LIGHTWEIGHT ASSISTIVE TECHNOLOGY: A WEARABLE, OPTICAL-FIBER GESTURE RECOGNITION SYSTEM

Sanjay Seshan
Fox Chapel Area High School

OBJECTIVES

- 18 years ago, a movie called Minority Report was first to introduce the world to a gesture-based interactive device.
- Though common in science fiction and comic book movies, it is yet to become reality.
- The goal of this project it to build a lightweight, low powered method to interact with computers around us
- The computer will recognize a wide range of hand gestures
- Does not require the use or special gloves or other intrusive hardware

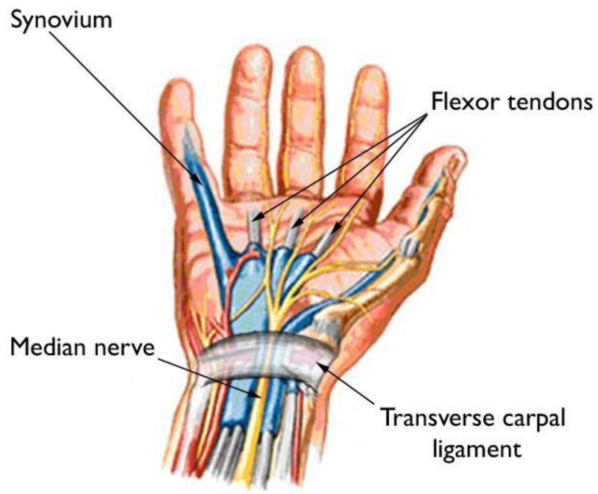
CURRENT METHODS VS. REQUIREMENTS

- Current methods are
  - Intrusive and cumbersome to wear (unnatural positions) – Tap, Cyber Glove, and other devices are uncomfortable and hard to put on and take off
  - Impractical (e.g. camera based) – camera may not face the wearer, Google radar system is also off-body
  - Prices range from $150 to $30,000

- Must be wearable
- Must be non-intrusive
- Must be intuitive
- Must be lightweight
- Must be low-powered
- Must be inexpensive

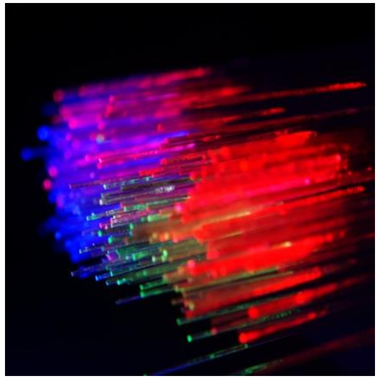
SOLUTION

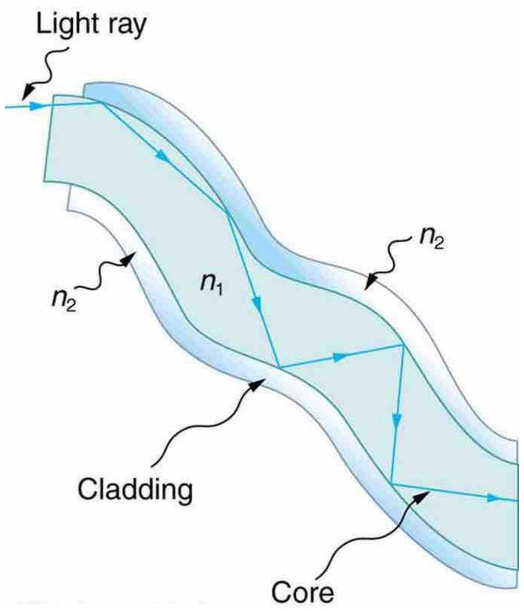
- Optical Fiber Based Lightweight Assistive Technology
- Strap around the wrist
  - 1 in 6 adults own a smart watch
- Use optical fibers as sensors along the strap to detect deflection
- Transverse carpal ligament area
  - Deflection occurs when fingers [tendons] are moved
  - Magnitude of deflection of different tendons can be used to identify gestures
  - 6 different tendons in wrist → each corresponds to a finger
- How to measure deflection accurately?
  - Must measure all tendons and measure tiny (< 1 mm) movement
  - Deploy a set of sensors across a watch strap

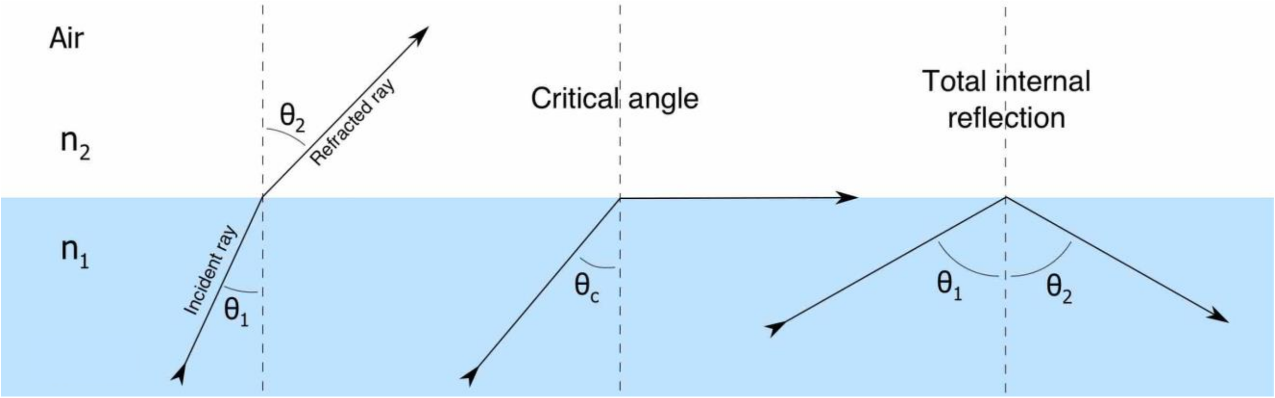


OPTICAL FIBERS AS SENSORS

- Optical fibers rely on the concept of total internal reflection to cause light to travel from one end of the fiber to the other
- The refractive index of the material is determined by the ratio of the speed of light in the medium to a vacuum: $n = \frac{c_{medium}}{c_{vacuum}}$
- The angle at which total internal reflection occurs is: $\theta_c = \arcsin(\frac{n_{atmosphere}}{n_{medium}})$
- For most glass fiber optics $n_{medium} = 1.444$ and $n_{atmosphere} = 1.000350$, therefore $\theta_c =$ 43.8 degrees

