A Kinect-Based Dataset for Determining Grasping Rectangles

Justin Kerekes jk2273
Katie Lee Meusling kam373

Abstract: With increasingly better and more widely available sensing technology capable of providing both a color image and associated point cloud, there is a great potential for advancement in the areas of object recognition and robotic manipulation. Here we present a large dataset consisting of 280 objects in 71 distinct categories with multiple views of each object, in addition to background images to allow for background subtraction. In addition, we have labeled both positive and negative grasping rectangles indicating good and bad grasping locations. The images, point clouds and labels are all publicly available to the robotics research community. The rest of the project focuses on finding good grasping rectangles in images. Prior work has shown that grasping rectangles can be an effective representation for finding grasping locations in an image [3,4]. We then use our data set to train and evaluate a supervised learning algorithm using SVM-light[5] using both pixel and point cloud features. Due to the wide variety of object types, this particular dataset proves challenging.

Keywords: Dataset, grasping, supervised learning

I. Introduction

There is a large-scale multi-view RGB-D dataset available to the research community provided by Lai et al [1]. Our dataset contains more objects than this, but with fewer orientations per object. The dataset by Lai et. al. is carefully hierarchically structured to allow for object classification tasks. Our dataset is instead focused on robotic grasping and the objects were chosen to be of appropriate size and shape as to be graspable. In addition, all orientations of the objects contain a viable grasping rectangle. The dataset includes both images and point cloud data, obtained from a Kinect Sensor, and is described in more detail below.

Grasping is one of the most important challenges in personal robotics. For a robot to be useful in a home setting, it needs to be able to identify objects and be able to pick them up or move them. Identifying a location on an object to grasp is a difficult problem even given a perfect 3d model of an object. The task is further complicated when relying on a single image and incomplete point cloud data. Much prior work has been done to identify grasping points in an image [1]. Jiang et. al. shows that using a rectangle representation for grasping instead of simply a point can perform better for robots with 2 parallel gripping plates as it better models the physical space that gripper will occupy [4]. Jiang uses a set of 17 filters to extract features from gripping rectangles and only uses the depth information for identifying the gripping location in physical 3-d space.

Our approach is similar to the methods employed by Jiang et al [4] in that we are identifying 2d grasping rectangles in an image and translating to a 3d robot position. However we will use a combination of pixel based features as well as 3D features extracted from the associated point

clouds, as Lim and Biswal do in[3]. We train our model using hand-labeled good grasping rectangles and bad rectangles on our training set of images. We extract the features from each of these oriented rectangles and use these features to train a linear SVM classifier described in [4]. To predict a grasping rectangle, we exhaustively search the image plane for rectangles of different sizes and orientations and return the rectangle with the highest rank using the SVM model we have trained.

II. Dataset

Our new dataset contains images and point clouds for 280 distinct objects (1,035 images/point clouds), which are of appropriate size and shape that a robotic arm equipped with a gripper capable of opening 4 inches would be able to grasp. The objects chosen are common household objects which could potentially be found in an environment where a personal robot would act. All images were collected with a Kinect sensor.

Figure 1: A sample of objects from the dataset

For cups, classes, and bowls, you can expect a good grasping point to be on the outer rim of these objects. Similarly, for pens, markers, and pencils, you can expect a good grasping point to be in the middle of the object, grasping around it. However, our dataset is much more comprehensive, containing objects that are not as obvious. For instance, you wouldn't grab the wire part of a whisk, or the cord of a pair of headphones. This dataset is variable enough to be representative of what a robot would typically see day-to-day household use, as well challenging enough to provide researchers with a tough benchmark to try to perform well against.

The background of the images is a white table with a white backdrop, and a sliver of the floor can be seen on the sides. Background images were also taken to allow for simple background subtraction. In a realistic setting, a robot will not be able to take a picture of the background without the object, but we are not as concerned with image segmentation as we are with the learning algorithm and data collection. We chose a solid, plain background behind each item to reduce the likelihood of confusing the object with the background. The background images are available with our dataset.

Each object is positioned in multiple poses or orientations with respect to the sensor, except in the case of perfectly symmetrical objects such as circular bowls, where only one image is taken per object. Poses were chosen so that a robotic arm would be able to grasp the object coming from a trajectory which is normal to the image plane.

There are 71 different categories of items, and each object belongs to one of these categories. A sample of the different categories of the objects is *aluminum can*, *bowl*, *scissors*, and *miscellanous* (to cover the 12 items which do not belong to any of the other categories). Each category of items contains 2-12 different objects which fall into that category.

