 2008.

[6] G. Ferretti, G. Magnani, and P. Rocco. Assigning virtual tool dynamics to an industrial robot through an admittance controller. In *14th International Conference on Advanced Robotics*, Munich, Germany, June 2009.

[7] T. Fukuda, Y. Fujisawa, K. Kosuge, and K. Uehara. Manipulator for man-robot cooperation. In *International Conference on Industrial Electronics, Control and Instrumentation*, pages 996–1001, Kobe, Japan, October–November 1991.

[8] R. Ikeura, H. Monden, and H. Inooka. Cooperative motion control of a robot and a human. In *3rd IEEE International Workshop on Robot and Human Communication*, pages 112–117, Nagoya, Japan, July 1994.

[9] H. Kazerooni. Human-robot interaction via the transfer of power and information signals. *IEEE Transactions on Systems, Man, and Cybernetics*, 20(2):450–463, 1990.

[10] P. Kormushev, S. Calinon, and D. Caldwell. Imitation learning of positional and force skills demonstrated via kinesthetic teaching and haptic input. *Advanced Robotics*, 2011. To be published.

[11] K. Kosuge, Y. Fujisawa, and T. Fukuda. Control of robot directly maneuvered by operator. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 49–54, Yokohama, Japan, July 1993.

[12] D. MacKenzie and R. Arkin. Evaluating the usability of robot programming toolsets. *International Journal of Robotics Research*, 17(4):381–401, 1998.

[13] P. Pastor, H. Hoffmann, T. Asfour, and S. Schaal. Learning and generalization of motor skills by learning from demonstration. In *IEEE International Conference on Robotics and Automation*, Kobe, Japan, May 2009.

[14] A. Steinfeld, T. Fong, D. Kaber, M. Lewis, J. Scholtz, A. Schultz, and M. Goodrich. Common metrics for human-robot interaction. In *1st ACM SIGCHI/SIGART conference on Human-robot interaction*, pages 33–40, Salt Lake City, Utah, USA, March 2006.

[15] T. Tsumugiwa, R. Yokogawa, and K. Hara. Variable impedance control with regard to working process for man-machine cooperation-work system. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1564–1569, Maui, HI, USA, October–November 2001.

[16] H. Yanco, J. Drury, and J. Scholtz. Beyond usability evaluation: Analysis of human-robot interaction at a major robotics competition. *Human-Computer Interaction*, 19(1):117–149, 2004.

Assessing grasp stability based on learning and haptic data

Yasemin Bekiroglu, Janne Laaksonen, Jimmy Alison Jørgensen, Ville Kyrki and Danica Kragic

Abstract— **An important ability of a robot that interacts with the environment and manipulates objects, is to deal with the uncertainty in sensory data. Sensory information is necessary to, for example, perform on-line assessment of grasp stability. We present a method for assessing grasp stability based on haptic data and machine learning methods. In particular, we study the effect of different sensory streams to grasp stability. This includes object information such as shape; grasp information such as approach vector; tactile measurements from fingertips and joint configuration of the hand.**

Sensory knowledge affects the success of grasping process both in the planning stage (before a grasp is executed) and during the execution of the grasp (closed-loop on-line control). In this paper, we study both of these aspects. We propose a probabilistic learning framework for assessing grasp stability and demonstrate that knowledge about grasp stability can be inferred using information from tactile sensors. Experiments on both simulated and real data are shown. The results indicate that the idea of exploiting the learning approach is applicable in realistic scenarios which opens a number of interesting venues for the future research.

Index Terms— **Grasping, Force and Tactile Sensing, Learning and Adaptive Systems.**

I. Introduction

Grasping is an essential skill for a general purpose service robot, working in an industrial or home-like environment. If object parameters such as pose, shape, weight and/or material properties are known, grasp planning using analytical approaches can be employed [1]. In unstructured environments these parameters are uncertain, which presents a great challenge for the current state-of-the-art approaches. Extraction and appropriate modeling of sensor data can alleviate the problem of uncertainty. Many approaches to robotic object grasping exist and most of these have been designed for dealing with known objects. To estimate the shape and pose of an object, visual sensing has been used [2, 3, 4, 5, 6, 7]. However, the accuracy of vision is limited, for example due to imperfect calibration and occlusions. Small errors in object pose are thus common even for known objects and these errors may cause failures in grasping. These failures are commonly difficult to prevent at the grasp execution stage if a hand is not equipped with proper sensors. Tactile and finger force sensors can be used to reduce some problems but are uncommon in practice [8, 9]. Due to uncertainty in the observations, a grasp may fail due to slippage or collision even when all fingers have adequate contact forces and the hand pose with respect to the object is not very different from the planned one.

The main contribution of our work is a new approach that incorporates knowledge of uncertainty in the observations when predicting the stability of a grasp. We show how grasp stability can be assessed based on data extracted both *prior to* and *during* execution. The data contain object information such as shape; grasp information such as approach vector; and online sensory and proprioceptive data including tactile measurements from fingertips and joint configuration of the hand. In real world scenarios the observations acquired from the environment are erroneous and associated with a degree of uncertainty. Our goal is to create a system capable of performing prediction of grasp stability from real world sensory streams. In order for the system to be robust, the uncertainty in the observations needs to be taken into account. Probabilistic methods provide a framework for dealing with uncertainty in a principled manner and will to this end provide the foundation that our system is built upon. Our aim is to model the embodiment specific and inherently complex relationship between grasp stability and the available sensory and proprioceptive information. Our approach is a learning based framework and relies on having a training data-set which can be assumed to sample the domain of possible scenarios well. This poses a challenge as acquiring such data is associated with a significant cost with respect to time and computation. In order to alleviate this problem we use a simulator from which we can generate a large set of synthetic training data in a controlled environment with relative ease. The approach of using synthetic training data is justified by performing inference on real-world examples. Moreover, the generalizability of the grasp stability estimation is experimentally evaluated. The results demonstrate that the stability estimation generalizes relatively well to new objects even with a moderate number of objects used in training. In summary, the paper demonstrates that knowledge about grasp stability can be inferred using information from tactile sensors while grasping an object before the object is further manipulated. This is very useful since, if an unstable grasp is predicted, objects can be regrasped before attempting to further manipulate them.

In the following section, the contributions of our work are discussed in detail in relation to the state-of-the-art work in

Y. Bekiroglu and D. Kragic are with the Centre for Autonomous Systems and Computation Vision and Active Perception Laboratory, School of Computer Science and Communication-CSC, KTH, Stockholm, Sweden. J. Laaksonen and V. Kyrki are with the Department of Information Technology, Lappeenranta University of Technology, Finland. J. A. Jørgensen is with Robotics group, Maersk Mc-Kinney Moller Institute, University of Southern Denmark, Denmark. Janne.Laaksonen,kyrki@lut.fi, yaseminb,danik@csc.kth.se, jimali@mmmi.sdu.dk.

This work was supported by EU through the project CogX, FP7-IP-027657, and GRASP, FP7-IP-215821 and Swedish Foundation for Strategic Research.

the area. This is followed by a presentation of the theoretical framework in Section III and the employed learning methodology. In Section IV the simulator, the database and the real data collection are described. We present the results of experimental evaluation in Section V and conclude our work in Section VI.

II. Contributions and Related work

In robotic object grasping there has been a lot of effort during the past few decades [1]. Grasp stability analysis is a tool often used in grasp planning, where the grasp is planned using grasp quality measures derived from stability analysis. Most of the work on grasp stability assessment relies on analytical methods and focuses on rigid objects, albeit some work has considered the analysis of grasps on deformable objects [10]. Compared to our approach, the analytical methods require exact knowledge of the contacts between the hand and the object to estimate the stability of a grasp.

Most of the grasp planning approaches tested in simulation have the common property of using a strategy that relies on the object shape. Modeling object shape with a number of primitives such as boxes, cylinders, cones, spheres [11, 4], or superquadrics [12] reduces the space of possible grasps. The decision about the suitable grasp is made based on grasp quality measures given contact positions. However, none of these approaches provide a principled way of dealing with uncertainties that arise in dynamic scenarios or the errors inherent to simplification with primitives, which can potentially be solved using tactile feedback. This is also the main objective and contribution of the work presented here.

One of the issues often faced in household scenarios are deformable objects. Planning grasps for these type of objects is not at all as well studied as rigid objects. Examples can be found in literature, such as [13], where the deformation properties of objects are learned and then a suitable grasping force is planned for the associated objects.

To cope with the fact that the exact knowledge of the object and the hand is not available, we employ tactile sensors measuring a range of pressure levels. Tactile sensing has been used for various purposes in prior studies and we focus on the use of tactile sensors in the remaining survey of the related work. There are recent examples which perform grasp generation from visual input and use tactile sensing for closed loop control once in contact with the object. For example, the use of tactile sensors has been proposed to maximize the contact surface for removing a book from a bookshelf [14]. Application of force, visual and tactile feedback to open a sliding door has been proposed in [15]. In our work the main difference is that the tactile sensors are used to assess the stability of a grasp. Thus, rather than using the tactile data for control, we use it in order to reason about grasp stability.

Learning aspects have been considered in the context of grasping mostly for the purpose of understanding human grasping strategies. In [16], it was demonstrated how a robot system can learn grasping by human demonstration using a grasp experience database. The human grasp was recognized with the help of a magnetic tracking system and mapped to the kinematics of the robot hand using a predefined lookup-table. Another approach is to use vision. However, measuring the contact between object and hand accurately is a non-trivial task. The system in [2] learns grasping points by using hand labeled training data in the form of image regions which indicate good grasping regions. A probabilistic decision system is employed on previously unseen objects to determine a good grasping point or region. In [3], vision is used to create grasp affordance hypotheses for objects and refine the grasp affordance hypotheses through grasping. The result is a set of grasps that will produce good grasps on a specific object.

Current learning approaches using tactile sensors are focused on either determining the properties of objects [17, 18, 19] or object recognition [19, 20, 21, 22]. Different properties of objects give valuable information that can be further used in grasp stability analysis. In [17], the pose of the object is determined using a particle filter technique based on the tactile information gained from the contacts between a gripper and the object. Similar work was presented by Hsiao et al. [23] where object localization was performed with knowledge of tactile contacts on specific objects. In [18], the surface type (edge, flat, cylindrical, sphere) of the tactile contact is determined using a neural network. In [19], tactile information extracted from the sensors on a two fingered gripper is used to determine the deformation properties of an object. However, learning or analyzing such object properties through tactile sensors do not answer the question of grasp stability directly compared to the work presented here.

Work on using tactile sensors for recognition of manipulated objects has been reported rather recently. The main approach is to use multiple grasp or manipulation attempts and then learn the object through the haptic input from the manipulations or grasps. Current approaches use either one shot data from the end of the grasps [21, 22] or temporal data collected throughout the grasp or manipulation execution [19, 20]. In [21], a bag-of-words approach is presented which aims to identify objects using touch sensors available on a two fingered gripper. The approach processes tactile images collected by grasping objects at different heights. In [22], a similar approach is taken for a humanoid hand. A more traditional approach to learning is employed with features extracted from tactile images in conjunction with hand joint configurations as input data for the object classifier. In [20] entropy is used to study the performance of various features in order to determine the most useful features in recognizing objects. In this case, a plate covered with tactile sensor was used as the manipulator. However, the object recognition using the recognized good features did not perform as well as in the other presented works. Thus, no attempts have been made on using tactile sensors placed on a robotic hand to predict the stability of a grasp. We have presented the idea of grasp stability prediction using tactile sensors in [24] with some initial results and we extend our work in this paper.

III. Problem formulation and modeling

Determining grasp stability is difficult when factors affecting the stability are uncertain or unknown. We show that with a probabilistic approach it is possible to assess grasp stability using tactile measurements. Mapping from tactile sensor measurements to grasp stability is complex and not injective because of variability in object parameters, grasp and hand types, and the uncertainty inherent in the process. Thus, we consider grasp stability as a probability distribution

$$P(S|H(t), j(t), O, G), \tag{1}$$

where grasp stability, denoted by S, depends on different measured and/or known factors. The factors taken into account in our model are: i) H, force/pressure measurements from tactile sensors; ii) j, joint configuration of the hand; iii) O, object information, e.g., object identity or shape class; and iv) G, information relevant to the grasp, e.g. approach vector and/or hand preshape. Grasp stability, S, is a discrete variable with two possible states: a grasp is either stable or unstable, while the other variables can be discrete or continuous. Our goal is to assess the effect of factors in Eq. (1) to grasp stability by considering different subsets of the variables.

We study the problem using both instantaneous measurements of variables and time-series measurements. With instantaneous measurements, the stability is assessed only from the instant when the robot hand is static and has closed around the object. This approach is referred to as one-shot classification. In contrast, the time-series approach takes into account measurements generated during the whole grasping sequence. The variables H and j are thus represented from time t_0 to t_n where t_0 and t_n represent the start and the end of the grasping sequence respectively. In the case of one-shot classification, we use the measurements once the hand has reached a static configuration, an approach similar to [21]. Thus, we compare the distribution defined by Eq. (1) to one which discards the time series:

$$P(S|H(t_n), j(t_n), O, G). \tag{2}$$

We show that both approaches described by Eq. (1) and Eq. (2) are valid and that grasp stability can be assessed based on them. To study the contribution of object O and grasp knowledge G, we have set up a hierarchy as depicted in Fig. 1. The hierarchy is divided into levels, each with increasing amount of sensory information being available. At the top level of the hierarchy only the information related to the hand itself, H, and j is used. Thus, we estimate

$$P(S|H, j) = \int\int P(S|H, j, O, G)\, p(G|O)\, p(O)\, dO\, dG\ . \tag{3}$$

Considering only sensor information, the overall distribution will be somewhat uninformative — there is significant uncertainty as the same sensor readings can be associated with both stable and unstable grasps for different objects, grasp approach vectors and hand preshapes. Subsequently, when more pieces of information are considered, the estimation of the distribution should be more specific resulting in better discrimination. At the second level, we consider that object

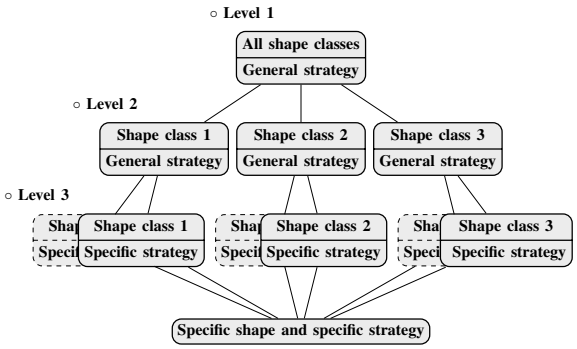


Fig. 1. Hierarchical recognition of grasp stability taking into account different type of sensory knowledge.

shape or object instance are known:

$$P(S|H, j, O) = \int P(S|H, j, O, G)\, p(G)\, dG\ . \tag{4}$$

Finally, at the third level we consider knowledge about the applied grasp, and estimate the stability through $P(S|H, j, O, G)$. Since knowledge of all the variables present in Eq. (1) is assumed, the uncertainty in the stability estimation is expected to decrease.

In the rest of the section, we describe methods for estimating the density functions using a classification approach. Support Vector Machines and AdaBoost are used to model the instantaneous model, according to Eq. (2) while Hidden Markov models are used for the general time series case, according to Eq. (1). Although the probabilistic framework is presented as a method to estimate grasp stability using haptic data, it is also possible to use the proposed framework with other types of sensory information.

A. Feature representation

First, we describe the input features for the classifiers. In this work, a three-fingered Schunk Dextrous Hand (SDH) with seven degrees of freedom and equipped with six two-dimensional Weiss Robotics pressure sensitive tactile pads [25] is used as a demonstration hardware platform. Tactile measurements are recorded from the first contact with the object until a steady state is reached. The whole measurement sequence is denoted by $x_1^i, \ldots, x_{T_i}^i$, where i is the index of the measurement. For one-shot classification, tactile measurements at the steady-state is used and denoted $x_{T_i}^i$. Training data is generated both in simulation and on real hardware and will be presented in Section IV. The notation used in this paper is as follows:

- $D = [o_i], i = 1, \ldots, N$ denotes a data set with N observation sequences.
- $o_i = [x_t^i], t = 1, \ldots, T_i$ is an observation sequence.
- $x_t^i = [M_f^{i,t} j_v^{i,t}], f = 1, \ldots, F, v = 1, \ldots, V$ is the observation at time instant t given the i-th sequence; F

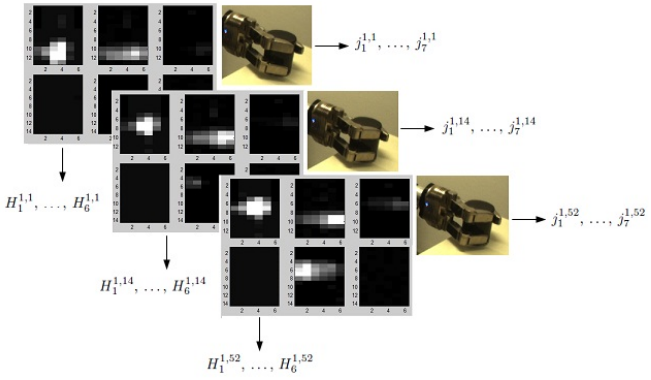


Fig. 2. An example grasping sequence of a cylinder and the corresponding tactile measurements.

is the number of tactile sensors and V is the number of joints of the robot hand.

- $M_f^{i,t}$ includes the moment features extracted from the tactile readings $H_f^{i,t}$ on the sensor f at time instant t given the i-th sequence. Details about the extraction of these features are given later in this section.
- $j_v^{i,t}$ is a joint angle at time instant t given the i-th sequence.

The acquired data thus consists of tactile readings $H_f^{i,t}$ and joint angles of the hand $j_v^{i,t}$. For the Schunk Dextrous Hand, we store $3 \times (14 \times 6)$ readings on proximal and $3 \times (13 \times 6)$ on distal sensors, and seven parameters representing the pose of the hand given the joint angles. Example images from the tactile sensors are shown in Fig. 2. The tactile images in the figure represent a stable grasp of a cylinder.

Tactile data is relatively high dimensional and redundant. Thus, we borrow ideas from image processing and consider the two-dimensional tactile patches as images. Each tactile image is represented using image moments. The general parameterization of image moments is given by

$$m_{p,q} = \sum_z \sum_y z^p y^q f(z, y), \tag{5}$$

where p and q represent the order of the moment, z and y represent the horizontal and vertical position on the tactile patch, and $f(z, y)$ the measured contact. We compute moments up to order two, $(p + q) \in \{0, 1, 2\}$, for each sensor array separately. These then correspond to the total pressure and the distribution of the pressure in the horizontal and vertical direction. Thus, there are in total six features for each sensor resulting in an observation $x_t^i \in \mathbb{R}^{6F+V}$. Normalizing the feature vector is a common step in machine learning methods. In our case, moment features and finger joint angles are normalized to zero-mean and unit standard deviation. Normalization parameters are calculated from the training data and then used to normalize the testing sequences.

B. One-shot recognition

In this section, we examine the learning of grasp stability based on tactile measurements acquired at the end of a grasping sequence, that is, once the final grasp has been applied to the object. We claim that if successful separation between stable and unstable grasps can be learned from examples, one-shot classification can determine the stability of the grasp from any haptic observation x_t^i measured during a grasp. This information can then be used in grasp control to determine when the robot hand has reached a stable configuration.

In this paper, two types of non-linear classifiers, AdaBoost and Support Vector Machine (SVM), are used in the experiments to demonstrate the ability to learn the stability of the grasps. AdaBoost and SVM were the best performing classifiers in [26]. AdaBoost is a boosting classifier, developed by Freund and Schapire [27], that works with multiple so-called weak learners to form a committee that performs as the classifier. Here, we use AdaBoost implementation from [28].

