Rover Manipulation

A project of the 2019 Robotics Course of the School of Information Science and Technology (SIST) of ShanghaiTech University

https://robotics.shanghaitech.edu.cn/teaching/robotics2019

Ruiqing Chen,Yang Li,Liang Chen
{ chenrq,liyang2,chenliang1}@shanghaitech.edu.cn

January 7, 2020

1 Abstract

Manipulation is one of the hottest topic in Robotics. Picking and place is one of the typical application in manipulation. There are various methods in this field can be performed. The important parts of pick and place operation include recognizing the object which will be grasped, and obtaining the global positions of object, how to grasp it and move it to destination smoothly and correctly . In this project, the robotic arm can obtain the height of sand by a ZED camera, then grasp the shovel to shovel the sand. Finally robotic arm will smoothly put the sand to destination.

2 Introduction

Nowadays people are basically superseded by robot in performing risky and dangerous task which are not to done due to size, environment constraints. One of the most important part in this is manipulation, which are undergoing important and dramatic changes daily. And advancements in the field of automation and robotics technologies are also contributions towards it. The project of the Robotics is rover manipulation, which need the robot find where is the shovel firstly, then move the robotic arm to close to the shovel, grasp the shovel and then move it to the area of scoop sand. Having done this, the robot needs to move the sand carefully to the specific place.

Machine vision has better sensitivity and resolution. It tends to application for identifying and inspecting details, guiding the robot controllers. Vision is very important in automation industry, which can be used for number of industry application such as welding, painting, automatic pick and place object, military operation, in space, in underwater operation, in agriculture operation etc. One of the function of vision is to identify shovel, which the robot know where it is and how to proceed in order. Specifically, we will fix the ZED Camera on the side of work scene to obtain the height of sand, which based on point cloud from ZED camera, the ZED point cloud was used algorithm to get the height of sand. Having done this, the robot can move the arm to the place of shovel and move it to do some tasks. The test tasks will be divided into two steps:

- **Experiment 1**, the robotic arm holds the shovel and knows where the sand is, and then uses an algorithm to move the robot arm to the sand, shove the sand, and finally pours the sand into the pre-prepared collection device.
- **Experiment 2**, the shovel is somewhere on the ground, the robotic know its position, then move there, and grab the shovel, move to the position of sand to shove the sand, but robot don't know the height of sand, so we need depth camera to get it, and finally the sand in the shovel is poured into the pre-prepared collection container.

In this project, a ZED camera will be used to get the height of sand. Then using ROS packages to drive the robot arm to grasp shovel and sands. Having done this, the robot smoothly transports the sands to the specific place or some place the robot don't know, the position of destination can be obtained by depth camera.

3 State of Art

3.1 liyang

(N. Roshni et al.,2017)[7]mainly concentrate on the method of differentiating the object from it surrounding using image processing and picked through its centre of gravity. They discuss about how the pick and place operation is performed with the artificial intelligence incorporated on the ABBIRB1200 robot so that the robot can itself pick objects oriented in different direction utilizing the centre of gravity which help in increasing the precision with which pick and place operation can be performed.

(G. Zhu et al.,2009) [10] used Differential Evolution Optimization Approach to do pick and place task. The Differential Evolution algorithm for pick-and-placing problem of multi-head surface mounting machines is studied, and the research work has two contributions to the multi-head case, one is that the algorithm has a special mutation operation; the other is the crossover operation. Experimental results show that the proposed approach gives good robot assembly time and less movement of the feeder carrier and PCB table.