The rest of the project focuses on identifying good grasping rectangles in an image. To achieve this, we need labeled data to train a supervised learning algorithm (we use support vector machine), and labeled rectangles to evaluate on the test set. For this purpose, we have hand labeled both good and bad grasping rectangles for each of the images in our dataset. The positive rectangles identify all of the good grasping rectangles in the image assuming a gripper orientation normal to the image plane. The positive rectangles overlap with each other to ensure that in the case of many equally good grasping rectangles, the automated testing will correctly determine whether a proposed rectangle is good or not. In addition, this provides many more training examples for the supervised learning system.

Different objects have more or fewer good grasping rectangles, so the number of positive rectangles varies from image to image. Some objects have as few as 2, or as many as 22, for example, on a frisbee. In addition to positive rectangles, we hand labeled negative rectangles, which are bad grasping rectangles. Roughly 3 negative rectangles on average are labeled for each image. There are a total of 3681 positive rectangles and 3367 negative rectangles labeled throughout our entire dataset.

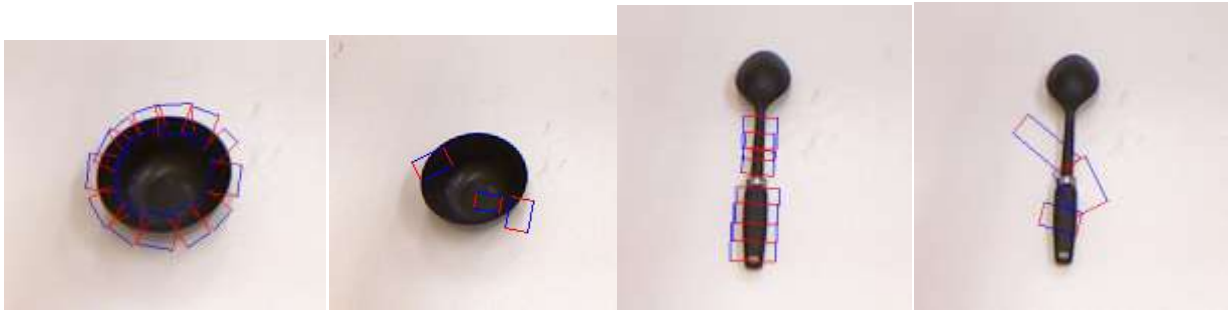

Figure 2: Good and bad grasping rectangles for a bowl and spoon. Blue lines indicate the orientation of the gripper plates

Including both positive and negative rectangles allows us to train a classifier to determine if a rectangle on a new object which the system has never seen before is a good rectangle or a bad one.

III. Approach

After collecting and labeling the data and dividing into training and test sets, we extract features from all the positive and negative rectangles from the image and point cloud files. These extracted features are used to train a SVM classifier. SVM's are good at classifying points in high dimensions, which is a significant concern here due to the large number of features used.

A rough evaluation of the hardness of our dataset can be seen by comparing how Marcus Lim's classifier performed on it. On Lim's dataset, he gets approximately 66% accuracy for his system's classifier on his dataset [3]. Using the same features that he uses - of which there are approximately 1,900 - we also train a classifier using Lim's code, but we use training data from our new dataset. We get approximately 22% accuracy on the new dataset, which is significantly lower than on Lim's dataset.

A. Features

The baseline system uses the features described by Lim and Biswal[3]. There are 17 RGB image filters, used by both Jiang et. al.[4] and Lim and Biswal[3]. There are 6 oriented edge filters to extract edge features and 9 Laws' masks. In addition, we use the z-position, surface normal in the z-direction and the approximated local curvature from the point cloud files and Fast Point Feature Histogram, as in Lim and Biswal [3]. The system we use is the same code that they wrote.

To compute the features for each rectangle, we use normalized fuzzy histograms with 15 bins. Additionally, the rectangles are divided into 3 strips because the middle strip, which contains the portion of the object to be grasped is likely to have different features from the external strips. Thus for each of the filters there are 3x15 features. In total, there are 1,902 features that are used in this system.

B. Background Subtraction

To prevent background items, or the edge of the table, from being identified as part of a good grasping rectangle, we employ background subtraction. We use the background images taken without an object and the background subtraction algorithms provided in openCV. Removing the background also significantly reduces our search space when looking for good grasping rectangles in the image plane.

C. Finding a Rectangle

We begin the search process by discretizing the search space significantly. We do not have the processing power to test every possible grasping rectangle, so we are restricting the properties of the rectangles that we search. For every pixel, we test 972 different rectangles. Their height and width of the rectangles range from 20 pixels to 140 pixels, and the rotation ranges from 0 to 165 degrees. We restrict rectangles to be on the plane of the test image for simplicity (recall we chose orientations for the objects in our dataset such that they would be graspable with a

grasping rectangle on the image plane). For each of the rectangles that we test, we first extract the same set of features which was used during the training. Then we use the SVM-rank model trained previously to calculate the score for the particular rectangle.

