Implementation of a Gaussian process-based machine learning grasp predictor

Alex K. Goins, Ryan Carpenter, Weng-Keen Wong & Ravi Balasubramanian

Autonomous Robots

ISSN 0929-5593

Auton Robot
DOI 10.1007/s10514-015-9488-2

Springer

Auton Robot
DOI 10.1007/s10514-015-9488-2

Implementation of a Gaussian process-based machine learning grasp predictor

Alex K. Goins[1] · **Ryan Carpenter**[1] · **Weng-Keen Wong**[2] · **Ravi Balasubramanian**[1]

Received: 26 August 2014 / Accepted: 5 August 2015

Abstract With the goal of advancing the state of automatic robotic grasping, we present a novel approach that combines machine learning techniques and physical validation on a robotic platform to develop a comprehensive grasp predictor. After collecting a large grasp sample set (522 grasps), we first conduct a statistical analysis of the predictive ability of grasp quality metrics that are commonly used in the robotics literature. We then apply principal component analysis and Gaussian process (GP) algorithms on the grasp metrics that are discriminative to build a classifier, validate its performance, and compare the results to existing grasp planners. The key findings are as follows: (i) several of the existing grasp metrics are weak predictors of grasp quality when implemented on a robotic platform; (ii) the GP-based classifier significantly improves grasp prediction by combining multiple grasp metrics to increase true positive classification at low false positive rates; (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.

Keywords Grasping · Machine learning · Grasp metrics

✉ Alex K. Goins
alexgoins301@gmail.com

Ravi Balasubramanian
ravi.balasubramanian@oregonstate.edu

[1] School of Mechanical, Industrial, and Manufacturing Engineering, Oregon State University, Corvallis, OR 97331, USA

[2] School of Electrical Engineering and Computer Science, Oregon State University, Corvallis, OR 97331, USA

1 Introduction

Robots have typically been used in highly structured industrial environments. In order to incorporate them into less structured outdoor or domestic environments, robots need to have automatic grasp algorithms which allow them to interact with everyday objects. With this in mind, researchers have developed a variety of approaches based on force modeling (Kehoe et al. 2012; Ferrari and Canny 1992), machine-learning (Pelossof et al. 2004), and human-inspired grasping (Faria et al. 2012) to generate and predict robotic grasp performance prior to execution. While significant progress has been made, recent results show that even the best method has a failure rate of 23 % when implemented on a physical robot (Balasubramanian et al. 2012). Such a high failure rate shows the complexity of the robotic grasping problem, in part due to the difficulty in modeling the multi-dimensional grasp space consisting of all the grasping parameters as well as non-linear effects such as contact friction, slip, compliance, and object movement during grasping.

In an attempt to simplify the grasp space and account for some of these effects, researchers have developed grasp metrics which capture some of the defining characteristics of stable grasps. For example, the physics-based grasp metrics "epsilon" and "volume" were developed to compute the external disturbance forces that the contacts established by the robot hand can resist (Ferrari and Canny 1992). Another example is "grasp energy", which creates virtual contact points on the hand with force magnitudes that are scaled based upon distance from the object, and calculates the grasp wrench of these potential contacts (Ciocarlie et al. 2007).

Surveys of grasping literature (Chinellato et al. 2005; Lin et al. 2000; Suárez et al. 2006; Shimoga 1996; El-Khoury et al. 2011; Lopez-Damian et al. 2005; Bohg et al. 2013) list as many as 24 grasp metrics which have been developed, each

Published online: 19 August 2015

Table 1 Grasp metrics

Heuristic	Description	Min	Max	Citation
Contact point equilateralness[a]	Equilateralness of the triangle made by the contact points of the finger tips	0	1	Chinellato et al. (2005)
Grasp volume[a]	Volume of the triangular prism consisting of the finger tips and the palm	0	669 cm^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	Saxena et al. (2008a, b)
Perpendicular symmetry[b]	Distance between center of mass of object and contact point perpendicular to the object principal axis	0.5	1	
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°	Balasubramanian et al. (2012)
Grasp Wrench (Epsilon)[a]	Minimum disturbance wrench that can be resisted	0	1	Ferrari and Canny (1992) and Miller and Allen (2004)
Grasp wrench volume[a]	Volume of grasp wrench space	0	2^6	
Grasp energy[b]	Distance of hand sample points to object	$-\infty$	∞	Ciocarlie et al. (2007)

[a] Larger $\geq$ better grasp; [b] smaller $\geq$ better grasp; [c] mid-range $\geq$ better grasp

representing a small aspect of the grasp performance (see Table 1 for a list of some of them). While some metrics, like finger spread, apply only to three finger grippers, the majority of the metrics are applicable to other multifinger grippers (Bounab et al. 2008; El-Khoury and Sahbani 2009) and even the human hand (Leon et al. 2012). However, as the metrics have been further studied, it was found that slight variations in hand placement relative to the object can significantly change the metric value and the resulting grasp performance (Chinellato et al. 2003; Weisz and Allen 2012). Furthermore, the metric values are typically correlated, as they are often calculated from dependent variables (such as finger contact location) which are based on independent variables (such as hand pose, orientation, finger spread, and object type). As such, adjusting one independent variable influences multiple dependent variables, affecting the computation of several metrics simultaneously.

In order to account for this broad grasp feature space and increase prediction performance, researchers have developed aggregate grasp metrics which merge several individual metrics, up to as many as nine metrics (Morales et al. 2004). For example, weighted sums of epsilon, volume, and energy have been used simultaneously in the open source software GraspIt! (Miller and Allen 2004) for grasp planning (also see (Kehoe et al. 2012; Chinellato et al. 2003; Li and Sastry 1988; Kirkpatrick et al. 1992; Miller and Allen 1999) and Table 1 for other examples).

