Improving Grasp Classification through Spatial Metrics Available from Sensors

Nigel Swenson, Garrett Scott, Peter Bloch, Paresh Soni, Nuha Nishat, Anjali Asar, Cindy Grimm, Xiaoli Fern, Ravi Balasubramanian

***Abstract*— We present a method for classifying the quality of near-contact grasps using spatial metrics that are recoverable from sensor data. Current methods often rely on calculating precise contact points, which are difficult to calculate in real life, or on tactile sensors or image data, which may be unavailable for some applications. Our method, in contrast, uses a mix of spatial metrics that do not depend on the fingers being in contact with the object, such as the object's approximate size and location. The grasp quality can be calculated *before* the fingers actually contact the object, enabling near-grasp quality prediction. Using a random forest classifier, the resulting system is able to predict grasp quality with 96% accuracy using spatial metrics based on the locations of the robot palm, fingers and object. Furthermore, it can maintain an accuracy of 90% when exposed to 10% noise across all its inputs.**

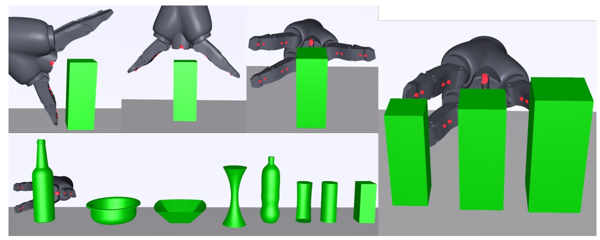

Fig. 1: Top left: Hand orientations: rotated, top down and normal. Bottom left: Shapes: tall bottle, rounded bowl, rectangular bowl, hourglass, short bottle, vase, cylinder and box. All shapes shown are medium sized. Right: The three sizes; all shapes have three sizes with these relative proportions.

I. Introduction

Robotic grasping is challenging for several reasons. First, it is difficult to understand, the physical interaction between the hand and the object. Second, there are a large number of potential grasp configurations, making grasp planning challenging and requiring large amounts of data collection. A wide variety of sensors attempt to remedy the first problem, including cameras, tactile sensors and distance sensors. Other works have built classifiers that can determine when a grasp is successful based on contact information, images or tactile sensor data [1]–[3]. This paper presents a grasp stability classifier that uses limited sensing information, reducing the potential cost and expanding its applicability to more environments. This classifier is also usable *before* the hand contacts the object (near-contact grasping), enabling grasp testing before, potentially, knocking the object over. It can also be used to ensure that a robot has a good grasp on an object before moving with the object [3], to train a grasp planner [4] or enable for regrasping [5].

Previous works that use contact-based grasp metrics have accuracy close to 85% [1], [6]. More recent work using physical tactile sensors achieved accuracy up to 93% on a variety of different objects, [3], [5]. Other methods instead rely on computer vision, using images to generate 3d representations of the objects and using these to classify grasps, resulting in an accuracy of 77% [2]. Lastly, some approaches combine tactile sensors with image data to plan grasps and evaluate grasp quality [4]. These approaches are promising, but many rely on the hand being in contact with the object to evaluate grasp quality.

Our **contribution** is a grasp classifier that uses spatial metrics that do not rely on contact points, is more resistant to noise and subtle variation, and works even when the hand is close, but not in contact with the object. We trained this on a wide array of shapes and hand orientations, as shown in Figure 1, to make it applicable to a large number of potential grasps.

II. Related Works

Previous works have demonstrated the usefulness of grasp quality classifiers and numerous methods to determine grasp quality. Grasp quality classifiers have been used for grasp planning, re-grasping, and accident prevention. These can be used to evaluate the necessary qualities of a good grasp classifier. Common methods used in grasping are calculated contacts, image-based analysis, 3D-shape analysis, and tactile sensors. We discuss each of these in turn.

A. Related Works: Grasp Classifier Uses

For grasp planning, Varley et al. [4] used contact forces and image data to calculate two grasp energy terms. These were used in conjunction with GraspIt! to plan grasps. Mahler et al. developed Dex-Net 4.0 [7], an ambidextrous grasp policy that used a depth camera to determine the optimal location to grasp based on wrench resistance. More accurate grasp classifiers can be used to further evaluate the potential grasps selected by the grasp planner.

