Air Drums: Playing Drums Using Computer Vision

Carl Tolentino and Agatha Uy

CS 282 Mini Project (2019)

Why did we do this project?

- We want to make a **low-cost solution for aspiring drummers**, on how they could play the drums
- A possible solution is thru using what they already possibly have, which is a device that has a **webcam**

Comparison to Existing Solutions

COMPARISON TO EXISTING SOLUTIONS

Air Drums Using Sticks

LAZADA .COM.PH

Aerodrums Air Drumming Percussion Instrument

₱19,622.00

Free

National Capital Region (NCR)

Electronic Lightweight Rhythm Sticks Music Drum Sticks Air Drumstick...

₱587.02

~~₱1,079.00~~ -46% Free

China

COMPARISON TO EXISTING SOLUTIONS

Roll-Up Drum Pads

W758 Digital Portable 9 Pad Musical Instrument Electronic Roll-up Drum...

₱1,163.00

~~₱2,999.00~~ -61%

★★★★★ (152) National Capital...

Portable Electronic Roll Up Drum Pad Set 9 Silicon Pads Built-in Speakers...

₱2,060.00

~~₱3,000.00~~ -31%

★★★★★ (53) National Capital...

COMPARISON TO EXISTING SOLUTIONS

Electric Drum Sets

Medeli DD315 Electric Drumset 2018 Edition

₱9,499.00

~~₱13,999.00~~ -32%

Philippines

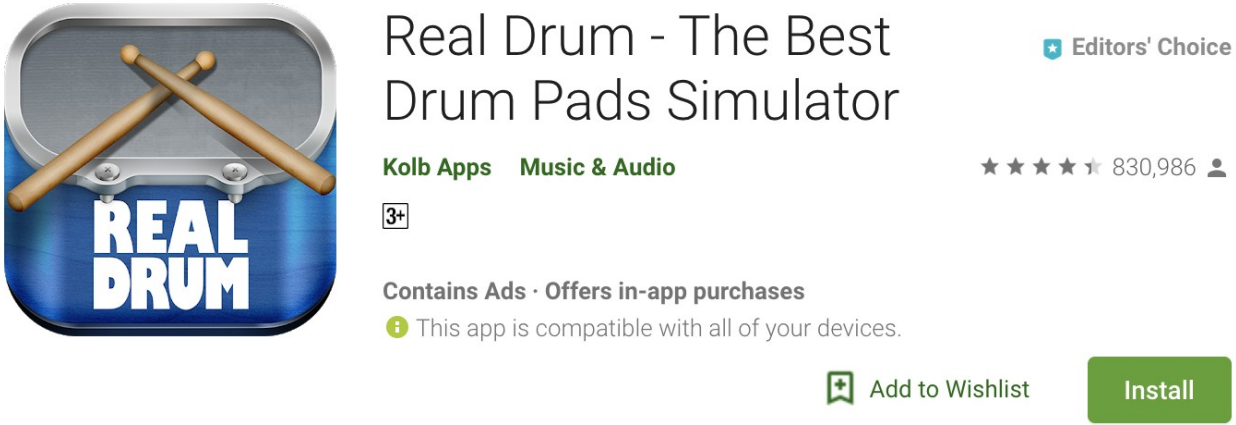
Alesis DM Lite Portable Electronic Drumset