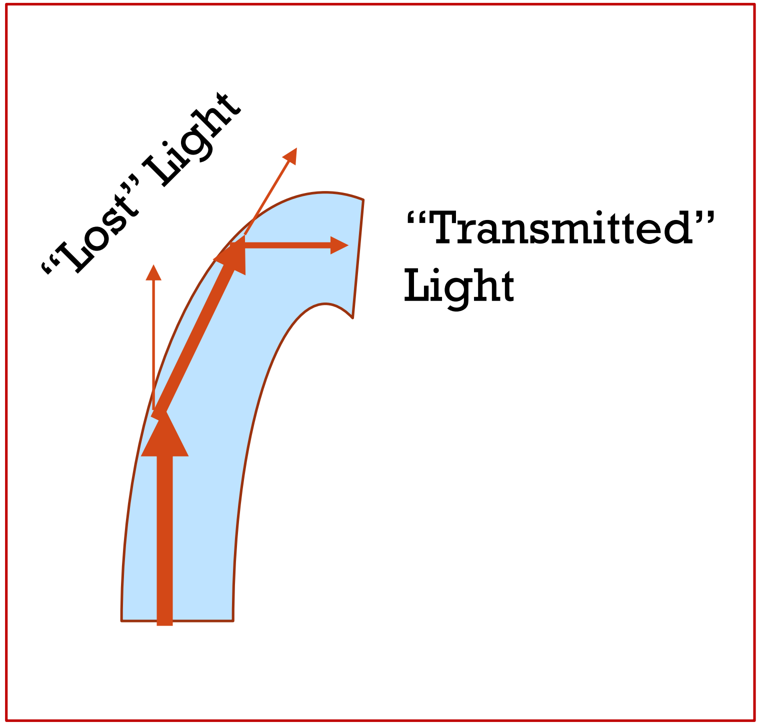




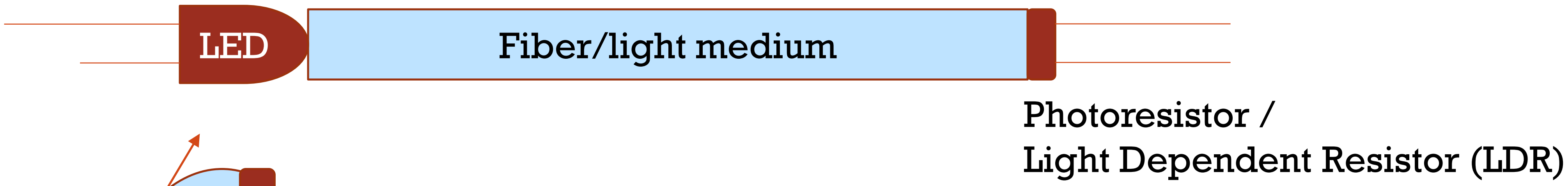


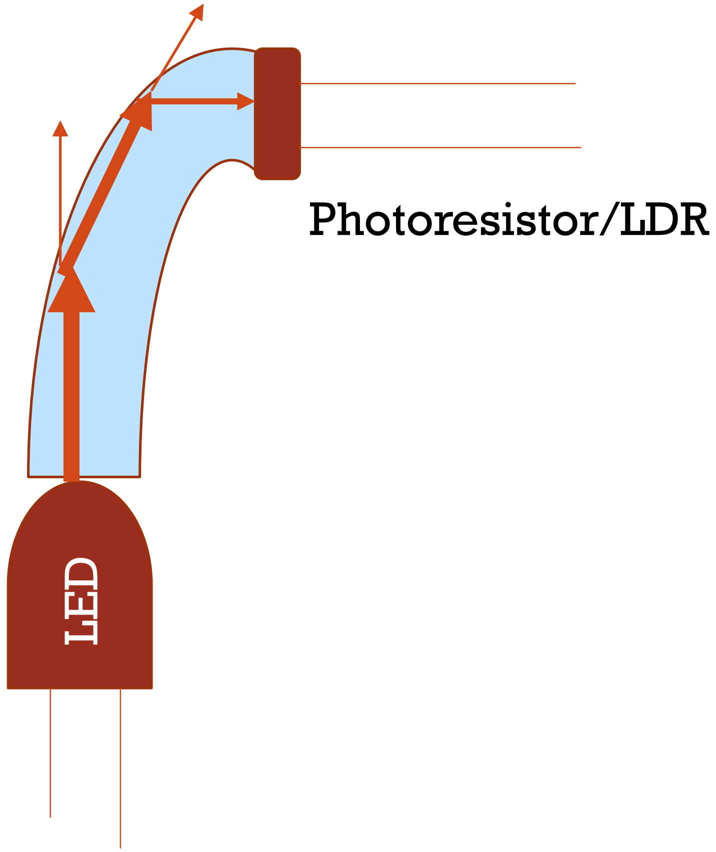
OPTICAL FIBERS AS SENSORS (CONTINUED)

- If light strikes the fiber-atmosphere boundary at less than the critical angle it is completely reflected
- Imperfections and rough surfaces within the fiber can cause light to be scattered and lost
- This loss is made worse when the fiber is bent or deflected
- If fibers were embedded in a wrist strap, deflection due to hand movement would cause characteristic changes in the amount of light transmitted
- Analyzing the transmitted light levels could allow computers to reconstruct gestures!



BASIC SETUP FOR DEFLECTION SENSOR





In order to detect the deflection using an LDR, a significant amount of light must be lost when the fiber is bent

35mm fiber length was determined to be the minimum for a noticeable light deflection when bent

PROJECT METHODOLOGY

**Step 1:
Proof of Concept**

- Determine best material
- Determine best shape
- Determine color of LED light

**Step 2:
Prototype**

- Create a wearable band
- Add sensors
- Develop data collection method

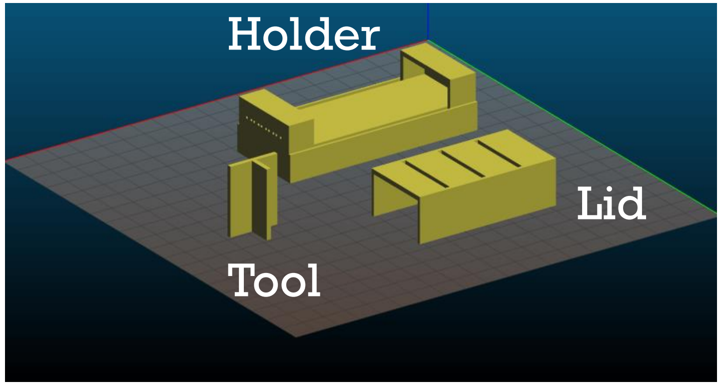
**Step 3:
Data Analysis**

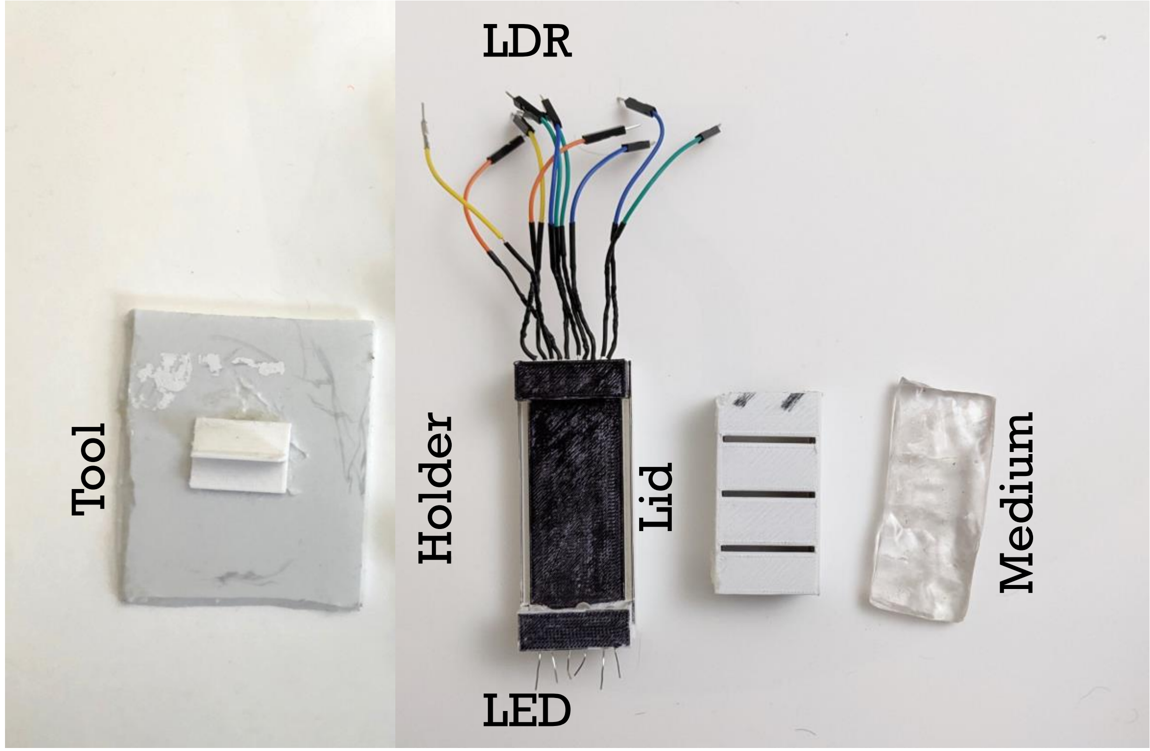
- Measure how light levels are impacted by gestures
- Uniquely identify different a selection of gestures
- Automate process

