Adaptive Autonomous Grasp Selection via Pairwise Ranking

David Kent* and Russell Toris†

Abstract—Autonomous grasp selection for robot pick-and-place applications makes use of either empirical methods leveraging object databases, which generate grasps for specific objects at the initial cost of modeling effort, or analytical methods, which generalize to novel objects but fail on object subsets that require specific grasping strategies not captured by the algorithm. We introduce a grasp selection algorithm that ranks grasp candidates with a set of grasp metrics augmented with object features, creating an approach that adapts its strategies based on user-specified grasp preferences. We formulate grasp selection as a pairwise ranking problem, which significantly reduces data collection compared to traditional grasp ranking methods and generalizes to novel objects. Our approach outperforms a state-of-the-art grasp calculation baseline and a pointwise ranking formulation of the same problem.

I. INTRODUCTION

Fully autonomous pick-and-place functionality for mobile manipulators is difficult to realize, due to the variety of objects that a robot may encounter. We focus specifically on the problem of teaching a robot to grasp a large and diverse set of objects, all of which may not be known in advance, with minimal time and effort on the part of a teacher. There are two approaches to this problem, each with their own advantages and disadvantages. On the one hand, empirical methods that leverage object models as prior knowledge produce reliable grasps for specific objects. They do not scale to unstructured environments (homes, public spaces), where the complete object set cannot be predicted. In practice, they also do not scale to large structured environments (warehouses, retail stores), where the number of objects to train can require prohibitive effort. On the other hand, analytical methods that directly calculate end-effector poses without prior object models grasp objects in general, but fail on subsets of objects that require specific grasping strategies.

In this work we develop a grasp selection algorithm to fuse analytical grasp calculation with the object-specific advantages of empirical approaches, without compromising generalizability. Our approach involves collecting and leveraging grasp preferences, which requires a human in the loop to specify ground truth preferences as training data. The preferences can generalize to novel objects, and as such our approach does not explicitly require additional training data; it works out-of-the-box as a standalone autonomous grasp calculator with some baseline competence.

With efficient data collection in mind, we base our approach on work in an analogous area: the ranking problem of document retrieval [1]. We equate document retrieval’s search query to a target pick-and-place object. Instead of ranking retrieved documents, we rank grasp pose candidates calculated from the target object’s point cloud. The document ranking problem has been studied extensively, with pairwise ranking being a useful approach to our grasp ranking problem. Pairwise ranking generates many training instances from a single instance of user input, simply selecting a preferred item from a list. Further, when using appropriate features, pairwise ranking can generalize to new items.

We present a pairwise ranking approach to autonomous grasp selection for robot pick-and-place. We characterize grasps using a novel object-centric extension to the grasp metrics used in the point-and-click approach presented in [2]. Our novel formulation of grasp selection as a pairwise ranking problem uses these grasp metrics as feature vectors, augmented with object features for context, which can generalize to novel objects. As pairwise ranking has not previously been applied to grasp selection, we perform an analysis of binary classification models over our pairwise grasp data. Additionally, we discuss data collection for our pairwise ranking model, which is included as a component of the Fetch Programming by Demonstration interface [†]. Finally, we show that our pairwise ranking approach outperforms a state-of-the-art grasp calculation baseline and a pointwise ranking formulation of the same problem.

II. RELATED WORK

With the advent of 3D sensing, empirical approaches to pick-and-place leveraging object model databases proliferated. These approaches are well-suited to tailoring specific grasp strategies to specific objects. Grasps for each model can be precomputed with simulators such as GraspIt! [3] or OpenRAVE [4], learned autonomously from trial-and-error [5], or learned from human demonstrators [6], [7]. With an appropriate database of models and associated grasps, grasp planning is reduced to either object recognition with 6-DoF pose estimation or shape matching. The object models themselves can come from existing shape databases [8], be defined as CAD models [9], or be constructed directly from depth data [10], [11], [12].

