A2 SHARE OUTS & CRITIQUE + INTRO TO A3

CSE 599 Prototyping Interactive Systems | Lecture 10 | May 2

Jon Froehlich • Jasper O'Leary (TA)

DESIGN USE BUILD

MAKEABILITY LAB

PAUL G. ALLEN SCHOOL OF COMPUTER SCIENCE & ENGINEERING

UNIVERSITY of WASHINGTON

A2 SHARE OUTS & CRITIQUE

A2: Fabrication: 3D-Printed Interactive Night Light

Published | Edit

Related Items
SpeedGrader™
Download Submissions
0 out of 9 Submissions Graded

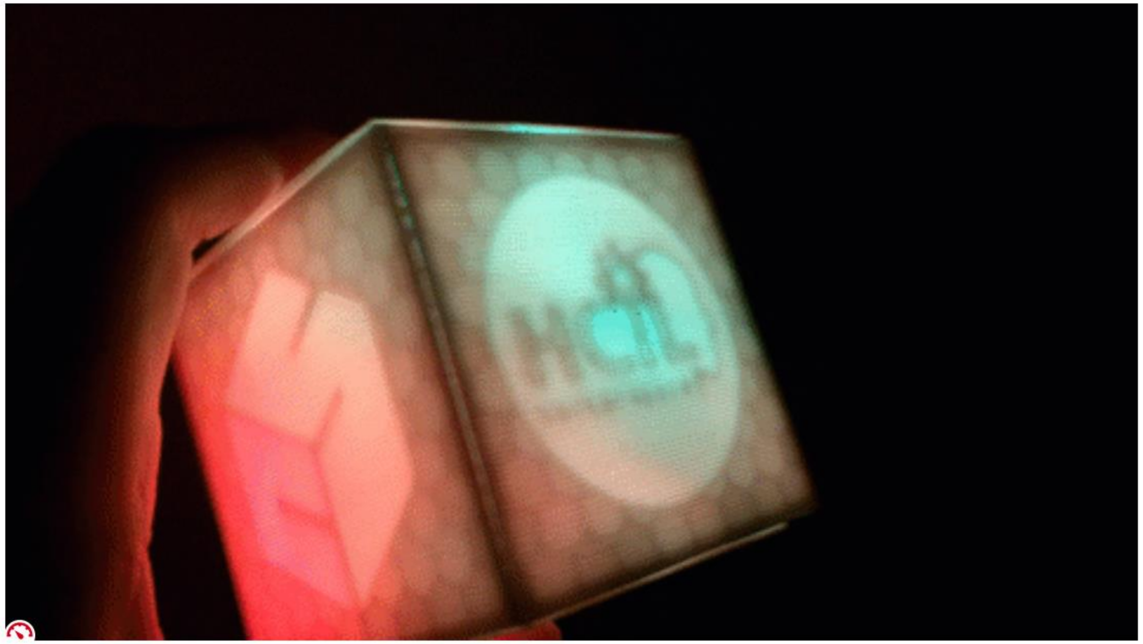
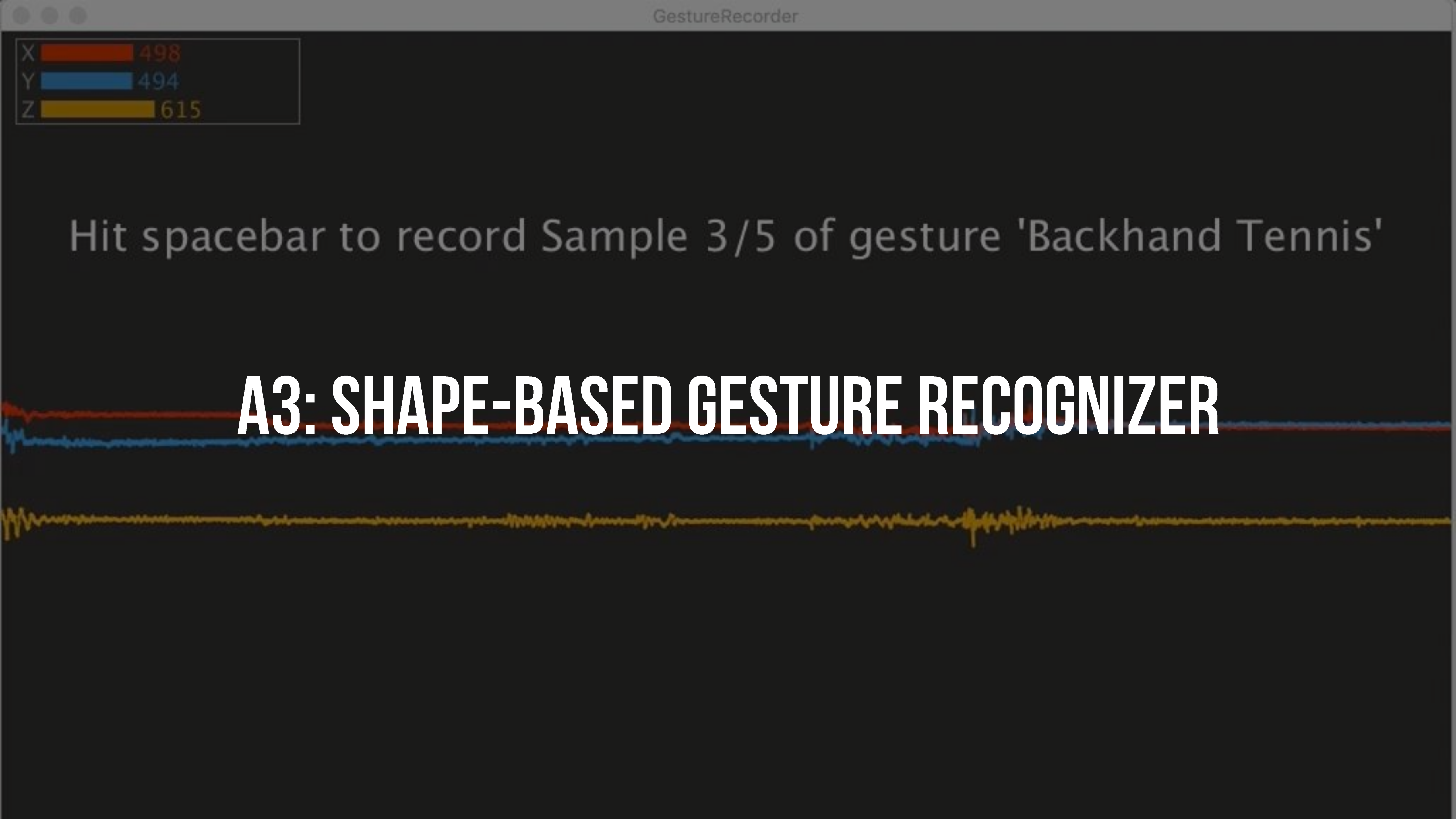