STEP 1: PROOF OF CONCEPT

HARDWARE

- Designed a hard casing using Solidworks to test concept
- 3D Printed design for consistency and durability
- Key features
  - A holder for the medium to be placed in
  - Lid to eliminate external light and isolated LED light
  - Holder for LED and LDR on each end
  - Plastic tool for consistent depression of medium





STEP 1: PROOF OF CONCEPT

Colour	Medium/Fibre material	Minimum Intensity	Maximum Intensity	Delta Intensity
Large blue	Humimic #2	160	990	**830**
Large red	Humimic #2	183	1013	**830**
Large white	Humimic #2	205	1021	816
Small red	Humimic #2	180	977	797
Large red	Earth's atmosphere	235	996	761
Large yellow	Humimic #2	210	971	761
Small green	Humimic #2	189	937	748
Large blue	Earth's atmosphere	228	957	729
Large green	Humimic #2	185	850	665
Small red	Earth's atmosphere	245	864	619
Small green	Earth's atmosphere	253	775	522
Large green	Earth's atmosphere	240	720	480

Light Colors Tested

- The optimal wavelength for the Light Dependent Resistor (LDR) depends on the type of LDR used
- For my Cadmium Sulfide LDR, Large red (630nm) and large blue (460nm) LEDs were determined to be the best (had the greatest delta intensity when depressed).
- Red was chosen due to availability.
- Meausrements in scaled light intensity (0-1023 which represents 0-5V)

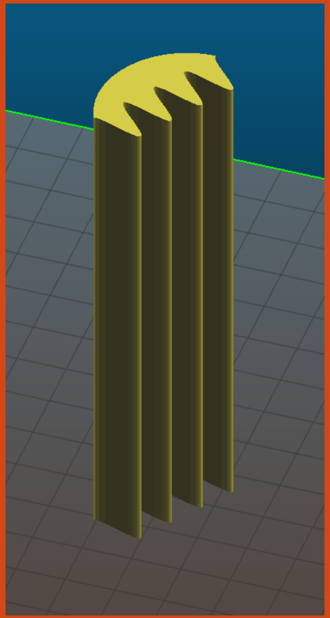
STEP 1: PROOF OF CONCEPT

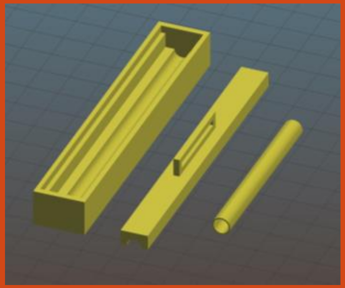
Mediums Tested

- Smooth On Ecoflex (30) – an opaque silicone
  - Too opaque and did not permit any light from the LED to pass
- Humimic gel – a transparent synthetic gelatin
  - Too soft. Melted even due to body heat
- **Smooth On Vytaflex (20) – a semi-transparent urethane rubber**
  - Worked the best. Transmitted light the best and sturdy. It also exhibited significant changes in intensity when depressed (~100).
- Standard optical fiber (from Adafruit)
  - Light intensity did not change with depression and was always at the maximum (1023)

Materials: Ecoflex, Humimic, Vytaflex, optical fiber

STEP 1: PROOF OF CONCEPT





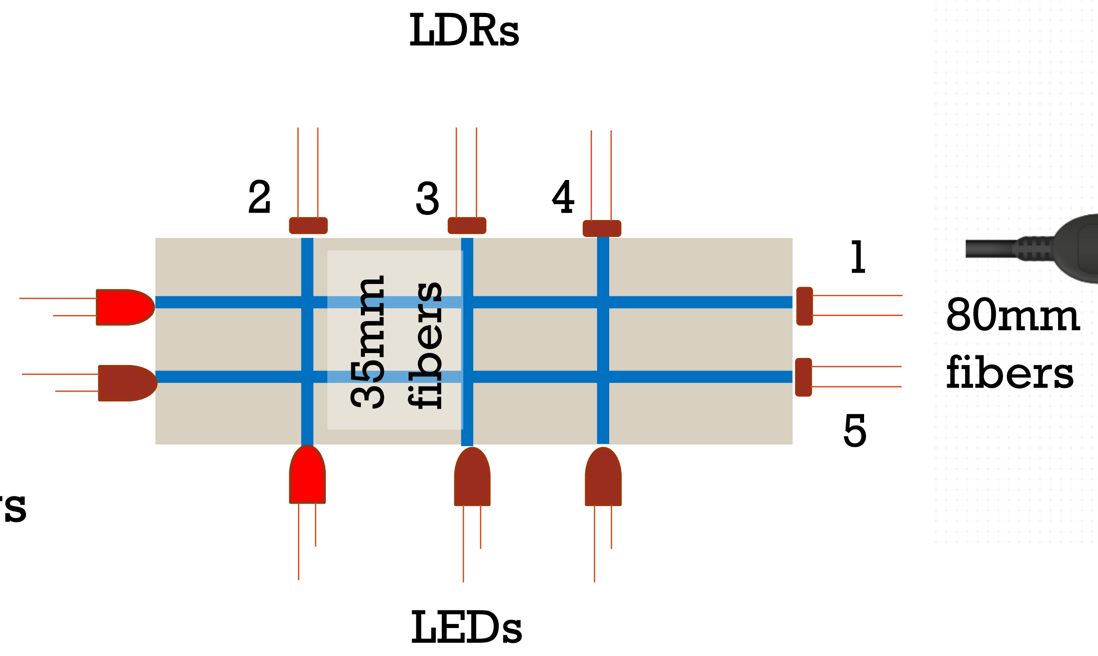
Moulds:
Straws,
heat shrink,
3d printed,
sinusoidal

Fiber Shapes

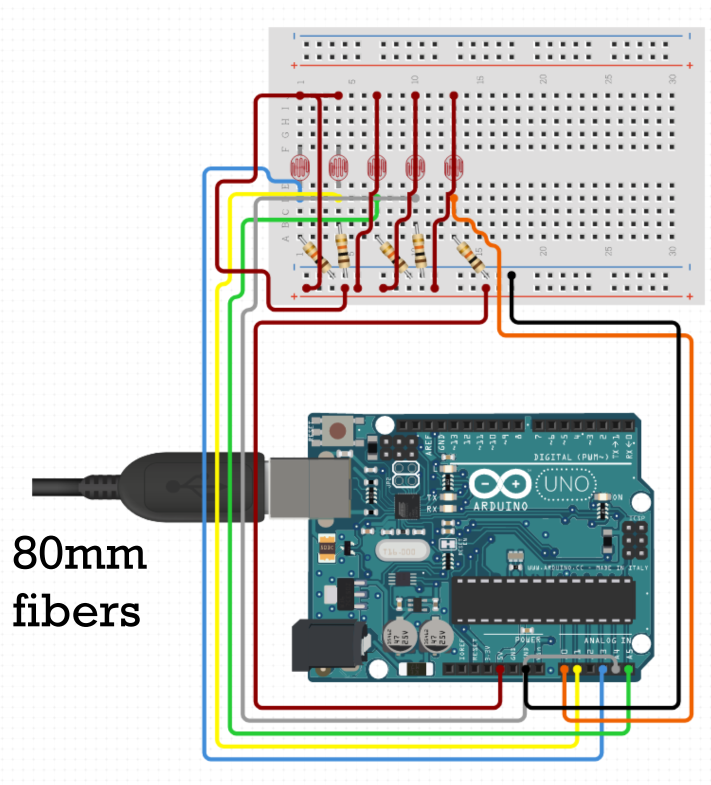
Tube Molding Material	Pros	Cons
Shrink tube (20mm, 5mm, 4mm)	Somewhat easy to obtain, many sizes	Material stuck to sides Did not mold well
3D Printed Tube (2mm, 4mm, 5mm)	Customizable	Material stuck to sides Material could not be removed
Straw Mold Tube (3mm, 5mm)	Easy to obtain 3mm proved to be best for LED shape and resolution	5mm proved to be too thick and inflexible
Rectangular	Easier to fabricate	sensors must be embedded → fragile
Sinusoidal	Degree of light path independence between sensors	Sensors must be embedded → fragile