₱22,999.00

~~₱25,000.00~~ -8%

Philippines

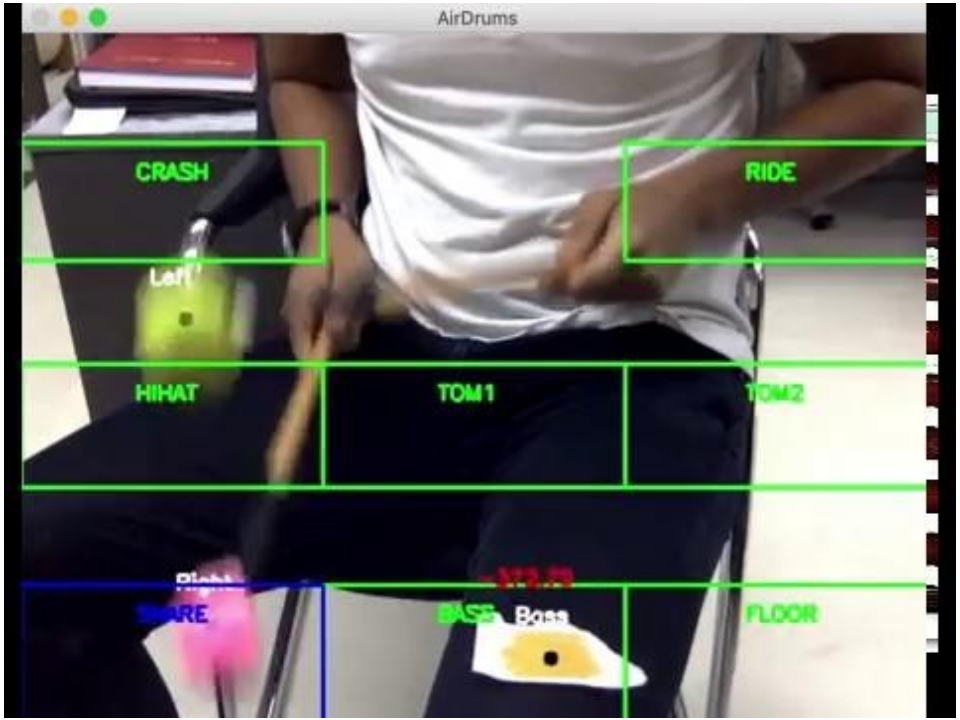
Phone Apps


REAL DRUM
Real Drum - The Best Drum Pads Simulator
Editors' Choice
Kolb Apps
Music & Audio
830,986
3+
Contains Ads · Offers in-app purchases
This app is compatible with all of your devices.
Add to Wishlist
Install

Overview of the Developed System

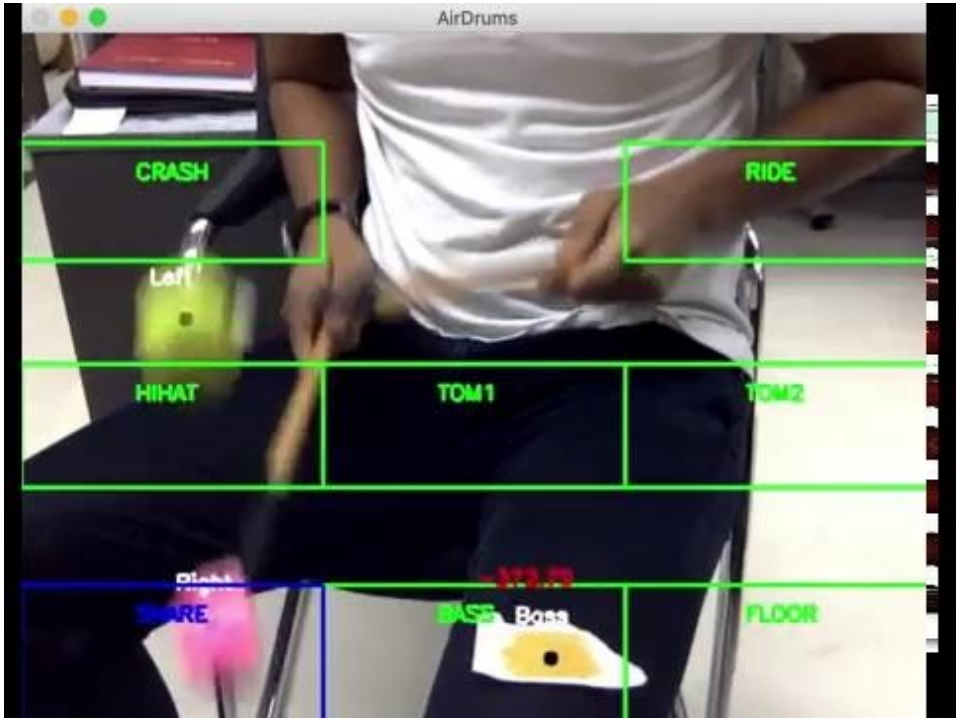
What You Need

1. Any two sticks, **two drum tips of different colors**, **one different colored patch** to stick to your knee
2. **Laptop with webcam**

