Grasp objects in 2D images

Minghuang Ma, Wei-chiu Ma
{*minghuam, weichium*}@*andrew.cmu.edu*

1. Introduction

In this project, we seek to understand how a robotic hand grasps an object given only a monocular image input. Although the grasp planning problem has been addressed by many previous works, most of these works relies heavily on 3D perception. We utilize the advances of computer vision in the past few years, especially the success of deep neural networks and developed a proof-of-concept framework to plan grasps given only a 2D image. In particular, given a 2D image and a robotic hand with fixed posture, our framework predicts how to grasp the object in the image.

From the perception point of view, grasp planning can be approached in a few different ways. (1) Data-driven approach: To solve this problem using machine learning algorithms, it usually requires a large amount of image data and grasp annotations. For example, Lenz *et al.* [2] trained a convolutional neural networks(CNN) to directly predict grasps using RGB-D data and achieved promising results. This approach can also work on RGB-only data. However, it is difficult to generalize the learned model to novel objects. (2) Geometry-based approach: This approach uses geometric information from images like edges and gradients or uses other information like depth data to reconstruct 3D geometry of the object. However, this requires large feature engineering efforts. In our particular problem, given only a monocular image, it would be very difficult to accurately obtain 3D geometry of the object. There are also other possible approaches. We instead adopt another approach based on an offline grasp database.

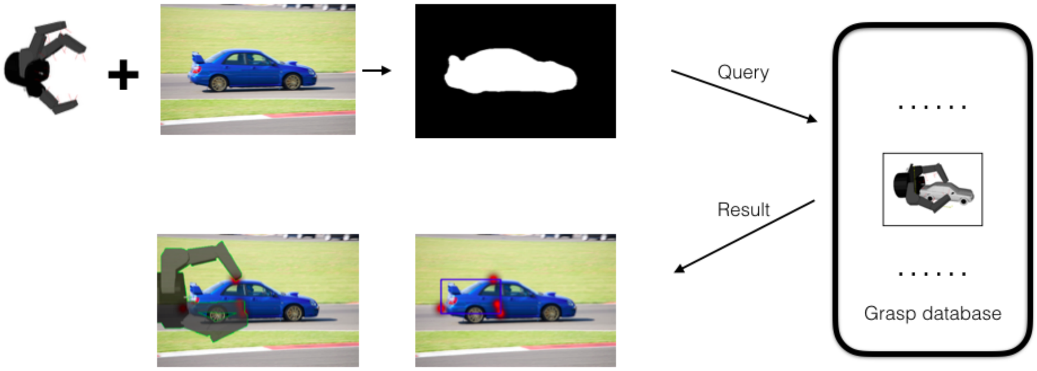

Figure 1: The overview of our approach.

The overview of our approach is shown in Fig. 1. Given a 2D image, we first detect the object in the scene and obtain the posture of the object. Then, we run a query using the object shape and match the object with the closest 2D view in the grasp database. At last, we map the grasp information back to the 2D image. Although there are better ways to estimate and match the pose of the object with the 3D object in the database, we only use the shape of the object mask for this proof-of-concept project.

2. Related Work

Grasp planning. In the robotic manipulation area, grasp planning has been discussed in many previous works. Bicchi *et al.* did a survey [1] on previous success in designing methods to achieve high-quality grasp. To measure the quality of robotic grasps, many different metrics have been proposed. Miller *et al.* [4] proposed a grasp analysis system that, when given a 3D object, hand and hand posture, can determine the types of contacts and compute two measures of quality for the grasp. In the simulation world, Miller *et al.* developed a 3D simulation tool GraspIt! [5] for grasping research. The tool provides an 3D interface and allows user to manipulate a robot or object and create contacts. The tool also includes a set of grasp planning algorithms and grasp quality metrics.

Machine learning based algorithms. Besides analytic solutions, a few other works also adopt machine learning algorithms for grasping research. Pelossof *et al.* proposed a SVM based system [7] to combine numerical methods and machine learning to find the optimal grasp. Lenz *et al.* [2] proposes a deep learning approach to detect robotic grasps in an RGB-D view of objects. A two-step cascaded structure based on neural networks are presented to optimize the speed and accuracy of grasp detection and achieved the state-of-art result in robotic grasp detection.

