Janne Välitalo

Md Aman Khan

MEI-56606 ASSIGNMENT 2: REPORT

Picking parts with machine vision and robot

CONTENTS

VISION PROGRAM 1
ROBOT PROGRAM 3
SOURCES 4

VISION PROGRAM

At first, the program uses simulated acquisition to load the necessary test images. In the real world this is replaced with camera inputs. We use the frontlit images because they allow recognition of the barcode and the letters dictating the side of the objects. These were not visible with backlighting. Images are not resized because time isn't a critical factor in this application. After that the system is calibrated using special calibration images. This is done specify the distances in the real world.

Inspection starts with counting the number of objects in the image. This is done using detect objects –function. It looks for dark objects in the region of interest which is slightly larger than the lit platform. We specified that it looks for objects between 20 pix^2 and 1900 pix^2 to avoid misclassifications due to too small or large objects. If no objects are detected, the program jumps straight into drawing the overlay graphics indicating no objects detected.

If there are objects in the image, the program moves to detecting the side of the objects. It tries to determine whether the objects have side A or B facing upwards towards the camera. This is done with pattern matching: the program searches for both letter A and B in the image and determines the sides of the objects this way. If neither A or B is present in the object, side is classified as "no side". The amount of "no side" -objects is determined by subtracting the amount of A and B found from the total number of objects. This state is also used for calculating the coordinates. The coordinates are fixed at the top edge of the letter indicating the side. This is where we planned the robot would pick up the objects from. The angle of the letter is the same as the angle of the object so it can be used as is. The only difference is that since the robot wants the orientation in range between –180 and 180 degrees, we subtract 180 from all of the coordinate values (since they are in range 0 to 360 degrees). Also, thanks to earlier calibration, the x and y coordinates can be extracted as millimetres rather than pixels.

Next step is to determine whether the objects are shaped L or T. This is done with geometric pattern matching. We use the whole object as the geometric pattern, but the letter is ignored in the template. If the object is neither of those shapes, it's classified as "no shape". However, the barcode and the hole in L shaped objects may be in a different place depending on which side is facing towards the camera. Therefore, we search for both L and mirrored L with different pattern matches. The total number of L shaped objects is then got from the calculator as a sum of both pattern match results. Similarly, the amount of "no shape" objects is the number of all found matches subtracted from total number of objects.

After this there is overlay to show the information about detected objects: total number of objects, the side facing towards the camera, objects shape and objects coordinates. Coordinates also include the angle of the object. Finally, there are placeholder states reserved for communication with the robot and possible modification of variables. These states are currently left empty for the lab. In these states the string of characters that contains the part information and coordinates would be constructed.

ROBOT PROGRAM

SPEL+ is a BASIC-like programming language that runs in the controller which supports multitasking, motion control, I/O control. As the requirement of this language, the provided program contains Function main and this function end with Fend command. This Function main call several subprograms which are as followed:

- Monitors collisions at gripper (line 254)
- Initializes robot by turning on the motor with defined speed, acceleration, etc. (line 222)
- Initialize MV parameters and positions: where we can modify the real drop positions of TA, TB, LA, LB, and Waste. (line 180)
- Sets the upper Z limit for the Jump command.

Communication with camera:
Function MVPickAndPlace contains the command for the camera.

Getting and solving pick point:
In the main loop(line 56), it receives input string from the camera , and retrieve & store the position, orientation and the type of object for pick point.

Solving place point:
In line 196 to 200, the place points are defined. At the time of lab work, we need to change the x and y value to place accurately.

Differences between robot move commands (Go, Jump, Move):

Points	Go instruction	Jump instruction	Move instruction
Cause	point to point motion	a point to point type movement, the robot end effector to first move up to the LimZ value, then in a horizontal direction until it is above the target point, and then finally down to the target point	movement in a straight line
Utilization	when concerned with the orientation of the arm when it arrives on point	for guaranteed object avoidance and to improve cycle times for pick and place motions	to control the path of the robot arm while it is moving

SOURCES