TACTILE VISION1.0 // SEEING THE WORLD THROUGH VIBRATIONS

EE376A (Winter 2019)

By Abhipray Sahoo & Sakshi Namdeo

Motivation

Human eyes are incredible sense organs. They can capture images with great detail and the vision system as a whole can perceive those images to extract complex pieces of information. We set out to capture the information in images and encode them as patterns of vibrations on the human skin. The goal is to enable a visually impaired person to perceive those vibration patterns as images.

Introduction

Back in the 1960s, Paul Bach-y Rita conducted some of the first experiments in translating images from a camera feed to vibrations on the skin. He installed a grid of 400 vibrating plates on a chair that displayed images coming from a camera feed. He had blind people train with it and interpret those vibrations as representing objects in front of the camera. Based on their responses, he believed those vibrations were being processed by the visual cortex, the part of the brain associated with perceiving images. This indicated that the brain is plastic enough so that in the blind, the new vibration stimuli was perceived as vision.

For our project, we focused on encoding single images (as opposed to a video feed) as time-varying vibration patterns. We used a chest strap called the Link built by NeoSensory. It has eight vibrating actuators that we can control wirelessly.

We experimented with several encoding schemes to map images onto vibration patterns which we will describe in this post but first, here's what elementary school kids thought of our final choice of the encoding scheme.

Outreach Event

Our project stall at the event and pictured in it – Daredevil our mascot

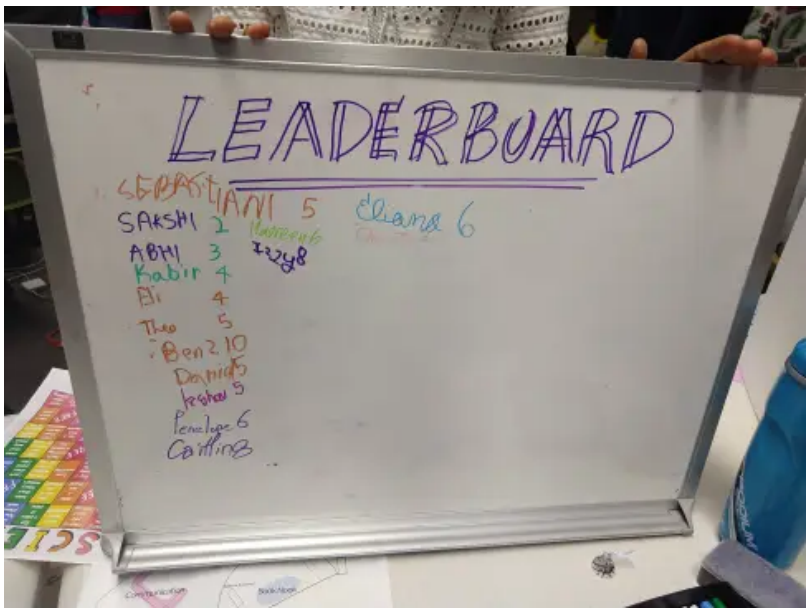
Over the last weekend, we presented our project to kids from Nixon Elementary School. As part of the event, we prepared a technical demonstration of the project prototype for the students – to wear the Link and play a game designed to test how well they could decode vibrations produced by the Link. Before they played the game, they trained up on six vibration patterns representing six different images – of a dog and a man doing one of the three different things: eating, sleeping or running. The kids wore the Link around their chest and the Link "told them" what the image was. For example, an image of a dog running on the beach produced a pattern corresponding to "A do-g i-s run-ning on the bea-ch". The objective of the game was to identify the subject and the action being performed by the subject in a particular image based only on the vibrations on the Link.

Picture of a dog running on a beach
PC – Wikipedia

During the test, the kids were shown 6 new images of combinations of the subject performing different activities and were scored for correctly identifying the two categories with 12 being the maximum points. The kids might have initially been lured with the prospect of having their names on the leaderboard and of course by the candy at our table, but once they started the challenge, they couldn't stop. One student, Ile came back a second time to try his luck! Ben was a curious and a rather serious kid, who took the game controls in his hands and navigated through the game-app on his own, scoring 10/12.