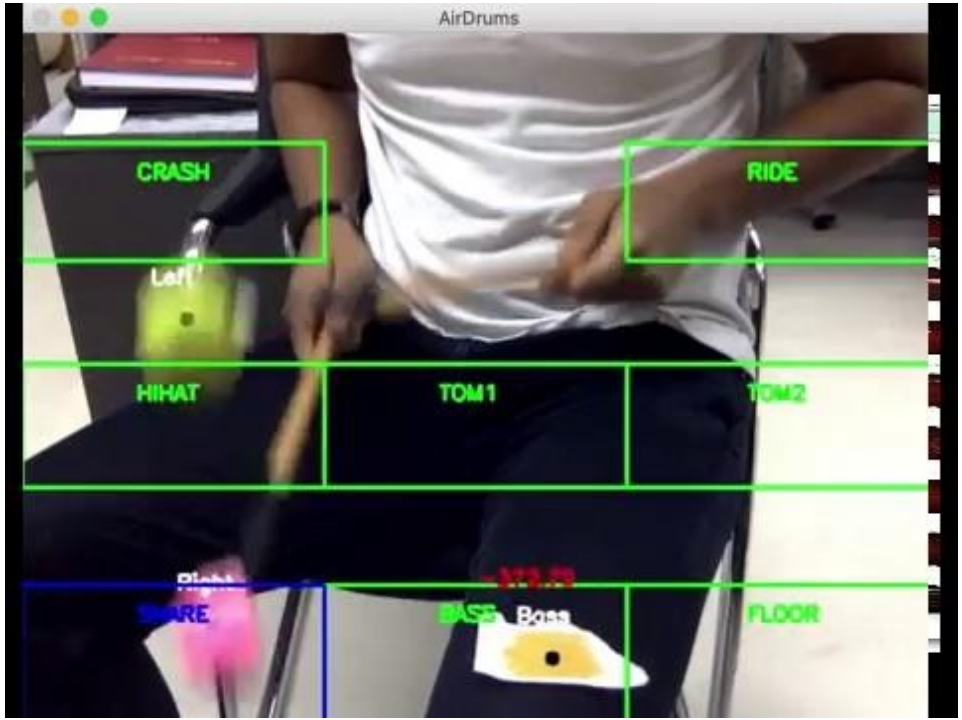


How to Use It

1. **Initial Calibration**
   - Click the drum tips: Left Stick tip, Right Stick tip, and/or Bass tip
2. **Hit the boxed areas** that define the drum pads to produce the appropriate drum sounds
3. Hit ESC to exit



Video of the Developed System



Methodology

Major Parts of the Methodology

Used OpenCV 3.4.5, and Python 3.6.8

1. Object Detection
2. Trigger Methods Explored

METHODOLOGY

#1 Object Detection

METHODOLOGY: #1 OBJECT DETECTION

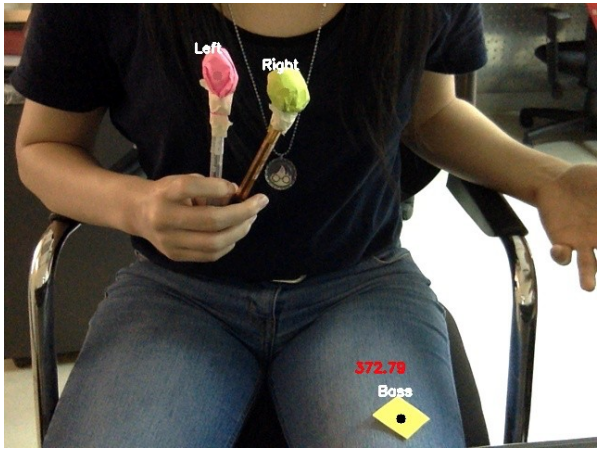
Blob Detection

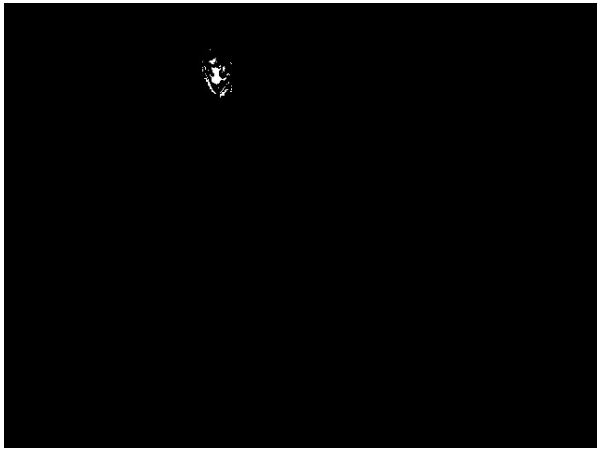
For each items = [Left Stick, Right Stick, Bass]:

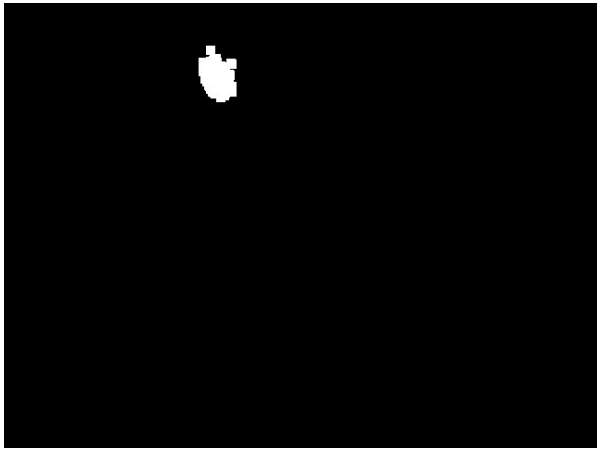
1. Do **blob detection given range of colors** from init. calibration
   a. Initial calibration gets 10x10 patch of RGB pixels, where top 10 and lowest 10 values are discarded
2. Given the extracted blobs, find **max contour and its centroid**
3. The extracted centroid is considered the item's **current point**

```python
def centroidDetection(self, img, item_num, img_counter):
    '''
        Detects the centroid of a given item
    '''

    item = self.DRUM_ITEMS[item_num]

    start = time.time()
    # Detect for blob
    maskLAB = cv2.inRange(img, self.min_rgb[item_num], self.max_rgb[item_num])
    kernel = np.ones((10,10),np.uint8)
    dilation = cv2.dilate(maskLAB,kernel,iterations = 1)
    im2, contours, hierarchy = cv2.findContours(dilation,cv2.RETR_TREE,cv2.CHAIN_APPROX_SIMPLE)
    height,width = dilation.shape[:2]
    c_areas = []

    for c in contours:
        c_area = cv2.contourArea(c)
        c_areas.append(c_area)

    # Find max contour, Check if no contours detected
    self.prev_pt_2[item_num, 0] = self.prev_pt[item_num, 0]
    self.prev_pt_2[item_num, 1] = self.prev_pt[item_num, 1]
    self.prev_pt[item_num, 0] = self.new_pt[item_num, 0]
    self.prev_pt[item_num, 1] = self.new_pt[item_num, 1]
    if (len(c_areas) != 0):
        max_c_area_index = c_areas.index(max(c_areas))
        logger.debug(max_c_area_index)
        # Calculate moment and center of contour
        M = cv2.moments(contours[max_c_area_index])
        cX = int(M["m10"] / M["m00"])
        cY = int(M["m01"] / M["m00"])
        self.new_pt[item_num, 0] = cX
        self.new_pt[item_num, 1] = cY
        end = time.time()
        logger.debug("[CENTROID DETECTION]: Seconds elapsed: {}".format(end-start))
        cv2.circle(img, (cX, cY), 5, self.blob_colors[item_num], -1)
        cv2.putText(img, item, (cX - 25, cY - 25),cv2.FONT_HERSHEY_SIMPLEX, 0.5, (255, 255, 255), 2)

    else:
        # No contours detected so set new pt to 0
        self.INIT_ITEM[item_num] = 0
        self.new_pt[item_num, 0] = 0
        self.new_pt[item_num, 1] = 0
```