Convolutional neural networks. The recent success of deep learning has been demonstrated in many classical computer vision problems such as image classification and object detection. In the robotics community, many works also used the same approach to solve robotics problems. In this project, we also use deep networks for the task of object detection in monocular images. In particular, Long *et al.* proposed a fully convolutional neural network [3] and achieved the state-of-the-art results in semantic segmentation. We use this model in our framework for object segmentation.

3. Grasp Database

To construct a grasp database, we utilize two open-source resources: (1) GraspIt! [5], a grasp simulation environment, and (2) Princeton Shape Benchmark [8], a 3D shape database for shape matching and classification.

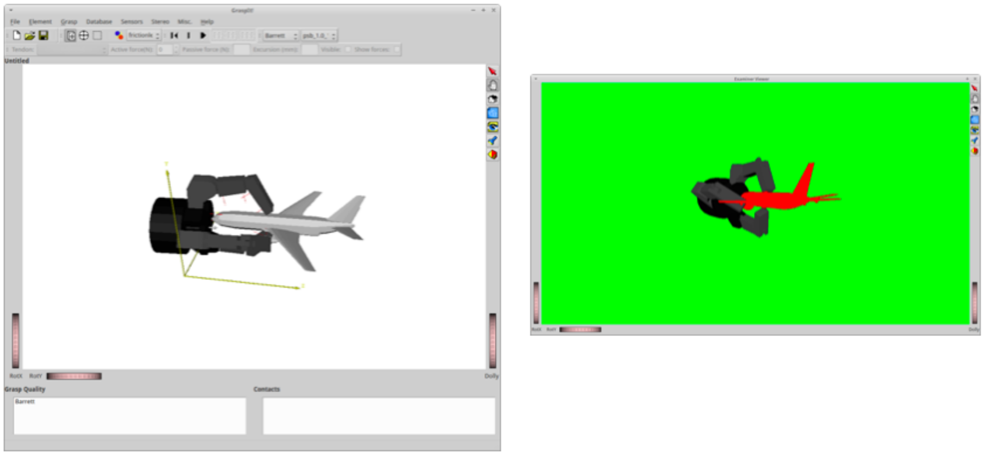

Figure 2: The GraspIt simulation tool user interface with our plugin

The GraspIt! tool. GraspIt! is an open-source tool for grasping research. It provides a 3D user-interface for users to interact with robots and objects. It also provides automatic grasp planning based on eigengrasps. This software was developed more than a decade ago and currently is *not* under active development. We spent efforts on understanding the code base and also fixed a few bugs in the original code. We host our modified version, which to the best of our knowledge is **bug-free**, at an open repository[1] in hope of advancing research in this area.

3D model database. To construct a grasp database of 3D objects, we use models from the Princeton Shape Benchmark [8]. This dataset was developed originally for shape matching and classification. In this project, we focus on three types of objects, *i.e.* cars, airplanes and horses. To simplify our task, we scale all objects into the size which a robot hand is able to grasp.

2D object views. Once the grasp planning is done, we then generate 360 views of all objects in the database in order to match with the 2D query object masks. To sample position of the camera in 3D space evenly, we use Fibonacci grids [9] as our sampling algorithm. We developed a plugin for view rendering in GraspIt!. The source code is also publicly available.

[1] https://github.com/minghuam/grasp-search

4. Object Detection

We use the deep neural network FCN-8s proposed in [3] to segment the interested object. Instead of using fully connected layers for the last layers in the network to predict labels for input images, this network converts these layers to fully convolutional layers to predict labels for each pixel. Given a input RGB image, the output of the network is a probability map for each pixel. The network is trained using the Pascal-Context [6] image semantic segmentation dataset. In our experiments, we use images from the test set. Example results for object detection are shown in Fig. 3.