Grasp classifiers are also useful for re-grasping, as shown by Chebotar et al. [5]. They used a grasp classifier to predict the success or failure of both their initial grasp and their re-grasp. This allowed them to determine the quality of their grasp before testing it, and if needed, pick a different position to re-grasp the object. In this use case it is important for the grasp classifiers to produce a smoothly-changing metric in order to support optimization search.

Lastly, grasp classifiers have been used to ensure a stable grasp before moving, thus preventing accidents. Kwiatkowski et al. used their grasp classifier to determine the quality of a grasp before attempting to pick up objects [3].

In summary, an effective grasp classifier should be fast, accurate and smooth. Next, we breakdown grasp classifiers by the metrics that they take as input.

B. Related Works: Grasp Classifier Inputs

Traditionally, classifiers were built from a small number of metrics calculated from finger and object contacts, forces at those contacts, and poses [8]–[10]. Balasubramanian et al. studied grasp metrics for their usefulness in [11], [12]. Later, Goins evaluated 12 of these metrics and were able to achieve a success rate of 88% on a set of 522 real world trials of nine objects of varying shapes [1]. Kim et al. utilized metrics that focused on the dynamics of the object and the uncertainty of its position [13]. Numerous works have found efficient ways to generate complex metrics from object shape and finger position to be used in grasp classification or planning [14]–[16]. Recent work developed metrics that worked relatively similarly across different hands and shapes [17]. However, almost all metric based approaches rely on the fingers being in contact with the object. By contrast, we use distances and relative orientations, which are more stable and are applicable *before* the fingers come into contact.

Some newer approaches build a classifier based on images of the hand and object [2]. This approach used RGBD images to generate 3d representations of the object that they were trying to grasp, then used these representations (in simulation) to evaluate the grasps.Similarly, Mahler et al., generates point clouds from image data to evaluate grasp quality and plan grasps, and similar approaches have been used elsewhere [7], [18], [19].

Tactile sensors are a promising area [3]–[5], [20], [21] that can also be used with image data to calculate grasp quality and plan grasps. Currently, despite giving a large amount of information about the interaction of the finger and the object, tactile sensors are often expensive and more prone to failure than simpler sensors such as time of flight [22]. However, our approach would support using both time of flight *and* tactile sensors.

The key difference in this work in relation to previous work is the focus on near-contact grasping and developing a fast and smooth classifier that could be utilized before the hand contacts the object.

III. Methods

For our grasp classifier, the overall approach was to begin with a large set of possible metrics and then use simulation-based data analysis to bring this large set down to a smaller set, reducing complexity and risks of over-fitting without losing prediction accuracy. We then used a small set of real-world tests to validate the resulting metrics.

We began with a large set of metrics grouped by both type and location on the hand (see Table I, Section III-A). We then specified a larger set of possible grasps with different objects and hand poses (Section III-B), using a Mujoco based simulation of a Kinova JACO 2 three-fingered gripper to label the grasp's success and determine which point in the grasp sequence to use for calculating the metrics (Section III-C). We used feature selection methods to determine which groups of metrics were the most effective at predicting grasp selection (Section III-D). We also tested the metric groups with noise to simulate real-world behavior of the metrics (Section III-E). Finally, we tested our best noise-trained model on real world shapes (Section III-F).

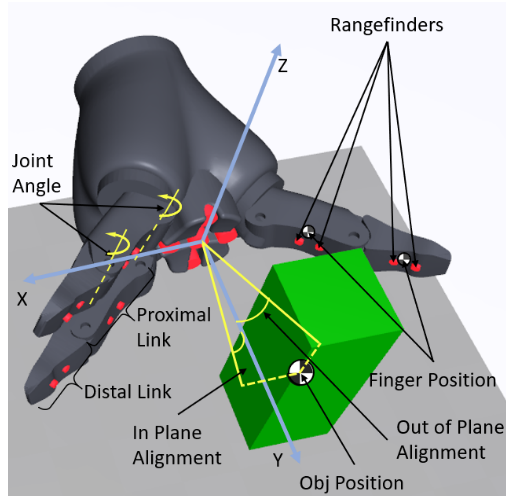

Fig. 2: All locations are relative to a coordinate system centered on the palm. The object position is the center of the object. We place two range finders on each link and five on the palm, shown as red dots in this image.

