Formulation and Validation of an Intuitive Quality Measure for Antipodal Grasp Pose Evaluation

Tian Tan, Redwan Alqasemi, Rajiv Dubey, and Sudeep Sarkar

***Abstract*—This paper describes a novel grasp quality measure that we developed for evaluating antipodal grasp poses in real-time. To quantify the grasp quality, we compute a set of object movement features from analyzing the interaction between the gripper and the object's projections in the image space. The normalization and weights of the features are tuned to make practical and intuitive grasp quality predictions. To evaluate our grasp quality measure, we conducted a real robot grasping experiment with 1000 robot grasp trials on 10 household objects to examine the relationship between our grasp scores and the actual robot grasping results. The results show that the average grasp success rate increases, and the average amount of undesired object movement decreases as the calculated grasp score increases. We achieved a 100% grasp success rate from 100 grasps of the 10 objects when using our grasp quality measure in planning top quality grasps. In addition, we compared our quality measure with the Q measure and deep learning-based quality measures.**

***Index Terms*—Computer vision for automation, Contact Modeling, grasping.**

I. Introduction

GENERATING grasp pose candidates and evaluating their qualities are core problems in grasp planning. In this work, we focus on addressing the antipodal grasp pose evaluation problem. Existing approaches to this problem can be divided into two categories, analytical [1]–[3] and data-driven [4], [5]. The analytical approaches extract various hand-crafted features from analyzing the input sensory data and calculating an overall quality score. On the other hand, data-driven approaches use auto-generated features. The grasp quality can be found by either mapping features to a quality score or comparing against known-quality grasps in the feature space.

Our grasp quality measure was developed using the analytical approach. We designed two low-level visual features that capture the grasp region's geometric properties, two high-level features that indicate target missing and collision, and four high-level quality features that measure the object movement during gripper closing. The overall quality score is calculated as the weighted sum of the normalized feature values. Fig. 1 shows the flowchart of the evaluation process. Throughout this paper, the "object movement" we mention is the object movement caused by gripper closing.

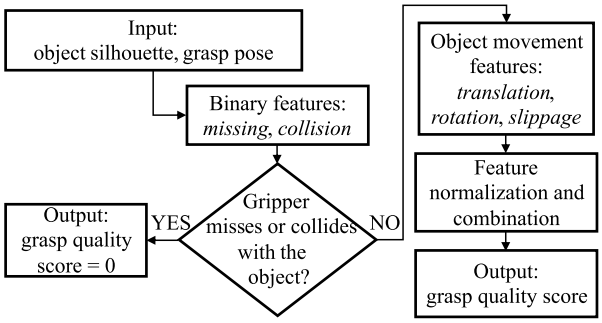

Fig. 1. Grasp quality evaluation flowchart.

Most researchers evaluate grasp planning systems only based on the success rate in picking up objects and overlook how much the gripper has moved the objects before they are grasped. In this work, we would like to emphasize the importance of considering object movement in grasp planning because it affects the interaction between the grasped object and its surroundings, the contents inside the grasped object, and the object's pose feasibility in the post-grasp task. As our grasp quality features are designed based on the object movement predictions, the grasp planning system using our quality measure yields more reliable grasp poses than other existing grasp planning systems. Please note that, even though we compute our quality measure in the image space, we can use it to evaluate antipodal grasp poses in the 3D space. We can take advantage of the parallel-jaw gripper's grasping mechanism to simplify the grasp evaluation problem from 6D (x, y, z, yaw, pitch, roll) to 3D (x, y, roll). We perform the simplification by projecting both the gripper and the object orthogonally to an image plane in the direction of the gripper's z-axis (which is defined by the gripper's yaw and pitch). When projecting the 3D object to a 2D silhouette, the change of the object's cross-sections, in the projection direction, should be considered as in [[6], Fig.6]. The projected silhouette should be able to represent the object's surface area of contact. The main contribution of our work is presenting a quality measure that provides a more accurate evaluation of antipodal grasp poses than other existing methods.

II. Related Work and Background

The existing approaches for grasp evaluation include data-driven and analytical approaches. The most successful data-

Manuscript received: February, 23, 2021; Revised May, 28, 2021; Accepted June, 29, 2021.

This paper was recommended for publication by Editor Hong Liu upon evaluation of the Associate Editor and Reviewers' comments. This work was supported by the National Science Foundation (NSF).

The first three authors are with the Department of Mechanical Engineering, University of South Florida, Tampa, FL, USA (tiantan@usf.edu; ralqasemi@gmail.com; dubey@usf.edu).

The fourth author is with the Department of Computer Science and Engineering, University of South Florida, Tampa, FL, USA (sarkar@usf.edu).

Digital Object Identifier (DOI): IEEE staff will add the DOI here.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LRA.2021.3096192, IEEE Robotics and Automation Letters

driven approaches are deep learning approaches. The most popular deep learning approaches [7]–[10] can achieve an 80-100% grasp success rate in grasping everyday household items. However, the grasp success rate alone is insufficient to validate the grasp network's auto-generated grasp quality measure since there can be significant quality differences between successful grasps. Grasp detection networks are suitable for such tasks as bin picking and storing/retrieving non-fragile goods but not reliable enough to be used alone on assistive robots.

There are two sub-categories of analytical approaches, the contact geometry-based and the contact wrench-based. The contact geometry-based approaches typically evaluate grasp quality by examining if the contact region's geometric properties/features are consistent with the designed criteria. Davidson [11] and Vahedi [12] presented geometric caging methods for finding immobilizing grasps. Calli [13] used the object contour curvature derived from its contour function to help determine graspable regions on the object contour. Blake [14] exploited the properties of object contour local reflectional and rotational symmetry. These contact geometry-based approaches often have many limitations due to the lack of a complete design of grasp quality features.