Support vector machine classification [29, 30] is also suitable for the problem. SVM is a maximum margin classifier, i.e. the classifier fits the decision boundary so that maximum margin between the classes is achieved. This guarantees that the generalization ability between the classes is not lost during the training of the SVM classifier. We use the libSVM implementation presented in [31]. Another critical feature of the SVM for our use is the ability to use non-linear classifiers instead of the original linear hyper-plane classifier. Non-linearity is achieved using different kernels, in this study the radial basis function (RBF)

$$K(x_i, x_j) = e^{-\gamma \|x_i - x_j\|^2}, \quad for \quad \gamma > 0, \tag{6}$$

is used as the kernel for SVM. Moreover, as an extension to the basic two-class SVM, probabilistic outputs for SVM are used to analyze the results given by the SVM. This idea was first presented in [32]. The SVM output $y(\mathbf{x})$ is converted to a probability according to

$$p(t = 1|\mathbf{x}) = \sigma(\Gamma y(\mathbf{x}) + \Lambda), \; y(\mathbf{x}) = K(\mathbf{w}, \mathbf{x}) + b \, , \tag{7}$$

where parameters Γ and Λ are estimated using training data, and $\sigma(\cdot)$ is the logistic sigmoid function. This probability is thus related to the earlier general discussion by

$$P(S = stable|H(t), j(t), O, G) = p(t = 1|\mathbf{x}) \, . \tag{8}$$

C. Temporal recognition using HMMs

Time-series grasp stability assessment is performed using Hidden Markov models (HMMs) [33]. We construct two HMMs: one representing stable and one unstable grasps. Classification of a new grasp sequence is performed by evaluating the likelihood of both models and choosing the one with higher likelihood. For the HMM, we use the classical notation $\lambda = (\pi, A, B)$ where π denotes the initial probability distribution, A is the transition probability matrix

$$A = a_{ij} = P(S_{t+1} = j|S_t = i), i, j = 1 \ldots N, \tag{9}$$

and B defines output (observation) probability distributions $b_j(x) = f_{X_t|S_t}(x|j)$ where $X_t = x$ represents a feature-vector for any given state $S_t = j$. In this work, we evaluate both ergodic (fully connected) and left-to-right HMMs.

The estimation of the HMM model parameters is based on the classical Baum-Welch procedure. The output probability distributions are modeled using Gaussian Mixture Models (GMMs):

$$f_X(x) = \sum_{k=1}^{K} w_k \frac{1}{2\pi^{L/2}\sqrt{|C_k|}} e^{-\frac{1}{2}(x-\mu_k)^T C_k^{-1}(x-\mu_k)}, \quad (10)$$

where $\sum_{k=1}^{K} w_k = 1$, μ_k is the mean vector and C_k is the covariance matrix for the k-th mixture component. The unknown parameters $\theta = (w_k, \mu_k, C_k : k = 1...K)$ are estimated from the training sequences $o = (x_1, ...x_T)$. Initial estimates of the observation densities in Eq. (10) affect the point of convergence of the reestimation formulas. Depending on the structure of the HMM (ergodic vs left-to-right), we use a different initialization method for the parameters of the observation densities. The two initialization procedures are given below:

- For an ergodic HMM, observations are clustered using k-means. Here, k is equal to the number of states in the HMM and each cluster is modeled with a GMM using standard expectation maximization. Initial parameters for the GMMs are found using k-means algorithm.
- For a left-to-right HMM, each observation sequence is divided temporally into equal length subsequences. Then, each GMM is estimated from the collection of corresponding subsequences. Thus, the GMMs represent the temporal evolution of the observations. Initial parameters are found as in the case of an ergodic HMM.

IV. Data collection

For a learning system to achieve good generalization capabilities, relatively large training data is typically required. Generating large datasets on real hardware is time consuming and in robotic grasping generating repeatable experiments is difficult due to the dynamics of the process. However, if suitable models are available, simulation can be used for generation of data for both training the learning system and performance evaluation. In our work, we generate both simulated and real training data as explained below.

A. The simulator

The grasp simulator RobWorkSim, described in [34], is used to generate training data including tactile measurements. The simulator is used in combination with the Open Dynamics Engine (ODE) physics engine and provides support for simulating articulated hands, PD joint controllers, grasp quality measures, camera sensors, range sensors and tactile sensors. The primary motivation for using RobWorkSim over the more widely used GraspIt! [35], is the integrated support for tactile array sensors.

1) Tactile sensor model: The tactile array sensor simulation in RobWorkSim is an experimental model that transforms the point contacts of the ODE to sensor measurements by describing the deformation of the sensor surface given a point force $\mathbf{f}$ applied perpendicular to it. The model was originally described in [36]. The model assumes that the deformation or response is linear with the magnitude of the point force, which is a fair assumption for small forces. Given the deformation function $h(x, y)$ where x and y are specified relative to the center (a, b) of the contact, the total deformation of the surface of an array of rectangular texels with size (A, B) can be found by integrating over the surface of each texel by

$$g_{m,n}(a, b) = \int_{(A-\frac{1}{2})m-a}^{(A+\frac{1}{2})m-a} \int_{(B-\frac{1}{2})n-b}^{(B+\frac{1}{2})n-b} h(x, y)dxdy, \quad (11)$$

where (a, b) is the center point of the contact and (m, n) is the texel index. This surface integration is approximated using the rectangle method. Point force experiments on the real sensors suggested that the deformation decreased with the inverse of the square of the distance from the point force. We use an isotropic function to approximate the deformation of the sensor surface

$$h(x, y) = (\mathbf{f} \cdot \mathbf{n_{texel}})\ max(-\beta + \frac{\alpha}{1 + x^2 + y^2}, 0), \quad (12)$$

where (x, y) is specified relative to (a, b) and $\mathbf{n_{texel}}$ is the normal of the texel on which the point force $\mathbf{f}$ is applied. The parameters (α, β) were found by fitting the model to experimental data extracted from real sensors. Fig. 3 shows a visual comparison between the real and the simulated sensor output where a sharp edge was pressed against both sensors.

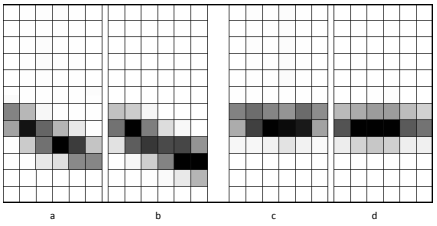


Fig. 3. Measured (a and c) versus simulated (b and d) sensor values. The tactile images were generated by pressing a sharp edge onto the sensor surface.

Assessing grasp quality requires taking properties of the hand (orientation, joint configuration, friction, elasticity, grasping force) and object (shape, mass, friction, contact locations and area, contact force) into account. In the simulated environment these parameters are known. We use a widely known grasp quality measure based on the radius, ϵ, of the largest enclosing ball in the grasp wrench space (GWS). We construct the GWS as proposed in [37] by calculating the convex hull over the set of contact wrenches $\mathbf{w}_{i,j} = [\mathbf{f}_{i,j}^T\ \lambda(\mathbf{d}_i \times \mathbf{f}_{i,j})^T]^T$, where $\mathbf{f}_{i,j}$ belongs to a representative set of forces on the extrema of the friction cone of contact i. $\mathbf{d}_i$ is the vector from the torque origin to contact i and λ weighs the torque quality relative to the force quality.

It is not obvious how to determine λ due to the differences between forces and torques. We therefore calculate force space and torque space independently and use the radius of the largest enclosing ball in each of these to give a 2 dimensional quality value $(\epsilon_f, \epsilon_\tau)$ for each grasp. A third quality measure ϵ_{cmc} based on the distance between the centroid of the contact polygon C and the center of mass CM of the object [38] is used: $\epsilon_{cmc} = ||CM - C||$. This measure captures the same properties as the torque measure, however it is more robust with regard to the point contact output of the simulator. Stable grasps are defined as those for which all three quality values are within a certain threshold. The thresholds have been determined experimentally.

B. Generating training data in simulation

The database includes examples of stable and unstable grasps on different objects. We examine stability starting from the most general case in the hierarchy specified in Fig. 1 and continue by including information about subsequent properties until reaching the most specific case. At the top level of the hierarchy, data is generated on objects with different shapes using approach vectors generated uniformly from a sphere, referred to as a spherical strategy. At the second level, the shape information is given, hence grasps are generated separately per object shape with the spherical strategy. At the third level, the approach vector is formed based on the object shape, namely side or top grasps are applied with more than one preshape. At the bottom level, the preshape is also chosen per object shape and approach vector. Fig. 4 shows examples of objects that are included in the database.

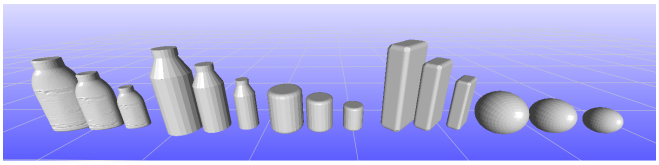


Fig. 4. Objects in simulation were generated in three sizes (75%,100%,125%): Hamburger sauce, Bottle, Cylinder, Box, Sphere.

Each grasping sequence in the database is generated by placing the hand in a specific configuration with respect to the object and then closing the fingers. For the recognition that relates to levels 1 and 2 in the recognition hierarchy (see Fig. 1), a simple spherical grasp strategy with a randomly chosen preshape is used. The spherical grasp strategy generates the approach direction for the hand by sampling the unit sphere around the center of mass of the object. Each sample then consists of a vector pointing toward the center of mass of the object.

The strategy and the preshapes used for level 3 in the recognition hierarchy are shape specific. Therefore strategies where developed for each shape used in the experiments. The hand preshapes for level 3 were generated with finger joint values in the interval $([-90; -70], [-10; 10])^\circ$, where the 7th joint was one of $90^\circ, 60^\circ, 0^\circ$ as shown in Fig. 5.

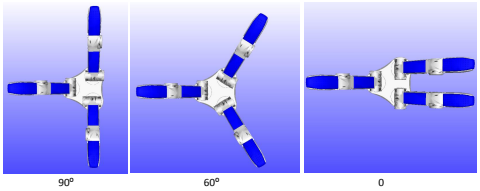


Fig. 5. Hand configuration when the 7'th joint is at 90°, 60° and 0°

The following grasp strategies are applied for the shape primitives:

- Sphere - The approach directions are sampled randomly from the unit sphere with origin in the center of gravity of the object. Both the ball preshape (60°) and the parallel preshape (0°) were used.
- Cylinder - The object is approached either from the top or from the side. When approaching from the top, a ball grasp preshape is used and the approach direction is pointing towards the object center of mass. For side grasps, the approach is sampled with an angle of $0 - 20^\circ$ with respect to the horizontal plane, pointing towards the center of mass of the object. The preshape in the side grasp uses an angle of 0 on joint 7, so that a parallel grasp is obtained.
- Box - The object is approached using a vector lying in the plane defined by the world z-axis and the longest axis of the box and pointing toward the center of gravity. A parallel preshape of the hand is used.

In addition, two natural objects, the hamburger sauce and the bottle (see Fig. 4), used the same strategy as the cylinder. The tactile information and the joint configuration are recorded from simulation at regular time intervals.

In general, the performance of the simulation is largely dependent on the level of detail of the geometries in both hand and objects. In our setup generating a simulated grasp using a modern quad core computer took approximately 2 seconds.

C. Generating training data on a robot

The real world experiments show the feasibility of assessing grasp stability on physical robot platforms. The experiments aim to serve as a proof-of-concept rather than assessing the exact performance rates in different use cases. The experimental evaluation on real data follows the methodology used in simulation such that similar objects and same grasp types are used. The objects are placed such that they are initially not well centered with respect to the hand to assess the ability of the methods to cope with the uncertainty in pose estimation. A few example grasps are shown in Fig. 6. The real data includes side grasps on the objects in Fig. 7 with the preshape shown in Fig. 5 where the 7th joint is 0°. After preshaping, the hand closes the fingers with equal speeds while limiting the maximum torque of each actuator until reaching a static state where the object does not move or a fully closed hand configuration is reached. The latter occurs in the case of an unsuccessful grasp.

Fig. 6. A few examples from the execution of real experiments.

Fig. 7. Objects used in real experiments, with last three deformable.

Tactile readings and corresponding joint configurations were recorded starting from the first contact until a static state is achieved. To generate stable/unstable label for a grasp, the object is lifted and rotated $[-120°, +120°]$ around the approach direction. The grasps where the object is dropped or moved in the hand were labeled as unstable. 100 stable and 100 unstable grasps were generated for each object. Data processing, training and classification followed the same methodology as described for the simulated data.

V. Experiments

We begin the experimental part by describing a simple demonstration scenario to show that the proposed approaches are viable in real applications. As the main experimental contribution, we proceed to study the effect of different types of information for the estimation of grasp stability.

A. Demonstration

The feasibility of the approach is demonstrated in a realistic scenario. The demonstration is included to better show how the proposed methodology can be integrated in a real robotic system. Quantitative evaluation of the methodology is presented after the demonstration.

A vision based system can provide information about the specific objects in the scene and their pose [4, 5, 6] or potential grasping points on the object [39, 7]. In our previous work, we have shown how this can be done for known [4], unknown [5, 6] and familiar objects [39, 7]. However, in the the previous work there were many cases that resulted in unsuccessful grasps. One example using system from [7] is shown in Fig. 8 and more examples are provided in the supplementary material[1].

The scenario that is demonstrated is as follows: Objects of known geometry are placed in the workspace of a robot in a known position similar to [4]. Grasp hypotheses from a planner [40] are applied on the real robot by placing each of the 5 objects (Fig. 9) in a known position. The planner is performing object decomposition for complex objects and plans grasps on the decomposed parts [4]. In our scenario, the planner is configured for a specific preshape. To demonstrate grasping of asymmetric objects in different poses, we place them in four different orientations with respect to the robot. After a suitable grasp is generated by the planner, the hand is moved to a preshape position and the fingers are closed. After a steady state is reached (no change is detected in the tactile sensors), the stability of the grasp is estimated. Finger closing is controlled by executing a constant velocity motion for the finger joints and simultaneously limiting the maximum force by limiting the current for the finger actuators.

Before the system can be operated, a training (calibration) process, required for each individual robotic hand, needs to be completed. The calibration process is described in Algorithm 1. The algorithm is run using the objects in Fig. 9, 114 stable and 114 unstable grasps are generated, including 58 grasps from the white spray bottle and 32 grasps from the pink detergent bottle in Fig. 9. While the calibration algorithm is not tied to a particular classification methodology, in the demonstration the HMM classifier presented in Sec. III-C is shown.

Algorithm 1 Calibration mode.

1: Choose a suitable grasping strategy for object O.
2: **for** $i = 1$ to n **do**
3: Preshape the hand
4: Grasp object O according to the chosen grasping strategy.
5: Record tactile and joint configuration data during the grasp.
6: Manipulate the object O along a predetermined path.
7: Record object motion relative to the hand ΔT.
8: **if** $\Delta T > 0$ **then**
9: Grasp i is unstable.
10: **else**
11: Grasp i is stable.
12: **end if**
13: **end for**
14: Using recorded data from each grasp i, train a classifier C.

The operation mode of the demonstration system is described in Algorithm 2. A grasp is estimated as stable if the probability of a stable grasp exceeds the probability of the grasp being unstable, that is, $P(S = stable) > P(S = unstable)$. The probabilities are estimated using the well-known HMM "forward algorithm" to compute the probability of the observed sequence of measurements, assuming equal prior probabilities for stable and unstable.

Figure 10 shows snapshot images from the operation of the system[2]. The robot is attempting to grasp a bottle by first placing the hand in a preshape position given by the planner mentioned above, as shown in Fig. 10a. Then, the fingers are closed as described above. The closed grasp is shown in Fig. 10b with the corresponding tactile measurements

[1]A supplementary video showing the demonstration is available at http://ieeexplore.ieee.org.

[2]Please see the supplementary video for a more detailed demonstration.

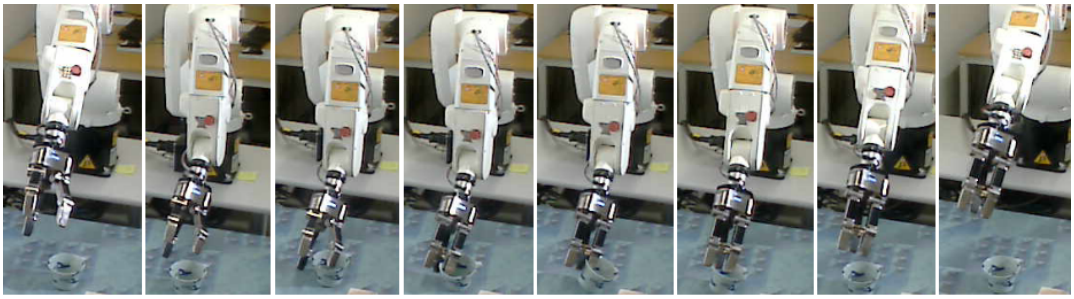


Fig. 8. An example of a failed grasp when only visual input is used. Details about the system are reported in [7].

Algorithm 2 Operation mode.

1: Generate a grasp using our grasp planner.
2: Preshape the hand.
3: Grasp object by closing fingers.
4: Evaluate classifier using sensor data.
5: **if** $P(S = stable) > P(S = unstable)$ **then**
6: Lift object.
7: **else**
8: Go to 1.
9: **end if**

in Fig. 10c. The grasp is predicted to be unstable, with the log-likelihood ratio $\log \frac{P(unstable)}{P(stable)}$ of the two models being $191.1270 > 0$, indicating unstable grasp. Now, in order to demonstrate that the failure was correctly predicted, instead of regrasping, the robot is nevertheless commanded to lift the object. The object drops as shown in Fig. 10d, demonstrating the ability to correctly recognize an unsuccessful grasp. Next, to demonstrate that the stable grasps are also successfully recognized, another grasp generated by the same grasp planner is shown in Fig. 10e. The closed grasp and the corresponding tactile measurements are shown in Figs. 10f and 10g. Based on the measurements, the grasp is predicted to be stable, with the difference across log-likelihoods of the two models being $-537.7687 < 0$, indicating a stable grasp. Lifting and rotating the object around demonstrates this in Fig. 10h, which concludes the demonstration.

Fig. 9. Objects used to generate a dataset for the demonstration.

B. Evaluation of Learning Capability

The experiments are divided according to the hierarchy presented in Section III. The goal is to evaluate the effect of the increasing knowledge on the classification results with both one-shot and temporal classification approaches.

1) Level 1: No constraints: On this level, no constraints are placed on the data used for training the classifiers. In other words, only tactile sensor measurements and the joint configuration are available and the other variables are unknown. The grasps are sampled from a sphere and the hand is oriented towards the object. The data is collected in simulation across multiple object shapes and scales.

2) Level 2: Constraints on object shape: The shape of the object is known, enabling the use of shape specific classifiers. The grasps are randomly sampled from a sphere and the hand is oriented towards the object. The data is collected in simulation.

3) Level 3: Constraints on approach vector, preshape and object shape: On level 3 of the hierarchy, constraints are placed on the approach vector, the grasp preshape and the object shape. The data are collected using a manually chosen approach vector, and the preshape is adjusted to the shape of the object. On this level, the shape is known so that shape specific classifiers can be used. Both simulated data and real data are available at this level.

C. Experimental setup

1) Data: The simulated data used in the experiments consists of five objects with three different grasp configurations applied to them. Three of the objects have primitive shape (box, cylinder, sphere), and two have natural shape (hamburger sauce, bottle). Each object is scaled to three different sizes, 0.75, 1.0, and 1.25 of the original size. For each object/size/grasp combination, 1000 unstable and 1000 stable grasps are randomly chosen from the database described in Sec. IV-B. Thus, each object/grasp dataset consists of 3000 stable and 3000 unstable grasps. When we refer to specific simulated object/grasp combination, terms *side* or *top* are used for grasps generated as side and top grasps, while *sph.* is used for grasps generated uniformly from a sphere around the object (random approach vector). Altogether, there are then 30000 samples for the five objects. We also refer to the root node of the information hierarchy, which contains all samples of primitives shapes, a total of 18000 samples.

The real data collected includes nine objects with 100 unstable and 100 stable grasps for each object. Thus, there are 1800 samples in the real data set. The details of the real data collection are described in Sec. IV-C.

