Grasp Evaluation With Graspable Feature Matching

Li (Emma) Zhang
Department of Computer Science
Rensselaer Polytechnic Institute
Troy, NY
Email: emma.lzhang@gmail.com

Matei Ciocarlie and Kaijen Hsiao
Willow Garage
Menlo Park, CA
Email: matei, hsiao@willowgarage.com

***Abstract*—We present an algorithm that attempts to identify object locations suitable for grasping using a parallel gripper, locations which we refer to as "graspable features". As sensor input, we use point clouds from a depth camera, from which we extract a gripper-sized voxel grid containing the potential grasp area and encoding occupied, empty, and unknown regions of space. The result is then matched against a large set of similar grids obtained from both graspable and non-graspable features computed and labeled using a simulator. While matching, we allow unknown regions of space in the input grid to match either empty or occupied space in our database of graspable feature voxel grids, which enables us to consider many possible geometries for occluded areas in our grasp evaluation. We evaluate our algorithm in simulation, using real-life sensor data of objects as input and evaluating the output using ground-truth object shape and pose.**

I. Introduction

The task of finding an appropriate placement for a robotic hand in order to pick up a desired object, commonly referred to as grasp planning, can be explored using a variety of data types as input. When an object recognition tool is available, coupled with a database of known objects, this input can take the form of a complete object model such a triangle mesh. This is a compelling approach, as grasp planning can be performed off-line, takes advantage of complete object information, and can potentially by complemented by additional semantic information from the database. However, while object identity can be an extremely valuable prior, it is not strictly necessary for grasping: the stability of a grasp is largely determined by the part of the object that the gripper makes contact with. The inertial properties of the object (mass and center of mass location) are still determined by its overall shape and are relevant for the grasping task, but a grasp with a stable local contact area should be able to handle at least some variation of these global parameters.

In this study we focus on the problem of identifying such object locations that enable stable grasps. We refer to them as "graspable features", and base our approach on the following intuitions. First, robotic hands with relatively simple geometry, such as parallel jaw grippers, will also exhibit the least variation in the shape of graspable features they can approach. This provides a tractable entry point to a problem that would be more difficult to approach for dexterous hands. Second, graspable features are likely to be shared between many objects with otherwise different global geometry and appearance, making it easier to generalize results from a limited training data set.

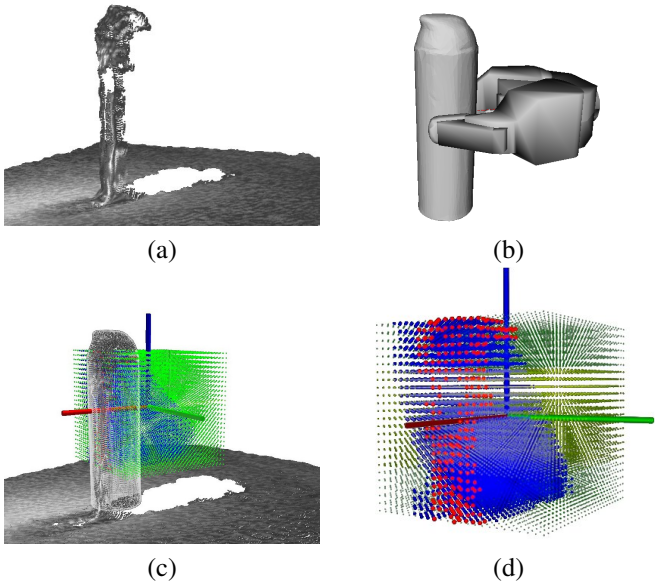

Fig. 1: Graspable feature encoded in a voxel grid. (a) the sensor image of a real object; (b) a grasp for that object computed in simulation using a 3D model of the object; (c) a voxel grid encodes the volume around the object surface at the grasp location, as seen by the depth camera. Red cells are occupied, green are empty and blue are unknown due to occlusions; (d) close-up of the voxel grid.

Since we assume that object identities and complete models are not available on-line, we are interested in identifying graspable features as they appear in real-life robot sensor data, of a type that can be expected to be available for a robot in a human environment. In particular, for a given scene, we use point clouds from a depth camera (such as a laser scanner, stereo camera, etc.). While commonly available for tasks in unstructured environment settings, each view from such a sensor only contains surface information on part of the scene, much of which remains occluded. We convert such point clouds into a volumetric, voxel grid representation of the depth sensor data, which has the advantage of being able to encode the difference between surface areas, empty regions of space, and occluded (and thus unknown) regions.