However, there are three key problems with the current grasp planning methods. First, most of the grasp metrics have been evaluated through simulation only, with limited validation of these metrics on physical robots (Balasubramanian et al. 2012; Morales et al. 2004, 2003; Saxena et al. 2008b). Second, current methods have largely failed to account for the interactions or correlations between the grasp metrics (Leon et al. 2012; Morales et al. 2004) 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 shortcomings of the metrics, others have tried learning grasp examples with various machine learning algorithms such as support vector machines, Bayesian networks, and neural networks to create a more robust grasp planner

using previous grasp examples (Fischinger et al. 2013; Song et al. 2011; Molina-Vilaplana et al. 2007). These methods learn from a small subset of the grasp metrics, usually limited to two or three. However, if the metrics chosen are highly correlated, or if the grasp space is not sufficiently explored, then the resulting grasp planner may not be robust enough in order to generate or plan new grasps.

This paper expands on previously published work (Goins et al. 2014) by (i) applying the GP algorithm to improve bad grasp examples by searching for neighboring grasps with better performance. (ii) And exploring the GP's performance as the data set size is varied to determine the number of grasp examples required to build a successful classifier.

2 Background

In this research, we use a Gaussian process (GP) as our machine learning algorithm because it is able to determine the non-linear relationships among the grasp metrics as well as providing the variance in the predicted values. This allows us to determine the predicted grasp success rate prior to execution as well as the uncertainty in the prediction. Other machine learning algorithms which are able to handle the non-linearities of the grasp space could be used and compared, but it is beyond the scope of this paper to explore them all.

A GP (Rasmussen 2006) is a generalization of a Gaussian distribution. Just as a multivariate Gaussian distribution models a probability distribution over vectors $\mathbf{x}$, a GP models a probability distribution over functions $f(\mathbf{x})$. To see this, we can consider a function to be an infinite-dimensional vector, with each component of the vector being the value of the function $f(\mathbf{x})$ at some value of $\mathbf{x}$. In addition, just as a Gaussian distribution is defined by a mean and a variance matrix, a GP is similarly defined by a mean function, which is often standardized to be the zero function, and a covariance function, which represents a similarity measure between two input vectors $\mathbf{x}$ and $\mathbf{x}'$. GPs have been widely used in robotics because they provide a way to model a non-linear response from input values and because they can provide a measure of uncertainty for the predicted output value.

In our work, each data instance $\mathbf{x_i}$ is a grasp, which has k features where $k = 12$. These twelve features are represented by the twelve metrics calculated for each grasp (see Table 1). We use an open-source Matlab[1] package known as GPML[2] to predict a continuous output value between 0 to 1 that represents the probability of the grasp being successful.

[1] http://www.mathworks.com/.

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

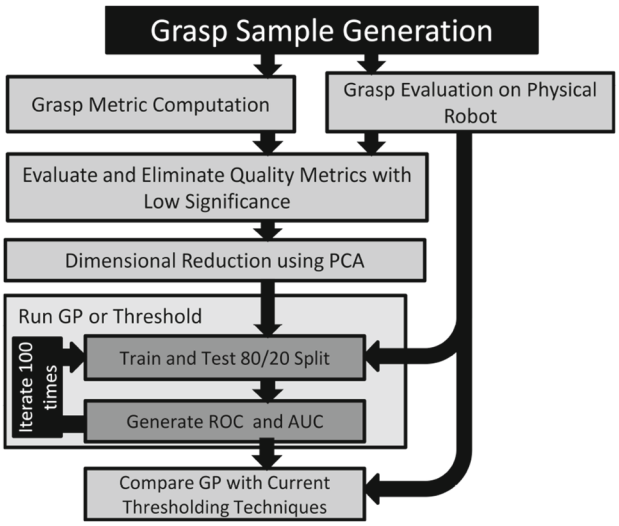

Fig. 1 Flow chart of experimental procedure

3 Experimental methods

Our approach begins with an evaluation of twelve of the most common kinematics-based metrics (see Table 1), both in simulation and on a physical robotic platform. Other metrics were not included since our Barrett robotic system does not have the force or tactile sensors to support their inclusion. For robotic systems that have these capabilities, the extra metrics can be easily added in the future by following the same procedure. The results of the metric evaluation is then used to select which metrics to include in the machine learning algorithm. An overview of the procedure is shown in Fig. 1.

3.1 Collection of the grasp sample set

Twenty two human subjects were recruited to provide example grasps for nine everyday objects (see Fig. 2) using a simulation environment developed in OpenRAVE (Diankov 2010). A total of 522 grasp examples were collected using three common human-robot interfaces, a gamepad controller, a three-dimensional mouse, and an "interactive marker" display developed in ROS (Gossow et al. 2011). Each human subject commanded the position, orientation, finger spread, and grasp closure of the virtual BarrettHand (Townsend 2000) robotic hand, and had the option of viewing the grasp from several angles. Each subject created, at most, one grasp per object for each interface. The variation in the interfaces, as well as the randomization of the robot hand starting location, were both done so as to avoid repeated grasps and increase the diversity in the grasp examples to prevent the data set from being skewed. Once the user was satisfied with the final grasp, both the robot hand's pose relative to the object's coordinate frame and the computed metric scores were recorded. The human-subject experiment procedure

Fig. 2 Nine everyday objects used for grasp generation. From *top left* to *bottom right*: pitcher, water bottle, cracker box, soap dispenser, martini glass, wire spool, remote control, soda can, cd case. Number of grasp examples collected for each object respectively: 56, 59, 57, 59, 61, 57, 54, 55, 64

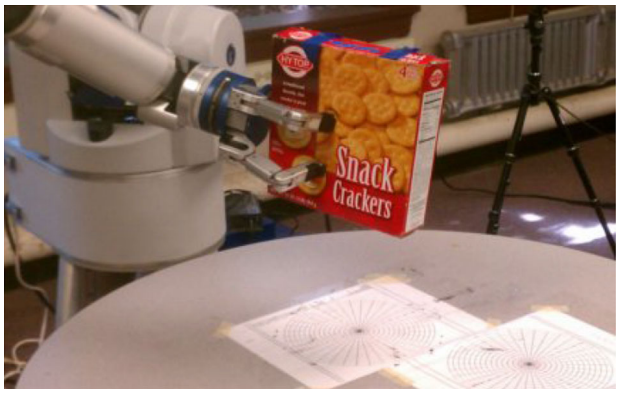

Fig. 3 Shake test setup using WAM and marked reference location for object placement

was approved by Oregon State University's Human Subjects Division.