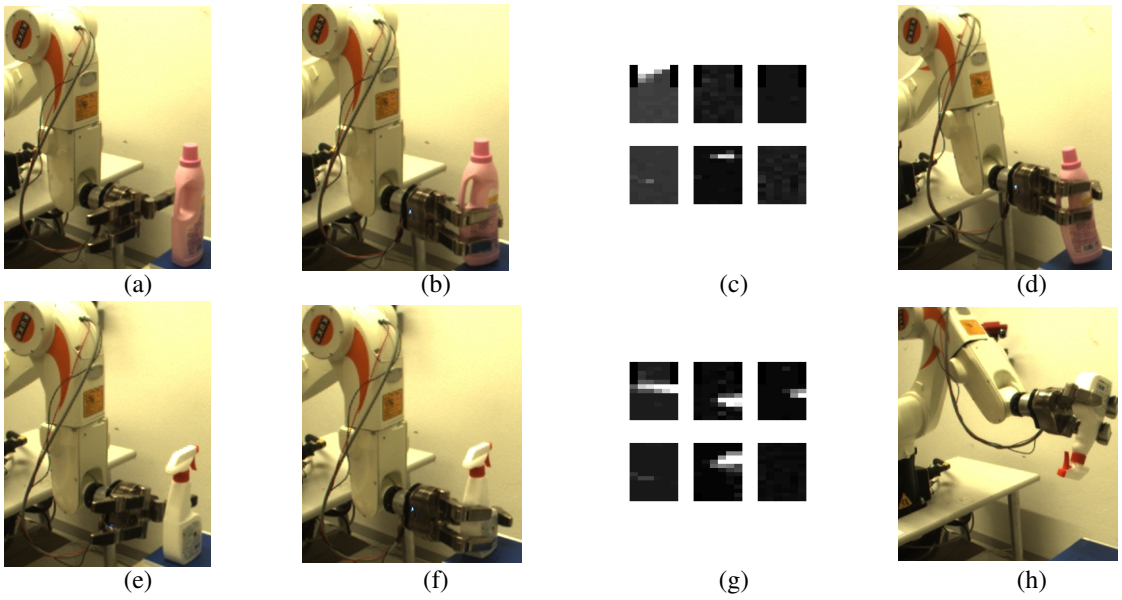


Fig. 10. Operation of the system. First row unsuccessful grasp, second row successful grasp: (a,e) Hand in a preshape position; (b,f) Closed grasp; (c,g) Tactile measurements; (d) The object dropped while lifting; (h) Lifting and rotating the object successfully.

2) One-shot recognition: As mentioned in Section III-B, we utilize the AdaBoost-algorithm in one-shot classification. Due to the formulation of the AdaBoost, a weak learner needs to be chosen. In the experiments, a decision tree with a branching factor of 1 was used as the weak learner, effectively reducing the tree to a series of linear discriminants. The branching factor was determined from series of tests that showed that using branching factor of 1 performed as good or better as larger branching factors on the data described in Sec. IV. 200 iterations of AdaBoost were run to find the final classifier in all experiments. For SVM classifier, $\gamma = 0.03$ and constant C related to the penalty applied to incorrectly classified training samples [29] is set to $C = 0.4$. Training time for both AdaBoost and SVM varies from a few minutes for simulated datasets with thousands of samples to a few seconds for real datasets which have only a few hundred samples. Classifying a single sample with the trained classifier, AdaBoost or SVM, takes a few milliseconds.

All experiments are reported as 10-fold cross validation averages, except where otherwise noted. In each case, the data sets used for training and testing the classifiers are balanced, i.e. the data sets contain equal number of unstable and stable grasps. Image moments are used as the feature representation for the one-shot classifiers. The joint data in addition to the tactile data is also included in the features unless otherwise noted.

3) Temporal recognition: To study if the temporal information improves the recognition performance, two HMMs, one for stable grasps and another for unstable ones, were trained. The stopping criteria for HMM training was a convergence threshold of 10^{-4} with a 10 iteration limit. In order to improve the reliability of the evaluation, both ergodic and left-to-right HMM were evaluated independently. The reason for these multiple experiments is that by evaluating multiple temporal models we aim to understand if the temporal ordering plays part in the modeling. The covariance of the mixture model component distributions was forced to be diagonal.

In the training of the temporal model, the structure of the HMM needs to be chosen in the form of structural parameters, which describe the number of HMM states and the number of mixture model components for each state. These were chosen experimentally such that the HMM was trained using different parameter settings and the setting producing at least lowest equal error rate result (equal number of false positives and negatives) or better performance than that was chosen. The number of states was varied between 2 and 6 while the number of mixture components was between 2 and 5.

Experiments were performed both on simulated and real data. For simulated data randomly chosen 80% of the samples were used for training and the rest 20% for testing. For the real data 10-fold cross validation was used to evaluate the performance and best parameter setting over all folds was chosen. With given parameters, the training time for the HMM varies from less than 20 minutes for the simulated data with thousands of samples to a few minutes for the real data with a few hundreds of samples. Classification of a single sample takes a few seconds.

Image moments were used as features, similar to one-shot learning. However, to reduce the number of parameters in HMM and speed up the training process, principal component analysis (PCA) was applied to the moment and joint measurements separately to reduce the dimensionality of the dataset. The number of principal components was chosen such that at least 99% of the total variance is retained.

D. One-shot recognition

In this section, we present a collection of experiments based on the information hierarchy in Fig. 1 using the AdaBoost classifier. Support vector machine classifier is used with image moments to examine the separability of the grasp stability at each level by means of log-likelihood histograms. We also study the effect of the joint configuration data on the classification by including or excluding it from the feature vector for the classifier when using real data.

1) Real data: The experiments begin by showing results using real data. Sampling grasps with a real hand is a slow process and thus the sample size is limited. To study the effect of the amount of samples used for training, we ran a series of tests with variable sample sizes. These tests are shown in Table I. The test shows that for a specific grasp on the cylindrical object, 100 samples are already enough to reach classification performance levels achieved with higher amount of samples, the differences in classification performance above 100 samples are not statistically significant. However, this is the case only when the stable and unstable grasps are distinctive, i.e. we achieve a high rate of correctly classified grasps. In the case of the white bottle data set, where the classification rate is lower, the results show that more than 200 samples could be useful in increasing the classification performance.

TABLE I

ADABOOST CLASSIFICATION RATES (IN PERCENT) ON DATA SETS WITH VARIABLE AMOUNT OF SAMPLES.

Samples	50	100	150	200
Def. cylinder	74.6 %	85.0 %	84.8 %	89.0 %
W. Bottle	64.6 %	68.0 %	68.5 %	75.5 %

Classification results for single object classifiers are presented in Table II. Classification rates are shown both with joint configuration data and without it, and the classification rates were computed for image moment feature representations. The main focus in this experiment is to study prediction of the grasp stability on objects the system has previously learnt. The average classification rate for known objects is 82.5% including joint data and 81.4% excluding it from the measurements. Thus, the inclusion of joint data seems to benefit the recognition but only to a minor effect. Moreover, the result indicates that at least with known objects the proposed approach seems to have adequate recognition rate for practical usefulness.

We also study how well the trained system can cope with unknown objects, i.e. objects that have not been used to train the system. The results are shown in Table III. The results are for a system that has been trained on all the objects except the object for which the classification rate is shown. The average recognition rate is 73.8% with joint data and 72.7% without it. The results show that while the classification rate is lower than with known objects it is still possible to make predictions of the grasp stability on unknown objects to some extent.

TABLE II

ADABOOST CLASSIFICATION RATES (IN PERCENT) ON OBJECT SETS WITH AND WITHOUT JOINT DATA.

	With joint data	Without joint data
Cylinder	88.9 %	90.3 %
Def. cylinder	91.0 %	89.0 %
Cone	79.5 %	81.0 %
O. Bottle	77.0 %	78.5 %
Shampoo	82.5 %	76.0 %
Pitcher	84.5 %	78.0 %
W. Bottle	76.0 %	73.5 %
B. Bottle	74.0 %	75.0 %
Box	89.0%	91.0 %

However, this holds true only when similar grasps are applied on unknown objects as were applied to the objects that the system were trained on. In comparison, including grasps from all objects, including the one being tested, for a single classifier yields a result of 78.6 % correct classification across all the objects in the real object set. This indicates that the variety of objects used in training plays an important role in order to attain good performance, and that the knowledge of object identity is useful but does not seem necessary if the training data includes same or similar objects.

TABLE III

ADABOOST CLASSIFICATION RATES (IN PERCENT) ON UNKNOWN OBJECTS WITH AND WITHOUT JOINT DATA.

	With joint data	Without joint data
Cylinder	80.4 %	81.9 %
Def. cylinder	76.0 %	76.5 %
Cone	73.0 %	68.0 %
O. Bottle	72.5 %	72.0 %
Shampoo	70.0 %	71.5 %
Pitcher	71.0 %	66.0 %
W. Bottle	75.0 %	76.0 %
B. Bottle	68.5 %	69.0 %
Box	78.0 %	73.0 %

Two objects of a primitive shape are included in the real data, a box and a cylinder. Table IV shows classification results when the classifier is trained only on one of the primitive objects. The classifier is then asked to classify the grasp stability of grasps made on real-world objects with different shapes. Cross validation was not needed in this case, because the training and test sets are naturally separate. The average classification rate for the cylinder model is 68.0 % and for the box model 66.4 %. These results do not anymore seem adequate for a real system, which again suggests that the variety in the training data is essential.

2) Simulated data: In contrast to the real data, in simulation we are able to sample a large number of grasps from different objects and using different grasp strategies. The following classification results were achieved using the simulated data sets described in Section IV. In Table V, the results are reported for each node in the information hierarchy. The root node (Level 1) was randomly subsampled to 12000 samples due to computational constraints and has classification rate of 75.3%. The average classification for Level 2 (known

TABLE IV

CLASSIFIER PERFORMANCE (IN PERCENT) WHEN TRAINING WITH A PRIMITIVE OBJECT.

Trained object	**Cylinder**	**Box**
Def. cylinder	76.0 %	73.5 %
Cone	66.0 %	69.5 %
O. Bottle	64.5 %	61.0 %
Shampoo	66.5 %	64.0 %
Pitcher	71.0 %	62.0 %
W. Bottle	73.5 %	69.5 %
B. Bottle	58.5 %	65.0 %

object, unknown approach vector) is 76.5% and for Level 3 (known object, known grasp) 77.5%. A trend that increasing knowledge increases classification rate appears, similar to the experiments with real data. However, the trend is significantly weaker compared to the real data. Somewhat surprisingly, the real data classification rates are notably higher when more information is available and the trend is stronger, compared to simulation.

TABLE V

ADABOOST CLASSIFICATION RATES (IN PERCENT) ACCORDING TO THE INFORMATION HIERARCHY ON SIMULATED DATA.

Level	**Node**	**Classification rate**
Level 1	Root	75.3 %
Level 2	Prim. cylinder sph.	73.5 %
	Prim. box sph.	79.2 %
	Prim. sphere sph.	77.0 %
Level 3	Prim. cylinder side	80.7 %
	Prim. cylinder top	67.6 %
	Prim. box side	83.5 %
	Prim. sphere side	78.5 %

While the primitive shapes used in Table V are simple shapes, we can use these primitive shapes to train the classifier and then use the classifier to classify grasps sampled from more natural, complex objects. The results are shown in Table VI. The table shows results of classifying the natural objects (hamburger sauce, bottle) with different training objects and grasp strategies shown in columns. Comparison results when training the classifier with the natural object and corresponding grasping strategy are shown italic font. The figures in the table show that having data from the correct object has a notable positive effect on the classification rates. This is again a positive argument for the beneficial effect of a variety of training data.

Using the SVM and its ability to output estimates of the prediction certainty, gives us a possibility to examine the performance of the classifier on different data sets in more detail compared to AdaBoost, which supports only the hard decision boundary. This comparison can be seen in Fig. 11. In the figure, log-likelihood ratios, $\log \frac{1-P(S)}{P(S)}$, calculated from the probabilities for stable and unstable samples are shown in histogram form, red for unstable and blue for stable. The classification errors are shown in filled color, with the filled area indicating the error probability. Fig. 11a-c are from simulated data and Fig. 11d is from the real cylinder grasped with the SDH hand. It is evident from the figure that increasing information makes the distributions for stable and unstable grasps more separate, which was also indicated by the earlier results. Moreover, the figure also supports the finding that classifying the real data seems to be easier than the simulated data. Finally, the figure supports the use of probabilistic approaches for grasp classification, as the ability to measure the uncertainty in classification is important as it can, for example, allow tuning the classification system to give fewer false positives.

E. Recognition based on temporal model

1) Real data: Similar to one-shot classification, we begin by investigating the general performance and the required number of samples for achieving good generalization properties. Table VII shows HMM results corresponding to Table I. The results demonstrate that the performance of HMM classifier does not change much for distinctive grasps such as the ones from the deformable cylinder. While the average classification rates are similar to the one-shot model, the temporal model seems to have better generalization capability in that the classification rate does not decrease significantly with smaller data sets.

TABLE VII

HMM CLASSIFICATION RATES (IN PERCENT) ON DATA SETS WITH VARIABLE AMOUNT OF SAMPLES.

Object	50	100	150	200
Def. cylinder	86.7 %	85.0 %	85.4 %	87.0 %
W. Bottle	78.3 %	82.0 %	74.8 %	75.0 %

Classification results for single object classifiers are presented in Table VIII both with joint configuration data (w/j) and without it (wo/j), to study the prediction capabilities on objects the system has previously learnt with the two HMM types (left-to-right: LR, ergodic: ERG). The average classification rate for known objects (with joint data) is 82.4% with LR and 81.7% with ERG which are on a par with the one-shot learning (Table II). Thus, with single object classifiers the inclusion of temporal information did not increase classification performance.

Table VIII also includes the results that study how well the trained system can cope with unknown objects, corresponding to Table III for the one-shot learning. The rates not included (marked with a dash) were below the level of chance. The results are similar in the way that the classification rates drop with unknown objects, average rate with joint data being 77.5% for LR and 77.0% for ERG. However, the rate for unknown objects is in most cases high enough such that while the classification rate is lower than with known objects, it is still possible to make useful predictions of the grasp stability on unknown objects. LR seems to outperform ERG slightly in both cases but the difference is not very significant. The reason for the difference is likely to be the simpler structure forced by the LR model, which in turn is likely to prevent

TABLE VI

ADABOOST TRAINING WITH A PRIMITIVE SHAPE AND CLASSIFYING GRASPS SAMPLED FROM A NATURAL OBJECT WITH SIMULATED DATA.

	Prim. cylinder sph.	**Prim. cylinder side**	**Prim. cylinder top**	**Prim. box sph.**	**Prim. box side**	**Prim. sphere sph.**	**Prim. sphere side**	**All classes sph.**
Hamb. sauce	71.5 %	74.0 %	62.9 %	76.8 %	73.6 %	61.4 %	62.7 %	73.4 %
	78.7 %	*83.5 %*	*72.4 %*	*78.7 %*	*82.0 %*	*78.7 %*	*83.5 %*	*78.7 %*
Bottle	68.6 %	77.4 %	56.2 %	72.6 %	76.9 %	59.4 %	66.9 %	69.7 %
	74.7 %	*82.0 %*	*65.2 %*	*74.7 %*	*82.0 %*	*74.7 %*	*82.0 %*	*74.7 %*

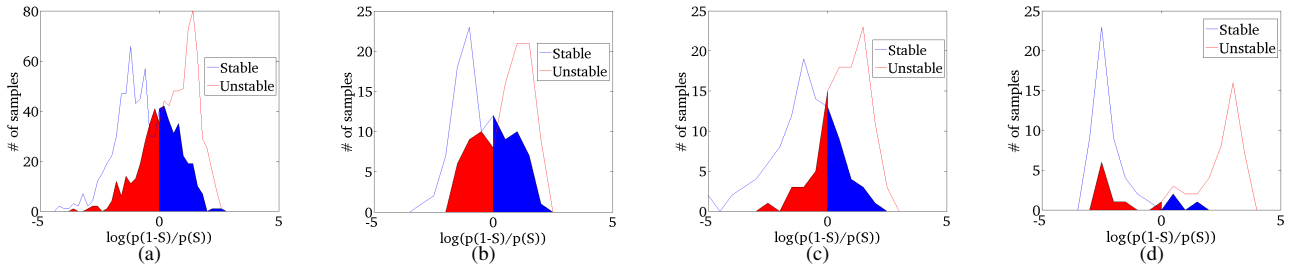


Fig. 11. Likelihood ratios for comparison of separability: (a) Root node, all objects, random grasp vector; (b) Cylinder, random grasp vector; (c) Cylinder side grasp; (d) Real cylinder side grasps.

overfitting. In comparison, using all data from all objects for a single classifier yields a result of 78.3% for LR model and 76.5% for ERG. It is remarkable that the difference between these and the results without the test object in the training data is less than 1%. Thus, with real data it seems that the generalizability of grasp stability across objects is surprisingly good.

TABLE VIII

HMM CLASSIFICATION RATES (IN PERCENT) ON KNOWN AND UNKNOWN OBJECTS.

	LR, Kn.		ERG, Kn.		LR, Unkn.		ERG, Unkn.	
	w/j	wo/j	w/j	wo/j	w/j	wo/j	w/j	wo/j
Cyl.	90.0	86.5	92.5	82.0	83.0	77.5	81.0	75.0
Def. cyl	87.0	83.5	85.0	83.0	76.0	75.5	76.0	-
Cone	83.0	80.0	81.0	85.0	77.0	73.5	76.0	69.5
O. Bott.	74.0	76.5	75.0	73.5	77.5	77.5	74.5	77.5
Shamp.	81.0	77.5	78.5	77.5	81.0	75.5	79.0	75.0
Pitcher	83.0	81.5	84.0	73.5	72.5	77.5	72.5	65.0
W. Bott.	75.0	69.0	74.0	59.5	77.0	65.0	77.5	-
B. Bott.	78.5	71.0	75.0	66.0	75.5	69.0	75.0	-
Box	90.5	67.0	90.5	68.0	78.5	79.0	81.5	-

Table IX shows classification results when the classifier is trained only on one of the primitive objects, corresponding to one-shot learning results in Table IV. The average rate for cylinder primitive is 64.6% for LR and 62.3% for ERG, which are below the results of one-shot recognition. For box primitive, the recognition rate for pitcher was below level of chance and is thus not shown. On average, the rates for box primitive are nevertheless higher than for the cylinder primitive and also higher compared to the one-shot learning. The cause of failure for the single object could not be identified. Altogether, the results are in agreement with those from one-shot learning in that the variety of training data seems important to attain good and stable performance.

TABLE IX

HMM CLASSIFICATION RATES (IN PERCENT) WHEN TRAINING WITH A PRIMITIVE OBJECT ONLY.

Node	Cylinder		Box	
	LR	ERG	LR	ERG
Def. cylinder	67.0	69.5	74.0	74.5
Cone	66.0	66.0	70.0	76.5
O. Bottle	63.0	60.0	72.0	74.5
Shampoo	61.5	57.5	75.5	77.5
Pitcher	79.5	78.5	-	-
W. Bottle	58.5	50.0	76.5	76.5
B. Bottle	57.0	55.0	73.5	74.5

2) Simulated data: Using the simulated data, Table X reports the results for each node in the information hierarchy, corresponding to Table V for the one-shot learning. For LR model, the average classification for Level 1 (root node, unknown object, unknown approach vector) is 64.9%, 69.9% for Level 2 (known object, unknown approach vector), and for Level 3 (known object, known grasp) 67.5%. The results for ERG are similar. There are two observations to be made. First, these are consistently lower than those with one-shot learning, which is the opposite behavior compared to the real data experiments, indicating that the simulated and real data do not match exactly. Second, the trend that increasing knowledge increases performance is broken for Level 3, although the difference is not very significant. A possible explanation for this is that the stability of top and side grasps is on average more difficult to model with the HMM compared to modeling the stability of a grasp with random approach vector, because it is possible that some of the grasps with random approach vector might be especially easy to recognize correctly.

TABLE XI

HMM TRAINING WITH A PRIMITIVE SHAPE AND CLASSIFYING GRASPS SAMPLED FROM A REAL-WORLD OBJECT WITH SIMULATED DATA.

	cylinder sph.		**cylinder side**		**cylinder top**		**box sph.**		**box side**		**sphere sph.**		**sphere side**		**All sph.**	
	LR	ERG	LR	ERG	LR	ERG	LR	ERG	LR	ERG	LR	ERG	LR	ERG	LR	ERG
Hamb. sauce	61.2	60.8	63.3	60.3	57.8	57.3	59.2	57.3	63.1	61.1	51.6	52.9	65.2	63.2	59.3	59.6
	60.1	*57.2*	*67.5*	*68.0*	*68.1*	*64.8*	*60.1*	*57.2*	*67.5*	*68.0*	*60.1*	*57.2*	*67.5*	*68.0*	*60.1*	*57.2*
Bottle	58.4	58.3	67.6	64.3	63.1	65.4	58.4	54.2	57.2	-	52.4	52.8	62.7	59.6	57.4	58.5
	57.8	*55.6*	*65.8*	*66.8*	*68.8*	*69.1*	*57.8*	*55.6*	*65.8*	*66.8*	*57.8*	*55.6*	*65.8*	*66.8*	*57.8*	*55.6*

TABLE X

HMM CLASSIFICATION RATES (IN PERCENT) ACCORDING TO THE INFORMATION HIERARCHY ON SIMULATED DATA.

