Team D1: Is Mayonnaise an Instrument?

Harry Fernandez, Min Gun Kim, Tomas Vancura

Carnegie Mellon University

Use Case:

Getting started is difficult

- Lots of complex musical concepts
- DAW and synthesizer UIs are unintuitive and huge in scope
- What does anything actually do?

???

Experimentation is clunky

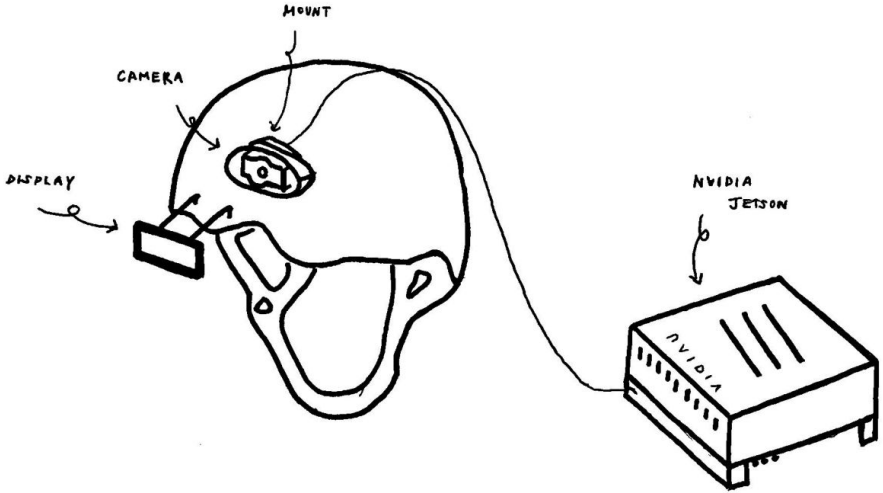
- Not a lot of room for play and creativity unless you really know what you're doing
- Almost all synthesizers use knobs, buttons, and faders
- Difficult to set up a quick workflow

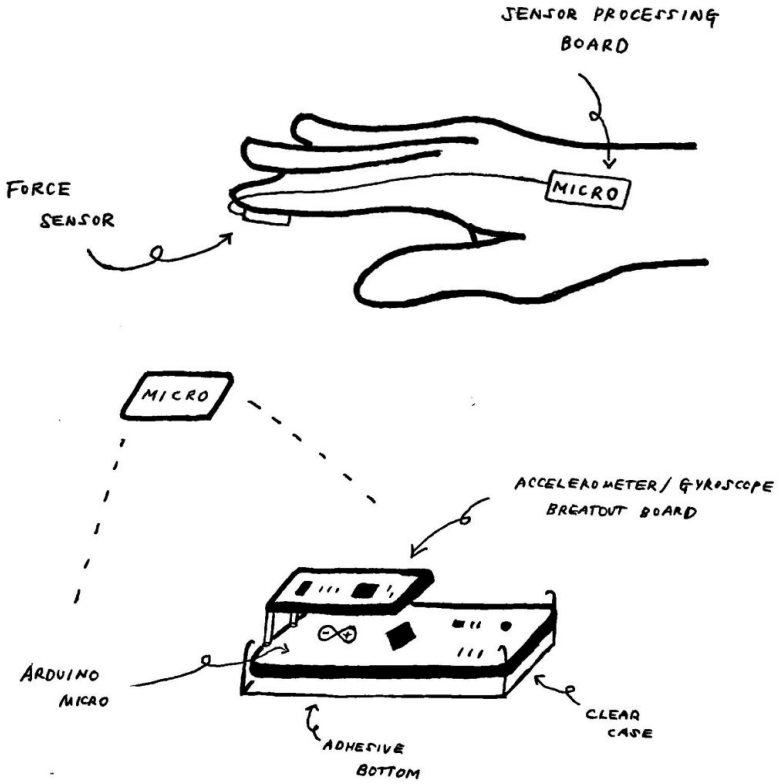
???

???

