CSC420 Project Report

Thomson Yu
1001276754
Partner: Bill (Jia Ning) Gan

Introduction

Music transcription is often not a simple task as it requires a professional musician or computer to listen to the audio to produce the transcripted music notes. Depending on the audio quality, some transcriptions may be wrong and will need another form of analysis such as video. Arias et al. (2017) proposes a method for automatic visual note estimation by detecting the fingertips of a clarinet player where they measure the displacement of the fingers with respect to the holes and keys of the clarinet. The method proposed achieves a mean accuracy of 47.2% using 8 classes, thus this is a problem that is not easy to solve. In this report, we attempt to tackle the problem except we use drums as our instrument as there is larger margin of visual area and is easier to detect. Other instruments such as the clarinet, violin, etc., have a much smaller margin of visual area for the different notes that they produce. The goal of this is to transcribe drumming visually through videos of drumming found online. Due to our limitations in knowledge of music theory, we will produce copies of the videos with drum sounds being played based off the transcription instead of producing proper musical transcriptions.

We expect to have videos where the drumming is clear and slow to have the correct output. Videos where the drumming is fast, or the camera is off to the side and the drumsticks cannot be seen clearly will be expected to have worse results. We expect other main challenges to be gathering the right data to analyze as some drumming videos may not have great quality or the camera may be moving around a lot and analyzing when the drumstick hits the drum as it will be hard to tell without depth.

Methods

The process can be broken down into 5 major steps:

1. Find drumming videos and clip out the parts where the drummer is playing
2. Detect the locations of the drum, drumstick, and drummer
3. Track the drumstick for each frame of the video
4. Determine whether the drumstick has hit the drum for each frame
5. Produce audio synced to the video of when the drumstick hits the drum

Step 1: Finding drumming videos

We primarily focused on gathering drumming videos off Youtube. We first began by searching for people playing on a regular drum kit but soon realized that videos with regular drumsticks produced very poor tracking results which is explained in further detail in Results & Discussion. We then searched for videos of people playing on a bass drum as the mallet was a much easier object to track.

Step 2: Object detection

In order to initialize the tracking for the drum, drumstick, and drummer, we needed to detect the locations of the objects. Originally the plan was to use a pre-trained model such as Mask R-CNN and RetinaNet to detect the drums and drumstick, however only COCOAPI had pre-trained weights, which did not include drums or drumsticks as one of

Thomson Yu
1001276754
Partner: Bill (Jia Ning) Gan

the categorical classes. Instead we used template matching to locate the drumsticks. At first, we cropped out the tip of the drumstick from the first frame of the video and used that for template matching but simplified it to using the built-in function cv2.selectROI [1] to manually select the drumstick, drum, and drummer.

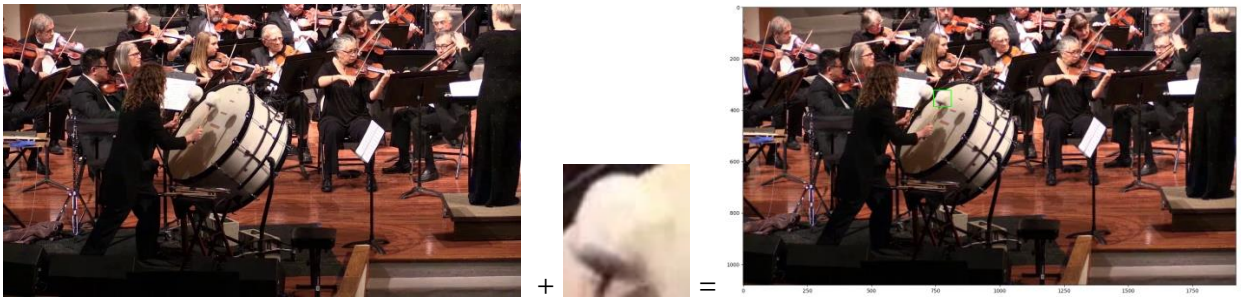

Figure 1: Template matching of drumstick/mallet

Step 3: Object tracking

In order to detect when the drumstick hits the drum, we first need to track the drumstick. Mallick (2017) wrote a tutorial on object tracking using a couple tracking methods which we followed. Of those methods, we found that CSRT tracking has the best tracking accuracy. In the code, the initial objects that were detected in step 2 are used to initialize the CSRT tracker [2]. Next, the tracker updates the position of the bounding box around the detected drumstick for every frame of the video.