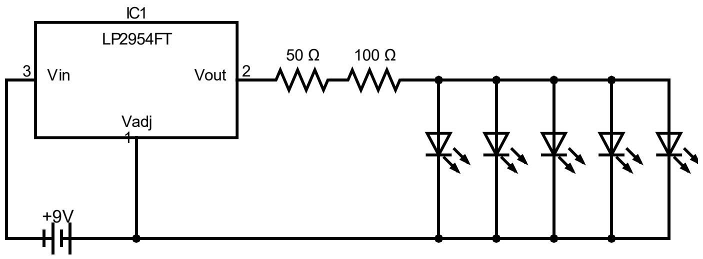
STEP 2: PROTOTYPE HARDWARE

- Three 35mm and two 80mm Vytaflex fibers molded using 3mm straw
- Large red (630nm) LEDs
- LEDs and LDRs are fastened using heat shrink to the fiber
- 1 Arduino Uno
- 5 LDRs and 5 10k Ohm resistors
- LED circuit: 9V battery, 5V regulator, 50 ohm resistor, 100 ohm resistor, Ohms resistors, 5 LEDs



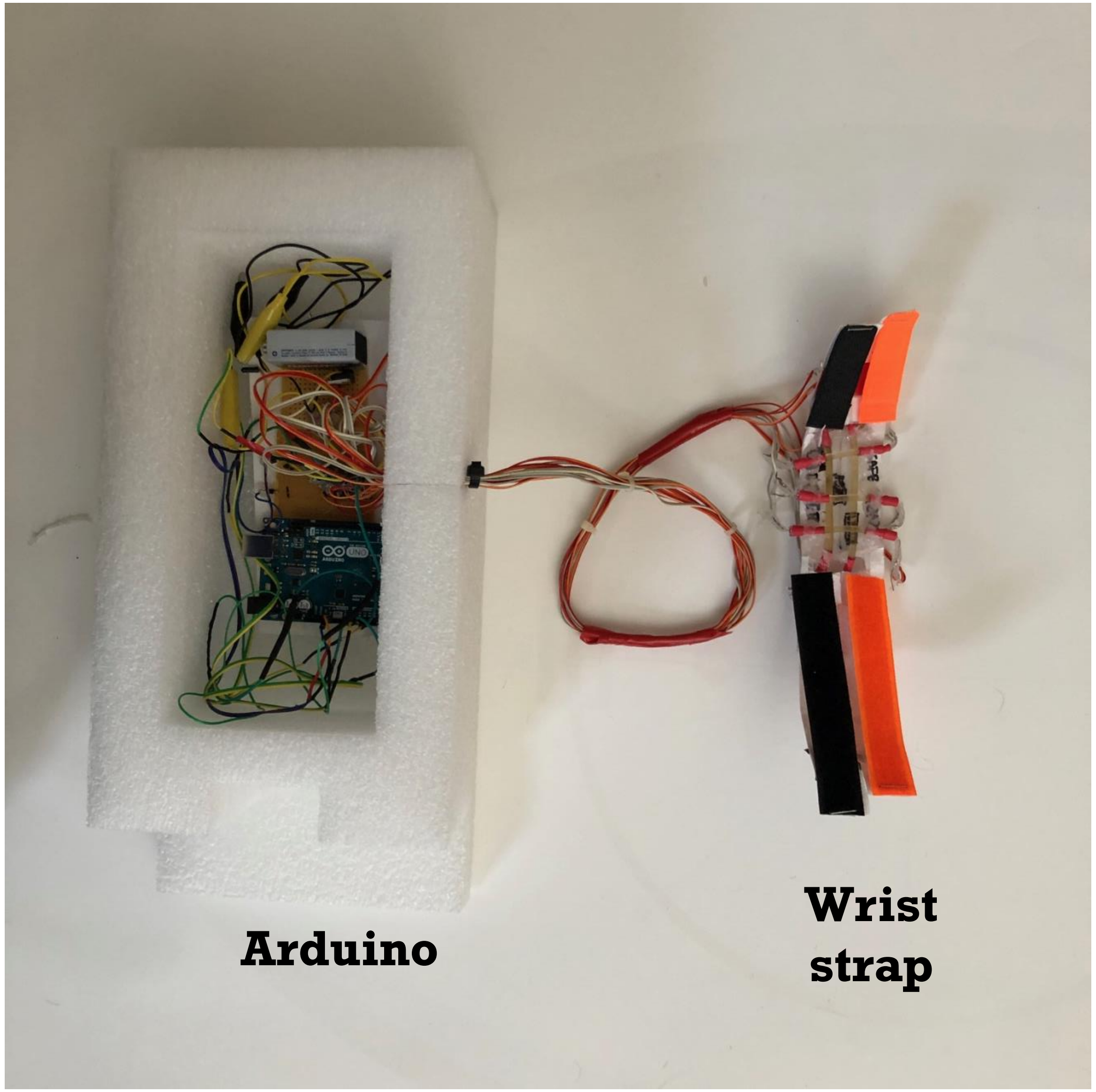
1cm separation between fibers





STEP 2: ACTUAL HARDWARE SETUP

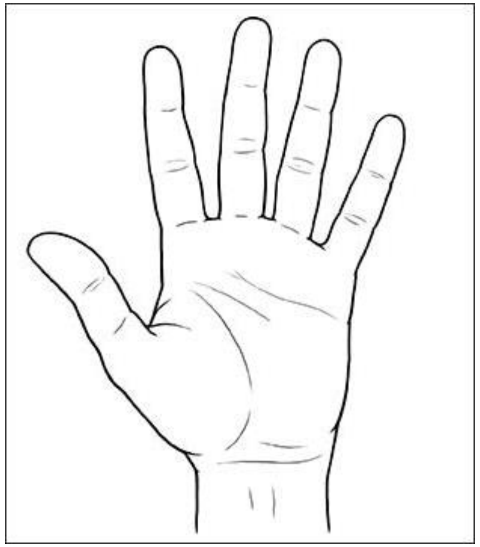
- Setup on the previous slide is mounted with Velcro bands
- The band is placed on the wrist
- The Arduino is used to collect data
- Code written in Python and C++



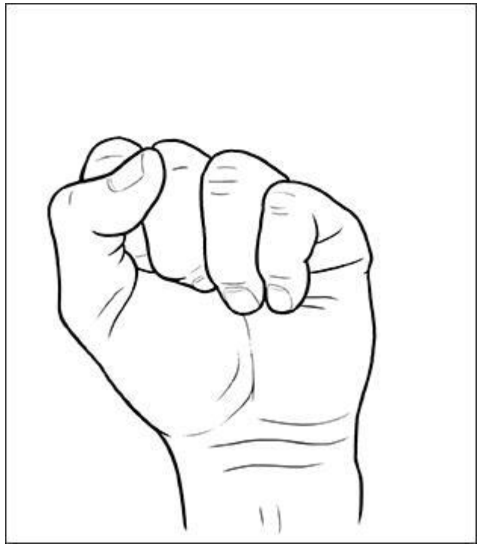
STEP 2: GESTURES TESTED

- A large collection of gestures can be tested using the prototype
- I focus this presentation on three gestures

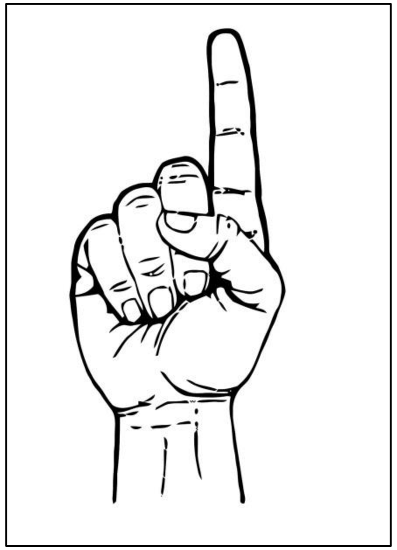
Finger Extension
[Extend]



Finger Flexion
[Fist]



One Finger
[One]



