Real Robot Challenge Stage 2

Team: **MiLAB**

Abstract

In the second stage, we solved the task of using a real robot to manipulate the dice to a random target position. Each task is broken down into three stages: select the corresponding starting point and endpoint, move the robot finger to the contact points to the dice, and take the dice to desired positions. In order to locate the specific positions of the dice, we perform motion sampling on the X-axis and Y-axis, and fit the picture and position according to the least-squares method.

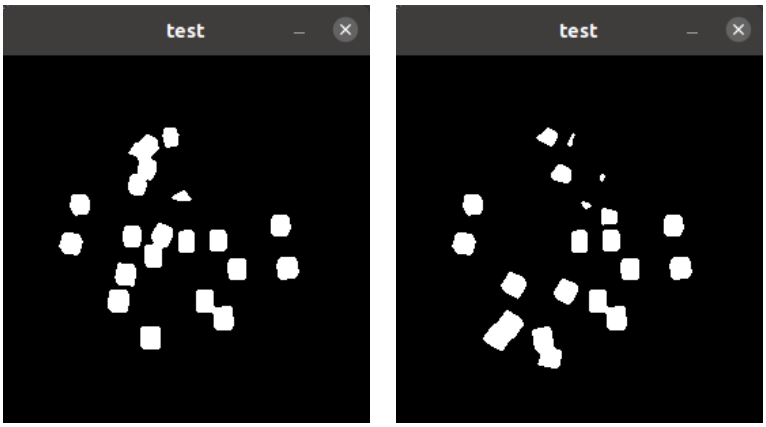
1 Introduction

With the amount of information, we could obtain from the task at this stage, calculating the state of each dice would be difficult taking into account of both the orientation and the coordination. Considering all the dice have the same Z-axis position, we decide to push them to the target location, avoiding the issues that we might have in grasping tasks. We plan to first map the relationship between global coordinate and robot fingers base coordinate. We calculate this coordinate transformation based on the motion of the robot arm and the changes in the mask. Then our goal is to move those dice to their desired position. We set the weight of the grasping position according to the target position and the current state of the dice. Then finish the push based on the relative distance.

2 Method

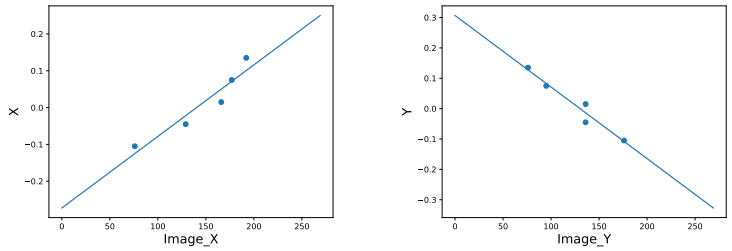
In order to obtain the positions of different pixels, we move the robotic arm in different directions and observe the changes in the mask image as shown in Fig. 1. Sampling the X and Y axes respectively, and finally calculate the corresponding curve according to the sampling results. The results are shown in the Fig. 2.

Facing the randomly located dice in the map, our first task is to design a method to extract the current dice positions, then we push the dice in undesired positions towards their desired positions. Here we need to match two types of masks. One



(a) Before the manipulator moves (b) After the manipulator moves

Figure 1: Comparison before and after the movement of the manipulator



(a) Corresponding function of image mask and real X-axis position (b) Corresponding function of image mask and real Y-axis position

Figure 2: Mask and real position corresponding function

is those located within the area of desired dice position but has no actual dice. The other one is coordinates masked with dice that are not in the desired dice area. As shown in Fig. 3, our goal is to push the gray squares to the white squares. The fewer gray squares there are, the closer we are to the goal. We further divide the map into 15x15 squares and calculate the values of the two masks discrepancy as shown in Fig. 4. Then we set the rules of dice pushing. The robot fingers first move along the X axis to the goal. Then the fingers move above the dice and reach the other side and push the dice along the Y axis till they reach the goal.

Figure 3: The goal mask not reached is the same as the error mask reached. The gray box is the mask at the wrong position, and the white box is the target position not reached

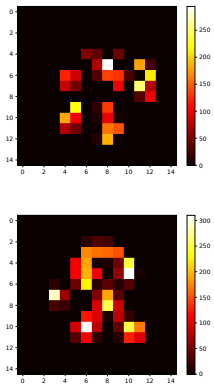


Figure 4: Thermodynamic diagram of non reached goal mask and reached error mask

3 Result and Discussion

We test the effectiveness of our method by applying it in a variety of situations in simulation. our method is compatible with stability and adaptability. According to the simulation results statistic, our method reduces the mask error to 39% from 16 test runs.