Image Processing for Robotic Manipulation

Himani Arora, Yuanfeng Mao, Aiyu Tang, Yuanqing Hong

***Abstract*—Robotic grasping demands the help of image processing to analyze and measure unknown objects. In this paper, we explore the use of object detection and classification to represent shape of objects for robotics. The project starts by taking photos of several objects in a clean background, then use object detection algorithm to extract the objects. Based on the extracted objects and the dataset gathered by ourselves, we can measure the properties for robotics grasping. Our experiments involve extracting boundary and objects from the picture, analyzing the properties of shapes, preparing training data sets and classifying objects. The results show that our methods are capable of measuring each object from the photos and providing related parameters for robotic grasping.**

Keywords—robotics manipulation; object detection; object shape properties; image classification;

I. Introduction

A. Motivation

The autonomous grasping of various objects presents an enormous potential for industry. Widely used applications include putting different accessories into a labeled box or tray and robotic kitting, which release humans from repetitive and dangerous tasks [1]. In addition, the implementation of robotic grasping can help entrepreneurs lower the laboring cost as well as improving picking and packaging efficiency. However, the cost of a robotic arm remains a puzzle, impeding the existing mechanisms from being widely used. Thus, we are looking for a novel way of reducing the cost of robotic grasping while retaining the effectiveness and accuracy of the work.

B. Previous Work

Computer vision is commonly used in robotic grasping. There is a significant body of modern robot designs, ranging from simple grasping tools to complex hand designs for a previously known object, to grasping of an unknown object such as ones based on friction cones [2], form- and force-closure [3], pre-stored primitives [4], or grasping based on location-identification algorithm [5]. All of above methods successfully solved the problem of grasping with single object in sight. However, in practical scenarios it is often difficult to grasp the target object when there are different types of objects in the surroundings.

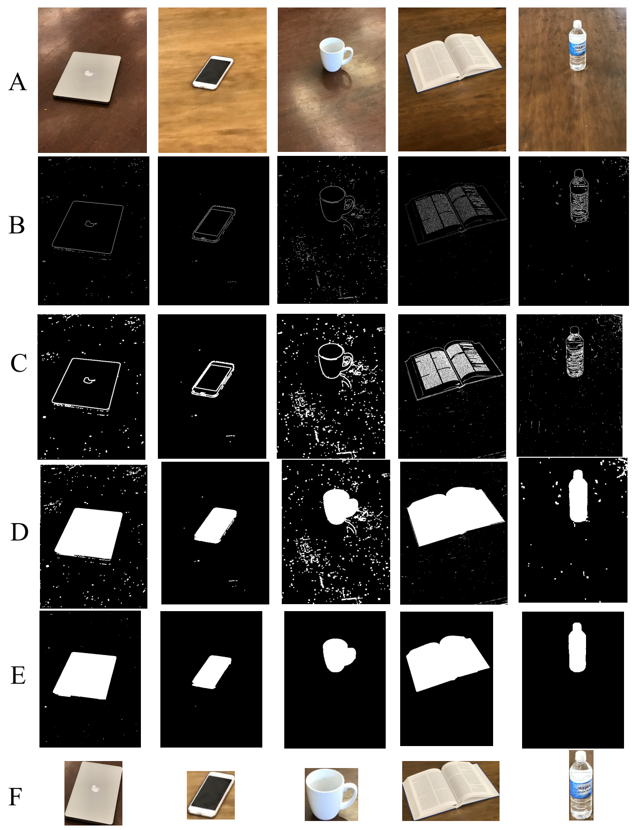
C. Goals

In contrast to these approaches, we propose a way of combining image processing and object classification to assist the robotic arm in grasping object. The task is defined as grasping of a target object in a multiple objects context. To begin with, a camera which is fixed to a specific place above the workspace takes a photo of the objects on the workspace. A plain background is used to simplify the problem. Then the controller conducts image processing of the photo with object detection algorithm and extracts each object out of the mixed background. With these single object pictures, the computer is able to classify each object into correct category and identify the target object. Then the properties of the target are calculated to assist the robotic arm in grasping.