Carnegie Mellon University

Solution





Carnegie Mellon University

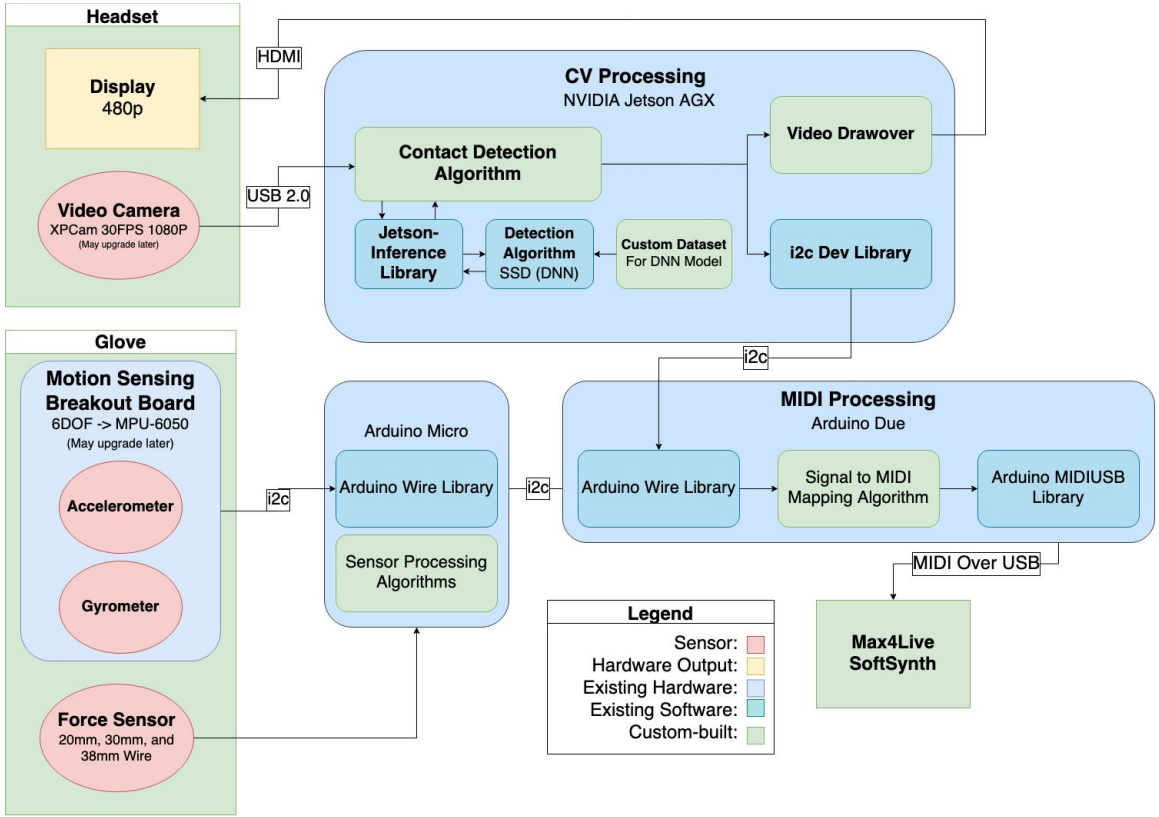
Use-Case Requirements

Requirement	Metric
Capture video	**60 FPS (min. 30 FPS)**
Identify objects in video	≥ 3 distinct objects, ≤ 1m range at 90% accuracy
Determine when user is touching object	**≤ 30[1] ms end-to-end latency,** ~ 99% accuracy
Determine position of held object	≥ 80% Movement detection accuracy
Translate/output positional, object data to MIDI	≤ 30 ms end-to-end latency
Visualize to a display	≤ 30 ms latency from picking up object to display

Carnegie Mellon University

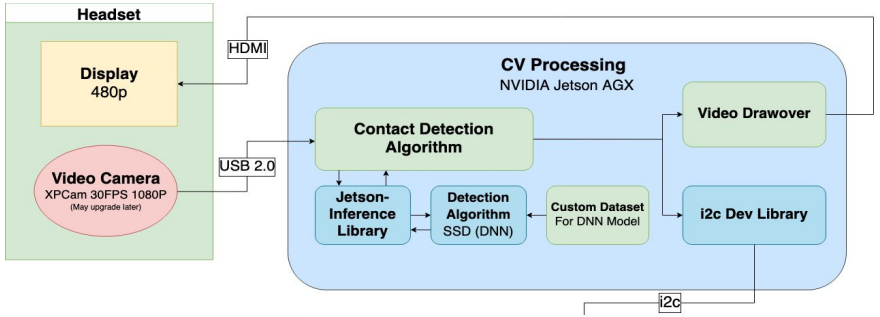
1. Xiaoyuan Gu, M. Dick, Z. Kurtisi, U. Noyer and L. Wolf, "Network-centric music performance: practice and experiments," in *IEEE Communications Magazine*, vol. 43, no. 6, pp. 86-93, June 2005, doi: 10.1109/MCOM.2005.1452835.

