AN ABSTRACT OF THE THESIS OF

Alex Keith Goins for the degree of Master of Science in Mechanical Engineering presented on May 19, 2014.

Title:

Improving Robotic Grasping Performance Using Machine Learning Techniques

Abstract approved: ____

Ravi Balasubramanian

Robots are being utilized in ever more complex tasks and environments to help humans with difficult or dangerous tasks. However, robotic grasping is still in its infancy and is one of the limiting factors which prevent the deployment of robots in the home and other assisted living scenarios. Traditional methods for grasp planning use grasp metrics, which are numerical computations of the kinematic arrangement of the hand and object. However, they are insufficient alone for accounting for all of the variables involved in the grasping process shown by their poor performance when implemented on a robotic platform. We use grasp testing data, along with a machine learning algorithm, in order to learn the complex relationship among all of the grasp metrics so as to improve grasp prediction performance. We then evaluate the resulting machine algorithm to validate the results and compare them to the individual metrics and state of the art grasp planners.

Improving Robotic Grasping Performance Using Machine Learning Techniques

by

Alex Keith Goins

A THESIS

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Master of Science

Presented May 19, 2014
Commencement June 2014

Master of Science thesis of Alex Keith Goins presented on May 19, 2014.

APPROVED:

Major Professor, representing Mechanical Engineering

Head of the School of Mechanical, Industrial and Manufacturing Engineering

Dean of the Graduate School

I understand that my thesis will become part of the permanent collection of Oregon State University libraries. My signature below authorizes release of my thesis to any reader upon request.

Alex Keith Goins, Author

ACKNOWLEDGEMENTS

The author thanks everyone involved in this research and successful degree completion. My advisor, Dr. Ravi Balasubramanian, who helped with paper submissions and provided the opportunity to participate in the DARPA Robotics Challenge. My lab co-workers Ryan Carpenter and Zhifei Zhang who helped with the data processing and analysis. Also, Jordan Meader who helped with a lot of the data collection and testing involved in the research. Thanks to all of my friends, family, and church for all of their support and prayers, and thanks to God for making it all happen.

TABLE OF CONTENTS

Page

1 Introduction 2

2 Evaluating the Efficacy of Grasp Metrics for Utilization in a Gaussian Process Based Grasp Predictor 5

2.1 INTRODUCTION 6

2.2 BACKGROUND 8

2.2.1 Gaussian Process 9

2.3 EXPERIMENTAL METHODS 10

2.3.1 Grasp Metric Selection and Evaluation 11

2.3.2 Collection of the Grasp Sample Set 11

2.3.3 Evaluation of the Grasp Sample Set 13

2.3.4 Quantitative Evaluation of Grasp Metrics 16

2.3.5 Dimensionality Reduction Using Principal Component Analysis and Statistical Testing 17

2.3.6 Building a Gaussian Process-based Classifier for Grasp Quality Prediction 17

2.4 RESULTS 19

2.4.1 Discriminative Ability of Individual Grasp Metrics 19

2.4.2 Principal Component Analysis of the Grasp Sample Set . . . 20

2.4.3 Performance of the GP-based Classifiers 21

2.5 DISCUSSION 23

3 Improving Robotic Grasping Using a Gaussian Process Based Grasp Predictor 27

3.1 INTRODUCTION 28

3.2 BACKGROUND 31

3.2.1 Gaussian Process 32

3.3 EXPERIMENTAL METHODS 33

3.3.1 Grasp Metric Selection and Evaluation 33

3.3.2 Collection of the Grasp Sample Set 34

3.3.3 Evaluation of the Grasp Sample Set 37

3.3.4 Quantitative Evaluation of Grasp Metrics 39

TABLE OF CONTENTS (Continued)

Page

3.3.5 Dimensionality Reduction Using Principal Component Analysis and Statistical Testing 40
3.3.6 Building a Gaussian Process-based Classifier for Grasp Quality Prediction 41
3.3.7 Gaussian Process Based Grasp Improvement 42

3.4 RESULTS 45
3.4.1 Discriminative Ability of Individual Grasp Metrics 45
3.4.2 Principal Component Analysis of the Grasp Sample Set . . . 46
3.4.3 Performance of the GP-based Classifiers 47
3.4.4 Grasp Improvement Results 51

3.5 DISCUSSION 52

4 Conclusions 57

Bibliography 60

LIST OF FIGURES

Figure		Page
1.1	Example robotic hands (a) Sarah (b) Shadow (c) iRobot (d) Sandia.	3
2.1	Flow chart of experimental procedure	12
2.2	Nine everyday objects used for grasp generation	13
2.3	Shake test setup using WAM and marked reference location for object placement	13
2.4	Representative ROCs of several grasp metrics and GP classifier (mean±standard error over one hundred trials).	21
2.5	Visualization of a two dimensional projection of the six-dimensional surface that the GP creates to predict grasp quality. The "x" indicates good grasps and "o" indicates bad grasps from the grasp sample set. The filled area represents the "good" grasp region with success rate greater than 83% and a 10% FPR classification level.	23
2.6	Performance of GP and Energy with threshold as data set size increases	24
3.1	Flow chart of experimental procedure	35
3.2	Nine everyday objects used for grasp generation	36
3.3	Shake test setup using WAM and marked reference location for object placement	36
3.4	Plot showing PC1 and PC2 of GP surface with color levels for predicted score and shading based on uncertainty. Current grasp location with GP predicted value shown.	44
3.5	Representative ROCs of representative grasp metrics and GP classifier (mean±standard error over one hundred trials).	48
3.6	Visualization of a two dimensional projection of the a six-dimensional surface that the GP creates to predict grasp quality. The "x" indicates good grasps and "o" indicates bad grasps from the grasp sample set. The filled area represents the "good" grasp region with success rate greater than 83% and a 10% FPR classification level.	49

LIST OF FIGURES (Continued)

Figure Page

3.7 Performance of GP and Energy with threshold as data set size increases 50

3.8 Example grasp (a) before (b) after optimization. 51

LIST OF TABLES

Table Page

2.1 Grasp Metrics 10

2.2 End-Effector Shake Test Magnitudes 15

2.3 Individual Grasp Metric Evaluation 20

2.4 GP performance using PCA on different number of grasp metrics: TPR and AUC values 22

3.1 Grasp Metrics 34

3.2 End-Effector Shake Test Magnitudes 39

3.3 Individual Grasp Metric Evaluation 47

3.4 GP performance using PCA on different number of grasp metrics: TPR and AUC values 49

3.5 GP grasp improvement performance using fifteen different grasps . . 52

DEDICATION

Whatever you do, whether in word or deed, do all to the glory of God. To Him belongs all the glory, honor, and praise.

Improving Robotic Grasping Performance Using Machine Learning Techniques

Chapter 1 – Introduction

Robots are created to help humans. As the complexity and capabilities of robots have increased, they are being used in ever more diverse situations, with latest research focusing on the introduction of robots to the home environment and disaster scenarios. With these new environments comes new challenges. These challenges include interacting with human made objects and executing tasks normally reserved for humans. To complete these tasks, the robot needs to be dexterous and smart enough for handling the complexity and variety of tasks and objects.

In order to increase the dexterity and capabilities of service robots, several multi-degree of freedom robotic hands have been developed such as the Shadow hand, Sarah hand, Barrett Hand, iRobot, and Sandia hand (see Fig. 1.1). But using the hands is not straight forward since objects can be used for a variety of tasks and handled a variety of ways. Even humans, who are adept at grasping objects, have instances where they drop objects, despite being able to detect if a grasp is secure, loose, or slipping. Robots have yet to catch up to the capabilities of humans, and still have issues trying to incorporate a variety of information such as tactile, visual, and memory, in order to make a logical choice on the proper grasp to execute.

The traditional approach used to select a grasp choice has been to use grasp metrics. These are numerical computations which define certain parameters about

(a) (b) (c) (d)

Figure 1.1: Example robotic hands (a) Sarah (b) Shadow (c) iRobot (d) Sandia.

the grasp such as the amount of the object enclosed by the grasp, the minimum force the grasp can resist, and the alignment of the hand with the object. However, even when the object shape and location are known, these metrics still have poor performance during execution. Recent research has demonstrated that by including multiple metrics in the grasp planner, the grasp performance can be further improved, showing that the grasping domain is complex and multidimensional [1]. For the grasp planner to be robust even in the presence of sensing and perception errors, it needs to be able to combine the various metrics in order to account for complex interactions of metrics and grasp success rate.

While human intuition was used to create the individual metrics, manually combining the metrics to create a comprehensive and accurate grasp planner is infeasible due to the large number of variables involved. Due to the complexities in the grasping process, and the multitude of grasping metrics, we would like to have an automated method for both identifying the key qualities which define a stable grasp, and for incorporating them into an aggregate metric.

Chapter 2 explains the machine learning method which is used to examine some of the grasp metrics and their performance as well as the performance of the learned grasp planner. Chapter 3 shows some of the results when the learned grasp planner is used to plan new grasps or improve existing grasps. Chapter 4 provides final comments about the machine learning technique, how it can be improved, and how it can be utilized in a robotic application.

Chapter 2 – Evaluating the Efficacy of Grasp Metrics for Utilization in a Gaussian Process Based Grasp Predictor

With the goal of advancing the state of automatic robotic grasping, we present a novel approach that combines machine learning techniques and rigorous validation on a physical robotic platform in order to develop an algorithm that predicts the quality of a robotic grasp before execution. After collecting a large grasp sample set (522 grasps), we first conduct a thorough statistical analysis of the ability of grasp metrics that are commonly used in the robotics literature to discriminate between good and bad grasps. We then apply Principal Component Analysis and Gaussian Process algorithms on the discriminative grasp metrics to build a classifier that predicts grasp quality. The key findings are as follows: (i) several of the grasp metrics in the literature are weak predictors of grasp quality when implemented on a physical robotic platform; (ii) the Gaussian Process-based classifier significantly improves grasp prediction techniques by providing an absolute grasp quality prediction score from combining multiple grasp metrics. Specifically, the GP classifier showed a 66% percent improvement in the True Positive classification rate at a low False Positive rate of 5% when compared with classification based on thresholding of individual grasp metrics.

2.1 INTRODUCTION

Developing automatic algorithms that enable robots to grasp objects robustly is fundamentally important to the field of robotics, since it would pave the way for the use of robots in domestic and outdoor environments and not just in structured industrial settings. Recognizing this need, a variety of approaches based on physics force modeling [2, 3], machine-learning based techniques [4], and human-inspired grasping [5] have been developed for the automatic generation and prediction of robotic grasp success prior to execution. While significant progress has been made, recent results show that even the best of these autonomous grasp generation methods has a failure rate of 23% when implemented on a physical robot [1]. Such a high failure rate shows the complexity of the robotic grasping problem. This may be attributed to the difficulty in modeling non-linear effects such as contact friction, slip, compliance, and object movement due to disturbances during grasping.

In order to overcome the challenges of modeling these effects, researchers have developed metrics with the intention of capturing the properties that make a grasp secure and robust even in the presence of such uncertainty. For example, the physics-based grasp metrics "epsilon" and "volume" were developed using grasp wrench-space computations based on the magnitude and direction of generalized forces applied by the gripper to evaluate the grasp stability [3]. Another example is "grasp energy", which measures the average distance between potential gripper contact points and the object to determine the extent to which the object is enveloped by the hand [6].