II. Technical Approach

A. Object Detection

Object detection is the segmentation of an image into different components in order to extract the part we care about [6]. In our case, the part we cared about consisted mainly of objects that were part of our dataset (A). We describe below our approach:



Figure(1) Process for object detection

1. Since there was sufficient contrast difference between the background and the objects of interest, we first calculated the gradient of the image using Sobel edge

detection method. This gave us a binary image where the white pixels represented the edges present in the image (B).

2. However, the edges still had some gaps in between and were not smooth. To create smooth edges, we then dilated the image using a vertical as well as a horizontal structural element (C).
3. Once a uniform thick outline was obtained, we then filled all the holes in the binary image (D).
4. However, this also magnified the noise present in the image which we then cleaned through morphological opening on the binary image with a disk shaped structuring element (E).
5. Now each object in the binary image became a connected component. We gave each connected component a unique label and then filtered out the object corresponding to that labelled component (F). Since the objects we care about had a significant size as compared to the noise components, we also applied a threshold on the area of the connected components to avoid considering any noise blob.

B. Training Dataset Preparation

In this paper, we focus on the everyday objects such as laptops, phones, cups, books and bottles in clean backgrounds. Since most of the datasets from the Internet have an enormous size, we want to gather datasets by ourselves. And we found that dozens of pictures are enough to get satisfied results. Therefore, in our project, pictures were taken by ourselves and the number is about 50 for each kinds. In the process of preparing for datasets, careful consideration was given. For the background of pictures, we took pictures at plain background. Given the big size of original pictures will slow down the processing of image classification, we used the method of compressing pictures to reduce the size of datasets.

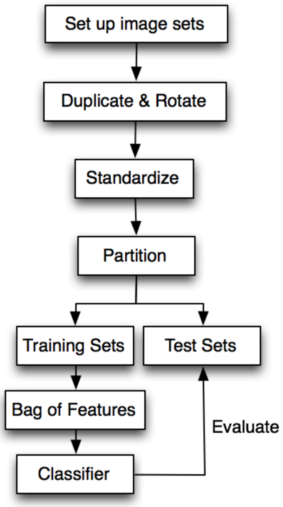
Some samples from our datasets are shown in figure(2), which include laptops, phones, cups, books and bottles:

Figure(2) Several samples from datasets

C. Image Classification

In order to identify the object shown in the input image, we will be using the method of comparing the image with a bag of visual words [7]. We will be using many of the built-in functions for image classification in order to achieve this purpose. The general idea of this method is as follows: features will be extracted from a subset of our dataset and will be used to create an image classifier, which serves to predict the object shown in the image.

Our code will first take in all of the images in our data set and determine the minimum number of pictures out of all the categories in the dataset. Then, we reduce the number of pictures in each category to equal the minimum number, that way we will have a balanced training set. After resizing, we will divide the image set- 70% of the images will be used as a training set and the rest will be the validation set, which will be used to determine the accuracy of the image classifier. In order to create the bag of visual words, we use the bag() function on the training set. Then, we use the trainImageCategory() function and the newly created bag to train the images in the training set; the resulting output will be an image classifier. Using the evaluate() function on the image classifier and validation set, we are able to determine the average accuracy of the classifier. Using the predict() function and the image classifier, we can classify the image shown[8]. The process of classification algorithm was shown in figure(3).



Figure(3) Image classification algorithm

By using our datasets, we test the algorithm above. The output of the code was shown in Figure(4). From these data, the average accuracy reach 0.87, which demonstrates the reliability of our algorithm.

	PREDICTED				
KNOWN	Book	Cup	Laptop	Phone	Water
Book	0.76	0.02	0.21	0.00	0.00
Cup	0.00	0.93	0.05	0.02	0.00
Laptop	0.24	0.00	0.76	0.00	0.00
Phone	0.07	0.00	0.05	0.88	0.00
Water	0.00	0.00	0.00	0.00	1.00

* Average Accuracy is 0.87.