A. Methods: Metrics

Our choice of metrics is motivated by three constraints. First, it should be feasible to implement the sensors on a real hand and/or calculate them with minimal reliance on computer-vision. Second, the metrics should change smoothly as the hand-object configuration changes. Third, the metrics should characterize the shape of the space *between* the hand and the object — not just the pose of the object relative to the hand. This last constraint is motivated by the fact that many grasps fail because of the fingers contacting the object at different times, which de-stabilizes the object *before* balancing contact forces can be applied.

The metrics are described in Table I and shown in Figure 2. They are grouped by type and by their location on the hand (palm, proximal, or distal link). Object location, size, and pose are similar to existing metrics, except we define them relative to the coordinate system of the palm. The range finders and distance metrics measure the space between the object and the hand; the rangefinder directly casts a ray to the object (time-of-flight sensor), while the distance metric — together with the object size — are a more indirect measure of the time to contact. Altogether, we have 19 distinct feature groups built from 76 total individual metric values.

Table I includes Location from Rangefinder and Object

TABLE I: Metric and Feature Group Description

Metric	Feature Groups	Description
Finger Position	• Proximal Link • Distal Link	X,Y,Z position of the center of the link
Object Position		X,Y,Z position of the center of the object
Hand Position		X,Y,Z position of the hand relative to its starting position
Joint Angle	• Proximal Joint • Distal Joint	Angle of the joint
Object Size		X,Y and Z dimensions
Finger Object Distance	• Proximal Link • Distal Link	Distance to center of object from center of finger link
Object-Palm Alignment	• In-plane Angle • Out-of-plane Angle	Angle of deviation of the object position measured from the Y axis about the X and Z axes
Rangefinder	• Palm • Proximal Link • Distal Link	Ray-cast distance to object
Gravity Vector		The direction of gravity
Location from Rangefinder		Approximate object position from rangefinder metrics
Object Ratio	• Side • Top	Ratios of the size of the object to the open space on the top and side of the hand

Location from Rangefinder and Object Ratio are derived from existing metrics.

Ratio as separate metrics, even though they are both derived from other metrics. We did this to determine if the classifier would be more accurate if it was trained on data with lower dimensions rather than the raw data. The in-plane and out-of-plane angles were found using the formulae below where Obj_X, Obj_Y and Obj_Z are the X, Y, and Z position of the object respectively.

$$\text{In-plane} = cos^{-1}(\frac{[Obj_X, Obj_Y, 0] * [0, 1, 0]}{|[Obj_X, Obj_Y, 0]|}) \quad (1)$$

$$\text{Out-of-plane} = cos^{-1}(\frac{[0, Obj_Y, Obj_Z] * [0, 1, 0]}{|[0, Obj_Y, Obj_Z]|}) \quad (2)$$

B. Methods: Object and Hand pose Test Set

We collected a data set utilizing a large variety of object shapes, sizes and hand approach angles (Figure 1) to ensure robustness. We initiated grasps from three different primary starting configurations, which we label as normal, top-down and rotated. We added noise both to the hand's pose (adding up to ± 5 degrees tilt in all axes) and to the finger starting angles. Finger joint noise was added by closing the fingers at different speeds and randomly moving the fingers in or out.

The objects used were a rectangular prism, cylinder, two bowls (rectangular and rounded), two different bottles and two vases with different cross sections (see Figure 1). The bowls, bottles, and vases were chosen to be similar to the shapes used in Goins [1] and Rubert [6] to make our results comparable to theirs. In addition, we used 3 different sizes of each shape, labeled as small, medium and large. The medium and small shapes were scaled down 15% and 30% from the large shape, respectively. In total we used $8 * 3 = 24$ shapes with three primary hand positions, for a total of 72 unique grasps, each of which were randomly tweaked with slight changes to the hand position and orientation to generate additional data points.

