Air Drums: Playing Drums Using Computer Vision

Carl Timothy Tolentino
Electrical and Electronics Engineering Institute
University of the Philippines, Diliman
Quezon City, Philippines
carl.timothy.tolentino@eee.upd.edu.ph

Agatha Uy
Electrical and Electronics Engineering Institute
University of the Philippines, Diliman
Quezon City, Philippines
agatha.uy@eee.upd.edu.ph

***Abstract*—The cost of drum sets is an investment that most aspiring drummers would eventually need to shoulder in order to continue their craft. What this research wants to do is to fasten that initial introduction of drummers to the drumming experience without the costs, and also to allow for drummers to be able to practice, at least casually, without a full drum set. This thus allows the experience of drumming to a wider audience. The solution we explore is the development of a prototype virtual drum set that would only require users to have a laptop with a camera, and easily accessible markers representing the tips of drum sticks and knee movement, such as colored paper. OpenCV based on Python was used to implement this, and used the concept of color-based blob detection for detecting the markers. This prototype has also shown to have potential for further development as a released application, along with the possibility of use as a USB controller and MIDI controller.**

I. Introduction

The drums is the most popular percussive instrument in the music industry today. Many beginners who aspire to be a percussive musician starts out with learning how to play the drums. However, a typical drum set is usually expensive, takes a lot of space, and not easily transportable unlike other instruments such as the guitar or keyboard. Fig 1 shows the different components that consist of a standard drum set.

Fig. 1: The different components of a standard drum kit [1]

The goal of this study is to build a system that will enable aspiring drummers to be able to play and practice the drums on thin air, with the use of computer vision. The idea is to translate a video being captured of a user playing the virtual drums, considering realistic movements with an actual drum set, to an audio synthesis of appropriate drum samples in real-time. Making use of a laptop's built-in web camera, the project aspires to create an implementation that would make it easier for people lacking the funds or equipment to practice and learn the actual drums.

II. Related Work

Some implementations of portable drum systems employed the use of attachable sensors to the drum sticks to detect the direction and the velocity associated with the sticks. The gathered information is fed into the system to play the sound sample that corresponds to the relevant drum component. The advantage of these sensors is that they are more sensitive to velocity changes, and they also require less setup with relation to using a camera. However, these sensors can still remain inaccessible to the general public due to cost considerations. An implementation of this in a commercial setting is Freedrum [2], which costs around $235 for a complete kit. Their implementation made use of having sensors on the drumsticks, as well as the feet.

There are also works which implemented the virtual drums using a computer vision approach. The advantage of these systems is that a camera is very accessible. However, the algorithms may be complex and difficult to implement in real-time. The work done by Bering and Famador [3] locates and tracks the position of the hands in the video without the need for drum sticks. However, their work does not involve the movement of the feet which is important in recognizing the bass drum and the hi-hat control. Another work implemented by Brown et. al. [4] uses orange markers attached to the end of the drum sticks to be able to locate and track the movement of the drum sticks . Again, their work does not involve the movement of the feet. The work by Rojo [5] also used colored markers and implemented a dynamics computation for the volume setting of the synthesized drum samples. The most popular, and marketable implementation making use of computer vision makes use of a high speed camera and reflective balls attached to the end of drum sticks. This product is called the Aerodrums [6], and retails for $199.

III. Methodology

The prototype virtual drum system was developed using OpenCV 3.4.5 running on Python 3.6.8. The methodology

is divided into three major steps: (1) *Object Detection and Tracking*, (2) *Event Detection*, and (3) *Drum Sound Synthesis*.

A. Object Detection and Tracking

1) Keypoints: For what can be said to be a complete camera-based drum system, four keypoints are needed to be detected and tracked: the two tips of the drum sticks, and the two feet for bass drum and hi-hat control. These four points should be used to control the sound that the drum system will produce.