Level	Node	LR	ERG
Level 1	Root	64.9	64.6
Level 2	Prim. cylinder sph.	70.2	70.2
	Prim. box sph.	62.1	59.0
	Prim. sphere sph.	77.4	76.9
Level 3	Prim. cylinder side	69.3	64.3
	Prim. cylinder top	69.5	69.3
	Prim. box side	68.6	69.0
	Prim. sphere side	62.8	63.2

The classification performance when training with primitive shapes but testing with real-world objects is shown in Table XI, corresponding to Table VI for the one-shot classification. The classification rates with the correct object are shown in italic for comparison. The results indicate that on average the classification is significantly improved by having the correct object model instead of a general primitive model, again indicating the importance of variety in training data. Moreover, the results are again inferior to one-shot recognition, strengthening the finding that the temporal information is not essential for recognition with the available simulated data. To conclude, the real-world cases seem to contain dynamic phenomena which can be modeled better using a temporal model.

VI. CONCLUSION AND FUTURE WORK

Uncertainty is inherent to the activities robots perform in unstructured environments. Probabilistic techniques have demonstrated the strength of coping with the uncertainty in robot planning, decision making, localization and navigation. In the area of robot grasping, there have been very few examples of solving problems such as assessing grasp stability by taking uncertainty into consideration.

In the present work, it was shown how grasp stability can be assessed based on uncertain sensory data using machine learning techniques. Our learning framework takes into account object shape, approach vector, tactile data and joint configuration of the hand. We have used a simulated environment to generate training sequences, including the simulation of the sensors. The methods were evaluated both on simulated and real data using a three-fingered robot hand. Our work demonstrates how grasp stability can be inferred using information from tactile sensors while grasping an object before the object is further manipulated or during the manipulation step. We have implemented and evaluated both one-shot and temporal learning techniques. One focus of the experiments was to study prediction capabilities of the proposed methods for known objects. We have also studied how the system can cope with unknown objects, i.e. objects that have not been used in the training step. The results show that while the classification rate is lower than with known objects it is still possible to make useful predictions of the grasp stability on unknown objects. In summary, the experimental results show that tactile measurements allow assessment of grasp stability. The aim of the paper was not a perfect discrimination between successful and unsuccessful grasps but rather a measure of certainty of grasp stability. This also means that a system may be built to reject some stable grasps while having fewer unstable grasps classified as stable ones. Experiments showed that using sequential data to evaluate grasp stability appears to be beneficial during dynamic grasp execution.

Our current work proceeds in several directions. First, we are in the process of integrating the presented system with a vision based pose estimation system and grasp planning. Second, we are implementing a grasping system based on the proposed ideas for local control of grasps and corrective movements. In both cases, the aim is to demonstrate a robust object grasping and manipulation system for both known and unknown objects based on visual and tactile sensing. Finally, we are developing a more elaborate probabilistic framework in which we study the joint probability of object-relative gripper configurations, tactile perceptions, and grasping feasibility. Here, we are developing a kernel-logistic-regression model of pose- and touch-conditional grasp success probability. The goal is to show how a learning framework can be used for grasp transfer, i.e. if the robot has learnt how to grasp one type or category of objects, to use this knowledge to grasp a new object.

REFERENCES

[1] D. Prattichizzo and J. C. Trinkle, "Grasping," in *Springer Handbook of Robotics*, B. Siciliano and O. Khatib, Eds. Springer, 2008, ch. 28.

[2] A. Saxena, J. Driemeyer, and A. Y. Ng, "Robotic grasping of novel objects using vision," *International Journal of Robotics Research*, vol. 27, no. 2, pp. 157–173, 2008.

[3] R. Detry, E. Baseski, M. Popovic, Y. Touati, N. Krueger, O. Kroemer, J. Peters, and J. Piater, "Learning continuous grasp affordances by sensorimotor exploration," in *From Motor Learning To Interaction Learning in Robots*, 1st ed., O. Sigaud and J. Peters, Eds. Berlin, Germany: Springer-Verlag, 2010.

[4] K. Huebner, K. Welke, M. Przybylski, N. Vahrenkamp, T. Asfour, D. Kragic, and R. Dillmann, "Grasping known objects with humanoid robots: A box-based approach," in *14th International Conference on Advanced Robotics*, Munich, Germany, June 2009.

[5] B. Rasolzadeh, M. Bjorkman, K. Huebner, and D. Kragic, "An active vision system for detecting, fixating and manipulating objects in real world," *International Journal of Robotics Research*, vol. 29, no. 2-3, pp. 133–154, 2010.

[6] M. Popovic, D. Kraft, L. Bodenhagen, E. Baseski, N. Pugeault, D. Kragic, T. Asfour, and N. Kruger, "A strategy for grasping unknown objects based on co-planarity and colour information," *Robotics and Autonomous Systems*, vol. 58, no. 5, pp. 551–565, 2010.

[7] J. Bohg and D. Kragic, "Learning grasping points with shape context," *Robotics and Autonomous Systems*, vol. 59, no. 4, pp. 362–377, 2010.

[8] M. Shimojo, T. Araki, A. Ming, and M. Ishikawa, "A high-speed mesh of tactile sensors fitting arbitrary surfaces," *IEEE SENSORS JOURNAL*, vol. 10, no. 4, pp. 822–830, 2010.

[9] M. Higashimori, M. Kaneko, A. Namiki, and M. Ishikawa, "Design of the 100g capturing robot based on dynamic preshaping," *International Journal of Robotics Research*, vol. 24, no. 9, pp. 743–753, 2005.

[10] H. Wakamatsu, S. Hirai, and K. Iwata, "Static analysis of deformable object grasping based on bounded force closure," in *International Conference on Robotics and Automation*, Minneapolis, USA, April 1996, pp. 3324–3329.

[11] A. T. Miller, S. Knoop, H. I. Christensen, and P. K. Allen, "Automatic Grasp Planning Using Shape Primitives," in *IEEE International Conference on Robotics and Automation*, 2003, pp. 1824–1829.

[12] C. Goldfeder, P. K. Allen, C. Lackner, and R. Pelossof, "Grasp Planning Via Decomposition Trees," in *IEEE International Conference on Robotics and Automation*, 2007, pp. 4679–4684.

[13] A. M. Howard and G. A. Bekey, "Intelligent learning for deformable object manipulation," *Autonomous Robots*, vol. 9, no. 1, pp. 51–58, 2000.

[14] A. Morales, M. Prats, P. Sanz, and A. P. Pobil, "An experiment in the use of manipulation primitives and tactile perception for reactive grasping," in *Robotics: Science and Systems, Workshop on Robot Manipulation: Sensing and Adapting to the Real World*, Atlanta, USA, 2007.

[15] M. Prats, P. Sanz, and A. del Pobil, "Vision-tactile-force integration and robot physical interaction," in *IEEE International Conference on Robotics and Automation*, Kobe, Japan, 2009, pp. 3975–3980.

[16] S. Ekvall and D. Kragic, "Learning and Evaluation of the Approach Vector for Automatic Grasp Generation and Planning," in *IEEE International Conference on Robotics and Automation*, 2007, pp. 4715–4720.

[17] A. Petrovskaya, O. Khatib, S. Thrun, and A. Y. Ng, "Bayesian estimation for autonomous object manipulation based on tactile sensors," in *IEEE International Conference on Robotics and Automation*, Orlando, FL, USA, May 2006, pp. 707–714.

[18] A. Jiméneza, A. Soembagijob, D. Reynaertsb, H. V. Brusselb, R. Ceresa, and J. Ponsa., "Featureless classification of tactile contacts in a gripper using neural networks," *Sensors and Actuators A: Physical*, vol. 62, no. 1–3, pp. 488–491, 1997.

[19] S. Chitta, M. Piccoli, and J. Sturm, "Tactile object class and internal state recognition for mobile manipulation," in *IEEE International Conference on Robotics and Automation*, Anchorage, AK, USA, May 2010, pp. 2342–2348.

[20] M. Schöpfer, M. Pardowitz, and H. J. Ritter, "Using entropy for dimension reduction of tactile data," in *14th International Conference on Advanced Robotics*, Munich, Germany, June 2009.

[21] A. Schneider, J. Sturm, C. Stachniss, M. Reisert, H. Burkhardt, and W. Burgard, "Object identification with tactile sensors using bag-of-features," in *IEEE/RSJ international conference on Intelligent robots and systems*, St. Louis, MO, USA, 2009, pp. 243–248.

[22] N. Gorges, S. E. Navarro, D. Göger, and H. Wörn, "Haptic object recognition using passive joints and haptic key features," in *IEEE International Conference on Robotics and Automation*, Anchorage, AK, USA, May 2010, pp. 2349–2355.

[23] K. Hsiao, L. Kaelbling, and T. Lozano-Perez, "Task-driven tactile exploration," in *Robotics: Science and Systems*, Zaragoza, Spain, June 2010.

[24] Y. Bekiroglu, J. Laaksonen, J. A. Jorgensen, and V. Kyrki, "Learning grasp stability based on haptic data," in *Robotics: Science and Systems Workshop on Representations for Object Grasping and Manipulation in Single and Dual Arm Tasks*, Zaragoza, Spain, June 2010.

[25] "Weiss robotics tactile sensor." [Online]. Available: http://www.weiss-robotics.de/en.html

[26] J. Laaksonen, V. Kyrki, and D. Kragic, "Evaluation of feature representation and machine learning methods in grasp stability learning," in *10th IEEE-RAS International Conference on Humanoid Robots*, 2010, pp. 112–117.

[27] Y. Freund and R. E. Shapire, "Experiments with a new boosting algorithm," in *Thirteenth Internation Conference on Machine Learning*. Morgan Kaufmann, 1996, pp. 148–156.

[28] A. Vezhnevets, *GML AdaBoost Matlab Toolbox*, 2006, available at http://graphics.cs.msu.ru/ru/science/research/machinelearning/ adaboost-toolbox.

[29] C. Cortes and V. Vapnik, "Support vector networks," *Machine Learning*, vol. 20, pp. 273–297, 1995.

[30] V. Vapnik, *The Nature of Statistical Learning Theory*. New York: Springer-Verlag, 1995.

[31] C.-C. Chang and C.-J. Lin, *LIBSVM: a library for support vector machines*, 2001, software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.

[32] J. C. Platt, "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods," in *Advances in Large Margin Classifiers*. MIT Press, 1999, pp. 61–74.

[33] L. R. Rabiner, "A tutorial on Hidden Markov Models and selected applications in speech recognition," in *Proceedings of the IEEE*, vol. 77, no. 2, 1989, pp. 257–286.

[34] J. Jorgensen, L. Ellekilde, and H. Petersen, "RobWorkSim - an open simulator for sensor based grasping," in *41st International Symposium on Robotics (ISR 2010) and ROBOTIK 2010*, Munich, Germany, June 2010.

[35] A. T. Miller and P. K. Allen, "Graspit! A Versatile Simulator for Robotic Grasping," *IEEE Robotics and Automation Magazine*, vol. 11, no. 4, pp. 110–122, 2004.

[36] J. A. Jorgensen and H. G. Petersen, "Usage of simulations to plan stable grasping of unknown objects with a 3-fingered schunk hand," in *IROS'08 Workshop on Robot Simulators*, Nice, France, September 2008.

[37] C. Ferrari and J. Canny, "Planning optimal grasps," in *IEEE International Conference on Robotics and Automation*, Nice, France, May 1992, pp. 2290–2295.

[38] R. Suárez, M. Roa, and J. Cornella, "Grasp quality measures," Technical University of Catalonia, Tech. Rep. IOC-DT-P-2006-10, 2006.

[39] J. Bohg and D. Kragic, "Grasping familiar objects using shape context," in *14th International Conference on Advanced Robotics*, Munich, Germany, June 2009.

[40] K. Huebner, "BADGr - A Toolbox for Box-based Approximation, Decomposition and GRasping," in *IROS 2010 Workshop on Grasp Planning and Task Learning by Imitation*, Taipei, Taiwan, October 2010.

Robust robot-camera calibration

J. Ilonen and V.Kyrki

Abstract— **Calibrating parameters of a vision system for robotics is crucial for many tasks where the robot has to interact with the environment. This paper introduces a robust method for calibrating the relative poses between the base frame of the robot and one or more cameras. The method is based on tracking a marker attached to the end-effector of the robot without requiring manual boostrapping. The method is robust to a large number of outliers (wrongly detected marker positions) and can provide covariance of the of the estimated parameters based on the variance of the observed errors, providing information on the accuracy of the estimate. The steps of the calibration procedure are presented comprehensively.**

I. IINTRODUCTION

Calibrating parameters of a vision system for robotics is crucial for many tasks where the robot has to interact with the environment. In this paper a robust method for calibrating the relative poses between the base frame of the robot and one or more cameras is introduced. The method is based on tracking a marker rigidly attached to the end-effector of the robot. The position of the marker relative to the end-effector is estimated, but the forward kinematics of the robotics must be known. No manual boostrapping is required, the robot can be moved to random joint positions and if marker is visible to the camera the data is stored. When some amount of data is available, the parameters (pose of the camera(s) and the marker position) can be estimated. The estimation can provide information on the accuracy of the estimate by backpropagating the variance of the residual errors. The estimation method is based on robust M-estimators and therefore some outliers, wrongly detected marker positions, do not affect the accuracy of the estimate.

The method is related to the (stereo) camera calibration method by Zhang [17] (additional details in [16]), where a planar calibration target is viewed from several viewpoints and the method then solves the intrinsic camera parameters and relative poses of the calibration objects to the camera. It provides a (non-robust) maximum likelihood estimate for the parameters using Levenberg-Marquadt algorithm. Zhang's method requires that the markers lie on a plane while in our case the requirement is that their relative poses are known from forward kinematics. Rotation matrix is parametrized using axis-angle presentation in both methods. In Zhang's method the main interest is in calibrating the intrinsic camera parameters and the pose of the calibration plane comes along for free. In our method the main interest is estimating the pose between camera and the robot, which holds the calibration target which is moved to different positions. In this article it is assumed that the intrinsic parameters of the camera are already known, but their estimation could also be easily added. Intrinsic camera parameters do not have to be estimated all that often compared to extrinsic and therefore the steps have been kept as separate.

The authors are with Machine Vision and Pattern Recognition research group, Lappeenranta University of Technology, Skinnarilankatu 34, 53850 Lappeenranta, Finland. {ilonen,kyrki}@lut.fi

Contributions and design goals of the method are:

- Simple and accurate method for robot-camera calibration (the relative pose between the base frame of the robot and one or more cameras).
- Robust to noise in detected marker positions and to complete outliers.
- Provide information on accuracy of the estimate.
- Reasonably quick, both in the sense of not requiring much data and fast computation.

Section II covers shortly the related work, Section III is the overview and details of the presented method, Section IV presents the experimental setup and results and Section V the conclusions.

II. RELATED WORK

The problem in our case is formulated similarly to extrinsic camera calibration using a known 3D object. Zhang's method [17] has been extended for 3D calibration objects for example in [11] and for moving 1D objects in multi-camera setups in [14]. There also numerous other calibration methods based on 3D calibration objects, for example [4], [5] where unit quaternions are used for handling rotations and the robustness (to noise in marker positions) of the method is of special interest. However, the extrinsic camera calibration methods are not directly applicable to our case because if a separate calibration object is attached to the robot, there is still the problem of finding out its position relative to the robot, or the required shape of the calibration object (which in our is equal to movement of the robot's hand and the attached marker) would be difficult to realize.

In the field of robotics there are many studies on hand/eye calibration, where the location of the camera mounted on the end-effector has to be determined. For example in the method by Tsai and Lenz [13] few images of a planar calibration object are taken and the hand/eye calibration can be then performed. Later, even automatic calibration methods have been designed [10].

The problem of calibrating the coordinate frames between the robot and the camera is often described only in scarce detail as the main problem presented in an article lies elsewhere. Here is a short review how the calibration is performed in two laboratories where the ARMAR III robotic head [1] is used, Karlsruhe Instutute of Technology (KIT) and KTH Royal Institute of Technology. The ARMAR III

robotic head has 7 degrees of freedom and has two "eyes" which both have wide and narrow field of view cameras which can focus on an object (foveation). Calibration of such system requires much more than calibration between robot (or world) and camera frames, but that is still a crucial part if a robotic hand has to interact with the world the head sees.

In KIT the ARMAR III calibration procedure [15] is based on a checkerboard pattern which defines the world frame, i.e., the location of the base frame of the robot in the world frame is not considered explicitly. The calibration starts from Zhang's method [17] and is mostly concentrating the kinematic calibration; how the relation between the camera and world frames change when the head or eyes move.

In KTH the calibration of ARMAR III [6] is performed using a LED attached to the robot's end effector which is then moved in a pattern. Combined stereo and robot-camera calibration is performed at the same time and the method is based to that of Zhang [17]; the LED is moved in regular planar patterns instead of using a actual planar checkboard. In comparison to this article, the method requires that the pattern in which the LED is moved is selected manually so that the camera sees all of the pattern, and the position of the LED is measured beforehand and not estimated.

III. Overview of the Method

The robot-camera calibration is based on a marker attached rigidly to the end-effector of the robot. The position of the marker in the end-effector frame and the pose between the camera and the robot base frame are estimated during calibration. The overview of the estimated parameters is presented in Fig. 1. When more than one camera is calibrated at the same time they all have separate poses, but share the same marker position.

Intrinsic parameters of the cameras are assumed to be calibrated separately, because effective methods already exist [8], [17] and the number of measurements would increase needlessly as the intrinsic parameters (in fixed focus cameras) do not usually change. The intrinsic parameters have been estimated in this work with Matlab Camera Calibration Toolbox [3]. The method can apply both radial and tangential distortion parameters of the camera.

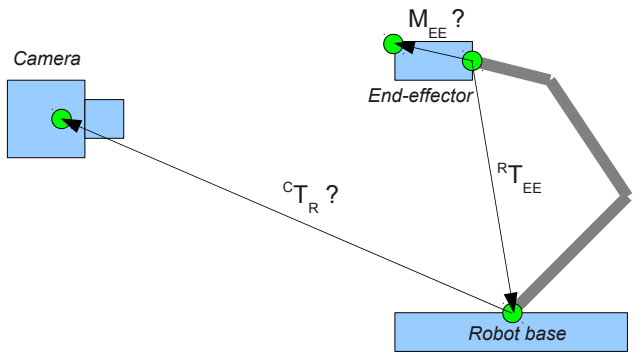


Fig. 1. Overview of the setup and what is being estimated.

In the following all necessary steps for calibration are described and in the end of this section the steps of the calibration procedure are enumerated.

A. From end-effector to camera frame

A doint rotation-translation matrix is composed as

$$T = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \tag{1}$$

where R is a 3×3 rotation matrix and translation vector $t = [t_x, t_y, t_z]^T$.

It is assumed that the joint rotation-translation matrix between the end-effector and the robot base, ${}^R T_{EE}$, is known from forward kinematics and can be computed for the current measurement from for example joint values.

In the camera frame the XYZ position of the marker M_{EE} is

$$M_C = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = {}^C T_R * {}^R T_{EE} * M_{EE}, \tag{2}$$

where ${}^R T_{EE}$ is known from the forward kinematics and M_{EE} and ${}^C T_R$ are the unknown parameters to be determined. The marker position $M_{EE} = [m_x, m_y, m_z, 1]^T$ is the position of the marker in the EE frame in homogenous coordinates. There are several possible parametrizations for rotation matrix, but the most natural for our case is the axis-angle presentation where vector $\theta = [\theta_0, \theta_1, \theta_2]^T$ defines the axis of rotation and the length of vector $|\theta|$ defines the rotation around the specified axis. The benefit of this formulation is that the partial derivatives, which are needed for estimation, have only one singular point (zero length vector) unlike Euler angles, and there are no extra constraints unlike with quaternions where the are four parameters presenting three degrees of freedom.