A kid wearing the Link and playing the game

He later told us his cheat code; he focused on the location of vibrations more towards the right for “man” and slightly to the left and the middle for “dog” or how each sentence started with a distinct heavy vibration on one side corresponding to “A” or eɪ .



The Leaderboard TactileVision1.0 3/16/19

This kind of reasoning was given by a few other kids, leading us to think of the demonstration/game as a good proof of concept for the viability of the proposed technology.

In total, we had 15 participants who played the game on the Link and scored an average score of 6.8/12 or 57%, all this with a basic training on 6 images shown within a one minute timeframe.

Implementation Details

Encoding Scheme Design

In designing an encoder, we used the following general guidelines:

- The encoding should be easy to learn or alternatively there needs to be an explicit training method to teach users how to make sense of the encoding
- There exists a decoder that can reconstruct the conveyed information with minimum distortion
- Users should be able to distinguish classes of objects (eg. dog vs human) in the image and what action is being performed (eg. running, eating, sleeping)

Encoding with language

The encoding scheme that the elementary school kids tried out relies on English speech comprehension. We transform an image into a vibration pattern with the following steps:

1. **Image to Text**: Using an image captioning deep learning model called im2txt trained on a large image dataset, we generated captions for our selected images.
2. **Text to phonemes**: The text output of im2txt is converted to phonemes. Phonemes are the building blocks of speech; there are 44 of them in the English language each representing a speech sound. The primary reason for doing this conversion is that there are fewer phonemes needed than letters per word. Eg. "A man is running" -> [ax, m, ae, n, pause, ih, z, pause, r, ah, n, ax, ng]. We used a python library called phonemizer to do this.
3. **Phonemes to vibration frames**: We convert each phoneme into a vibration frame to be played out for a fixed duration of 80ms. To build the mapping itself, we first found 44 distinguishable vibration patterns experimentally and rank-ordered them by how easy they were to distinguish (i.e patterns with low human decoding confusion error). Second, we listed the phonemes in the order of frequency of occurrence in English speech. Then we associated the most frequent phoneme symbols with the most distinct vibration patterns following their rank-ordering. To see a more sophisticated approach, see the last section of the blogpost (modeling the human as a noisy channel).

We think this is a practical approach to the problem because it captures in words, what is going on in an image. Can a human reconstruct the original image from these words? With perhaps enough descriptive words as demonstrated by work done by Tsachy's group: "Humans are still the best lossy image compressors". There, a human was asked to reconstruct an image based solely on the image description and textual directions communicated by another human. They found some of the reconstructions to outperform some of the state of the art results in lossy image compression.

Other encoding schemes

We experimented with several other schemes. While the previous encoding scheme relied on language, in the other schemes we tried encoding the low-level features that make up an image.

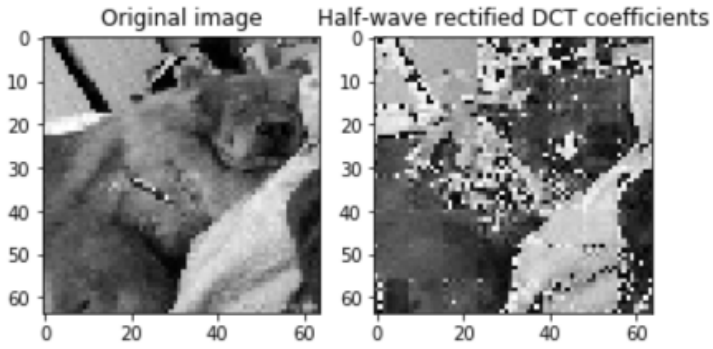
JPEG-like Discrete Cosine Transform

JPEG is a popular lossy image compression format. At its heart is the discrete cosine transform which is a linear transformation of the image data. It changes the basis of the image to a set of cosines with different frequencies. After the DCT transformation, most of the energy is concentrated in a few coefficients usually representing the low frequency cosine components. We can then keep the top K coefficients to represent the entire image and zero out the rest. This works well if we take small blocks of the image at a time. We experimented with 8×8 image blocks, taking their DCT and keeping the top eight lowest frequency coefficients. Thus, for a 64×64 pixel image, we have 64 8×8 blocks. Then we use the Hilbert space-filling curve to arrange each encoded 8×8 block across time i.e we get 64 8-dimensional vibration frames to be played out on the Link (for the motivation behind the Hilbert curve, check out this video on the subject). If each vibration frame is played out for 32ms, each image has a vibration pattern that lasts roughly two seconds.