For this prototype, we limit the scope to tracking the two ends of the drum sticks, and the right knee of the player. The movement of the knee instead of the foot was chosen for detection in order to allow a user of the system to be near the camera of his/her laptop. The removal of the detection of the left foot was done since its movement is usually used for the open and close state of the hi-hat. The implementation of this would be better suited after the perfection of accuracies of knee or foot movement detection, as this would lead to more accurate states of the hi-hat. It should be noted that the hi-hat is usually closed for most drum sequences thus making this design choice acceptable. Figure 2 shows an example setup and camera view given the desired keypoints.

Fig. 2: The setup of a user playing the virtual drum set.

2) Blob Detection: The prototyped system detects the three keypoints through blob detection based on color. As such, this method assumes that there are three different color ranges for the three keypoints, and that there are three differently colored markers attached to the ends of the two sticks and the right knee of a user. It should be emphasized that these colors should be different from each other, and that the camera view should not contain items of the same color bigger than the sizes of the markers as seen by the camera.

For each frame captured by a camera, thresholding is done for each keypoint to be identified. Dilation is done to each set of extracted pixels in order to make the extracted pixels be more blob-like. The largest blob for each threshold is determined to contain desired keypoint, and then the center of this blob is computed in order to get the position of the keypoint. Figure 3 shows an example frame where blob detection was implemented.

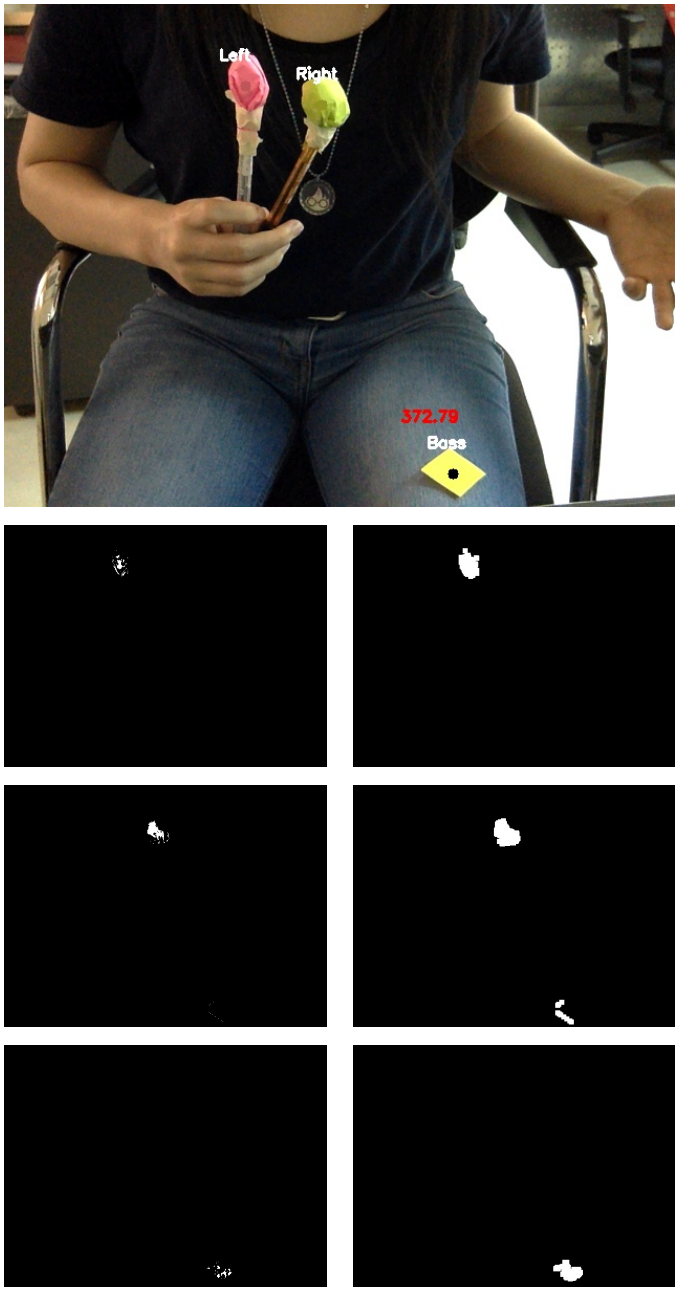

Fig. 3: Extracted blobs for the three keypoints (from top to bottom: left stick tip, right stick tip, and right knee area). The first image shows the original photo along with the overlayed computed centers of the keypoints, the left column shows the thresholds based on color extraction, and the second column shows the dilated thresholds from those on the left column.