C. Methods: Data Collection

For each generated hand and object pose, we needed to both label the grasp (successful or not) and produce metrics for when the fingers were *near* the object (less than 5 cm away), but not quite grasping. To do this, We pick an initial hand-object pose, add noise to the orientation and object position, then run a simulation where the hand first attempts to grasp the object, then to lift it.

More specifically, we try to grasp each object/size pair at 5000 randomly generated object positions and hand orientations. Across 9 shapes and 3 orientations, this resulted in 135,000 total grasp attempts. To add noise to the finger angles the fingers were set to close with random speeds 80% of the time and move completely randomly (open or close) the other 20% of the time. After 1 second of motion, the state was saved and the hand would try to lift the object by moving straight up. If the object reached a height of 20 cm above the table, the grasp was labeled a success and the metrics from before lifting were recorded. Using this approach we ended up with 24% positive examples and 76% negative examples. This is beneficial because it is better that a classifier be biased towards bad grasps to prevent accidents caused by moving before a firm grasp is established.

D. Methods: Feature Selection

We used backward elimination to remove the least-contributing feature group until only one group remained. We performed elimination on groups — rather than individual metrics — both for efficiency and because all metrics in one feature group would be collected by the same sensor. We created 19 models with 1 different feature group omitted from each model. The feature groups contained in the model with the maximum Area Under the Curve (AUC) were used as the starting point for the next iteration of model creation. Figure 3 shows this process with n feature groups remaining. Although we used Area Under the Curve (AUC) for the backwards elimination criterion, we also tracked accuracy; the results were similar. We tested the best-performing model 15 times in order to generate means and standard deviations.

We then applied forward selection, starting with the remaining feature group, to ensure re-addition of any necessary features that may have been eliminated prematurely. This was implemented using a similar procedure as backward elimination. The last two remaining feature groups with and without rangefinder were used to seed forward feature selection. This was continued until there was less than a 1% increase between additions.

Since rangefinder metrics are both the most numerous and more difficult to implement in the real world than our

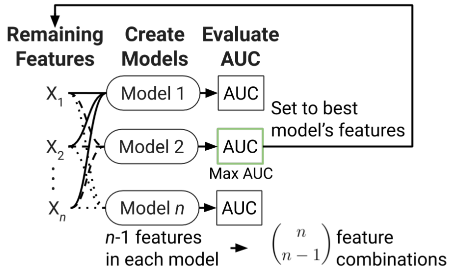

Fig. 3: Backwards elimination with n feature groups remaining. First, n models are created with 1 of the feature groups omitted, then the models are evaluated based on their AUC. The feature groups contained within the best performing model are used for the next iteration.

Fig. 4: Real world test setup. The ArUco markers were used to measure the finger, object and palm location and the box was used to tell if grasps were successful or not.

other metrics, we performed a separate cycle of backward elimination and forward selection on a set of feature groups that did not include the rangefinder metrics.

For this and future sections, we used random forest classifiers (20 trees) in PyTorch [23]. We also attempted Neural Networks, but found that they trained far slower for the same performance. For this reason, we used random forest classifiers for all our experiments.

The best performing set of metrics are referred to as the reduced input space in the remainder of this paper.

E. Methods: Noise Generation

In order to model real world sensor errors and imperfect input data, we performed two different tests with noise added to the grasp metrics. In the first comparison we trained on a model with noise steadily added to both the training and the test sets. In the second comparison noise was only introduced to the metrics of the test set.

The "noisy" metrics m'_k, were calculated from the original metric values m_k using the Equation 3 below. v_k is the difference between the maximum and minimum range of variation from metric k in the training set and p is the magnitude of noise added:

$$m'_k(p) = m_k + \text{rand}(-0.5, 0.5)v_k p \tag{3}$$

Percentage p was set to increments of 2.5%. Each value m_k was clamped in order to insure that the new values were not outside of a realistic case (eg. Rangefinder values should never be less than 0).

F. Methods: Real-world Evaluation

After testing and training in simulation, we evaluated our results on a physical Kinova JACO 2 with a three finger gripper. We attempted to grasp four shapes (Cube, Cylinder, Vase, Hourglass) of two sizes (small and large) for 10 different hand-object poses (normal orientation), then recorded the metrics and the grasp success. This built a database of 80 grasps (51 successful, 29 unsuccessful) to validate our classifier's performance.