Surveys of grasping literature [7, 8, 9, 10, 11, 12, 13] list as many as 24 grasp metrics which have been developed, mostly based on kinematic models (see Table 2.1 for a list of some of them). While some metrics, like finger spread, apply only to three finger grippers, the majority of metrics are applicable to other multifinger grippers [14, 15] and even the human hand [16]. However, each grasp metric individually captures only a small aspect of what makes a good grasp. As was found in [17, 18], slight variations in hand placement relative to the object can significantly change the metric value and grasp performance. In addition to variable sensitivity is the issue of correlation. The metrics are often calculated from dependent variables (such as finger contact location) which are based on the independent variables (such as hand pose, orientation, finger spread, and object type). Adjusting one independent variable could affect multiple dependent variables causing correlation among the various metrics.

In order to capture broader aspects of grasping and potentially improve grasp prediction performance, researchers have also developed aggregate grasps metrics that merge the evaluation signals from several individual metrics up to as many as nine metrics [19]. For example, weighted sums of epsilon, volume, and energy have been used simultaneously as a quality measure in the open source grasp planning and evaluation software GraspIt! [20] (also see [2, 21, 22, 23, 17] and Table 2.1 for other examples).

However, there are three key problems with the state of the art. First, most of the grasp metrics have been evaluated through simulation only [1], with limited validation of these metrics on physical robots [24, 19, 25]. Second, current methods have largely failed to account for the interactions or correlations between the grasp metrics [19, 16] which can lead to erroneous grasp quality prediction if unaccounted for. Third, most metrics only provide a measure of relative grasp quality, thus making it difficult to assess the grasp performance in absolute terms prior to execution. Ideally, we would like to know the probability of success for a grasp.

Given the state of grasp generation and grasp quality prediction algorithms, this paper uses machine learning techniques and rigorous validation on a physical robotic platform to develop an absolute grasp quality prediction algorithm. This paper's key contributions are: (i) An evaluation of individual grasp metrics commonly used in the robotics literature. (ii) The development of a data-driven approach to use a state-of-the-art classification algorithm to predict grasp quality and quantitatively compare its performance with prediction using current grasp metrics individually.

2.2 BACKGROUND

In this research, we use a Gaussian Process as our machine learning algorithm because it can model the non-linear relationship among the grasp metrics as well as create a non-linear decision surface between good and bad grasps. In addition, Gaussian Processes also provides the variance of its predictions, thereby providing a measure of the confidence or uncertainty regarding the prediction. This learning algorithm allows us to generate an estimate of the absolute grasp quality at a desired false positive rate, rather than a relative quality measure which current techniques provide. Other machine learning methods that can deal with the non-linear nature of the grasp space could also be used, but exploring all of them is not within the scope of this paper.

2.2.1 Gaussian Process

A Gaussian Process (GP) is a non-parametric model that can be used for supervised learning [26]. Specifically, given a set of n training samples $D = \{(x_1, y_1), \ldots, (x_n, y_n)\}$, where x_i is a feature vector and y_i is the output value, the algorithm learns a non-linear function $f(x)$ that generalizes from the training data in order to predict the output value y for some new data instance x.

GPs may be thought of as a generalization of a multivariate Gaussian distribution to infinite dimensions, such that any finite subset of the components of this infinite-dimensional vector is jointly Gaussian. Rather than just modeling a single function $f(x)$, a GP is a stochastic process that models a distribution over functions $f(x)$.

In our work, each data instance x_i is a grasp, which has k features that correspond to k grasp metrics used to represent it. Table 2.1 shows the $k = 12$ grasp metrics used in this paper. We use the GP to predict a continuous output value between 0 to 1 that represents the probability of the grasp being successful. We use an open-source GP package known as GPML[1] which was implemented in Matlab[2].

[1]http://www.gaussianprocess.org/gpml/code/matlab/doc/
[2]http://www.mathworks.com/

Table 2.1: Grasp Metrics

Metric	Description	Min	Max	Source
Contact Point Equilateralness[a]	Equilateralness of the triangle made by the contact points of the finger tips	0	1	[7]
Grasp Volume[a]	Volume of the triangular prism consisting of the finger tips and the palm	0	$669cm^3$	
Finger Extension[b]	Average finger flexion	0	1	
Finger Spread[a]	Amount of spread of the fingers	0	1	
Finger Limit[c]	Total flexion of all the fingers	0	1	
Parallel Symmetry[b]	Distance between center of mass of object and contact point parallel to the object principal axis	0	0.5	[27]
Perpendicular Symmetry[b]	Distance between center of mass of object and contact point perpendicular to the object principal axis	0	0.5	
Object Volume Enclosed[a]	Normalized volume of the object enclosed by the hand	0	1	
Skewness[c]	Alignment of the hand principal axis parallel to the object principal axis	0°	180°	[1]
Grasp Wrench (Epsilon)[a]	Minimum disturbance wrench that can be resisted	0	1	[3, 20]
Grasp Wrench Volume[a]	Volume of grasp wrench space	0	2^6	
Grasp Energy[b]	Distance of hand sample points to object	$-\infty$	∞	

[a]Larger = Better grasp; [b]Smaller = Better grasp; [c]Mid-range = Better grasp

2.3 EXPERIMENTAL METHODS

Our approach includes a combination of grasp generation and evaluation on a physical robotic platform and machine learning techniques to develop an algorithm for grasp quality prediction. An overview of the process used to develop the algorithm, including intermediate steps to perform dimensionality reduction on the data, is shown in Fig. 2.1.

2.3.1 Grasp Metric Selection and Evaluation

We selected twelve of the most common kinematic based metrics for evaluation and testing (see Table 2.1). Other metrics which depend on having force or contact sensors were not included in this study since our Barrett manipulator system does not have the capabilities to support them (see Fig. 2.3). While we did not analyze the other metrics, they can easily be included using the same procedure outlined below to increase the performance with grasping systems that have more capabilities.

2.3.2 Collection of the Grasp Sample Set

Twenty two human subjects were recruited to provide a total of 522 robotic grasp examples across nine everyday objects (see Fig. 2.2) using a simulation environment developed in OpenRAVE [28]. Each human subject commanded the position, orientation, finger spread, and grasp closure of the virtual BarrettHand [29] robotic hand, and had the option of viewing the grasp from several angles. Subjects used one of three common human-robot interfaces to grasp and pick up an object, a gamepad controller, a three-dimensional mouse, and the recently popular "interac-

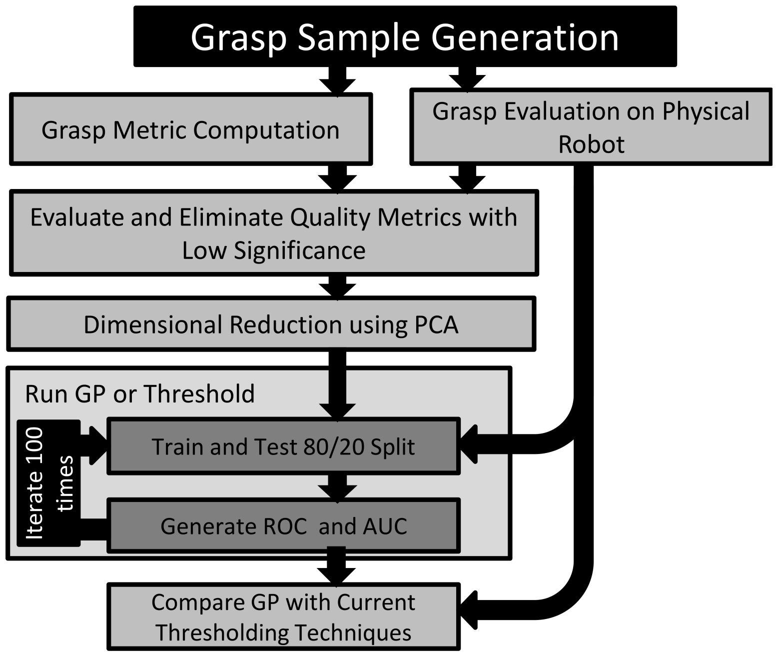


Figure 2.1: Flow chart of experimental procedure

tive marker" display [30]. Different human-robot interfaces were utilized to ensure that the grasp sample set was diverse and that one particular interface did not skew the grasp examples. Also, the robot hand's starting location was randomized between the objects so that a subject would not repeat the same grasp across multiple objects. When the user finished grasping the object and was satisfied with the final grasp, both the robot hand's posture relative to the object's coordinate frame and the computed metric scores were recorded. The human-subject experiment procedure was approved by Oregon State University's Human Subjects Division.

Figure 2.2: Nine everyday objects used for grasp generation

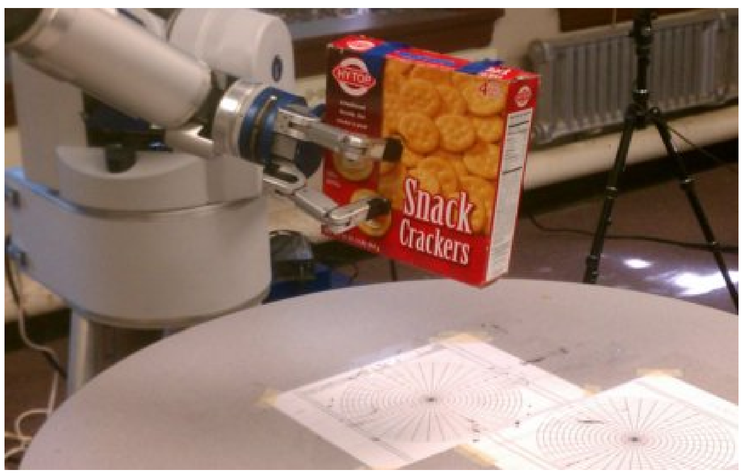


Figure 2.3: Shake test setup using WAM and marked reference location for object placement

2.3.3 Evaluation of the Grasp Sample Set

To determine the quality of the grasps provided by the human subjects, we tested the example grasps on a BarrettWAM and BarrettHand with standard rubber fingertips. This process was done in order to validate the predictive capability of the metrics, as well as provide ground truth data for the machine learning algorithm. Foam spacers were added during testing to the box and the soda can to prevent crushing but allow for minor flexing. The test procedure involved placing an object on a table at specific reference locations that were accurately measured and marked on the table (see Fig. 2.3). These locations had a series of evenly spaced radial and axial lines such that the object centroid could be placed accurately on the reference point in the correct position and orientation.

Extra care was taken to ensure that all fingers would make contact simultaneously and the final grasp would closely resemble simulation. This was performed by computing the pre-grasp finger posture for each grasp which would place the fingers at a uniform distance away from the object's surface but at the desired finger spread. This is important because if the fingers did not make contact simultaneously, they would push away the object resulting in grasps and metric values that do not match those planned in simulation. We did this to minimize such effects and ensure that the physical testing results were closely associated to the generated metric values. Thus, any noise in the grasping process was due to the precision in positioning of the object and robotic hand.

When grasping the object, the grasp controller used was the default controller provided by Barrett which closes all of the fingers simultaneously and stops each finger when a force or torque threshold is exceeded. After the robot hand closed on the object, the object was lifted and subjected to a series of rigorous disturbances. The disturbance was created by rotating the each of the three wrist joints sequentially from the current joint position to the furthest joint limit and then

Table 2.2: End-Effector Shake Test Magnitudes

Type	Peak	Mean
Angular Velocity (rad/s)	41.932	4.67
Linear Velocity (m/s)	87.05	0.4947
Angular Acceleration (rad/s^2)	40.76	3.49
Linear Acceleration (m/s^2)	86.94	0.44

back to its starting position. This was done so that the object would be subjected to forces in all of the gripper's primary axes and would experience translational as well as rotational forces. The acceleration and velocity magnitudes created by the disturbances are provided in Table 2.2 and are comparable or greater in magnitude to disturbances used in evaluation procedures in prior work [1].