3.2 Evaluation of the grasp sample set

After grasp generation, all of the grasps were tested using the BarrettWAM and BarrettHand platform. The default BarrettHand fingers with rubber pads, and default Barrett grasp controller were utilized for all of the tests. This grasp controller closes all of the fingers simultaneously at constant velocity until a force limit is reached in each finger. A table was placed in front of the BarrettWAM with a marker placed at a known distance from the WAM. This marker had evenly spaced radial and axial lines to help align the object centroid with respect to the robot so as to accurately recreate the grasps as they were planned in simulation (see Fig. 3). Foam spacers were placed inside the soda can and cracker box which allowed some flexing, but prevented crushing of the object.

During grasp execution, it was ensured that all of the fingers would make contact simultaneously by computing the finger joint values in simulation. If the fingers do not make simultaneous contact, then the object position would be disturbed by those fingers making premature contact, resulting in a final grasp that does not match the one created in simulation. If there is a discrepancy between the simulated and the executed grasp, then the test results will not correspond to the metrics as calculated in simulation. In this work, we are trying to validate the metrics' classification performance rather than test the metrics' sensitivity to sensing or placement errors during the grasping process.

The hand was first commanded to the grasp pose location with all of the fingers open. Then, using the joint values obtained from simulation, the hand was closed to a pre-grasp pose with all of the fingers a uniform distance away from the object prior to the final closure of the grasp. Finally, the fingers were commanded closed at a constant velocity until contact was made, and the object lifted and subjected to a series of shake motions. This was done by rotating the three wrist joints, in sequential order, toward one joint limit and back to its starting location, so as to subject the object to a variety of forces and torques from all directions. This shake motion was performed so as to emulate the effect of external forces acting upon the object. The magnitude of the shake disturbances are shown in Table 2 and are greater than those used in previous work (Balasubramanian et al. 2012). If the object was retained in the hand it was labeled a success, and a failure if it fell and hit the table. Each grasp was tested ten times, for a total of 5220 shake tests, and the overall success rate for each grasp was recorded. After testing, the grasps were separated into two categories, "good" and "bad", based upon their overall success rate. If a grasp had an 80 % or greater success rate, it was labeled "good", and otherwise labeled "bad". This threshold was selected based upon current research which showed that existing grasp planners have an average success rate of 80 % (Balasubramanian et al. 2012), although a higher threshold could easily be selected as well.

3.3 Quantitative evaluation of grasp metrics

Prior to evaluating the metrics, the data was normalized to a mean of 0 as

Table 2 End-effector shake test magnitudes

Type	Peak	Mean
Angular velocity (rad/s)	41.93	4.67
Linear velocity (m/s)	87.05	0.49
Angular acceleration (rad/s^2)	40.76	3.49
Linear acceleration (m/s^2)	86.94	0.44

$$x'_{(m,n)} = \frac{x_{(m,n)} - \overline{x}_m}{\sigma_m}, \quad (1)$$

where for a given metric m and n data points, $x'_{(m,n)}$ is the normalized value for the observation $x_{(m,n)}$ with sample mean $\overline{x}_m$ and sample standard deviation σ_m. Normalizing the data is important because it places all of the metrics on a uniform scale prior to performing PCA. If the metrics are not uniformly scaled, then the results from PCA will be skewed by those metrics which have larger ranges. Normalization also does not affect the predictive performance of the metrics since it is simply translation and a scaling factor applied to the data.

Two methods were used to evaluate the discriminative ability of each metric. First, a two-tailed t test (p value $\leq$ 0.05) was used to compare the good and bad grasp sets to determine if the grasp metric was a good predictor of grasp success rate. A metric which showed showed a statistically significant difference was considered to be a potentially good metric for inclusion in a grasp planner. Second, a simple classifier was built based on thresholding over the grasp metric value. If the metric value was greater or less than a desired threshold value, the grasp was considered a good grasp. The results from this simple classifier were compared with the GP based classifier to see how well the machine learning based classifier compares to the performance of the individual metrics (see Sect. 3.5).

3.4 Dimensionality reduction using principal component analysis and statistical testing

Even though we are exploring twelve metrics, the grasp space may be easily reduced to fewer dimensions due to correlation among the metrics and the poor classification performance of some of the metrics. First we reduce the number of metrics used in the classifier by eliminating metrics which did not prove statistically significant in the t tests (see Sect. 3.3). Then, in order to deal with correlated metrics, we apply principal component analysis (PCA) to the remaining metrics and reduce the data to only a few dimensions in the full dimensional space (Hastie et al. 2009). We then evaluate the machine learning performance as the number of principal components included in the GP are varied, starting with just the first principal component and adding each successive principal component until all are included.

3.5 Building a Gaussian process-based classifier for grasp quality prediction

In order to avoid creating a composite metric by hand, we use Gaussian processes to merge the information provided by each metric. This allows us to easily expand the grasp planner when extra information or data about new objects and grippers becomes available. In this work, we utilize a GP with a squared exponential covariance function, and an Automatic Relevance Determination distance measure. This distance measure was selected to better fit the nonlinearities of the metric space, as it creates a set of independent relevance weights for each metric and automatically determines the values for these weights. After selecting the desired principle components created from performing PCA (see Sect. 3.4), a cross-validation technique was used for training the GP. In this step, 80 % of the data is selected at random to train the GP classifier, and the remaining 20 % of the data is used for testing (Hastie et al. 2009). This process was repeated one hundred times and the average performance of the GP-based classifier using a threshold was recorded.

In order to determine the performance of a classifier based on the individual metrics, a threshold was applied to the metrics using 20 % of the data randomly selected from data set. This process was also repeated one hundred times and the results averaged so as to obtain the individual metrics' performance. This was then compared with the GP classifier to examine the performance improvement gained by using the machine learning algorithm.

