Blueberry Jam: Universal Gripper With Computer Vision

Junji Hanatani, Jooyeop Kim, Renjie Li, Mike Park
Department of Mechanical Engineering
Columbia University
10027 New York, NY

***Abstract*—The objective of this project was to build a holistic system by building a jamming gripper attachment for the baxter and integrating computer vision for autonomous manipulation. The motivation for this project was to develop an intelligent robot with versatile grasping capabilities. The jamming gripper can achieve this versatility without complex mechanics and controls like multifingered hands. We were able to successfully integrate all the components and create a fully functional system. The mission of the project was to accurately recognize the object and carry it to the corresponding hole. Through 25 experiments, success criteria have been met for all objects except the square.**

I. Introduction

The overall goal for this project was to build a functional jamming gripper attachment for the Baxter Robot, as well as incorporating computer vision to enable the robot to autonomously pick and place specified objects into the corresponding hole. The baseline goal for this project was to design and build a working jamming gripper that is able to grasp objects using the camera on the Baxter and lifting the object. This baseline goal acted as proof that our design operated as a functional robotics manipulator. Therefore, if the gripper can grab any object and lift it, we regard it as success. Our main goal is to enable the robot to perform the assembly process. To be specific, through computer vision, the robot must recognize the object, find the hole that matches the surface of the object, and implement the process of accurately placing the object in the hole. Therefore, in addition to grasping, this goal includes recognition with vision sensor and comparing geometry between objects. The motivation for this project is to develop an intelligent robot with versatile grasping capabilities. The jamming gripper can achieve this versatility without complex mechanics and controls like multifingered hands. This technology can be used for manufacturing process (e.g. assembly process). The jamming gripper itself is so soft that it does not damage the object when moving it. Additionally, it is more advantageous in motion planning in terms of shape diversity of grasping, compared to normal gripper.

II. System

A. System Diagram

Figure 1 shows the diagram of our entire system. The jamming gripper with a vacuum pump is attached to the Baxter's left hand. The pump is driven by a motor, which is powered by 12V external power supply, and its driver communicates with the ROS server via Arduino (see details in II-C and II-E). The ROS server generates all signals for controlling the Baxter's arm as well as the pump of the jamming gripper. The motion of the baxter arm is basically based on vision feedback. In each time step, the Baxter's built-in camera captures a 2D image and sends it to the ROS server. In the ROS server, our object detection algorithms identify the pixel positions of the objects in the image and generate feedback signals based on PID method. (See details in II-G and II-F)

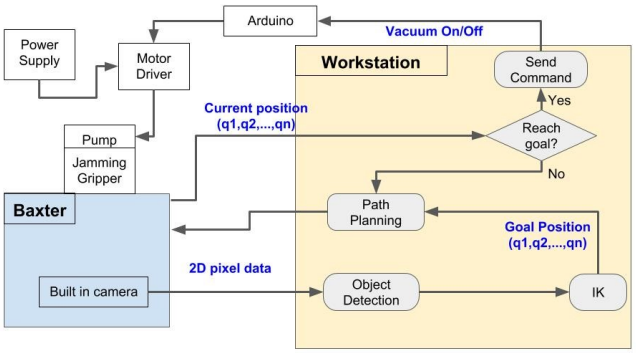

Figure 1. System Diagram

B. Differences from PDD

In our PDD, we suppose to use a RGB-D camera (e.g. Microsoft Kinect, Asus Xtion Pro) mounted on Baxter to identify the rough position of the objects and then move the Baxter arm to the corresponding location. But in our final product, this idea was abandoned due to the software issues. At the very beginning, we believe the port on the back of the Baxter robot will natively support the RGB-D camera. However, it doesn't. We tried another method: connect the Asus Xtion Pro to the computer and use the computer as a communication bridge between the Baxter and our jamming gripper. Disappointedly, it still cannot work. The RGB-D camera could only work properly (but not stably) on the administrator account in this computer. Because of this (plus we do not have enough time left – we find ourselves unable to solve this problem one month ago) , we decide to abandon

this idea. Instead, we would hardcode the rough position of the objects. This is the only difference from the PDD in our final product.

C. Hardware Components

The Jamming gripper works as follows: First, after the gripper arrives at the target object, wait for the balloon to fully enclose the object, and the pump works to suck the air and grasp the object.

Our gripper consists of the following components: balloon, coffee, air pump, tube, motor driver, Arduino, case, funnel and filter.