Each grasp was tested ten times for a total of 5220 trials, and a binary score (success or fail) was recorded for each test. A specific grasp execution was considered a failure if the object fell or slipped and hit the table during the shake process. The success and failure binary scores from the ten trials were averaged to compute a mean performance score for each grasp. A grasp sample was labeled "good" if it had a performance score greater than or equal to 80%, and labeled "bad" otherwise. This 80% threshold was based on a realistic consideration of the state of the art in automatic robotic grasp generation, where one in four automatically generated grasps failed even in ideal laboratory conditions [1]. However, our algorithms could easily be extended to higher thresholds of performance.

2.3.4 Quantitative Evaluation of Grasp Metrics

The grasp metric data was normalized to a mean of 0 as

$$x_{(m,sph)} = \frac{x_{(m,n)} - \overline{x}_m}{\sigma_m}, \tag{2.1}$$

where for a given metric m and n data points, $x_{(m,sph)}$ is the normalized value for the observation $x_{(m,n)}$ with sample mean $\overline{x}_m$ and sample standard deviation σ_m. Normalizing data is important when using dimensionality reduction techniques such as PCA so that raw metric values with large ranges do not skew the analysis. Most importantly, normalization does not alter the ability of each metric to predict grasp quality.

A two-tailed t-test (p-value ≤ 0.05) was used to determine if the grasp metric's values were significantly different between good and bad grasps. A metric that showed a statistically significant difference between good and bad grasps was considered to be a good metric which will benefit a grasp planner. In addition, a simple classifier was built based on thresholding over the grasp metric value to determine if a grasp was good or bad. Specifically, if the result was greater or less than a desired threshold value, the grasp was considered a good grasp. These two methods help provide a baseline of how discriminative a grasp metric is. This simple classifier was compared with the GP based classifier (see section 2.3.6).

2.3.5 Dimensionality Reduction Using Principal Component Analysis and Statistical Testing

Even though multiple grasp metrics are utilized to describe the grasp, it is possible that the grasp sample data may have smaller intrinsic dimensionality due to (i) strong correlations between the grasp metrics and (ii) poor predictive ability of some grasp metrics. In order to deal with the correlated metrics, we use Principal Component Analysis (PCA) to perform a dimensionality reduction of our data by reducing the data to only a few dimensions in the full dimensional space [31].

First, those metrics that did not show statistical significance in the t-tests between good and bad grasps (see section 2.3.4) were eliminated. Then PCA was applied to all the remaining dimensions and the data variance captured by the different principal components was analyzed to determine if some principal components contributed more to the data variance than others.

2.3.6 Building a Gaussian Process-based Classifier for Grasp Quality Prediction

The high complexity of the grasp space makes it prohibitively difficult to manually develop a custom, composite metric, and is ideally suited for a machine learning algorithm to merge the information provided by each metric. In this work, we utilize a GP with a squared exponential covariance function with an Automatic Relevance Determination distance measure. Once the desired grasp metrics and principal components from PCA were selected (see section 2.3.5), a cross-validation technique using a randomized 80/20 split, where 80% of the grasp sample set was randomly chosen to train the GP classifier and the remaining 20% of the grasp sample set was used to test the classifier [31]. This process was repeated one hundred times and the average performance of the GP-based classifier using a threshold was recorded.

The GP-based classifier's prediction was used to create a receiver operating characteristic (ROC) curve to analyze performance trade-offs. ROC is a common tool used in the machine learning community for evaluating a classifier's performance [32]. The ROC curve's shape indicates how good the classifier is at keeping False Positive Rates (FPR) low and True Positive Rates (TPR) high. The TPR represents the success rate of correctly labeling successful grasps and FPR incorrectly labeling the unsuccessful grasps as successful. After one hundred iterations of cross-validation, the area under the curve (AUC) for all the iterations was averaged and the TPR at values of 5%, 10%, and 15% FPR were found. The AUC value represents the classifier's robustness by showing its probability to correctly classify a grasp. An AUC value of 1 indicates perfect performance, and an AUC value of 0.5 indicates random classification. To benchmark the GP classifier, we completed a similar ROC analysis for the simple classifiers based on thresholding on the grasp metrics (see section 2.3.4).

2.4 RESULTS

Of the 522 grasps in the dataset, 376 (72%) grasps were good (average success greater than 80%) and the remaining 146 were bad (28%).

2.4.1 Discriminative Ability of Individual Grasp Metrics

Table 2.3 and Fig. 2.4 provide a quantitative analysis of each grasp metric in terms of two aspects: (i) The statistical significance of each metric to discriminate between good and bad grasps based on t-tests, (ii) The performance of a simple classifier built by thresholding on each grasp metric. The table's rows are sorted based on increasing t-test p-values, which indicate that only six of the twelve grasp metrics can individually differentiate between good and bad grasps for this set of grasps at the $p = 0.05$ statistical significance level. In Fig. 2.4, the ROC curves (mean$\pm$standard error over one hundred trials) for the best classifiers built by thresholding individual grasp metrics and the best GP-based classifier are shown. It is evident that the GP-based classifier performs better than classification using individual grasp metrics in the regions of low FPR values. Furthermore, classification based on all individual grasp metrics, except energy, is only marginally better than random guessing as shown by the low AUC values and low TPR values in Table 2.3.

Table 2.3: Individual Grasp Metric Evaluation

Grasp metric	**t-test p-value**	**AUC value**	**TPR at 10% FPR**	**% Success at 10% FPR**
*Finger Extension	4.62e-13	0.65	0.24	70.6
*Skewness	2.78e-11	0.65	0.21	67.7
*Grasp Energy	1.67e-10	0.79	0.43	81.1
*Object Volume Enclosed	1.12e-8	0.65	0.24	70.6
*Parallel Symmetry	1.63e-6	0.62	0.14	58.3
*Perpendicular Symmetry	1.80e-6	0.56	0.15	60.0
*Point Arrangement	1.14e-5	0.57	0.13	56.5
*Finger Spread	2.56e-4	0.56	0.13	56.5
*Finger Limit	4.56e-3	0.61	0.12	54.5
Triangle Size	0.28	0.51	0.05	33.3
Epsilon	0.79	0.53	0.12	54.5
Grasp Wrench Volume	0.97	0.52	0.02	16.7

$*$p-value < 0.05, which indicates strong discriminative power

2.4.2 Principal Component Analysis of the Grasp Sample Set

The results from performing principal component analysis on all twelve dimensions of the grasping data showed that there is significant information in all of the components. Specifically, the cumulative variance explained by each additional principal component increases almost linearly (correlation to a 45° slope line is 0.97). However, comparing the AUC values for a GP classifier using varying numbers of principal components (PC), the AUC increased from 0.76 with one PC to 0.82 with four PCs, after which there was no further improvements for adding additional PCs. While the variance explained data implies that there is significant information in each PC, the AUC values from the GP shows that more than half

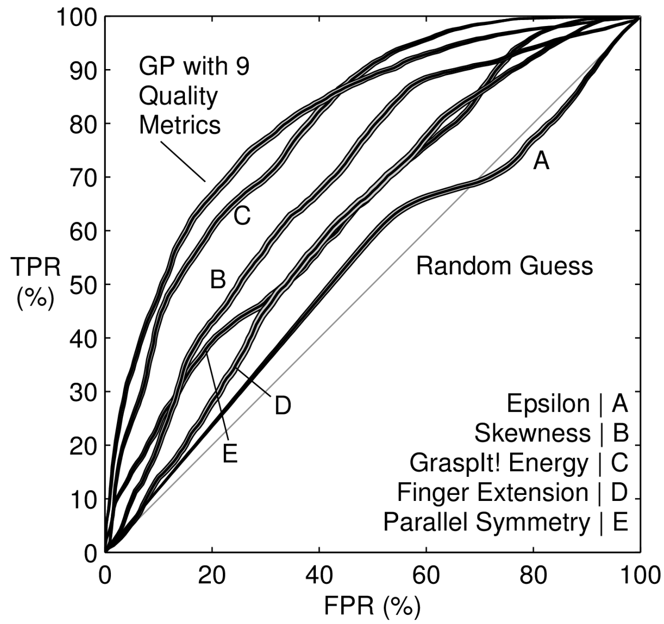


Figure 2.4: Representative ROCs of several grasp metrics and GP classifier (mean±standard error over one hundred trials).

of the PCs can be excluded without affecting the performance of the GP classifier. However, testing would need to be done on a case by case basis to confirm that some of the PCs could be excluded since the variance explained is insufficient alone to account for this.

2.4.3 Performance of the GP-based Classifiers

Table 2.4 shows the results from building and testing GP-based classifiers using all the grasp metrics and using all the principal components derived from subsets of the statistically significant grasp metrics. The results show that decreasing the number of grasp metrics used in the PCA process (but using all the principal

Table 2.4: GP performance using PCA on different number of grasp metrics: TPR and AUC values

Number of Grasp Metrics Used	**TPR**			**AUC**
	FPR =5%	**FPR =10%**	**FPR =15%**	
1	0.11	0.22	0.31	0.65
2	0.08	0.22	0.33	0.71
3	0.20	0.37	0.46	0.78
9	0.38	0.50	0.58	0.81
12	0.32	0.47	0.56	0.80

∗All scores are statistically different ($p < 0.05$)

components) based on the t-test performance significantly improves the TPR values of GP-based classifiers at a FPR of 5%. However, at the 10% and 15% FPR values, the data shows that using nine grasp metrics provides the best TPR values. Additionally, comparing Table 2.3 to the 10% FPR column of Table 2.4 shows the significantly improved performance of the GP classifier over simple thresholding of the individual metrics.

Figure 2.5 presents a visualization of a two-dimensional projection of the classification surface the GP creates for evaluating grasp quality. This particular GP is built using all principal components of the top six grasp metrics from Table 2.3. Despite the non-linearities, it is clear that the GP has been successful in finding a boundary that divides the good and bad grasp region. Fig. 2.6 shows how the performance of this classifier improves (measured in terms of AUC values) as the data set size increases. As expected, the GP-based classifier performs worse than thresholding using the energy metric for small datasets. However, when the grasp

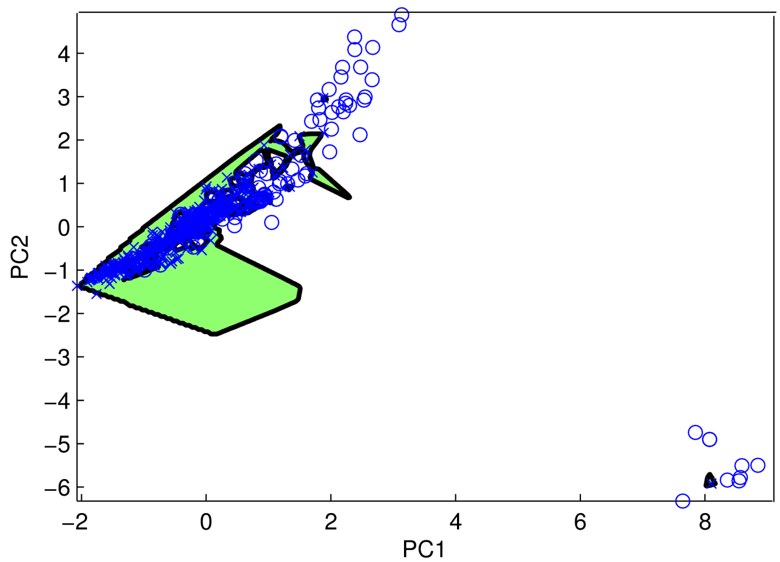


Figure 2.5: Visualization of a two dimensional projection of the six-dimensional surface that the GP creates to predict grasp quality. The "x" indicates good grasps and "o" indicates bad grasps from the grasp sample set. The filled area represents the "good" grasp region with success rate greater than 83% and a 10% FPR classification level.

sample set size goes beyond 300, the GP-based classifier performs better than energy-based thresholding.