The GP classifier and the metric thresholding were both examined using 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 (Hanley and McNeil 1982). The shape of the ROC indicates the performance of the classifier at correctly labeling good grasp, or the True Positive Rate (TPR), as well as incorrectly labeling bad grasps as good, or the false positive rate (FPR). Also, the area under the curve (AUC) helps to give a quantitative value to the overall performance of the classifier, with an AUC of 1 indicating perfect performance, and an AUC of 0.5 indicating random classification. After running the cross-validation on the GP and the metrics, the results were averaged, and the AUC and TPR at 5, 10, and 15 % FPR levels were found.

3.6 Gaussian process based grasp improvement

Once the GP based classifier is generated, it can then be used to evaluate the performance of new grasps. If the grasp's predicted success rate is high, the grasp may 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.

3.6.1 Calculation of grasp score

Starting with an initial grasp pose, the metric scores of the grasp are calculated and transformed into PCA space

and the performance of the grasp is then evaluated 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 PCA space which has better performance according to the GP.

While a desired grasp can be found in PCA space and subsequently mapped to the corresponding metric values, there are multiple solutions in end-effector space which can satisfy the resulting metric values. Furthermore, as previously mentioned in Sect. 1, the metrics are calculated from several dependent variables. Because of this and the complex surface curvatures possible, there is no direct mapping from the metric scores back to the end-effector space, preventing 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 (Hunter 2007) (see Fig. 4) using the first two principal components (PC1 and PC2). For more complex surfaces, the plot can be used to select a region of better performance and drive the grasp search algorithm towards that region. This approach helps to avoid the problem that a gradient search has, namely that of finding local maximums. If the surface is simple, then the human interaction step can be skipped, and the grasp can be optimized solely based on improving the GP predicted score over each successive iteration.

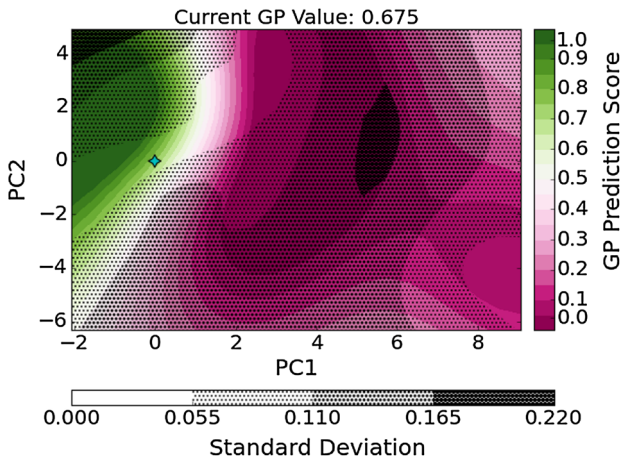

Fig. 4 Plot showing the first principal component (PC1) and second principal component (PC2) of the PCA space with *color* based on GP predicted score and *shading* based on uncertainty. Current grasp location with GP predicted value shown by the *blue diamond* (Color figure online)

Algorithm 1 Grasp Search Protocol

```
m = seed grasp(s)
S_m = GP predicted success rate for all m
Goal = GP target
while S_m1 < Goal do
    for all i in m do
        for j in (1,2,...,20) do
            C_n = m_i + (Δpose, Δspread)
        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
```

3.6.2 *Grasp search protocol*

In order to improve the grasp performance, an optimization algorithm is used to perturb the grasp in the robot end-effector space 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). Since a direct mapping cannot be found from the grasp metric space to the end-effector space, a gradient assent search method cannot be utilized. This limits the search to an iterative, forward search method such as Particle Swarm optimization. More about this will be discussed in Sect. 4.

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. If the desired predicted grasp score 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. For each of the four seed grasps, twenty new grasp candidates are generated, for a total of eighty candidate grasps for the next round of optimization testing. Each subsequent round of searching uses the four best grasps from the previous round in order to seed the next round of grasp candidates. 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. Other, more sophisticated search algorithms could be utilized in order to decrease the search time, however, the focus of this research is to show that the GP can be utilized for new grasp generation. Once the performance of the GP has been validated, new search algorithms can be implemented to speed up this process, which will be included in future work.

Table 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
† TPR/(FPR+TPR)

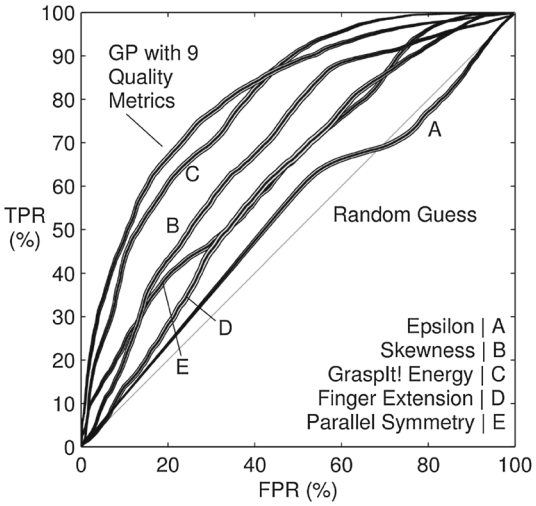

Fig. 5 Representative ROCs of representative grasp metrics and GP classifier (mean ± standard error over 100 trials)

4 Results

Of the 522 grasps in the dataset, 376 (72 %) grasps were successful (average success >80 %) and the remaining 146 failed (28 %).

4.1 Discriminative ability of individual grasp metrics

Table 3 shows the statistical significance of each metric to discriminate between good and bad grasps based on t tests and the results of thresholding on each grasp metric. Sorting the table's rows based on increasing t test p values reveals that only nine of the twelve grasp metrics can discriminate between good and bad grasps for this set of grasps at the $p = 0.05$ statistical significance level. The ROC curves in Fig. 5 shows that the best GP-based classifier performs better than thresholding on the individual grasp metrics especially at low FPR values (50 vs 43 % TPR for GP and Energy respectively). Furthermore, for the individual metric performance, only grasp energy shows significant classification ability, whereas the remaining metrics have marginal performance as shown by the low AUC values and low TPR values in Table 3.