The contact wrench-based approaches judge the grasp quality by analyzing the contact wrenches (forces and torques) acting on the object [2]. Most of the contact wrench-based methods are built upon the grasp force closure property [15]. Nguyen [16] presented a method to construct force closure grasps, which is one of the representative works in the field. Ferrari and Canny [17] developed the most popular grasp quality measure(often referred to as Q measure) for force closure grasps. The Q measure is a measure of how well a grasp can resist external disturbances. It is currently the most versatile and reliable analytical grasp evaluation method. Many variants of the Q measure have been developed [18]–[21] to improve different aspects of the method. Besides the Q measure, other contact wrench-based approaches [22] often have many restrictions similar to the geometry-based approaches. In the method comparison section, we provide more details in comparing our grasp quality measure, Q measure, and the embedded quality measure in grasp detection networks.

III. Grasp Quality Evaluation

A. Grasp Representation

We established mathematical models for both the gripper and the object projections in the image to evaluate the gripper-object interaction. An object's projection is a silhouette image of the object, which is mathematically a binary matrix. Each entry of this matrix links to a pixel of the object silhouette image, and the entry is 1 or 0 as the corresponding pixel belongs to the object or background, respectively. As for the gripper projection, we designed a different grasp representation in the image space. Instead of describing an antipodal grasp as a rectangle, we treat it as a set of line segments, $\{gl_1, gl_2, \ldots, gl_n\}$, as shown in Fig. 2. Therefore, the gripper projection can be discretely modeled as a set of line segment functions. Let

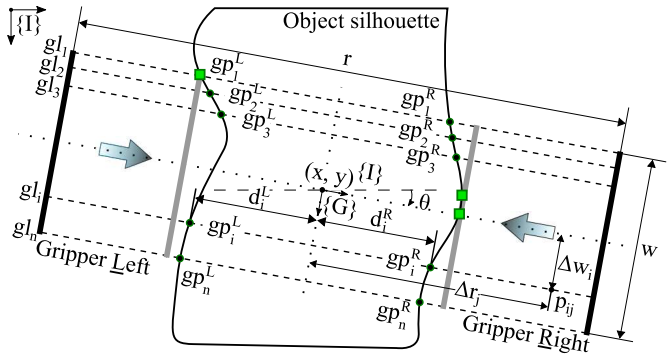

Fig. 2. Illustration of grasp lines(gl_i), grasp points(gp_i^e), grasp distances(d_i^e), contact points and grasp pose parameters(x, y, θ, r, w, n). The thicker lines labeled as L and R represent the left and right gripper fingers' contact lines. The arrow indicates the gripper closing direction.

point P_{ij} be the j^{th} point on grasp line *i* (gl_i), then the function of gl_i can be expressed as the coordinates of $P_{ij}(P_{ij}^x, P_{ij}^y)$ in the image frame {I}:

$$P_{ij}^x = x - \Delta w_i \sin\theta + \Delta r_j \cos\theta \tag{1}$$

$$P_{ij}^y = y + \Delta w_i \cos\theta + \Delta r_j \sin\theta \tag{2}$$

$$\Delta w_i = \frac{w(2i - n - 1)}{2(n - 1)} \tag{3}$$

Where $i = 1, 2, \ldots, n$ is the index of a grasp line. Δw_i is the distance from the center of the gripper projection to the i^{th} grasp line. Δr_j is the distance from a grasp line's center to its point j. Also, (Δr_j, Δw_i) is the coordinate of P_{ij} in the gripper projection frame {G}. Δr_j is positive if P_{ij} is on the right side of the grasp line center, and negative if otherwise ($\Delta r_j \in [-\frac{r}{2}, \frac{r}{2}]$). The parameters of this grasp representation, (x, y, θ, w, r, n), are composed of the planar center location (x, y), planar orientation (θ), gripper finger width(w), gripper open width(r), and the number of grasp lines ($n \in [1, w]$) of the gripper projection in the image coordinates.

B. Terminology

Here we introduce some terminologies and annotations to help describe the gripper-object interaction and better explain our algorithm. We use superscripts L and R to differentiate the gripper's left and right sides in our equations. We use superscript e to indicate that the term is side specific, and it should be replaced by either L or R. The intersections of the grasp lines and the object outline are referred to as *grasp points* of the object, such as gp_i^e ($i = 1, 2, ..., n$) in Fig. 2. The distances between grasp points and the center points of the corresponding grasp line segments are referred to as the *grasp distances* (d_i^e) of those grasp points. When the gripper closes along the grasp line, some of the grasp points will contact the gripper, and these grasp points become *contact points*. The green square shape grasp points in Fig. 2 are the contact points corresponding to that grasp pose. The collection of grasp points is the *grasp region*, and the collection of contact points is the *contact region*.