The core of our approach is a data-driven method, attempting to match a voxel grid extracted from one or more depth images of a potentially graspable feature against a very large set of similar data, extracted and labeled in simulation. In previous work[1], we have introduced a database of object models along with a large number of grasps for a parallel jaw gripper, computed off-line using a simulator. In this study, we augment the database with simulated laser scans of these object models, focusing on the local grasp areas. We then use the resulting dataset to evaluate images of new, potentially graspable features.

II. Related Work

There are several related data-driven grasp planning methods. Glover et al.[3] use probabilistic models of object geometry to complete outlines of objects for grasp planning, and Goldfeder et al.[4] match object geometries with similar models in a database, whose grasps are likely to be useful. Both use the database of objects to generate potential grasps, not to evaluate arbitrary grasps under consideration, and try to match entire objects, not just the parts of objects being grasped.

Other grasp evaluation methods exist; for instance, Saxena et al.[14] estimate the probability of success of a grasp using logistic regression on features based on 2D images and corresponding 3D point clouds; Pelossof et al.[13] use SVMs to evaluate grasps of superquadrics, and Kamon et al.[9] evaluate grasps for parallel-jaw grippers based on features of the object silhouette. Hsiao et al.[6] combine results from multiple object recognizers and multiple grasp evaluators/planners to estimate the probability of success of arbitrary grasps.

A key feature of our method is that it allows unknown regions of our input voxel grid (due to occlusions) to match arbitrary geometries in voxel grids from our database. Other methods exist that attempt to plan grasps based on single/incomplete views of objects, such as [5, 7, 11, 10, 8]; however, all of these make implicit assumptions about the geometry filling the occluded regions, either that the regions are empty, or that smoothness constraints cause a shroud to drop off from known edges. Our method, on the other hand, takes into account many possible geometries filling in the unknown space, and is thus less likely to confidently suggest grasps based on incorrect assumptions about the occluded geometry.

III. Graspable features

A graspable feature is a part of an object's geometry that affords a stable grasp when using a given type of robotic gripper. In this study, we use a simple 1DOF parallel jaw gripper, which minimizes the variance in shape of grasped object locations. Our final goal is to identify graspable features in sensor data acquired by the robot at run-time. The approach presented in this paper aims to tackle the following problem: given a particular sub-area of a sensed 3D object (or collection of objects), estimate how likely it is that the local geometry is a graspable (vs. ungraspable) feature; or in other words, estimate the likelihood of success of the corresponding grasp.

A. Voxel grids

To encode sensor data (Fig. 1a) for a graspable feature, we use a voxel grid defining a volume centered at the possible grasp location (Fig. 1b). While a simple point cloud is a much more compact representation, a volumetric approach has the advantage of being able to encode not only surface area, but also known empty or unknown regions of space (Fig. 1c,d). The dimensions of the voxel grid are just large enough to contain the gripper if a grasp were executed on the object at the given location, as we are interested in characterizing strictly the contact area and the volume inside and around the gripper, rather than global geometry of the object. In this study,the grid dimensions we used were 12cm X 12cm X 12cm.

The sensor data that the voxel grid is constructed from at run-time consists of one or multiple scans of the scene, registered together. For a given candidate grasping position and orientation, we use this data to build the associated voxel grid. Using the method which we will describe below, we compute the "graspable feature" score of the grid, which is then used to decide whether the grasp should be performed or not.

When using real scans, the inside of an object will always show up as unknown, with a relatively thin volume of known occupied points on the object's surface. This representation is thus sensitive to small shifts in camera or object location, with a small change in either of those causing a large part of the occupied cells to shift into either empty or unknown space. To mitigate this problem, we use multiple resolutions of our grids throughout the matching process described below. The downsampling process, while obviously losing some information, can also capture the proximity relationship between neighboring cells. In our study, we use cell dimensions ranging from 5mm to 40mm.

B. Generating the database

Our approach to recognizing voxel grids of graspable features is data-driven, based on matching against a large corpus of data computed and labeled in simulation. The starting point is a database of object models and grasps which we have introduced in a previous study [1]. This database contains 180 3D models of household objects, available from common retailers, and an average of 600 grasps for each object, computed using the *GraspIt!* simulator [12]. Due to computational constraints, in this study we used a subset of 79 object models from the database, and an average of 167 grasps per object.