The empirical methods discussed above assume that every interactable object is modeled, which, while achievable in controlled environments, does not hold in unstructured settings. Further, even if the full set of objects is known, any objects not already in the database require additional modeling effort. Shape-based methods provide generalization to novel objects by decomposing them into parts [6], or by

*Georgia Institute of Technology dekent@gatech.edu
†Fetch Robotics, Inc. rtoris@fetchrobotics.com

†The fetch_pbd package is available at https://github.com/fetchrobotics/fetch_pbd
generating grasps over local regions [13], but still require additional modeling effort or training for objects that do not contain the database’s known shapes.

Analytical grasp calculators forego model databases in favor of more recent approaches that seek to identify reliable grasps on any object. This can be accomplished by sampling over depth data to locate hand-engineered grasp features [14], [15], [16], [17], ranking sampled grasp poses with human-inspired heuristics [18], [2], discovering grasp features through physical trial-and-error [19], or learning reliable grasp features on adversarial synthetic data [20]. These approaches have also proved effective, with the advantage of being immediately deployable in new environments, but they lack the ability to adapt to outlier objects for which the predefined grasp features, heuristics, or learned features fail.

An ensemble approach can mitigate the disadvantages of both methods, accounting for grasps returned from both shape matching and analytical calculation [21]. The result is a system that prioritizes grasps from good shape matches, while calculating new grasps when shape matching fails. This method requires maintaining a traditional model database, however. As an alternative to an ensemble approach, we propose fusing the object-focused advantages of empirical methods with the generalizability of analytical methods in one algorithm, without explicit object modeling. We accomplish this by treating grasp calculation as a ranking problem. Ranking problems found in information retrieval provide a useful analogy for grasp ranking. In particular we are interested in the object ranking problem, a class of preference learning [22]. Given a search query, object ranking seeks to determine an ordering of returned items, with only a subset of exemplary pairwise preferences given as training data. Common object ranking problems include web page ranking [23], [24], [25], document retrieval [25], [26], [27], [28], image retrieval [29], and recommender systems [30], [27]. More recently the approach has been applied successfully outside of information retrieval to scheduling [31], where the query is a set of observations over current tasks, and the items are the set of scheduling decisions at a current time step. As we will present in this work, we take a similar approach to grasp ranking for pick-and-place, where we consider the search query to be the object itself.

The ranking problem has been formulated with three distinct approaches: pointwise methods that directly calculate a score for each item, pairwise methods that establish the relative ranking of a pair of items, and listwise methods that minimize loss over a complete list ordering. For a comprehensive discussion, see [1]. We formulate our approach as a pairwise ranking model, the advantages of which are discussed in III-C.

III. APPROACH

In keeping with analytical grasp calculation, the goal of our grasp selection algorithm is to calculate and rank a set of grasps by their likely success rate from input depth data. We accomplish this with three steps: grasp sampling to identify initial candidate grasps, grasp metric calculation to describe each grasp’s relationship to the object-of-interest and the surrounding environment, and pairwise ranking to determine an ordering of the grasp candidates under context derived from the object-of-interest. We detail each step below.

A. Grasp Sampling

Our approach does not require a specific grasp sampling strategy, and is designed for use with any algorithm that can analyze a point cloud to produce a set of candidate grasps. In this work, we use the antipodal grasp sampler included in agile_grasp [15] to populate candidates. In practice any grasp sampler can be used, such as the height accumulated feature sampler of [16], the handle-like sampler of [17], or the antipodal grasp sampler of [20]. We selected AGILE’s sampler as it is available open source, and it is the first step of the full AGILE pipeline, which we use as an established grasp calculation baseline to compare against in Section IV.