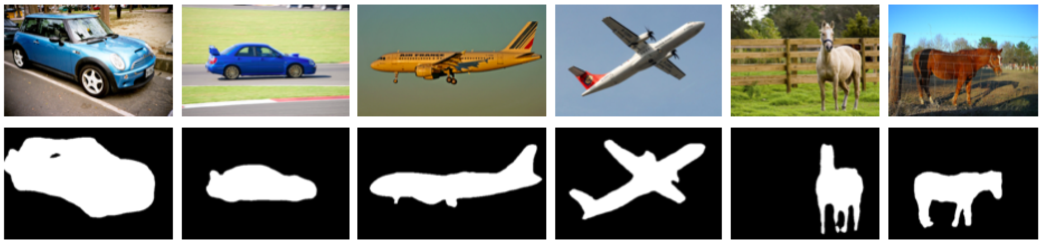

Figure 3: Object segmentation results using CNN

5. Object Pose Matching

Given the object mask and views of 3D objects in the database from 360 degrees, we match the object mask with each view and return the closest view. To accurately match the two images, ideally we need to match both the shape of objects and object attributes like edges. In our experiments, we only match the shape of the object by downsize the two binary masks and compute the hamming distances. Then we align the centroids of the masks and translate the robotic hand and contact points as return results for the query.

6. Results

We show results on three types of objects with increasing geometric complexity: car, airplane and horse.

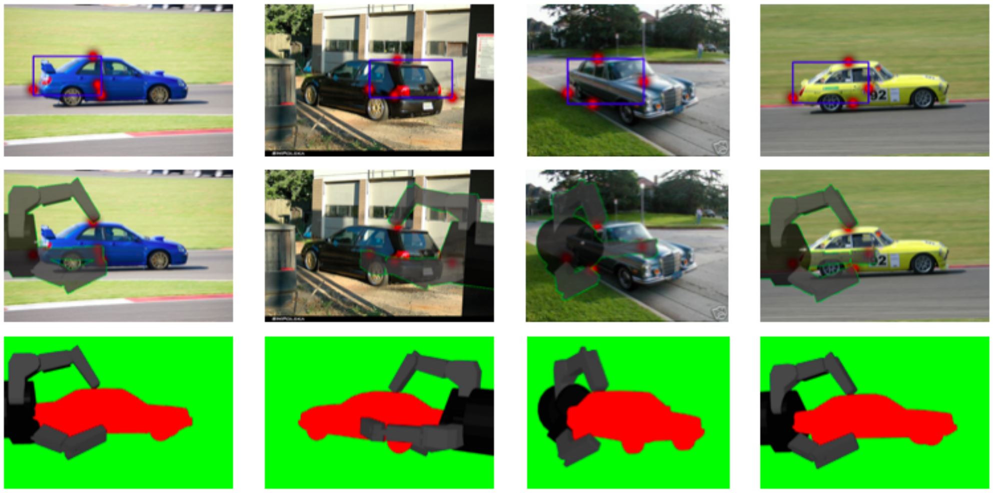

Figure 4: Grasp planning results on **car** images.

7. Conclusion

In this project, we developed a framework to combine analytic grasp planing techniques and latest advances in computer vision to predict how to grasp an object in a RGB image. In particular, we use GraspIt! to plan robotic grasp and build a

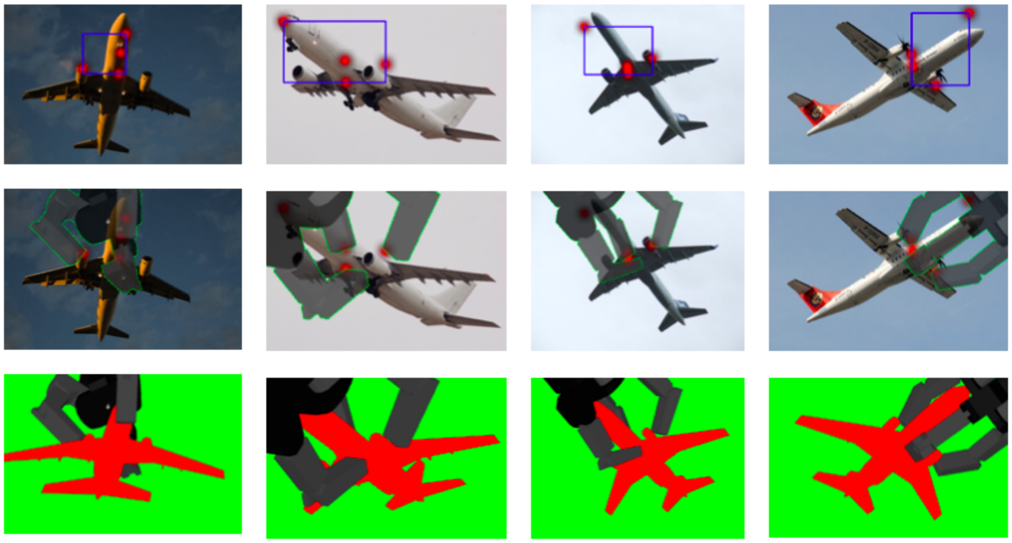

Figure 5: Grasp planning results on **airplane** images.