4.2 Principal component analysis of the grasp sample set

The cumulative variance explained by each additional principal component increases almost linearly (correlation to a 45° slope line is 0.97) implying that all the principal components account significantly for the variance in the data. However, the AUC value for performing GP with the first PC was 0.76 which increased to a maximum of 0.82 when using the first four PCs. Since there was no further gains in the AUC from adding more than four PCs, then the remaining PCs could safely be excluded in order to reduce computation time. However, since retaining more principal components does not necessarily improve predictive accuracy, testing would need to be done for each data set to determine how many PCs are necessary to build the best classifier in terms of AUC and TPR.

4.3 Performance of the GP-based classifiers

Table 4 shows the results from creating a GP-based classifier using various subsets of the statistically significant grasp metrics and all the principal components derived from those metrics. The results show that using all of the grasp metrics which were found to be statistically significant from the t test provides for the best performing classifier. Adding any additional metrics beyond this only decreases the performance, especially at low FPR levels. Additionally, comparing

Table 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 ($p < 0.05$)

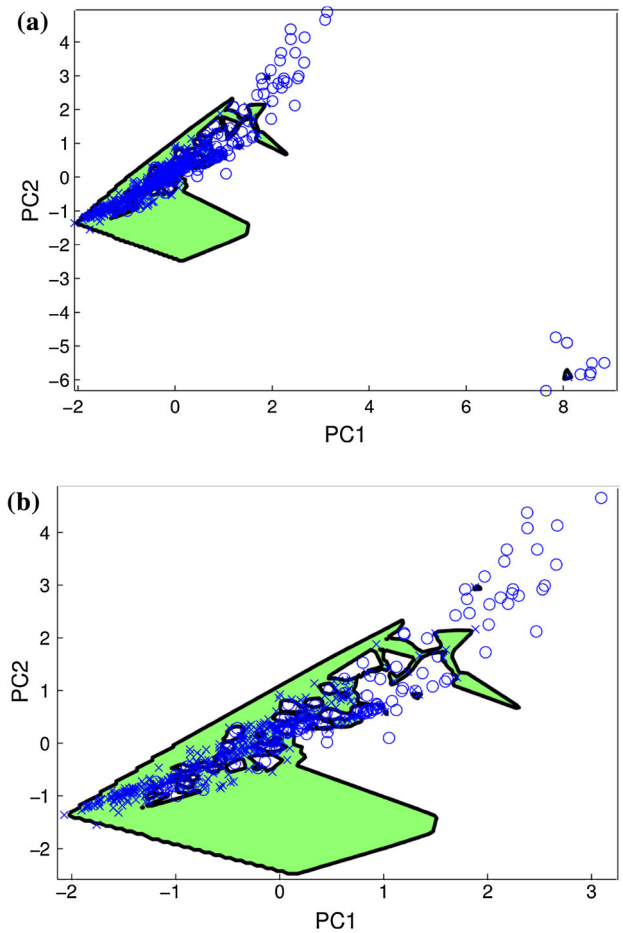

Fig. 6 **a** Visualization of a two dimensional projection of the a nine-dimensional surface that the GP creates to predict grasp quality. The "×" indicates good grasps and "○" indicates bad grasps from the grasp sample set. The *filled area* represents the "good" grasp region with success rate >83 % and a 10 % FPR classification level. **b** Scaled subregion from (**a**)

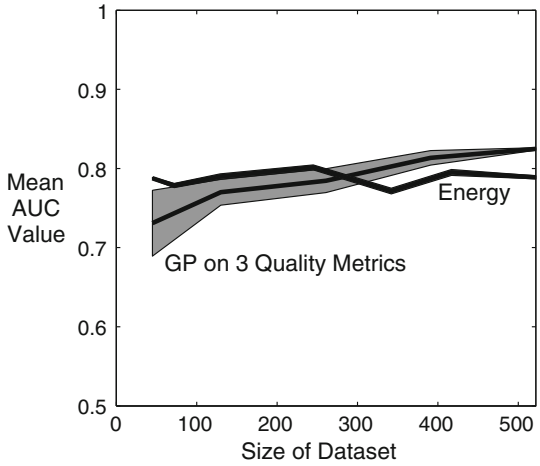

Fig. 7 Performance of GP and Energy with threshold as data set size increases (mean ± standard error)

Table 3 to the 10 % FPR column of Table 4 shows that GP classifier has increased performance over thresholding of the individual metrics. Furthermore, at the 5 % FPR level, the GP classifier has a true positive rate of 38 % for an overall success rate of 88 % [TPR/(TPR + FPR)]. This is better than other similar work which had an overall success rate of 81 % Saxena et al. (2008b).

Figure 6 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 nine grasp metrics from Table 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. Figure 7 shows how the performance of this classifier improves (measured in terms of AUC values) as the data set size increases. As expected for small datasets, the GP-based classifier performs worse than thresholding using the energy metric. However, when the grasp sample set size goes beyond 300, the GP-based classifier performs better than energy-based thresholding.

4.4 Grasp improvement results

Using thirty-two 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. Figure 8 shows two of the grasps before and after using the GP to improve the grasp.