Tests were performed by moving the hand to a position near the object, closing the fingers, then lifting the hand and moving it over a box on the edge of the table. If the object was moved to the box, the grasp was considered to be a success. This setup is shown in Figure 4

We used an overhead camera with ArUco markers [24] placed on each of the fingers and the object to implement the metrics. These markers gave us the position of the fingers, object and hand, which were used to calculate the joint angles, finger object distance and in/out-of-plane alignment. The object size was added in based on the shape being tested, but could easily be found from image data.

We chose to use an overhead camera to implement the sensors because we did not want to modify the Kinova gripper. However, these metrics could all be determined via sensors that would not require an overhead camera. The finger position and joint states could be collected from sensors built into the hand, or by tracking the end-poses of the fingers using inertial measurement units (IMUs). The object position, object size and object-palm alignment could be collected from a small hand-mounted camera, and the remaining metrics from time of flight sensors.

IV. Results

In this section, we present results on which metrics and feature groups were the most effective (Section IV-A) followed by a noise analysis. The noise analysis focuses on the groups that were either highly effective or ineffective (Section IV-B). Lastly, we discuss our results on real world test data (Section IV-C).

In summary, with only three feature groups (12 metrics total) we were able to achieve $\approx$ 96% accuracy with a random forest classifier.

A. Results: Feature Selection

In Figure 5 we plot the best-performing model's AUC and accuracy at each iteration. None of the forward additions increased model AUC or accuracy.

Rangefinders on the distal link were the last feature group to remain from the full set, while finger object distance distals were the last to remain when rangefinder metrics were removed. This indicates that the most important information is how far the fingertips are from the object, which is to be expected since they are usually the part of the hand that makes contact with the object. Table II summarizes the accuracy for the last three groups in the process. For both cases, object position and size were kept, along with distance information on the distal ends of the fingers. The standard deviation of the accuracy per iteration was negligible (maximum 0.145%). A confusion matrix for the random forest trained on 21 metrics is given in Table III.

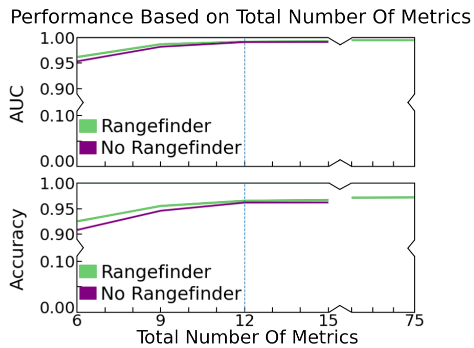

Fig. 5: Figure 5 displays the AUC and accuracy as feature groups are removed. The vertical line marks the performance with only 3 feature groups (12 metrics total) from both sets of experiments. Models with and without rangefinder achieved 96.7 ± .1% and 96.2 ± .1% accuracy respectively.

TABLE II: Accuracy and AUC of Random Forests Trained on Reduced Feature Groups

Total Metrics	Features Groups	AUC	Accuracy
	With Rangefinder		
18	• Obj Pos • Rangefinder Distals • Obj Size • Finger Obj Dist Distals	0.994±.0	96.9±.1%
12	• Obj Pos • Rangefinder Distals • Obj Size	0.992±.1	96.7±.1%
9	• Obj Pos • Rangefinder Distals	0.987±.1	95.5±.1%
	Without Rangefinder		
21	• Obj Pos • Finger Obj Dist Distals • Obj Size • Finger Pos Dist	0.992±.0	96.3±.1%
12	• Obj Pos • Finger Obj Dist Distals • Obj Size	0.991±.1	96.2±.1%
9	• Obj Pos • Finger Obj Dist Distals	0.982±.1	94.6±.1%

TABLE III: Confusion Matrix for Testing on Simulation Data

	Good Grasp	Bad Grasp
Predicted Good	92.6 ± .4%	2.2 ± .1%
Predicted Bad	8.8 ± .3%	97.8 ± .1%

B. Results: Noise Generation

Figure 6 shows the results of the two noise tests for selected sets of feature groups, each run ten times and plotted at the mean with error bars. The sets were chosen based on their performance in the backwards elimination (kept vs eliminated early). Broadly speaking, robustness to noise was correlated with the number of metrics in the set. The blue horizontal line marks the "always assume a bad grasp" policy. Up to 25% noise introduction, all classifiers trained with noise outperform this policy.