Step 4: Hit Detection

Once the drumstick, drum, and drummer has been detected and successfully tracked, the next step is detecting when the drumstick hits the drum. Due to the lack of depth information, it will be difficult to calculate when the drumstick hits the drum. Therefore, we implemented algorithms based on patterns that were observed of the drumstick. We noticed two patterns; when the drumstick makes contact with the drum, the drumstick is further away from the drummer and the other pattern that we noticed was that once the drumstick has made contact, the drumstick changes direction.

Algorithm 1 [3]

When the drumstick hits the drum, the bounding box on the drumstick is further away from the drummer. We can use this to estimate when contact is made through the following calculation,

If the drum is on the right side of the drummer:

$Hit = drumstick_x > min_x + (max_x – min_x) * threshold$

If the drum is on the left side of the drummer:

$Hit = drumstick_x < min_x + (max_x – min_x) * (1\text{-}threshold)$

Max and min are the rightmost and leftmost points of the drumstick, and threshold is essentially how far the drumstick needs to be before being considered a hit.

Algorithm 2 [4]

Thomson Yu
1001276754
Partner: Bill (Jia Ning) Gan

Once the drumstick hits the drum, the direction of the drumstick changes, thus we can analyze the locations of the drumstick in the neighboring frames to know if a hit has occurred.

If the drum is on the right side of the drummer:

Let i be the current frame number

$Hit = dStickFrame(i-1)_x < dStickFrame(i)_x \text{ and } dStickFrame(i+1)_x < dStickFrame(i)_x$

If the drum is on the left side of the drummer:

$Hit = dStickFrame(i-1)_x > dStickFrame(i)_x \text{ and } dStickFrame(i+1)_x > dStickFrame(i)_x$

Essentially a hit is when there is a local maximum/minimum depending on the location of the drum

Stereo [5]

The above algorithms are based on the observed patterns and may not work on all drumming videos. Ideally, we would want to check if the drumstick hit the drum through a disparity map. However due to lack of access to a stereo camera and drum kit, we decided to take a more general approach by using a 3D camera app that we found on the Google Play Store and took stereo photos of a water bottle against a poster surface to mimic a drumstick against a drum.

Step 5: Generating audio and visualizing the hit detection

The final step is to visualize and add audio to the frames of when the drumstick hits the drum. We create an audio clip that is the same length as the video. We then add the drum audio depending on if the neighboring frames include a hit [6]. Then the audio and the video clips are synced together using an open source library called ffmpeg [7].

Results & Discussion

Object detection:

Since using a pre-trained model did not work for us, we went with manually selecting the drumstick, drum, and drummer which is more accurate and easier to use.

Object tracking:

Tracking fails completely when the drumstick either moves too fast or moves off screen. This problem becomes worse for videos that use regular drumsticks, thus we needed to look for videos that use a larger mallet. The CSRT tracking algorithm works well for tracking larger mallets and occasionally fails when the tempo is fast, or the drumstick moves off screen.