STEP 3: DATA ANALYSIS

Light Intensity over Time

↓

Manually Label Hand Position

↓

Moving Averages vs. Original

↓

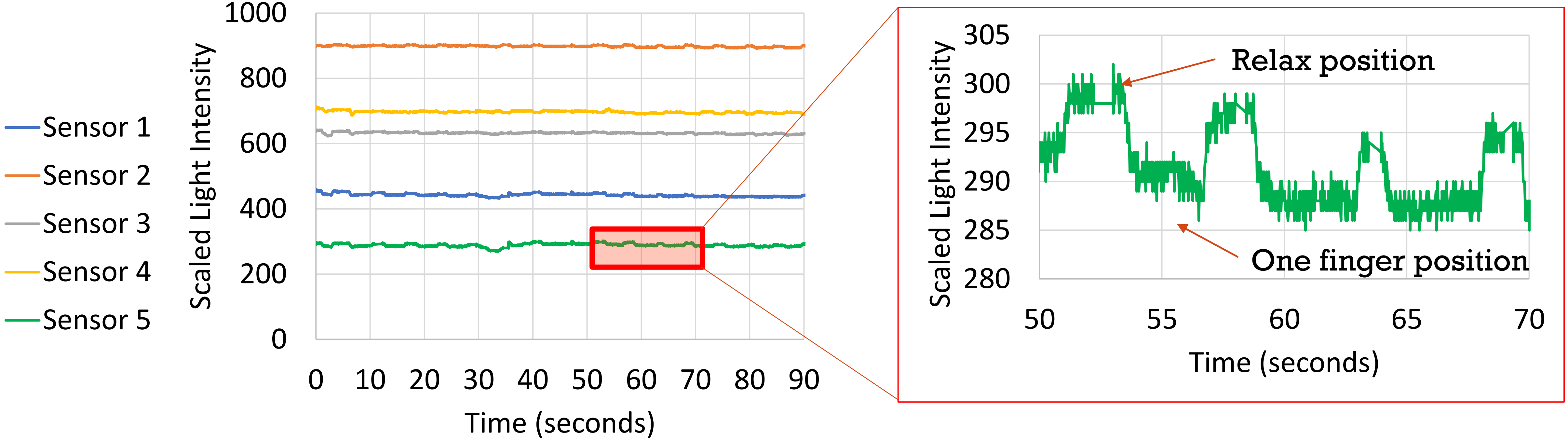
Compute the Gesture Label

↓

Compare

- 5 seconds of data was collected for each gesture
- Includes a brief relax and transition period before and after gesture
- Approximately 1 reading per 0.02s (250 readings/s)
- Light intensity in scaled units (0 to 1023) based on the voltage drop due to the LDR (0 to 5V)

STEP 3: DATA ANALYSIS [ONE FINGER]

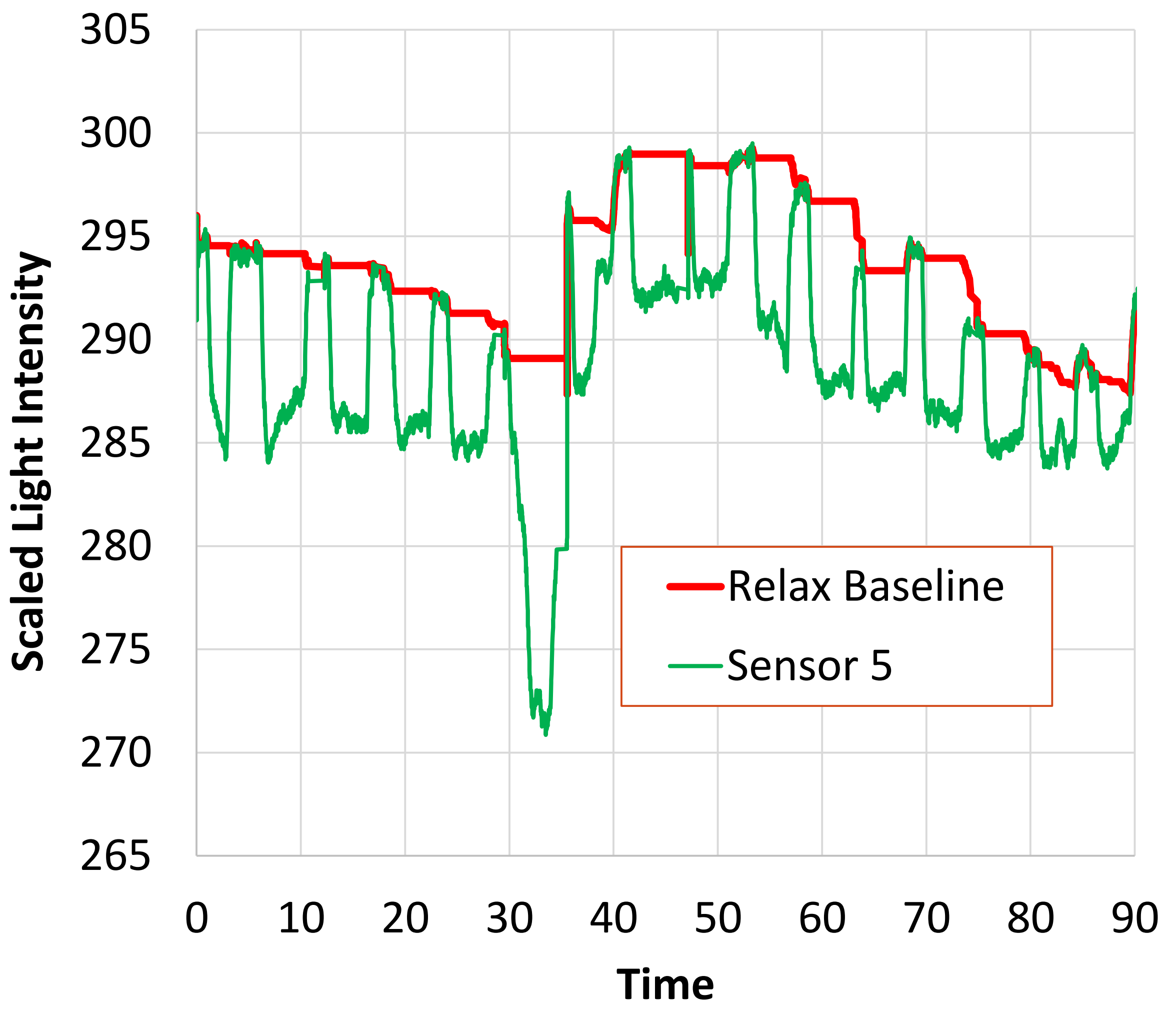


- Plot of scaled light intensity of each sensor over time as I repeatedly performed a one finger gesture
- Each sensor (LDR) has its own unique baseline intensity value
  - Each sensor need to be calibrated to a specific value separately
- Each sensor detects the change in intensity when hand gestures are made