The rotation matrix from vector θ is defined as [7]

$$R(\theta) = cos|\theta| I + \frac{sin|\theta|}{|\theta|}[\theta]_x + \frac{1 - cos|\theta|}{|\theta|^2}\theta\theta^T, \tag{3}$$

where I is the identity matrix, $[\theta]_x$ is skew-symmetric matrix

$$[\theta]_x = \begin{bmatrix} 0 & -\theta_2 & \theta_1 \\ \theta_2 & 0 & -\theta_0 \\ -\theta_1 & \theta_0 & 0 \end{bmatrix}. \tag{4}$$

and

$$\theta\theta^T = \begin{bmatrix} \theta_0^2 & \theta_0\theta_1 & \theta_0\theta_2 \\ \theta_0\theta_1 & \theta_1^2 & \theta_1\theta_2 \\ \theta_0\theta_2 & \theta_1\theta_2 & \theta_2^2 \end{bmatrix}. \tag{5}$$

In case of one camera and one marker the parameters to be estimated are

$$X = \{\theta_0, \theta_1, \theta_2, t_x, t_y, t_z, M_{EE_x}, M_{EE_y}, M_{EE_z}\}. \tag{6}$$

B. From camera frame to pixel position

To get from a 3D point in the camera frame to the actual camera pixel position a pinhole camera model with radial and tangential distortion is used [8].

The basic pinhole camera can be defined as

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{7}$$

where f_x, f_y is the focal length of the lens in x and y directions and c_x, c_y is the principal point (center of the image) and the actual pixel position is $(u/w, v/w)$. Distortion parameters are (k_1, k_2, k_3) for the radial distortion and (p_1, p_2) for the tangential distortion. The pixel position (u, v) can then be calculated as follows:

$$\begin{aligned} \begin{bmatrix} \dot{a} \\ \dot{b} \end{bmatrix} &= \begin{bmatrix} a \\ b \end{bmatrix} (1 + k_1 r_2 + k_2 r_2^2 + k_3 r_2^3) + \\ &\quad \begin{bmatrix} 2p_1 ab + p_2(r_2 + 2a^2) \\ p_1(r_2 + 2b^2) + p_2 ab \end{bmatrix} \\ u &= f_x \dot{a} + c_x \\ v &= f_y \dot{b} + c_y, \end{aligned} \tag{8}$$

where $a = x/z$, $b = y/z$ and $r_2 = a^2 + b^2$.

C. Partial derivatives

For maximum-likelihood estimation of the parameters (position of the marker M_{EE} and joint rotation/translation matrix from robot to camera frame ${}^C T_R$) partial derivatives of Eq. 2 with respect to each parameter are needed. Applying the product rule of derivation it can be seen that partial derivatives of M_{EE} and ${}^C T_R$ can be solved separately and the joint rotation-translation matrix ${}^C T_R$ can be divided to its rotation and translation parts.

The partial derivative of a rotation matrix defined by axis-angle vector θ (Eq. 3) is

$$\begin{aligned} \frac{\partial R(\theta)}{\partial \theta_i} &= -\theta_i \frac{\sin|\theta|}{|\theta|} I + \\ &\quad \theta_i \left(\frac{\cos|\theta|}{|\theta|^2} - \frac{\sin|\theta|}{|\theta|^3} \right) [\theta]_x + \\ &\quad \frac{\sin|\theta|}{|\theta|} \frac{\partial [\theta]_x}{\partial \theta_i} + \\ &\quad \theta_i \left(\frac{\sin|\theta|}{|\theta|^3} + 2\frac{\cos|\theta| - 1}{|\theta|^4} \right) \theta\theta^T + \\ &\quad \frac{1 - \cos|\theta|}{|\theta|^2} \frac{\partial(\theta\theta^T)}{\partial \theta_i} \end{aligned} \tag{9}$$

where the partial derivatives of $\frac{\partial [\theta]_x}{\partial \theta_i}$ and $\frac{\partial(\theta\theta^T)}{\partial \theta_i}$ can be trivially solved from Eq. 4 and 5. Partial derivatives of the translation vector and the position of the marker are also trivial, for example, $\frac{\partial t}{\partial t_x} = [1, 0, 0]^T$. The complete partial derivatives of Eq. 2 are

$$\begin{aligned} \frac{\partial M_C}{\partial \theta_i} &= \begin{bmatrix} \frac{\partial R(\theta)}{\partial \theta_i} & 0 \\ 0 & 0 \end{bmatrix} *^R T_{EE} * M_{EE} \\ \frac{\partial M_C}{\partial t_{x,y,z}} &= \begin{bmatrix} 0 & \frac{\partial t}{\partial t_{x,y,z}} \\ 0 & 0 \end{bmatrix} *^R T_{EE} * M_{EE} \\ \frac{\partial M_C}{\partial M_{EE_{x,y,z}}} &= {}^C T_R *^R T_{EE} * \frac{\partial M_{EE}}{\partial M_{EE_{x,y,z}}} \end{aligned} \tag{10}$$

One thing to note is that a partial derivative of a rotation matrix is not a proper rotation matrix, e.g., the sum of rows/columns is not necessarily 1, and the results in general are not in homogenous coordinates as the last value of the resulting vector is zero.

To estimate the parameters with the maximum-likelihood estimator the derivatives in the camera coordinates are also needed to calculate to which direction the marker would move in camera pixel coordinates if a of the model parameter is adjusted. For this derivatives of Eq. 8 are needed, i.e, $\frac{\partial u}{\partial X}$ where X is one of the model parameters. As the equations are the same for all parameters a simplified notation is used, where $\frac{\partial u}{\partial X} = u'$. The derivatives can be calculated from Eq. 8 and

$$\begin{aligned} \begin{bmatrix} \dot{a}' \\ \dot{b}' \end{bmatrix} &= \begin{bmatrix} a' \\ b' \end{bmatrix} (1 + k_1 r_2 + k_2 r_2^2 + k_3 r_2^3) + \\ &\quad \begin{bmatrix} a \\ b \end{bmatrix} (k_1 r_2' + 2k_2 r_2 r_2' + 3k_3 r_2^2 r_2') + \\ &\quad \begin{bmatrix} 2p_1(a'b + ab') + p_2(r_2' + 4aa') \\ p_1(r_2' + 4bb') + 2p_2(a'b + ab') \end{bmatrix} \\ u' &= f_x \dot{a}' \\ v' &= f_y \dot{b}', \end{aligned} \tag{11}$$

where $a' = \frac{zx' - xz'}{z^2}$, $b' = \frac{zy' - yz'}{z^2}$ and $r_2' = \frac{z^2(2xx' + 2yy') - 2zz'(x^2 + y^2)}{z^4}$.

D. Weighted Gauss-Newton approximation and M-estimators

The problem is formulated as a bundle adjustment [12] problem using a non-quadratic M-estimator which can explicitly handle outliers, unlike more classical maximum-likelihood or least-squares formulation. In this case when using a single camera and a single marker the problem is not sparse, but when several cameras (non-calibrated stereo system, for example) is estimated the problem becomes sparse because pixel position of the marker in a camera is not dependent of other cameras, they only share the same marker position, M_{EE}.

A single iteration of weighted Gauss-Newton approximation consists of solving Δ in equation

$$(J^T W J)\Delta = -J^T W \epsilon \tag{12}$$

where J is the Jacobian, W is the weight matrix and ϵ are the residuals (prediction error), $\epsilon = \overline{z} - z(X)$, where $\overline{z}$ are the measured and $z(X)$ the predicted values. For least-squares estimate the weight matrix is the identity matrix,

other choices for weight are discussed later. The estimated parameters are then adjusted,

$$X_{i+1} = X_i + \alpha\Delta, \tag{13}$$

where $\alpha =]0, 2]$ is selected so that the weighted residual is minimized [12],

$$\arg\min_{\alpha\in]0,2]} W(\overline{z} - z(X + \alpha\Delta)). \tag{14}$$

Using the assumption $\alpha = 1$ often leads to non-optimal improvement (i.e., more iterations are needed) and computing just the residuals instead of the full Jacobian and solving new Δ is considerably faster. Iterations are continued until no improvement is found or a limit on number of iterations is reached.

Two M-estimators have been used to increase robustness to outliers. Initial estimation is started with $L_1 - L_2$ M-estimator and after convergence more outlier-resistant Welsch M-estimator is used. When used with Gauss-Newton approximation the M-estimator defines the diagonal elements in the weight matrix W [18]. In case of least-squares, L_2, estimation $w(x) = 1$, meaning that no weighting is used, but the estimate is not robust to outliers. With L_1 norm $w(x) = \frac{1}{|x|}$ which means that the weight decreases with increasing residual, however the weight goes to infinity when x approaches zero. Therefore, $L_1 - L_2$ M-estimator is initially used, because like L_1 estimator the influence of large errors is reduced and the function is defined everywhere like L_2. The weight function of $L_1 - L_2$ M-estimator is

$$w(x) = \frac{1}{\sqrt{1 + x^2/2}}. \tag{15}$$

When initial converge has been reached, the M-estimator is changed to Welsch function [18],

$$w(x) = e^{-\left(\frac{x}{c}\right)^2}, \tag{16}$$

where with suitably chosen value of c the weight for large outliers approaches zero. Few examples of weight functions of $L_1 - L_2$ and Welsch M-estimators are presented inf Fig. 2. The value of $c = 2.9848$ for Welsch-function has been selected as one of the presented graphs because then it reaches 95% asymptotic efficiency with standard normal distribution [18].

E. Backward propagation of covariance

In general given a non-linear function $f : \mathbb{R}^M \rightarrow \mathbb{R}^N$ and v a random vector in $\mathbb{R}^M$, the approximation of mean and covariance of $f(v)$ can be computed in the vicinity of the mean $\overline{v}$ of the distribution. The approximation of f is $f(v) \approx f(\overline{v}) + J_f(v - \overline{v})$, where J_f is the Jacobian $\frac{\partial f}{\partial v}$ evaluated at $\overline{v}$. The first-order approximation of random variable $f(v)$ has mean $f(\overline{v})$ and covariance $\Sigma_f = J_f \Sigma J_f^T$. In our case we can calculate or have a reasonable assumption of covariance of $f(v)$ (the pixel positions given estimated parameters) and would like to propagate the covariance backwards. One option would be to create inverse mapping function f^{-1} to map pixel positions to parameters and their partial derivatives $J_{f^{-1}}$, but fortunately that is not needed, because the inverse covariance propagation can be calculated as [2], [7]

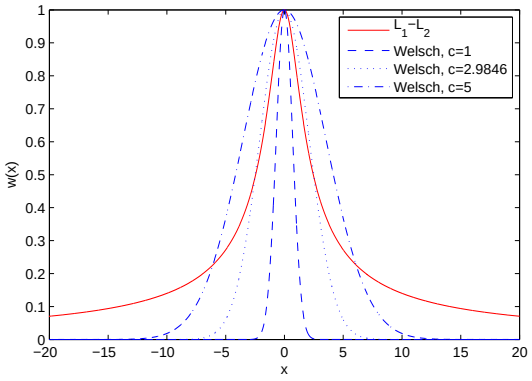


Fig. 2. Weight functions of $L_1 - L_2$ and Welsch M-estimators.

$$\Sigma_{f^{-1}} = \left(J_f^T \Sigma_f^{-1} J_f\right)^{-1}. \tag{17}$$

F. The complete calibration procedure

1) Collect data; move the robot to (random) positions, store the marker position $\overline{z_i}$ and ${}^R T^i_{EE}$.
2) Start the parameter estimation, initialize the parameters X.
3) For each measurement i, calculate the predicted measurements $z_i = z(X, {}^R T^i_{EE})$ (Eq. 2 and Eq. 8), residuals $\epsilon_i = \overline{z_i} - z_i$ and partial derivatives J_i (Eq. 10 and Eq. 11).
4) Calculate M-estimator weights W based on the residuals (Eq. 15 or Eq. 16).
5) Do one iteration of Gauss-Newton estimation (Eq. 12 and Eq. 14).
6) If an improved solution was found, apply Eq. 13 and go back to step 2, otherwise stop.
7) Parameters estimated, check backpropagated variances if needed (Eq. 17).

IV. Experiments

A. Setup

The experiments have been performed using Mitsubishi RV3SB industrial robot with an attached Schunk PG 70 gripper and Bumblebee 2 stereo camera with 640×480 resolution. The marker has been a blinking red LED which has been gripped in arbitrary pose by the gripper.

The stereo camera has been calibrated using camera calibration toolbox for matlab [3]. The calibration includes intrinsic parameters of both cameras and their relative pose. The camera-robot calibration method presented here requires that the intrinsic parameters are known and can apply the information of respective poses of cameras in calibrated stereo by modifying the Eq. 2 to include the joint rotation-translation between left and right cameras, ${}^{C_r} T_{C_l}$, assuming

the pose between robot and the left camera is being calibrated. Eq. 2 then becomes two separate equations for left and right cameras, but the number of estimated parameters (6 for ${}^{C_l}T_R$ and 3 for M_{EE}) stays the same as in one camera case,

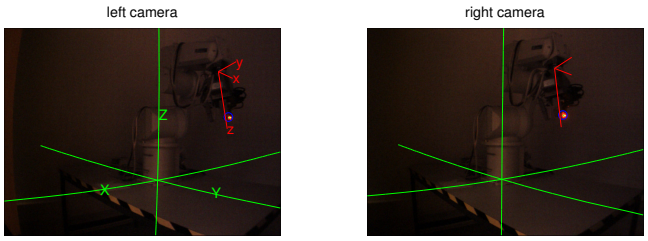
$$M_{C_l} = {}^{C_l}T_R *^R T_{EE} * M_{EE}$$
$$M_{C_r} = {}^{C_r}T_{C_l} *^{C_l} T_R *^R T_{EE} * M_{EE}. \quad (18)$$

Some tests have been performed so that the two cameras of the calibrated stereo have been treated as separate and both ${}^{C_l}T_R$ and ${}^{C_r}T_R$ have been estimated, which means that there are now 15 parameters to estimate in total as M_{EE} is still shared,

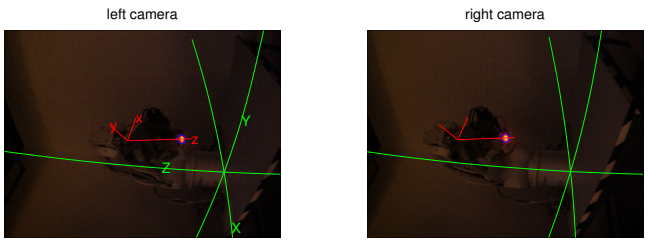
$$M_{C_l} = {}^{C_l}T_R *^R T_{EE} * M_{EE}$$
$$M_{C_r} = {}^{C_r}T_R *^R T_{EE} * M_{EE}. \quad (19)$$

This way, we have a "groundtruth" pose between two cameras measured with the camera calibration toolbox and we can compare the result between two different calibration methods.

Tests were repeated with the camera and the marker LED in several different locations. In the tests the robot arm was set in random joint positions until 50 such poses were found where both cameras found the position of the blinking LED in the same frame. Each test set includes 50 stereo image pairs with the marker location found with sub-pixel accuracy, however, in some of the images instead of the true LED marker location an erroneous reflection from a metallic part of the robot or surrounding environment has been found instead. Examples of two test setups can be seen in Fig. 3.



(a) From set 'Static camera 1'



(b) From set 'Static LED 4'

Fig. 3. Examples of test setups. Green lines and capital letters X, Y and Z mark the axes of the robot base frame, red lines and small letters mark the axes of the end-effector frame and a blue circle marking the estimated position of the marker LED.

The parameter estimation was initialized so that the origin of the robot base frame was assumed to be one meter directly in front of the camera, the marker LED 0.2m directly in front of the robot's end effector frame and the three axis angle parameters were initialized randomly. This initialization strategy was used because there is a reasonable assumption for the two translations but the rotation matrix can be almost anything depending on which side of the robot the camera is.

Unless otherwise noted, the tests have been run initially using the $L_1 - L_2$ M-estimator and after convergence the M-estimator have been changed to Welsch wich $c = 5.0$ to remove the effect of outliers. The value of $c = 5.0$ means that the weight of a measurement approaches zero for residuals larger than 10 pixels (see Fig. 2) and the value should in general be based on the resolution of the camera, accuracy of the forward kinematics of the robot and how accurately the marker can be detected. Note that the residuals have been calculated as euclidean distance between the detected and estimated marker positions of the marker, not separately for x and y coordinates.

The speed of the parameter estimation was not of special interest, but the C implementation estimates parameters for 50 measurements in under one second.

B. Gaussianity of errors

Propagating covariance requires that the errors are distributed roughly according to Gaussian distribution. Therefore, the distribution of residuals was the point of interest in this experiment. In the experiment non-robust least-squares estimation was used with a test-set where there were very few outliers. In addition the potential bias caused by estimating only the position of the left camera and using the pre-calibrated pose between cameras (Eq. 18) and estimating both cameras separately (Eq. 19) was studied.

The results are presented in Fig. 4. Fig. 4(a)&(c) present results when estimating only one camera and Fig. 4(b)&(d) the results when estimating cameras separately. In both cases the residuals were roughly normally distributed, however, as is reflected by the kurtosis values (≈ 20) the distributions have sharper peaks and fatter tails [9].

There was a slight bias between the right and left cameras (Fig. 4(a), means marked with bold 'x' and '+'), the distance between means was 0.194 pixels. The bias is mainly caused by the fact that the marker LED is not truly a point and it is seen from slightly different points of view by the cameras, in the view of the right camera the marker is always left of the position compared to what the left camera sees. In the case where both cameras were estimated separately (Fig. 4(b)) there was no bias.

C. Estimation accuracy

Eight test sets in total were collected where each has 50 stereo image pairs. In four of them the camera was kept in the same position and the marker LED was attached to a different position in the gripper. The results for these four test are presented in Table I. The $\pm$ values report the inaccuracy

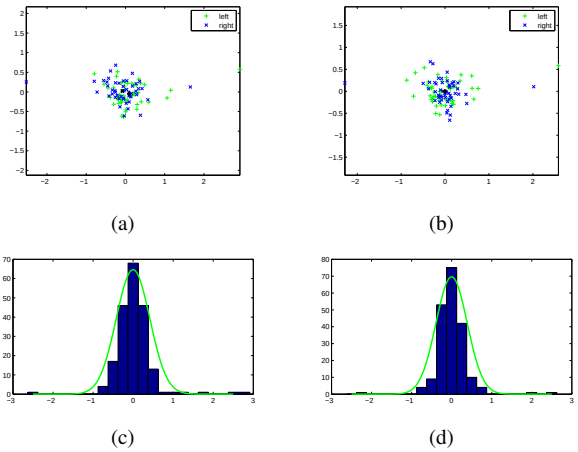


Fig. 4. Distribution of residuals; (a) residuals when estimating one camera; (b) residuals when estimating cameras separately; (c)&(d) histograms of residuals for both cases, green line presents the Gaussian distribution fit to the data.

calculated from backpropagated variance of residuals. With 50 stereo image pairs there is maximum of 100 inliers. With the camera staying stationary in best case $t_{x,y,z}$ would be identical in every test. Angle* is the average difference between rotation matrices to the other sets. The differences in the coordinates were a few millimetres and also the angular differences between the test sets were small, about $0.4°$. One thing to note is the large number of outliers in Set 3, which was caused by the LED being positioned so that its reflection was being detected often instead of the actual LED.

TABLE I

RESULTS WITH FOUR TEST SETS WHERE THE CAMERA HAS STAYED STATIONARY AND THE MARKER LED HAS BEEN CHANGED TO DIFFERENT POSITIONS. THE UNITS ARE MILLIMETRES.