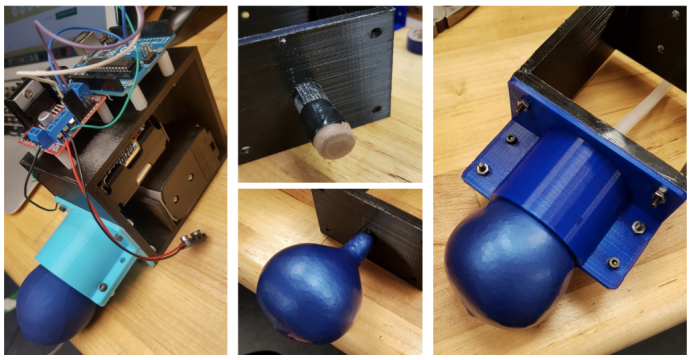

Figure 2. Hardware Components (from left to right: a,b,c)

In Figure 2(a), all the components are shown. The case has space for attaching the air pump and battery, while the Arduino and motor driver are attached on the outer wall of the case with spacers. Also, the case has a stem to attach filter and balloon, and connect the part with tube for air pump in Figure 2(b). The inner diameter of the stem is designed to match the outer diameter of the tube, and the outer diameter is set so that the balloon has sufficient tension. The balloon is filled with coffee as granular material. In Figure 2(c), the funnel is attached in a way that two separate parts are assembled, which adds to the usability in assembly.

Arduino board receives the signal from the ROS and operates the motor and controls the motor through the motor driver. The battery serves to power the motor driver to drive the pump, which was later replaced by a power supply for more powerful power.

D. Trial & Error for Design

While implementing the gripper performance with baxter, several problems occurred, and the design was modified accordingly as follows.

1) Hanging: When attaching the balloon to the stem without removing the band of the balloon, we can find the ballon hanging (which is can not be used as a jamming gripper) as shown in Figure 3(a). This problem was solved by shortening the balloon.

2) Moment: A moment occurred in the initial design because there was a distance between the point at which the gripper touches the object (ie, the center of the balloon) and the point at which the force of the baxter arm is applied. As

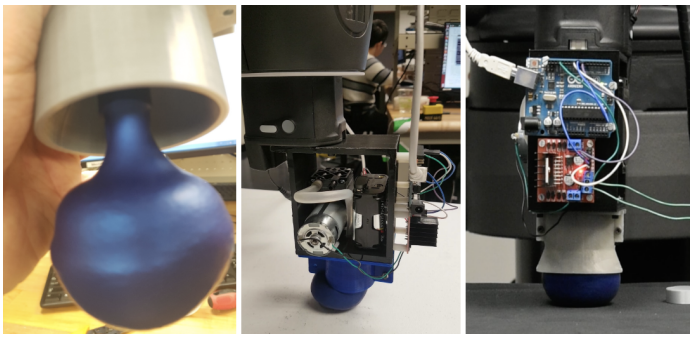

Figure 3. Modification for Gripper Design (from left to right: a,b,c)

a result, in Figure 3(b), the balloon was dragged, which has affected negatively the grasping performance. To solve this problem, the design was modified to reduce the distance. That is, the balloon is positioned at the center of the arm as much as possible.

3) Diameter of Funnel: For the jamming gripper to work properly, there must be enough space for balloons to spread. In previous design, however, the funnel used a minimum diameter to prevent interference with the camera's field of view(because the balloon was moved as close to the camera as possible to eliminate moment). It gave bad influence. Thus, even though there was some visual disturbances, the diameter of the funnel was increased and the better results were obtained.

4) Battery Power: During the experiment, due to the vibration generated by the pump and baxter, a stronger pump force was required. When using alkaline batteries, we could only use 30% of the pump's power, so we increased the power by connecting an external power supply instead of battery sets.

E. Serial Communication & Motor Control

Shown in Figure 1, Arduino Uno is the communication bridge between the motor driver (via Pin 7-9 & GND in Arduino, and in1-2 & enA in driver) ,the Ubuntu desktop (via USB Type B cable) and the Baxter. The motor driver that we use is L298N Dual H Bridge Motor Driver, it is powered by 12V power supply. The vacuum pump is connected to the vout port of the motor driver.

In our Baxter program, every time the Baxter moves to the right position, which is our approaching part (will discuss later), the program will publish an Empty topic named 'pump_on' and publish another 'pump_off' after all the grasping tasks are finished. The communication between Arduino and Baxter (ROS) is implemented by Rosserial. Arduino will continuously wait for the message published from ROS. When 'pump_on' is received, the Arduino will set pin 8 as high, pin 7 as low, (pin 9 is an analog pin that may control the power of the motor by PWM)which may turn on the motor and vice versa.