C. Low-level Visual Features

The visual features are the grasp region profile vector (GRPV) and the contact region feature vector (CRFV), both of which carry information about the geometry and the contact properties of the object contour inside the grasp region. The GRPV is a vector of grasp distances of all grasp lines.

$$GRPV = \begin{bmatrix} GRPV^L \\ GRPV^R \end{bmatrix} = \begin{bmatrix} d_1^L & d_2^L & \dots & d_n^L \\ d_1^R & d_2^R & \dots & d_n^R \end{bmatrix} \quad (4)$$

We find the grasp distances from searching the left and right outermost intersections of the object outline and the grasp lines. With the parameters of the input grasp pose (x, y, θ, w, r, n), we calculate the searching points using (1) and (2) with Δr_j as the only variable. When we find the left and right outermost intersections, their corresponding searching variables are the grasp distances of the grasp line. We check the searching points' pixel values to find intersections. An intersection point must satisfy two conditions: (a) its pixel value is 1, (b) at least one of its two neighboring points on the same grasp line has a pixel value of 0. I.e., the pixel values of an intersection (underlined) and its two neighboring points must be [0, $\underline{1}$, 1] (left contact), [1, $\underline{1}$, 0] (right contact), or [0, $\underline{1}$, 0] (left and right contacts). To ensure the intersections found are the outermost intersections, we start searching from the outermost point of each side. For each grasp line, if any one of the endpoints has a pixel value of 1, then the gripper will collide with the object, the *collision indicator (I_c)* will be set to 1, and we can skip extracting other features. After collision checking, we first perform a coarse searching that uses d_{step} as step size to find a point of pixel value 1 on each side of the grasp line. Then we switch to a fine backward searching with step size 1 to find the contact point. The maximum number of searches in the coarse and fine searching processes are $\frac{r}{d_{step}}$ and $d_{step}-1$, respectively. To minimize the maximum number of searches, we use $d_{step} = \sqrt{r}$. If no grasp point found after searching all grasp lines, the GRPV will be an empty vector, and we set the *missing indicator (I_m)* to 1. I_c and I_m are 0s by default, and this is the only case that we need to calculate the high-level movement features.

After calculating the GRPV, we can define the left and right *primary contact points* as the grasp points with maximum absolute grasp distances in the left and right grasp regions, respectively. If the grasp distances' difference between a grasp point and the primary contact point of the same side is below a threshold, this grasp point is a *secondary contact point* on that side. Adding secondary contact points into consideration decreases the effect of false contact prediction caused by imperfect object silhouette detection. Once the contact points are found, we can calculate their normals and center offsets and organize them in the CRFV:

$$CRFV = \begin{bmatrix} CRFV^L \\ CRFV^R \end{bmatrix} = \begin{bmatrix} CPN^L \\ CPD^L \\ CPN^R \\ CPD^R \end{bmatrix} \quad (5)$$

CPN^e and CPD^e are the vectors of contact point normals and contact point center offsets, respectively. The contact point center offset equals to the distance from the gripper center to the grasp line that the contact point is on, which is the Δw_i in (3).

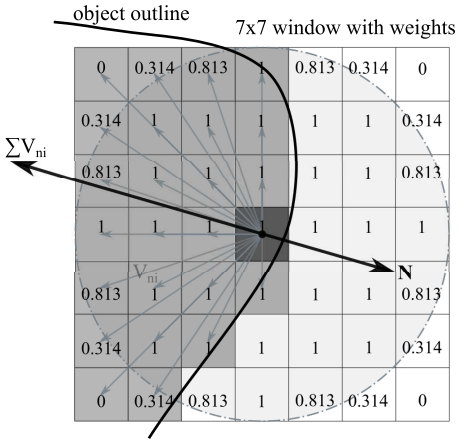

Fig. 3. The visualization of object silhouette boundary pixel normal approximation, N is the pixel normal, V_{ni}'s are the neighboring vectors.

To extract the normals of contact points, we developed a surface normal approximation method for binarized objects. As shown in Fig. 3, the normal of an object boundary pixel can be approximated by inverting the sum of its non-vacant neighboring vectors. Here, a neighboring vector (V_{ni}) is a vector from the pixel of interest to one of its neighboring pixels, and a neighboring pixel is non-vacant if its pixel value equals 1. Moreover, the shape of the neighborhood is an essential factor affecting the approximation result, and it should be a circle since the neighbors in a circle are evenly distributed in all directions. Therefore, we assign weights to the square window for the neighborhood to form a circular shape. In mathematical terms, this method can be expressed as:

$$N = normalize(-(\sum_i^m \sum_j^m M_{xij}, \sum_i^m \sum_j^m M_{yij})) \quad (6)$$

$$M_x = M_{x0} \circ M_p \circ M_w \quad (7)$$

$$M_y = M_{y0} \circ M_p \circ M_w \quad (8)$$

Where M_{x0} and M_{y0} are, respectively, the matrices of the original x and y coordinates of the corresponding pixels in a window of size $m \times m$ and centered at the point of interest. M_p is the matrix of pixel values of the pixels inside the window, which is a binary masking matrix. M_w is the weight matrix of the window. M_x and M_y are the matrices of the weighted x and y coordinates of the non-zero pixels inside the window. The sum of all entries in M_x and M_y, respectively, are the x and y coordinates of the sum of all non-vacant neighboring vectors, and the normal N is in the opposite direction of this vector. These calculations are performed in the window coordinate system, which has the point of interest as the origin and the same orientation as the image coordinate system. Good normal approximation results are obtained using $m = 7$. This method can also be used to estimate the surface normal of 3D objects represented by voxels.

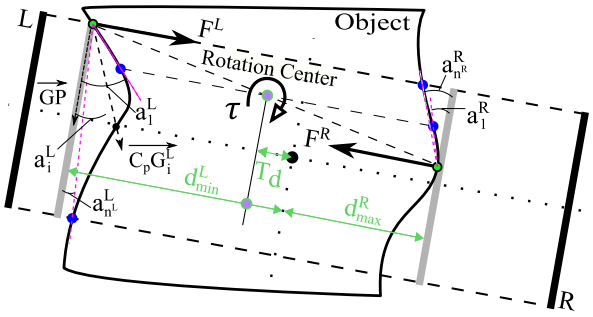

Fig. 4. The object *translation* and *rotation* features that measure the amount of object translation in the gripper closing direction and its planar rotation, respectively.