2.5 DISCUSSION

Accurately predicting grasp quality is a challenging problem, given the significant amount of uncertainty in the grasping process and lack of clarity in which grasp metrics correctly predict grasp performance. Table 2.3 shows that many of the grasping metrics commonly used in the robotics literature are weak predictors of grasp quality. However, the t-test procedure proved to be a good method for deter-

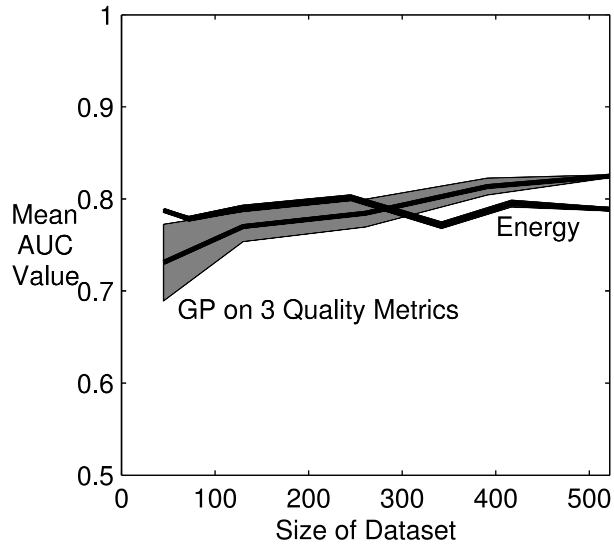


Figure 2.6: Performance of GP and Energy with threshold as data set size increases

mining which grasp metrics were important and may be used to build a classifier that combines the metrics to improve classification performance.

Using the grasp metrics in Table 2.1, the GP-based classifier (TPR=0.38) significantly improved over a classifier based on simple thresholding of individual grasp metrics (energy TPR=0.23) resulting in a 66% improvement in the TPR rate at an FPR of 5%. This was because the GP-based classifier non-linearly merged the signals from multiple metrics. The key finding was that the grasp metrics which have low discriminative ability only serve to introduce noise into the classifier and make it more difficult for GP to learn the grasp quality function. These metrics had low discriminative power due to the users' preference of power grasps over precision grasps. Some of the objects and grasps were such that the fingers wrapped around the object, but there was no palm contact when initially created in simulation.

This resulted in grasps which did not have force closure, and thus had small grasp wrench values and zero epsilon. However, when executed, these grasps performed very well as they were able to fully enclose some or all of the object. As such, the successful grasps had widely varying grasp wrench and epsilon scores, resulting in their low performance and exclusion from the GP. If different objects and grasps were selected, then these metrics could prove significant and be reintroduced into the GP.

As the number of data points used to build the classifier increases, the classification success rate of the GP classifier also increases (see Fig. 2.6). From the current data analysis, it is unclear if the GP classifier's performance has plateaued when using the full data set. However, it is clear from Fig. 2.5 that the current data set's spread can be improved, given the clustering of grasp samples in the $(-2 < \mathrm{PC}_1 < 2, -1 < \mathrm{PC}_2 < 4)$ range. More experiments are needed for exploring other regions of the grasp space.

Similar experiments have been performed but usually on smaller data sets. Specifically, ninety grasps were generated across four planar objects and tested a total of 920 times and were able to achieve an average prediction success rate of about 76% [19]. Another group tested thirteen novel 3-D objects across 150 trials and achieved a prediction success rate of 81% across all objects [25]. In our work, our experiment used 522 grasps on nine objects a total of 5220 trials and was able to achieve a high TPR at low FPR levels and an overall success rate of 88% (at 5% FPR). A key advantage of our work is the ability to select a desired FPR level for the prediction performance. However, given the complexity of the grasping problem, more grasp examples and validation over more platforms is needed to improve the grasp predictor's performance and to find regions of strong or weak performance.

One key advantage the GP-based classifier offers over the individual metrics is the significantly higher TPR at a low FPR values. This will significantly reduce the online computation time required by reducing the number of rejected good grasps. For example, the online grasp planner GraspIt! searches about seventy five grasps a second in order to provide about thirty valid grasps [1]. With GP's higher TPR, this number can be increased to forty valid grasps which can improve the grasp performance especially in constrained environments where typical grasps are not possible. Alternatively, the computation time could be reduced for the same number of candidate grasps, resulting in better performance for real-time robotics.

One limitation of this research is the lack of including the dynamics of the grasping process in the metric computation. While great care was taken to ensure that the object moved negligibly during the grasping process, additional development to include grasping dynamics in grasp quality prediction would further improve the results as well as open up a new field for making grasp predictions of flexible and compliant objects. Second, we used a robotic platform commonly used for both research [33] and development[3] to make the results broadly applicable. However, more testing is needed to transfer the results to other robotic platforms with differing capabilities.

[3] www.thearmrobot.com

Chapter 3 – Improving Robotic Grasping Using a Gaussian Process Based Grasp Predictor

With the goal of advancing the state of automatic robotic grasping, we present a novel approach that combines machine learning techniques and rigorous validation on a physical robotic platform in order to develop an algorithm that predicts the quality of a robotic grasp before execution. After collecting a large grasp sample set (522 grasps), we first conduct a thorough statistical analysis of the ability of grasp metrics that are commonly used in the robotics literature to discriminate between good and bad grasps. We then apply Principal Component Analysis and Gaussian Process algorithms on the discriminative grasp metrics to build a classifier that predicts grasp quality. The resulting classifier is then used to generate new grasps to validate its performance and compare the results to existing grasp planners. The key findings are as follows: (i) several of the grasp metrics in the literature are weak predictors of grasp quality when implemented on a physical robotic platform; (ii) the Gaussian Process-based classifier significantly improves grasp prediction techniques by providing an absolute grasp quality prediction score from combining multiple grasp metrics; (iii) The GP classifier can be used generate new grasps to improve bad grasp samples by performing a local search to find neighboring grasps which have improved contact points and higher success rate. Specifically, the GP classifier showed a 66% percent improvement in the True Positive classification rate at a low False Positive rate of 5% when compared with classification based on thresholding of individual grasp metrics.

3.1 INTRODUCTION

Developing automatic algorithms that enable robots to grasp objects robustly is fundamentally important to the field of robotics, since it would pave the way for the use of robots in domestic and outdoor environments and not just in structured industrial settings. Recognizing this need, a variety of approaches based on physics force modeling [2, 3], machine-learning based techniques [4], and human-inspired grasping [5] have been developed for the automatic generation and prediction of robotic grasp success prior to execution. While significant progress has been made, recent results show that even the best of these autonomous grasp generation methods has a failure rate of 23% when implemented on a physical robot [1]. Such a high failure rate shows the complexity of the robotic grasping problem. This may be attributed to the difficulty in modeling non-linear effects such as such as contact friction, slip, compliance, and object movement due to disturbances during grasping.

In order to overcome the challenges of modeling these effects, researchers have developed metrics with the intention of capturing the properties that make a grasp secure and robust even in the presence of such uncertainty. For example, the physics-based grasp metrics “epsilon” and “volume” were developed using grasp wrench-space computations based on the magnitude and direction of generalized forces applied by the gripper to evaluate the grasp stability [3]. Another example is "grasp energy", which measures the average distance between potential gripper contact points and the object to determine the extent to which the object is enveloped by the hand [6].

Surveys of grasping literature [7, 8, 9, 10, 11, 12, 13] list as many as 24 grasp metrics which have been developed, mostly based on kinematic models (see Table 2.1 for a list of some of them). While some metrics, like finger spread, apply only to three finger grippers, the majority of metrics are applicable to other multifinger grippers [14, 15] and even the human hand [16]. However, each grasp metric individually captures only a small aspect of what makes a good grasp. As was found in [17, 18], slight variations in hand placement relative to the object can significantly change the metric value and grasp performance. In addition to variable sensitivity is the issue of correlation. The metrics are often calculated from dependent variables (such as finger contact location) which are based on the independent variables (such as hand pose, orientation, finger spread, and object type). Adjusting one independent variable could affect multiple dependent variables causing correlation among the various metrics.

In order to capture broader aspects of grasping and potentially improve grasp prediction performance, researchers have also developed aggregate grasps metrics that merge the evaluation signals from several individual metrics up to as many as nine metrics [19]. For example, weighted sums of epsilon, volume, and energy have been used simultaneously as a quality measure in the open source grasp planning and evaluation software GraspIt! [20] (also see [2, 17, 21, 22, 23] and Table 2.1 for other examples).

However, there are three key problems with the state of the art. First, most of the grasp metrics have been evaluated through simulation only [1], with limited validation of these metrics on physical robots [19, 24, 25]. Second, current methods have largely failed to account for the interactions or correlations between the grasp metrics [16, 19] which can lead to erroneous grasp quality prediction if unaccounted for. Third, most metrics only provide a measure of relative grasp quality, thus making it difficult to assess the grasp performance in absolute terms prior to execution. Ideally, we would like to know the probability of success for a grasp.

Several methods for learning grasps have been tried including support vector machines (SVM), Bayesian networks (BN), and neural networks [34, 35, 36]. These methods learn off of a small set of the grasp space, usually limited to two or three constraints such as hand approach vector, finger joint configuration, eigengrasps, or contact point symmetry, in order to create a more robust grasp planner. The problem is that these approaches require information about each object and initial grasp examples. Generalization of grasp information to new objects is not always straightforward, and often requires new grasp examples and retraining of the algorithm.

Because the learning approach is still new and requires a lot of data, there is still active research in human grasping controls for robots [37, 38]. However, pure teleoperation can be a significant burden to the user if the task is sufficiently difficult. By increasing the amount of autonomy of the robot for difficult tasks, the user burden can be reduced and the success rate and time of completion improved [39]. One way that the level of autonomy can be increased, is to reduce the amount of human input to high level commands only. Since grasping directions are often confined based upon the object and the task required [35], the user can provide grasp direction or task information, while the robot decides the low level details of contact point location and grasp pose. Grasps chosen this way then can be optimized by the robot, in order to find a neighboring good grasp [40, 41].

Given the state of grasp generation and grasp quality prediction algorithms, this paper uses machine learning techniques and rigorous validation on a physical robotic platform to develop an absolute grasp quality prediction algorithm. This paper's key contributions are: (i) An evaluation of individual grasp metrics commonly used in the robotics literature. (ii) The development of a data-driven approach to use a state-of-the-art classification algorithm to predict grasp quality and quantitatively compare its performance with prediction using current grasp metrics individually. (iii) Use of the learned algorithm to optimize a grasp which is initially provided by the user in order to create a successful grasp once executed on the robot.

3.2 BACKGROUND

In this research, we use a Gaussian Process as our machine learning algorithm because it can model the non-linear relationship among the grasp metrics as well as create a non-linear decision surface between good and bad grasps. In addition, Gaussian Processes also provides the variance of its predictions, thereby providing a measure of the confidence or uncertainty regarding the prediction. This learning algorithm allows us to generate an estimate of the absolute grasp quality at a desired false positive rate, rather than a relative quality measure which current techniques provide. Other machine learning methods that can deal with the non-linear nature of the grasp space could also be used, but exploring all of them is not within the scope of this paper.

