Rachel Walter

CMSC388E

Rachel_Walter_Project4.zip

Overview of the Piece

This semester I have been thinking a lot about movement: joining a Latin dance club and learning salsa, watching my mother struggle with hip problems, trying to work out after being sick all winter break. When people think of art, they are quick to list visual art and singing or playing music. Yet, dance and movement seem like an after though to the arts. I thought using movement as a medium of user interaction would create a unique kinetic piece and allow users a new way of interacting and thinking of art.

Message/Themes

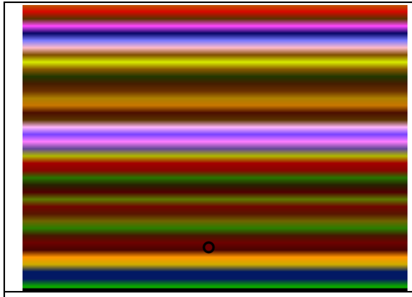
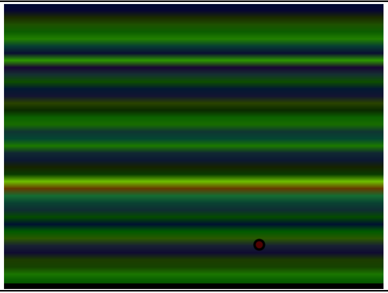
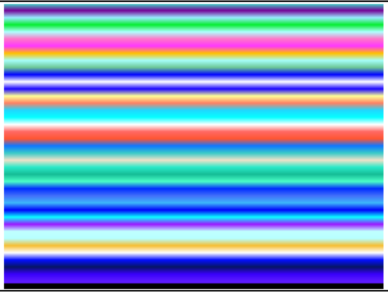
As stated above, dance and movement seem like an after though of arts. However, our movement is as unique as our finger prints. It is an affordable form of expression that doesn't need extra training or art supplies or musical instruments (though fancy shoes are optional). There are international styles and infinite ways to dance.

One goal of this piece is to show the unique nature of every individual's movement. Depending on the direction, intensity, and timing of someone's movement, an art piece of colors blending into each other will be created (red value of color determined by X-axis movement, green by Y-axis movement, and blue by the brightness). The likelihood that anyone will move in the exact say way at the exact same timing is incredibly unlikely; this generates all movement into a completely unique piece of art.

The second goal of the project is to manifest movement as a visual art form. By changing physical movement to visual representation, we are forming a new way of seeing it. It will be interesting to experiment with how changing physical movement to numbers and data might change the emotional quality of the piece. In Laban Movement Analysis, an analytic practice done in theatre and dance, movements are described along four axes: body, effort, shape, and space. Together, these qualities form certain categories of movement labelled names like “Float” or “Press.” This way of categorizing movement allows in more emotional connotation, but how could you express an entire dance piece in just combinations of Laban labels? This project will make dance and movement a more palatable visual art version, but it will not be as full of an artistic experience as watching the dance itself. The beauty if the visual piece along with the realization of detail lost in translation will hopefully reveal the power of movement and dance as an art form.