Another DCT based-approach we tried emphasized spatial information. We took a grayscale image, and split it into two halves vertically — the left side was to be represented by the left four motors of the Link and the right side was to be represented by the right four motors. Then we broke up the image horizontally into overlapping frames and did a DCT based encoding. For example, for an image of 100×100 pixels and 50×50 sized horizontally overlapping blocks, we get three 8-dimensional motor encoding frames to be played out one by one over time. The second frame in this example is redundant and within each frame, the left motors only represent the left side of the image and similarly for the right

four motors. The idea behind this scheme was to encode some notion of where the objects are in the image as actual spatial location on the Link.

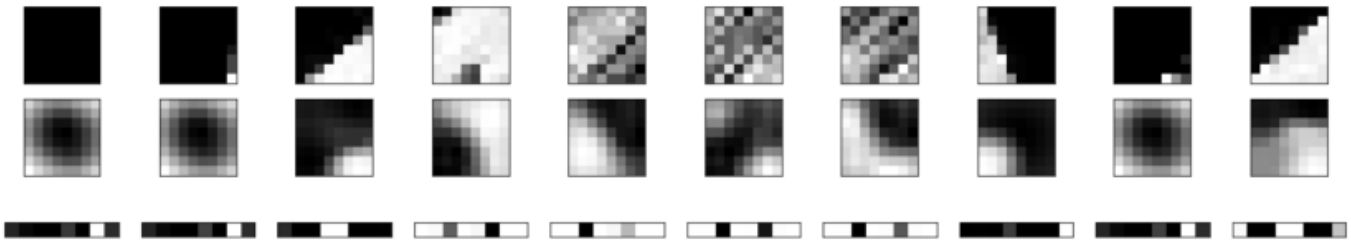
The non-negativity problem: The motor encoding is inherently non-negative since the motors can vibrate only at fixed non-negative intensity levels. One solution to this problem is to zero out negative DCT coefficient values. This leads to large reconstruction error. Another approach is to simply shift up the DCT coefficients so that they are never negative but this leads to an encoding that is not-so-easy to interpret — zero (black image) does not map to zero (no vibrations).



To solve the non-negativity problem, we experimented with the idea of using data and machine learning to come up with an alternate to the DCT which has similar compression properties but leads to non-negative coefficients. This was based on good results in lossy compression using a method called projective non-negative matrix factorization (pnmf) in this paper. To test it for ourselves, we designed an auto-encoder to achieve the same effect as pnmf.

Non-negative linear auto-encoder

We designed an auto-encoder that transforms each 8×8 image block (64 dimensional) into an 8 dimensional embedding vector using a linear transformation matrix W. The decoder part of the auto-encoder is simply W^T . We enforced non-negativity in the weights and the embedding vector. Using stochastic gradient descent we optimized the weights for W on reconstruction mean squared error on a small image dataset. The results were mixed:



Top row: Original 8×8 image blocks
Middle row: Decoded/reconstructed image
Last row: 8-dimensional motor intensity embedding vector

The auto-encoder approach is a promising avenue for designing non-negative embeddings. With perhaps a larger training dataset and a more expressive non-linear machine learning model (eg. WaveOne), we could design good data-driven encoding schemes.

Modeling human perception as a noisy channel

In our final encoding scheme that used language we had to find a good mapping for each of the 44 phonemes to a vibration pattern. We revisited this problem after the outreach event and tried to solve it

more algorithmically.

