Collision Avoidance
With
Depth Detection Camera

Winter Quarter Review
# Development Team

<table>
<thead>
<tr>
<th>Name</th>
<th>Responsibilities</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jiaqi Tang</td>
<td>Leader, collision detection</td>
</tr>
<tr>
<td>Ming Wen</td>
<td>STM32, Motor</td>
</tr>
<tr>
<td>Kerr Ding</td>
<td>Color Detection, Communication</td>
</tr>
<tr>
<td>Joe Yu</td>
<td>STM32, Motor, SLAM</td>
</tr>
<tr>
<td>Zhanglu Wang</td>
<td>Color Detection, Communication, 3D Modeling</td>
</tr>
</tbody>
</table>
Goals and General Design

• In short, we need to avoid the robotic arm from collision using a mounted depth sensing camera.
System Design

Depth Detection Camera

USB 3.0

3D printed mount

Step Motors

PWM

Raspberry Pi

Edge detection/object detection

Color detection & arm tracking

Multi-system coordination

STM32 Microcontroller

USART

x 2 or more
3D Printed Mount
3D Printed Mount
Raspberry Pi Software Development

- Using multithread (multiprocessing in python) to get better performance
- Using synchronous lock to synchronize
Color Detection

• Use stickers on the arm to get the precise shape of the whole arm

• Find the color range in HSV space for our stickers

• Find the geometric center of the robotic arm using the color mask
Color Detection Demo
STM32 and Step Motor

ARM tracking:

- Two motors
  - One for up and down
  - One for right and left
- Center boundary in the image frame
  - Whenever excess the boundary
  - Raspberry pi will send signal to STM32
    (microcontroller) to make the frame centered with
    the arm
Arm Tracking Demo
Object Detection

1. Edge detection on a single layer of the RGB frame
Object Detection

2. Find the point from the color detection mask

points to make sure the distance between every detection avoid repeated detection, which will slow down the program.
# Object Detection

3. Find nearby object based on the gap

<table>
<thead>
<tr>
<th>Detection center</th>
<th>Detection Direction</th>
<th>Depth Data</th>
<th>Robotic Arm</th>
<th>Gap</th>
<th>Blocking Object</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>d1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>d2</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>d3</td>
<td></td>
</tr>
</tbody>
</table>

If \( d_2 \gg d_1 \) and \( d_3 \approx d_1 \)

Then, there is a blocking object nearby
Move to Jetson Nano

- A lot 32-bit operations
- Need more powerful chip to run the program
- Jetson has such multi-core chips and more powerful GPU on it
- Run the program much faster
  - Using multi-threads
Progress in this Winter

• Complete core software developing
   - Object detection
   - Arm tracking

• Build the communication system between subsystems
   - Between Raspberry Pi and STM32
   - 3D printed mount

• Debug and improve the performance of the system
Plans for Spring

• Do test on real device in Alcon Lab

• Solve the multi-system coordination (At least 2)

• Try some communication with robotic arm controller
Thanks to

- Yogananda Isukapalli
  - Boning Dong
  - Trenton Rochelle
- Yuepei Hu (Alcon)
Questions