B. Grasp Metrics

We adapted the point-and-click approach described in [2] for use specifically in pick-and-place applications. Our adapted approach takes two point clouds as input: an object point cloud \( p_e \) segmented from the scene, and an environment point cloud \( p_c \) constructed by cropping the full scene to a padded bounding box containing \( p_e \). Candidate grasp poses are sampled over \( p_c \) to eliminate grasps in collision with the environment. Using the sampled grasps and the two point clouds, we developed a broader set of metrics that more fully describe the relationship of a candidate grasp to an object-of-interest and the surrounding environment.

The metrics are as follows:

- \( m_1 \): difference in orientation between the grasp approach vector and the normal vector to the dominant plane, fit over \( p_c \). This metric represents grasps perpendicular to large planes in the environment, such as table surfaces or shelf walls.
- \( m_2 \): difference in orientation between the grasp approach vector and the normal to a plane fit over a local region of \( p_e \) centered at the grasp point. This represents grasps perpendicular to the object-of-interest.
- \( m_3 \): difference in orientation between the grasp orientation and the principal directions of \( p_e \), computed by principal component analysis, i.e. grasps aligned with the object-of-interest.
- \( m_4 \): distance from the grasp point to the center of \( p_e \), approximating grasps near the object center of mass.
- \( m_5 \): distance from the grasp point to the nearest point in \( p_c \). This mainly serves to deter grasps erroneously sampled from the environment rather than the object, but also differentiates grasps centered on an object point from grasps that are offset from the object.

Each metric is normalized on \([0, 1]\). For a grasp \( g_i \), we can calculate a combined score \( m_i \) to approximate its grasp...
quality using a linear combination of the metrics (lower scores are preferred):

\[ m_i = w_1 m_1^i + w_2 m_2^i + w_3 m_3^i + w_4 m_4^i + w_5 m_5^i, \]

where \( 0 \leq w_i \leq 1 \) and \( \sum w_n = 1 \).

In practice this is not sufficient, as each metric should be used differently depending on the object-of-interest. For example, when grasping a small object such as a pen from a desk, \( m^1 \) should be weighted more heavily than \( m^4 \), since a successful grasp should be perpendicular to the desk (grasping the center of the pen is less important). Conversely \( m^1 \) should contribute less to ranking grasps on a large object, where relationships between the grasp and the object are more important than the orientation of the grasp with respect to the environment. One could learn a different set of weights per object, but this requires extensive training data. As we show below, we achieve the same result with a pairwise ranking model that compares the above grasp metrics differently based on context provided by object features, which additionally improves training data collection and has the ability to generalize to novel objects.

C. Grasp Ranking

Taking the analogous ranking problem of information retrieval, we formulated a pairwise ranking problem by equating the search query to the object-of-interest, and the returned items to the calculated grasps. We describe the pairwise ranking formulation below, discuss its advantages with respect to data collection and training, evaluate classifiers for our application, and present an algorithm that leverages the pairwise classifier to produce grasp rankings.

1) Pairwise Ranking Formulation: For a pair of grasps \( g_i \) and \( g_j \), represented by grasp feature vectors \( x_i = [m_1^i, m_2^i, m_3^i, m_4^i, m_5^i] \) and \( x_j = [m_1^j, m_2^j, m_3^j, m_4^j, m_5^j] \) (the metrics of Section III-B), we calculate a pairwise feature vector by taking the difference of the grasp feature vectors:

\[ \hat{x}_{ij} = x_i - x_j. \]

We append the result to a vector of object features \([f_0, f_1, \ldots, f_n]\) calculated from the object point cloud \( pc_o \), resulting in a context-enhanced pairwise feature vector:

\[ x_{ij} = [f_0, f_1, \ldots, f_n, \hat{x}_{ij}]. \]

In effect, the context-enhanced feature vector allows the grasp ranking method to adapt its ranking strategy to different types of objects differentiated by the object-derived features. In this work, we use \( x_{ij} = [l, a, b, x, y, z, \hat{x}_{ij}] \) as our context-enhanced pairwise feature vector, where \([l, a, b]\) is the average color of \( pc_o \) in the CIELAB color space, and \([x, y, z]\) are the dimensions of the minimum-area bounding box containing \( pc_o \). We chose these features as a simple feature vector that can differentiate the set of objects used in our experiments. Depending on the degree to which the grasping strategy should change per object, more specific features, such as color histograms, point feature histograms, etc., or even object labels, could be used instead.