3.2.1 Gaussian Process

A Gaussian Process (GP) is a non-parametric model that can be used for supervised learning [26]. Specifically, given a set of n training samples $D = \{(x_1, y_1), \ldots, (x_n, y_n)\}$, where x_i is a feature vector and y_i is the output value, the algorithm learns a non-linear function $f(x)$ that generalizes from the training data in order to predict the output value y for some new data instance x.

GPs may be thought of as a generalization of a multivariate Gaussian distribution to infinite dimensions, such that any finite subset of the components of this infinite-dimensional vector is jointly Gaussian. Rather than just modeling a single function $f(x)$, a GP is a stochastic process that models a distribution over functions $f(x)$.

In our work, each data instance x_i is a grasp, which has k features that correspond to k grasp metrics used to represent it. Table 2.1 shows the $k = 12$ grasp metrics used in this paper. We use the GP to predict a continuous output value between 0 to 1 that represents the probability of the grasp being successful. We use an open-source GP package known as GPML[1] which was implemented in Matlab[2].

3.3 EXPERIMENTAL METHODS

Our approach includes a combination of grasp generation and evaluation on a physical robotic platform and machine learning techniques to develop an algorithm for grasp quality prediction. An overview of the process used to develop the algorithm, including intermediate steps to perform dimensionality reduction on the data, is shown in Fig. 3.1.

3.3.1 Grasp Metric Selection and Evaluation

We selected twelve of the most common kinematic based metrics for evaluation and testing (see Table 3.1). Other metrics which depend on having force or contact sensors were not included in this study since our Barrett manipulator system does not have the capabilities to support them (see Fig. 3.3). While we did not analyze the other metrics, they can easily be included using the same procedure outlined below to increase the performance with grasping systems that have more capabilities.

[1] http://www.gaussianprocess.org/gpml/code/matlab/doc/
[2] http://www.mathworks.com/

Table 3.1: Grasp Metrics

Metric	Description	Min	Max	Source
Contact Point Equilateralness[a]	Equilateralness of the triangle made by the contact points of the finger tips	0	1	[7]
Grasp Volume[a]	Volume of the triangular prism consisting of the finger tips and the palm	0	669cm^3	
Finger Extension[b]	Average finger flexion	0	1	
Finger Spread[a]	Amount of spread of the fingers	0	1	
Finger Limit[c]	Total flexion of all the fingers	0	1	
Parallel Symmetry[b]	Distance between center of mass of object and contact point parallel to the object principal axis	0	0.5	[27]
Perpendicular Symmetry[b]	Distance between center of mass of object and contact point perpendicular to the object principal axis	0	0.5	
Object Volume Enclosed[a]	Normalized volume of the object enclosed by the hand	0	1	
Skewness[c]	Alignment of the hand principal axis parallel to the object principal axis	0°	180°	[1]
Grasp Wrench (Epsilon)[a]	Minimum disturbance wrench that can be resisted	0	1	[3, 20]
Grasp Wrench Volume[a]	Volume of grasp wrench space	0	2^6	
Grasp Energy[b]	Distance of hand sample points to object	$-\infty$	∞	

[a]Larger = Better grasp; [b]Smaller = Better grasp; [c]Mid-range = Better grasp

3.3.2 Collection of the Grasp Sample Set

Twenty two human subjects were recruited to provide a total of 522 robotic grasp examples across nine everyday objects (see Fig. 3.2) using a simulation environment

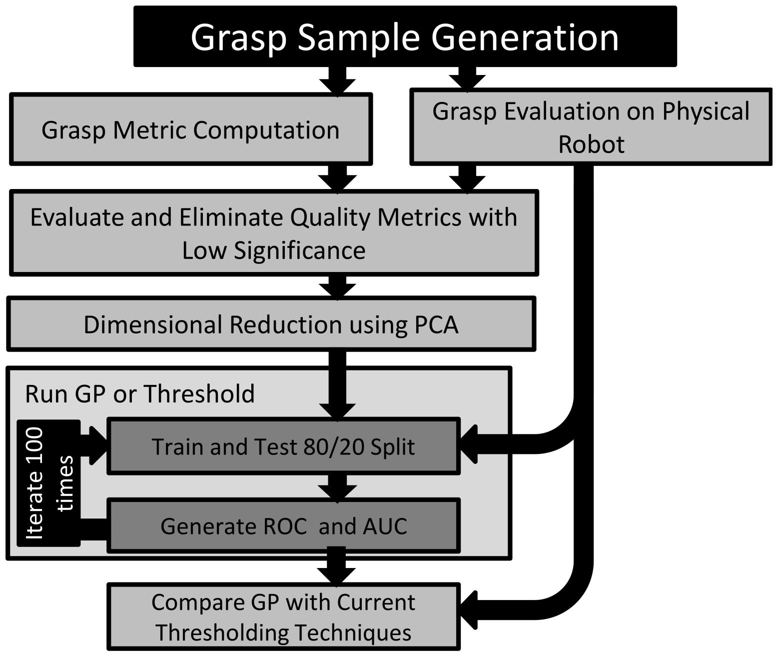


Figure 3.1: Flow chart of experimental procedure

developed in OpenRAVE [28]. Each human subject commanded the position, orientation, finger spread, and grasp closure of the virtual BarrettHand [29] robotic hand, and had the option of viewing the grasp from several angles. Subjects used one of three common human-robot interfaces to grasp and pick up an object, a gamepad controller, a three-dimensional mouse, and the recently popular "interactive marker" display [30]. Different human-robot interfaces were utilized to ensure that the grasp sample set was diverse and that one particular interface did not skew the grasp examples. Also, the robot hand's starting location was randomized between the objects so that a subject would not repeat the same grasp across multiple objects. When the user finished grasping the object and was satisfied with the final grasp, both the robot hand's posture relative to the object's coordinate frame

Figure 3.2: Nine everyday objects used for grasp generation

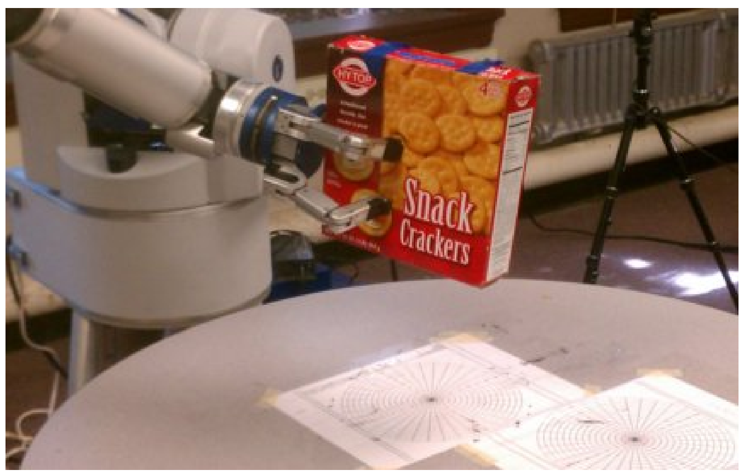


Figure 3.3: Shake test setup using WAM and marked reference location for object placement

and the computed metric scores were recorded. The human-subject experiment procedure was approved by Oregon State University's Human Subjects Division.

3.3.3 Evaluation of the Grasp Sample Set

To determine the quality of the grasps provided by the human subjects, we tested the example grasps on a BarrettWAM and BarrettHand with standard rubber fingertips. This process was done in order to validate the predictive capability of the metrics, as well as provide ground truth data for the machine learning algorithm. Foam spacers were added during testing to the box and the soda can to prevent crushing but allow for minor flexing. The test procedure involved placing an object on a table at specific reference locations that were accurately measured and marked on the table (see Fig. 3.3). These locations had a series of evenly spaced radial and axial lines such that the object centroid could be placed accurately on the reference point in the correct position and orientation.

Extra care was taken to ensure that all fingers would make contact simultaneously and the final grasp would closely resemble simulation. This was performed by computing the pre-grasp finger posture for each grasp which would place the fingers at a uniform distance away from the object's surface but at the desired finger spread. This is important because if the fingers did not make contact simultaneously, they would push away the object resulting in grasps and metric values that do not match those planned in simulation. We did this to minimize such effects and ensure that the physical testing results were closely associated to the generated metric values. Thus, any noise in the grasping process was due to the precision in positioning of the object and robotic hand.

When grasping the object, the grasp controller used was the default controller provided by Barrett which closes all of the fingers simultaneously and stops each finger when a force or torque threshold is exceeded. After the robot hand closed on the object, the object was lifted and subjected to a series of rigorous disturbances. The disturbance was created by rotating the each of the three wrist joints sequentially from the current joint position to the furthest joint limit and then back to its starting position. This was done so that the object would be subjected to forces in all of the gripper's primary axes and would experience translational as well as rotational forces. The acceleration and velocity magnitudes created by the disturbances are provided in Table 3.2 and are comparable or greater in magnitude to disturbances used in evaluation procedures in prior work [1].

Each grasp was tested ten times for a total of 5220 trials, and a binary score (success or fail) was recorded for each test. A specific grasp execution was considered a failure if the object fell or slipped and hit the table during the shake process. The success and failure binary scores from the ten trials were averaged to compute a mean performance score for each grasp. A grasp sample was labeled "good" if it had a performance score greater than or equal to 80%, and labeled "bad" otherwise. This 80% threshold was based on a realistic consideration of the state of the art in automatic robotic grasp generation, where one in four automatically generated grasps failed even in ideal laboratory conditions [1]. However, our algorithms could easily be extended to higher thresholds of performance.

Table 3.2: End-Effector Shake Test Magnitudes

Type	Peak	Mean
Angular Velocity (rad/s)	41.932	4.67
Linear Velocity (m/s)	87.05	0.4947
Angular Acceleration (rad/s^2)	40.76	3.49
Linear Acceleration (m/s^2)	86.94	0.44

3.3.4 Quantitative Evaluation of Grasp Metrics

The grasp metric data was "spherized" or rescaled to a mean of 0 as

$$x_{(m,sph)} = \frac{x_{(m,n)} - \overline{x}_m}{\sigma_m}, \tag{3.1}$$

where for a given metric m and n data points, $x_{(m,sph)}$ is the normalized value for the observation $x_{(m,n)}$ with sample mean $\overline{x}_m$ and sample standard deviation σ_m. Normalizing data is important when using dimensionality reduction techniques such as PCA so that raw metric values with large ranges do not skew the analysis. Most importantly, normalization does not alter the ability of each metric to predict grasp quality.

A two-tailed t-test (p-value ≤ 0.05) was used to determine if the grasp metric's values were significantly different between good and bad grasps. A metric that showed a statistically significant difference between good and bad grasps was considered to be a good metric which will benefit a grasp planner. In addition, a simple classifier was built based on thresholding over the grasp metric value to determine if a grasp was good or bad. Specifically, if the result was greater or less than a desired threshold value, the grasp was considered a good grasp. These two methods help provide a baseline of how discriminative a grasp metric is. This simple classifier was compared with the GP based classifier (see section 3.3.6).

3.3.5 Dimensionality Reduction Using Principal Component Analysis and Statistical Testing

Even though multiple grasp metrics are utilized to describe the grasp, it is possible that the grasp sample data may have smaller intrinsic dimensionality due to (i) strong correlations between the grasp metrics and (ii) poor predictive ability of some grasp metrics. In order to deal with the correlated metrics, we use Principal Component Analysis (PCA) to perform a dimensionality reduction of our data by reducing the data to only a few dimensions in the full dimensional space [31].

