Intro to Robotics II - Final Project Report

Abhinav Jain[1], Ritika Avadhanula[2], Satyam Bhawsinghka[3]

[1]jainab@oregonstate.edu
[2]avadhanr@oregonstate.edu
[3]bhawsins@oregonstate.edu

1 Project Objective

The objective of this project is to develop an autonomous manufacturing process for braille signs. Braille is a tactile writing system for visually impaired humans in which a combination of raised dots represents the literature scheme. They are able to perceive literature written in braille via their sense of touch.

In this project, braille signs are indented on a piece of paper using a stylus attached to the end-effector of the hiwonder arm (viz. a *6*-DOF robotic arm) depicting the autonomous manufacturing process.

2 Setup

For this project, we have been provided by a 6DOF arm, a an RPi Camera which needs to be calibrated to the position of the Arm. The arm runs a flavor of ubuntu on a Raspberry Pi which is used to manipulate the servos of the arm. The paper on which we emboss the braille sits right below the camera and under the arm. as shown in figure 1 and figure 3

The input of the system is a text string which is transformed into braille characters and mapped onto the paper in the workspace. After mapping, the arm's end effector which is the stylus, creates indentations on the surface of paper.

The position of the paper in the workspace is perceived using the camera as seen in the fig[1]. The indentations are then generated on the paper by moving the arm over the visible plane to the corresponding mapped locations. The deformations generated on the opposite plane of the paper are the desired braille markings.

2.1 Camera Setup

Figure 1: An image depicting the camera, the robot setup along with the calibration results.

The camera was calibrated using the custom code given by in the hiwonder's arm software. This provides us with the frame transformation information from the camera frame to the robot's base frame (world coordinate).

2.2 Addition of stylus as end effector

Figure 2: An image depicting the stylus being attached to the arm as the end effector.

A stylus was attached to the end of the fourth servo to act as an end effector as shown in the fig[2] which is used to indent the braille characters on paper. To account for the end point of the stylus in the calculation of joint angles are done through the inbuilt IK of the arm. The length of link L4 was changed in the builtin *ArmMoveIK* script present in *ArmIK* directory.

3 Project Pipeline

3.1 Perception

After calibrating the camera the next step is to generate the locations of the required indentations in the robot base frame to print the braille locations.

The the workspace includes a white rectangular piece of paper with a soft cardboard background as seen in the fig[3]. The cardboard used in the workspace has two advantages, it first helps to identify the paper coordinates perfectly from the camera feedback and secondly, it provides soft background to generate perfect indents with the tool tip as our stylus.

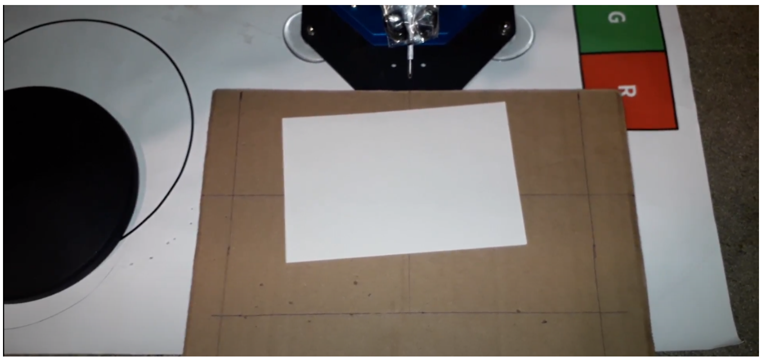

Figure 3: An image depicting the tooltip being attached to the arm's as the end effector.

To get the exact locations of the indents corresponding to the braille characters, the following steps are followed using our perception system,

1. We first get a mask for all the white pixels from our camera feedback.
2. Using them, we generate all the contours in the image.

3. The biggest contour is considered the piece of paper
4. We extract the corners and orientation of the contours in pixel space
5. Convert these pixel space values to real world space.
6. With the corner locations, a margin size of our choice is added to the paper and the start location is obtained.
7. Later, the motion class is used to iteratively obtain all the indent locations corresponding to the braille characters.

. References were taken from pyimagesearch PyImageSearch n.d.

3.2 Braille Text Translation

Braille characters consists of a series of $3 * 2$ matrices arranged in a unique manner. Any alphabets, numeral or punctuation mark correspond to a set of matrices assigned according to ADA committee. In our project we use a open source library called **Braille-Translator** by LazoCoder [LazoCoder n.d.] to generate characters in the form of an encoded in UTF-8
The output generated from the library was transformed in this project into the matrix shown in the fig[4]

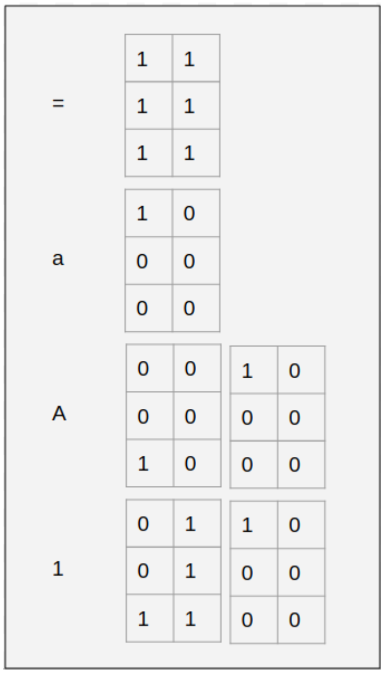

Figure 4: An image depicting an example of text to braille translation. Here dots are represented as 1s and spaces are represented with 0s.