CSC420 Project Report

Thomson Yu
1001276754
Partner: Bill (Jia Ning) Gan

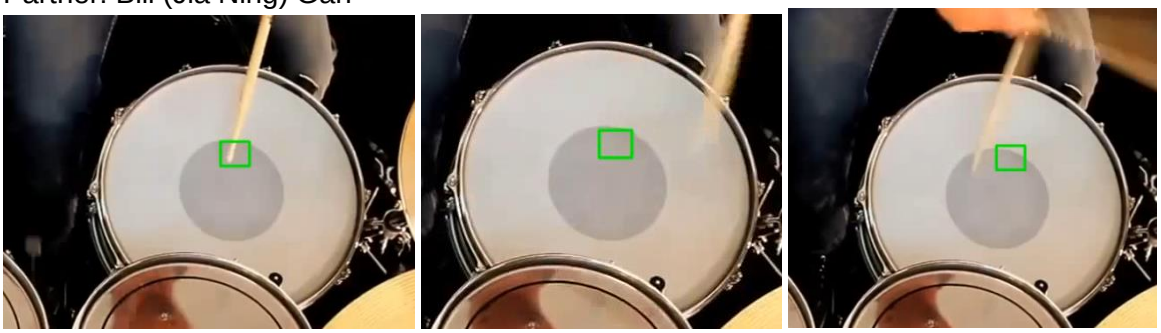

Figure 2: Tracking small drumsticks fail when they move too fast

Figure 3: Tracking the mallet fails when it moves out of view

Hit detection:
Algorithm 1 produces results however the accuracy of detecting a hit is low. The algorithm just checks whether the drumstick passes over the threshold line. This method works if the drummer hits the exact same spot and lifts the drumstick away from the drum after each hit but ideally, we would want a method that works for any drumming video. This method fails when the drummer does not lift the drumstick enough before hitting the drum again, causing the algorithm to think that the drum is constantly being hit.

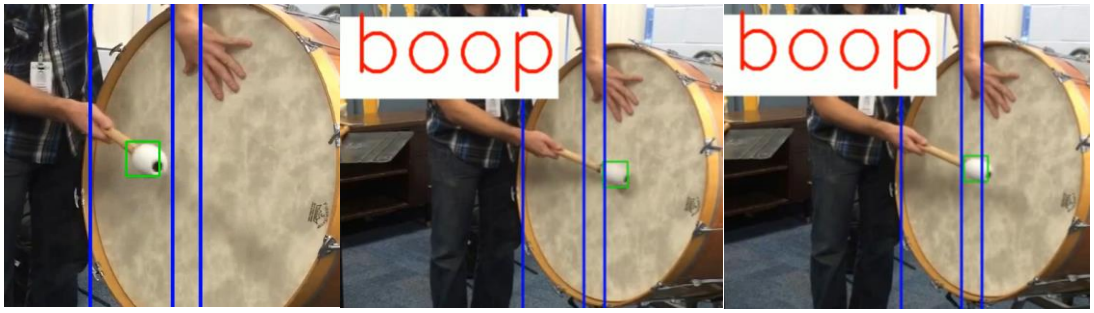

Figure 4: Algorithm 1 identifies the frames in the third image where the drummer is lifting the mallet off the drum as a hit

Algorithm 2 produces better results and the method can be used in more cases. The accuracy is higher as it detects a change in direction based on the location of the drum

Thomson Yu
1001276754
Partner: Bill (Jia Ning) Gan

relative to the drummer. However, there are some cases where the algorithm fails, such as the drumstick changing direction without hitting the drum.

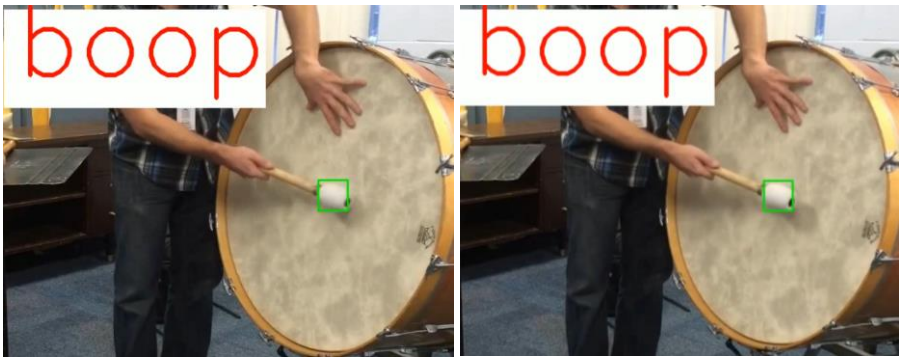

Figure 5: The first image is a correct hit but algorithm 2 misidentifies the second image as a hit since the drummer moved the drumstick to the left and then back to the right

Stereo:

The two algorithms above may not work for all drumming videos. Ideally, we would use stereo to get the depth maps and check when the disparity of the drumstick is the same as the drum. However, we did not have access to a real stereo camera or a drum kit, so we took photos of a water bottle against a poster using a 3D camera app. We were able to get decent results however we were only able to take photos and not video.