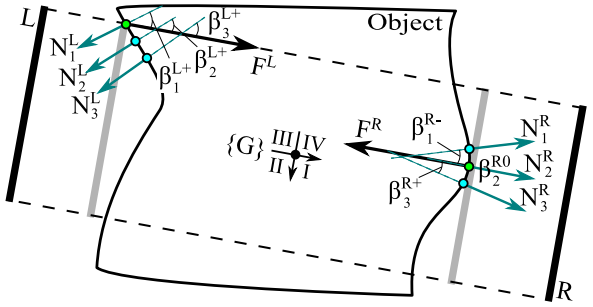

Fig. 5. The type 1 *slippage* predictor that estimates the likelihood of slippage caused by angled contact.

D. High-level Quality Features

Based on the two low-level features, we derived a set of more in-depth features that can be used to quantify the quality of the grasp. These high-level features are the object translation predictor, the object rotation predictor, the type 1 and type 2 object slippage predictors.

The object translation predictor (D_t) estimates the ratio of the object travel distance during gripper closing to half of the gripper open width ($\frac{r}{2}$). As shown in Fig. 4, T_d is the object translation distance in the gripper closing direction. It is measured from the center of the two primary contact points to the grasp rectangle's center. Using GRPV, the ratio of the object translation distance to the gripper half gripper open width is calculated as:

$$D_t = \left| \frac{min(GRPV^L) + max(GRPV^R)}{r} \right| \tag{9}$$

We use the distance ratio instead of the distance as the grasp quality measure because using the ratio allows this feature to adapt to objects and grippers of different sizes.

The object rotation predictor (R_r) predicts the angle of object rotation during grasping. The object rotates when there is torque, and the torque occurs when the two primary contact points are not on the same grasp line. With the two primary contact points, the rotation center and the rotation direction can be determined. The *potential after-rotation contact region (PARCR)* is defined using the center of rotation and the rotation direction.

As shown in Fig. 4, the PARCR on each side is the object outline segment between the two grasp points highlighted in blue. One grasp point is on the same grasp line as the rotation center, and the other is the last grasp point of the region approaching the gripper in the rotation process. Once the PARCRs are determined, we find the angle of rotation by selecting the minimum required rotation angle for a grasp point in the PARCR to be the new contact point after rotation. Geometrically, the rotation angle (a_i) is defined as the angle between the gripper plate and the line that connects the primary contact point (approximate pivot point) and the new after-rotation contact point. This can mathematically be expressed as:

$$R_r = min(a_1^L, a_2^L, \ldots, a_{n^L}^L, a_1^R, a_2^R, \ldots, a_{n^R}^R) \tag{10}$$

Where a_i^e's are the rotation angles. Each rotation angle is calculated as the angle between the gripper plate vector $\overrightarrow{GP}$ and $\overrightarrow{C_pG_i^e}$ the vector from the primary contact point to the i^{th} grasp point in the PARCR.

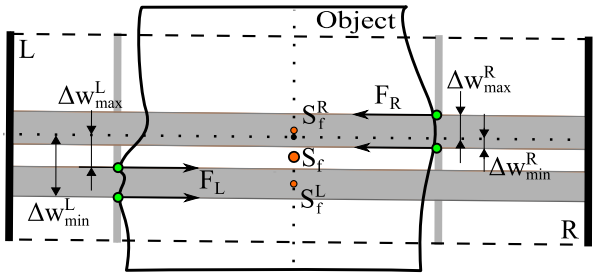

Fig. 6. The type 2 *slippage* predictor that estimates the likelihood of slippage caused by the offset of overall contact force to the center of gripper finger. The shaded areas are the force zones.

The type 1 slippage predictor (S_a) is a feature that measures how likely the object will slip during grasping due to the slope of the contact surfaces. This feature is defined as the average *contact angle* of the most slippery contact region. As shown in Fig. 5, the contact angle ($\beta \in [-90, 90]$) is defined as the angle between the contact force ($\mathbf{F}$) and the contact point's inward normal ($-\mathbf{N}_i$). If the contact point normal ($\mathbf{N}_i$) points towards the 1^{st} and 2^{nd} quadrant of the gripper frame {G}, then the contact angle is positive; and it is negative otherwise. If one contact region's contact angles have the same polarity, then the contact region is slippery, and the slipperiness is indicated by the average of all the contact angles. The larger the average contact angle, the more likely the object will slip. The following equations describe this slippage prediction:

$$\beta_i^e = \begin{cases} \angle(\mathbf{F}^e, -\mathbf{N}_i^e) & \mathbf{N}_{iy}^e > 0 \\ -\angle(\mathbf{F}^e, -\mathbf{N}_i^e) & \mathbf{N}_{iy}^e < 0 \end{cases} \tag{11}$$

$$SI^e = \sum_{i=1}^{n^e} \left|\beta_i^e\right| - \left|\sum_{i=1}^{n^e} \beta_i^e\right| \tag{12}$$

$$S_a = \begin{cases} \frac{1}{2}\left(\left|\frac{\sum_{i=1}^{n^L} \beta_i^L}{n^L}\right| + \left|\frac{\sum_{i=1}^{n^R} \beta_i^R}{n^R}\right|\right) & SI^e = 0 \\ 0 & SI^e \neq 0 \end{cases} \tag{13}$$

Where N_{iy}^e is the y coordinate of the normal N_i^e in {G}, and n^e is the number of contact points of the e side of the gripper. SI^e is the slippage indicator. When SI^e is 0, slippage

is likely to happen because all contact angles of the same side have the same polarity.

