Robotic Pick-up of Objects

"Robotics and other combinations will make the world pretty fantastic compared with today" ∼Bill Gates

Aniksha Mahala
B21ME008
mahala.1@iitj.ac.in

Bhavya Mehta
B21ME018
mehta.7@iitj.ac.in

Jyoti Bhookar
B21ME031
bhookar.1@iitj.ac.in

***Abstract*—**This project proposes a grasping system for diverse objects on a flat surface. This proposes novel algorithms for object detection and optimal grasping point determination for 3D point clouds and 2D images in robotic applications. For 3D data from CAD models, techniques like Convex Hull, Fast Geometry-based Grasping, and Minimum Value Bounding Box are introduced. 2D image processing employs methods such as Convex Hull, single-shot detection, aspect ratio analysis, and Distance Transform. The implemented algorithms leverage image processing techniques to enhance autonomous robotic grasping and pick-and-place capabilities, advancing automation across industries.

I. Introduction

In the realm of robotics, the capability to detect and grasp objects autonomously stands as a fundamental challenge and opportunity. Object detection and grasping are pivotal components of automation, with applications across industries ranging from manufacturing to healthcare. Understanding the evolution and complexities of object detection and grasping in robotics is essential for industries seeking to leverage automation to optimize efficiency and productivity. Sophisticated object detection, grasping points detection, and path planning algorithms enable robots to navigate obstacles and ensure safe object movement.

Literature Review:

Robotic pick-and-place operations with various approaches are proposed to address the challenges of object recognition, pose estimation, and grasp planning.
Traditional methods have relied heavily on supervised learning techniques, which require extensive task-specific training data. For instance, several studies have explored the use of deep learning models trained on large datasets of labeled RGB images for object detection and pose estimation. While these approaches have shown promising results in controlled environments, their performance can degrade when dealing with novel objects that differ significantly from the training data.
Given are the examples of some already existing researches in this field:

Robotic hand is used in image processing and for carrying out various applications for robotic hand. The system extracts edge information from a webcam image, differentiates objects using color coding, and optimizes the data for smooth and efficient drawing by a robotic arm. Here, the Canny algorithm is applied for edge detection.To draw efficiently, the robotic arm starts from a "start pixel" (one edge point with only one neighbor) in each contour. Neighboring edge points are then sorted by distance to the current point, guiding the arm smoothly along the contour. [1]

The paper describes a robotic application that can simultaneously detect a moving object and avoid obstacles in real-time. The architecture of the system comprises of: Image Input Phase: Web Camera is used to capture robots images. Object Detection Phase: Pre-processing is done on the captured image which is stored in the hard-drive to convert it to a gray-scale image. A histogram is then developed from the grayscale image to identify the region containing the desired object. This region from the captured image is compared against a previously stored reference image of the target object. Robot Control Phase: If captured image matches with the image of an object then robot moves towards the object, else it takes a turn and re-capture image. The system also comprises Obstacle Detection Phase and Obstacle Avoidance Phase to tackle the obstacles in path using IR sensor. [2]

An autonomous robotic system - DoraPicker- is represented that can perform picking and placement operations in a simplified warehouse scenario. They proposed the LLSDPso method to identify dynamic parameters and friction coefficients of robotic manipulators using only torque measurements from the joint motors. It divides parameters into linear (identified by least squares) and nonlinear parts (identified by dual-swarm PSO). A key novelty is simultaneously identifying parameters of two joints without error accumulation. It also includes a verification step to ensure robustness. Experiments on a 7-axis robot showed most parameter errors were below 10 percent compared to ground truth. [3]

It involves determining object poses for proper end-effector orientation. Comparing object geometries to recognize the target. Identifying suitable gripping points ensures gripper compatibility. Image processing techniques are essential for accurate decision-making in these steps. [4]

II. Proposed Method

In this project, we have primarily worked on the development of object detection and determining optimal grasping points in 3D and 2D objects. For developing the 3D objects data we have converted the CAD's .stl file into point cloud data. Below are some of the proposed algorithms which can be used for the specified tasks:

Convex Hull-based Object Detection:

The Convex Hull algorithm is a fundamental concept in computational geometry used to determine the smallest convex polygon that encloses a set of points in a plane or a convex polyhedron in 3D space. One of the classic algorithms for computing the convex hull is the Jarvis march, also known as the gift wrapping algorithm due to its analogy of wrapping a string around a set of points like a gift. The algorithm starts by selecting the leftmost point in the point cloud as the starting point of the convex hull. It then iteratively finds the next point on the hull by considering the angle formed between the current point, the next point under consideration, and all other points. The point with the smallest polar angle (counterclockwise) relative to the current point is chosen as the next point on the hull. This process continues until the algorithm returns to the starting point, completing the convex hull.

Fast Geometry based Grasping-

Initially, the scene can be divided into separate objects using point cloud data. This involves removing background points and identifying distinct objects. For each object, areas suitable for grasping are identified. This involves finding the centroid (center of mass) and main axis of the object. A cutting plane can be determined to isolate a portion of the object's surface. Grasping areas are then can be defined based on geometric rules, considering the orientation of the object relative to the table and camera. Points within the grasping areas are evaluated based on factors such as their distance from a cutting plane, curvature of the surface, alignment for stable grip, and perpendicularity to the object's main axis. A ranking function can assess these factors to determine the best pair of grasping points. Once the optimal grasping points are identified, the hand pose needed to grasp the object is determined. This involves positioning the hand so that its fingers align with the selected points on the object's surface, considering factors such as finger spread and orientation relative to the object's axis. [5]

GG-CNN (Grasping Geometry CNN):

GG-CNN takes images as input, typically RGB-D images. These images represent the scene containing objects that the robot needs to grasp. Like traditional convolutional neural networks (CNNs), GG-CNN consists of multiple convolutional layers. These layers extract features from the input images, gradually learning hierarchical representations of the visual information relevant to grasp detection. GG-CNN is designed as a fully convolutional neural network. This means that it operates on the entire input image, pixel by pixel, without the need for fully connected layers. This architecture allows GG-CNN to output grasp configurations for each pixel in the input image. The output of GG-CNN is a grasp configuration map, which provides predictions for grasp parameters at each pixel location in the input image. These parameters typically include the position, orientation, and width of the gripper needed to grasp an object effectively.GG-CNN is trained on annotated data, where each image is labeled with ground truth grasp configurations. During training, the network learns to minimize the difference between its predicted grasp configurations and the ground truth labels.

Minimum value bounding box:

This algorithm is a fit-and-split approach for approximating a 3D point cloud by a constellation of minimum volume bounding boxes (MVBBs). It begins by fitting a root bounding box and then estimating the best split by utilizing 2D projections of the enclosed points onto each of the box surfaces. Depending on a volume gain parameter, the algorithm may produce two child boxes, which are then tested for further splitting. Subsequently, the identified regions are extracted from the contours based on four main conditions: low curvature at any point, limited total accumulated curvature. The algorithm is used for determining the graspable points. [6]

III. Result

Our focus has been on object detection and grasping planning for objects in a captured image. We first generated CAD models that contained different geometric shape data. Then we converted the CAD-generated .stl file into point cloud data for further implementation.

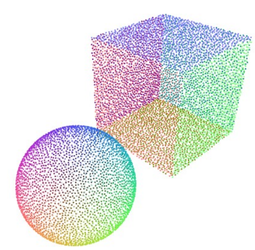

Fig. 1. Point cloud object

Object detection: We have implemented the use of Convex Hull Algorithm for Object detection in 2D and 3D images. For 2D, OpenCV is used to perform grasping point detection on an image using the Convex Hull-Based Grasping technique. Canny edge detection is applied to identify object boundaries, followed by contour detection to find continuous curves outlining regions in the image. For 3D, The indices of the vertices forming the convex hull are extracted. Subsequently, the actual vertices of the convex hull are obtained from points using these indices. The center and size of the boundary boxes are computed by calculating the minimum and maximum coordinates along each axis of the convex hull vertices to define the bounding box.