The problem can be defined as finding an encoder that assigns to every source output symbol (phoneme) one channel input symbol (vibration). Here, our channel is the human skin+perception. Our vibration frame for every phoneme is defined as an 8-dimensional vector (for 8 motors) with each value indicating vibration strength. If we are to assume a noiseless channel i.e a human has perfect decoding capability for every possible vibration pattern, then we have no problem — we can map a phoneme to any arbitrary 8-D vector. We can simplify our solution by placing constraints on the 8-D vector. Two simple constraints are (a) imposing a sparsity of 2 so that only 2 motors are on at a time (b) each motor can either be on or off. With these constraints, we only get $\binom{8}{1} + \binom{8}{2} = 34$ non-zero possibilities so that's not enough for 44 phoneme symbols. What if we allowed three motor states: 0 for OFF, 1 for half intensity, 2 for full intensity? Now, we have $\binom{8}{1}2 + \binom{8}{2}2^2 = 128$ possibilities. Thus, a good start might be to use a sparsity of two and, three states for each motor. Now we can choose any 44 of the 128 possibilities if the human channel is noiseless.

If you have ever tried decoding vibration patterns before you will know that it's not easy; the human skin is a very noisy channel especially without training. Are 128 possibilities really enough for a noisy channel? A noisy channel is characterized by the conditional probability of the output given the input $P(Y = y|X = x)$ which captures for each input x, how likely a human is to decode it as y. Through actual testing on humans, we can perhaps come up with these conditional probabilities for all combinations (128×128) but that is tedious. Maybe we can design a simple model for the human decoding confusion graph. What could be some safe assumptions regarding the expected human decoding error?

- There is a high probability of decoding error between each consecutive motor state eg. half intensity (state 1) and full intensity (state 2).
- There is a high probability of decoding error between neighboring motors. An active motor some distance away from another active motor leads to a higher probability of error than one further away.

With these two assumptions regarding decoding error probabilities, our next step is designing a good distance metric between the 128 8-D vectors that captures the error probabilities. We used a ternary Gray code for generating the 128 possibilities in an order which constraints consecutive codes from having a hamming distance of 1. The Gray codes look like:

```
'00000001': 0
'00000002': 1
'00000012': 2
'00000011': 3
'00000010': 4
'00000020': 5
'00000021': 6
'00000022': 7
'00000120': 8
'00000110': 9
'00000102': 10
.
.
.
'01000010': 81
'01000002': 82
'01000001': 83
```

```
'01000000': 84
'02000000': 85
'02000001': 86
'02000002': 87
'02000010': 88
.
.
.
'20100000': 124
'20200000': 125
'21000000': 126
'22000000': 127
```

The codes above have 8 digits each representing a motor and as before, {0, 1, 2} represent the motor state. We removed the zero encoding with the assumption that a user can decode the no-vibrations state reliably. We also filtered out all the Gray codes that have more than 2 motors on at a time. You'll notice that this ordering captures distances between encodings well. For example, the first state '00000001' has one motor ON at half-intensity; the last state is '22000000' which have **2** motors farthest away in spatial distance and at maximum intensity. The codes we generated used the algorithm in this paper by Guan. It has the nice property that the count of the motor state alternately rises and falls as opposed to cycling through.

Our goal is to assign to the most frequently occurring phoneme in English speech to the most distinguishable vibration. To find a rank-ordered list of distinguishable vibration encodings, we can start with the first gray code, find the one farthest away from that one, then the gray code most equidistant from both the first and second encodings and so on. This translates to clustering the 128 possibilities into 44 uniformly spaced intervals but ordering them by the above procedure.

Results and Conclusions

In the game played with the elementary school students we observed scores better than random chance in identifying the underlying image through only vibrations on the chest. The proposed design can be further explored to achieve better performance with better training, incentives and encoding schemes. The test results indicate that good encoding of information from a conventional domain (images seen through eyes) to another sensory modality (tactile signals felt on skin) is possible and useful; it a promising step towards an alternative assistive tech empowering the visually impaired.

Acknowledgments

We would like to extend our gratitude to our project mentor Prof. Tsachy Weissman for his invaluable support and words of encouragement to keep pushing and challenging the limits. We would further like to thank Shubham and Kedar, and the entire team of EE376A for their assistance and commitment to make it one of the best taught courses at Stanford. The outreach event was an exciting and thrilling addition, which made our entire experience all the more memorable. 🙂

LEAVE A REPLY

Enter your comment here...

Fault Tolerant Quantum Computational Models from Topological Quantum Field Theories

Tabular Compression Using Feature Dependencies