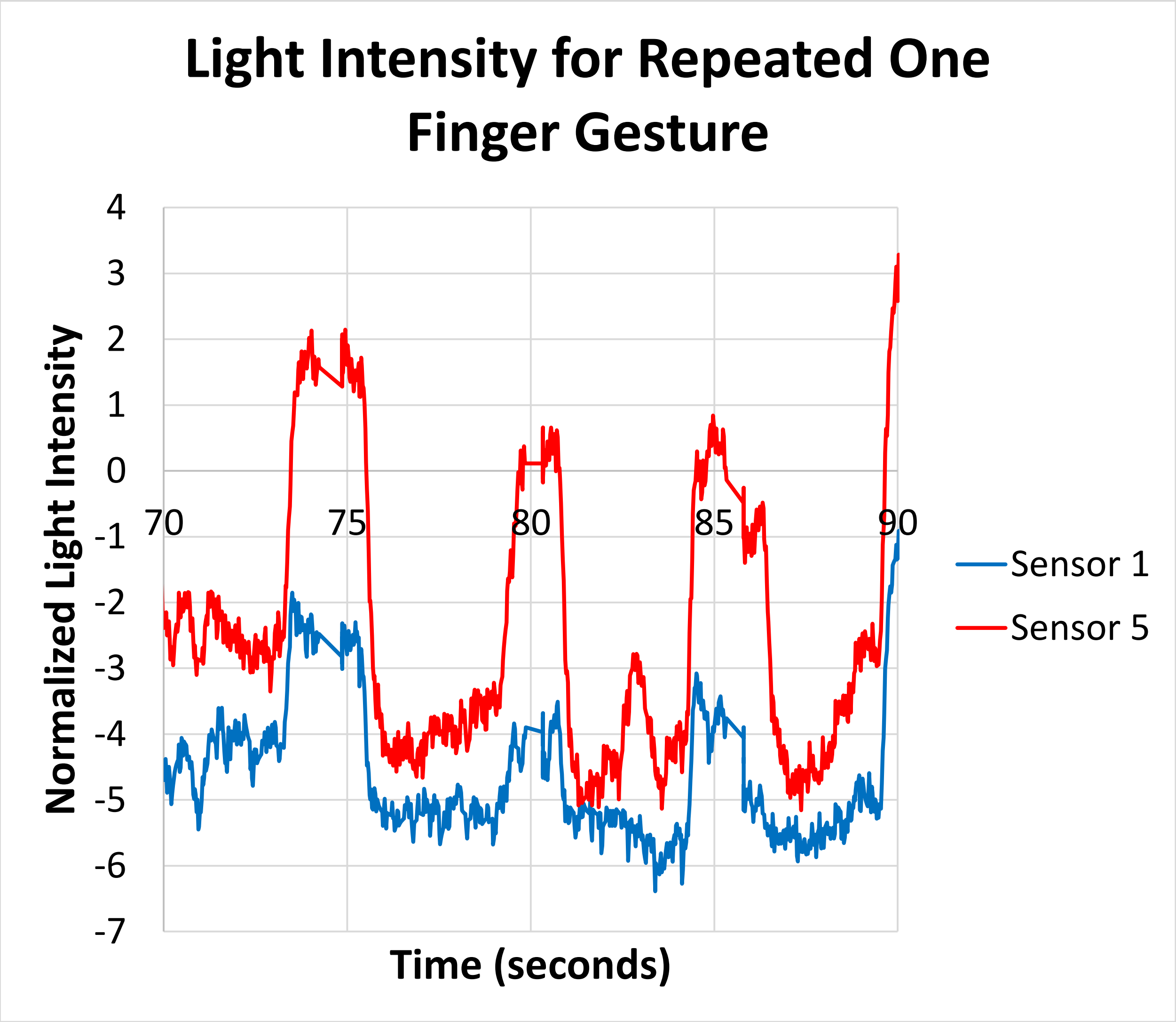
STEP 3: HANDLING CHANGING READINGS

- Graph of sensor 5 for the one finger gesture shows that the baseline readings for a sensor can change over time
  - E.g., due to movement of the wristband
- Need to distinguish between changes due to gestures and changes in the baseline
- Two techniques to establish a baseline

1. Exponentially-weighted Moving Average: I use a moving average remove outliers and smooth noisy data
   - $A_n = \alpha * x_n + (1 - \alpha) * A_{n-1}$
   - α determines how many "past values" to include in the average
     - α close to 0 weighs past readings heavily
     - α close to 1 makes average closely to current reading
     - Tuning α to 0.2 worked well
2. Sliding Window: I keep track of the maximum reading in the past 4 seconds. This (shown in red line) followed the changing reading well



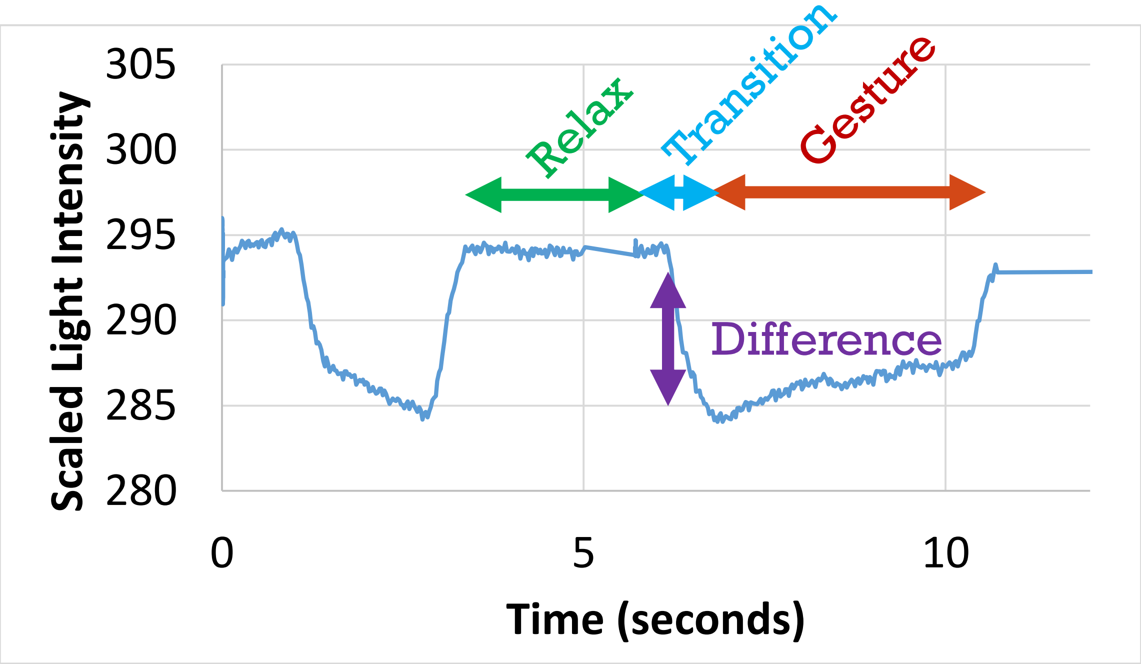
STEP 3: IDENTIFYING A SINGLE GESTURE



- The graph shows readings of two sensors "normalized" by subtracting the average value of their readings (i.e. centering graph around value of 0)
- Some sensors react more significantly to specific gestures
  - Sensor 5 detects the deflection more intensely than sensor 1
- By examining the magnitude of the change across all sensors, a unique "signature" can be determined for each type of gesture

STEP 3: CREATING A GESTURE "SIGNATURE"