Each database grasp has an associated quality metric score, with a lower number indicating a better grasp. We note that the quality metric is specific to our parallel jaw gripper, with a good grasp requiring both finger pads to be aligned with the surface of the object and further rewarding postures where the palm of the gripper is close to the object as well. Our previous study also shows the experimentally determined relationship between the value of this quality metric and the

probability of success of the given grasp in real life execution. Based on those results, we consider any grasp with a quality score between 0 and 10 as a good grasp, while scores above 10 indicate a bad grasp. In practice, this is a conservative threshold, as grasps with quality scores in the 10 to 20 range also succeed in many cases.

The original database only contained good grasps (quality scores below 10). For the purpose of this study, we also needed examples of bad grasps. We thus used the same simulation engine to compute, for each object in our set, 90 additional grasps evenly distributed in the 10-100 range for quality scores, as well as 10 grasps where the gripper was in collision with the object.

For each grasp location contained in our corpus, we created a "perfect information" voxel grid, of the dimensions described earlier, describing the volume around the grasped object at the given grasp location. These voxel grids essentially represent our knowledge base; they are characterized by the fact that, unlike grids computed from scans, they contain perfect information: both the surface and the inside volume of the object are marked as occupied, with everything else marked as known empty. We created this type of voxel grid for both the good and bad grasps in our data set, resulting in voxel grids from a total of 13225 grasps (5323 good and 7900 bad) stored in our database.

C. Comparing graspable features

Once we have a large database of both graspable and ungraspable features, we can estimate how likely it is that an input grasp will succeed by comparing it to our feature database. First, we must define a distance metric between two grids. Our cell-distance metric b is a weighted sum of binary differences; for an input grid g_o and database grid g_i, with K being the set of cell indices into both grids:

$$b(g_o, g_i) = \sum_{k \in K} w_g p(g_o^k, g_i^k)$$

$$p(g_o^k, g_i^k) = \begin{cases} 1 \text{ if } g_o^k \text{ is occupied and } g_i^k \text{ is known-empty} \\ 1 \text{ if } g_o^k \text{ is known-empty and } g_i^k \text{ is occupied} \\ 0 \text{ otherwise} \end{cases} \tag{1}$$

where g_o^k is the point in grid g_o with index k, and w_g is a weight that is 0.9 for points within the gripper volume and 0.1 for points outside of it.

We compute the cell-distance between two grids at several resolutions, coarse to fine. When generating a coarser-resolution grid, each coarse-grid cell contains several fine-grid cells. If all of the fine-grid cells are known-empty, the coarse-grid cell is also set to known-empty; if any of the fine-grid cells are occupied, the coarse-grid cell is set to occupied, and otherwise the coarse-grid cell is set to unknown. For database grids, the voxel grids at all resolutions are precomputed and stored for fast matching. In our implementation, we used four different resolutions: 40, 25, 10, and 5 grid cells on a side. We note that the weight w_g as described above is only used at the highest two resolutions, whereas at the lowest two resolutions all cells weigh the same when computing the distance b.

The cell-distances from all grid resolutions are used to compute an overall weighted distance function between the two grids, $d(g_o, g_i)$:

$$d(g_o, g_i) = \sum_j w_j b_j(g_o, g_i) \tag{2}$$

where j is one of h hierarchy levels (in our implementation, $h = 4$, so $1 \leq j \leq 4$), $b_j(g_o, g_i)$ is the cell-distance between the voxel grids for grasp g_o and g_i sampled at resolution level j, and w_j is the weight assigned to distance values at resolution level j. In our implementation, w_1 = 0.681, w_2 = 0.266, w_3 = 0.043, and w_4 = 0.010; these values were chosen to compensate for the different numbers of points in the grid at each resolution ($0.681 = 40^2/(40^2 + 25^2 + 10^2 + 5^2)$), although the w_j in general are parameters that can be adjusted to improve performance.

Using the coarser-resolution cell-distances as well as the highest-resolution cell-distance provides robustness to small shifts in object position; a positional shift of half a grid cell width at any resolution can be enough to cause an exact match to suddenly have no matching grid cells. In our implementation, the finest resolution grid cells are 5 mm on a side; except in rare edge cases, a shift of 2.5 mm is typically not enough to drastically change the quality of a grasp.