B. Event Detection

Two methods were explored for detecting the event of striking a drum pad on the virtual drum set: (1) *By Acceleration Computation*, and (2) *By Points Comparison*.

1) By Acceleration Computation: With this explored method, we determine event of a drum pad being hit by calculating the dynamics of the points tracked by our system. Specifically, we calculate the position $\mathbf{p}(n)$, velocity $v(n)$, and acceleration $a(n)$ of the three keypoints. The calculation of the dynamics is given by the following equations.

$$\mathbf{p}(n) = \begin{bmatrix} x(n) \\ y(n) \end{bmatrix} \tag{1}$$

$$v(n) = \frac{\|\mathbf{p}(n) - \mathbf{p}(n-1)\|}{\Delta t} \tag{2}$$

$$a(n) = \frac{v(n) - v(n-1)}{\Delta t} \tag{3}$$

where Δt is the time step corresponding to the number of frames per second (FPS) captured by the camera. For simplicity, we set this parameter to be a constant value since the time step between frames is varying due to the processing done in between frames captured.

Since the points detected and tracked by the camera contain some measurement noise, we implement a Kalman filter for the calculation of the dynamics. Through the use of this filter, instead of obtaining the current position $\mathbf{p}(n)$ directly through computations made with the next frame captured by the camera, we make an estimated prediction of the the current position $\hat{\mathbf{p}}(n)$ using past data. The estimate of this position is given by the following equation:

$$\hat{\mathbf{p}}(n) = K(n) \cdot \mathbf{p}(n) + (1 - K(n)) \cdot \hat{\mathbf{p}}(n-1) \tag{4}$$

where $K(n)$ is the Kalman gain at time n. We use this estimated position instead of the measured position in calculating the dynamics of the points.

Once the dynamics have been calculated, we detect that a drum pad trigger has occurred when a certain point has achieved the following criteria: (1) its acceleration has exceeded a certain threshold, (2) its direction of travel is towards the downward direction, and (3) it has not been triggered for a certain number of time steps. For the first criterion, we assume that the virtual drum set has been hit when the point has been driven to a complete stop, which is characterized by a large negative acceleration. The direction of travel must be towards the downward direction for it to be characterized as an appropriate strike to the virtual drum set. This is easily determined by comparing the estimated position from the previous position. Lastly, to avoid the triggering of events for consecutive frames, we trigger events only after a certain number of time steps.

2) By Points Comparison: In this method, the event of a drum pad being hit is determined by the comparison of a keypoint's current position with its previous position. For a keypoint's given $\mathbf{p}(n)$, if it is detected to be inside the bounding box of a drum pad pre-defined with the system, the point from the previous frame $\mathbf{p}(n-1)$ is checked as to whether it is above or beside a bounding box. This event would then trigger a hit. In the case of the keypoint not being detected in the previous frame due to blurring caused by high speeds, the point $\mathbf{p}(n-2)$ is then used for comparison.

C. Drum Sound Synthesis

There are eight drum components included in our virtual drum set: (1) Crash Cymbal, (2) Ride Cymbal, (3) Hi-hat, (4) Left Tom, (5) Right Tom, (6) Snare, (7), Bass, and (8) Floor Tom. The positioning of these components are shown in Figure4.

After the detection of a drum pad hit, the corresponding drum sound is generated based on the computed location of the hit given the pre-defined bounding boxes.