Iterative Process

1. Original Design (Project 3 Submission)

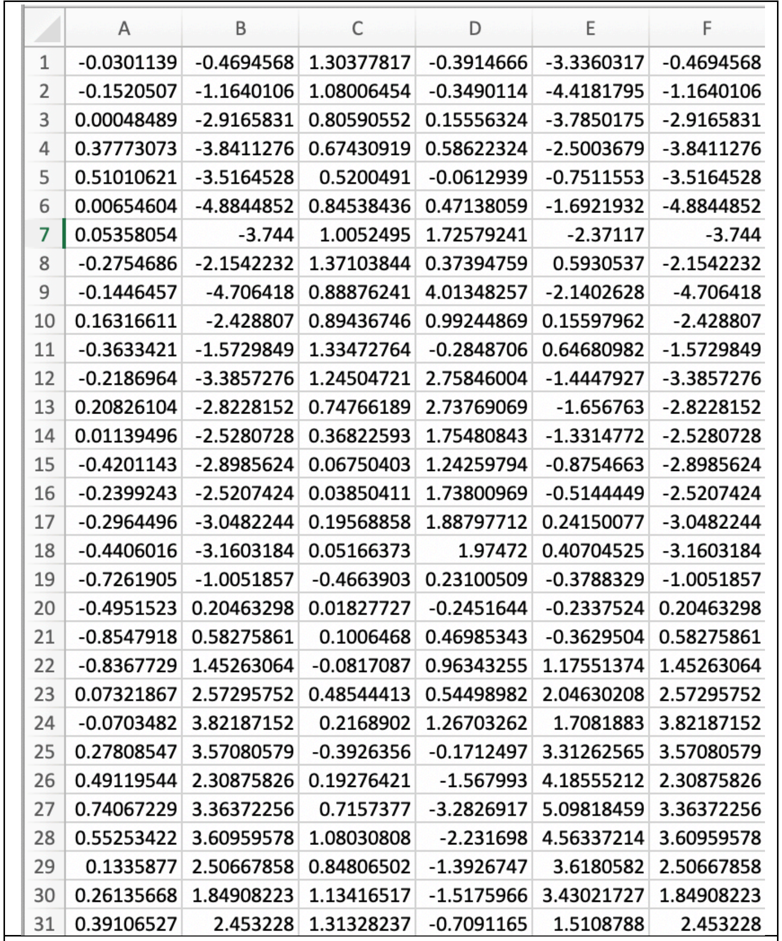
		
Left/Right Movement (Higher Red Values)	Up/Down Movement (Higher Green Values)	Brightness (Higher Blue Values)

Capturing movement in hard! In my first iteration of the piece I made use of color tracking as motion detection. After a user selects a color to track, the program calculates which pixel on the screen at the time is closest to the pre-selected color. The program then tracks how the X, Y, and brightness values of the new pixel compare to the last position it was located at. The magnitude of X, Y, and brightness translate to the Red, Green, and Blue values of the color on the screen. This method for tracking for motion had a lot of issues, however. One of the biggest was that because it tracked a color using a video feed, it was very finnicky. The room being dark, something in the background of the same color, or any other imperfect condition made it so that the motion detection did not work. My other big problem was that since the motion detection was being done on via video (a 2D space), it was difficult to represent a Z-axis of movement. The closest approximation I could reach was brightness, but this did not contribute much to the piece since a specific shade/brightness of color was being tracked.

2. Using Hardware

My solution to the downfalls of color tracking via video was to make use of hardware. I chose to use a Raspberry Pi 3 Model B+ with a Sense HAT with Gyroscope and Accelerometer sensor capabilities. To make the setup portable (and easy to move and dance with!) I made use of a portable battery that was originally an on-the-go phone charger. The sensors directly capture data on X, Y, and Z movement and rotation. The Sense HAT also has a convenient Python API to interface with, which made capturing data and getting it to Processing much easier.

I had 2 options for data transfer: send data in packets over a WiFi network or writing the data to a file on the Pi which would later be transferred to my laptop with the Processing code to make it art. Because I wanted a minimal viable product with the sake of timing for the project and because I mainly interface with the Pi over SSH, it made most sense to me to just write it to a file and use the Unix scp command to get it local for Processing.