(P. Andhare et al.,2016) [1]give an idea about finding the x,y coordinate of object. This is found by converting pixel co-ordinate into real world coordinate with the help of 2D transformation. They incorporate robotic arm control based on location and orientation of object. Fundamentally, the problem taken in this work is to pick and place objects which are placed under vision sensor. Machine Vision is vast area of interest and is related to computer vision. It tends to application for identifying and inspecting small parts, guiding the robot controllers. The objective of these visual servoing approach is control 6 degree of freedom robot with help of information provided by vision sensor. The whole work is divided into 4 subprocess. The first is scene analysis, which consists of object position and orientation calculation. The second is co-ordinate conversion between camera co-ordinate and robot co-ordinate. The third is path planing based on vision system.The last is interfacing vision system to robot, which consists of ethernet and socket communication.

Because we need to install a ZED Stereo Camera to recognize the object so the package of ZED will be used in our project. The ZED Stereo Camera is a lightweight depth camera based on passive stereo vision. It outputs up to 2208x1242 high-resolution stereo video on USB 3.0, and can work in high-speed mode in VGA at 100 FPS. The ZED SDK computes a depth map of the environment on the GPU of the host machine at the frame rate of the camera. It also gives you access to a strong odometry that learn its environment, making it able to relocate itself at any moments.The ZED ROS wrapper lets you use the ZED stereo cameras with ROS. It provides access to the following data:Left and right rectified unrectified images;Depth map;Colored 3D point cloud; Visual odometry: Position and orientation of the camera;Pose tracking: Position and orientation of the camera fixed and fused with IMU data (ZED-M only).

3.2 chenliang

(Tong Li et al.,2018)[6]propose a method to distinguish whether grasping operation is completed. With this method, grasping operation can be achieved with low contact force and fine stability. Then, the paper do some experiments to verify the significant advantage with tactile sensor array in grasping operation than other kinds of sensors. In the end, the paper said that more work such as recognition, exploration and perception can be devoted with tactile sensors with the information. (Kene Li et al.,2018)[5] present a cucumber picking motion planning scheme at velocity level based on the pseudo inverse method that in order to realize the automation and intelligence of agriculture product picking operations, and to improve the efficiency of picking operation. In the conclusion, the neural-dynamic design method has a simple architecture and makes the actual tacking path rapidly convergent to the desired path. The simulation results demonstrated the accuracy and effectiveness of the presented motion scheme used for the picking robot.

(Tomasz Szczesny et al.,2018)[8] aims to design control algorithm of the robotic arm in Mars rover analogue. The paper describes control of 6 degrees of freedom robotic arm. There are three parts in the paper: first step is the determination of DH parameters and transformation matrix for simple kinematics, and then, the using geometric and analytic method to determine the inverse kinematics, later, determine interpolation algorithm, finally, carry out a simulation of manipulator movement made of real components. Based on the results of the simulation, the maximum velocity of the robotic arm movement can be determined which help to create the most natural control scenario for the rover's operator.

In order to control the robotic arm, kinova-ros package will be used. The ros package include joint position control which could command a certain joint node to rotate a specific angle and print the other massage about the joint, Cartesian position control which could command arm to move a specific distance along certain direction, figure position control which could open or close the figures, speed control of joint and Cartesian space which could obtain the linear velocity and angular velocity of a certain joint node.

Also, there are some other functions like ROS service order which could make the arm returns its original position, Cartesian admission mode which could give users permission that move the robot arm by applying force/torque to the end effector/joint, recalibrate the torque sensor that ensure zero torque at the joint, torque control which could release torque/force command like joint/Cartesian speed.

3.3 chenruiqing

(Daniel M et al.,2018) [2]propose a fast way to detect,locate and grasp object with a robotic arm. To find the object, they propose a method based feature, being the feature extractor the ORB algorithm. Together with RANSAC, it is going to find the object position in the RGB image, and find the pixels coordinates in the point cloud. After the object detection, the propose a way to grasp based on the estimated position by the point cloud and the robotic arm.

(Nishchal K et al.,2016)[9] present a vision base object grasp methodology for a robotic manipulator,they discussed about kinematical modeling of five DOF robotic manipulator which deals with both forward and inverse kinematics based on analytical approach and a MSER features based methodology for object localization with help of density based clustering and homography transform. Detected centroid of object is transformed into base frame with the help of calibration. Algorithm is verified on hardware for object grasping mainly for rectangular boxes with RGB-D camera.