System

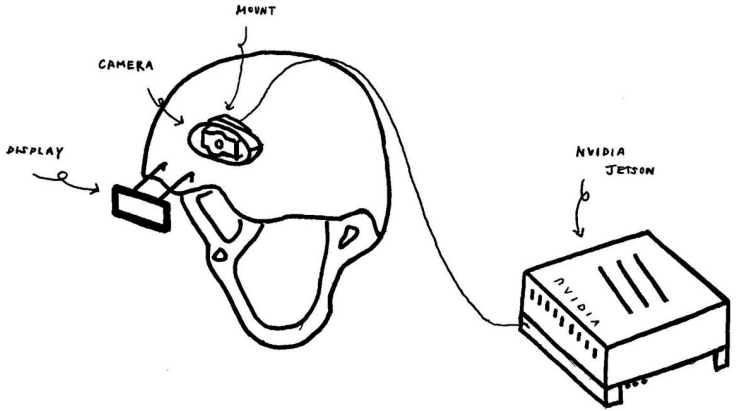


Carnegie Mellon University

Headset & Computer Vision

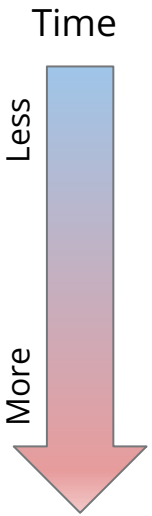


- Object detection and contact identification
  - What objects are in the camera's view?
  - Which object is the user (potentially) holding?
- Single Shot Detector - DNN
  - Detect position, size, type of recognized objects in scene
  - Research suggests SSD can hit our performance metrics on Jetson boards
- *Jetson-Inference*
  - DNN AI image library from NVIDIA
- Unit Test: Three distinct benchmark objects
  - Measure success of accuracy across trials
  - Measure latency in detection



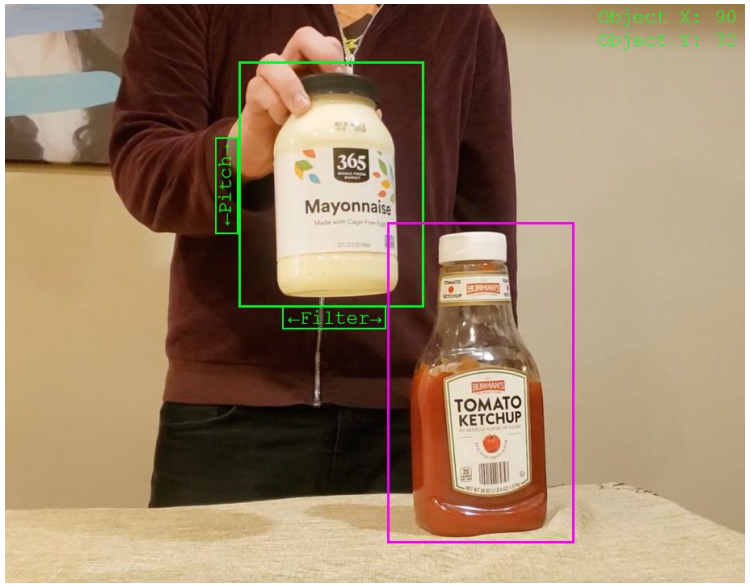
Object Detection Deep Learning Models

- Multiple options for the SSD model
  - Pre-trained model (via Jetson-Inference Library, trained on COCO dataset)
    - *Pros: Convenient*
    - *Cons: 91 object classes, many are irrelevant*
  - Train new model (on Open Images data set)
    - *Pros: Comprehensive training data (600 classes, 1.9M images, 16M bounding boxes)*
    - *Cons: Household object classes are very general*
  - Train new model (on custom dataset)
    - *Pros: Control over specific types of classes*
    - *Cons: Collecting and labeling hundreds of images*