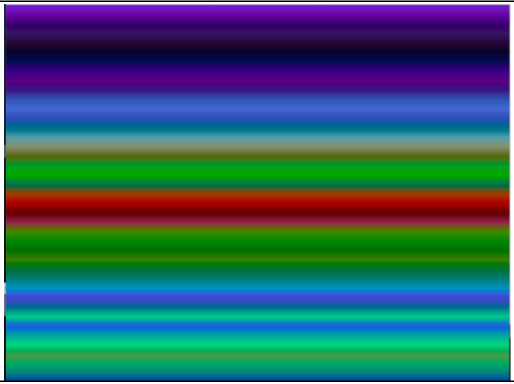
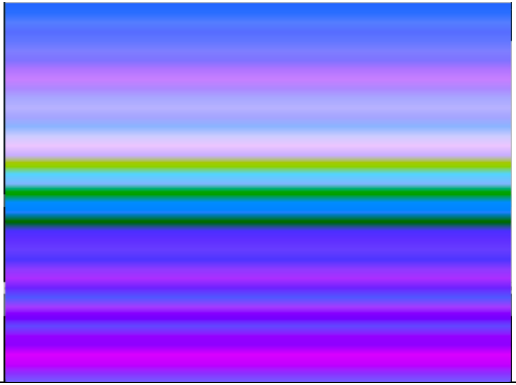
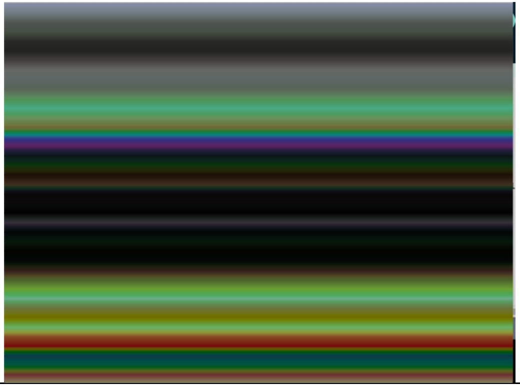
	A	B	C	D	E	F
1	-0.0301139	-0.4694568	1.30377817	-0.3914666	-3.3360317	-0.4694568
2	-0.1520507	-1.1640106	1.08006454	-0.3490114	-4.4181795	-1.1640106
3	0.00048489	-2.9165831	0.80590552	0.15556324	-3.7850175	-2.9165831
4	0.37773073	-3.8411276	0.67430919	0.58622324	-2.5003679	-3.8411276
5	0.51010621	-3.5164528	0.5200491	-0.0612939	-0.7511553	-3.5164528
6	0.00654604	-4.8844852	0.84538436	0.47138059	-1.6921932	-4.8844852
7	0.05358054	-3.744	1.0052495	1.72579241	-2.37117	-3.744
8	-0.2754686	-2.1542232	1.37103844	0.37394759	0.5930537	-2.1542232
9	-0.1446457	-4.706418	0.88876241	4.01348257	-2.1402628	-4.706418
10	0.16316611	-2.428807	0.89436746	0.99244869	0.15597962	-2.428807
11	-0.3633421	-1.5729849	1.33472764	-0.2848706	0.64680982	-1.5729849
12	-0.2186964	-3.3857276	1.24504721	2.75846004	-1.4447927	-3.3857276
13	0.20826104	-2.8228152	0.74766189	2.73769069	-1.656763	-2.8228152
14	0.01139496	-2.5280728	0.36822593	1.75480843	-1.3314772	-2.5280728
15	-0.4201143	-2.8985624	0.06750403	1.24259794	-0.8754663	-2.8985624
16	-0.2399243	-2.5207424	0.03850411	1.73800969	-0.5144449	-2.5207424
17	-0.2964496	-3.0482244	0.19568858	1.88797712	0.24150077	-3.0482244
18	-0.4406016	-3.1603184	0.05166373	1.97472	0.40704525	-3.1603184
19	-0.7261905	-1.0051857	-0.4663903	0.23100509	-0.3788329	-1.0051857
20	-0.4951523	0.20463298	0.01827727	-0.2451644	-0.2337524	0.20463298
21	-0.8547918	0.58275861	0.1006468	0.46985343	-0.3629504	0.58275861
22	-0.8367729	1.45263064	-0.0817087	0.96343255	1.17551374	1.45263064
23	0.07321867	2.57295752	0.48544413	0.54498982	2.04630208	2.57295752
24	-0.0703482	3.82187152	0.2168902	1.26703262	1.7081883	3.82187152
25	0.27808547	3.57080579	-0.3926356	-0.1712497	3.31262565	3.57080579
26	0.49119544	2.30875826	0.19276421	-1.567993	4.18555212	2.30875826
27	0.74067229	3.36372256	0.7157377	-3.2826917	5.09818459	3.36372256
28	0.55253422	3.60959578	1.08030808	-2.231698	4.56337214	3.60959578
29	0.1335877	2.50667858	0.84806502	-1.3926747	3.6180582	2.50667858
30	0.26135668	1.84908223	1.13416517	-1.5175966	3.43021727	1.84908223
31	0.39106527	2.453228	1.31328237	-0.7091165	1.5108788	2.453228