(Cagri K et al.,2018)[4] using raspberry pi to implement the algorithm on a robotic arm for object detection and recognition, with these algorithms, the objects that are desired to be grasped by the gripper of the robotic arm are recognized and located.They using 4 degrees of freedom (DOF) OWI–535 robotic arm, it is purposed that the recognized object is grasped in the workspace of the robotic arm and dropped to the desired target. For this purpose, an appropriate experimental setup is established. Object recognition and localization operations are performed with software written in C++ language using Open Source Computer Vision (OpenCV) library on Raspberry Pi, which is a Single-Board Computer (SBC) based on Linux operating system. Such a study is performed for the first time on OWI–535 robotic arm. Its shows that local feature-based algorithms are useful for applications of education and industry.so we think maybe we can apply ORB algorithm to the kinova robotic arm.

we use ros package Moveit to process the motion planning when robotic arm proceed tasks. Moveit is a easy-to-use robotics manipulation platform for developing applications, evaluating designs, and building integrated products, which have six main function as the following: First, Motion Planning: Generate high-degree of freedom trajectories through cluttered environments and avoid local minimums. Second, Manipulation:Analyze and interact with your environment with grasp generation. Third, Inverse Kinematics:Solve for joint positions for a given pose, even in over-actuated arms. Fourth, Control:Execute time-parameterized joint trajectories to low level hardware controllers through common interfaces. Fifth, 3D Perception:Connect to depth sensors and point clouds with Octomaps. Sixth, Collision Checking:Avoid obstacles using geometric primitives, meshes, or point cloud data.

3.4 Relevant Paper

(García, Néstor and Suárez at.,2018)[3] addresses the problem of obtaining human-like motions on hand-arm robotic systems performing grasping actions. This paper captures different grasps of human movements performing and mapped to the robot in order to compute the human hand synergies. In addition, this paper proposes a sampling-based planner, which guides the motion planning following the synergies and considering different types of grasps. In this paper, the main key points of the proposed approach are the following:

- Capture the motions of a human operator performing different types of grasp on several objects
- Computing the synergies in the human motions through the captured information
- A bidirectional sampling-based planner is designed to bias the tree growth towards the directions of the computed synergies and to reduce the dimension of the search space.

There are three steps of their work:Motion capture and mapping, motion analysis and motion planning. In their experiments:

AVERAGE RESULTS OF THE MOTION PLANNING WHEN RUNNING THE CLASSIC RRT-CONNECT (a) AND THE PROPOSED APPROACH WITH THE PROPER (b) AND WITH MISMATCHED GRASP SYNERGIES (c).

Case	Success rate	Planning time	# iterations	# collision checks	Valid segments	Path length	Human-likeness
a	97 %	51.80 s	1834	32231	68.3 %	14.18 rad	73.6 %
b	100 %	6.21 s	274	10649	80.0 %	7.79 rad	83.1 %
c	100 %	11.79 s	484	13667	75.3 %	8.35 rad	81.9 %

4 System Description

Our task is that use robotic arm to grasp shovel, and then move the robotic arm to close to the sand in box(or ground),shovel the sand and put them into another specific box. In the initial case, we assume that we already know where the sand, the container, the shovel are. We needn't to get the poses of those objects, but we need to use pointcloud segmentation algorithms to get the height of the sand pile so that the robot arm can decide where to shovel the sand. Having done this, we can use the RGB-D camera to get the position of those objects.

4.1 Algorithm

In order to calculate the height of sands based on point cloud, we use ZED stereo camera get the point cloud of the scene,then use Sample Consensus(SAC) Segmentation algorithm to extract the planes of ground and wall. Having done this, we can obtain the point cloud of sand and get the peak of the sand.