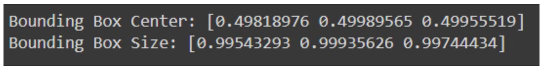

Fig. 2. Bounding Box dimensions

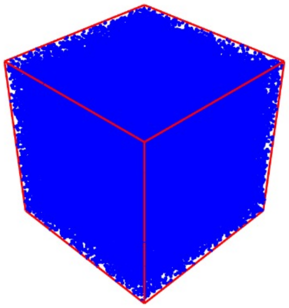

Fig. 3. 3D object detection

Another Approach for 2D image object detection: We have employed single-shot detection techniques for object detection. They use a single convolutional neural network (CNN) to predict bounding boxes and class labels for objects in an image, making them faster and more efficient than other methods. We implemented it using a pre-trained model of OpenCV python library. Also, we have implemented YOLO for obtaining 3D bounding box on video data.

Fig. 4. Grasping points

Grasping point algorithm: For 2D images, aspect ratio has been utilized to identify the positions from which the actuators can grasp the objects for an improved grip. For 3d objects, grasping point detection is one by using Distance Transform-based algorithm. Canny algorithm is performed on the grayscale version of the image. Subsequently, the edges are dilated to enhance their visibility and continuity. A distance transform is then computed on the dilated edges, which assigns intensity values based on the distance to the nearest background pixel. A thresholding operation is applied to isolate grasping points based on their distance values. Contours are extracted from the thresholded image, representing potential grasping regions. Link for the colab file for the above code: Colab File

IV. Conclusions

This project focused on developing object detection and optimal grasping point determination for 3D and 2D objects. The project involved generating CAD models, converting them to point clouds, and implementing these algorithms on both 2D and 3D data. Various algorithms like Convex Hull, Fast Geometry-based Grasping, and Minimum Value Bounding Box were proposed and implemented. For 2D images, techniques like Convex Hull, single-shot detection with pre-trained models (OpenCV, YOLO), aspect ratio analysis, and Distance Transform-based algorithms were utilized for object detection and grasping point identification.

V. Contributions

Aniksha Mahala- Researched about the various methods and algorithms for grasping point detection. Worked on the determination of grasping points for the generated point cloud data using algorithms like Fast Geometry-based Grasping. Also worked on object detection using SSD, with the help of CNN model using PyTorch.

Bhavya Mehta- Worked on the use of Convex Hull-based Object Detection for both 2D and 3D objects. Canny edge detection is applied to identify object boundaries, followed by contour detection. Also worked on converting the stl files into the point cloud data an rgbd data. Worked on the implementation of Distance Transform-based algorithm.

Jyoti Bhookar- Came up with the approach for grasping point allocation of 2D objects using aspect ratio. Also worked on finding the grasping point using Distance Transform-based algorithm. Worked on the implementation MVBB algorithm for determining the grasping points for 2.5D data.

References

[1] Dragusu, M., Mihalache, A. N., & Solea, R. (2012, October). Practical applications for robotic arms using image processing. In 2012 16th International Conference on System Theory, Control and Computing (ICSTCC) (pp. 1-6). IEEE.

[2] Borse, H., Dumbare, A., Gaikwad, R., & Lende, N. (2012). Mobile robot for object detection using image processing. Global Journal of Computer Science and Technology Neural & Artificial Intelligence, 12(11), 13-14.

[3] Zhang, H., Long, P., Zhou, D., Qian, Z., Wang, Z., Wan, W., ... & Pan, J. (2016, August). Dorapicker: An autonomous picking system for general objects. In 2016 IEEE International Conference on Automation Science and Engineering (CASE) (pp. 721-726). IEEE.

[4] Hameed, F. S., Alwan, H. M., & Ateia, Q. A. (2020). Pose estimation of objects using digital image processing for pick-and-place applications of robotic arms. Engineering and Technology Journal, 38(5A), 707-718.

[5] Zapata-Impata, B. S., Gil, P., Pomares, J., & Torres, F. (2019). Fast geometry-based computation of grasping points on three-dimensional point clouds. International Journal of Advanced Robotic Systems, 16(1), 1729881419831846.

[6] Geidenstam, S., Huebner, K., Banksell, D., & Kragic, D. (2010). Learning of 2D grasping strategies from box-based 3D object approximations.