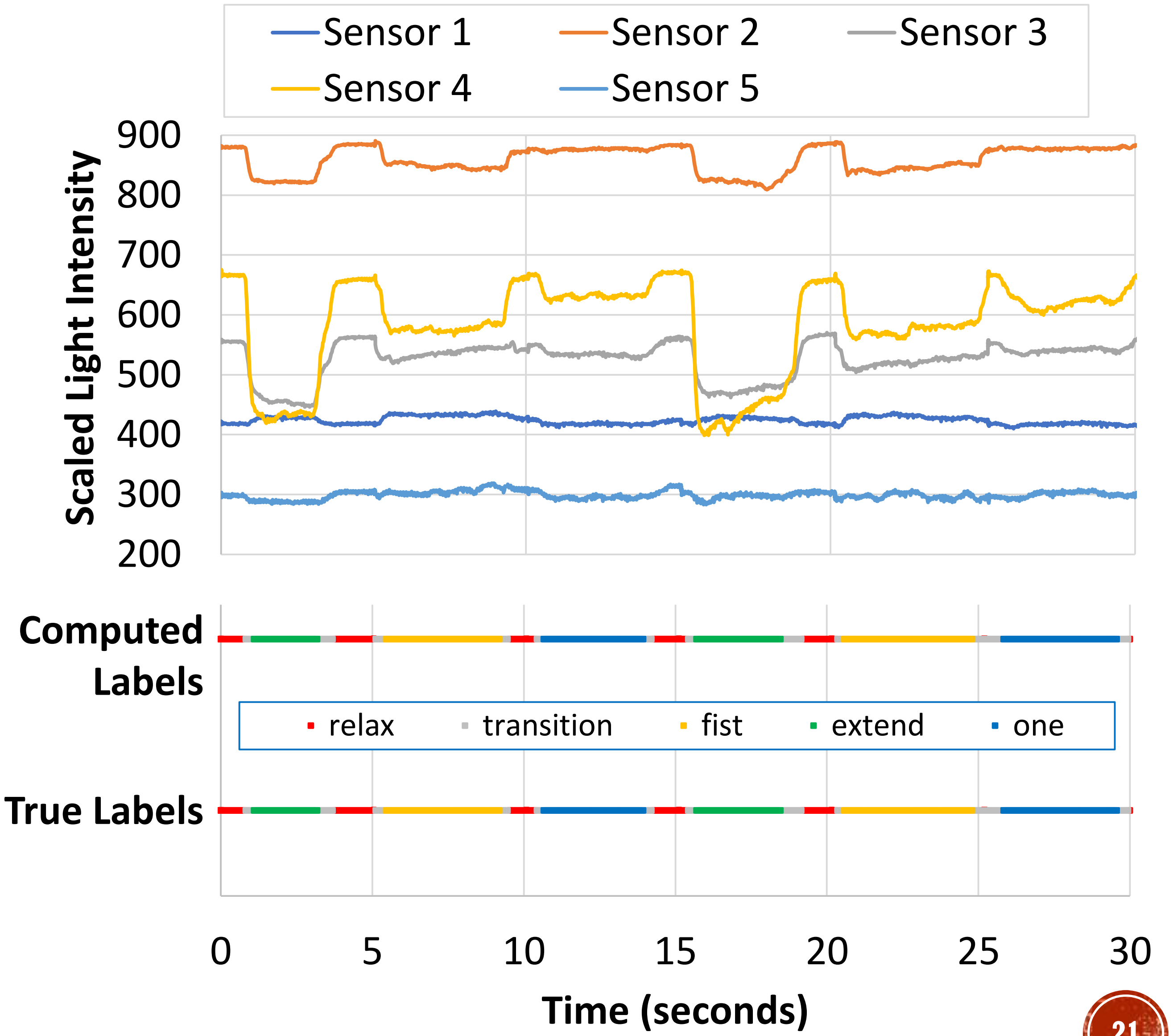
- For each gesture, I manually labelled the data readings of a single trace as "Relax Baseline", "Transition Between Relax and Gesture" and "Gesture" periods
- I computed a simple average ($\bar{x} = \frac{\sum_{i=1}^{n} x_i}{n}$) reading of each period
- I then computed the difference in Light Intensity between a Relax period and the subsequent Gesture period
- This was done for each sensor and for each gesture
- The table shows the average light intensity difference for each sensor/gesture
- From this, I chose to use Sensors 2,3 and 4 to identify gestures



	Sensor 1	Sensor 2	Sensor 3	Sensor 4	Sensor 5
Extend	8	-48	-90	-223	-12
Fist	12	-34	-32	-60	1
One	-3	2	-24	-49	-4

STEP 3: AUTOMATIC GESTURE LABELING

- I collected sensor readings from two gestures of each type performed over 30 seconds
- I used the computed signatures to automatically compute labels for each data point
  - If the difference of the current reading from the baseline was within 10% of the signature value, it was assigned that label
  - If the current reading was not within 10% of any label, it was marked as a "relax" (i.e. baseline gesture)
  - If the computed label had changed within the past 10 readings, the label was changed to "transition"



STEP 3: LABELING ACCURACY

- Accuracy of a classification system is typically summarized using a confusion matrix
- The bulk of the computed classification are correct (highlighted along the diagonal)
- The misclassification are mostly associated with the transition state
  - Manual inspection suggested that these were associated slightly different estimates for the start/end of a gesture
  - Note that there is some uncertainty in the true labeling of the transition state
  - The impact of this misclassification on a gesture system is unlikely to be significant

Computed Label

True Label	Relax	Transition	Fist	Extend	One
Relax	**652**	3	0	0	2
Transition	62	**83**	14	17	21
Fist	0	10	**339**	0	0
Extend	0	8	0	**206**	0
One	0	13	0	0	**295**

CONCLUSIONS AND NEXT STEPS

- Fiber optics are a practical method of detecting deflection in the wrist
- This is inexpensive: LEDs, LDRs, resistors, and basic electronic equipment are ~$0.10 each, the Vytaflex is $27 for 0.91 kg (my setup uses roughly 4 mL or ~15 g), and the Arduino Uno is $19, so the total prototype cost was about $25-30.
- The project is a promising first step in creating a light-weight assistive technology
- The next step of the project is to increase the number of gestures recognized and complete the machine learning training on the data to auto-detect gestures
- The final design for the wrist strap will be to have the system in a watch band and powered by a smart watch rather than an Arduino

APPENDIX: CODE

```
int sensorPin = A0; // select the input pin for LDR(s)
int sensorPin2 = A1;
int sensorPin3 = A2;
int sensorPin4 = A3;
int sensorPin5 = A4;

int sensorValue = 0; // variable(s) to store the value coming from the sensor
int sensorValue2 = 0;
int sensorValue3 = 0;
int sensorValue4 = 0;
int sensorValue5 = 0;

void setup() {
    Serial.begin(9600); //sets serial port for communication
    pinMode(LED_BUILTIN, OUTPUT);
}
void loop() {
    sensorValue = analogRead(sensorPin); // read the value(s) from the sensor(s)
    sensorValue2 = analogRead(sensorPin2);
    sensorValue3 = analogRead(sensorPin3);
    sensorValue4 = analogRead(sensorPin4);
    sensorValue5 = analogRead(sensorPin5);

    Serial.print(sensorValue); //prints the values coming from the sensors on the screen
    Serial.print(",");
    Serial.print(sensorValue2);
    Serial.print(",");
    Serial.print(sensorValue3);
    Serial.print(",");
    Serial.print(sensorValue4);
    Serial.print(",");
    Serial.println(sensorValue5);

}
```

C++ Arduino Code

Python data collector

```python
#!/usr/bin/python

import serial
import time
import sys

# Determine the file to save data to and Arduino port
fileOut = sys.argv[2]
port = sys.argv[1]

f = open(fileOut, "w")

# Open Arduino serial
ser = serial.Serial(port, 9600, timeout=1)
def getData(timeout, label):
    # measure delta time of a single gesture cycle for timeout
    timeI = time.time()
    while time.time()-timeI < timeout:
```

CODE

```
        try:
            reading = ser.readline() # Read sensor values from Arduino
            # Record delta Time, Unix time, ldr readings, and label to a
file
            data = [0,0,0,0,0,0,0,""]
            data[0] = time.time()-timeIA
            data[1] = time.time()
            data[2] = int(reading.split('\n')[0].split(",")[0])
            data[3] = int(reading.split('\n')[0].split(",")[1])
            data[4] = int(reading.split('\n')[0].split(",")[2])
            data[5] = int(reading.split('\n')[0].split(",")[3])
            data[6] = int(reading.split('\n')[0].split(",")[4])
            data[7] = label
            for i in data:
                print str(i)+",",
                f.write(str(i)+",")
            print ""
            f.write("\n")
        except:
            pass
```

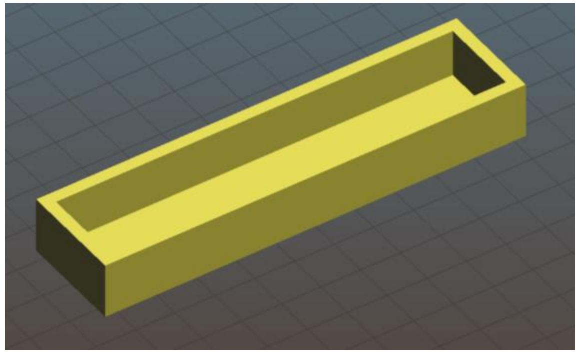
Python data collector cont.