The general convention for all small case letters and punctuation mark consists of just a single $3 * 2$ matrix. Where as a capital case letter or a numeral is represented with two matrices of the size $3 * 2$ as seen in the fig[4].
The 1s and Os obtained from this section are used by the motion class to iteratively produce all the indenting locations in the robot base frame which is considered as our world frame.

3.3 Motion

After obtaining the starting location from where the robot arm can begin the indenting process, for each $3 * 2$ matrix present in the string translation, the following is done.

1. First the positions for each of the six cells of the matrix are calculated.
2. Later, for each cell, if the the cell value equals to 1 then the arm is given a move command to go etch at that location.
   The move command uses the inbuilt inverse kinematics code of the arm to individually indent each dot of the character according to braille rules of that character.

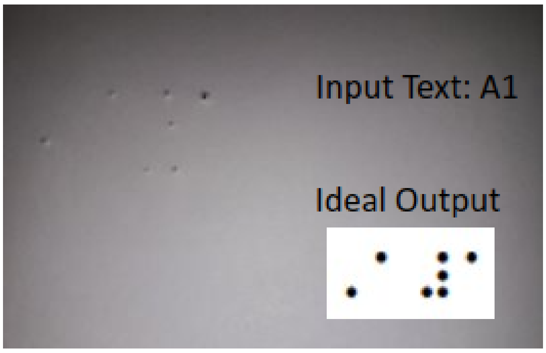

Figure 5: results

3. After looping over all the six cell values, a new starting location is generated, which corresponds to the position of the first cell of the next $3 * 2$ matrix obtained from the translated string.
   The above three steps are repeated until the all the resulting $3*2$ matrices from the string translation have been indented completely.

4 Flowchart

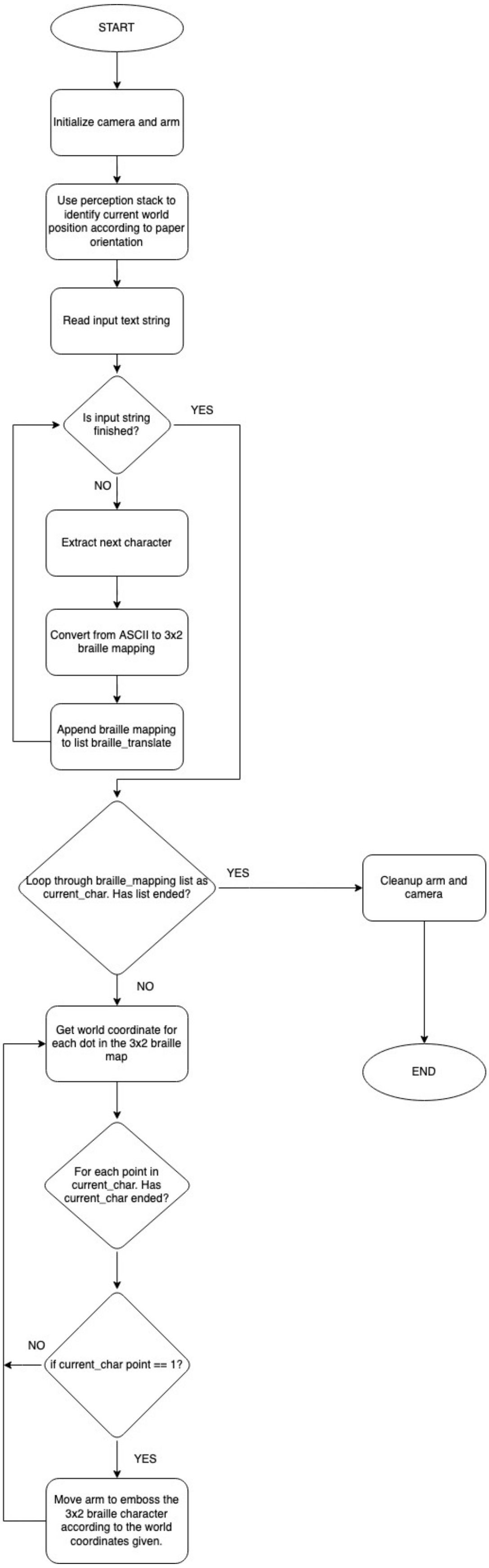

Figure 6: Flowchart showing the flow of high level tasks

5 Difficulties Faced

The robot was made to move with a higher velocity such that the tooltip had enough impulse to pierce through the paper. Even though its worked well for few characters, there the resulting shaking caused by the robot, slightly added noise in the print as the lent of the string increased.

The tooltip had a stylus attached to to by the scotch tape. So after few indentations, the tooltip begins to slide back which lead to improper piercing on the paper. To counteract this error, the tooltip was slid out such that slight sliding back did not effect the piercing profile. But this led to the tooltip end related vibrations in lateral and longitudinal directions resulting in a bad accuracy in positioning the indent

The cardboard base as shown in the fig[3] was also shaking alot initially, we later had to hold it firmly to counteract the impulse transfer from the tooltip action.

6 Results

The results of this project can be seen in Figure 5.Further the videos can be viewed at:

1) Screen Grab
2) Working Arm Demo

As we can observe, the indentation is not close enough and not perfectly aligned. This is because the lower resolution of arm servos. The frame transformations, IK and software development were the easiest part to complete. With the robot are hardware, we came across challenges which proved that our idea is many steps behind from gaining complete autonomy in the field of braille sign manufacturing.

References

LazoCoder (n.d.). *Braille-Translator*. URL: `https://github.com/LazoCoder/Braille-Translator`.

PyImageSearch (n.d.). *Sort Rectangle corners*. URL: `https://pyimagesearch.com/2016/03/21/ordering-coordinates-clockwise-with-python-and-opencv/`.