Notably, in the train without noise plot, all of the classifiers that incorporated the rangefinder as a key grasp metric suffered a 10% accuracy drop at only 5% untrained noise. We hypothesize that the rangefinder data (which has 17 metrics) was over-fitting the data.

C. Results: Physical Testing

We tested our results on the physical kinova with the grasp classifier trained on the object position, finger object distance distals, object size and finger position distals. On the collected data, our simulation trained network had a maximum accuracy of 76%. The accuracy of all tested networks is given in Figure 7.

While this accuracy is lower than our simulation accuracy, this is likely due to differences between simulation and the real world. When we tested the same starting positions in simulation, 40% had different outcomes.In addition, our training set was 76% failed grasps, while the real world data was 36% failed grasps. This likely preconditioned the network to expect a failed grasp.

V. Discussion

There are potentially a wide variety of sensors that can be used to understand the interaction between the hand and the object. In particular, we are interested in sensors that can operate in near-contact situations and with limited data. Using simulation, we have evaluated these metrics for predicting grasp successes with a variety of poses and objects.

At around 96% accuracy on the objects we tested, the results from backwards elimination (shown in Figure 5) provides evidence that only 12 metrics are sufficient for grasp classification. Compared to the performance of grasp classifiers with a similar number of metrics, our classifier achieves higher accuracy without contact locations and contact forces [1], [6]. Moreover, these metrics are feasible to implement with time of flight sensors and simple image processing.

For the reduced input spaces, both with and without the rangefinder data, 6 of the 12 metrics contain object size and position, providing a general location of where it is in space. The remaining 6 metrics — finger object distance distals or rangefinder distals — provide a measurement of how close the tips of the fingers are to the object in space. Conceptually, this *is* the right minimal information needed to evaluate grasps on the types of shapes we used.

Our experiments indicate that these 12 metrics are the basis of grasp classification for the objects tested and the other metrics provide redundant information. For example, in-plane angle, out-of-plane angle, and object ratios can all be approximated from a combination of the 12 metrics. From

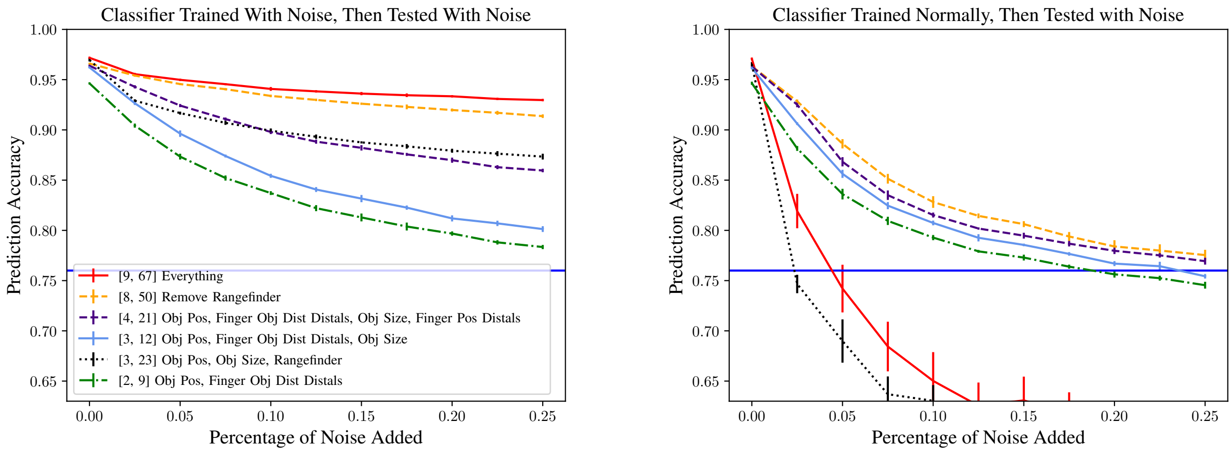

Fig. 6: Left: Testing and training with noise. Right: Just testing with noise. Legend policies ordered by feature groups and length of grasp metrics. The blue horizontal line in each plot indicates the accuracy of an "assumed bad grasp" policy.