Display and GUI

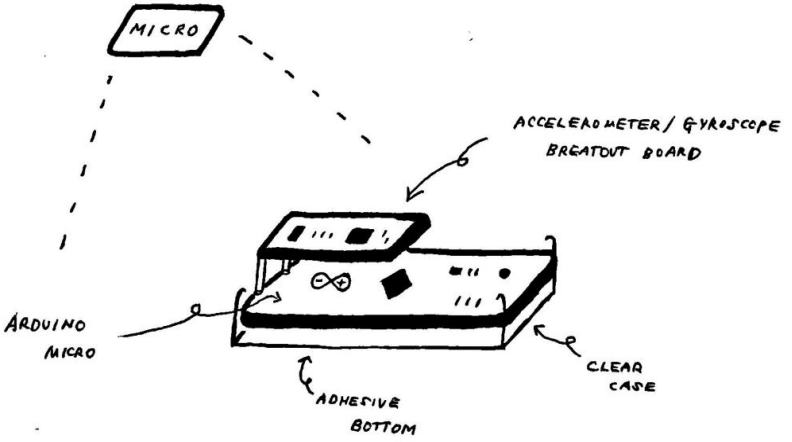
- Display will act as an AR HUD
- Visualizes the system's response to user actions
- Serves as a tutorial/entry point
  - Want to minimize learning curve
  - Will determine ease of use through user testing

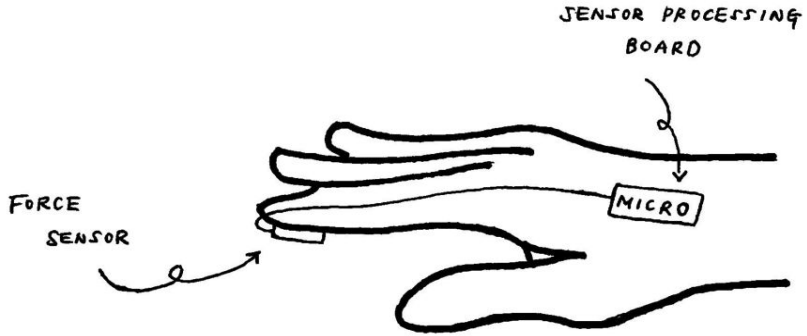


Carnegie Mellon University

Glove

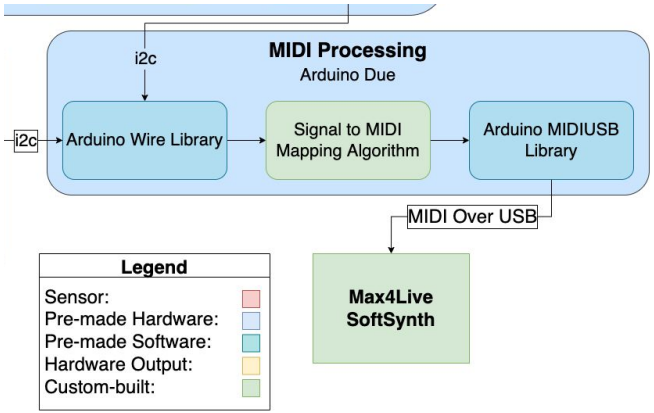
- Force Sensor
  - Lies flat, detects contact with objects
- Accelerometer and Gyrometer
  - Track movements
- Sensor Data Aggregation Board (Arduino Micro)
  - Packages data and sends it off to Due
- Unit Test
  - Force sensor sensitivity
  - Motion simulation
  - Ergonomics/usability





Communication Protocol & MIDI Mapping

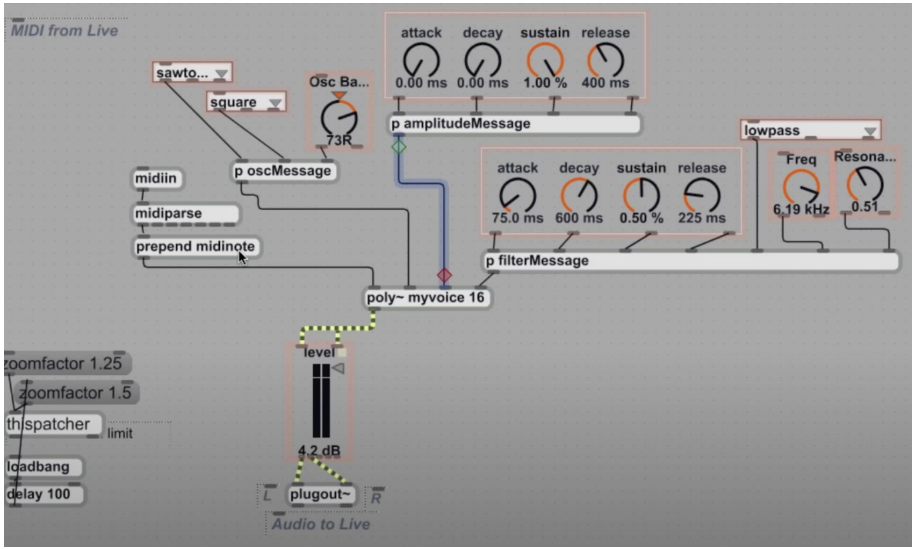
- Board-to-board communication over I2C (400kbps)
  - Jetson sends contact-detection packets
  - Arduino Micro sends sensor packets
- MIDI Mapping
  - MIDIUSB Library
  - Four main parameters for MVP
    - *Object Type*
    - *Contact*
    - *X Coordinate*
    - *Y Coordinate*
- Unit Test: I2C Latency with Unos