We can also use the multi-resolution grids to prune database grids in a hierarchical fashion, matching from coarse to fine and discarding grids that are too far apart at coarser levels to bother matching further. This process greatly speeds up the matching process. In our current implementation, we prune to 4000 neighbors at the coarsest level, then 2000, 1000, and finally keep only the closest 100 grids in the final neighbor pool. At the same time, we also limit the maximum cell-distance at each level to avoid keeping grids that are too dissimilar; in our current implementation, the cell-distance limits are 9, 27, 250, and 1000, from coarse to fine. Currently the full matching process for approximately 20 input grids can be done in one second on a modern computer with 8 cores. The most computationally intensive step is voxel grid matching at the highest resolution, which we hope to speed up by using a fast approximate nearest neighbor algorithm.

When evaluating an input grid g_o, we would like to take into account all matching database grids in the final neighbor pool, but we would like more distant matches to have less weight than closer matches. Our weight function for grid g_i in the neighbor pool $N(g_o)$ is as follows:

$$w(g_i) = \left(1 - \frac{d(g_o, g_i) - d_{min}}{d_{max} - d_{min}}\right)^2 \tag{3}$$

where d_{min} is the smallest distance of any grid in $N(g_o)$ and d_{max} is the largest. This weight function weights more distant grids less than nearer ones; the exact weight function used is somewhat arbitrary, and trying different functions and observing their effect on performance is the subject of future work.

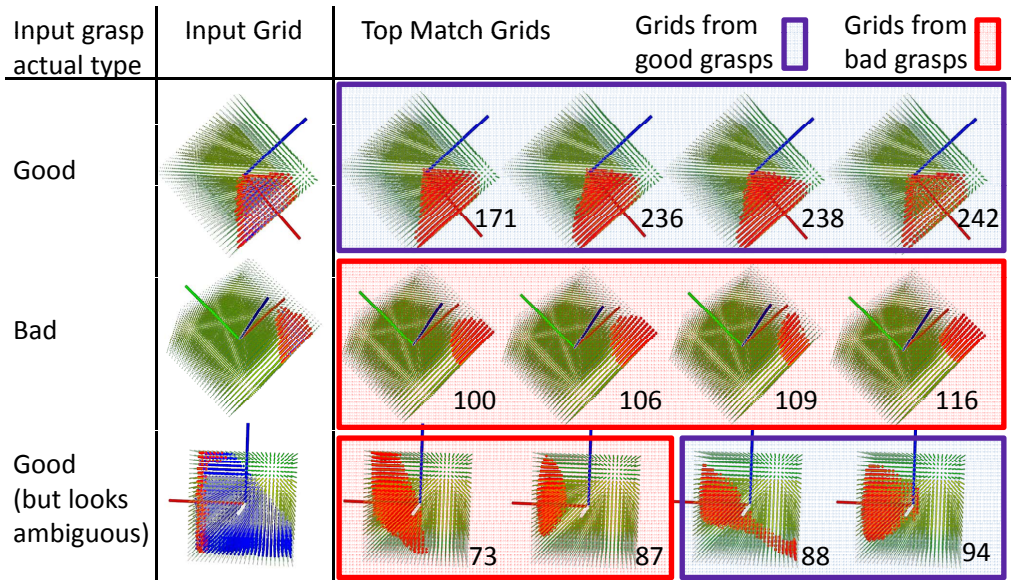

Fig. 2: Input grids and top database matches for examples of good, bad, and ambiguous grasps. Numbers next to each match are cell-distances at the resolution shown. Note that, for the first two rows, the input grid simulates ideal sensing conditions, with no unknown space. The third grid does contain a significant amount of occlusion, explaining the fact that it matches both good and bad grasps, as well as the lower (better) matching scores (as an unknown cell will match anything).