The type 2 slippage predictor (S_f) is a feature that predicts slippage through the grasp force placement. This feature favors grasp forces that are balanced and centered on the gripper contact surface. It is calculated as the average of the left and right force zone center offsets.

As shown in Fig. 6, the force zone of one side of the gripper is the continuous region that contains all contact points of that side. The center offset is the distance from the center of the force zone to the gripper's center. Using CPD^e, this feature can be calculated as:

$$S_f^e = \frac{min(CPD^e) + max(CPD^e)}{2} \tag{14}$$

$$S_f = \frac{S_f^L + S_f^R}{2} \tag{15}$$

Where $min(CPD^e)$ and $max(CPD^e)$ are the Δw^e_{min} and Δw^e_{max} respectively. Note that we do not directly consider friction force in predicting slippage. However, our slippage predictors are related to how the gripper forces are applied to the object. Minimizing the slippage predictors' values maximizes the friction forces applied to the object.

E. Grasp Quality Scoring

The quality features are in different units and scales. Before combining them, we first normalize them with linear functions defined by two endpoints (0, 1) and (τ_i, 0), where the x-coordinate is the feature value and the y-coordinate is the normalized feature score. τ_i is the threshold feature value for the 0 feature score. Since D_t is a ratio, its threshold value is 1. R_r and S_a are measures of angles, and their thresholds are set to 60 degrees based on our experiments. S_f is the force placement feature, its value range from 0 to the gripper projection's half-width ($\frac{w}{2}$). We used two feature thresholds $\frac{w}{4}$ and $\frac{w}{2}$ for feature scores 0.7 and 0, respectively, which makes the function has two different slopes when normalizing low and high feature values. We found this yields more practical quality scores than a single slope function. After feature normalization, we calculate the grasp quality measure S as the weighted sum of the feature scores:

$$S = ks_{min} + (1-k)s_{o^-} \tag{16}$$

Where s_{min} is the minimum feature score, and s_{o^-} is the average of other feature scores. Since the grasp pose's quality mainly depends on its worst quality feature, the weight k should be assigned in a minimum-dominant fashion ($k >> 0.5$). It can be empirically determined by specifying the grasp quality to the desired value when $s_{min} = 0$ and $s_{o^-} = 1$ (in this paper, we used $k = 0.9$).

IV. REAL ROBOT GRASPING EXPERIMENT

A. Experiment Setup

We used the Baxter robot [23] from Rethink Robotics as our test platform. The computations were programmed in Python and performed on a laptop PC running Ubuntu 18.04 with a 2.2 GHz Intel Core i7-8750H CPU, 8 GB of RAM, and an NVIDIA GeForce GTX 1060 graphics card. The graphics card was only used for computing the objects' silhouettes, and other computations are all performed on the CPU. We used a parallel jaw gripper with a pair of narrow fingers(1.3 cm wide). We masked the fingers' rubber contact surfaces with scotch tape to reduce friction so that slipping is more likely to happen when the grasp pose is prone to slippage. Fig. 7 shows the hardware and the 10 objects used in the grasping experiment.

Fig. 7. The robot gripper and the objects used in the robot grasping experiment (top). Some of the objects from the Cornell and DexNet grasping dataset used in the method comparison test (bottom).

This experiment examines how our calculated grasp quality score relates to the actual robot grasping performance. In the experiment, we used an eye-to-hand camera (Intel Realsense L515) to locate the object. Because controlling the gripper pitch and yaw is outside the scope of this work, we set them as -90 and 0 degrees, respectively, to form top-down grasps. Then, we used an eye-in-hand camera (Baxter hand camera) to take a closer shot at the object from the top. Facebook Detectron2 [24] was used to extract the object's silhouette from the close shot object image. Assuming the gripper was at the object's location, we calculated the gripper's projection in the object silhouette image using calibrated camera intrinsics and the object's distance to the camera. With the object and gripper projections, we randomly generated grasp candidates within the object's bounding box. Then we evaluated them with our quality measure to select the one with the desired quality score for robot execution. Randomly generated grasp poses help prove our quality measure's robustness since to get the desired pose, we need to evaluate hundreds and thousands of candidates.

To test the whole grasp score value domain, we categorized the grasp scores into ten score levels, 0-0.1, 0.1-0.2, …, 0.9-1. We performed ten grasp trials within each score level for each object, which is 1000 total grasps. The number of evaluated grasp candidates, the time used in grasp evaluation, and the robot grasp outcome were recorded for each grasp trial. The robot grasp outcome includes: (1) a binary term that indicates if the grasp is successful, and (2) two hand camera images

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LRA.2021.3096192, IEEE Robotics and Automation Letters