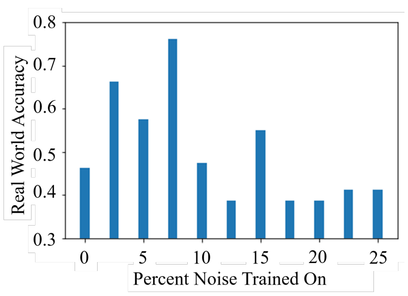

Fig. 7: Accuracy is highest when trained on mild noise

Table II, adding more than 12 metrics increases accuracy by less than 1%. It is likely that they contain the vast majority of the information for classifying a grasp and that adding more metrics only accounts for noise.

When testing for noise, we also found that these metrics (Object Position, Finger Object Distance Distals, Object Size, and Finger Position Distals) are robust to noise, making them good candidates for the transition from simulation to real world grasping. Furthermore, there is evidence that the increased size of the rangefinder metric groups may make them susceptible to over-fitting.

From the real world tests, it is interesting that the best performing network was trained on 7.5% noise. We suspect that this is because the Aruco markers were not accurate, and in general were off by about the same amount. This indicates that training on noisy data is useful for real world performance if the noise added in simulation matches the noise present in the sensors.

Although simulation does not always accurately reflect reality, our tests show that it is feasible to use simulation to perform a "culling" of possible metrics to train an initial classifier. Given the challenges and costs of physical implementation of sensors — and the costs of collecting real-world training data — we believe this approach can be useful to focus development on specific sensor types.

Note that our approach does not exclude using tactile sensors in conjunction with the metrics outlined here. We have chosen not to include them in our analysis for two reasons. First, they are only active when the fingers are actually in contact, and we are explicitly looking for effective near-contact metrics. Second, the behavior of these sensors is (currently) not well-modeled in existing physical simulators because of the complex physical contact interactions.

Limitations: We have only evaluated this approach with a single robot hand (Kinova arm). This is, in part, because we wanted to validate our approach on a physical hand, and we currently have limited access to robotic hardware. Although this hand is emblematic of 3-fingered grippers, more testing is needed to determine if these results will generalize to other hands (and other objects). In particular, our objects were mostly symmetric — asymmetric objects may require more nuanced metrics.

VI. Conclusion and Future Work

We found that, in general, information about the relationship between the fingertips and the object is more important than information about the relationship between the base of the finger and the object. In addition, our noise testing indicates that relatively simple metric based classifiers can be made resilient to noise, and that this can improve its performance on real world data.

Future work should focus on collecting additional, more accurate real world test data and apply the methods here to additional hands. Our classifier could also be used in grasp planning by using it as a heuristic to pick actions. In addition, this problem is an interesting avenue for transfer learning, as it could be used to both improve the real world performance and to broaden the applicability of the classifier to additional hands.

Acknowledgment

This work supported in part by NSF grants CCRI 1925715 and RI 1911050.

REFERENCES

[1] A. K. Goins, "Improving robotic grasping performance using machine learning techniques," 2014.

[2] X. Yan, J. Hsu, M. Khansari, Y. Bai, A. Pathak, A. Gupta, J. Davidson, and H. Lee, "Learning 6-dof grasping interaction via deep geometry-aware 3d representations," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 3766–3773.

[3] J. Kwiatkowski, D. Cockburn, and V. Duchaine, "Grasp stability assessment through the fusion of proprioception and tactile signals using convolutional neural networks," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 286–292.

[4] J. Varley, J. Weisz, J. Weiss, and P. Allen, "Generating multi-fingered robotic grasps via deep learning," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015, pp. 4415–4420.

[5] Y. Chebotar, K. Hausman, Z. Su, G. S. Sukhatme, and S. Schaal, "Self-supervised regrasping using spatio-temporal tactile features and reinforcement learning," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016, pp. 1960–1966.

[6] C. Rubert, D. Kappler, J. Bohg, and A. Morales, "Predicting grasp success in the real world - a study of quality metrics and human assessment," *Robotics and Autonomous Systems*, vol. 121, p. 103274, 09 2019.