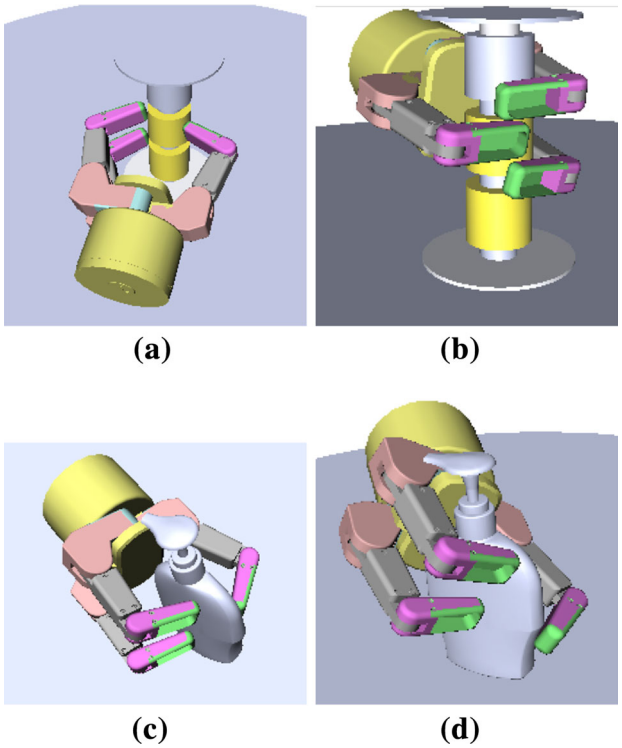

Fig. 8 Example grasp for wire coil **a** before and **b** after optimization and soap bottle **c** before and **d** after optimization

Table 5 GP grasp improvement performance using 32 different grasps

Average GP score		Average success rate	
Before	After	Before	After
0.39	0.68	0.61	0.83

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 29 % improvement in the predicted success rate and a 22 % improvement in the overall success rate (see Table 5).

5 Discussion

This section discusses the statistical and machine learning methods used pertaining to what can be learned from the work done and applied to future research.

5.1 Metric performance

Our testing revealed that many of the grasp metrics found in robotics research are weak predictors of grasp success rate (see Table 3). However, the *t* test proved to be a good method for determining which grasp metrics were significant and can be used to build a combined classifier with improved classification performance. Using this method on the metrics found in Table 1, we were able to create a GP-based classifier (TPR = 0.38) which is significantly improved over the best individual grasp metrics (energy TPR = 0.23), which is a 66 % increase in the true positive rate at an FPR of 5 %. This was accomplished because the machine learning algorithm is able to merge the metrics in a non-linear fashion across the grasp space.

An important finding is that adding in metrics with low predictive ability reduces the performance of the GP by decreasing classification performance while simultaneously increasing the computation time. These metrics had low discriminative performance since the users who provided the grasp examples tended to prefer power over precision grasps, and also utilized grasping features of the objects such as handles. For the case of the glass and the pitcher, the grasps created were such that the fingers wrapped around the glass stem or the pitcher handle, but there was no palm contact when initially created in simulation. This caused many of the grasps to not have full force closure, resulting in 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, or were able to obtain force closure due to the object moving in the hand during grasp closure. As such, the successful grasps had widely varying grasp wrench and epsilon scores, resulting in their low performance and eventual exclusion from the GP. If the grasp data were expanded to include different objects and more force closure grasps, then these metrics could prove significant and be reintroduced into the GP.

5.2 GP classification performance

As the number of data points used to build the classifier increases, the classification success rate of the GP classifier also increases (see Fig. 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. 6 that the current data set's spread can be improved, given the clustering of grasp samples in the $(-2 < \mathrm{PC}_1 < 2, -1 < \mathrm{PC}_2 < 2)$ range. More experiments are needed for exploring other regions of the grasp space in order to realize more gains in performance.

Similar experiments have been performed but usually on smaller data sets and using computer vision for object pose detection. Specifically, one group tested ninety grasps across four planar objects and tested a total of 920 times and were able to achieve an average prediction success rate of about 76 % while using nine quantifiable grasp measurements (Morales et al. 2004). Another group tested thirteen novel 3-D objects across 150 trials and achieved a prediction success rate of 81 % across all objects using three grasp features (Saxena et al. 2008b). In our work, we used 522 grasps on

nine objects with 5220 grasp trials and was able to achieve a higher TPR at low FPR levels and an overall success rate of 88 % (at 5 % FPR) using nine of the twelve tested grasp metrics. A key advantage of our work is the ability to use prior grasp performance data to select a desired FPR level for future prediction performance. However, given the complexity and number of dimensions of the grasp space, more grasp examples and validation over more objects is needed to improve the grasp predictor's performance and to find regions of strong or weak performance of the grasp metrics.

5.3 Grasp quality improvement using GP-based classifiers

Using thirty-two low performing grasps, we were able to show an overall improvement of 35 % in the grasp performance. This shows the GP's ability to not only correctly classify grasps, but to generate new grasps which have high robustness. Looking at some of the images of grasps before and after optimization (Fig. 8) shows that the GP is able to increase the number of contact points and optimally place the contact points so as to better resist any disturbances.

While the GP classifier shows promise to improve bad grasps, only one in three grasps were improved. One reason why may be that, for many of the objects, the subjects preferred power grasps to precision grasps. Since the grasp set was skewed toward power grasps, the amount of information available for precision grasps was limited, and thus proved difficult to plan stable precision grasps. Furthermore, the disturbance magnitudes utilized to validate the robustness of the grasp was large enough to dislodge many of the objects from the precision type grasps. These two factors favored power grasps over precision grasps, which means that if a suitable power grasp cannot be found, then the final grasp will have a low chance of succeeding. This was especially evident with the television remote control, which because of its low profile, was restricted to precision only grasps. For this object, the GP guided search was unable to find any grasps which had a predicted success rate over 80 %. To fix this, more testing is needed with precision type grasps so that there will be enough information to properly train the GP using the metrics grasp wrench and epsilon which may be able to compensate for these issues.

A second reason for the search to fail could be due to the grasp search algorithm. In this work, we utilized one method which is similar to an unguided particle swarm optimization algorithm. However, other algorithms which use a dynamic or adaptive search would have better success rates and increase the number of grasps which could be improved. Because the end-effector search space and the grasp space are not linearly related, a small change in the end-effector pose can cause large changes or even jumps in the grasp space. If a mapping could be made between the end-effector space and the grasp space, then a hill-climbing optimization search could be utilized whereby the desired grasp is created not by random sampling, but by directly solving for the final end-effector location given a desired position in the grasp space.