4.1.1 RANSAC Segmentation

RANSAC (Random Sample Consensus) is a randomized algorithm for robust model fitting.The Random Sample Consensus (RANSAC) algorithm proposed by Fischler and Bolles is a general parameter estimation approach designed to cope with a large proportion of outliers in the input data. Unlike many of the common robust estimation techniques such as M-estimators and least-median squares that have been adopted by the computer vision community from the statistics literature, RANSAC was developed from within the computer vision community. RANSAC is a resampling technique that generates candidate solutions by using the minimum number observations (data points) required to estimate the underlying model parameters. As pointed out by Fischler and Bolles , unlike conventional sampling techniques that use as much of the data as possible to obtain an initial solution and then proceed to prune outliers, RANSAC uses the smallest set possible and proceeds to enlarge this set with consistent data points. The basic algorithm is summarized as follows:

Algorithm 1 Ransac Algorithm

1: Select randomly the minimum number of points required to determine the model parameters.
2: Solve for the parameters of the model.
3: Determine how many points from the set of all points fit with a predefined tolerance θ.
4: If the fraction of the number of inliers over the total number points in the set exceeds a predefined threshold γ, re-estimate the model parameters using all the identified inliers and terminate.
5: Otherwise, repeat steps 1 through 4 (maximum of N times).

We use the RANSAC algorithm to extract the planes like the ground and walls, and the point cloud of sand. Having done this, we could calculate the height of sand then publish it and others can obtain the height to determine where to shovel the sand.

4.2 Ros Package

The mainly ros packages we used in this project are kinova, moveit and pcl. Kinova is a ros package about robotic arm, which include some driver and description of hardware, making hardware can work well not only in virtual world but also in real world. PCL (Point Cloud Library) ROS interface stack. PCL-ROS is the preferred bridge for 3D applications involving n-D Point Clouds and 3D geometry processing in ROS. MoveIt is the most widely used software for manipulation and has been used on over 100 robots. It provides an easy-to-use robotics platform for developing advanced applications, evaluating new designs and building integrated products for industrial, commercial, RD, and other domains. By incorporating the latest advances in motion planning, manipulation, 3D perception, kinematics, control and navigation, Moveit is state of the art software for mobile manipulation.So we use it to plan our robotic arm to move more rationally and smoothly.

4.3 Code

This code mainly based on kinova arm moveit demo,pick and place, so we just attach function about the process of robotic arm proceed rover manipulation, not attach the details about each function appeared in the following code. Code as follow:

```
bool PickPlace::my_pick()
{
    clear_workscene();
    build_workscene();
    add_obstacle();
    //initial the position of robotic arm and gripper
    group_->clearPathConstraints();
    group_->setNamedTarget("Home");
    evaluate_plan(*group_);
    gripper_action(0*FINGER_MAX,0*FINGER_MAX,0*FINGER_MAX);

    group_->setPoseTarget(shovel_pose_);
    evaluate_plan(*group_);

    gripper_action(0.87*FINGER_MAX,0.87*FINGER_MAX,0.87*FINGER_MAX);

    group_->setPoseTarget(sand_pose_);
    evaluate_plan(*group_);
```

```
    setup_constrain(grasp_sand_pose_.pose, true, false);
    group_->setPoseTarget(grasp_sand_pose_);
    evaluate_plan(*group_);

    group_->setPoseTarget(start_transport_pose_);
    evaluate_plan(*group_);

    group_->setPoseTarget(box_pose_);
    evaluate_plan(*group_);

    group_->setPoseTarget(release_pose_);
    evaluate_plan(*group_);

    group_->setPoseTarget(shovel_pose_);
    evaluate_plan(*group_);

    //release the shovel
    gripper_action(0*FINGER_MAX,0*FINGER_MAX,0*FINGER_MAX);

    //move to the home position
    group_->setNamedTarget("Home");
    evaluate_plan(*group_);

    clear_workscene();
    return true;
}
```