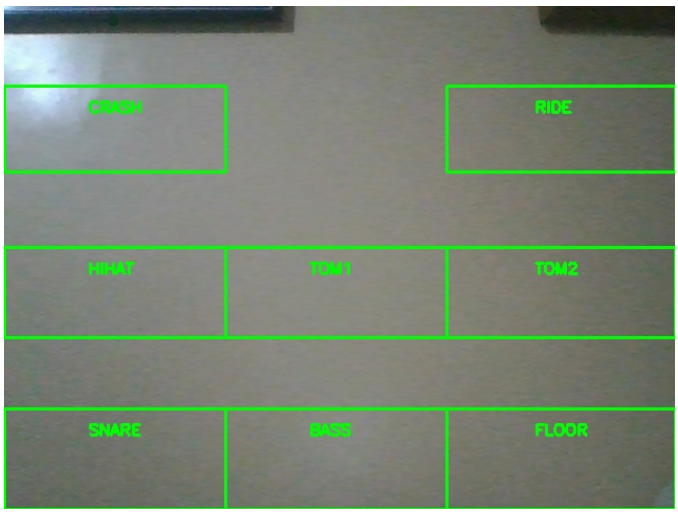

Fig. 4: Prototype virtual drum positioning for each frame.

D. System Overview

1) Computer Vision Algorithms Used: For the algorithms used, the over-all system used blob detection by color for detecting the three desired keypoints since it is a simple and efficient technique useful for real-time applications. For event detection, we detect an event *By Points Comparison* for the two drum sticks, while for the right knee, we detect an event *By Acceleration Computation*. The reasoning behind this is that detecting *By Points Comparison* tends to be more robust since the requirement is just to detect where the keypoints for the sticks were in the previous frames, and then compare where they are in the current frame. And as for the right knee, we wanted a user to not be restricted to keeping his/her knee positioned at a certain location to comply with a bounding box. Through using *By Acceleration Comparison*, a user is then free to just move the knee up and downwards.

2) Modes of Play: Two modes of play were included in our system. The user is asked to play the virtual drum set either by: (1) using two drum sticks, or (2) using two drum sticks with the right knee for bass drum control. For the first mode of play, the user can strike any of the drum pad in Figure 4 including the bass drum. For the second mode of play, the user can strike any of the drum pads with the sticks except for the bass drum, but the user can play the bass drum by moving

the right knee similar to the movement for an actual drum set. Figure 5 shows the interface of a user playing the first mode of the system, while Figure 6 shows the second mode of the system.

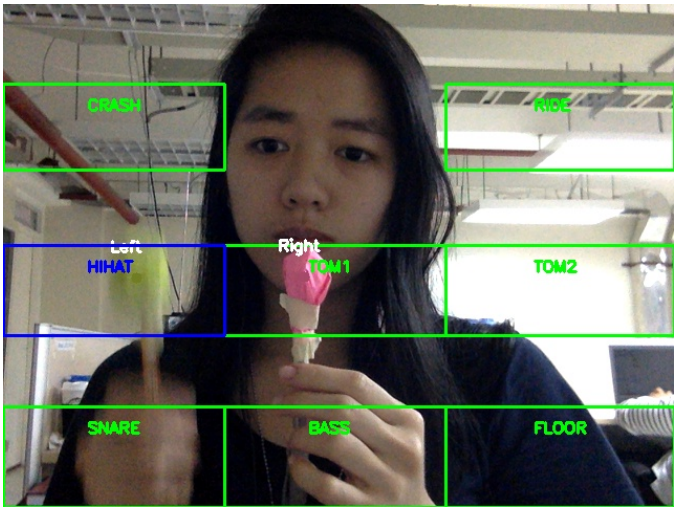

Fig. 5: Example display seen by a user playing the first mode of our virtual drum system

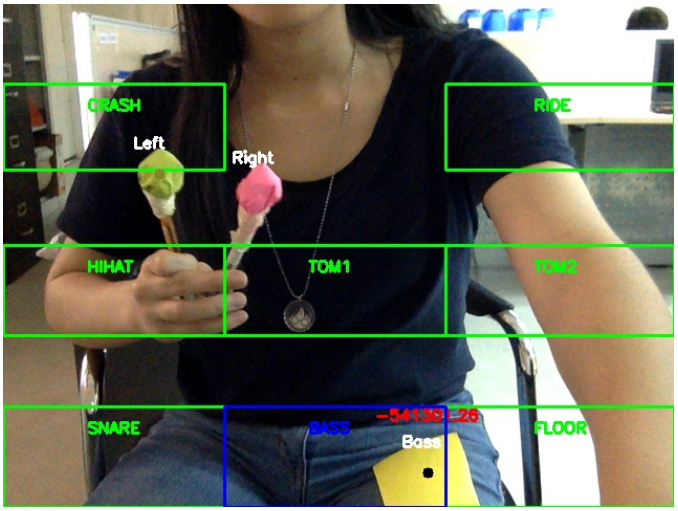

Fig. 6: Example display seen by a user playing the second mode of our virtual drum system