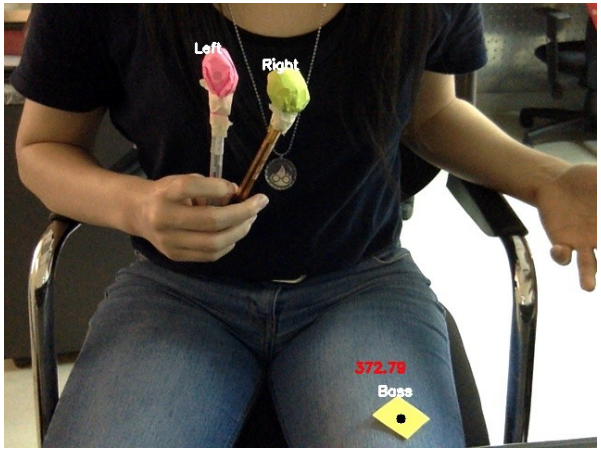
Blob Detection: Left Stick

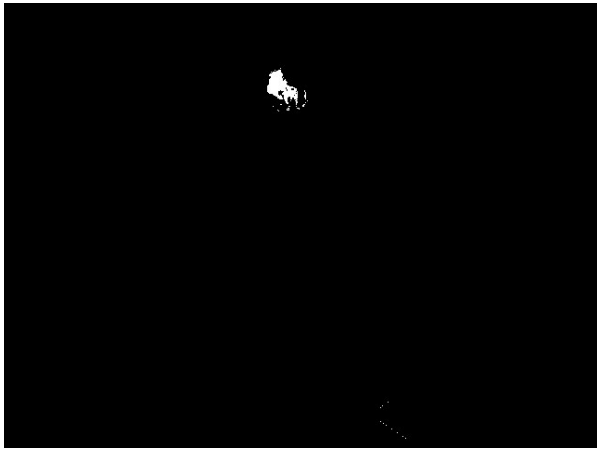


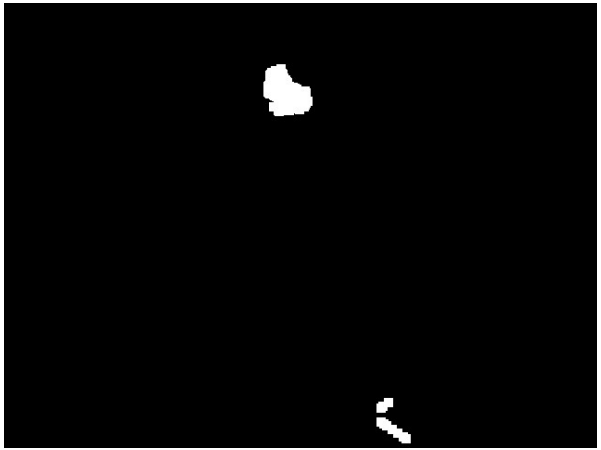




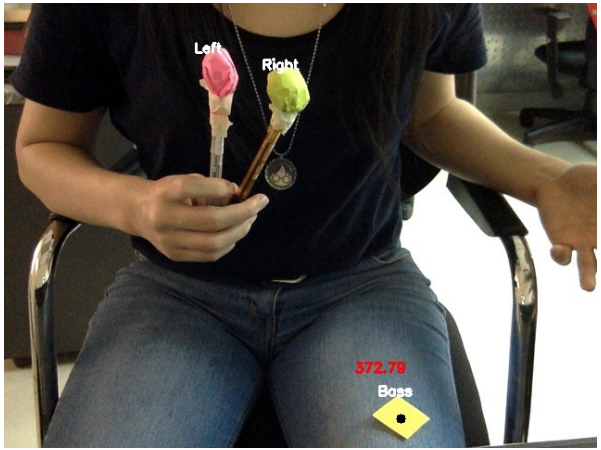
Blob Detection: Right Stick

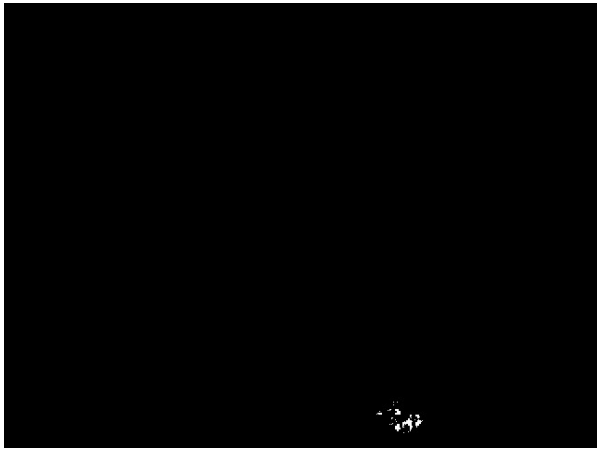


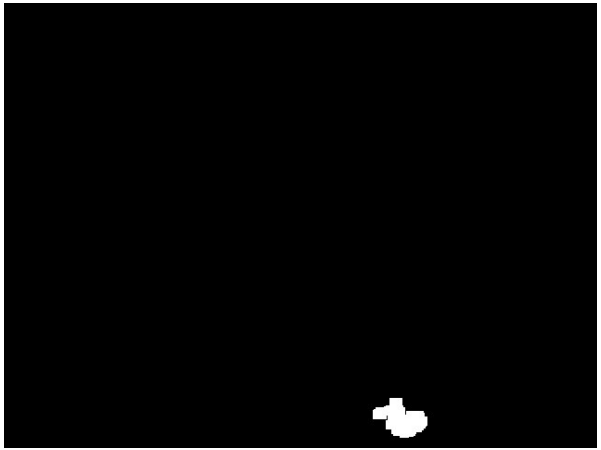




Blob Detection: Bass







METHODOLOGY

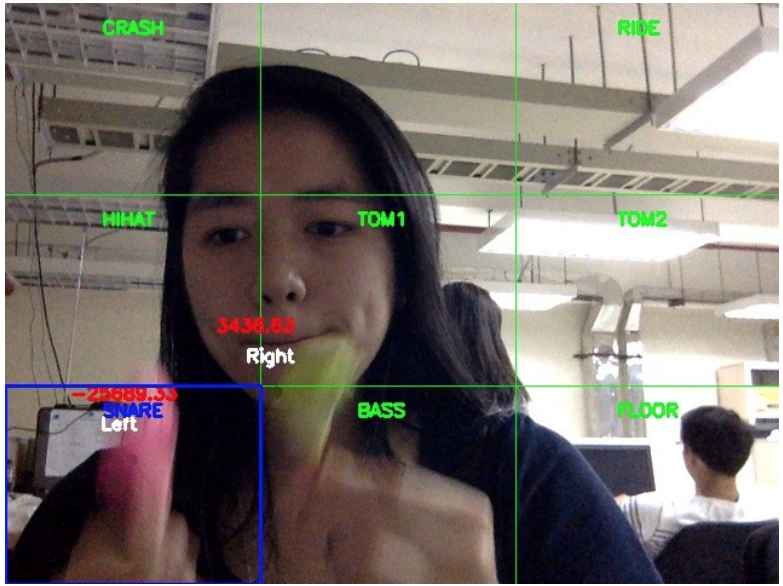
#2 Trigger Methods Explored

Two Explored Methods

1. By Acceleration Computation
2. By Points Comparison

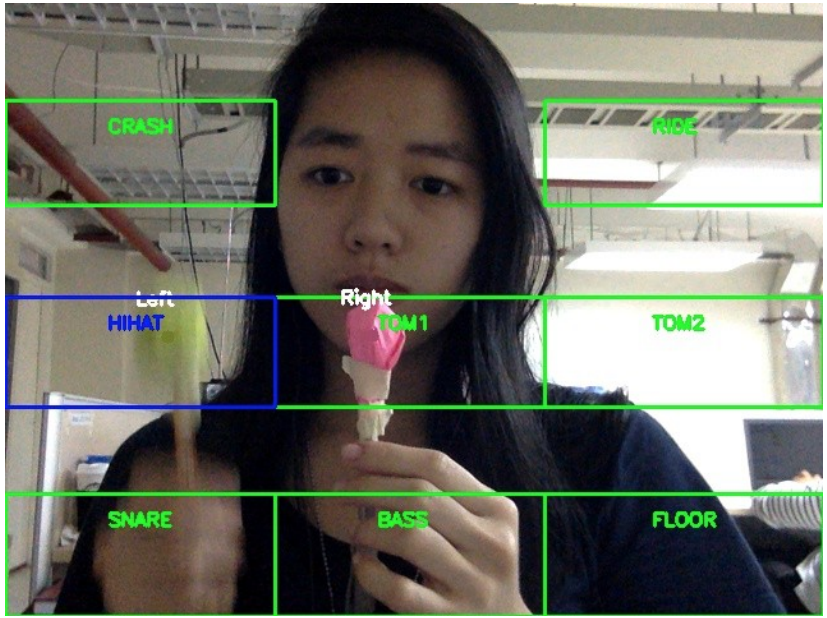
#1 By Acceleration Computation

1. For a given keypoint, **compute its current acceleration**
   a. Uses **Kalman filtering** for robustness against jitter
2. If the **acceleration is negative and downwards**, it means that a drum pad was hit
   a. Used a threshold for determining hit
3. If a drum pad was hit, **find location of present point for sound**



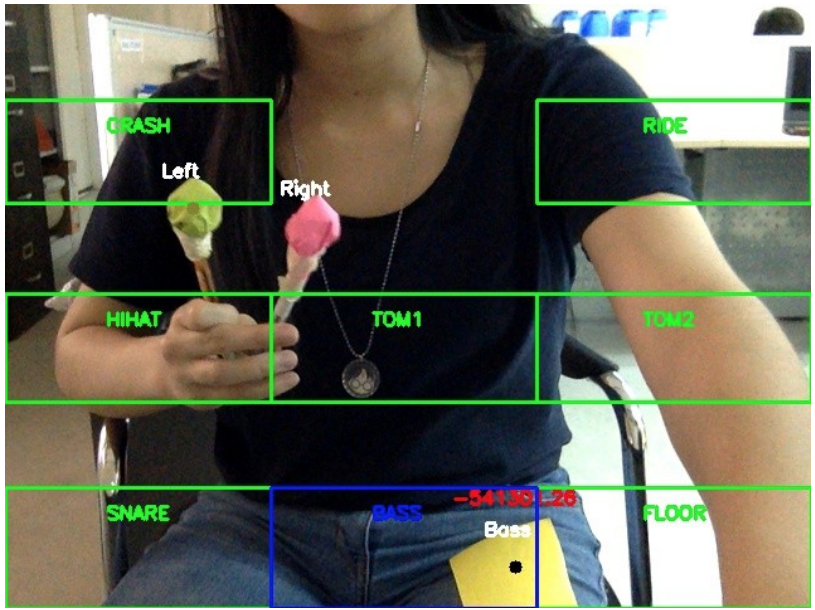
#2 By Points Comparison