The main procedure of robotic arm is that, first, use clear_workscene() function and build_workscene() to initial robotic arm's original work scene, and then move robotic arm to the home position, making robotic arm end-effector (i.e. finger) open as max angle. Next, move arm to the location of shovel, and use gripper_action() making finger close to grasp the shovel with appropriate power. Next, arm will move to the coordinate of sand, proceed grasp operation according to the poseStamped msg type sand_pose_ and grasp_sand_pose_. And then arm will adjust grasp pose and use setup_constrain() to avoid sand leaking in the process of transportation, when achieve the position of box, arm will adjust grasp angle to pour sand into box. Finally, arm put the shovel to its original position and return to home pose.

5 System Evaluation

After we implement the algorithm, we need make some experiments to evaluate our system. In order to do it, a series of experiments were designed.

- **Experiment** , the shovel is somewhere on the ground, the robotic know the position of it, then move there, and grab the shovel, move to the position of sand to shove the sand. But we do not know the height of the sand, so we will obtain the height of sand through the camera, and then decide the posture and the force of the robotic arm, finally the sand in the shovel is poured into the pre-prepared collection container.

As for the definition of successful of our system, when we put the shovel on an arbitrary position, then tell the robotic arm where the shovel is, the arm will do the following a series of continuous actions:

- Grad shovel
- Move to sand heap
- Go forward to obtain sand
- Keep shovel parallel to the horizontal plane
- Move to the top of container
- Put shovel to the original place

We define that all the above actions can be successfully completed twice is called a successful experiment. So, after that we will do ten experiments and record the number of successes and failures. When the probability of success is greater than 60 percent, we define it as a successful systemThe root reason why in the same situation, we will get different results when executing the same code is that the path planned by IK is not unique.

We have done ten times experiments, we recorded the time-consuming per test, and we have the following table: where S means the current experiment is successful and F means the current experiment is failure

Table 1: Experiments Results

1	2	3	4	5	6	7	8	9	10
38.68	39.19	35.41	39.82	37.84	42.23	39.77	45.23	38.88	39.87
S	S	S	F	S	F	S	F	S	S

We could obtain an average value by many times experiments. The calculate average time robot cost of formula as follows:

$$Time_c = \frac{1}{N}\sum_{i=1}^{N} t_i \tag{1}$$

where N is the number of repetitions, t_i is the time-consuming every experiment.

The average value of total experiment is 38.892s, the success rate is 70%. According to system description, we think our system is a successful system. The figure and picture of experiment show as follow:

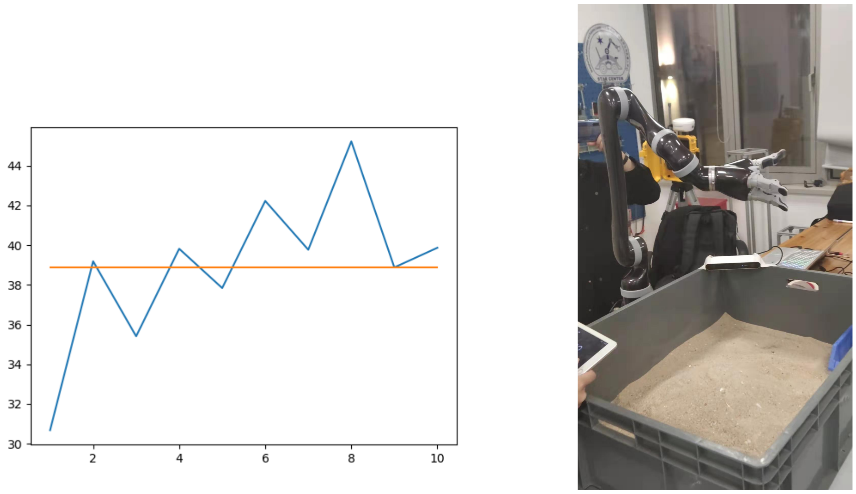

Figure 1&2. Line chart and experiment environment figure

6 How to Reproduce the Project

We have uploaded the code to our repo. **kinova_arm_moveit_demo**: the code of to proceed the process of grasp the sand and modify the position according to the depth of sand received by camera. **pcl_conversions/pcl_test**/:the code of pcl-segmantation and get the height of sands.