D. Graspable feature score

Our graspable feature score, $S(g_o)$, is an estimate of the likelihood of success of a particular grasp g_o based on its nearest neighbors:

$$S(g_o) = \frac{\sum_{g_i \in N(g_o)} q(g_i) w(g_i)}{\sum_{g_i \in N(g_o)} w(g_i)} \tag{4}$$

$$q(g_i) = \begin{cases} 1 \text{ if } g_i \text{ is a good grasp} \\ 0 \text{ if } g_i \text{ is a bad grasp} \end{cases} \tag{5}$$

If all of the n nearest matches in the database are good grasps, then it will have a value of 1, and if all of the nearest matches are bad grasps, it will have a value of 0. Grasp evaluation values in-between indicate ambiguity, either due to missing information (due to occlusions or sensor noise) or due to a grasp feature being similar to both good and bad grasp features.

Figure 2 shows three examples of input voxel grids and their closest four matches in the database. The first two rows show one good grasp and one bad grasp with "ideal" voxel grids (as described in sectionIV-A); the top four match grids are shown for each grasp, with the relevant distance scores for the highest resolution shown next to each match. The good grasp in this example matches all good grasps, and the bad grasp matches all bad grasps, as one would hope. The third row is a voxel grid from an actual stereo camera point cloud–the same grasp, object, and grid shown in Figure 1–and has significant unknown space intersecting the proposed grasp. Thus, its nearest matches vary greatly in geometry, particularly in the input grid's unknown region; in this case, the first two matches are from bad grasps and the next two are from good grasps. All four have similar scores, however, and thus if those four were the only matches considered, the resulting grasp evaluation score would be nearly 0.5.

IV. Simulation Results

A. Full object point clouds

The first test we performed was intended as a baseline measure of the voxel grid matching function and our graspable feature database, isolating it from the effects of missing data or sensor noise. To this end, we used the method described above to analyze each grasp (both good and bad) in our database against the rest of the database, simulating ideal sensing conditions.

For each grasp, we built an "ideal" grasp voxel grid by performing simulated object scans, with the following characteristic: object surface was marked as occupied, object interiors were marked as unknown, and everything else was marked as known empty. This procedure is intended to simulate a scenario were depth images of the target object are available from enough viewpoints to resolve any occlusions. The resulting voxel grid was then matched against the rest of our database using the method described in the previous section, and a graspable feature score was computed. This score was then compared against the known quality metric of the initial grasp. As matched voxel grids are free of both occlusions and sensor noise, any mislabeling after this procedure can only be due to the matching procedure itself.

The results, shown in Fig. 3, confirmed the strong correlation between the grasp quality metric and its graspable feature score computed using the method presented in this paper. Both good grasps (energy metric below 10.0, graspable feature score close to 1.0) and bad grasps (energy metric at 30.0 or above,

graspable feature score close to 0.0) were generally labeled correctly, with ambiguous results in the range of "mediocre" grasps (energy metric between 10.0 and 30.0). In particular, the resulting plot is almost completely clean in the critical upper right corner (bad grasps incorrectly labeled as definitely good), as well as the bottom left corner (good grasps incorrectly labeled as definitely bad). This test established a blueprint for what successful results should look like on real-life sensor data and grasps.

Note that, in Fig. 3, the red points with a grasp energy metric of -1 are grasps for which the fingertips are in collision with the object, which would ideally be labeled as bad, but which are often very similar to non-colliding grasps. In practice some such grasps can be filtered out based on obvious collisions between the hand geometry and known points, and some others can be fixed by reactive grasp adjustment as in [5]; matching performance may be improved in the future by adding more colliding grasps to our graspable feature database. The points with a database match score of exactly 0.5 are grids for which no matching grids were found in the database (due to the limits on cell-distance at each resolution level); the number of these is an indicator of insufficient database size. Currently there are already few such grasps, but in general exploring the effect of database size is the subject of future work.

B. Stereo camera point clouds

In order to test the graspable feature evaluation method using real-life sensor data, we used the Grasp Playpen dataset introduced in a previous report [2]. This dataset contains real depth images of the objects in our database, manually labeled with ground-truth object identity and pose. We can thus evaluate the grasps from the database, using voxel grids extracted from real life sensor images of their respective objects. The "ground truth" that we use to evaluate our results against is still the energy metric of the grasps, which was computed in simulation using *GraspIt!* with the ground-truth object geometry and pose. As such, we consider these to be simulation results, even though they were obtained using real sensor images.