	Set 1	Set 2	Set 3	Set 4
inliers	96	90	56	93
angle*	$0.444°$	$0.444°$	$0.508°$	$0.398°$
t_x	104.1 ± 1.2	100.3 ± 0.8	102.2 ± 1.2	103.9 ± 0.9
t_y	275.8 ± 1.6	283.8 ± 0.9	280.5 ± 1.4	281.9 ± 1.2
t_z	1379.4 ± 2.0	1373.3 ± 1.3	1380.8 ± 2.9	1377.5 ± 1.6
M_{EE_x}	5.8 ± 0.5	-13.0 ± 0.5	-5.5 ± 0.8	-0.3 ± 0.5
M_{EE_y}	8.8 ± 0.4	-29.5 ± 0.4	-9.5 ± 0.7	0.0 ± 0.4
M_{EE_z}	204.5 ± 1.1	215.4 ± 0.7	170.4 ± 1.0	214.7 ± 0.8

In four of the tests the marker LED was kept in the same position and the camera was moved to other positions. The results are presented in Table II. In this experiment the marker LED positions $M_{EE_{x,y,z}}$ would be equal in the best case. In x and y directions the changes were very small, in the order of $0.5mm$, but in the z direction the changes were slightly larger in one of the test sets, Set 2, the difference was about $3mm$.

The results comparing estimation of joint rotation-translation betweeen ${}^{C_r}T_{C_l}$ cameras in a stereo system are presented in Table III. The translation between cameras estimated by the Camera Calibration Toolbox for Matlab [3] was $(120.5, -0.5, 0.8)$mm with $0.30°$ difference between directions of the cameras (which are designed to be parallel). Average trasnlation with the eight test sets estimated with

TABLE II

RESULTS WITH FOUR TEST SETS WHERE THE CAMERA HAS BEEN MOVED AND THE MARKER LED HAS BEEN KEPT IN THE SAME POSITION. THE UNITS ARE MILLIMETRES.

	Set 1	Set 2	Set 3	Set 4
inliers	96	100	93	96
t_x	-235.2 ± 0.7	-57.6 ± 0.7	-110.7 ± 0.6	529.2 ± 0.6
t_y	282.8 ± 0.8	258.2 ± 0.8	405.8 ± 0.6	216.8 ± 0.8
t_z	905.5 ± 1.2	1950.4 ± 2.7	1309.6 ± 1.2	1356.7 ± 1.4
M_{EE_x}	-0.3 ± 0.5	-0.3 ± 0.6	-0.9 ± 0.2	-0.7 ± 0.3
M_{EE_y}	-0.7 ± 0.5	-0.6 ± 0.4	-0.5 ± 0.2	-0.0 ± 0.3
M_{EE_z}	216.3 ± 0.7	212.8 ± 1.0	215.5 ± 0.5	216.2 ± 0.7

Eq. 19 was $(118.6, 0.2, 0.8)$mm. The specifications of the Bumblebee 2 stereo camera state that the distance between cameras is 120mm and with both estimation methods the difference was below 2%. There was a slight constant bias between the results of the two estimation methods in the x and slightly also on y translation. The same bias was also noticeable in Fig. 4(a) and the underlying reason is probably the same – a non-point marker and a constant difference in camera viewpoints.

TABLE III

RESULTS COMPARING ${}^{C_r}T_{C_l}$ ESTIMATED BY CAMERA CALIBRATION TOOLBOX [3] AND BY ESTIMATING TRANSFORMATION BETWEEN THE ROBOT BASE FRAME AND BOTH CAMERAS SEPARATELY (EQ. 19).

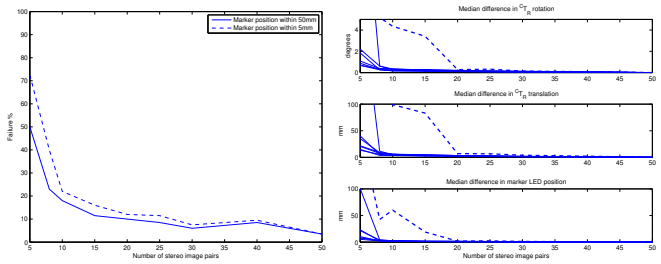
Test set	Difference in angle	Difference in translation
Static cam 1	$0.115°$	$(-1.56, 0.88, -1.33)$ mm
Static cam 2	$0.062°$	$(-1.49, 0.45, -0.87)$ mm
Static cam 3	$0.236°$	$(-1.72, 1.75, 3.59)$ mm
Static cam 4	$0.057°$	$(-0.36, 0.71, 0.17)$ mm
Static LED 1	$0.103°$	$(-1.10, 0.33, -0.22)$ mm
Static LED 2	$0.214°$	$(-5.35, 0.58, -2.49)$ mm
Static LED 3	$0.159°$	$(-1.25, 1.78, 0.29)$ mm
Static LED 4	$0.172°$	$(-2.37, -0.03, -0.31)$ mm

D. *Effect of number measurements*

These experiments study the effect of used number of measurements (stereo image pairs). For each number of measurements and test set the tests were repeated 25 times.

Fig. 5 shows the results when measuring the position of only the left camera (Eq. 18). All 8 datasets are included. Fig. 5(a) shows how often the estimated position of the marker LED was within $50mm$ or $5mm$ of the position estimated when using all 50 measurements. After 15 measurements the failure percent was fairly constant. There are 9 parameters to estimate and a single measurement gives 4 datapoints (x and y positions in two frames) so in theory three measurements are sufficient, but the estimation often fails to find the correct parameters when only 5 measurements were used. When repeating the test with the full dataset, there still were some failures because the estimation starts from partly random initialization and Gauss-Newton sometimes fails to converge. However, in those cases the errors were extremely large and the median difference was exactly zero, i.e., when the estimation succeeded the same parameters were always found.

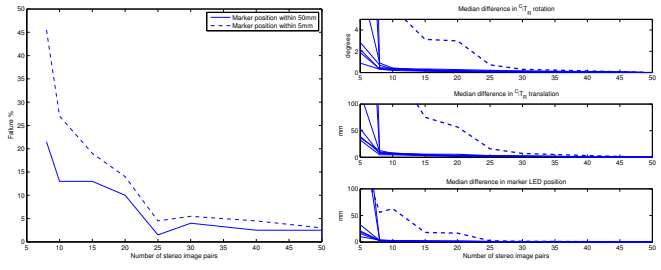
Fig. 5(b) shows the median differences in estimated ${}^{C}T_{R}$ angle and translation and in marker position separately for all 8 datasets. Results for dataset 'Static camera 3' are highlighted, because the results differ from other datasets due to the dataset having a large number of outliers (as seen in Table I). That dataset required 20 measurements for the median errors to become near the values estimated with the full dataset, but in all other datasets 10, or even 5, measurements gave very low median errors (under 10mm for marker position and under 20mm for the robot-camera translation).



(a) (b)

Fig. 5. The effect of number of measurements; (a) How often the estimation failed to find the marker position with specified accuracy; (b) Median differences in estimated ${}^{C}T_{R}$ angle and translation and in marker position for all 8 datasets, dataset 'Static camera 3' highlighted.

Similar tests were repeated also when estimating the position of the both cameras separately (Eq. 19). The results are presented if Fig. 6. For simplicity of the presentation results are presented only for the left camera. In this case there was 15 parameters to estimate, so theoretically 4 measurements are enough. With some test sets 5 measurements gave reasonably small median errors, but the increased need for measurements can be seen that for dataset 'Static camera 3' 25 measurements were needed instead of 20 which was enough in the previous experiment.

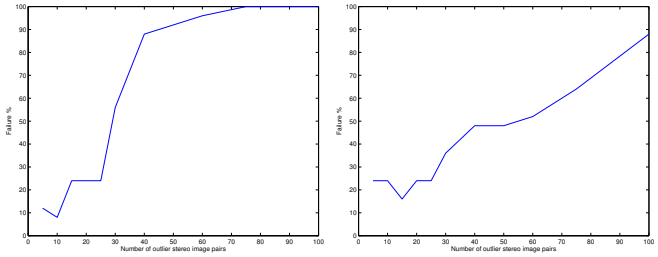


(a) (b)

Fig. 6. The effect of number of measurements when estimating position of both cameras; (a) How often the estimation failed to find the marker position with specified accuracy; (b) Median differences in estimated ${}^{C_l}T_{R}$ angle and translation and in marker position for all 8 datasets, dataset 'Static camera 3' highlighted.

E. Effect of outliers

Here the effect of number of outliers is studied. In these tests 25 best inliers, those having the largest weight values after the estimation, from one of the test sets were selected and then a number of outliers were added and the parameter estimation was performed again. For each number of outliers the test was repeated 25 times with different set ouf outliers. The results are shown in Fig. 7 where the graphs show how often the marker LED was not found within $5mm$ compared to the outlier-free estimation. In Fig. 7(a) the "outliers" are selected randomly from all other test sets and in Fig. 7(b) the marker positions are additionally randomized. In the first case the estimation begun to fail very often when the number of outliers grew over 25 but in the second case the estimation only grew linearly and still succeeded over 10% of the time with 100 outliers (4 times more than true inliers). With outliers selected from real data the estimation fails earlier because there actually are several valid parameter sets and the estimation may end up in converging to a wrong one.



(a) (b)

Fig. 7. The effect of number of added outliers with 25 valid inlier measurements; (a) Outliers added from other test sets; (b) Outliers in randomized marker positions.

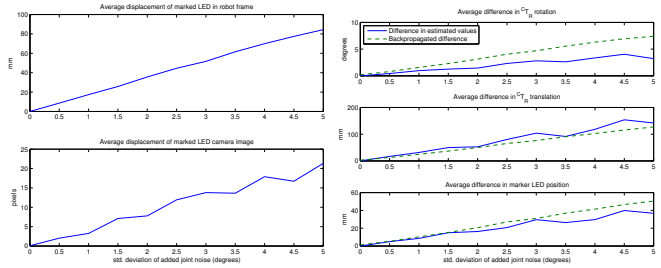
Note that the performance in the second case (randomized marker positions) would be considerably higher if the Welsch M-estimator used some adaptive scheme for selecting the c parameter instead of constant $c = 5.0$.

F. Joint noise and error backprogation

This experiment studies the effect of added noise in the joint values, adding inaccuracy to ${}^{R}T_{EE}$, and the use of backpropagating variance of residuals to the estimated parameters. Some robotic arms have considerable errors in their forward kinematics, i.e., they are not very stiff or there are other errors in joint configurations and it is useful if the error can be measured from the camera-robot calibration.

In these tests zero mean Gaussian noise was added to the joint values of the robot which are used to calculate the position of the marker LED in the base frame of the robot. To avoid having to tune the M-estimators for increasing amount of noise, a non-robust least-squares estimator was used. The test set having smallest number of outliers was therefore used, same as in Fig. 4. The results are presented in Fig. 8. The effect of added noise in the joint positions is presented Fig. 8(a), both in the world coordinates and in camera pixels coordinates. Fig. 8(b) presents the differences to non-noisy estimates. Actual realized average differences are solid line and the dashed lines presents the backpropagated variances from the variance of the residuals (pixel positions errors), see Eq. 17. Backpropagated variances were very close to the realized differences, except for the rotation matrix angle

where the overestimated error is caused by the non-linear nature of the axis-angle presentation.



(a) (b)

Fig. 8. The effect of added joint noise; (a) Displacement caused to world coordinates and pixel values; (b) Differences to parameter estimates with no added noise.

V. Conclusions

In this work a new camera-robot calibration method was presented. It is based on tracking a marker attached to the end-effector of the robot. The estimation can be performed for one or multiple camera simultaneously and a known pose between cameras in a stereo system can also be applied. The accuracy of the estimate can be established using backpropagation of the variance of the measurement residuals.

In the experiments various aspects of the parameter estimation procedure were studied. In tests where the camera or the marker was moved to different position, it was noticed that the estimations agreed with few millimeters, as well as the stereo calibration result between this method and the Camera Calibration Toolbox [3]. As few as 5 measurements were noticed to give reasonably accurate estimations, but a larger number improves the result and makes the method robust to a large number of outliers. The backpropagation of the variance of the residuals to the estimated parameters was noticed to very accurately reflect the realized inaccuracy in an artificial test where noise was added to the joint positions.

Future improvements could be changing the Gauss-Newton approximation method to more robust Levenberg-Marquardt algorithm, using an adaptive scheme for selecting the M-estimator parameters for cases where the measurement errors are unknown, and the estimation of camera intrinsic parameters could be easily added, which however would increase the amount of needed data would considerably.

VI. Acknowledgments

This research has been funded by European Commission GRASP project (IST-FP7-IP-215821).

References

[1] T. Asfour, K. Welke, P. Azad, A. Ude, and R. Dillmann. The karlsruhe humanoid head. In *Humanoid Robots, 2008. Humanoids 2008. 8th IEEE-RAS International Conference on*, pages 447 –453, 2008.

[2] M. Bauer, M. Schlegel, D. Pustka, N. Navab, and G. Klinker. Predicting and estimating the accuracy of n-occular optical tracking systems. In *Mixed and Augmented Reality, 2006. ISMAR 2006. IEEE/ACM International Symposium on*, pages 43–51, 2006.

[3] J. Y. Bouguet. Camera calibration toolbox for matlab. http://www.vision.caltech.edu/bouguetj/calib_doc/, February 2011.

[4] F. Dornaika F and C. Garcia. Robust camera calibration using 2d-to-3d feature correspondences. In *Proceedings of Videometrics V – SPIE's Optical Sience, Engineering and Instrumentation'97*, pages 123–133, 1997.

[5] C. Garcia. Fully vision-based calibration of a hand-eye robot. *Autonomous Robots, Special issue on Perception-Based Intelligent Robots*, 6(2):223–238, 1999.

[6] X. Gratal, J. Bohg, M. Björkman, and D. Kragic. Scene representation and object grasping using active vision. In *IROS'10 Workshop on Defining and Solving Realistic Perception Problems in Personal Robotics*, 2010.

[7] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision, 2nd. edition*. Cambridge University Press, 2003.

[8] J. Heikkila and O. Silven. A four-step camera calibration procedure with implicit image correction. In *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, pages 1106 –1112, June 1997.

[9] A. Hyvärinen, J. Karhunen, and E. Oja. *Independent Component Analysis*. John Wiley & Sons, 2001.

[10] Andreas Jordt, Nils T. Siebel, and Gerald Sommer. Automatic high-precision self-calibration of camera-robot systems. In *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, pages 1244 –1249, May 2009.

[11] Bo Sun, Qing He, Chao Hu, and M.Q.-H. Meng. A new camera calibration method for multi-camera localization. In *Automation and Logistics (ICAL), 2010 IEEE International Conference on*, pages 7 –12, 2010.

[12] Bill Triggs, Philip McLauchlan, Richard Hartley, and Andrew Fitzgibbon. Bundle adjustment - a modern synthesis. In *Vision Algorithms: Theory and Practice*. Springer-Verlag Berlin, Germany, 2000.

[13] R.Y. Tsai and R.K. Lenz. Real time versatile robotics hand/eye calibration using 3d machine vision. In *Robotics and Automation, 1988. Proceedings., 1988 IEEE International Conference on*, pages 554 –561 vol.1, April 1988.

[14] L. Wang, F.C. Wu, and Z.Y. Hu. Multi-camera calibration with one-dimensional object under general motions. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1 –7, 2007.

[15] K. Welke, M. Przybylski, T. Asfour, and R. Dillmann. Kinematic calibration for saccadic eye movements. Technical report, Institute for Anthropomatics, Universität Karlsruhe, 2008.

[16] Z. Zhang. A flexible new technique for camera calibration. Technical report, Microsoft Research, 1998 (last update in 2009). http://research.microsoft.com/ zhang/Calib/.

[17] Z. Zhang. A flexible new technique for camera calibration. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(11):1330 – 1334, November 2000.

[18] Zhengyou Zhang. Parameter estimation techniques: a tutorial with application to conic fitting. *Image and Vision Computing*, 15(1):59–76, 1997.

Emptying the box using blind haptic manipulation primitives

Javier Felip, Jose Bernabé and Antonio Morales

***Abstract*— This paper shows the application of the manipulation primitives paradigm in order to solve a complex manipulation task. The paradigm uses a set of atomic elements to define actions called manipulation primitives. The primitives are basic controllers that define simple actions such as grasp, lift, transport place and release. All the primitives are sensor based and implement a reactive behavior that adapts its actions to the uncertain and changing real environment. The task consists in emptying a box which location is barely known, and which contains an undefined number of unknown objects.**

In this paper we compare two different approaches, blind and vision based, to obtain the main parameters of the task: starting pose, end pose and hand preshape.

I. Introduction

Robotic manipulation on real and unstructured environments is one of the current challenges in robotics. One of the main problems is the inherent uncertainty that shows up in this scenarios. There are several sources of uncertainty, lack of knowledge about the physical properties and shape of the objects, inaccuracy in hand-eye calibration, sensing errors, limitation of planner models and so on.

As a result, it is very difficult to have enough precision in order to move the fingertips to a determined position. Thus establishing a precise grasp is quite difficult. An approach to deal with the environmental uncertainty are the independent contact regions [1] that compute areas to place the fingers instead of points allowing some error in the final position of the fingertips. If the contact points are not taken into consideration, sensor based strategies are the most popular approach to adapt actions to environment conditions. Teichmann and Mishra [2] proposed the use of a light beam sensor to align the gripper with an unknown object. More recently, other solutions such as IR proximity sensors [3], tactile sensors [4] [5], force and tactile feedback [6] were proposed. A totally different approach is to design a compliant hand that adapts to the object [7].

Executing manipulation tasks and actions for complex humanoid robots has always been a tough control problem. In this paper we use the manipulation primitive paradigm to define atomic actions, the reduction of possible actions reduces complexity and eases planning. There are several definitions of primitives: to control a hand [8], to define object movements [9] and its relations [10] or to control a manipulator [11]. Despite the different approaches to define primitives, they have a common purpose: discretize and reduce the complexity of controlling the robotic setup to reduce the search space for planning issues.

J. Felip, J. Bernabé and A. Morales are with Robotic Intelligence Laboratory at the Department of Computer Science and Engineering, Universitat Jaume I, 12006 Castellón, Spain {jfelip,jbernabe,morales}@uji.es

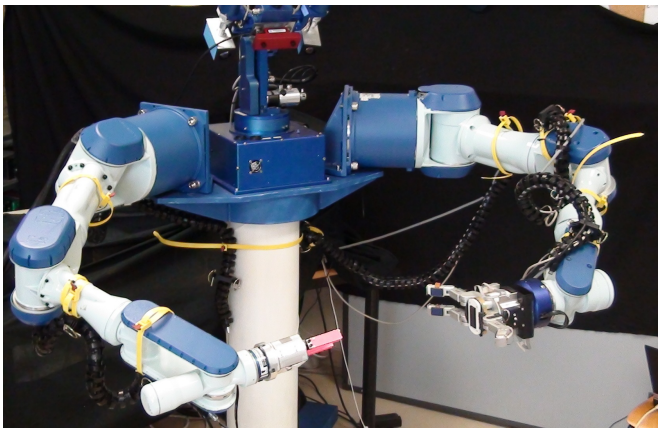


Fig. 1. The experimental robotic platform: Tombatossals, the UJI humanoid torso.

In literature, terms like primitive, task or plan have many different hues. In this paper a *manipulation primitive* is a single controller that performs a specific action on a particular embodiment. From an abstract point of view, primitives are the simplest pieces of a vocabulary to define tasks. Hence, a *task* is defined by a sequence of manipulation primitives that require some information (i.e. parameters) to tune its behavior to the specific situation. Although some of the parameters are defined by the task, there are still some parameters that must be defined. A task with defined parameters is called *plan*.

Paper outline

In this paper, we use manipulation primitives that relay in force, tactile and visual feedback to adapt its behavior to the real environment. We use our primitive approach to define and solve a complex manipulation task like emptying a box full of objects in any position. Section II describes the design principles of tasks and primitives. It also presents two different approaches to parametrize the defined task with and without vision. This two approaches are tested and validated, results are described and discussed in sections III and IV.

II. Methodology

Emptying a box consists on repeating a pick and place task for each object in the box. As pointed in section I, a task is defined using a set of manipulation primitive controllers. Each set of primitives is structured and executed as a Finite State Machine (FSM) where each primitive represents a state. In this paper we have defined a pick and place task using

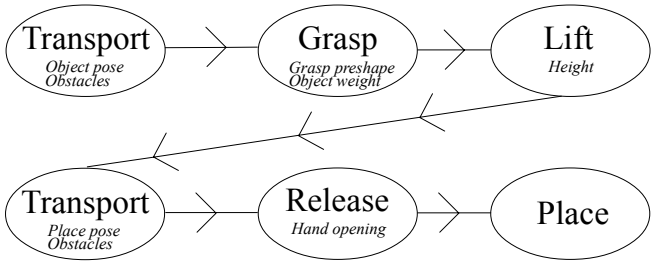


Fig. 2. Diagram of the sequence of primitives that compose a pick and place task. Inside each primitive, some examples of parameters are written in italics.