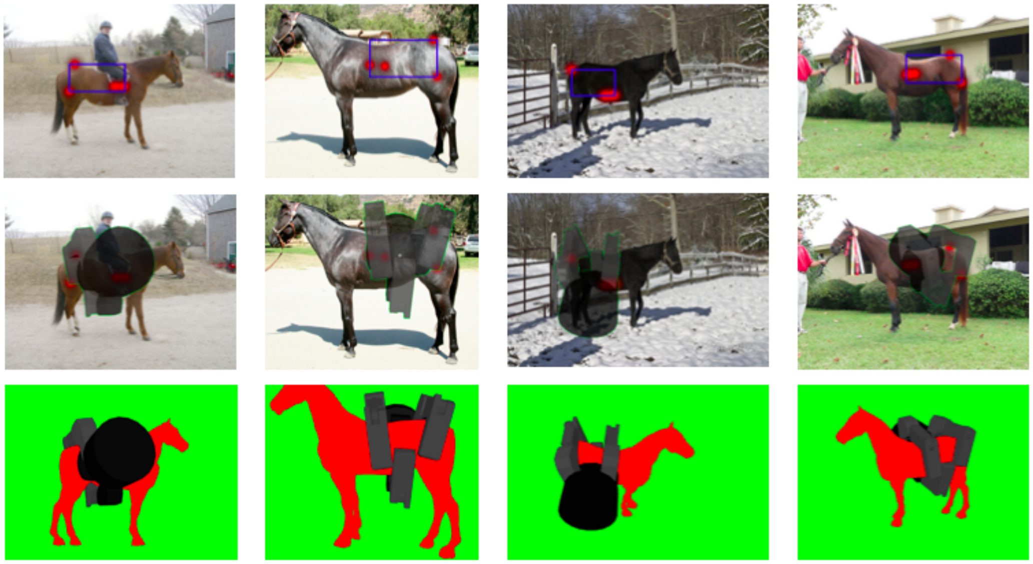

Figure 6: Grasp planning results on **horse** images.

grasp database using 3D object models. We use CNN to detect object in the image. We then query the database and find the closest object pose in the database and then map the grasp information back to the 2D image. We test our framework on test images from the Pascal-Context dataset and show qualitative results.

References

[1] A. Bicchi and V. Kumar. Robotic grasping and contact: A review. In *ICRA*, pages 348–353. Citeseer, 2000. 1

[2] I. Lenz, H. Lee, and A. Saxena. Deep learning for detecting robotic grasps. *The International Journal of Robotics Research*, 34(4-5):705–724, 2015. 1, 2

[3] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. *arXiv preprint arXiv:1411.4038*, 2014. 2, 3

[4] A. T. Miller and P. K. Allen. Examples of 3d grasp quality computations. In *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on*, volume 2, pages 1240–1246. IEEE, 1999. 2

[5] A. T. Miller and P. K. Allen. Graspit! a versatile simulator for robotic grasping. *Robotics & Automation Magazine, IEEE*, 11(4):110–122, 2004. 2

[6] R. Mottaghi, X. Chen, X. Liu, N.-G. Cho, S.-W. Lee, S. Fidler, R. Urtasun, and A. Yuille. The role of context for object detection and semantic segmentation in the wild. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. 3

[7] R. Pelossof, A. Miller, P. Allen, and T. Jebara. An svm learning approach to robotic grasping. In *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, volume 4, pages 3512–3518. IEEE, 2004. 2

[8] P. Shilane, P. Min, M. Kazhdan, and T. Funkhouser. The princeton shape benchmark. In *Shape modeling applications, 2004. Proceedings*, pages 167–178. IEEE, 2004. 2

[9] R. Swinbank and R. James Purser. Fibonacci grids: A novel approach to global modelling. *Quarterly Journal of the Royal Meteorological Society*, 132(619):1769–1793, 2006. 2