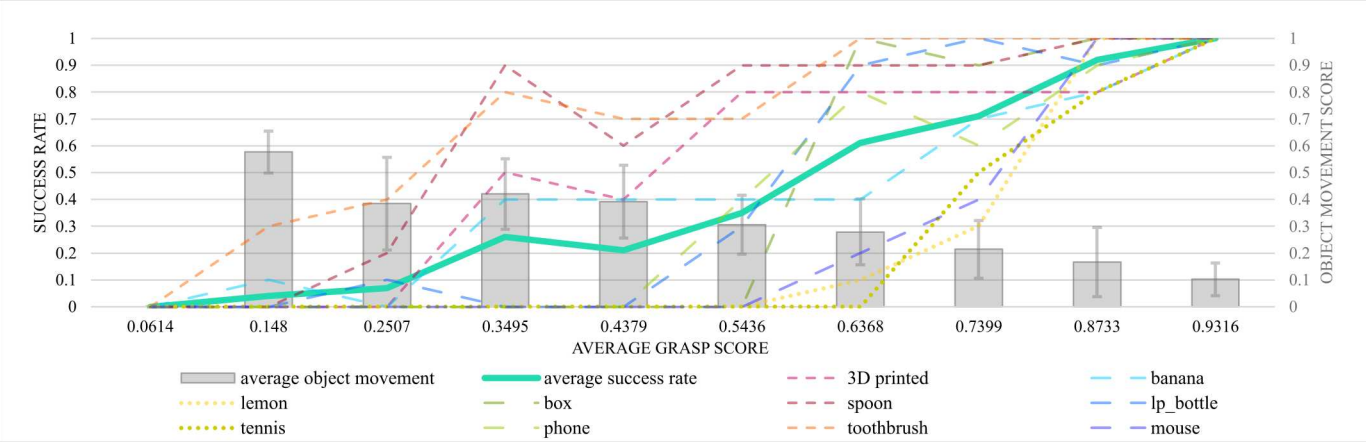

Fig. 8. The success rate and the object movement score of the robot grasping experiment vs. the grasp score.

(BA images) taken right before and after the gripper closes, which capture the object's movement during gripper closing. If the robot can lift the object for 2 seconds and put it back without visible slippage, the grasp is recorded as successful. The object's rotation and its translation in the gripper closing direction were measured as the key lines' orientation change and the key points' position change, respectively, in the BA images. We drew the key lines using the object's appearance features that are visible in both BA images. We drew the position key point as the object's center on the gripper's center grasp line in each of the BA images.

B. Results

In terms of the computational cost, the recorded average grasp generation time for the 1000 grasps is 0.534 s. 261868 grasp candidates were evaluated to generate those grasps; the average grasp evaluation time is 2.04 ms, with a standard deviation of 0.264 ms.

We analyzed the accuracy of our quality measure through the results of our robot grasping experiment, as shown in Fig. 8. The left y-axis is the success rate; the right y-axis is the normalized object movement score. Because the objects' movements in failed grasp trials are unpredictable and not helpful in evaluating our method, we ONLY measured the objects' movements in succeeded grasp trials to reveal the quality difference among those successful grasp poses. The object movement score is calculated as the average of the normalized object rotation and translation (in the gripper closing direction). The thresholds used for normalizing the object translation and rotation are 100 pixels and 60 degrees, respectively.

The x-axis is the average grasp score. The thin non-solid lines show each object's average success rate at each score level. The thick solid line shows the average success rate, and the bars show the average object movement of all objects at each score level. To help interpret the results, we classified our test objects into four categories based on their geometrical properties. As shown in Fig. 7, objects in categories 1 and 2 both have flat contact surfaces. However, the graspable parts of category-1 objects are much thinner than category-2 objects. Category-4 objects have curved contact surfaces, and category-3 objects have flat and curved contact surfaces.

From the object-wise success rate, we can see that objects in the same category tend to have similar success rates under the same score level. Also, category-1 objects and category-4 objects have the highest and lowest overall grasp success rate, respectively. The category-1 objects are long and thin and have flat contact surfaces in the projection, making them the easiest for parallel-jaw grippers to grasp, while category-4 objects are the hardest to grasp because they are round and our gripper fingers are narrow. The category-1 objects and the banana have relatively high success rates even when the grasp qualities are low. This result does not disprove our quality measure because the object movements in those grasps are very high. Therefore, even though the grasps were successful, they were evaluated as low quality because they were expected to move the objects a lot.

Despite the differences between different objects' success rates in the medium quality score levels, all objects' success rates merge to the same points at the highest and lowest score levels. This observation indicates that our quality measure can distinguish good and bad grasps regardless of the object type. We achieved a 100% success rate from grasping the ten objects 100 times with grasps of an average score of 0.93 and an average movement score of 0.1 with a standard deviation of 0.0618. Overall, the average grasp success rate increases, and the average object movement score decreases as the quality score increases.

V. METHOD COMPARISON

Now that the real robot experiment has shown that our quality measure is effective, in this section, we compare the effectiveness of our quality measure with other grasp quality measures through grasp planning results comparison.

A. Implementation of the Grasp Planning Systems

We have implemented 4 grasp planning systems for this comparison, which includes 2 analytical systems based on our quality measure and the Q measure [17], and 2 deep learning-based systems, the FC-GQCNN [25] and the GGCNN [10].

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LRA.2021.3096192, IEEE Robotics and Automation Letters

For the grasp planning system using our quality measure, we used a normally distributed random search around the object's geometric center to find a high-quality grasp pose (quality score > 0.95). If no such high-quality pose is found, then we perform a uniformly distributed small region random search around the grasp pose with the maximum score to find a pseudo local optimum. We used the same searching method for the Q measure-based grasp planning system, except that we did not stop searching at a quality score threshold. Instead, we keep searching until we find the pseudo global optimum. We used 0.8 and 1 as the friction coefficient and the torque scaling factor, respectively, in evaluating grasp poses' Q measure scores.

The deep learning-based grasp planning systems are end-to-end networks, which do not require a separate grasp pose candidates sampling system. This makes them more efficient than the analytical systems. However, they are often much harder to implement than the analytical ones in real-world applications since they usually need to be re-trained to adapt to different working environment setups. For this comparison, we used the pre-trained models named FC-GQCNN-4.0-PJ and GGCNN for the FC-GQCNN and the GGCNN methods, respectively, and we used their training datasets as test inputs.