Figure(4) Predicted accuracy

D. *Parameters Calculation*

The related parameters for robotics grasping include label of numbers, areas, centroid, orientation, length, width which were listed as followed [9]:

- Label: label number of each object from 1 to N
- Area: the area of each object
- Centroid: the coordinate of the center of mass.
- Orientation: the angle between the x-axis and the major of the ellipse that has the same second-moments as the region (x-axis: horizontal line towards right)
- Length: the length of the object measured along the second-moment axis
- Width: the length of the object measured perpendicular to the second-moment axis

These parameters were calculated by following formulas:

Area: $A = \sum_{i=1}^{n} \sum_{j=1}^{m} b_{ij}$

Centroid:

$$\bar{x} = \frac{1}{A} \sum_{i=1}^{n} \sum_{j=1}^{m} i b_{ij} \qquad \bar{y} = \frac{1}{A} \sum_{i=1}^{n} \sum_{j=1}^{m} j b_{ij}$$

Orientation: Suppose

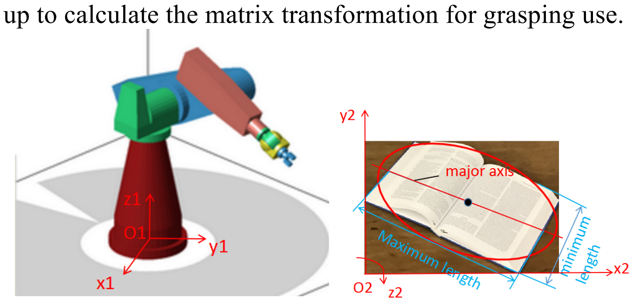
$$a = \sum_{i=1}^{n} \sum_{j=1}^{m} (i - \bar{x})^2 b_{ij} \quad c = \sum_{i=1}^{n} \sum_{j=1}^{m} (j - \bar{y})^2 b_{ij}$$

$$b = \sum_{i=1}^{n} \sum_{j=1}^{m} (i - \bar{x})(j - \bar{y}) b_{ij}$$

Orientation: $\theta = \frac{atan2(b, a-c)}{2}$

An example of the parameters of the target is shown in Figure(5) . Origin is denoted as a black dot. The red straight line denotes the major axis of the object. In implementation, the major axis is capable of guiding the grasper to grasp the object in a proper angle. As long as the center of mass and the orientation of the object are found, a basic grasping algorithm involving coordinate matrix transformation would give a suitable position and orientation of the end-effector of the robotic arm to grasp. In more advanced algorithms, the area and the shape of the object are required to minimized the error [10].

In the robotic grasping part, PUMA560, a type of robotic arm(shown in Figure(5)) was used to grasp the target. The frames for the robot(O1-x1y1z1) and the target(O2-x2y2z2) are set up to calculate the matrix transformation for grasping use.



Figure(5) An industrial robotic and target's parameters

III. Experiments

We first collected the training and testing dataset. These were then uploaded into the computer for image processing, which were mainly four stages: image preparation, object detection, image classification and parameter calculation. The details of these main steps are as follows.

Also, these experiments can be performed directly by running the code (main.m) provided with paper.

A. *Image Preparation*

In the experiments, we chose five types of objects for robotic grasping: water bottle, cup, laptop, book and phone. Fifty photos for each object were taken as training dataset from different orientations. Then for the testing dataset, these objects were placed randomly on the plain workspace. A photo of the overview of the workspace was taken. We collected 20 such sets of pictures each of them containing several objects. Through our test, there are 18 sets of pictures have been successfully detected and calculated. In this paper, we only show one set and the remaining sets could be saw in the code.

B. *Object Detection*

We used the object detection algorithm described above to detect multiple objects from the scene. The image shown in Figure(6) consisted of four different random objects. Our algorithm detected these objects and tagged each with a different color.