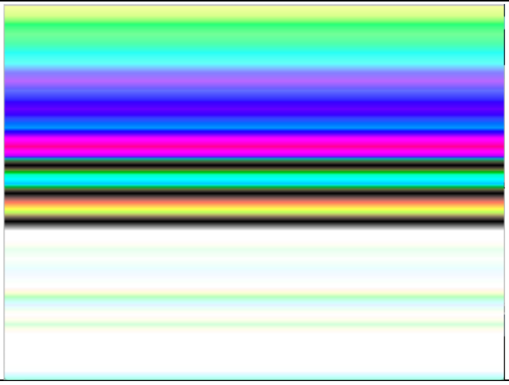
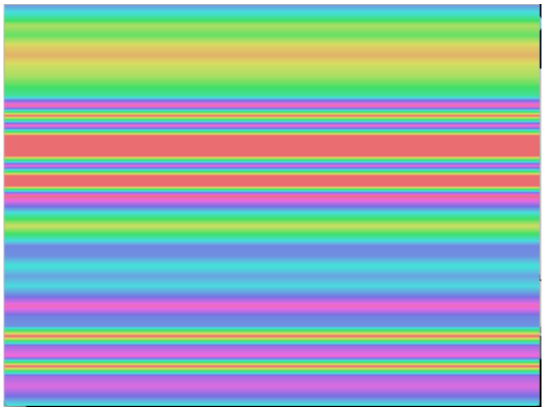
Example of Data Capture Converted to CSV

3. New User Interaction Methodology
   a. SSH into the Raspberry Pi and run the `acceldata.py` program
   b. Type what you want to CSV file generated to be called
   c. Dance and move your heart out! I tried captures both holding the Pi and keeping it in my pockets
   d. Press the joystick button on the SenseHAT to stop the recording at any time
   e. SCP the file back onto a local machine and see the art that you created!

4. Color Experimentation

My first draft with hardware used the same scheme of color creation by having X, Y, and Z values represent R, G, and B values respectively. This created much prettier and consistent results than the motion tracking, but I was curious as to how I could play around with color. There were many ways to map the movement values into a color scheme. I experimented with a few and compared how they looked on the same dataset. Below are some ways I changed color settings:

Method	Image Example	Explanation
Absolute Value RGB		This is the methodology I had been using in the first iteration of the project. In this style, we find the absolute value of the X, Y, and Z and map that onto a value from 0 through 255 to represent R, G, and B respectively
RGB with Direction		When taking the absolute value, we know the intensity of the movement but not the directionality. In the second RGB version, and 0 value is mapped to 127.5 and left/down/away or otherwise negative movements approach zero, while right/up/toward or otherwise positive movements approach 255
HSB Translation (Absolute Value)		I also remembered working with HSB color and thought it would be interesting to play around with that. This is similar to the original RGB model where X maps to Hue, Y to Saturation, and Z to Brightness

HSB Translation with Direction		This is similar to the second version of the RGB model where X maps to Hue, Y to Saturation, and Z to Brightness, but each is adjusted so 0 is the center point of values.
HSB Axis		The math behind this one is very sketchy but has interesting results. For this prototype, I divided the 360-degree HSB color wheel into 6 quadrants and had each diving line act like an axis for X, Y, or Z. For example, high positive z values map to 0/360 the red region of the wheel, so if movement was only positive Z it would show up as red. I also preset the brightness and saturation to values that didn't look like a weird heat map.

After much experimentation, I came to the conclusion that the original color choice of translating to absolute RGB values seemed the most organic and looked the best at least with my original test data set. I did a few more captures tested with the various color mappings and reached the same conclusion. It was fun to be able to experiment and think critically about color and how to "data-tize" movement. This way makes it clear there is an intense movement within one of the axes, which is useful for seeing the emotional and physical intensity.

5. Future Work

I have bulleted some potential areas of future work:

- Make use of Pi networking capabilities and live feed data into the Processing program, rather than relying on CSV files and the scp command

- Make a case for holding the Pi and portable battery to make capturing movement easier for the user and safer for the hardware ☺
- Allow for CSV file selection rather than it being hardcoded into the Processing file (I was having issues with the Processing API command for file picking)
- Not have to use UMD WiFi which is tough to connect to with a device like the Pi. I am hoping to find a router I can use for the presentation, but I don't know if it will work ☹

Example Pieces

You can view sample pieces via this link:
https://drive.google.com/open?id=1h45Oms38uVvkJriLnZx-S8HidMudsHiT

This includes my "I'm alone in my home/apartment" happy dance, the macarena, and a pretty bad attempt at the In My Feelings challenge!

Sources

Coding

- Gradient Tutorial: https://processing.org/examples/lineargradient.html
- SenseHat Documentation: https://pythonhosted.org/sense-hat/api/#imu-sensor
- Many, MANY videos and information pages on setting up and working with Raspberry Pis

Art Inspiration

- *Coronation* (1979) by Leon Berkowitz (https://www.nga.gov/collection/art-object-page.177135.html)
- Laban Movement Analysis (https://en.wikipedia.org/wiki/Laban_movement_analysis)