Image caption: The *Tangible Interactive Computing* Top Maker Award from CMSC838f, Spring 2015 designed by Jon Froehlich based on the Holocron Nightlight by CMSC838f student Philip Dasler.

Overview

In this assignment, you will design and fabricate a 3D-printed interactive night light, which responds to user interaction, creatively diffuses the light (*e.g.*, playfully or elegantly), and fully encloses your Arduino and electronics. The specific model is up to you but should include: (1) a mounting stand for the internal electronics; (2) a carefully measured and tightly fit input slot for the USB micro cable to power your design; (3) and similarly well-designed fittings for any input controls you want to expose. You will likely need to design and print a multi-part model, which can be reassembled into a full form (e.g., similar to this Arduino case ... but, of course, your night light will have to

A3: SHAPE-BASED GESTURE RECOGNIZER



A3: Signal Processing + Machine Learning 1: Shape-Matching Gesture Recognizer

Overview

Imagine working for a new hardware startup designing new input controllers. You've been asked to prototype a new, custom input device with gesture recognition (using an accelerometer)--for example, to use the input controller as a paddle in tennis, as a "ball" in bowling, or to recognize the "overhand throwing motion" in baseball. In this assignment, you will build your own 3D gesture recognizer to automatically recognize these gestures.

While in the "real world" you would ultimately need to create a real-time gesture recognizer, for this assignment you will make an *offline* version in Jupyter Notebook (we strongly recommend the Anaconda Distribution). Specifically, in this assignment you will build a *shape-matching* (or *template-matching*) recognizer such as via a Euclidean distance metric or Dynamic Time Warping. In A4, you will build a *feature-based* (or *model-based*) recognizer using a support-vector machine (SVM) (recommended) or an alternative supervised learning approach of your choosing (*e.g.*, an HMM).