3) Initial Calibration: Before the user can start playing the virtual drum set, he/she is asked to calibrate the system for the colored markers attached to the end of the drum sticks and on top or near the right knee. The user is asked to click on the colored markers on the screen as seen by the camera, so that the system can determine the colors associated with the keypoints. This allows for the user to use any markers he/she desires, as long as the colors of these keypoints are distinct from each other, and are larger as seen by the camera as compared to an object of the same color in the background. This size comparison should also consider the shape of the marker, where a suitable marker should be something that would have its center as the centroid given computation of moments, assuming the center of that item is the center considered by the user. In other words, circular, or square-ish objects are recommended as markers.

4) Actual Play: After the *Initial Calibration*, the user is free to play the virtual drum set until he/she closes the application's window by pressing the *ESC* key.

IV. Results and Discussions

The virtual drum system was tested by using colored paper as the markers. Paper was crumpled to be the tips of the sticks to be used, where both pens and actual drum sticks were used in testing the system. A piece of colored paper was used to be the marker for the right knee as well. The motivation behind using the colored paper and pens as sticks was that we wanted our system to be robust enough to work on objects that people would have easy access to and in a low-cost manner. Other objects such as bottle caps, hair clips, and a drinking straw were also recognized as valid markers by the system due to its use of color detection. The application was also tested to run invariant of the laptop it is used on, as it was tested on a Macbook Pro and a Xiaomi laptop with the same application calibrations without any noticeable difference in terms of usability.

For the test bed for the evaluation of the virtual drum set, the two markers used for the drum sticks were of the color yellow green and pink. These markers were attached to the ends of actual drum sticks. The marker used for the knee is of the color orange. The laptop used for the tests was a Macbook Pro 2015 (A1502), running on an Intel Core i5 processor (5287U).

A. Virtual Drum System Performance

Through continuously hitting the snare drum as fast as possible in a span of one minute, it has been observed that the system could detect approximately 513 hits per minute.

The precision and recall of the system has also been tested through two separate experiments for the drum pads and the snare drum. Out of 100 hits for each, the drum pads had 4 false positive hits, thus leading to a precision of 96.15% and a recall of 100%. For the bass drums, 4 false negative hits were obtained, thus leading to a precision of 100% and a recall of 96.15%.

B. Observations on the Algorithms Used

1) Blob Detection: It has been observed that this method does prove to be quite robust for a variety of situations. However it has been observed that a disadvantage of using this method is its sensitivity to lighting conditions with regards to the initial calibration and the actual use. What has been observed that makes the system fail is the presence of bright white light that thus renders the markers as mostly white as seen by the camera. However, it has also been noticed that a way to bypass this limitation was when calibrating the camera, the markers should be close to the laptop's camera. During tests, the system has not failed to detect the keypoints if calibrated this way. A possible explanation of this is that in

the initial calibration, the actual range of colors of the marker given its hue is what's acquired in the patch taken during calibration, and not mostly the whites from the light, nor the blacks from the folds of the paper markers.

2) By Acceleration Computation: It has been observed that further improvement of this algorithm is needed in order to make the detection of knee movement more robust, as it has been noticeable that the triggering of the bass drum can be inconsistent, especially from person to person.

3) By Points Comparison: Through testing the system it has been observed that the case of just checking the history of the past two frames for a current frame is enough for calculating whether a drum hit was done. However, to make the system more robust, it would be better to make the algorithm more flexible in terms of the number of past frames it could look into, especially with regards to differing hardware and CPU speeds.