Algorithm 1 Pairwise Training Data Generation

Require: \( grasps, pc_o, pc_e \)

1: \( s \leftarrow userSelectedGraspIndex(grasps) \)
2: \( x \leftarrow calculateMetrics(grasps, pc_o, pc_e) \)
3: \( f \leftarrow calculateFeatures(pc_o) \)
4: for \( i = 0 : size(grasps) \mid i \neq s \) do
5: \( saveTrainingInstance([f, x_s - x_i], 1) \)
6: \( saveTrainingInstance([f, x_i - x_s], 0) \)
7: end for

With the pairwise feature vector defined, we create a binary classification problem by adding a label \( y_{ij} \) that denotes the ordering of the two grasps used to construct \( x_{ij} \):

\[ (x_{ij}, y_{ij}) \text{ where } y_{ij} = \begin{cases} 1 & \text{if } g_i \prec g_j, \\ 0 & \text{if } g_j \prec g_i, \end{cases} \]

i.e. a label of 1 denotes \( g_i \) should be preferred over \( g_j \), and a label of 0 denotes \( g_j \) should be preferred over \( g_i \). Given training pairs \( (x_{ij}, y_{ij}) \) we train a binary classifier to predict \( y_{ij} \), thus predicting a grasp ordering.

2) Data Collection: Our training data collection does not require grasp demonstration, as users instead perform selection over a list of autonomously calculated grasps. Pairwise training data requires only relative differences in grasp quality, which is advantageous as training examples can be collected by having the user simply select the best grasp from a list, without providing a score or any additional ranking information. Further, a single grasp selection generates \( 2(n - 1) \) pairwise instances from a single click, where \( n \) is the total number of grasps, shown in Algorithm 1.

We collected training data over a diverse set of 15 objects, shown on the left of Figure 1. Training data was generated by having an expert user place an object randomly on a table, look at a visualization of the set of grasp candidates produced by antipodal grasp sampling, and select what they deem to be the best grasp at each instance. We define the “best” grasp as the grasp most likely to stably lift the object. This process was conducted 4 times for each object in the training set.

Fig. 1: (Left) Trained object set for classifier evaluation, model training, and grasp rate evaluation. (Right) Novel object set for generalization experiment.

---

4Our approach can perform other task-specific grasps by giving the expert different instructions as to what constitutes a good grasp; for the scope of this paper we evaluate only pick-and-place tasks, and leave additional task evaluation to future work.
We note that these training instances represent grasp preferences, rather than the absolute best grasp, since we cannot evaluate ground truth success rates for grasps generated on-the-fly. As such, there’s an element of subjectivity present in the training data collection.

3) Model Evaluation: Many binary classification techniques have proven successful for pairwise ranking, including boosting [30], neural networks [24], support vector machines [23], [28], regularized least-squares [26], and decision trees [31]. Since we are applying this method to a new domain, we evaluated a set of models to determine the best binary classifier for pairwise grasp feature vectors. Each model was trained using scikit-learn with cross-validated parameters, on the data from Section III-C.2.

The random forest classifier resulted in the highest pair ordering prediction accuracy over the training data using 10-fold cross-validation, at 0.93 ± 0.01 (the second best classifiers were the neural network and k nearest neighbors, both with an accuracy of 0.91 ± 0.01). The random forest precision-recall curve dominated the precision-recall curves of all other models, shown in Figure 2. The training curve for the random forest classifier, shown in Figure 3, has the ideal characteristics of producing a high classification rate with few training examples, while still improving with additional data. As such, we trained a random forest model over the complete data set described in Section III-C.2 which we use for all evaluation presented in Section IV.

