Robotic Vision Project - Object Recognition

Yu-Hsien Chang[1] and Yu-Ting Lai[1]

***Abstract*— In this project, we will implement Multi-view Self-supervised Deep Learning for 6D Pose Estimation Algorithm, developed by MIT-Princeton for Amazon Picking Challenge 2016, to recognize objects in images. The Fully Convolutiion Networks used in this algorithm is trained by VGG16 dataset and built with Marvin toolkit. The algoritm can perform object segmentation and pose estimation.**

I. INTRODUCTION

[1] [2] [3] [4] [5]

Robotic Grasping is a cruical task in the real world, it is composed of object recognition and pose estimation in order to grasp objects successfully.Object Recognition is and aplication to recognize objects in the view and it has two phases, one is ofline learning phase and the other is online recognizing phase. It can improve the efficiency in warehouse and reduce labor force. This technique is not only interesting but also useful in many scenarios. However, its performance will encounter difficulties due to different view points, various light conditions and transparent objects. We implement Multi-view Self-supervised Deep Learning for 6D Pose Estimation in the project and solve the difficulties mentioned above. Therefore, we first captured images using Intel RealSense SR300, then do the image segmentation to find the location of known objects using Fully Convolution Network(FCN). After getting the confidence score from the FCN, the pose estimation algorithm predict the pose of the objects. Bounding boxes are drawn on the RGB images for visualizing the result. The organization of the report is as follows: first, in Section II, we give an overview of Multi-view Self-supervised Deep Learning for 6D Pose Estimation Algorithm. In Section III, we will give a detail explanation of the hardware and the algorithm we use. Our results will be shown in Section IV. Finally, in Section V, we make a conclusion of our work.

II. RELATED WORK

A. Amazon Picking Challenge

It is the competition organized by Amazon Robotics. The goal of the challenge is to strengthen the ties between the industrial and academic robotic communities and promote shared and open solution to some of the big problems in unstructured automation.Each team builds up the hardware on its own and developes the object recognition algorithm and pose estimation algorithm to complete the tasks in the challenge.

*This work was not supported by any organization

[1]Yu-Hsien Chang and Yu-Ting Lai are first grade master students in Institute of Electrical and Control Engineering, National Chiao Tung University, HsinChu, Taiwan

B. Team Delft

Team Delft was one of 16 finalists for the Amazon Picking Challenge, and it is the championship of APC 2016. Team Delft used two industrial stereo camera's with RGB overlay camera's. One for detection of objects in the tote and the other on the robot gripper to scan the bins. For object recognition, Team Delft used deep learning neural network based on Faster RCNN then performed localization using the implemetation of super4PCS to do global optimation of the pose estimation. Finally, ICP was used to refined the estimation.

III. TECHNICAL PARTS

The system architecture is shown in figure 1. We capture RGB images and depth images using Realsense SR300 and Realsense-standalone function provided by MIT-Princeton. Realsense-standalone outputs four information, which are RGB image, raw depth image, image of depth aligned to color and the Cam-Info.txt respectively. RGB images, depth-aligned-to-color images and Cam-Info.txt are used for object recognition and pose estimation. Cam-Info.txt contains camera parameters, such as color intrinsics, depth intrinsics and depth-to-color extrinsics. First, we get HHA features from depth images, in which we address its horizontal disparity, height above ground, and angle between surface normal and gravity to form HHA features. Next, we do image segmentation using pre-trained fully convolutional neural network. In this step, object will be recognized and segmented from the image. After segmentation, we do pose estimation and show the bounding boxes on the RGB images.

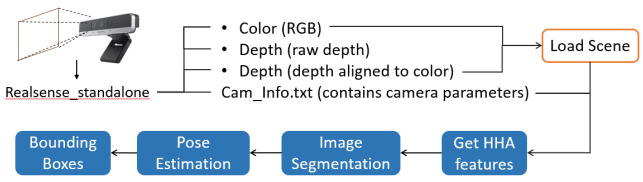

Fig. 1. System Architecture

A. Intel Realsense SR300

Intel Realsense SR300, shown in figure 2, is a RGB-D camera. In this project, the RGB-D camera is used to capture RGB image and depth image of the target object. Both of the two kinds of image data are used in object segmentation stage and pose estimation stage of the algorithm. The resolution of RGB image is 640*480 pixels and the resolution of depth image is 640*480 pixels. The range of depth capturing is 0.2m to 1.5m.