For each of 134 real-sensor scans from the Grasp Playpen dataset, we generated 16 test grasps, 8 good and 8 bad. For each grasp, we extracted the corresponding voxel grid from the real life sensor image of its target object; all voxel grids were extracted from a single image, and thus contained significant occlusions. The voxel grid was then matched against our grid database, and its graspable feature score was computed. This score was then compared to the "ground truth" energy metric of the grasp.

The results are shown in Fig. 4. We notice that the computed graspable feature score is still a good indicator of grasp quality, with good grasps clustered in the top left corner of the scatter plot. As expected, the correlation is not as strong as when using occlusion-free, simulated data. In this plot, the grasps whose graspable feature score is precisely 0.5 are cases where the grasp was performed from "behind" the object relative to the camera location, grasping in a fully occluded region and thus creating completely unknown voxel grids. Such voxel grids are automatically labeled with a 0.5 score and do not undergo regular matching.

V. Conclusions and Future Work

We have presented a method that aims to determine graspable features, or object locations suitable for grasping with a robotic gripper. We encode such locations in voxel grids that encompass an approximately gripper-sized volume at each proposed gripper pose. These voxel grids can be constructed from data of a scene captured using a depth sensor or range scanner, preserving information about occupied, free, or occluded areas.

A voxel grid of a potentially graspable feature is evaluated by matching against a large database of fully known volumes built around both good and bad grasps, constructed using both scan data and grasp labels computed in simulation. A test voxel grid that matches primarily against grids of known good grasps is labeled as graspable. Conversely, matches amongst grids of known bad grasps indicate a non-graspable feature. Equal proportions of good and bad matches indicate an ambiguous candidate, presumably due to either significant occlusions in the scan, or a grasp that is marginal.

The preliminary results presented in this study show that the graspable feature score computed using this method is a good predictor of the quality of the grasp, as evaluated in simulation. This result also holds when using real-life, single-view-point sensor images of objects with significant occlusions. These images are also labeled with ground truth object identity and location, allowing the results of graspable feature evaluations to be compared against grasp quality metrics computed in simulation.

The obvious next step along the directions presented in this study is to test grasps evaluated with our method by executing them with a real manipulator. In addition, we will study the problem of not only evaluating grasps, but also generating a list of possible candidates starting from a sensor image of a scene. Finally, reducing the computational effort associated with evaluating one grasp can allow more candidates to be tested, and increase the potential applicability of this method in real-life manipulation tasks.

References

[1] M. Ciocarlie, K. Hsiao, E. G. Jones, S. Chitta, R. B. Rusu, and I. A. Sucan. Towards reliable grasping and manipulation in household environments. In *Intl. Symp. on Exp. Robotics*, 2010.

[2] M. Ciocarlie, C. Pantofaru, K. Hsiao, G. Bradski, P. Brook, and E. Dreyfuss. A side of data with my robot: Three datasets for mobile manipulation in human environments. *IEEE Rob. and Autom. Mag. Special Issue: Towards a WWW for Robots (in press)*, 2011.

[3] J. Glover, D. Rus, and N. Roy. Probabilistic models of object geometry for grasp planning. *RSS*, 2009.