Additionally, we verify that incorporating the object features is beneficial to our model. Figure 3 also includes the training curve calculated over the same training data with object features removed (i.e. using $x_{ij}$ in place of $x_{ij}^\text{o}$. The object features significantly increase classification accuracy, regardless of the number of training instances.

4) Producing a Ranked Grasp List: The final ranked list is produced by voting, shown in Algorithm 2. For every permutation of two grasps, we use the random forest binary classifier to predict a pairwise ordering. The highest ranked grasp is simply the grasp that was ordered first for the greatest number of pairs. The voting formulation increases robustness to misclassifications from the binary classifier, as any misclassification has a chance to be mitigated by classifying the reverse-ordered pair (e.g. if $x_{ij}$ is classified as a false positive, $x_{ji}$ can still be classified as a true positive).

**Algorithm 2 Complete Pairwise Ranking Algorithm**

```
Require: pc_o, pc_e
1: grasps ← sampleAntipodalGrasps(pc_e)
2: x ← calculateMetrics(grasps, pc_o, pc_e)
3: f ← calculateFeatures(pc_o)
4: for all $g_i, g_j$ in grasps | $i \neq j$ do
5:     if classify($f, x_{ij}$) = 1 then
6:         incrementRank($g_i$)
7:     else
8:         incrementRank($g_j$)
9: end if
10: end for
11: return sort(grasps)
```

While classification rate over training and testing data is useful for comparing binary classifiers, what we care about in practice is grasp success rate. We evaluate grasp rates on a trained object set (over which the pairwise classifier was trained) and a set of novel objects to examine the model’s ability to generalize. Additionally, we evaluate a set of difficult-to-grasp objects identified from the two object sets, for better insight into the pairwise method’s advantages.

We compare our pairwise ranking grasp selection approach to a set of analytical baselines. For consistent comparison of the grasp selection algorithms, all methods use the same antipodal grasp sampler to calculate the list of grasp candidates as input. The approaches are:

1) Random: Uniformly select a random antipodal grasp.
TABLE I: Grasp rates for all methods, on each object set, reported as the mean grasp success rate over all trials for the object set ±1 standard error.

<table>
<thead>
<tr>
<th></th>
<th>Trained Set</th>
<th>Novel Set</th>
<th>Difficult Set</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random</td>
<td>0.592 ± 0.073</td>
<td>0.625 ± 0.042</td>
<td>0.361 ± 0.053</td>
</tr>
<tr>
<td>AGILE</td>
<td>0.575 ± 0.067</td>
<td>0.706 ± 0.072</td>
<td>0.438 ± 0.060</td>
</tr>
<tr>
<td>Pointwise</td>
<td>0.779 ± 0.072</td>
<td>0.863 ± 0.045</td>
<td>0.653 ± 0.106</td>
</tr>
<tr>
<td>Pairwise</td>
<td>0.883 ± 0.038</td>
<td>0.938 ± 0.021</td>
<td>0.840 ± 0.051</td>
</tr>
</tbody>
</table>

2) AGILE: Select a grasp returned by the full AGILE pipeline; for details see [15]. We note that this approach diverges from our approach immediately after antipodal grasp sampling. AGILE calculates a set of grasp features using HOG feature encodings of grasp images, which are then classified using an SVM to denote grasp quality. We include AGILE in our results as an established grasp calculation baseline.

3) Pointwise: Select the highest-ranked grasp using a score calculated from Equation 1. We include this approach as a pointwise ranking comparison to our novel pairwise ranking formulation using the same grasp metrics. For our experiment, we uniformly weighted each of the five metrics.

4) Pairwise: Select the highest-ranked grasp using our novel pairwise ranking method summarized by Algorithm 2. The classifier was the binary random forest classifier trained on the full set of grasp preference training data described in Section III-C.2.

