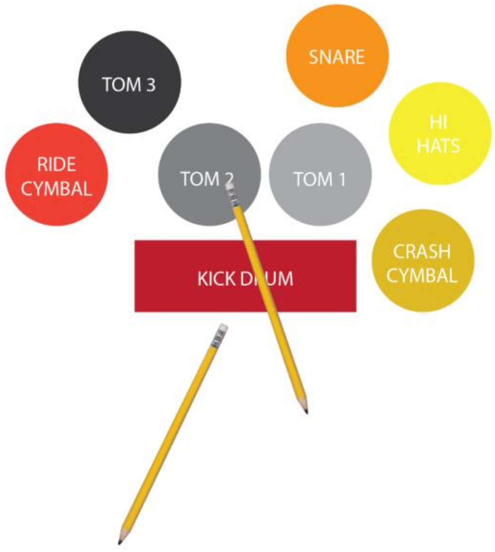
Paper Drums: Using Color Detection to Play Sounds

Omar Waheed, Rehan Rasool

Faculty Supervisor: Professor John Magee

Department of Math & Computer Science, Clark University



Abstract:

We created a computer vision system to detect colored objects on a paper and produce sounds associated with each object. The product that we imitated through our project is a drum kit. We used different colored objects to depict different parts of a drum: brown (hit hat), red (snare), green (tom 1), blue (tom 2), purple (bass tom). We made a masking strategy when detecting the colored objects to limit computations to a small space and used HSV color space to improve detection. The area covered of an object was used, with a threshold value, to decide whether to play a particular sound or not. To make the program more reliable when the user hits the objects, we added another camera on a low elevation, with similar logic.

Motivation:

Drum kits are expensive and not readily available to all people who are interested in playing. The only option online is to play drums using a keyboard. This project will help people play drum interactively and in a more realistic way, rather than using their keyboard. The cost of making a paper drum is negligible, the users will only be needing a colored printed paper, which distinguishes each drum part from the others. The user will also need a webcam connected to the computer. 2 webcams should be attached for better, more reliable results; one higher than the other. We can use this concept of colored paper products to replicate expensive objects for poor people, easy learning and personal use.

Goal:

Our goal was to use colored detection and user interaction to replicate a musical instrument. It was a difficult problem because there has not been much research done in the field of sound processing with computer vision. Hence, we had to come up with solutions and workarounds to many problems that we faced during the project.

Related work:

There is one related project [1] done before but it uses on-screen areas to play sounds, which does not give a realistic feeling of playing a musical instrument. It plays a certain sound when a sectioned area on the screen is filled by user's motion.

Methods:

We started off by analyzing how a drum kit works and figured out a way to create an accurate representation of it on a paper. Our implementation logic comprised of the following steps:

1. Detect initial colored circles on the paper using HSV thresholding to create a mask

2. Save the initial colored area for all circles using our mask
3. Run the program and wait for a circle to be covered by the user object
4. If a circle is covered by a certain threshold (e.g. 50%), play its corresponding sound

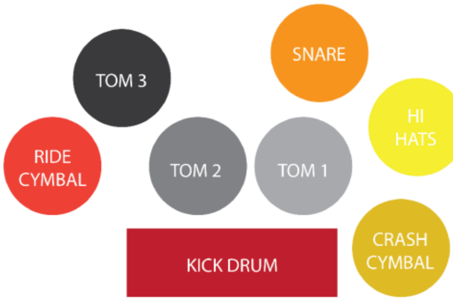


Figure shows initial layout of the drum we came up with.

We added another camera on a low elevation to make the results better and to avoid the sound playing when the user is covering the circle in air. Through the low-elevation camera, we ensured that the sound is only played when the user touches the circle.

We encountered some major issues during the project. They are described below with the solutions:

Inaccurate colored tracking under different lighting: We changed the algorithm to use HSV space instead of RGB. Moreover, we picked ROI (regions of interest) from the camera output for each color to reduce the computations significantly and ignore any similar colors that may have been in the environment. The user has the ability to create a sub-frame from the camera output when picking each colored object and limit the processing for that color to that subspace.