First, those metrics that did not show statistical significance in the t-tests between good and bad grasps (see section 3.3.4) were eliminated. Then PCA was applied to all the remaining dimensions and the data variance captured by the different principal components was analyzed to determine if some principal components contributed more to the data variance than others.

3.3.6 Building a Gaussian Process-based Classifier for Grasp Quality Prediction

The high complexity of the grasp space makes it prohibitively difficult to manually develop a custom, composite metric, and is ideally suited for a machine learning algorithm such as GP to merge the information provided by each metric. In this work, we utilize a GP with a squared exponential covariance function with an Automatic Relevance Determination distance measure. Once the desired grasp metrics and principal components from PCA were selected (see section 3.3.5), a cross-validation validation technique using an randomized 80/20 split, where 80% of the grasp sample set was randomly chosen to train the GP classifier and the remaining 20% of the grasp sample set was used to test the classifier [31]. This process was repeated one hundred times and the average performance of the GP-based classifier using a threshold was recorded.

The GP-based classifier's prediction was used to create a receiver operating characteristic (ROC) curve to analyze performance trade-offs. ROC is a common tool used in the machine learning community for evaluating a classifier's performance [32]. The ROC curve's shape indicates how good the classifier is at keeping False Positive Rates (FPR) low and True Positive Rates (TPR) high. The TPR represents the success rate of correctly labeling successful grasps and FPR incorrectly labeling the unsuccessful grasps as successful. After one hundred iterations of cross-validation, the area under the curve (AUC) for all the iterations was averaged and the TPR at values of 5%, 10%, and 15% FPR were found. The AUC value represents the classifier's robustness by showing its probability to correctly classify a grasp. An AUC value of 1 indicates perfect performance, and an AUC value of 0.5 indicates random classification. To benchmark the GP classifier, we completed a similar ROC analysis for the simple classifiers based on thresholding on the grasp metrics (see section 3.3.4).

3.3.7 Gaussian Process Based Grasp Improvement

Once the Gaussian process based classifier is generated, it can then be used to evaluate the performance of new grasps. If the grasp predicted success rate is high, it can be executed. If low, then either a new grasp must be generated, or the provided grasp can be improved. This section gives more detail about how the GP classifier is used to guide a selected grasp into a more desirable configuration which has higher predicted and actual performance.

3.3.7.1 Calculation of Grasp Score

First, a seed grasp must be generated, either from human example, or based upon environment or kinematic constraints. The metric scores of the grasp are calculated and transformed into PC space and the performance of the grasp is then predicted by the GP classifier. If the grasp score is low, then it can be improved by slightly altering the grasp so that the calculated metric values place the grasp in a nearby region in the GP PCA space which has better performance.

As previously mentioned in Section 3.1, the metrics are calculated from several dependent variables. Because of this, slight variations in the end-effector location can have large effects on the calculated metric values. This means that there is no direct mapping between the end-effector location and the metric scores, which prevents direct back-computation from the PCA space to the end-effector space. In order to find a grasp which matches the desired metric values, a forward search must be done where the end-effector is perturbed first, then the metric values calculated to determine the GP prediction score.

After the GP score is calculated, the GP surface plot with the resulting grasp score are displayed to the user using Matplotlib [42] (see Fig. 3.4). For more complex surfaces, the plot can be used to select a region of better performance to drive the grasp search algorithm towards. This approach helps to avoid the problem that a gradient search has, namely that of finding local maximums which are suboptimal. If the surface is simple, then a grasp search solely based on GP predicted score is adequate enough.

3.3.7.2 Grasp Search Protocol

In order to improve the grasp performance, an optimization algorithm is used to perturb the grasp to improve the GP prediction score. We then repeat this process until the grasp cannot be improved any further or until the desired grasp performance is obtained (see Algorithm 1).

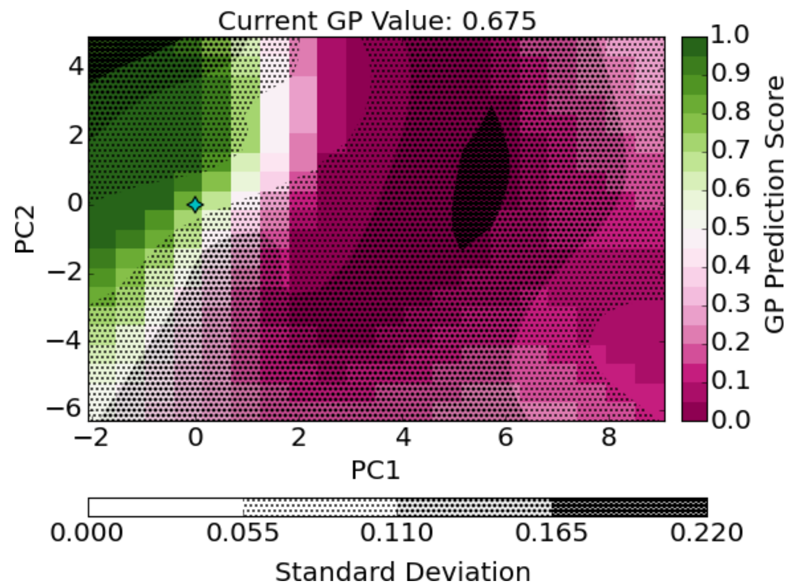


Figure 3.4: Plot showing PC1 and PC2 of GP surface with color levels for predicted score and shading based on uncertainty. Current grasp location with GP predicted value shown.

Algorithm 1 Grasp Search Protocol

---

m = seed grasp(s)

S_m = GP predicted success rate for all m

Goal = GP target

while S_{m_1} < Goal **do**

    for all i in m **do**

        for j in (1,2,...,20) **do**

            $C_n = m_i + (\Delta pose, \Delta spread)$

        end for

    end for

    S_n = GP success rate for all C_n

    m = best 4 grasps

    S_m = GP success rate for all m

end while

First, a seed grasp must be generated, either from human example, or based upon environment or kinematic constraints. For each seed grasp, twenty new candidate grasps are generated by adding random offsets to the translation, orientation, and finger spread of the end-effector. Next, the new performance metrics are calculated for each candidate grasp, and the grasps are ranked based upon predicted performance and distance to the target goal. If the goal has been reached, then the optimization algorithm finishes and returns the best performing grasp. If the target is not reached, then the top four candidate grasps are then used as the seed grasps for the next round of optimization. Since there are now four seed grasps, there will be a total of eighty candidate grasps for the next round of optimization testing. Once the optimization routine is finished, the final grasp pose is then executed on the robot in order to compare the predicted success rate to the actual success rate.

3.4 RESULTS

Of the 522 grasps in the dataset, 376 (72%) grasps were good (average success greater than 80%) and the remaining 146 were bad (28%).

3.4.1 Discriminative Ability of Individual Grasp Metrics

Table 3.3 and Fig. 3.5 provide a quantitative analysis of each grasp metric in terms of two aspects: (i) The statistical significance of each metric to discriminate between good and bad grasps based on t-tests, (ii) The performance of a simple classifier built by thresholding on each grasp metric. The table's rows are sorted based on increasing t-test p-values, which indicate that only six of the twelve grasp metrics can individually differentiate between good and bad grasps for this set of grasps at the $p = 0.05$ statistical significance level. In Fig. 3.5, the ROC curves (mean$\pm$standard error over one hundred trials) for the best classifiers built by thresholding individual grasp metrics and the best GP-based classifier are shown. It is evident that the GP-based classifier performs better than classification using individual grasp metrics in the regions of low FPR values. Furthermore, classification based on all individual grasp metrics, except energy, is only marginally better than random guessing as shown by the low AUC values and low TPR values in Table 3.3.

3.4.2 Principal Component Analysis of the Grasp Sample Set

The results from performing principal component analysis on all twelve dimensions grasping data showed that there is significant data in all of the components. Specifically, the cumulative variance explained by each additional principal component increases almost linearly (correlation to a 45° slope line is 0.97). However, comparing the AUC values for a GP classifier using varying numbers of principal components (PC), the AUC increased from 0.76 with one PC to 0.82 with four PCs, after which there was no further improvements for adding additional PCs. While the variance explained data implies that there is significant data in each

Table 3.3: Individual Grasp Metric Evaluation

Grasp metric	**t-test p-value**	**AUC value**	**TPR at 10% FPR**	**% Success at 10% FPR**
*Finger Extension	4.62e-13	0.65	0.24	70.6
*Skewness	2.78e-11	0.65	0.21	67.7
*Grasp Energy	1.67e-10	0.79	0.43	81.1
*Object Volume Enclosed	1.12e-8	0.65	0.24	70.6
*Parallel Symmetry	1.63e-6	0.62	0.14	58.3
*Perpendicular Symmetry	1.80e-6	0.56	0.15	60.0
*Point Arrangement	1.14e-5	0.57	0.13	56.5
*Finger Spread	2.56e-4	0.56	0.13	56.5
*Finger Limit	4.56e-3	0.61	0.12	54.5
Triangle Size	0.28	0.51	0.05	33.3
Epsilon	0.79	0.53	0.12	54.5
Grasp Wrench Volume	0.97	0.52	0.02	16.7

$*$p-value < 0.05, which indicates strong discriminative power

PC, the AUC values from the GP shows that more than half of the PCs can be excluded without affecting the performance of the GP classifier. However, testing would need to be done on a case by case basis to confirm that some of the PCs could be excluded since the variance explained is insufficient alone to account for this.

3.4.3 Performance of the GP-based Classifiers

Table 3.4 shows the results from building and testing GP-based classifiers using all the grasp metrics and using all the principal components derived from subsets of the statistically significant grasp metrics. The results show that decreasing

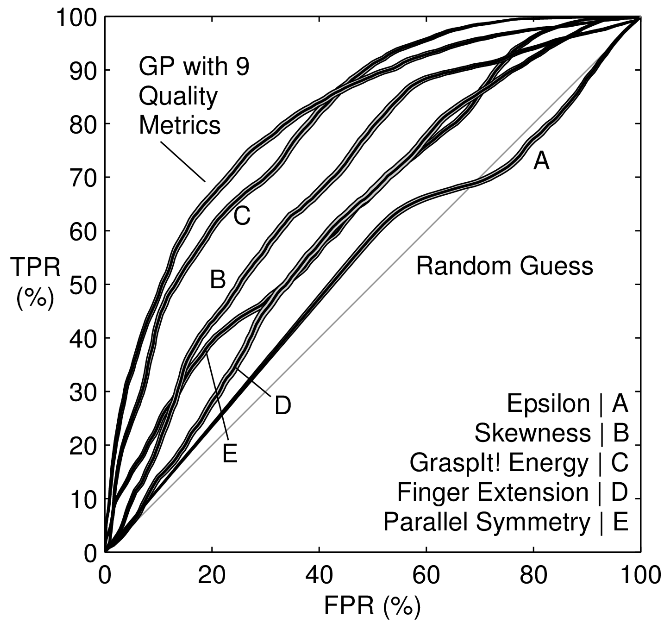


Figure 3.5: Representative ROCs of representative grasp metrics and GP classifier (mean±standard error over one hundred trials).

the number of grasp metrics used in the PCA process (but using all the principal components) based on the t-test performance significantly improves the TPR values of GP-based classifiers at a FPR of 5%. However, at the 10% and 15% FPR values, the data shows that using nine grasp metrics provides the best TPR values. Additionally, comparing Table 2.3 to the 10% FPR column of Table 3.4 shows the significantly improved performance of the GP classifier over simple thresholding of the individual metrics. Furthermore, at the 5% FPR level, the GP classifier has a true positive rate of 38%, which means that 38 out of 43 grasps will be successful, for an overall success rate of 88%. This is better than other similar work which had an overall success rate of 81% [25].