Within Jupyter Notebook, we will use Python 3 and these amazing libraries numpy, scipy, matplotlib, and scikit-learn. Numpy and scipy provide numeric array handling and signal processing, matplotlib provides visualization, and scikit-learn is the de facto machine learning library in Python. You are welcome to use other libraries as well (*e.g.*, this DTW library).

For your deliverables, you will turn in your Jupyter Notebook, your recorded gestures, and a slide deck report on your algorithmic approaches and performance results.

Things You Need

- Download and install Jupyter Notebook for offline data analysis, signal processing, and machine learning. We've created an initial skeleton to parse the data, which is available here.
- The ADXL335 3-axis accelerometer with soldered header pins hooked up to an Arduino running ADXL335GestureRecorder.ino (the ADXL335 is a $15 part--the 2nd most expensive in your kits behind the Arduino itself--so, please solder these pins only after you have completed the in-class LED flashlight soldering exercise and feel comfortable doing so. If you need help, please ask)

ASSIGNMENT

A3: SHAPE-MATCHING GESTURE RECOGNIZER

Parts

- [1 pt] **Record your own gesture set**. See above.

- [2 pts] **Visualize and explore both gesture sets** (my recordings of the gestures and your recordings of the gestures) in Jupyter Notebook. Your visualizations should include anything that helps you analyze the signals and aid your signal processing and classification approaches. At the very least, you should visualize the raw x, y, and z accelerometer signals as line graphs (as we did in class) as well as processed versions. Please appropriately label axes, titles, and include legends.

- [3 pts] **Design and implement a shape-matching gesture** recognition approach (*e.g.*, using DTW). What transformations of the signal are necessary here (*e.g.*, smoothing, detrending, *etc.*)?

- [2 pts] **Evaluate your shape-based matching approach** using k-fold cross validation . For each user (my gesture set and your gesture set), randomly split the data into k-folds (k=5). Use four folds for training and one for testing and repeat this five times (with a different fold reserved for testing each time). You do **not** need to examine cross-user performance (*e.g.*, training on my gesture set and testing on your gesture set); however, see the Bonus section. For performance metrics, your Notebook should print out the following for both the shape-matching and model-based approaches: (i) overall accuracy; (ii) per-gesture accuracy; (iii) and a confusion matrix.

- [+1 pts] **You will receive one bonus point** if your shape-matching approach perfectly classifies my gesture data.

