Title: Spatial map and haptic feedback of textures using a tactile sensor-equipped Soft finger.

Mentor: Prof Nitish Thakor, Johns Hopkins University

Name:
Neel Patel
Mechanical Engineering, IIT Gandhinagar
neel.kirankumar@iitgn.ac.in

Colleague:
Rwik Rana
Mechanical Engineering, IIT Gandhinagar
rwik.rana@iitgn.ac.in

Problem Statement:
The main of the project is to help a finger amputee get a sense of touch using a soft finger. Upon palpation, with various known and unknown textures, the amputee must be able to correctly get an idea of the texture he/she is touching. Using the input textures, we intend to create a spatial map of the texture and also give haptic feedback to the subject.

APPROACH:
In this project, we intend to:
Make a spatial map from given textures.
Convert the textures to haptic sensation.
To achieve both the outcomes, we follow the same common steps initially.
The process would include collecting data from the soft finger. This step involves making a set up analogous to a 2D/3D-plotter, which could palpate over the whole texture. We intend to use the most readily available 2D/3D plotter i.e. the 3D printer. A custom mount would be made over the nozzle of the 3D printer, thus helping the soft finger (which contains the tactile sensors) to move over the texture.
Using neuromorphic computing and spiking neural net- works (SNNs) to classify the textures using the input data from the tactile sensors.
Along with the SNNs, one of our objective is to visualize texture which are not known. For this, we intend to use Variational Autoencoders (VAE).
The intermediate weights i.e. the hidden features can be used to get useful information about the spatial mapping of the system. Moreover, as we would be palpating over the texture continuously, we will use LSTMs to capture the temporal aspect for the same. This spatial and temporal data combined will help us give the spatial texture map.

To convert the neuromorphic encodings to haptics, we intend to use the weights and hidden features as an input to the haptic system. The haptic system consists of a cluster of small vibration motors, which would rest on the wrist of the subject. So, on suitable input to the vibration motors, one could feel the sensation of the particular texture.

Things Done Till Now:

Gcode: Our first aim was to collect the palpation data on different textures. For that, we need to move our tactile sensor-equipped finger over all the 2d points of textures and collect the palpating data. We decided to use the 3D printer for the data collection part. We made our custom g-code and passed it to the 3D printer, the finger moved as desired motion. We made gcode by considering various parameters like offset, speed of palpation, etc. Our g-code was robust, precise and easy to understand. It will help us in collecting accurate palpation data.

3D printer and design:

We made an attachment that was attached to the 3D printer for the data collection part. We designed the attachment on the extruder of the 3D printer which was carrying the finger with a little suspension system in it for the degree of freedom of the finger. We 3D printed the attachment for the finger and different textures namely, 3 Bumps, 6 Bumps, 4 Ridges, 6 Ridges, 4 Waves, 6 Waves, 4x4 Blobs, 6x6 Blobs, Plain surface. A bucket for carrying a tactile board was also a part of the attachment so that it assisted us in data collection. The attachment was lightweight and hence it doesn’t harm the printer and it also didn’t affect the motion of the printer. We purchased the 3d printer magnetic bed and attached the holder for carrying the texture. We just had to put one texture in the holder and after the data collection part was done, we replaced it with another texture.

Overall circuit integration

Finally, we integrated the 3D printer, PC and tactile board for collecting the data. When the 3d printer starts the motion, it gives feedback about it and after receiving the feedback the tactile board gets on. We used the python library serial for the feedback part. So, the tactile board and 3D printer communicated with each other via PC.

Data collection

Once the overall setup was made, we moved on to the data collection part. In this step, we had to precisely calibrate and tune the finger's movement over various textures, i.e. ensure that x,y,z are aligned perfectly. A total of 9 specimens were used. The first iteration of the overall data collection led to us noticing that collecting the time data for the temporal part of the algorithm was not accurate, given there were small movements of the 3D printer that a standard stopwatch and human eye cannot gauge. Hence, this led to us collecting the 2D data using a series of 1D movements. Palpation was done twice on each specimen. A compact python package has been made to streamline the whole process of data collection.

Data processing

The tactile sensor is made of 16 small sensors arranged in a 4x4 manner. So the next step was to resolve the voltage data collected into each of the 16 sensors. Moreover, given that we knew the speed of the 3D printer and the total time of each pass, we could calculate the instant at which each data was collected. Our main aim is to make a spatial map from the collected data. Hence the next step was to get the corresponding x,y,z coordinates of the voltage data collected. Hence, a python package is made, giving the z value of the texture given x and y.

For every pass, the x value is kept constant, and the movement is done in the y-direction. This leads to us only estimating the y value given velocity and time. Thus we have made concise data that gives x,y,z, voltage and time for each sensor for every iteration of every specimen.

Data visualisation

Lastly, we combined the x,y, and z values of all the 16 sensors for each specimen's iteration. This will be further useful while making the Deep learning architecture.

Observations:

The data looks very similar to the human eye. 4 Waves and 4 Ridges seem very similar at first. However, a closer look gives us an idea that as 4 ridges are very sharp, the sensors relay a sudden voltage drop of large magnitude on the first ridge, and thus the successive ridges also have high drops compared to the ridges. Thus, we can say that many underlying features are not visible to the human eye, especially the temporal features. This is where we will take help from deep learning and, more precisely, VAE to get the underlying latent features into a simple array and thus will help differentiate and extract features for even unknown data.