Figure 3.6 presents a visualization of a two-dimensional projection of the clas-

Table 3.4: GP performance using PCA on different number of grasp metrics: TPR and AUC values

Number of Grasp Metrics Used	**TPR**			**AUC**
	FPR =5%	**FPR =10%**	**FPR =15%**	
1	0.11	0.22	0.31	0.65
2	0.08	0.22	0.33	0.71
3	0.20	0.37	0.46	0.78
9	0.38	0.50	0.58	0.81
12	0.32	0.47	0.56	0.80

∗All scores are statistically different ($p < 0.05$)

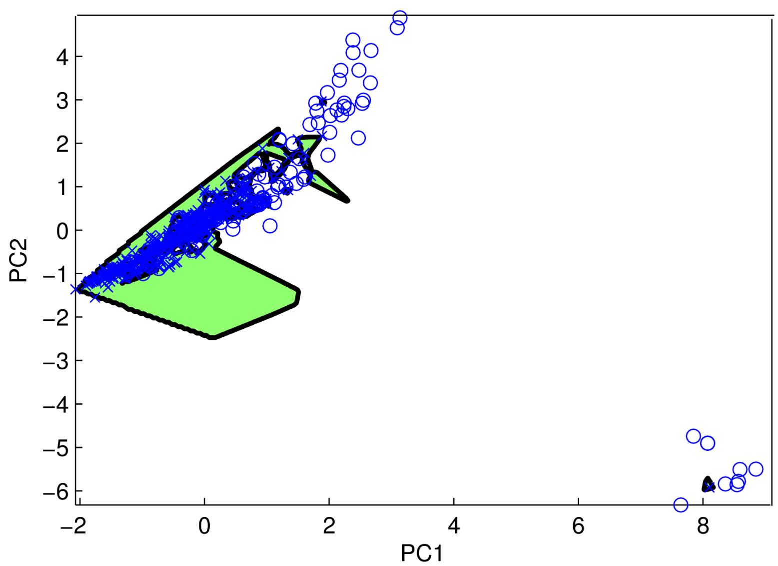


Figure 3.6: Visualization of a two dimensional projection of the a six-dimensional surface that the GP creates to predict grasp quality. The "x" indicates good grasps and "o" indicates bad grasps from the grasp sample set. The filled area represents the "good" grasp region with success rate greater than 83% and a 10% FPR classification level.

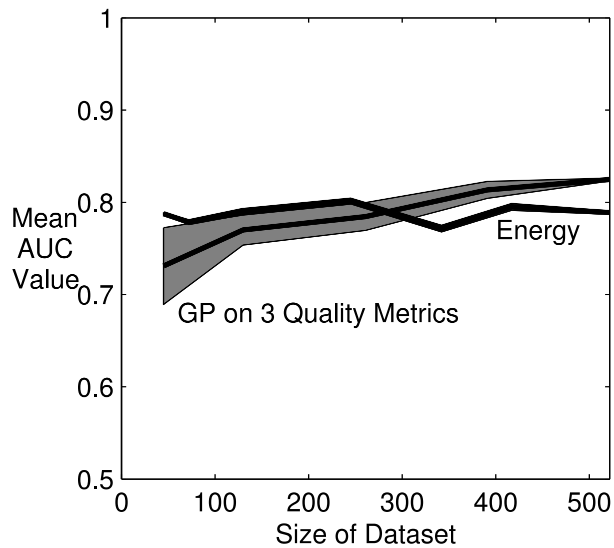
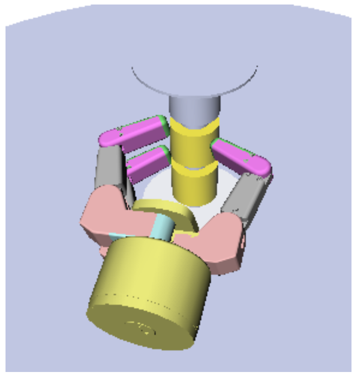
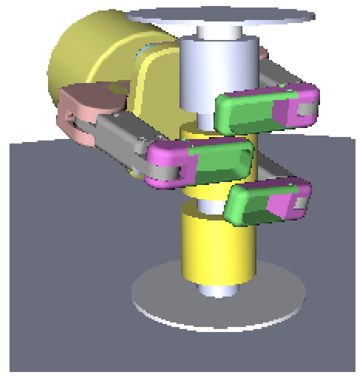


Figure 3.7: Performance of GP and Energy with threshold as data set size increases

sification surface the GP creates for evaluating grasp quality. This particular GP is built using all principal components of the top six grasp metrics from Table 3.3. Despite the non-linearities, it is clear that the GP has been successful in finding a boundary that divides the good and bad grasp region. Fig. 3.7 shows how the performance of this classifier improves (measured in terms of AUC values) as the data set size increases. As expected, the GP-based classifier performs worse than thresholding using the energy metric for small datasets. However, when the grasp sample set size goes beyond 300, the GP-based classifier performs better than energy-based thresholding.



(a)



(b)

Figure 3.8: Example grasp (a) before (b) after optimization.

3.4.4 Grasp Improvement Results

Using fifteen grasps which had predicted success rates below the 80% threshold, we used the GP classifier previously constructed to perform a search in an attempt to improve the predicted success rate. After searching, the final grasps were tested and the success rates and predicted scores were compared for the grasps both before and after improvement. From the comparison, there was a 50% improvement in the predicted success rate and a 40% improvement in the overall success rate (see Table 3.5). Of the fifteen grasps tested, five had an improved GP score higher than the 80% threshold which was determined to be a successful grasp. Of these five, all were ultimately successful when executed on the robot showing that the GP classifier can not only correctly classify, but can be used to improve grasp performance from a previous bad seed grasp.

Table 3.5: GP grasp improvement performance using fifteen different grasps

Object	**GP Score**		**Success Rate**	
	Before	**After**	**Before**	**After**
soap bottle	0.56	1	3	10
soda	0.57	1	4	10
soda	0.78	1	0	10
soap bottle	0.52	0.94	0	10
cup	0.53	0.84	0	10
spool	0.46	0.75	0	10
bottle	0.63	0.75	0	10
bottle	0.54	0.75	2	0
cd case	0.55	0.62	2	0
pitcher	0.36	0.62	0	1
box	0.31	0.62	0	0
remote	0.53	0.54	4	2
remote	0.53	0.54	0	2
remote	0.41	0.54	0	5
remote	0.41	0.54	0	0
average	0.47	0.74	1	5.33

3.5 DISCUSSION

Accurately predicting grasp quality is a challenging problem, given the significant amount of uncertainty in the grasping process and lack of clarity in which grasp metrics correctly predict grasp performance. Table 3.3 shows that many of the grasping metrics commonly used in the robotics literature are weak predictors of grasp quality. However, the t-test procedure proved to be a good method for determining which grasp metrics were important and may be used to build a classifier that combines the metrics to improve classification performance.

Using the grasp metrics in Table 2.1, the GP-based classifier (TPR=0.38) significantly improved over a classifier based on simple thresholding of individual grasp metrics (energy TPR=0.23) resulting in a 66% improvement in the TPR rate at an FPR of 5%. This was because the GP-based classifier non-linearly merged the signals from multiple metrics. The key finding was that the grasp metrics which have low discriminative ability only serve to introduce noise into the classifier and make it more difficult for GP to learn the grasp quality function. These metrics had low discriminative power due to the users' preference of power grasps over precision grasps. Some of the objects and grasps were such that the fingers wrapped around the object, but there was no palm contact when initially created in simulation. This resulted in grasps which did not have force closure, and thus had small grasp wrench values and zero epsilon. However, when executed, these grasps performed very well as they were able to fully enclose some or all of the object. As such, the successful grasps had widely varying grasp wrench and epsilon scores, resulting in their low performance and exclusion from the GP. If different objects and grasps were selected, then these metrics could prove significant and be reintroduced into the GP.

As the number of data points used to build the classifier increases, the classification success rate of the GP classifier also increases (see Fig. 3.7). From the current data analysis, it is unclear if the GP classifier's performance has plateaued when using the full data set. However, it is clear from Fig. 3.6 that the current data set's spread can be improved, given the clustering of grasp samples in the $(-2 < \mathrm{PC}_1 < 2, -1 < \mathrm{PC}_2 < 4)$ range More experiments are needed for exploring other regions of the grasp space.

Similar experiments have been performed but usually on smaller data sets. Specifically, ninety grasps were generated across four planar objects and tested a total of 920 times and were able to achieve an average prediction success rate of about 76% [19]. Another group tested thirteen novel 3-D objects across 150 trials and achieved a prediction success rate of 81% across all objects [25]. In our work, our experiment used 522 grasps on nine objects a total of 5220 trials and was able to achieve a high TPR at low FPR levels and an overall success rate of 88% (at 5% FPR). A key advantage of our work is the ability to select a desired FPR level for the prediction performance. However, given the complexity of the grasping problem, more grasp examples and validation over more platforms is needed to improve the grasp predictor's performance and to find regions of strong or weak performance.

While the GP classifier shows promise to improve bad grasps, only one in three grasps were able to be improved. The issue is not necessarily with the GP (since the improved grasps had high success rate), but the search algorithm which was insufficient for finding a better grasp. One reason why the grasps could not be improved may be due to the constraints in the end-effector space which would prevent reaching a target in the grasp space. For example, because of its low profile, the remote was restricted to precision type grasps and had a low number of successful grasps. If the grasp space preferred power grasps, then the remote would have a reduced number of options to achieve a higher performance. Additionally, the algorithm may have difficulties improving the grasp if neighboring grasps have similar poor performance [41]. Because of these issues, some grasps cannot be improved without drastically altering the grasp, while some objects are inherently difficult to grasp because of their size or shape.

Because the metrics were designed to be easily generalized to new objects, the learned GP is in essence a meta-metric which is also well suited for generalization. This means that the learned GP can be easily transferred to new objects without the need for more grasp examples or retraining. However, if the object or robotic hand were to change significantly, such as the contact forces or friction increasing, then this will result in grasps which reside in a different location in the grasp search space where grasps would have higher grasp wrench volume and higher success rate. This does not invalidate previous data, but simply would require more testing if the new region of the grasp space has not been thoroughly explored before. The challenge with this method is that an accurate model of the object must be available for planning, or else the calculation of the metrics will be incorrect. Incomplete or partial object information would give erroneous data and result in incorrect prediction scoress.

One key advantage the GP-based classifier offers over the individual metrics is the significantly higher TPR at a low FPR values. This will significantly reduce the online computation time required by reducing the number of rejected good grasps. For example, the online grasp planner GraspIt! searches about seventy five grasps a second in order to provide about thirty valid grasps [1]. With GP's higher TPR, this number can be increased to forty valid grasps which can improve the grasp performance especially in constrained environments where typical grasps are not possible. Alternatively, the computation time could be reduced for the same number of candidate grasps, resulting in better performance for real-time robotics.

One limitation of this research is the lack of including the dynamics of the grasping process in the metric computation. While great care was taken to ensure that the object moved negligibly during the grasping process, additional development to adaptively adjust finger contact points and forces according to object motion and to include grasping dynamics in grasp quality prediction would further improve the results as well as open up a new field for making grasp predictions of flexible and compliant objects [36, 43, 44]. Second, we used a robotic platform commonly used for both research [33] and development[3] to make the results broadly applicable. However, more testing is needed to transfer the results to other robotic platforms with differing capabilities.