6 Conclusions

Since 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. This would then increase the confidence of predictions made in the new region of the grasp space. 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 scores. More work is necessary to generalize the grasp prediction to incomplete data.

One key advantage the GP-based classifier offers over the individual metrics is the significantly higher TPR at low FPR values compared to other grasp planners. This can improve online computation results by providing more candidate grasp choices at the same FPR value as other grasp planners, thereby reducing the time required to find a suitable grasp choice. Alternatively, the grasp performance can be increased by reducing the FPR level such that the number of grasp choices remains the same, but the chance of failure is reduced.

One limitation of this research is the absence of grasping dynamics, which include finger and object motion, in the metric computation and grasp planning. While great care was taken to ensure that the object moved negligibly during the grasping process, this approach may not be feasible in real world applications. If grasping dynamics were accounted for, then grasps could be adaptively adjusted to compensate for disturbances to the object while grasping, and could further improve performance. Furthermore, the ability to calculate and plan for object motion would provide the ability to plan grasps for flexible and compliant objects (Molina-Vilaplana et al. 2007; Platt et al. 2006; Kim et al. 2012). Also, many of the grasp metrics were created for obtaining stable grasps with pick and place operations in mind, and may not be well suited for resisting large external forces as was tested in this work. A better measure of the grasp metrics' performance

would be to test the grasps at different levels of force, starting at zero and increasing until the grasp failed. This would provide a more continuous measure of the grasp metrics performance as it would show how well the metric performed at predicting overall grasp performance instead of grasp success rate at one force magnitude.

A second limitation is that of transferring information across robotic platforms. While we did use a common robotic platform used for both research (Srinivasa et al. 2010) and development[3], more testing would be needed to transfer the results to other robotic platforms which incorporate extra modalities such as tactile information. Ideally, we would like to validate as many metrics as is possible at one time, and use a reduced subset size of these metrics for a given robotic platform based upon the new platforms capabilities. This will reduce the amount of testing and setup time required for each new robotic platform since all of the necessary data will have already been collected. Finally, while data was collected for a variety of objects, we did not investigate the role that the shape of the object plays in the grasping process. Further gains could be realized if objects could be categorized and specific heuristics or machine learning algorithms applied to those categories. Another possibility would be to provide a way to encode the shape of the object into the learning process to create a GP which is able to plan object and context sensitive grasps.

In this work, we developed a method to select metrics to train a machine learning algorithm to create a robust grasp planner. In practice, data would be collected for a variety of object types which then could be utilized in various applications. With an automatic grasp planner, the level of autonomy of the robot can be increased and 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 (Song et al. 2011), 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 (Ekvall and Kragic 2007; Bodenhagen et al. 2011).

References

Balasubramanian, R., Xu, L., Brook, P. D., Smith, J. R., & Matsuoka, Y. (2012). Physical human interactive guidance: Identifying grasping principles from human-planned grasps. *IEEE Transactions on Robotics*, *28*(4), 899–910.

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

Bohg, J., Morales, A., Asfour, T., & Kragic, D. (2013). Data-driven grasp synthesis—A survey. *IEEE Transactions on Robotics*, *PP*(99), 1–21.

Bounab, B., Sidobre, D., & Zaatri, A. (2008). Central axis approach for computing n-finger force-closure grasps. In *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 1169–1174.

Chinellato, E., Fisher, R. B., Morales, A., & Del Pobil, A. P. (2003). Ranking planar grasp configurations for a three-finger hand. *Proceedings of IEEE International Conference on Robotics and Automation*, *1*, 1133–1138.

Chinellato, E., Morales, A., Fisher, R. B., & del Pobil, A. P. (2005). Visual quality measures for characterizing planar robot grasps. *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, *35*(1), 30–41.

Ciocarlie, M., Goldfeder, C., & Allen, P. (2007). Dimensionality reduction for hand-independent dexterous robotic grasping. In *IEEE International Conference on Intelligent Robots and Systems*, pp. 3270–3275.

Diankov, R. (2010). Automated construction of robotic manipulation programs. PhD thesis, Carnegie Mellon University, Robotics Institute.

Ekvall, S., & Kragic, D. (2007). Learning and evaluation of the approach vector for automatic grasp generation and planning. In *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 4715–4720.

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

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

Faria, D. R., Martins, R., Lobo, J., & Dias, J. (2012). Extracting data from human manipulation of objects towards improving autonomous robotic grasping. *Robotics and Autonomous Systems*, *60*(3), 396–410.

Ferrari, C., & Canny, J. (1992). Planning optimal grasps. *Proceedings of IEEE International Conference on Robotics and Automation*, *3*, 2290–2295.

Fischinger, D., Vincze, M., & Jiang, Y. (2013). Learning grasps for unknown objects in cluttered scenes. In *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 609–616.

Goins, A. K., Carpenter, R., Wong, W. K., & Balasubramanian, R. (2014). Evaluating the efficacy of grasp metrics for utilization in a Gaussian process-based grasp predictor. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*.

Gossow, D., Leeper, A., Hershberger, D., & Ciocarlie, M. (2011). ROS topics: Interactive markers: 3-d user interfaces for ROS applications. *IEEE Robotics and Automation Magazine*, *18*(4), 14–15.

Hanley, J. A., & McNeil, B. J. (1982). The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology*, *143*(1), 29–36.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning: Data mining, inference, and prediction*. New York: Springer.

Hunter, J. D. (2007). Matplotlib: A 2d graphics environment. *Computing In Science & Engineering*, *9*(3), 90–95.

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

Kim, J., Iwamoto, K., Kuffner, J., Ota, Y., & Pollard, N. (2012). Physically-based grasp quality evaluation under uncertainty. In

[3] www.thearmrobot.com.