F. Object Detection

The goal of our object detection algorithms is to identify the location of objects that have simple shapes such as a square, a

triangle, and a circle, in the input image. To achieve that, we implement object detection algorithms as described below:

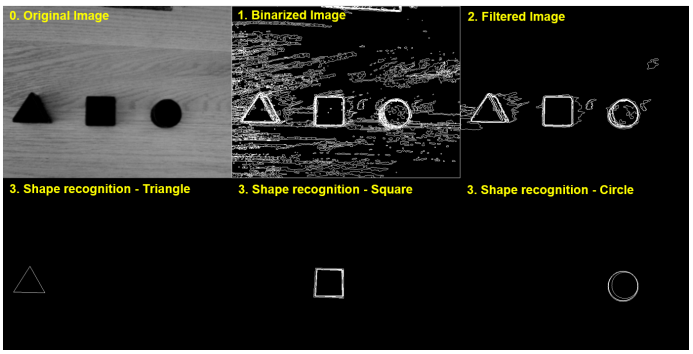

Figure 4. Object Detection

1) Image Binarization: Split original color image into R,G,B data and binarize each image using Canny edge detector and 5 threshold values. Thus, finally we have 6×3 (R,G,B) = 18 binary images.

2) Object Segmentation: For each binary image, draw contours and extract objects.

3) Filtering(Size&Aspect Ratio):

- Square: Object consists of 4 lines and the angles are approximately 90deg.
- Triangle: Object consists of 3 lines and the angles are approximately 60deg.
- Circle: Object area is nearly equal to the area of minimum enclosing circle.

4) Center Position Calculation: For triangle and square, we can obtain center positions by simply taking the average over all vertex positions. For the circle, we just use the center of the minimum enclosing circle of the objects (The function is available in the OpenCV library.)

G. Motion Planning

To achieve our goals, we divide the tasks into several steps and generate the motions for each task in an appropriate manner. These motions are sequentially executed as described below:

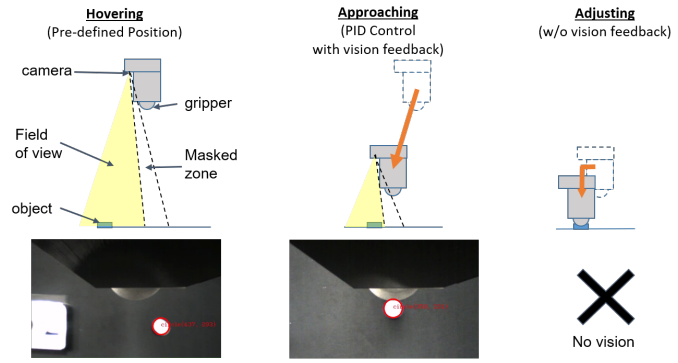

Figure 5. Motion Planning

1) Hovering: The arm moves to the pre-defined position (roughly above the objects) and wait in the air.

2) Searching: When the camera cannot find the object in the picture, Baxter's arm move around in a pre-defined area. Once the camera finds the target object, it will move on to the next step.

3) Approaching: The arm approaches to the object with vision feedback. In our PID controller, the reference value is given as a pixel position and compared with the current pixel position of the object, which is identified by the object detection algorithm in each time step.

4) Adjusting: Even after the approaching step, the gripper cannot reach just above the object. This is because when the gripper approaches closer to the object, the target object disappears from the image due to overlapping with the jamming gripper itself, and thus the vision feedback no longer functions. Therefore, in this step, the gripper moves a predefined distance to reach just above the object.

5) Grasping: The gripper pushes the object to the table and send an activation signal to Arduino to turn on the pump, then wait for 7 seconds to make sure that the air is sufficiently evacuated from the gripper.

6) Retracting: Move back to the hovering position described above (Step.1) Repeat the same motion for the corresponding hole.

For the "Adjusting" step, we repeated trials to calibrate the amount of motion. However, due to the baxter's motion error, we found that the resulting position is slightly different each time. We will discuss this motion error in III-E.

III. Experiment

A. Description

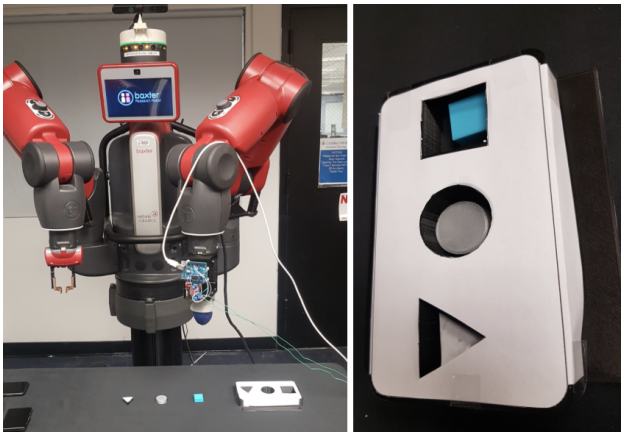

Figure 6. Experiment Environment

In the experiment, the robot recognizes the shape matching the command, picks up the object and carries it to the corresponding hole.