1. For a given keypoint and its current location, **detect if it's currently in desired area**
2. If its **prev. point, is above or beside the area**, then it is considered a **hit**, and sound is played
   a. If prev. point is blank due to blurring, look at stored point from two frames back



Actual Usage

1. **By Acceleration Computation**
   a. Used for **Bass drum trigger detection**
2. **By Points Comparison**
   a. Used for **drum pad triggers**



Evaluation

Experimental Results

- Can detect approximately **513 hits per minute**
- Out of 100 hits:
  1. For the sticks: 4 false positive hits
     - **Precision: 96.15%, Recall: 100%**
  2. For the bass drum: 4 false negative hits
     - **Precision: 100%, Recall: 96.15%**
- **96% Reliability**

Comparison to Existing Work

Both don’t state their accuracies nor speed capacity.

- **Virtual Drum Simulator Using Computer Vision by Bering, S., and Famador, S. (2017) from UP Cebu**
  - Used hand detection based on color blob detection
  - Required users to wear a jacket
  - Didn’t release code
- **Drum master by Francisco Rojo (2012)**
  - Static ball/drum tip color
  - Still figuring out how to run his code

Over-all Improvements from Existing Work

- Can use any drum tips
  - Size-invariant drum tips, depending on color comparisons to current background
- Has bass drum
- Easier to setup
  - The others require fixed calibrations

Conclusion

Conclusion

- It is possible to create **a webcam-invariant, camera-based drumset,** that detects triggers thru **color-based blob detection**
- **Further calibrations** are needed for more robust detections
- Need to further **evaluate method via surveys**
- Convert to **releasable app**
- Explore use cases as: **usb controller for drum rhythm games** e.g. GTX Mania, and as a **MIDI controller**

Air Drums: Playing Drums Using Computer Vision

Carl Tolentino and Agatha Uy

CS 282 Mini Project (2019)