To consistently measure grasp rates for every object, we used a Fetch robot [32] to perform 16 controlled grasping trials per object—at 4 preselected testing positions, with 4 preselected orientations at each position. A grasp is considered successful if the robot lifts the object off of the table with the object remaining stably in the gripper. This process was repeated for each approach, for a total of 1600 grasp executions.

A. Grasp Quality on Trained Objects

Our pairwise ranking approach is the most effective method on the trained object set with a grasp rate of 0.883 ± 0.038, as shown in the Trained Set column of Table I outperforming both the random antipodal grasp and the state-of-the-art baseline approaches, as well as our pointwise ranking approach that makes use of the same metrics.

As the grasp rates were calculated from controlled trials, we perform further analysis with a one-way repeated measures Analysis of Variance (ANOVA), where grasp rates measured on the same object with different grasp selection methods were correlated. ANOVA showed a significant effect of grasp selection method for grasp success rate ($F(3, 42) = 14.08, p < 0.0001$). Conducting post tests with Tukey’s HSD test showed the Pairwise method’s grasp rate was a statistically significant improvement over Random ($p < 0.01$) and AGILE ($p < 0.01$), as shown in Figure 4.

B. Generalization to Novel Objects

The Pairwise approach’s biggest disadvantage over the other approaches is the need to train the model with training data collected over a particular object set. This problem drove the decision to append general object features to the pairwise feature vectors, rather than more specific features or object labels, so that the model could to generalize to new objects.

To evaluate this, we performed the same grasping experiment on a new object set, without retraining the random forest classifier for the Pairwise method. The novel objects are shown on the right of Figure 1.

The results, shown in the Novel Set column of Table I still show Pairwise as the best method, with a grasp success rate of 0.938 ± 0.021. As with the trained objects, one-way repeated measures ANOVA shows a statistically significant effect of grasp selection method on grasp success rate ($F(3, 27) = 25.44, p < 0.0001$), with Tukey’s HSD post tests confirming that the Pairwise method’s improvement is statistically significant when compared to Random ($p < 0.01$) and AGILE ($p < 0.01$).

C. Pairwise vs. Pointwise

Although we found no statistically significant difference between the Pairwise and Pointwise approaches for either the trained or novel object set, examining grasp rates for individual objects showed Pairwise’s greatest performance improvements occurred for the most difficult to grasp objects in our object sets. We verify this result by performing an additional ANOVA on grasp rates for difficult objects, selected as any object with a grasp failure rate of over 50% when selecting random antipodal grasp.

*This included the mug, food can, game controller, tongs, squat bottle, hammer, laundry detergent bottle, scrub brush, and cooking utensil box.
post tests showing the Pairwise method significantly outperforms all of the other approaches \((p < 0.01\) for Random and Agile, \(p < 0.05\) for Pointwise). This result supports the original motivation for developing the Pairwise approach, in that through the addition of an efficient training process and a pairwise ranking formulation, autonomous grasp calculation can adapt its grasp selection approach to handle difficult objects that require specific grasping strategies.

V. DISCUSSION AND FUTURE WORK

We have shown that, with minimal training data, our pairwise ranking implementation for autonomous grasp selection outperforms state-of-the-art grasp calculation approaches. We believe the real strength of this method comes from its adaptability. We have shown that the pairwise model trained on an initial object set can generalize to novel objects, resulting in improvements over the set of baseline methods. As such the model can be used effectively for many objects without additional training. It is easy to envision the robot encountering a complex new object that requires a different grasping strategy, however. In such a case the classifier can be updated with training data collected specifically for that object, all with minimal data collection due to the advantages of the pairwise ranking formulation.

We are interested in studying the effects of object feature selection for the context-enhanced pairwise feature vectors. In particular, it would be interesting to explore the tradeoffs between the object features’ effect on the degree to which the grasping preferences change per object and the ability of the algorithm to generalize to novel objects. Additionally, we would like to characterize the approach’s robustness to suboptimal training data by evaluating models trained on data collected from naïve users.

REFERENCES