The experimental environment is as follows: Place a triangular, square, or circular object in a row on a desk and a frame with each type of hole next to it. To solve the shadow problem,

the desk was covered with a black cloth, and the color of each object was made in a color contrasting with black.

B. Success Criteria

- The jamming gripper will succeed in grasping 3/3 objects at a success rate of at least 75% each.
- The jamming gripper will succeed in identifying the shape of 2/3 objects and placing them into the corresponding holes at a success rate of 75% each.

Here, grasping success indicates the robot lifts the object vertically. Detecting success is when the robot places the gripper on an object that matches the command. Transporting success indicates the robot moves objects into the corresponding holes.

C. Result

The experiment was implemented 25 times for all objects:

Action	Detect			Grasp			Transport		
Object	△	□	○	△	□	○	△	□	○
Success	100%	100%	100%	88%	72%	92%	84%	52%	88%
PDD	75%	75%	75%	75%	75%	75%	75%	75%	75%

As a result, we successfully performed except the grasping and transporting object for the square (partial success).

D. Result Analysis

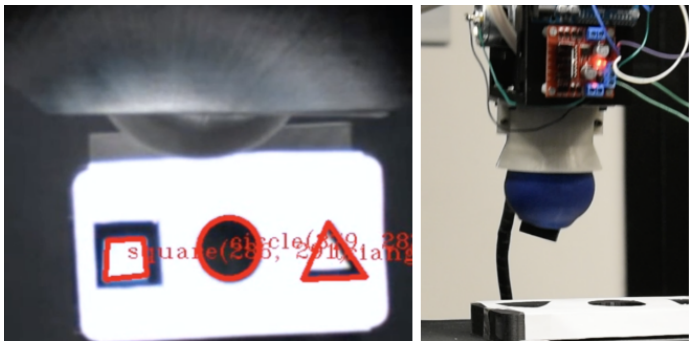

Figure 7. Result Analysis (from left to right: a,b)

In the experimental results, only square case did not meet the criteria. This may be due to the failure of motion planning derived from incorrect pose estimation. As shown in Figure 7(a), the detection of square was not necessarily a square, but a *distorted* rectangle. In that case, it was difficult to estimate the position of object accurately, and it was hard to locate the gripper at the center of the object. On the other hand, almost perfect recognition was achieved for the circle, showing a success rate close to 90%.

E. Unexpected Problems

1) Stickiness: Unexpectedly, due to the stickiness of the balloon, the object was still stuck from the gripper even though the pump was turned off.

Possible Solution: To avoid sticking, the jamming gripper needs to have a blower pump as well as a suction pump. When the gripper releases the object, a blower pump helps the object detach from the sticky rubber by extending the gripper surface.

2) Inaccuracy of Baxter: Because of baxter's inaccuracies, the parameters were modified every time and different results were obtained with the same parameters.

Possible Solution: As mentioned in II-G, Baxter's motion error in the "Adjusting" step cannot be corrected by vision feedback, because the target object disappears from the image. This motion error is considered to be mitigated when the baxter approaches closer to the object with vision feedback, i.e., the amount of the motion in the "adjusting" step is reduced. Thus, the success rate can be improved by two approaches below:

- Re-design more compact outer case for the gripper than the current design so that the masked zone in the image is reduced.
- Develop another algorithms for object detection that can detect the object even if the part of the object is missing due to overlapping.

IV. Conclusion

A. Recap

This project designed a jamming gripper for baxter and implemented an integrated manipulation system using computer vision. In terms of design, several issues have emerged and, in particular, the issue of moment generation has been a good lesson in designing robots. Our object detection algorithms and PID controller individually works well. However, when they are combined with the hardware components we found some problems, i.e. a narrow field of view and the baxter's motion error. To complete our goal, we should have tightly integrated all components in the earlier phase. The mission of the project was to accurately recognize the object and carry it to the corresponding hole. Through 25 experiments, success criteria have been met for all objects except the square.

B. Future Works

1) Global Camera: With the 3D camera installed, it is possible to measure the z-axis value that can not be observed at present, and it is possible to exclude the searching part from motion planning to find an object that deviates from the 2D camera field of view.

2) Simultaneous Control of Blower & Suction Pump: In the current design, there is some restriction on grasping because it uses only suction. If the system is modified to take advantage of the blower function, the sticky problem will be solved, and by injecting a little air into the balloon just before grasping, it is expected to wrap the object more completely and increase the probability of grasping success.

Acknowledgment

We would like to thank Professor Matei Ciocarlie and Teaching Assistant Emily Hannigan for their help in this project.