[3] www.thearmrobot.com

Chapter 4 – Conclusions

In Chapter 2, we showed that the individual grasp metrics had poor classification performance and that the best grasp metric, energy, had a 23% TPR at 5% FPR for an overall success rate of 82%. In previous work, two grasp metrics, energy and skewness, were combined to increase the grasp classification performance [1]. Since manually combining the metrics across the whole grasp space can be intractable, we implemented a machine learning method for automatically identifying the significant metrics to increase grasp prediction performance. Using this machine learning approach, we were able to create a classifier with 38% TPR at 5% FPR, an overall success rate of 88%, which is a 6% improvement over using individual metrics alone. Because the machine learning algorithm is able to learn the complex dependencies between the metrics, it is able to outperform the classification performance of the individual metrics.

Furthermore, we were able to provide a correlation between the metric and GP score, and that of the grasp success rate, providing for an absolute measure for predicting grasp success rate. Previously, the grasp success rate was unknown until the grasp is executed on the robot. Using information from previous experiments, this method is able to provide the user with a success prediction rate to aid in the decision making process when selecting grasps to execute.

In Chapter 3, we implemented a grasp search algorithm which is able to use the learned machine algorithm to create grasps with a high success rate. We were able to achieve an average increase in predicted grasp performance of 50%. Testing of some of the improved grasps showed that we had a 40% increase in success rate and a 100% TPR at a 10% FPR level for the given data set. While some of the grasps could not be improved, there are two plausible reasons for this. First, it could be due to the grasp space, that neighboring grasps have poor performance or that the amount of grasp examples tested in this region are low. This would cause the search algorithm to turn up similarly poor performing grasps, or grasps which have very high uncertainty. Secondly, the grasp search algorithm may not be optimized so as to be able to search intelligently to find an optimized grasp. It may have the issue of getting stuck in a locally optimal grasp which does not meet the global criteria of grasp successfulness. In the first case, more data would need to be collected in this region of high uncertainty in order to increase the performance of the classifier. However, in the second case, the search algorithm can be modified so as to increase the space that it searches and have an intelligent, guided approach, so as to get out of locally optimal grasps. It is, however, possible for an improved grasp to not be found due to environment clutter or joint constraints. This would then necessitate other fixes, such as moving the robot to a better standing location, or introducing intermediate steps, whereby the robot moves neighboring objects to make a solid grasp available.

Previous robotic setups required the environment to be highly structured with objects designed so that they are easy to grasp, and robotic hands designed to pick up specific objects. For more unstructured environments, grasping and manipulation tasks are performed by humans teleoperating the robotic gripper into position. With this new approach, we hope to increase the level of autonomy of current robotic systems by creating a grasp planner which is able to successfully plan grasps without the need for human intervention. With this increased level of autonomy, the robot can be employed in a wider range of tasks and will be better equipped to help humans in assisted living conditions, or perform crucial tasks in disaster areas and locations hazardous to humans.

Bibliography

[1] R. Balasubramanian, L. Xu, P. D. Brook, J. R. Smith, and Y. Matsuoka, "Physical human interactive guidance: Identifying grasping principles from human-planned grasps," *IEEE Transactions on Robotics*, vol. 28, no. 4, pp. 899 – 910, 2012.

[2] B. Kehoe, D. Berenson, and K. Goldberg, "Toward cloud-based grasping with uncertainty in shape: Estimating lower bounds on achieving force closure with zero-slip push grasps," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 576 – 583, 2012.

[3] C. Ferrari and J. Canny, "Planning optimal grasps," *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 3, pp. 2290 – 2295, 1992.

[4] R. Pelossof, A. Miller, P. Allen, and T. Jebara, "An svm learning approach to robotic grasping," *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2004, no. 4, pp. 3512 – 3518, 2004.

[5] D. R. Faria, R. Martins, J. Lobo, and J. Dias, "Extracting data from human manipulation of objects towards improving autonomous robotic grasping," *Robotics and Autonomous Systems*, vol. 60, no. 3, pp. 396 – 410, 2012.

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

[7] E. Chinellato, A. Morales, R. B. Fisher, and A. P. del Pobil, "Visual quality measures for characterizing planar robot grasps," *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, vol. 35, no. 1, pp. 30 – 41, 2005.

[8] Q. Lin, J. W. Burdick, and E. Rimon, "Stiffness-based quality measure for compliant grasps and fixtures," *IEEE Trans. Rob.*, vol. 16, no. 6, pp. 675 – 688, 2000.

[9] R. Suárez, J. Cornellà, and M. R. Garzón, *Grasp quality measures.* Institut dÓrganització i Control de Sistemes Industrials, 2006.

[10] K. B. Shimoga, "Robot grasp synthesis algorithms: A survey," *The International Journal of Robotics Research*, vol. 15, no. 3, pp. 230–266, 1996.

[11] S. El-Khoury, A. Sahbani, and P. Bidaud, "3d objects grasps synthesis: A survey," *13th World Congress in Mechanism and Machine Science*, pp. 573 – 583, 2011.

[12] E. Lopez-Damian, D. Sidobre, and R. Alami, "A grasp planner based on inertial properties," *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2005, pp. 754 – 759, 2005.

[13] J. Bohg, A. Morales, T. Asfour, and D. Kragic, "Data-driven grasp synthesis—a survey," *IEEE Transactions on Robotics*, vol. PP, no. 99, pp. 1–21, 2013.

[14] B. Bounab, D. Sidobre, and A. Zaatri, "Central axis approach for computing n-finger force-closure grasps," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 1169 – 1174, 2008.

[15] S. El-Khoury and A. Sahbani, "On computing robust n-finger force-closure grasps of 3d objects," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 2480 – 2486, 2009.

[16] B. Leon, J. L. Sancho-Bru, N. J. Jarque-Bou, A. Morales, and M. A. Roa, "Evaluation of human prehension using grasp quality measures," *International Journal of Advanced Robotic Systems*, vol. 9, 2012.

[17] E. Chinellato, R. B. Fisher, A. Morales, and A. P. Del Pobil, "Ranking planar grasp configurations for a three-finger hand," *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 1, pp. 1133 – 1138, 2003.

[18] J. Weisz and P. K. Allen, "Pose error robust grasping from contact wrench space metrics," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 557 – 562, 2012.

[19] A. Morales, E. Chinelalto, P. Sanz, A. Del Pobil, and A. H. Fagg, "Learning to predict grasp reliability for a multifinger robot hand by using visual features,"

Proceedings of the Eighth IASTED International Conference on Artificial Intelligence and Soft Computing, pp. 249 – 254, 2004.

[20] 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.

[21] Z. Li and S. Sastry, "Task-oriented optimal grasping by multifingered robot hands." *IEEE journal of robotics and automation*, vol. 6, no. 2, pp. 32 – 44, 1988.

[22] D. Kirkpatrick, B. Mishra, and C.-K. Yap, "Quantitative steinitz's theorems with applications to multifingered grasping," *Discrete & Computational Geometry*, vol. 7, no. 1, pp. 295–318, 1992.

[23] A. T. Miller and P. K. Allen, "Examples of 3d grasp quality computations," *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2, pp. 1240 – 1246, 1999.

[24] A. Morales, E. Chinellato, A. H. Fagg, and A. P. Del Pobil, "Experimental prediction of the performance of grasp tasks from visual features," *IEEE International Conference on Intelligent Robots and Systems*, vol. 4, pp. 3423 – 3428, 2003.

[25] A. Saxena, L. L. Wong, and Y. Andrew, "Learning grasp strategies with partial shape information," *Proceedings of the National Conference on Artificial Intelligence*, vol. 3, pp. 1491 – 1494, 2008.

[26] C. E. Rasmussen, *Gaussian processes for machine learning.* MIT Press, 2006.

[27] A. Saxena, L. Wong, M. Quigley, and A. Y. Ng, "A vision-based system for grasping novel objects in cluttered environments," *Springer Tracts in Advanced Robotics*, vol. 66, no. STAR, pp. 337 – 348, 2010.

[28] R. Diankov, "Automated construction of robotic manipulation programs," Ph.D. dissertation, Carnegie Mellon University, Robotics Institute, August 2010.

[29] W. Townsend, "The barretthand grasper-programmably flexible part handling and assembly," *Industrial Robot: An International Journal*, vol. 27, no. 3, pp. 181–188, 2000.

[30] D. Gossow, A. Leeper, D. Hershberger, and M. Ciocarlie, "ROS topics: Interactive markers: 3-d user interfaces for ROS applications," *IEEE Robotics and Automation Magazine*, vol. 18, no. 4, pp. 14 – 15, 2011.

[31] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction.* Springer, 2009.

[32] J. A. Hanley and B. J. McNeil, "The meaning and use of the area under a receiver operating characteristic (ROC) curve." *Radiology*, vol. 143, no. 1, pp. 29–36, 1982.

[33] S. S. Srinivasa, D. Ferguson, C. J. Helfrich, D. Berenson, A. Collet, R. Diankov, G. Gallagher, G. Hollinger, J. Kuffner, and M. V. Weghe, "HERB: a home exploring robotic butler," *Autonomous Robots*, vol. 28, no. 1, pp. 5–20, 2010.

[34] D. Fischinger, M. Vincze, and Y. Jiang, "Learning grasps for unknown objects in cluttered scenes," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 609 – 616, 2013.

[35] D. Song, C. Ek, K. Huebner, and D. Kragic, "Multivariate discretization for bayesian network structure learning in robot grasping," *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pp. 1944–1950, May 2011.

[36] J. Molina-Vilaplana, J. Feliu-Batlle, and J. Lopez-Coronado, "A modular neural network architecture for step-wise learning of grasping tasks," *Neural Networks*, vol. 20, no. 5, pp. 631 – 645, 2007.

[37] A. E. Leeper, K. Hsiao, M. Ciocarlie, L. Takayama, and D. Gossow, "Strategies for human-in-the-loop robotic grasping," *HRI'12 - Proceedings of the 7th Annual ACM/IEEE International Conference on Human-Robot Interaction*, pp. 1 – 8, 2012.

[38] D.-J. Kim, R. Hazlett-Knudsen, H. Culver-Godfrey, G. Rucks, T. Cunningham, D. Portee, J. Bricout, Z. Wang, and A. Behal, "How autonomy impacts performance and satisfaction: Results from a study with spinal cord injured subjects using an assistive robot," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 42, no. 1, pp. 2 – 14, 2012.

[39] A. Dragan and S. Srinivasa, *Formalizing Assistive Teleoperation.* MIT Press, July 2012.

[40] S. Ekvall and D. Kragic, "Learning and evaluation of the approach vector for automatic grasp generation and planning," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 4715 – 4720, 2007.

[41] L. Bodenhagen, R. Detry, J. Piater, and N. Kruger, "What a successful grasp tells about the success chances of grasps in its vicinity," *2011 IEEE International Conference on Development and Learning, ICDL 2011*, pp. IEEE Computational Intelligence Society; Frankfurt Institute for Advanced Studies (FIAS); Bielefeld Univ., Cognitive Interact.; Technol. Cent. Excellence (CITEC); italk –, 2011.

[42] J. D. Hunter, "Matplotlib: A 2d graphics environment," *Computing In Science & Engineering*, vol. 9, no. 3, pp. 90–95, 2007.

[43] R. Platt, R. Grupen, and A. Fagg, "Learning grasp context distinctions that generalize," *Humanoid Robots, 2006 6th IEEE-RAS International Conference on*, pp. 504–511, Dec 2006.

[44] J. Kim, K. Iwamoto, J. Kuffner, Y. Ota, and N. Pollard, "Physically-based grasp quality evaluation under uncertainty," *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pp. 3258–3263, May 2012.