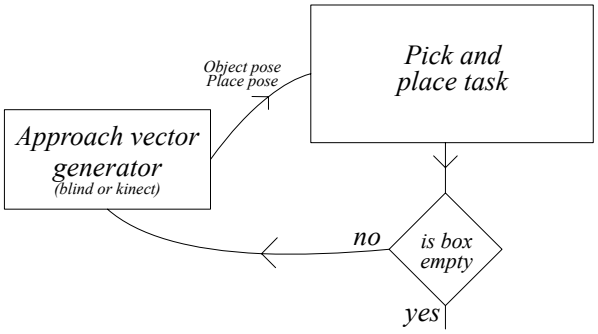


Fig. 3. Diagram to solve the empty the box task using a loop of pick and place tasks and the generation of approach vectors.

a subset of the available manipulation primitives. This task allows the robot to pick up an object from a staring position and place it to a destination position.

The set of primitives that form the pick and place task was defined in advance and is composed of: transport, grasp, lift, place and release. The FSM formed by this primitives is depicted in Fig. 2. As long as the primitives are parametrizable, this sequence defines a parametrizable pick and place task. The required parameters are the approach vector to the object to be grasped and the target position to place it. It is possible to set up some optional parameters to improve the performance of the primitives such as object size, weight or hand preshape.

The approach vector defines the starting position and direction to start grasping the object, in section II-D we detail two different strategies to provide the approach vectors: blind and vision based. An approach vector is defined as a 6D vector containing position and orientation using the Roll-Pitch-Yaw (RPY) agreement, see eq.1.

$$\vec{p} = (p_x, p_y, p_z, p_\gamma, p_\beta, p_\alpha) \tag{1}$$

A. Algorithm

The algorithm that solves the problem is structured in two levels of abstraction. The lower level (Fig. 2) is the primitive level, primitives could be combined to define a task. The upper level (Fig. 3) uses sequences of tasks to define more complex manipulation tasks.

For the specific problem of emptying a box, we have defined the pick and place task (see Fig. 2) that allows the robot to grasp an object and place it on the target position. This pick and place task, is repeated as long as there are objects left in the box to get the task done (see Fig. 3).

The output of the algorithm is the execution of the task, the minimum input needed is the approach vector for each object and the target pose to place it. For this work we have considered the place position to be common to all the objects. The approach vector generation methods are detailed in Sec. II-D.

B. System description and Assumptions

1) Hardware description: The torso system, called *Tombatossals* has 23 DOF (see Fig. 1). It is composed of two 7 DOF Mitsubishi PA10 arms. The left arm has a 4 DOF Barrett Hand [12] and the right arm has a parallel jaw gripper. Each arm has a JR3 Force-Torque sensor attached on the wrist between the arm and the hand. The visual system is composed of a TO40 4 DOF pan-tilt-verge head with two Imaging Source DFK 31BF03-Z2 cameras. Attached to the centre of the pan-tilt there is a $Kinect^{TM}$ sensor from $MicrosoftCorp$. For this work only the left arm, the pan-tilt head and the kinect system was used.

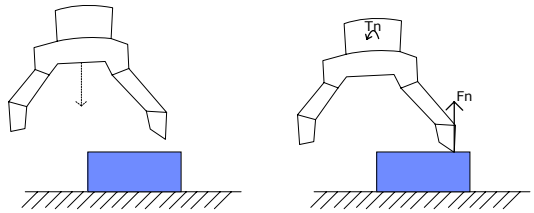
2) Assumptions: The object position inside the box is not restricted, objects can be in any position and orientation inside the box, except that it must be possible to grasp any object with the Barrett hand without needing to move it before. This means that the objects that are shorter than the box walls cannot be too close to the box walls, the separation between this kind of objects and the walls must be enough for the robot fingers to fit in (3cm).

The object maximum and minimum size is defined by the Barrett hand dimensions. All the objects must fit inside the hand and be graspable. The box must be on an even plane (i.e. table) inside the arm workspace.

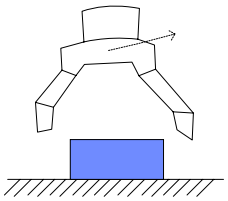
C. Manipulation primitives

Primitives are parametrizable and all of them require a common parameter: a pose. All other parameters are optional and, if present, will help to improve performance and robustness. Each primitive may read the pose with a different purpose, for example, if using a transport primitive the pose will represent the destination position to move the arm to, on the other hand, if using the grasp primitive the pose will be used as the approach vector to the object (i.e. starting position). In the next subsections a brief description of each primitive used for this work is presented.

1) Grasp primitive: The main features of the grasp primitive and its parameters are described in [6]. That grasp primitive has been improved with another two correction methods: translation error correction (see Fig. 4) and sliding grasp (see

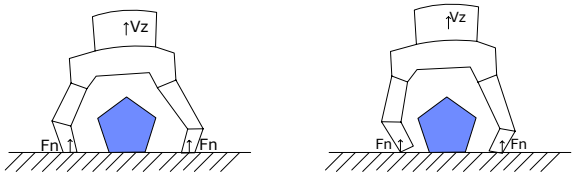


(a) Arm moving towards the object. (b) Contact generates torque in the wrist.



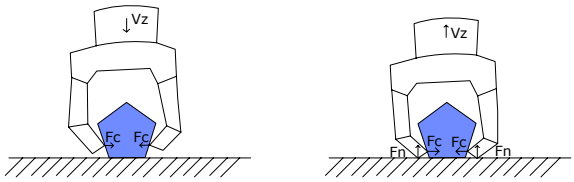
(c) Correction movement is performed.

Fig. 4. Translation error correction strategy.



(a) The fingers contact the table while closing. Thus the controller sets the velocity to move the hand back.

(b) The fingers continue closing and the contact force keeps moving back the hand.



(c) The fingers are closing and the contact with the table is lost. Vz is set forwards.

(d) The hand contacts the table again but the object is already grasped.

Fig. 5. Sliding grasp strategy.

Fig. 5) the latter replaces the parallel face detection phase of this primitive.

The execution of the slide grasp is depicted in Fig. 5. The hand starts closing and when the fingers make contact with the surface, the force they are applying is felt in the wrist Fig. 5(a), thus the arm moves back. The fingers continue closing and because there is no force felt, the arm moves forward Fig. 5(c). When the fingers are not able to continue closing and there is no force felt in the wrist Fig. 5(d), the primitive ends successfully.

2) Transport primitive: Moves the arm while it holds an object to the specified target position. It can also be used to move the arm without any object. This primitive has some optional parameters such as obstacle definition. If the obstacles are defined it will use a force-field [13]

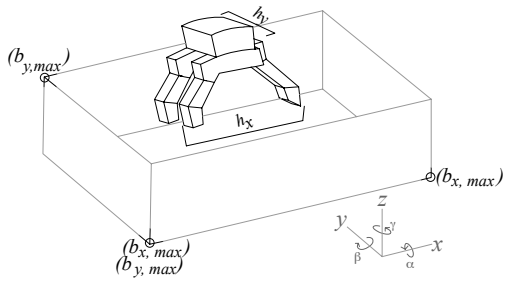


Fig. 6. Input parameters for blind and vision based approach vector generation.

based collision avoidance strategy to generate a collision free motion from current to target position.

3) Place primitive: The arm moves down until a contact is detected. It uses the force/torque sensor to detect a force opposing the movement direction, when it happens, the controller assumes that the object is placed.

4) Release primitive: This primitive opens the hand slowly. The movement of the arm is force-controlled and the arm moves back if a contact between the opening fingers and the environment is detected.

D. Generating approach vectors

We propose two strategies to generate approach vectors. A blind method and a simple vision based method. Both methods have been proposed in order to compare the results of adding vision to an already working blind task.

Both approach vector generators assume that the arm is able to perform top grasps on any position over the box. All the approach vectors generated have the same direction defining always top grasp. The box position is defined by the user, in the blind grasping is introduced in a configuration file while in the kinect version the user clicks on the image the corners of the box to fix its location.

An approach vector is defined as (eq. 1) where p_x, p_y and p_γ (roll) are computed by the proposed methods, p_z is fixed by the user and p_β (pitch),p_α (yaw) are fixed by top grasp.

1) Blind method: Variable elements of approach vectors (p_x,p_y,p_γ) are randomly generated as shown in (eq. 2) where $U(0,1)$ is a random uniform function and $p_{x,min}$, $p_{x,max}$, $p_{y,min}$, $p_{y,max}$ are defined in (eq. 3) and are function of p_γ due to Barrett hand is not symmetrical. The parameters $b_{x,min}$, $b_{x,max}$, $b_{y,min}$, $b_{y,max}$ are the dimensions of the box (See Fig. 6) and h_x,h_y are the dimensions of the hand in x and y axis of hand frame.

$$\begin{aligned} p_\gamma &= round(U(0,1)) * \frac{\pi}{2} \\ p_x &= p_{x,min} + U(0,1) \cdot (p_{x,max} - p_{x,min}) \\ p_y &= p_{y,min} + U(0,1) \cdot (p_{y,max} - p_{y,min}) \end{aligned} \tag{2}$$

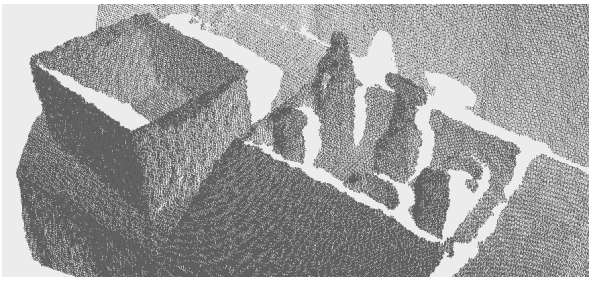
$$\rho = \frac{(h_x)^2+(h_y)^2}{4}$$
$$\phi = \arctan\left(\frac{h_y}{h_x}\right)$$
$$p_{x,min} = b_{x,min} + \sqrt{\rho \cdot \cos(p_\gamma + \phi)^2}$$
$$p_{y,min} = b_{y,min} + \sqrt{\rho \cdot \sin(p_\gamma + \phi)^2} \quad (3)$$
$$p_{x,max} = b_{x,max} - \sqrt{\rho \cdot \cos(p_\gamma + \phi)^2}$$
$$p_{y,max} = b_{y,max} - \sqrt{\rho \cdot \sin(p_\gamma + \phi)^2}$$

2) Vision Based Approach: The vision system uses the $Kinect^{TM}$ sensor from $MicrosoftCorp$. Kinect outputs a depth image and a common RGB image. By the combination of both images a RGB 3D point cloud is obtained (Fig. 7(a)). The use of point clouds enables the use of clustering to detect separate objects and also PCA to calculate each cluster main orientations. For the voxel filter and clustering algorithm implementation we have used the Point Cloud Library (PCL) [14] from ROS.

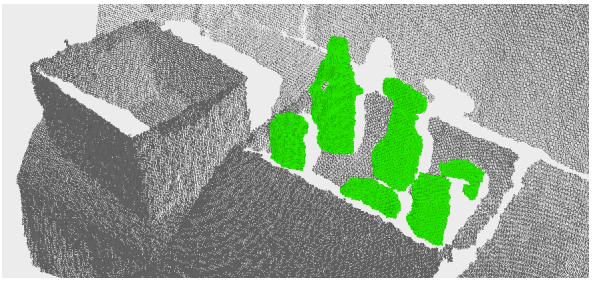
The process to obtain an approach vector over an object is divided in the following phases (see Fig. 7):

- Background extraction (see Fig. 7(c)): Using a virtual box with the same dimensions as the real one, all the points outside the box are labeled as obstacles if they are inside the arm workspace or background if they are outside. The points inside the box are labeled as object points.
- Voxel filtering: Clustering has a high computational cost which depends on the number of points to be clustered. A voxel filter reduces the number of points while keeping object geometry almost untouched.
- Clustering (see Fig. 7(d)): Using an implementation of the the Kd-Tree method [15], all the object points are labeled to belong to one cluster. Usually each cluster represents one object but it is possible that, if some objects are in contact, they might be classified as the same cluster.
- Centroid and PCA [16]: At this point, one cluster is selected and its approach vector is calculated as follows: p_x,p_y and p_z are extracted from the centroid of the selected cluster. p_β and p_α are fixed to perform a top grasp (i.e. perpendicular to the table plane) and p_γ is obtained from the orientation of the clusters' main axis obtained using Principal Component Analysis (PCA).

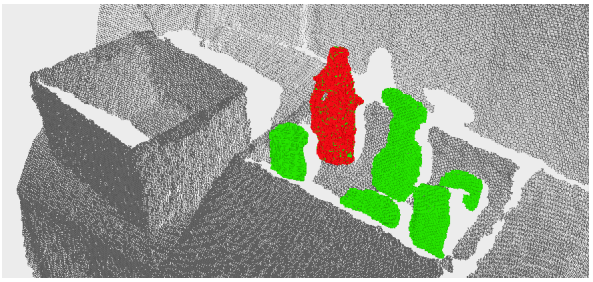
(a) Original 3D image.



(b) Original 3D point cloud read from Kinect sensor.



(c) Virtual box background filtering. Background points are colored in gray and objects are in green.



(d) Object clustering and selection. Background points are marked in gray, objects in green, and the selected cluster is labeled in red

Fig. 7. 3D point cloud segmentation phases.

III. Experimental results

A. Experimental setup

The experimental setup consists on a table in front of the robot. On the table there is a box full of graspable objects in any position (even stacked), the only restriction is that no pregrasp movement of the object is needed.

B. Results

We have obtained only some qualitative preliminar results. For the blind grasping we have executed the experiment and the robot succeeded emptying the box 4 out of 5 times. Although the first objects are grasped quickly, when there is only one object left it takes more time for the random approach vector generator to generate a vector close enough to the object to grasp it. The unsuccessful attempt was caused by an ungraspable position of an object, the object moved during a grasping attempt to a corner of the box. All the objects graspable by the Barrett hand that are in a graspable position will be grasped sooner or later.

On the other hand, using vision approach vectors are generated always over an object and the whole process is faster because less tries are required. Using the kinect vision the success rate was 2 out of 2 and the time taken to end the task was much lower.

IV. Conclusion

In this paper we have presented the application of manipulation primitive controllers to define a parametrizable pick and place task. Using the defined task, a solution to a complex manipulation task has been implemented and validated. It was also demonstrated that it is possible to solve the task with and without vision relaying on sensor based primitives that adapt their behavior to the environment. Moreover it was also shown that adding vision to the process improves the performance of the task. Thus, selecting good approach vectors is a key point for this task and we can use our current setup as a testbench for top grasp approach vector generators.

As future work, some of the assumptions taken to simplify the task and ease the programming will be removed. To solve the issue of objects too close to the box walls, we plan to add pregrasp movements to slide the objects away from the wall before grasping them.

References

[1] M. Roa and R. Suarez, "Computation of independent contact regions for grasping 3-d objects," *Robotics, IEEE Transactions on*, vol. 25, no. 4, pp. 839 –850, 2009.

[2] M. Teichmann and B. Mishra, "Reactive algorithms for grasping using a modified parallel jaw gripper," in *Robotics and Automation, 1994. Proceedings., 1994 IEEE International Conference on*, pp. 1931 –1936 vol.3, May 1994.

[3] K. Hsiao, P. Nangeroni, M. Huber, A. Saxena, and A. Y. Ng, "Reactive grasping using optical proximity sensors," in *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, pp. 2098 –2105, May 2009.

[4] K. Hsiao, S. Chitta, M. Ciocarlie, and E. Jones, "Contact-reactive grasping of objects with partial shape information," in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pp. 1228 –1235, 2010.

[5] D. Gunji, Y. Mizoguch, S. Teshigawara, A. Ming, A. Namiki, M. Ishikawa, and M. Shimojo, "Grasping force control of multi-fingered robot hand based on slip detection sing tactile sensor," in *SICE Annual Conference, 2008*, pp. 894 –899, 2008.

[6] J. Felip and A. Morales, "Robust sensor-based grasp primitive for a three-finger robot hand," in *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pp. 1811 –1816, 2009.

[7] A. Dollar and R. Howe, "Simple, reliable robotic grasping for human environments," in *Technologies for Practical Robot Applications, 2008. TePRA 2008. IEEE International Conference on*, pp. 156 –161, 2008.

[8] T. Speeter, "Primitive based control of the utah/mit dextrous hand," in *Robotics and Automation, 1991. Proceedings., 1991 IEEE International Conference on*, pp. 866 –877 vol.1, Apr. 1991.

[9] P. Michelman and P. Allen, "Forming complex dextrous manipulations from task primitives," in *Robotics and Automation, 1994. Proceedings., 1994 IEEE International Conference on*, pp. 3383 –3388 vol.4, May 1994.

[10] J. Morrow and P. Khosla, "Manipulation task primitives for composing robot skills," in *Robotics and Automation, 1997. Proceedings., 1997 IEEE International Conference on*, vol. 4, pp. 3354 –3359 vol.4, Apr. 1997.

[11] Y. Hasegawa, M. Higashiura, and T. Fukuda, "Object manipulation coordinating multiple primitive motions," in *Computational Intelligence in Robotics and Automation, 2003. Proceedings. 2003 IEEE International Symposium on*, vol. 2, pp. 741 – 746 vol.2, 2003.

[12] B. T. inc., "www.barrett.com."

[13] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," in *Robotics and Automation. Proceedings. 1985 IEEE International Conference on*, vol. 2, pp. 500 – 505, Mar. 1985.

[14] R. Rusu, "www.pointclouds.org."

[15] J. L. Bentley, "Multidimensional binary search trees used for associative searching," *Commun. ACM*, vol. 18, pp. 509–517, September 1975.

[16] I. T. Jolliffe, *Principal Component Analysis*. Springer, 1986.

Acknowledgment

This paper describes research carried out at the Robotic Intelligence Laboratory of Universitat Jaume I. The research leading to these results has received funding from the European Community's Seventh Framework Programme FP7/2007-2013 under grant agreement ICT-215821 (GRASP project).

Probabilistic Approach to Sensor-based Grasping

Janne Laaksonen and Ville Kyrki

Abstract— **In this paper, we present a probabilistic framework for grasping. In the framework, we consider grasp and object attributes, tactile information and the stability of a grasp, as probability distributions. We describe how the probability distributions can be used to plan for a stable grasp and how object attributes can be updated simultaneously using tactile information gained during grasping. We demonstrate the framework in simulation.**

I. INTRODUCTION

Current grasp planning approaches are usually based on perfect knowledge of objects. While geometric models are good approximations of the objects in the real world, they never are exactly the same, especially when speaking of household items. From these approximations, arises the error between the expected and the realized grasp, although usually small enough to achieve a stable grasp. This error is usually left unused.

On the other side, we have methods that use sensor information to grasp, using corrective motions or reacting to the tactile sensor information. Contrary to grasp planners, accurate object models are not usually available in this type of grasping. Thus, the only way to model the stability of a grasp is through the sensors of a manipulator.

In this paper, we present a probabilistic framework, which unifies the ideas behind grasp planning and reactive grasping. The probabilistic framework considers all required variables for grasping and allows the variables to be represented as probability distributions. The framework allows interplay between grasp planning and corrective motions, in situations that object attributes, such as pose, are not precisely known, by utilizing sensor information gained during grasping. We demonstrate the framework with a simple 2D example. In the demonstration, we use particle filters, a MCMC (Monte Carlo Markov Chain) method, to estimate the probability distributions.

Section II collects the related work and Section III describes the probabilistic framework. In Section IV, a practical implementation based on the presented framework is presented. We conclude in Section V.

II. RELATED WORK

Our approach to find good grasps is closely related to the field of grasp planning. In grasp planning the goal is to find as good as possible grasp on a given object. The goodness of the grasp is usually measured with a grasp quality measure [1]. However, compared to our method, most current grasp planning methods do not account for the uncertainty present in the object or in the object's pose information. Also most of the grasp planning methods require a known geometric model of the object.