C. Comparison of Results with Existing Work

We compare our work to the work of Rojo (2012) [5] and Bering and Famador (2017) [3]. It should be noted that both didn't release results on the reliability of their systems, along with a result along the lines of hits per minute that their system could do. Both used color-based blob detection. Our work is similar to Rojo's in the way that he used a red and blue colored ball attached to the tips of drum sticks. We can say that our system has improved compared to his method, as he used fixed calibrations and objects more easily identified by a camera as a blob, whereas our system is more flexible with regards to this. Bering and Famador's work relied on detecting the hands as blobs, or using skin tone. Their work is thus limited to the user needed to wear long sleeves, and to not include the face. We have also implemented bass drum using the right leg which both have not implemented. We can also say that as a prototype, our system is easier to setup due to the requirements one needs to play the system, where our system allows for size-invariant markers for the drum tips, of course with respect to color comparisons to the background, along with flexible initial calibrations.

D. User Surveys

We evaluate our system qualitatively by conducting a survey taken from 20 respondents. The respondents range from beginners who have never played the drums to intermediate-level drummers. We asked them to try the two modes of play of the system, and then asked them to answer our survey. The questions in this survey included the following items:

1) *How skilled would you rate yourself as a drummer?*
2) *How usable is the prototype system with just two sticks for casual drumming?*
3) *How usable is the prototype system with two sticks and bass drum control for casual drumming?*
4) *Do you think further developments to this project is something ok to do? Aside from improvements to the algorithms, future developments could include compatibility as a USB device input and MIDI controller.*
5) *Do you think that the development of a mobile app for this would make sense?*
6) *Would you use a released version of the application?*

For item numbers 1 to 3, the respondents were asked to give a rating from a scale of 1 to 5, while for item numbers 4 to 6, the respondents were simply asked a *yes* or *no* answer. The following results from the survey were obtained.

Figure 7 shows how the users evaluated themselves as drummers, with "1" being how they have never played the drums, and with "5" being they are expert drummers. We can say that our respondents are divided into 40% who have no experience with drumming, and 60% who have experience with drumming.

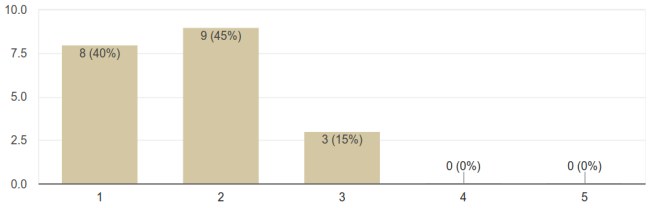

Fig. 7: Answers from the respondents on item number 1 (*How skilled would you rate yourself as a drummer?*)

Figure 8 shows how the users rated the usability of the system with just two sticks, with "1" being not usable at all, and with "5" being very usable. It can be said that the results gathered were positive for the prototype system. It has been observed that the problems of users who had experience with drumming with the system were the cases where they were hitting the drums from outside the frame of the camera, and the speeds were fast enough such that just checking the previous two points as past frames wasn't enough. For the inexperienced drummers, the problems they encountered were how they weren't that used to the movements of playing the drums, such that they lacked precision of handling. This led to wild gesticulations with the sticks that led to them not hitting the inside of the bounding box at all, along with the sticks being frequently outside of the frames. The problem of not hitting the inside of the bounding boxes were also encountered by the experienced drummers, however this is less frequently so. Their comments on this issue agree with our insights that the position and sizes of the drum pads should be made freely adjustable by a user according to preference.

Figure 9 shows how the users rated the usability of the system with the two sticks and the bass drum. In comparison with Figure 8 which only uses two sticks, it is can be said that the problem with this is the need to further refine the method used for detecting right knee movement.

Table I summarizes the results of the survey for question numbers 1 to 3.