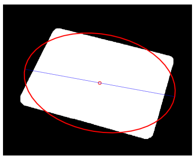
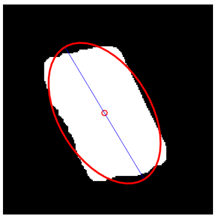
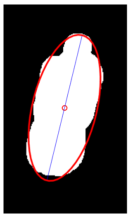
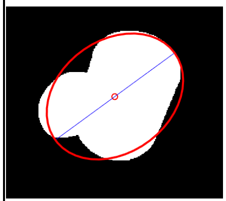
Figure(6) The photo of original objects and detected objects

C. *Target Identification*

After performing object detection, we extracted the detected objects from the whole image and used the separated objects for classification. Using the Bag of Features method described above, we first trained our classifier on the training dataset. After achieving an accuracy of 87% on the validation dataset, we saved our model. We then tested it on the testing dataset, the name of each objects were tagged. The results for the above example image has been shown below. From the result in Figure(7), we can see all objects have been classified correctly.

laptop phone bottle cup

Figure(7) Classification results

Object Name	Laptop	Phone	Water Bottle	Cup
Center of Mass($pixels$)	(426.39, 298.04)	(101.38, 111.69)	(100.06, 189.86)	(174.37, 153.38)
Area ($pixels^2$)	82458.00	7380.00	12538.00	17148.00
Orientation ($degree°$)	-10.29	-58.93	76.32	36.10
Maximum Length ($pixels$)	416.06	127.04	193.87	173.54
Minimum Length ($pixels$)	263.81	76.35	83.76	132.01
Diagrams (red point: center of mass blue line: orientation)				

Figure(8) Parameters of detected four objects

D. Calculation of Parameters

Having attained the photo of the target, the grasping parameters could be calculated using the function 'Object properties'. We set each of the classified object as the target object one by one.

Based on the photo of the target, a set of result is calculated and shown in the Figure(8). Combined with the length and width measured along this direction, the grasping location and direction could be determined. For visualization, the target object along with its ellipse approximation and the orientation (given by the blue axis) is also shown for a robotic arm to grasp.

IV. Discussion

Our algorithm can successfully distinguish multiple objects from a single photo, and classify them into specific categories. From the classified single-object pictures, 2-Dimensional kinematic parameters are calculated for each desired object. Having been designed and programmed in a general-purpose manner it can easily be extended for future use cases. With additional work this could be productionized to industrial grasping implementations and reduce the cost of a robotic arm.

We attempted to increase the accuracy of the image classifier by increasing the size of our dataset. More specifically, we created a code that duplicates the existing images and rotates each duplicated image, but this method did not significantly improve the accuracy of the classifier.

For the purpose of this project, we restricted our dataset to a small number of objects so that we can train it easily. We also tested by taking pictures on different backgrounds and found that on highly textured backgrounds as well as reflective ones, we are not able to correctly extract the objects. This presents a scope for future work where we can look into incorporating a larger variety of objects in our dataset and by designing a more robust object detection technique to handle different backgrounds.

References

[1] http://blog.robotiq.com/bid/30680/Robotic-Kitting

[2] Mason, Matthew T., and J. Kenneth Salisbury Jr. "Robot hands and the mechanics of manipulation." (1985).

[3] Bicchi, Antonio, and Vijay Kumar. "Robotic grasping and contact: A review." ICRA. 2000.

[4] Miller, Andrew T., et al. "Automatic grasp planning using shape primitives." Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on. Vol. 2. IEEE, 2003.

[5] Saxena, Ashutosh, Justin Driemeyer, and Andrew Y. Ng. "Robotic grasping of novel objects using vision." The International Journal of Robotics Research 27.2 (2008): 157-173.

[6] https://www.mathworks.com/help/images/examples/detecting-a-cell-using-image-segmentation.html

[7] https://www.mathworks.com/help/vision/examples/image-category-classification-using-bag-of-features.html

[8] https://www.mathworks.com/help/vision/ug/image-classification-with-bag-of-visual-words.html?refresh=true

[9] Mahler J, Patil S, Kehoe B, et al. Gp-gpis-opt: Grasp planning with shape uncertainty using gaussian process implicit surfaces and sequential convex programming[C]//2015 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2015: 4919-4926.

[10] Yu Z, Gu J. Optimized planar grasping synthesis algorithm for multi-fingered robotic hand[C]//2009 IEEE International Conference on Automation and Logistics. IEEE, 2009: 1052-1057.