To simplify the grasp planning, many methods employ some form of decomposing the object. The goal of the decomposition is to reduce the amount of feasible grasps without trying every grasp on an object. In [2], the object is decomposed to minimum volume bounding boxes, in an effort to understand the underlying shape of the object. The primitive shape is then used to reduce the search space for stable grasps. Instead of boxes, superquadrics are used in [3]. In addition to the construction of the superquadric decomposition, heuristic is used to define the trial grasps based on the superquadric form of the object limiting the space of grasps significantly.

The Columbia Grasp Database [4] takes a different approach to most grasp planners and compute best grasps for a set of hundreds of objects. The grasp planning problem is then transformed to a problem of matching a new object with an object found in the precomputed database of grasps. The work has also been extended to consider partial data [5].

If the object is not known, i.e. a geometric model is not available, the grasp planning methods can still be used if the model of the object can be constructed. The model construction can either be done by vision or tactile exploration. However, the geometric model in this case is usually a mesh or a point cloud, and contains no information about the inherent uncertainty related to the perception. Approaches such as [2] can be applied here as well but the results can be worse than in the cases where the full geometric model is known and the decomposition may fail in cases where large volumes are missing from the perceived object.

Another approach for finding grasps is object affordance modeling. While object affordance is a broader subject, the affordances can also be thought in the sense of grasp stability. In some of the grasp related studies, grasp affordances consider the overall stability of the grasp [6], [7] or, for example, the grasp affordance in specific tasks [8].

Learning to find good grasps is another view on the problem. [6] utilizes learning on a real robot to learn the grasp affordance of an object. The learning process reduces a vision bootstrapped distribution of grasps to a smaller set of grasps containing only good grasps. Reinforcement learning [9] can also be applied, so that a sequence of grasps can be learned which will lead to a stable grasp of an object.

The research leading to these results has received funding from the European Community's Seventh Framework Programme under grant agreement n° 215821.

J. Laaksonen and V. Kyrki are with Department of Information Technology, Lappeenranta University of Technology, P.O. Box 20, 53851 Lappeenranta, Finland, jalaakso@lut.fi, kyrki@lut.fi

Our approach to grasping is more related to the methods found in [10] and [11]. The aim of [10] is to reduce the uncertainty of a object's pose to enable grasping the object. In [11], the shape of the object is also uncertain in addition to the pose. In both of the studies, the method is presented with a parallel jaw gripper grasping a 2D-object. However, these methods do not utilize sensor information gained during grasping. Also in [12], the authors propose a decision-theoretic controller which minimizes the uncertainty of the object pose using arm trajectories to enable task specific grasps on objects. Tactile sensors were used to detect contacts between the hand and the objects.

This paper will present a probabilistic approach for finding a stable grasp and if necessary, refine the grasp by regrasping, so that even better grasp can be found for an object and at the same time reduce uncertainty of an object's pose. As can be seen from our survey of recent grasp planners and other grasping methods, similar grasping frameworks have not been yet published to our best knowledge.

III. GRASPING IN PROBABILISTIC FRAMEWORK

We model sensor-based grasping using the following variables: S denotes the stability of a grasp as a binary value, G the grasp attributes (e.g. the pose of the end-effector), O the object attributes (e.g. the pose of the target object) and T represents measurable tactile information. The variables have characteristics: G, the grasp attributes, can be controlled, T can be measured for each grasp attempt, while O is uncertain, that is, we assume we only have an uncertain initial estimate of the object attributes.

In our framework, traditional grasp planning algorithms try to maximize the stability, S, by controlling the grasp attributes, G, with perfect knowledge of the object attributes O,

$$\max_G \; P(S|G,O) \; . \tag{1}$$

In our model, O is not precisely known but instead represented as a probability distribution.

It has been shown that grasp stability can be estimated using tactile information [13]. Thus, we can build a probabilistic model for the stability given the other variables, $P(S|G,O,T)$. That model can be used to assess the stability of a single grasp attempt, as shown in the reference cited above. Moreover, for stability detection with uncertain object knowledge, we can marginalize over the uncertain object attributes, such that the probability of a stable grasp given the grasp attributes and tactile measurements is given by

$$P(S|G,T) = \int P(S|G,O,T)P(O|G,T)\,\mathrm{d}O \; . \tag{2}$$

If the grasp attributes are also uncertain, we can marginalize over them in a similar fashion to find $P(S|T)$. This is also the model for grasp stability for the case where no information about the object or grasp is used for stability recognition.

In order to perform grasp planning, we need also to marginalize over the distribution of object attributes. That is we need to find the mode of $P(S|G)$. This gives,

$$P(S|G,T) = \int P(S|G,O,T)P(O|G,T)\mathrm{d}O \; . \tag{3}$$

Since the tactile information for a future grasp attempt is not available, we approximate the first term in the integral by $P(S|G,O)$ and use the tactile information only to update the posterior distribution for the object attributes. Thus, after some tactile information has been collected, for grasp planning we find the maximum

$$\max_G P(S|G) \approx \max_G \int P(S|G,O)P(O|G,T)\mathrm{d}O \; . \tag{4}$$

Equation (4) shows that the stability S can be maximized by finding the best grasp G, when G and T (from the previous attempt) are known. To build a working system based on the Equation (4), two models are needed:

- Model for $P(O|G,T)$, describing relation between tactile information and grasp and object attributes.
- Model for $P(S|G,O)$, stability as a function of grasp and object attributes

Unfortunately, these models are not trivial to build and depend on the object and the manipulator used to grasp the object. Still, there are existing models for both cases, e.g. see [14] for a model for $P(O|G,T)$ and [15], [13] for a model for $P(S|G,O,T)$. One approach to generate the models is to simulate the object and the manipulator to produce the required tactile information and stability models. We have used this approach to demonstrate the framework in action in Section IV.

Our framework does not place constraints on the actual models, and the attributes G, O, T can be freely chosen. For example, G and O can include the poses of the manipulator and the object. The benefit of the presented probabilistic framework is that throughout the grasping process uncertainty of the actions arising from equation (4) is known. Also, measurement errors can be accounted for during both grasp planning as well as on-line grasp stability detection.

IV. DEMONSTRATION

We will demonstrate the validity of our method using a simulated environment. The environment is depicted in Figure 1. The environment consists of a parallel jaw gripper with finger width $l_{\texttt{finger}}$ and a rectangular object with side lengths of 6 and 2. The angle of the gripper in degrees is denoted by θ, which is zero when the gripper is perpendicular to the long side of the object. The gripper center is denoted with (x, y), which is relative to the object center (x_0, y_0). In the demonstration, the object is static. When grasping, we can close the two fingers of the gripper independently of each other and we will use the distances d_1 and d_2 as the measurements, representing the tactile information T. We assume that the fingers have the capability to detect when they come into contact with the object and that we can stop the fingers at that instance. We will use 3-tuple (x, y, θ) to denote the gripper variables, which are in relation to the object center (x_0, y_0, θ_0). (x, y, θ) represent G, the

grasp attributes, while (x_0, y_0, θ_0) represents the O, object attributes.

Note that with this setup, there is always ambiguity about the orientation of the gripper when the fingers are in contact with the top and bottom sides of the object. Also due to the symmetry, we can not reduce uncertainty in the x-axis.

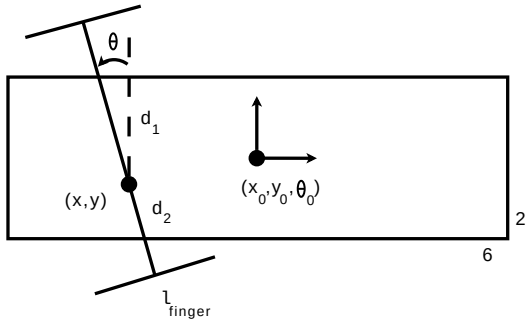


Fig. 1. Simulation environment.

1) Implementation: Our general approach is based on a sequence of actions, shown in Figure 2. We assume that some type of initial estimate of the object pose is given (1), e.g. from vision. Using the estimate, we can plan for a grasp with the uncertainty from the initial estimate (2). Then a grasp is performed (3), giving measurement data (we assume tactile and joint configuration data is available). Using the measurement data, we can make a decision of the grasp stability (4), if the grasp is stable, the object can be manipulated, if not, we can plan for a new grasp (5) with the new information from the attempted grasp. This loop can then be further iterated until grasp stability conditions are satisfied.

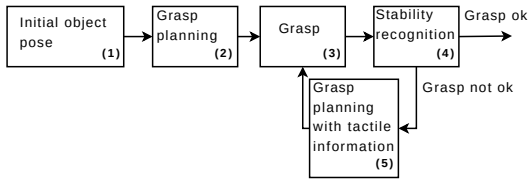


Fig. 2. Sequence of actions.

The theoretical framework described in Section III is implemented with particle filters to make the computation of probability distributions tractable. The particle filter method is a MCMC method, and estimates probability distributions with a cloud of particles. More information on particle filtering, especially applied to robotics can be found in [16]. Particle filters has been used in manipulation, for example in [14], to estimate object pose using tactile sensors. We use two different particle filter processes to estimate the two different models, $P(O|G,T)$ and $P(S|G,O)$, introduced in Section III. Likelihoods, which are shown in Figure 3, were chosen by hand for the purposes of this example. Figure 3(a) shows how likely a measurement d is a correct measurement in relation to the true measurement d^*, this model is used for both d_1 and d_2. Figure 3(b) presents how likely a grasp is stable relative to the belief of object pose O.

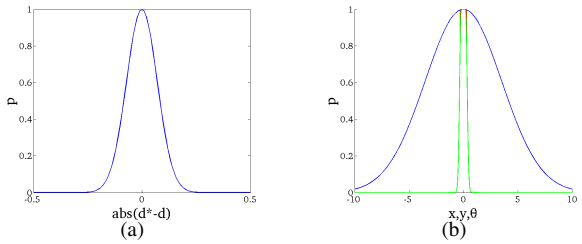
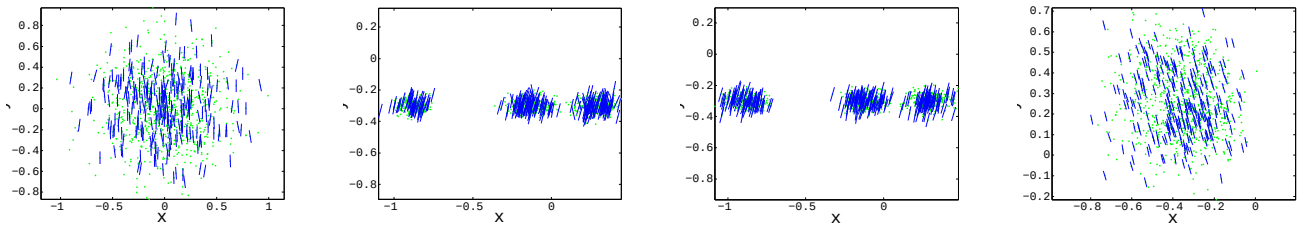


Fig. 3. Likelihoods for: (a) Measurement model, $p(d|d^*)$; (b) Grasp stability, $P(S|G,O)$, for (x, y, θ), x in red, y in green, θ in blue.

Algorithms 1 and 2 describe our method of finding stable grasps. The algorithms also contains the variables and distributions that we have used in particle filter processes. Algorithm 1 requires the initial estimates of the uncertainty, given in $\sigma_{\mathbf{init}}$, for each of the variables (x, y, θ). The particle set O_1 in Algorithm 1 represents the probability distribution of the object, i.e. $P(O|G,T)$, while particle set G_1 in Algorithm 2 represents the relative or corrective motion to the actual grasp, and by applying the relative motion to each of the particles in O_1, we can find the probability of a stable grasp $P(S|G,O)$. In Algorithm 2 the maximum of distribution, $\max_G \int P(S|G,O)$, is searched for and the corresponding relative motion is then applied.

Referencing Figure 2, Algorithm 2 takes care of the grasp planning, that is, steps (2) and (5). Algorithm 1 handles step (3), grasping the object and updating the belief of object pose. In line 13 of Algorithm 1, the grasp stability probability is computed and corresponds to step (4) of the action sequence.

2) Results: Figure 4 shows a single example run of the Algorithm 1. The example was run with the initial object pose $(x_0,\ y_0,\ \theta_0)$ set to $(0,\ -0.3,\ -15)$. Particle locations are shown in green and $\frac{1}{4}$ of the particles are plotted with blue line, indicating the orientation, θ_0. Figure 4(a) shows the initial distribution of O_1, where $\sigma_{\mathbf{init}} = [0.3\ 0.3\ 6]$. The grasp planning stage will produce a near zero relative motion after the initial object pose distribution is given, as there are no measurements yet. In Figure 4(b), the first grasp attempt has been made, and the distribution of object pose changes to account for the measurements, d_1 and d_2. The figure also shows the symmetry of the problem and two modes arising from this symmetry, one for $\theta_0 = 15$, other for $\theta_0 = -15$. This grasp does not satisfy the threshold of 0.5 for the grasp stability probability. Maximizing $P(S|G,O)$ yields solution $(0.07,\ -0.32,\ -14.3)$ for the grasp G. Figure 4(c) shows the final distribution of O_1, after the information from the second grasp. This grasp is stable as the probability of a stable grasp is 0.503. The mean of the distribution O_1 was $(-0.24,\ -0.32,\ -14.28)$ compared to the set pose which was $(0,\ -0.3,\ -15)$. As can be seen, the method was able to find a corrective motion for the gripper and produce a stable grasp and after the two grasps, the particle cloud converged to



(a) (b) (c) (d)

Fig. 4. Sequential distributions of particles modeling $P(O|G,T)$, (a)-(c), for a single run of Algorithm 1: (a) Initial distribution ; (b) Distribution after the first grasp; (c) Distribution after second grasp, for which $P(S|G,O) = 0.503$; (d) An example of distribution $P(S|G,O)$

Algorithm 1 find_stable_grasp($\sigma_{\mathbf{init}}$)

1: Generate initial particle set, O_1 according to $\mathcal{N}(0, {\sigma_{\mathbf{init}}}^2)$
2: $q \leftarrow 1$
3: **while** $q = 1$ **do**
4: $(x,\ y,\ \theta) \leftarrow$ **find_best_relative_motion**(O_1,σ_{init})
5: Apply motion $(x,\ y,\ \theta)$ to gripper
6: Grasp object
7: **while** O_1 is not converged **do**
8: For each particle, simulate the finger lengths, d_1 and d_2
9: Weigh particles O_1, $w_1 \propto p(d|d^*)$, i.e. estimate $P(O|G,T)$
10: Do importance filtering according to w_1
11: Use $\mathcal{N}(0, {\sigma_{\mathbf{1}}}^2)$ as proposal distribution with $\sigma_{\mathbf{1}} \leftarrow [0.02\ 0.02\ 2]$
12: **end while**
13: Approximate $P(S|G,O)$ by $\sum_i P(S|G,O_{1_i})$
14: **if** $\sum_i P(S|G,O_{1_i}) > 0.5$ **then**
15: $q \leftarrow 0$
16: **end if**
17: **end while**

Algorithm 2 find_best_relative_motion(O_1,σ_{init})

1: Generate particle set, G_1 according to $\mathcal{N}(0, \mathbf{5}{\sigma_{\mathbf{O_1}}}^2)$
2: **while** G_1 is not converged **do**
3: Weigh particles G_1, $w_2 \propto P(S|G,O)$
4: $(x_{max}, y_{max}, \theta_{max}) \leftarrow \max_G P(S|G,O)$
5: Do importance filtering according to w_2
6: Use $\mathcal{N}(0, {\sigma_{\mathbf{2}}}^2)$ as proposal distribution with $\sigma_{\mathbf{2}} \leftarrow 0.2\ \sigma_{\mathbf{init}}$
7: **end while**
8: **return** $(x_{max}, y_{max}, \theta_{max})$

near optimal values for the object pose, except the x-variable for which the uncertainty can not be reduced.

However, as the method is probabilistic, maximizing the probability $P(S|G,O)$ can produce a motion that is opposite to the correct one in θ, but after the mistaken grasp, the two modes have been eliminated, and during the next iteration of Algorithm 1, the correct motion will be found. In the case of this example, the system will usually make two or three grasps before finding a stable grasp, depending on the first corrective motion.

One of the benefits of probabilistic approach is also that we always know the uncertainty behind the actions. Figure 4(d) shows an example of probability distribution of $P(S|G,O)$. From this distribution, the uncertainty of the corrective motion can be observed and if needed, constraints can be placed, for example, to allow only very precise motions in relation to the grasp stability. The same can be applied to the probability distribution $P(O|G,T)$, for example, if the accuracy is not enough for grasping at a certain time instance, we can use additional measurements from vision to update the object pose.

V. CONCLUSIONS AND FUTURE WORK

We have presented a novel framework for grasping, which operates in a probabilistic setting. The framework allows grasp planning and corrective motions to interact, leading to a system where we can estimate uncertain object attributes, such as pose, and improve grasp stability simultaneously. We also presented a practical implementation of our framework utilizing particle filters. We showed that our method is able to find a stable grasp and simultaneously update the pose estimate of the object.

However, we were able to only show our implementation with a simple simulated environment consisting only of a simple object and of a simple manipulator. In the future we will try to apply the framework presented here to more complex manipulators and to more complex objects. Our goal is to first concentrate on simulation and building useable models of objects to use with our probabilistic framework.

REFERENCES

[1] C. Ferrari and J. Canny, "Planning optimal grasps," in *Robotics and Automation, 1992. Proceedings., 1992 IEEE International Conference on*, May 1992, pp. 2290 –2295 vol.3.

[2] K. Huebner, S. Ruthotto, and D. Kragic, "Minimum volume bounding box decomposition for shape approximation in robot grasping," in *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, May 2008, pp. 1628 –1633.

[3] C. Goldfeder, P. K. Allen, C. Lackner, and R. Pelossof, "Grasp Planning Via Decomposition Trees," in *IEEE International Conference on Robotics and Automation*, 2007, pp. 4679–4684.

[4] C. Goldfeder, M. Ciocarlie, H. Dang, and P. K. Allen, "The Columbia Grasp Database," in *IEEE International Conference on Robotics and Automation*, 2009.

[5] C. Goldfeder, M. Ciocarlie, J. Peretzman, H. Dang, and P. K. Allen, "Data-Driven Grasping with Partial Sensor Data," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009.

[6] R. Detry, D. Kraft, A. Buch, N. Kruger, and J. Piater, "Refining grasp affordance models by experience," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, May 2010, pp. 2287 –2293.

[7] C. Barck-Holst, M. Ralph, F. Holmar, and D. Kragic, "Learning grasping affordance using probabilistic and ontological approaches," in *Advanced Robotics, 2009. ICAR 2009. International Conference on*, June 2009, pp. 1 –6.

[8] D. Song, K. Huebner, V. Kyrki, and D. Kragic, "Learning task constraints for robot grasping using graphical models," in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, Oct. 2010, pp. 1579 –1585.

[9] R. Platt, "Learning grasp strategies composed of contact relative motions," in *Humanoid Robots, 2007 7th IEEE-RAS International Conference on*, Dec. 2007, pp. 49 –56.

[10] K. Goldberg and M. Mason, "Bayesian grasping," in *Robotics and Automation, 1990. Proceedings., 1990 IEEE International Conference on*, May 1990, pp. 1264 –1269 vol.2.

[11] V. Christopoulos and P. Schrater, "Handling shape and contact location uncertainty in grasping two-dimensional planar objects," in *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, Nov. 2007, pp. 1557 –1563.

[12] K. Hsiao, L. Kaelbling, and T. Lozano-Perez, "Task-driven tactile exploration," in *Proceedings of Robotics: Science and Systems*, Zaragoza, Spain, June 2010.

[13] Y. Bekiroglu, J. Laaksonen, J. A. Jorgensen, and V. Kyrki, "Learning grasp stability based on haptic data," in *Robotics: Science and Systems (RSS 2010) Workshop on Representations for Object Grasping and Manipulation in Single and Dual Arm Tasks*, 2010.

[14] C. Corcoran and R. Platt, "A measurement model for tracking hand-object state during dexterous manipulation," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, May 2010, pp. 4302 –4308.

[15] Y. Bekiroglu, J. Laaksonen, J. A. Jorgensen, V. Kyrki, and D. Kragic, "Assessing grasp stability based on learning and haptic data," Draft accepted to IEEE Transactions on Robotics.

[16] S. Thrun, W. Burgard, and D. Fox, *Probabilistic robotics*. MIT Press, 2005.