All of the examined rectangles are scored by the classifier, and we chose the top scoring rectangle as our best guess for a good grasping rectangle. We describe below the metric we used to quantify the effectiveness of this system performing on our new dataset.

III. Experiments

A. Data

We train our model using the dataset described above. We divide the data into a training set, and two test sets. The first test set consists of novel (unseen during training) objects and is 10% of the full dataset. The second test set contains objects seen during testing but in different poses or orientations as seen before and is 15% of the full dataset. The novel objects test set is smaller to allow more variety in the training set. A representative set of different categories of objects is included in different test sets. The wide variety of object types makes our algorithm more robust and extensible to novel objects. We trained our model on the remaining 75% of the data, the training set, and evaluate on the two test sets separately.

B. Evaluation Metric

To evaluate the accuracy of the system, we use the metric proposed by Jiang, et al [4] and used by Lim and Biswal[3]. The top predicted rectangle is compared to the ground truth labeled rectangles. If the difference in angle of orientation is less than 30 and the ratio of the area of intersection to the area of the rectangle is more than 0.5, we consider it a correct prediction. With multiple ground truth rectangles, we compare the predicted rectangle with all ground truth rectangles and use the best score.
Score= 1{Orientation difference<30}Area of intersectionArea of prediction.

C. Results

The table below summarizes the results of our offline tests of both testing sets, the Novel Objects (unseen during training) and the Seen Objects, in different poses than in the training set.

Features Used	Novel Objects	Seen Objects, new pose
RGB features only	**21.36%** (22/103)	**23.37%** (36/154)
RGB + point cloud features	**22.33%** (23/103)	**23.37%** (36/154)

With the Novel Object test set, the algorithm performed best on the bowls and frisbees, correctly finding a good grasping rectangle for 2 out of 2 frisbees and 2 out of 2 bowls. It also correctly labeled 3 out of 4 markers and 2 out of 4 remote controllers. On the Seen Objects test set, the algorithm performed well on the razors, getting 2 out of 4 correct and the toothbrush again getting 2 out of 4. As shown below, many of the proposed rectangles appear on the edges of

objects where one side of the rectangle is good, but the other side would hit the object. The majority of the highly ranked rectangles were very small, and not capable of grasping some of the objects. The reason behind this occurrence is not clear. See the images below for a sample of a good test result and a bad test result. In general, the algorithm was best at identifying very thin grasping rectangles, such as the handle of a toothbrush or the rim of a bowl, and was less successful at larger block-like objects.

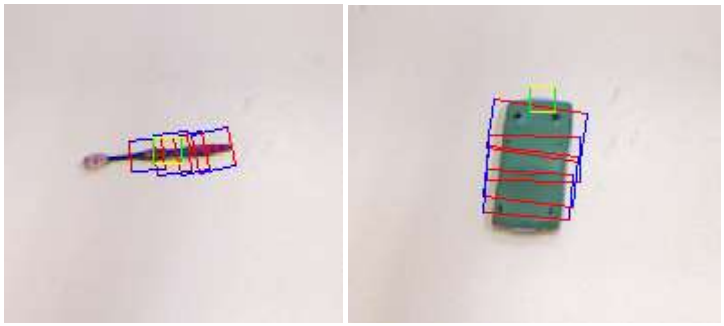

Figure 3: Good and bad grasping rectangles chosen by the system (yellow edge indicates the orientation of the gripper plates) along with our hand-labeled rectangles.

The wide range of objects that make up our dataset have a variety of shapes and sizes. Some of the good rectangles on one object have feature vectors which are similar to negative rectangles on another object. For example, on a laptop charger, a dangling end of a cord is not a good location to grasp the object, however a toothbrush handle is small enough to look quite similar to a piece of cord. In this regard, our dataset is somewhat self-contradictory. However, it does represent a range of common real-world items that a successful robotic manipulator would need to be able to grasp. Perhaps the limited success on this larger dataset suggests that a more complex approach is required to successfully grasp such a wide range of objects.

Sources

[1] Kevin Lai, Liefeng Bo, Xiaofeng Ren, and Dieter Fox. "A Large-Scale Hierarchical Multi-View RGB-D Object Dataset." *ICRA* 2011, May 2011.

[2] A. Saxena, J. Driemeyer, J. Kearns, A. Ng. "Robotic Grasping of Novel Objects" ***NIPS*** 19, 2006

[3] Lim, Marcus and Biswal, Biswajit. "Determining Grasping Regions using Vision."

[4] Y. Jiang, S. Moseson, and A. Saxena, "Learning to Grasp: What Should We Actually Learn?" 2010

[5] T. Joachims, "Optimizing search engines using clickthrough data," in SIGKDD, 2002