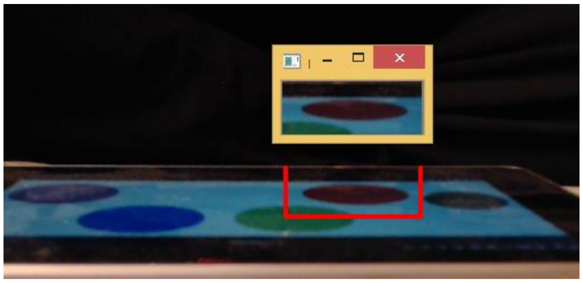
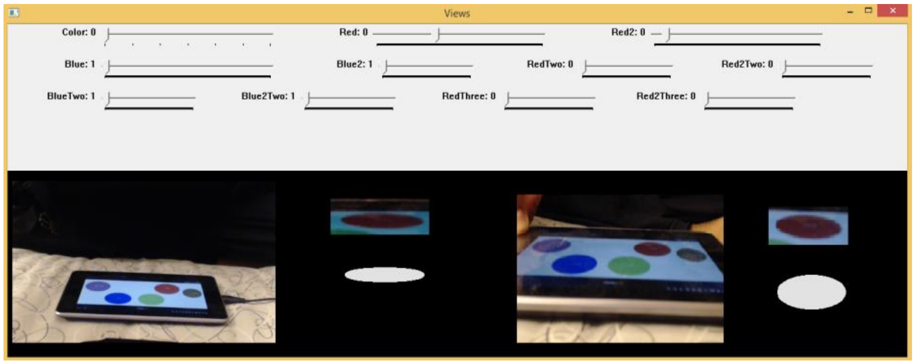


Figure shows the sub-frame selected by the user for the red object.

Primitive built-in C++ sound library: We used an open source library (irrKlang) [2], which greatly sped up the sound processing and allowed for overlapping sounds to be produced.

Complicated User Interface: We used an open source project [3] that merged multiple screen outputs to a single one. By doing so, we were able to create a user-friendly environment with minimal amount of windows.



Color: 0
Red: 0
Red2: 0
Blue: 1
Blue2: 1
RedTwo: 0
Red2Two: 0
BlueTwo: 1
Blue2Two: 1
RedThree: 0
Red2Three: 0

Figure shows the combined user interface of the program. Each feature is easily accessible through this layout.

Testing:

We ran tests to ensure that all circle were being detected properly. We also ran tests to ensure that our threshold logic was working for each circle. Lastly, we ran tests to ensure that both the camera's outputs coincide for each circle, for example, "snare" on one camera should be "snare" on the other camera as well, otherwise the output will be erroneous.

We asked 5 students who have played drums before to test our program and rate each area of the program, on a scale of 1-5 (5 being best). This is how their results looked like:

Student	Realistic	Sounds	Color Detection	Speed	Overall
1	4	5	5	3	4.25
2	3	4	5	4	4.00
3	3	5	5	4	4.25
4	4	4	4	4	4.00
5	3	5	4	3	3.75
Overall	3.4	4.6	4.6	3.6	4.05

Results & Evaluation:

The program is still in the development stage and a lot of improvements can be made, in order to achieve a more realistic and faster program. As we can see by the user ratings, we specifically need to improve on the speed of the program, which in turn will make it more realistic. We can include motion detection to calculate the intensity of the sound produced. Moreover, with a better colored detection algorithm, we can improve the performance for the threshold logic and limit the effect of lighting on the outcome. Our sounds produced are reasonable but an improvement on them can be easily made and will help us get closer to our goal of the near-perfect program.

Discussion:

We are very satisfied with the results and think that it is a big step towards achieving the ultimate goal of realistic programs through paper products. As of now, the limitation of our program is that it is affected by light intensity depending on the user-specified color thresholds.

Conclusion:

We were successfully able to create a computer vision program that used colored object detection to produce sounds. In the near future, we will turn it into a web application, accessible to everyone with a webcam. We also plan to improve on the program by including motion detection, better color recognition and optimizing our code. Through this logic, we also want to create other projects such as Paper Piano, Paper Games etc.

Acknowledgment:

We thank Professor John Magee (Clark University) for teaching us with the concepts of Computer Vision and for providing the two-camera logic, which results in a more realistic program, as the sound is only played when the user hits the object.

References:

[1] Fidan, K.C., Kehribar, I., Şahin, M.T., Cosar, S. Air drums: A computer vision based drums simulator. In Signal Processing and Communications Applications Conference (SIU), 2010 IEEE 18th, pages 844 - 847, Diyarbakir, Turkey, 2010.

[2] Ambiera. irrKlang: a cross platform sound library for C++, C# and all .NET languages.

[3] Vin Vinod, Showing multiple opencv images/frames on the same canvas. http://stackoverflow.com/questions/5089927/show-multiple-2-3-4-images-in-the-same-window-in-opencv. April 2014.