IEEE International Conference on Robotics and Automation (ICRA), pp. 3258–3263.
Kirkpatrick, D., Mishra, B., & Yap, C. K. (1992). Quantitative steinitz's theorems with applications to multifingered grasping. *Discrete & Computational Geometry*, *7*(1), 295–318.
Leon, B., Sancho-Bru, J. L., Jarque-Bou, N. J., Morales, A., & Roa, M. A. (2012). Evaluation of human prehension using grasp quality measures. *International Journal of Advanced Robotic Systems 9*.
Li, Z., & Sastry, S. (1988). Task-oriented optimal grasping by multifingered robot hands. *IEEE Journal of Robotics and Automation*, *6*(2), 32–44.
Lin, Q., Burdick, J. W., & Rimon, E. (2000). Stiffness-based quality measure for compliant grasps and fixtures. *IEEE Transactions on Robotics and Automation*, *16*(6), 675–688.
Lopez-Damian, E., Sidobre, D., & Alami, R. (2005). A grasp planner based on inertial properties. In *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 754–759.
Miller, A. T., & Allen, P. K. (1999). Examples of 3d grasp quality computations. *Proceedings of IEEE International Conference on Robotics and Automation*, *2*, 1240–1246.
Miller, A. T., & Allen, P. K. (2004). Graspit: A versatile simulator for robotic grasping. *IEEE Robotics and Automation Magazine*, *11*(4), 110–122.
Molina-Vilaplana, J., Feliu-Batlle, J., & Lopez-Coronado, J. (2007). A modular neural network architecture for step-wise learning of grasping tasks. *Neural Networks*, *20*(5), 631–645.
Morales, A., Chinellato, E., Fagg, A. H., & Del Pobil, A. P. (2003). Experimental prediction of the performance of grasp tasks from visual features. *IEEE International Conference on Intelligent Robots and Systems*, *4*, 3423–3428.
Morales, A., Chinelalto, E., Sanz, P., Del Pobil, A., & Fagg, A. H. (2004). Learning to predict grasp reliability for a multifinger robot hand by using visual features. In *Proceedings of the Eighth IASTED International Conference on Artificial Intelligence and Soft Computing*, pp. 249 – 254.
Pelossof, R., Miller, A., Allen, P., & Jebara, T. (2004). An svm learning approach to robotic grasping. *Proceedings of IEEE International Conference on Robotics and Automation*, *4*, 3512–3518.
Platt, R., Grupen, R., & Fagg, A. (2006). Learning grasp context distinctions that generalize. In *International Conference on Humanoid Robots, 2006 6th IEEE-RAS*, pp. 504–511.
Rasmussen, C. E. (2006). *Gaussian processes for machine learning*. Cambridge, MA: MIT Press.
Saxena, A., Driemeyer, J., & Ng, A. Y. (2008a). Robotic grasping of novel objects using vision. *International Journal of Robotics Research*, *27*(2), 157–173.
Saxena, A., Wong, L. L., & Andrew, Y. (2008b). Learning grasp strategies with partial shape information. *Proceedings of the National Conference on Artificial Intelligence*, *3*, 1491–1494.
Shimoga, K. B. (1996). Robot grasp synthesis algorithms: A survey. *The International Journal of Robotics Research*, *15*(3), 230–266.
Song, D., Ek, C., Huebner, K., & Kragic, D. (2011). Multivariate discretization for bayesian network structure learning in robot grasping. In *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1944–1950.
Srinivasa, S. S., Ferguson, D., Helfrich, C. J., Berenson, D., Collet, A., Diankov, R., et al. (2010). HERB: A home exploring robotic butler. *Autonomous Robots*, *28*(1), 5–20.
Suárez, R., Cornellà, J., & Garzón, M. R. (2006). Grasp quality measures. Institut dÓrganització i Control de Sistemes Industrials.
Townsend, W. (2000). The BarrettHand grasper-programmably flexible part handling and assembly. *Industrial Robot: An International Journal*, *27*(3), 181–188.
Weisz, J., & Allen, P. K. (2012). Pose error robust grasping from contact wrench space metrics. In *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 557–562.

Alex K. Goins received his bachelors degree in Mechanical Engineering from Oklahoma State University in 2008 and his masters degree in Mechanical Engineering from Oregon State University in 2014. During his study at Oregon State University, Alex participated in the DARPA Robotics Challenge developing grasp controllers for Team ViGIR. Alex is currently an engineer at Southwest Research Institute in the Automation and Data Systems Department providing automation services to industrial partners and contributing to the development of ROS Industrial.

Ryan Carpenter completed his BS in Mechanical Engineering at Oregon State University in 2009 and has worked for ESCO Corporation since graduation where he held many roles in design and process engineering. Since 2013, Ryan has also been working toward completion of his MS in Mechanical Engineering with an emphasis on robotics at Oregon State University. Ryan's graduate studies centers around robotic gripper design and analysis for industrial processes.

Weng-Keen Wong is an Associate Professor of Computer Science at Oregon State University. He received his Ph.D. (2004) and M.S. (2001) in Computer Science at Carnegie Mellon University, and his B.Sc. (1997) from the University of British Columbia. His research areas are in data mining and machine learning, with specific interests in anomaly detection and computational sustainability.

Ravi Balasubramanian is an Assistant Professor in the Mechanical Engineering department at Oregon State University, where directs the work in Robotics and Human Control Systems Laboratory. He received his undergraduate degree in Mechanical Engineering from the National University of Singapore with top honors and his PhD from the Robotics Institute at Carnegie Mellon University in 2006. He did a post-doctoral fellowship at the University of Washington in the emerging area of neurobotics, focusing on identifying the principles of human learning and execution in physical interaction tasks. During this time, he led a personal-robotics project (a collaboration with Intel Labs Seattle) and conducted a workshop on the biology of the human hand. He then was a research scientist in Yale University developing novel mechanisms for robotic hands. He has received several awards including the Best Student Paper finalist award at the International Conference on Robotics and Automation in 2004 and the Outstanding Researcher Award from the NIH National Center for Simulation in Rehabilitation Research in 2012. The key principle behind Dr. Balasubramanian's work is to simultaneously draw inspiration from the human control system to advance robotic systems and to use robotic systems to develop a stronger understanding of the human body and improve quality of life. Website: http://web.engr.oregonstate.edu/~balasubr/.