Fig. 2. Intel Realsense SR300

B. Object Segmentation

After we captured depth images from realsense, we extract the HHA features from them. HHA features are self-defined features, which address horizontal disparity, height above ground, and angle between surface normal and gravity, for each point. We will create a HHA map like this. The higher intensity means the point has a greater value of the sum of these three features. Then, we see this HHA feature map as the input to the pre-trained neural network. The neural network was trained using Marvin toolkit provided by Princeton. It will output the confidence score of the object with a 16-bit grayscale image. The lighter part in the image indicates that the object is more possible to appear here. Therefore, we complete object segmentation according to every items. For example, if the image contains three objects, it will create three confidence maps. These maps will be used in pose estimation.

[add figures]

C. Pose Estimation

(not yet)

The equations are an exception to the prescribed specifications of this template. You will need to determine whether or not your equation should be typed using either the Times New Roman or the Symbol font (please no other font). To create multileveled equations, it may be necessary to treat the equation as a graphic and insert it into the text after your paper is styled. Number equations consecutively. Equation numbers, within parentheses, are to position flush right, as in (1), using a right tab stop. To make your equations more compact, you may use the solidus (/), the exp function, or appropriate exponents. Italicize Roman symbols for quantities and variables, but not Greek symbols. Use a long dash rather than a hyphen for a minus sign. Punctuate equations with commas or periods when they are part of a sentence, as in

$$\alpha + \beta = \chi \tag{1}$$

Note that the equation is centered using a center tab stop. Be sure that the symbols in your equation have been defined before or immediately following the equation. Use (1), not Eq. (1) or equation (1), except at the beginning of a sentence: Equation (1) is . . .

IV. EXPERIMENT

In this section, we will explain the experiments we did in this project. First, we run the images captured by MIT-Princeton in APC. Second, we will generate the images by ourselves.

A. Benchmark images from APC

We run the images provided by MIT-Princeton to test the validity of the algorithms. We picked one of the benchmark image set from the shelf in APC scenes. We obtain the HHA features and calculate the confidence map using object segmentation algorithm described above. Next, we predict the poses of the segmented objects by drawing bounding boxes. The results are shown in the figure.

[add figure]

B. Self-generated Images

(not yet)

Positioning Figures and Tables: Place figures and tables at the top and bottom of columns. Avoid placing them in the middle of columns. Large figures and tables may span across both columns. Figure captions should be below the figures; table heads should appear above the tables. Insert figures and tables after they are cited in the text. Use the abbreviation Fig. 1, even at the beginning of a sentence.

TABLE I
AN EXAMPLE OF A TABLE

One	Two
Three	Four

We suggest that you use a text box to insert a graphic (which is ideally a 300 dpi TIFF or EPS file, with all fonts embedded) because, in an document, this method is somewhat more stable than directly inserting a picture.

Fig. 3. Inductance of oscillation winding on amorphous magnetic core versus DC bias magnetic field

Figure Labels: Use 8 point Times New Roman for Figure labels. Use words rather than symbols or abbreviations when writing Figure axis labels to avoid confusing the reader. As an example, write the quantity Magnetization, or Magnetization, M, not just M. If including units in the label, present them within parentheses. Do not label axes only with units. In the example, write Magnetization (A/m) or Magnetization A[m(1)], not just A/m. Do not label axes with a ratio of quantities and units. For example, write Temperature (K), not Temperature/K.

V. CONCLUSIONS

In this project, we explore the object segmentation and pose estimation algorithms provided MIT-Princeton in Amazon Picking Challenge. We started the whole experiments from setting up Realsense camera to capture depth images, then applying the fully convolutional neural network provided by Marvin toolkit, to pose estimation for the objects. We also captured the images on our own to test the algorithms. However, we need to calculate the exact camera extrinsic parameters between each poses, and calibrate our camera poses with the camera poses in MIT-Princeton. The bounding boxes results are not as good as expected. In the future, we can estimate the exact poses of the objects with a correct calibration parameters.

REFERENCES

[1] T. Hodan, X.Zabulis, M.Lourakis, S. Obdrzalek, and J. Matas, "Detection and fine 3d pose estimation of texture-less objects in rgb-d images," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sept 2015, pp. 4421–4428.

[2] D. Holz, A. Topalidou-Kyniazopoulou, J. Stuckler, and S. Behnke, "Real-time object detection, localization and verification for fast robotic depalletizing," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sept 2015, pp. 1459–1466.

[3] R. Jonschkowski, C. Eppner, S. Hofer, R. Martin-Martin, and O. Brock, "Probabilistic multi-class segmentation for the amazon picking challenge," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct 2016, pp. 1–7.

[4] A. C. Muller and S. Behnke, "Learning depth-sensitive conditional random fields for semantic segmentation of rgb-d images," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, May 2014, pp. 6232–6237.

[5] S. Li, S. Koo, and D. Lee, "Real-time and model-free object tracking using particle filter with joint color-spatial descriptor," in *2015 IEEE/RSJ International COnference on Intelligent Robots and Systems (IROS)*, Sept 2015.