ASSIGNMENT

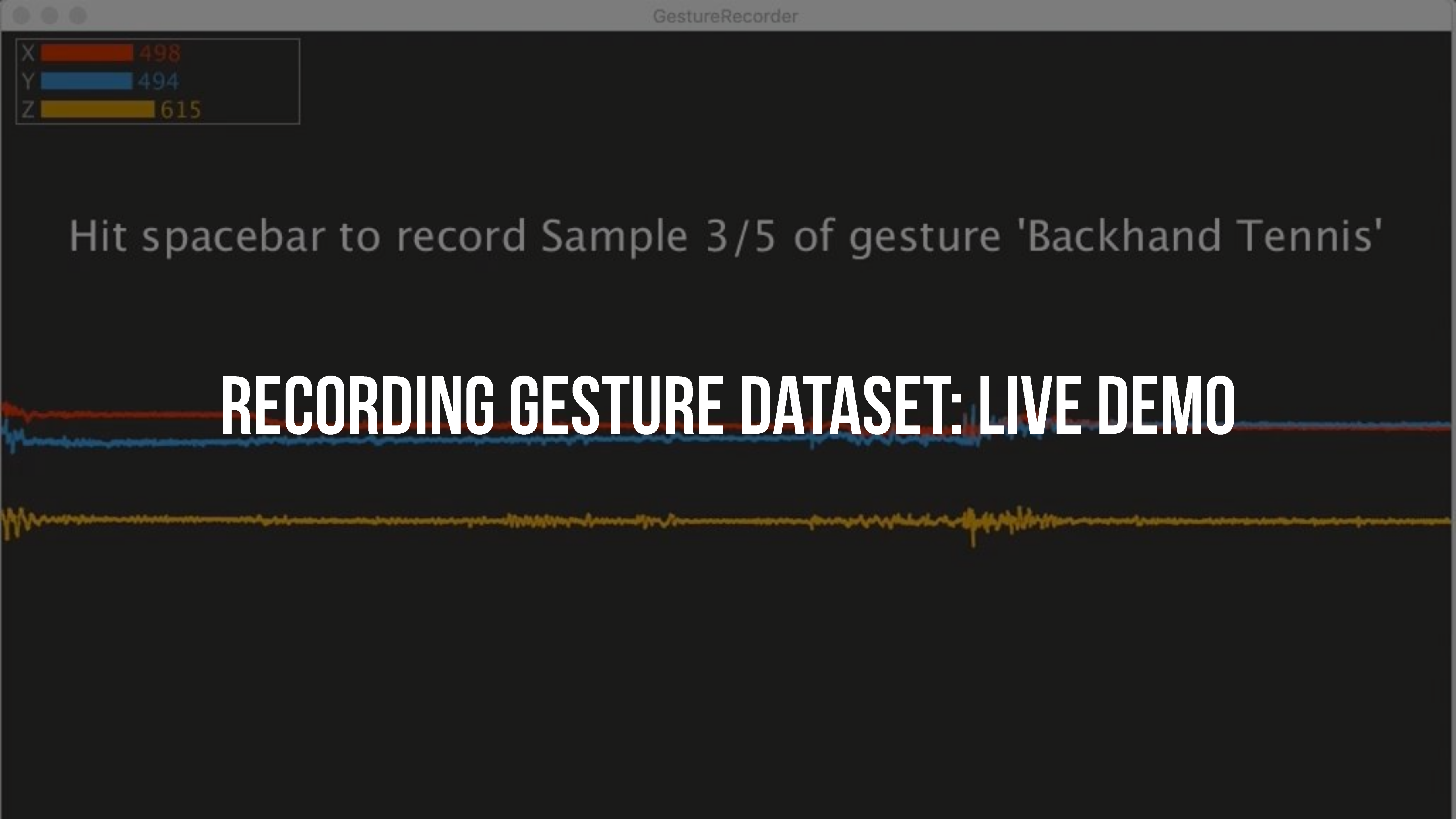
A3: SHAPE-MATCHING GESTURE RECOGNIZER

Deliverables

- [1 pts] Your **Jupyter Notebook** + your gesture set (either a github or gitlab link). Your Notebook should include all the code you wrote to visualize, process, and classify the gestures along with an evaluation framework + performance results. Your Jupyter Notebook should be clear, well-organized, and sufficiently commented (with additional markdown as necessary). As always, please acknowledge any websites or other sources you used to inform your solutions and code.

- [1 pts] A **16:9 slide deck** that with: (i) a description of your shape matching approach and its performance including overall accuracy, per-gesture accuracy, and a confusion matrix; (ii) a description of your model-based classification approach and its performance including overall accuracy, per-gesture accuracy, and a confusion matrix; (iii) an enumeration of key challenges; (iv) and a reflection of what you learned. You can include as many images as you want (*e.g.*, copy/pasted from your Notebook). Images are free.

RECORDING GESTURE DATASET: LIVE DEMO



Project Pitches

Published Edit

Please read the Final Project assignment first then loop back to read the project pitch assignment.

While this assignment is individual, you have the choice to work on your course project individually or with a partner.

Project Pitches

For your pitches, you must submit:

- **A brainstorm sheet or sheets** (can be scanned from paper or born digitally) enumerating at least 10 different project ideas (at least a sentence or two per idea of explanation)

- You will also **downselect** to your top **two favorite ideas.** For those two ideas, we would like you to write 1-2 paragraphs explaining the idea (with sketches, if you'd like), why its interesting, how it fulfills the design prompt, and feasibility for completion in five weeks. Unlike the brainstorm sheet, we would like these paragraphs written digitally

Jasper and I will review all of the downselected ideas and help you reflect on possible pursuits. We will also spend in-class time to share ideas and form teams (as necessary).

Points 2
Submitting Nothing

Due	For	Available from	Until
May 3	Everyone	-	-

Related Items
SpeedGrader™