B. Grasp Planning Dataset and Comparison Procedure

The dataset we used for this grasp planning comparison is comprised of 100 single-object images that we extracted from the DexNet 2.0 grasping dataset (synthetic) [7] and the Cornell grasping dataset (real-world) [26]. From the DexNet dataset, we randomly selected 50 images; from the Cornell grasping dataset, we manually selected 50 images with good object silhouette detection results (this rules out the impact of flawed inputs on the grasp planning results). We tested the two analytical grasp planning systems with all of the 100 images, the FC-GQCNN and the GGCNN methods with the DexNet subset and the Cornell subset, respectively. Some example objects are shown in Fig. 7.

After obtaining the grasp planning results, we created an anonymous online survey (USF IRB# Pro00040871) of 100 questions (one for each input image). For each question, the participants see 3 copies of the input image, and each image shows one grasp planning result (drawn on the image as a rectangle) from the three grasp planning methods tested on that image. Then the participants are asked to compare and evaluate the quality of each grasp pose (without knowing which pose is from which planning method) based on a 5-point Likert scale. We explicitly instructed the participants to score the grasp poses using the following criteria:

1 - The worst case as the robot gripper fingers will collide with the object, or the gripper will miss the object.

2 - Better than 1 as the grasp does not miss and there is no collision, but the robot still has a very low chance of picking up the object.

3 - Better than 2 as the robot has a very high chance of picking up the object. However, there will be a large amount of object movement.

4 - Better than 3 as the robot can pick up the object, and there is only a small amount of object movement.

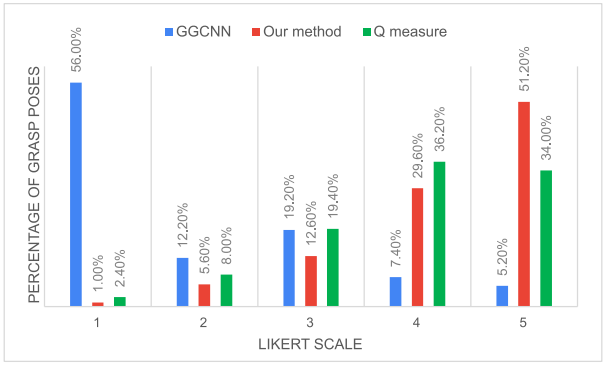

Fig. 9. The results of grasp planning on the Cornell sub-dataset using GGCNN, our method, and the Q measure. 1 and 5 indicate the worst and best quality grasp poses, respectively.

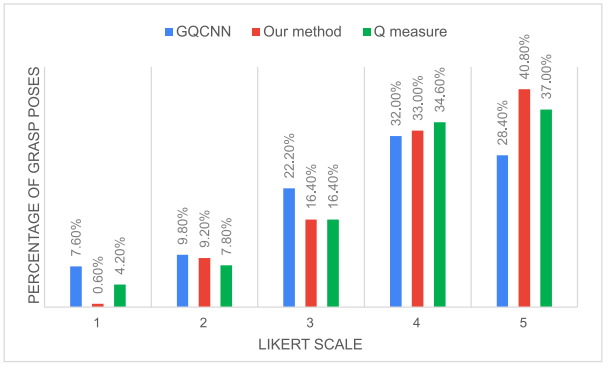

Fig. 10. The results of grasp planning on the DexNet sub-dataset using FC-GQCNN, our method, and the Q measure. 1 and 5 indicate the worst and best quality grasp poses, respectively.

5 - The best grasp pose as the robot can pick up the object, and there is almost no object movement.

This survey aims to use the human evaluation as a baseline to compare the results of different grasp planning methods, which can indicate the effectiveness of the corresponding grasp quality measures.

C. Survey Results and Discussions

Ten subjects participated in our anonymous survey. Fig. 9 and 10 show the percentage of the planned grasp poses based on their human-evaluated Likert score. For each dataset, the resulting grasp poses percentages are calculated based on 500 human evaluations (50 instances x 10 participants). These results show that the grasp planning system using our quality measure generates significantly more top-quality grasp poses (scored as 5) and less poor-quality grasp poses (scored as 1 or 2). In addition, we can see that both analytical grasp planning systems work almost consistently across datasets. There is a non-negligible percentage reduction of the top-quality grasp poses in the DexNet dataset results, which is as expected since the objects in this dataset are significantly more complex than those in the Cornell dataset. Despite the performance difference between the two grasp planning networks, both generated the largest number of poor quality poses, and the smallest number of top quality poses.

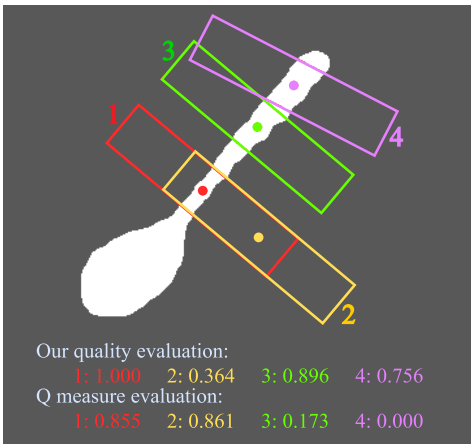

Fig. 11. Comparison of our quality measure and the Q measure in evaluating grasp poses.