[7] J. Mahler, M. Matl, V. Satish, M. Danielczuk, B. DeRose, S. McKinley, and K. Goldberg, "Learning ambidextrous robot grasping policies," *Science Robotics*, vol. 4, no. 26, p. eaau4984, 2019.

[8] G. M. Bone and Yonghui Du, "Multi-metric comparison of optimal 2d grasp planning algorithms," in *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No.01CH37164)*, vol. 3, 2001, pp. 3061–3066 vol.3.

[9] A. T. Miller and P. K. Allen, "Examples of 3d grasp quality computations," in *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No.99CH36288C)*, vol. 2, 1999, pp. 1240–1246 vol.2.

[10] C. Rubert, D. Kappler, A. Morales, S. Schaal, and J. Bohg, "On the relevance of grasp metrics for predicting grasp success," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 265–272.

[11] R. Balasubramanian, L. Xu, P. D. Brook, J. R. Smith, and Y. Matsuoka, "Human-guided grasp measures improve grasp robustness on physical robot," in *2010 IEEE International Conference on Robotics and Automation*, 2010, pp. 2294–2301.

[12] ——, "Physical human interactive guidance: Identifying grasping principles from human-planned grasps," *IEEE Transactions on Robotics*, vol. 28, no. 4, pp. 899–910, 2012.

[13] Junggon Kim, K. Iwamoto, J. J. Kuffner, Y. Ota, and N. S. Pollard, "Physically-based grasp quality evaluation under uncertainty," in *2012 IEEE International Conference on Robotics and Automation*, 2012, pp. 3258–3263.

[14] Y. Zheng, "An efficient algorithm for a grasp quality measure," *IEEE Transactions on Robotics*, vol. 29, no. 2, pp. 579–585, 2013.

[15] S. Liu and S. Carpin, "Partial convex hull algorithms for efficient grasp quality evaluation," *Robot. Auton. Syst.*, vol. 86, no. C, p. 57–69, Dec. 2016. [Online]. Available: https://doi.org/10.1016/j.robot.2016.09.004

[16] Y. Zheng and W.-H. Qian, "Coping with the grasping uncertainties in force-closure analysis," *The International Journal of Robotics Research*, vol. 24, no. 4, pp. 311–327, 2005. [Online]. Available: https://doi.org/10.1177/0278364905049469

[17] E. Dessalene, Y. H. Ong, J. Morrow, R. Balasubramanian, and C. Grimm, "Using geometric features to represent near-contact behavior in robotic grasping," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 2772–2777.

[18] A. t. Pas and R. Platt, *Localizing Handle-Like Grasp Affordances in 3D Point Clouds*. Cham: Springer International Publishing, 2016, pp. 623–638. [Online]. Available: https://doi.org/10.1007/978-3-319-23778-7_41

[19] C. Gabellieri, F. Angelini, V. Arapi, A. Palleschi, M. G. Catalano, G. Grioli, L. Pallottino, A. Bicchi, M. Bianchi, and M. Garabini, "Grasp it like a pro: Grasp of unknown objects with robotic hands based on skilled human expertise," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 2808–2815, 2020.

[20] J. Venter and A. M. Mazid, "Tactile sensor based intelligent grasping system," in *2017 IEEE International Conference on Mechatronics (ICM)*, 2017, pp. 303–308.

[21] S. Varkey and C. Achy, "Learning robotic grasp using visual-tactile model," in *2018 International Conference on Circuits and Systems in Digital Enterprise Technology (ICCSDET)*, 2018, pp. 1–3.

[22] M. I. Tiwana, S. J. Redmond, and N. H. Lovell, "A review of tactile sensing technologies with applications in biomedical engineering," *Sensors and Actuators A: Physical*, vol. 179, pp. 17 – 31, 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0924424712001641

[23] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché Buc, E. Fox, and R. Garnett, Eds. Curran Associates, Inc., 2019, pp. 8026–8037. [Online]. Available: http://papers.nips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

[24] R. Munoz-Salinas, "Aruco: a minimal library for augmented reality applications based on opencv," *Universidad de Córdoba*, 2012.