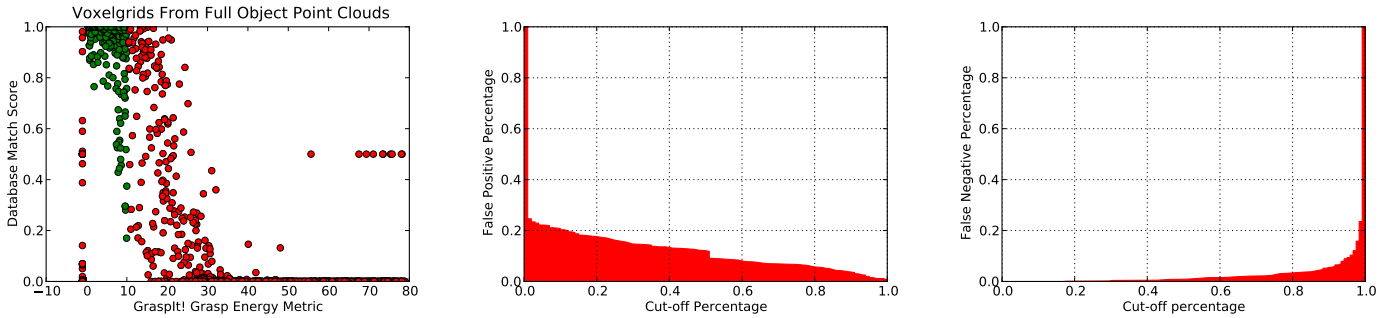

Fig. 3: Matching results on simulated, "ideal" sensing conditions. Left image plots the quality metric of the grasps in the database against their graspable feature score computed using the method described here. Grasps with an energy metric below 10.0 are considered good, and marked with green dots; bad grasps are marked with red dots. Middle and right plot show, for bad and good grasps respectively, the percentage of grasps (vertical axis) whose graspable feature score is above a given threshold (horizontal axis). For example, these plots show that if we were using a threshold of 0.9 or above for graspable feature scores, we would keep approximately 4% of the bad grasps (false positives) and discard approximately 6% of the good grasps (false negatives).

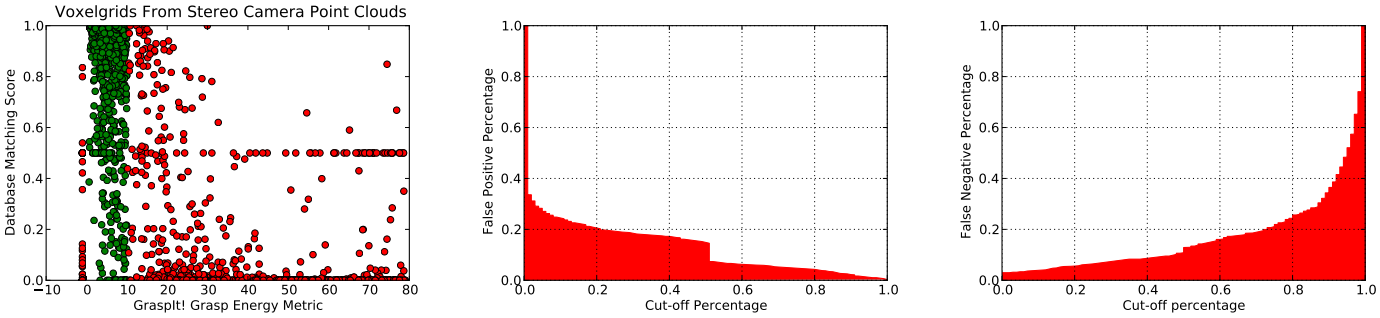

Fig. 4: Matching results using real sensor images of objects and simulated grasp quality values. Plots are interpreted in analogous fashion to Fig. 3.

[4] C. Goldfeder, M. Ciocarlie, J. Peretzman, H. Dang, and P. Allen. Data-driven grasping with partial sensor data. In *IROS*, 2009.

[5] K. Hsiao, S. Chitta, M. Ciocarlie, and E. G. Jones. Contact-reactive grasping of objects with partial shape information. In *IROS*, 2010.

[6] K. Hsiao, M. Ciocarlie, and P. Brook. Bayesian grasp planning. *ICRA Workshop on Mobile Manipulation: Integrating Perception and Manipulation*, 2011.

[7] A. Jain and C. Kemp. EL-E: an assistive mobile manipulator that autonomously fetches objects from flat surfaces. *Autonom. Rob.*, 2010.

[8] Y. Jiang, S. Moseson, and A. Saxena. Efficient grasping from rgbd images: Learning using a new rectangle representation. *ICRA*, 2011.

[9] I. Kamon, T. Flash, and S. Edelman. Learning to grasp using visual information. In *Intl. Conf. on Robotics and Automation*, 1996.

[10] E. Klingbeil, B. Carpenter, O. Russakovsky, and A. Ng. Grasping with application to an autonomous checkout robot. *ICRA*, 2011.

[11] A. Maldonado, U. Klank, and M. Beetz. Robotic grasping of unmodeled objects using time-of-flight range data and finger torque information. *IROS*, 2010.

[12] Andrew Miller and Peter K. Allen. GraspIt!: a versatile simulator for robotic grasping. *IEEE Rob. and Autom. Mag.*, 11(4), 2004.

[13] R. Pelossof, A. Miller, P. Allen, and T. Jebara. An SVM learning approach to robotic grasping. In *Intl. Conf. on Robotics and Automation*, 2004.

[14] A. Saxena, L. Wong, and A.Y. Ng. Learning grasp strategies with partial shape information. In *AAAI*, 2008.