Carnegie Mellon University

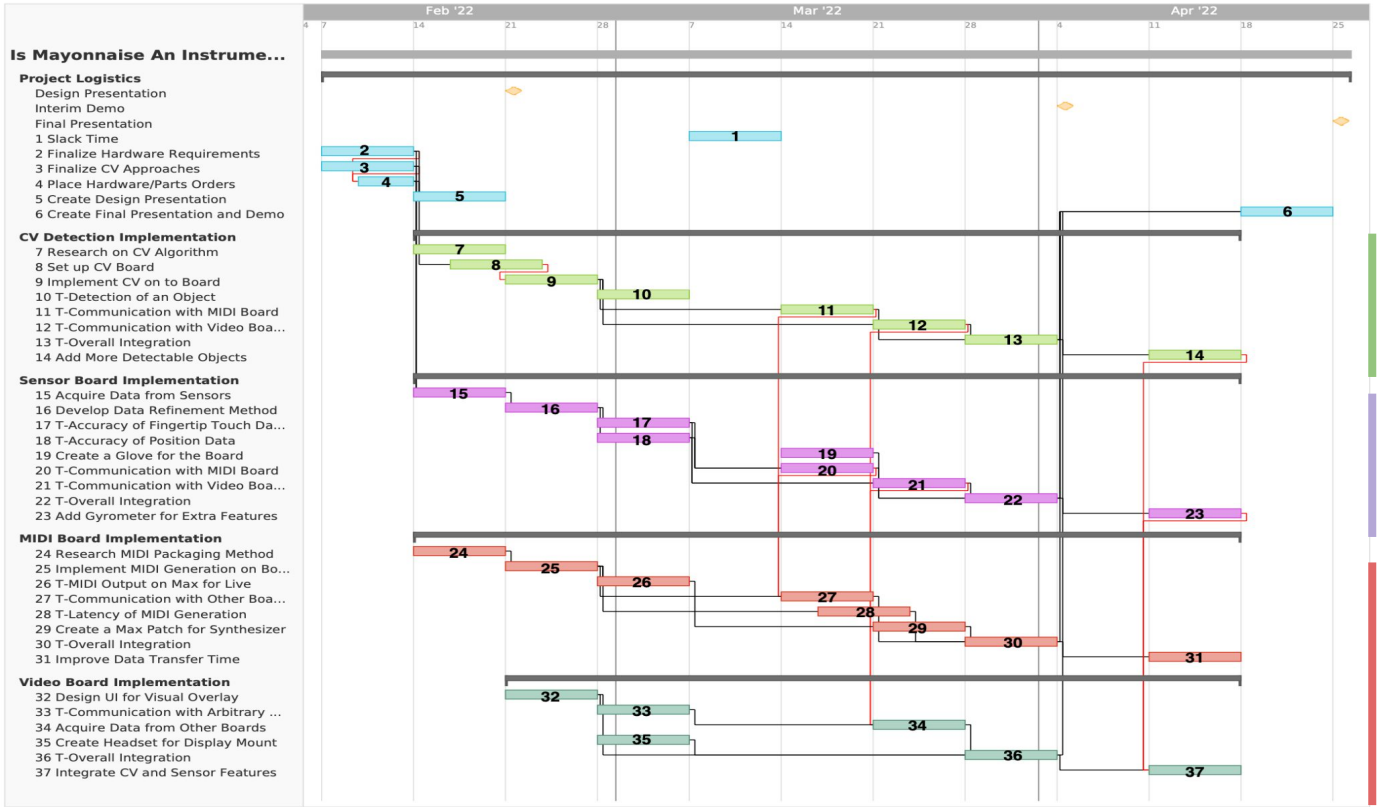
Integration

- Final product will be two wearables connected to a computer via USB
- Computer will be running a bespoke Max4Live Software Synthesizer



Carnegie Mellon University

Schedule



Tomas

Kevin

Harry

Carnegie Mellon University