If you want to run this project, first, you need visit link to download ros package Jaco and visit link to download ros package Pcl.and visit link to install ZED driver

After installed those package, you can execute the following command to get the running code.

run "roslaunch zed_wrapper zed.launch" in terminal

run "roslaunch kinova_bringup kinova_robot.launch"

run "roslaunch j2n6s300_moveit_config j2n6s300_demo.launch"

run "rosrun kinova_arm_moveit_demo pick_place"

run "rosrun pcl_conversation pcl_test input=/zed/point_cloud/cloud_registered"

Having done this, you should wait the robot arm run the code. After all processes are done, push 'q' in terminal to end the code.

7 Conclusion

In this project, we use RANSAC segmentation algorithm to calculate the height of sands, and also use ros package kinova, pcl and moveit to process information and proceed task. We have met some trouble when proceed the motion plan, but finally we find method to solve it and accomplish this task successfully. This project still have some problem need to improve, for instance, use laser or camera to locate the position of object automatically, find more efficient ways to decrease the probability of obtain wrong motion planning trajectory and how to design motion to grasp more sand. Maybe it will be improved in the future.

References

[1] P. Andhare and S. Rawat. Pick and place industrial robot controller with computer vision. In *2016 International Conference on Computing Communication Control and automation (ICCUBEA)*, pages 1–4, Aug 2016.

[2] A. G. S. Conceição, D. M. Oliveira, and M. P. Carvalho. Orb algorithm applied to detection, location and grasping objects. In *2018 Latin American Robotic Symposium, 2018 Brazilian Symposium on Robotics (SBR) and 2018 Workshop on Robotics in Education (WRE)*, pages 176–181, Nov 2018.

[3] Néstor García, Raúl Suárez, and Jan Rosell. Planning hand-arm grasping motions with human-like appearance. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3517–3522. IEEE, 2018.

[4] C. KAYMAK and A. UCAR. Implementation of object detection and recognition algorithms on a robotic arm platform using raspberry pi. In *2018 International Conference on Artificial Intelligence and Data Processing (IDAP)*, pages 1–8, Sep. 2018.

[5] Kene Li and Yang Qi. Motion planning of robot manipulator for cucumber picking. In *2018 3rd International Conference on Robotics and Automation Engineering (ICRAE)*, 2018.

[6] Tong Li, Cheng Chi, Chunkai Wang, Ning Xue, Chunxiu Liu, and Chang Liu. Real-time and autonomous grasping operation of manipulator based on tactile sensor array. In *2018 2nd IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)*, pages 732–736. IEEE, 2018.

[7] N. Roshni and T. K. S. Kumar. Pick and place robot using the centre of gravity value of the moving object. In *2017 IEEE International Conference on Intelligent Techniques in Control, Optimization and Signal Processing (INCOS)*, pages 1–5, March 2017.

[8] Tomasz Szczęsny and Maciej Rećko. Control of robotic arm for mars rover analogue. In *2018 19th International Carpathian Control Conference (ICCC)*, pages 259–264. IEEE, 2018.

[9] N. K. Verma, A. Mustafa, and A. Salour. Stereo-vision based object grasping using robotic manipulator. In *2016 11th International Conference on Industrial and Information Systems (ICIIS)*, pages 95–100, Dec 2016.

[10] G. Zhu and Z. Chen. A differential evolution optimization approach to solve the pick-and-placing problem. In *2009 Fifth International Conference on Natural Computation*, volume 4, pages 66–70, Aug 2009.