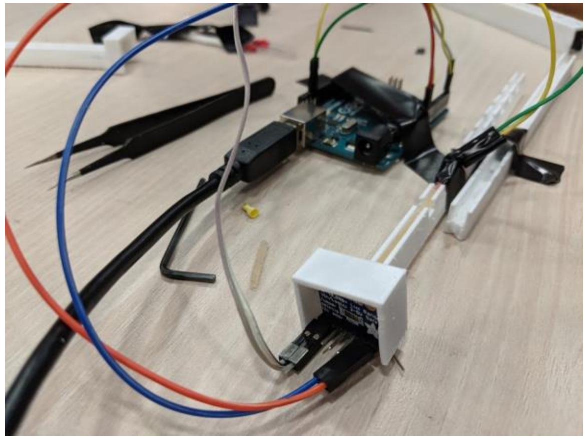
CODE

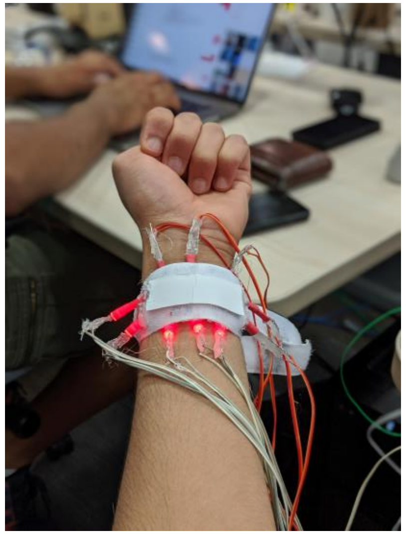
```
gestures = ["extend","fist","one"]*2 # declare labels of which gestrues are being
 tested

print("delta Time, Unix Time, pr1, pr2, pr3, pr4, pr5, label") # file headers
f.write("delta Time, Unix Time, pr1, pr2, pr3, pr4, pr5, label\n")

# Collect data for each gesture cycle for 5.0s
timeIA = time.time()
for gesture in gestures:
    continueQ = raw_input("Do "+gesture+"? [Y/n]: ")
    if continueQ == "y" or continueQ == "" or continueQ == "Y":
        getData(5.0, gesture)
    elif continueQ == "n" or continueQ == "N":
        continueQ = "n"
f.close()
```

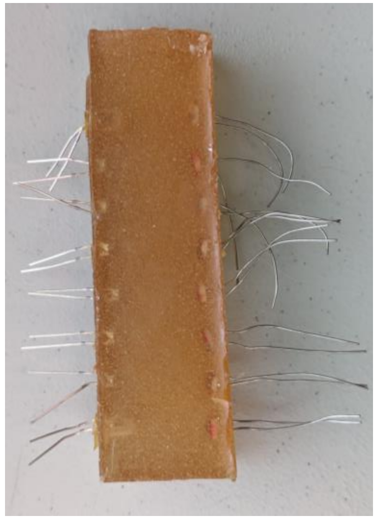
Python data collector cont.

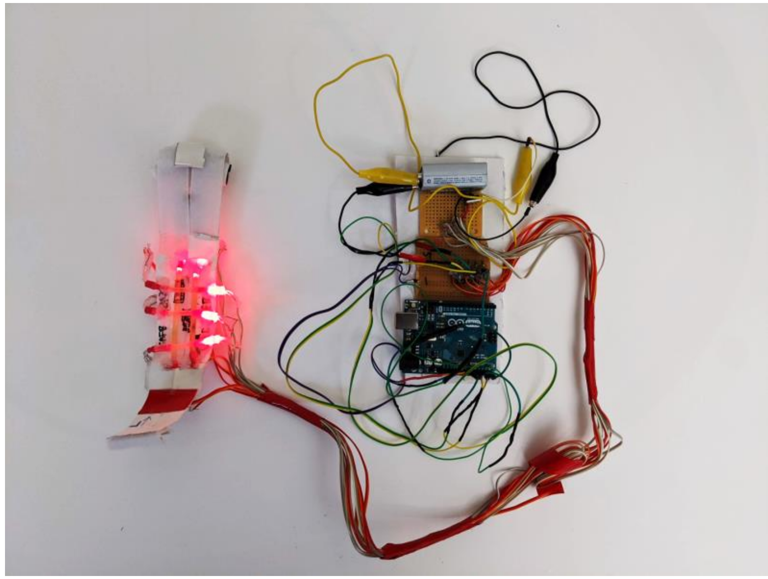
APPENDIX: MISCELLANEOUS PICTURES

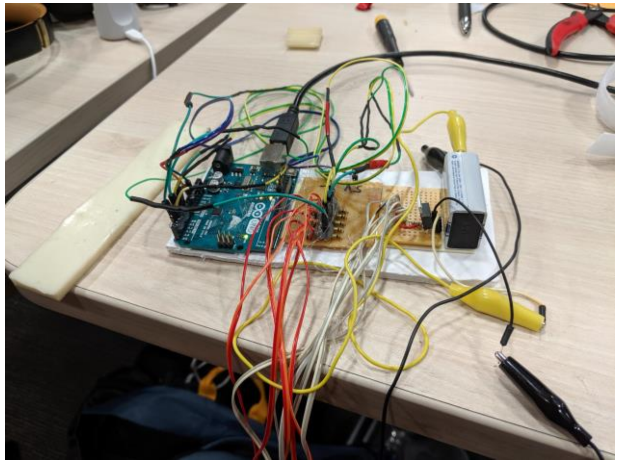


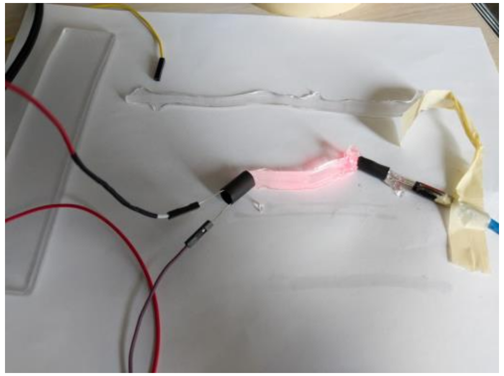












BIBLIOGRAPHY

- To, Celeste, Tess Lee Hellebrekers, and Yong-Lae Park. "Highly Stretchable Optical Sensors for Pressure, Strain, and Curvature Measurement."
- Meerbeek, I. M. Van, et al. "Soft Optoelectronic Sensory Foams with Proprioception." Science Robotics, Science Robotics, 28 Nov. 2018, robotics.sciencemag.org/content/3/24/eaau2489.abstract.
- Cho, Haedo, et al. "Design of an optical soft sensor for measuring fingertip force and contact recognition." International Journal of Control, Automation and Systems 15.1 (2017): 16-24.
- Butt, Jawad Mehmood, Hesheng Wang, and Radan Pathan. "Design, Fabrication, and Analysis of a Sensorized Soft Robotic Gripper." 2018 IEEE 8th Annual International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER). IEEE, 2018.
- Chen, Wenbin, et al. "Fabrication and Dynamic Modeling of Bidirectional Bending Soft Actuator Integrated with Optical Waveguide Curvature Sensor." Soft robotics (2019).
- Heintz, Brenner. "Training a Neural Network to Detect Gestures with OpenCV in Python." *Medium*, Towards Data Science, 17 Dec. 2018, towardsdatascience.com/training-a-neural-network-to-detect-gestures-with-opencv-in-python-e09b0a12bdf1.
- Poupyrev, Ivan. "Google I/O 2015." *Google*, Google I/O Conference, 2015, www.youtube.com/watch?reload=9&v=mpbWQbkl8_g#t=810.

PHOTO CREDIT

- https://orthoinfo.aaos.org/en/diseases--conditions/carpal-tunnel-syndrome/
- https://www.semanticscholar.org/paper/Wrist-worn-hand-gesture-recognition-based-on-Zhu-Jiang/
- https://thegadgetflow.com/
- http://hyperphysics.phy-astr.gsu.edu/hbase/optmod/fibopt.html
- http://oer2go.org/mods/en-boundless/www.boundless.com/physics/textbooks/boundless-physics-textbook/geometric-optics-24/reflection-refraction-and-dispersion-169/total-internal-reflection-and-fiber-optics-609-6258/index.html
- https://en.wikipedia.org/wiki/Wired_glove
- https://www.flickr.com/photos/brixendk/4349817216
- https://www.amazon.com/
- https://www.adafruit.com/
- Minority Report (2002 Movie)
- Sanjay Seshan