As the Q measure grasp planning system has similar performance to our method, we present a more detailed individual case comparison to highlight the difference between the two quality measures. As shown in Fig. 11, there are 4 grasp poses evaluated by both quality measures. Firstly, the Q measure cannot identify the quality difference between grasps 1 and 2 because it evaluates grasp quality by grasp contact wrenches, and grasp 1 and 2 have nearly identical contact wrenches. Secondly, comparing the two same-quality grasps 1 and 3, we can see that the Q measure is more sensitive to flawed input. Because of the imperfect object silhouette, our quality evaluation has a score change of 10.4%, while the Q measure has a score change of 79.8%. Lastly, the score of grasp 4 shows that the Q measure cannot evaluate grasp poses that have contact forces outside of the predetermined contact friction cone.

VI. CONCLUSIONS

This paper presents the detailed design and experimental results of our novel grasp evaluation method, which calculates grasp poses' quality by analyzing the interactions between the gripper and the object through their projections in the image space. The real robot grasping results show that our grasp quality measure is practical and intuitive. And through method comparison, we show that the grasp planning system using our quality measure outperforms the other three grasp planning systems in generating grasp poses that cause minimum object movements in the gripper closing phase. Although the presented quality measure addressed only parallel-jaw grippers, future work will include expansion to multi-fingered grippers.

REFERENCES

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

[2] A. Bicchi and V. Kumar, "Robotic grasping and contact: A review," in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*, vol. 1. IEEE, 2000, pp. 348–353.

[3] M. A. Roa and R. Suárez, "Grasp quality measures: review and performance," *Autonomous robots*, vol. 38, no. 1, pp. 65–88, 2015.

[4] L. E. Zhang, M. Ciocarlie, and K. Hsiao, "Grasp evaluation with graspable feature matching," in *RSS Workshop on Mobile Manipulation: Learning to Manipulate*, 2011.

[5] S. Caldera, A. Rassau, and D. Chai, "Review of deep learning methods in robotic grasp detection," *Multimodal Technologies and Interaction*, vol. 2, no. 3, p. 57, 2018.

[6] Y. Domae, H. Okuda, Y. Taguchi, K. Sumi, and T. Hirai, "Fast graspability evaluation on single depth maps for bin picking with general grippers," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 1997–2004.

[7] J. Mahler, J. Liang, S. Niyaz, M. Laskey, R. Doan, X. Liu, J. A. Ojea, and K. Goldberg, "Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics," *arXiv preprint arXiv:1703.09312*, 2017.

[8] 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, 2019.

[9] U. Viereck, A. Pas, K. Saenko, and R. Platt, "Learning a visuomotor controller for real world robotic grasping using simulated depth images," in *Conference on Robot Learning*. PMLR, 2017, pp. 291–300.

[10] D. Morrison, P. Corke, and J. Leitner, "Closing the loop for robotic grasping: A real-time, generative grasp synthesis approach," *arXiv preprint arXiv:1804.05172*, 2018.

[11] C. Davidson and A. Blake, "Error-tolerant visual planning of planar grasp," in *Sixth International Conference on Computer Vision (IEEE Cat. No. 98CH36271)*. IEEE, 1998, pp. 911–916.

[12] M. Vahedi and A. F. van der Stappen, "Caging polygons with two and three fingers," *The International Journal of Robotics Research*, vol. 27, no. 11-12, pp. 1308–1324, 2008.

[13] B. Calli, M. Wisse, and P. Jonker, "Grasping of unknown objects via curvature maximization using active vision," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2011, pp. 995–1001.

[14] A. Blake, M. Taylor, and A. Cox, "Grasping visual symmetry," in *1993 (4th) International Conference on Computer Vision*. IEEE, 1993, pp. 724–733.

[15] X. Markenscoff, L. Ni, and C. H. Papadimitriou, "The geometry of grasping," *The International Journal of Robotics Research*, vol. 9, no. 1, pp. 61–74, 1990.

[16] V.-D. Nguyen, "Constructing force-closure grasps," *The International Journal of Robotics Research*, vol. 7, no. 3, pp. 3–16, 1988.

[17] C. Ferrari and J. F. Canny, "Planning optimal grasps." in *ICRA*, vol. 3, no. 4, 1992, p. 6.

[18] F. T. Pokorny and D. Kragic, "Classical grasp quality evaluation: New algorithms and theory," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2013, pp. 3493–3500.

[19] S. Liu and S. Carpin, "A fast algorithm for grasp quality evaluation using the object wrench space," in *2015 IEEE International Conference on Automation Science and Engineering (CASE)*. IEEE, 2015, pp. 558–563.

[20] Y. Zheng, "Computing the best grasp in a discrete point set with wrench-oriented grasp quality measures," *Autonomous Robots*, vol. 43, no. 4, pp. 1041–1062, 2019.

[21] M. Pozzi, A. M. Sundaram, M. Malvezzi, D. Prattichizzo, and M. A. Roa, "Grasp quality evaluation in underactuated robotic hands," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 1946–1953.

[22] S. Liu, Z. Hu, H. Zhang, M. Kwon, Z. Wang, Y. Xu, and S. Carpin, "Grasp quality evaluation and planning for objects with negative curvature," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 2223–2229.

[23] R. Robotics, "Baxter," https://github.com/RethinkRobotics/sdk-docs/wiki/Baxter-Overview.

[24] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick, "Detectron2," https://github.com/facebookresearch/detectron2, 2019.

[25] V. Satish, J. Mahler, and K. Goldberg, "On-policy dataset synthesis for learning robot grasping policies using fully convolutional deep networks," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1357–1364, 2019.

[26] I. Lenz, H. Lee, and A. Saxena, "Deep learning for detecting robotic grasps," *The International Journal of Robotics Research*, vol. 34, no. 4-5, pp. 705–724, 2015.