Table II shows the answers of the users for question numbers 4 to 6. All of our respondents responded positively on how the system is something that should be developed further for what it is trying to achieve. 3 out of our 20 respondents said

TABLE I: Answers from the 20 respondents on item numbers 1 to 3

	Rating					
Item	1	2	3	4	5	**Mean**
1	8	9	3	0	0	**1.75**
2	0	1	10	6	3	**3.55**
3	0	6	8	4	2	**3.10**

TABLE II: Answers from the 20 respondents on item numbers 4 to 6

Item	Yes	No
4	**20**	0
5	**17**	3
6	**18**	2

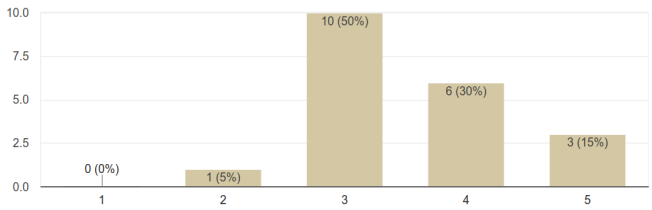

Fig. 8: Answers from the respondents on item number 2 (*How usable is the prototype system with just two sticks for casual drumming?*)

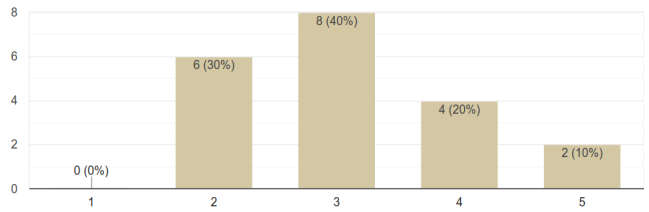

Fig. 9: Answers from the respondents on item number 3 (*How usable is the prototype system with two sticks and bass drum control for casual drumming?*)

that the system should probably not be made into a mobile application, with reasons of how it would probably be not useful and cumbersome to setup and position for a potential user. And only 2 out of our 20 respondents said that they would not be using a released application of this system, with reasons of that they are not really interested in playing the drums, and a possibly negative experience, which is from our observation of how the system can get unusable if the user is not coordinated enough or precise in using the drum sticks. Over-all, we can say that we have received positive reviews for our system.

V. Conclusions

Based on our tests, we can say that we have achieved our goal of developing a prototype system for air drums, usable by beginner drummers and at a very low monetary cost.

The use of color based detection for real-time detection is simplistic, but due to its speed it is thus viable for the goal of gaining the fastest hits per minute possible in real time. We are able to achieve an estimate 513 hits per minute for the triggering of the drum pads. The conversion of the current code base to C++ is something to be explored in making the code run faster, and thus increase our hits per minute.

Further refinement of the algorithms used for whether a drum pad was hit or not, along with the knee movement detection is something that is also needed to be done. As our tests show, we need to further refine them to achieve 100% usability even if for just casual drumming, as reliability of musical instruments is the most important factor for playing musical instruments. The achievement of this would also allow for the inclusion of hi-hat control which would then make the standard drum kit experience complete.

Improvements to the user interface and user experience of the application is also something to be done for future work as the virtual drum system gets out of the prototype stage. Further development on using it as a USB controller and MIDI controller, would depend on how accurate the system as it is further developed could be. However, those purposes are indeed use cases that have potential for this system.

References

[1] ”Janne”, “Standard 5-Piece Drum Kit Parts,” 2018. [Online]. Available: https://drumfortress.com/standard-5-piece-drum-kit-parts/

[2] “Freedrum: The Drumkit That Fits in Your Pocket,” 2017. [Online]. Available: https://www.kickstarter.com/projects/freedrum/freedrum-the-drumkit-that-fits-in-your-pocket

[3] S. R. F. Bering and S. M. W. Famador, “Virtual Drum Simulator Using Computer Vision,” pp. 370–375, 2017.

[4] B. Brown, D. Quenneville, and M. Shashoua, “CCS453 Computer Vision Spring 2016 Final Project: Air-drumming through Video Object-detection,” 2016. [Online]. Available: http://www.cs.middlebury.edu/{~}dquenneville/cs453/final-project/

[5] F. Rojo, “Air Drums Using OpenCV and Python,” 2012. [Online]. Available: https://www.youtube.com/watch?v=MAnWwxTjL3k

[6] “Aerodrums,” 2019. [Online]. Available: https://aerodrums.com/home/