Figure 6: Stereo photos of water bottle being further away from the door

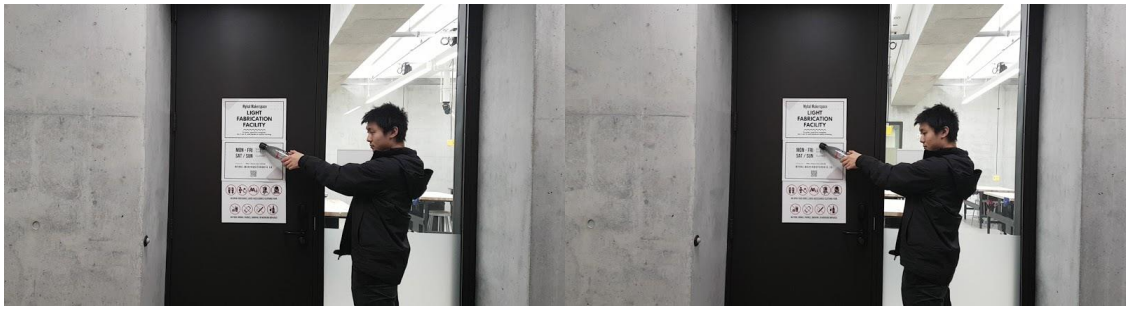

Figure 7: Stereo photos of water bottle making contact with the door

Thomson Yu
1001276754
Partner: Bill (Jia Ning) Gan

Figure 8: Depth maps of water bottle tip against poster surface. Left image = tip of water bottle being further away. Right image = tip of water bottle making contact with poster

Main Challenges

The main challenges that we faced were detecting the drums and drumsticks which we solved by manually selecting the objects and doing template matching instead of using a pre-trained model. Tracking small drumstick tips which we instead went for videos with larger mallets that were easier to track. Detecting the drum hits without depth information which we instead observed the motion of the drumstick and implemented algorithms based on the patterns that we observed. Using stereo to detect drum hits was also quite challenging as we did not have access to a stereo camera or a drum kit, so we improvised by using a 3D camera app that we found on the Google Play Store to take stereo photos of a water bottle over a poster surface to mimic a drumstick on a drum.

Conclusion

Transcribing music via visual computing techniques can be done as long as the videos are clear, and the instrument can be successfully tracked. First template matching was done to locate the drumsticks, drum, and drummer. Then the drumsticks are tracked for every frame in the video and are locations are used to compute whether the drumstick has hit the drum. We primarily used two algorithms to do this, the first algorithm which was less accurate, checked whether the drumstick passed over a threshold line on the drum indicating a hit. The second algorithm which was more accurate looked for a change in direction of the drumstick based on the position of the drum relative to the drummer which thus would indicate a hit. These algorithms were only based on patterns observed in the videos and may not work in every video, thus using stereo to check the depth of the drumstick would be more accurate. Once we have determined when the drum was hit, drumming audio and hit visualization was added as a way to transcribe the drums. To further improve results, a better setup that can take stereo video would be crucial to gaining a higher accuracy for determining hits. In our approach, we simply detect if there was a hit or not. To improve the quality of the transcription, we would detect the type of collisions such as a soft or a hard hit for varying loudness so that the transcription more closely resembles what's being played.

CSC420 Project Report

Thomson Yu
1001276754
Partner: Bill (Jia Ning) Gan

References:

Camarada: 3D Camera, 3D Video, 3D Selfie, 3D Photo - Apps on Google Play. (n.d.). Retrieved December 1, 2018, from https://play.google.com/store/apps/details?id=com.aimfire.camarada

Gomez, Emilia, Pablo Arias, Pablo Zinemanas, and Gloria Haro. "Visual Music Transcription of Clarinet Video Recordings Trained with Audio-Based Labelled Data." *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, 2017. doi:10.1109/iccvw.2017.62.

Mallick, S. (2017, February 13). Object Tracking using OpenCV (C /Python). Retrieved December 1, 2018, from https://www.learnopencv.com/object-tracking-using-opencv-cpp-python/

Videos:
Nofmusic. (2014, June 26). IMAGINE DRAGONS RADIOACTIVE- BRAVALLA 2014 - KNOCKS THE BIG DRUM OVER. Retrieved from https://www.youtube.com/watch?v=EylIxrEKY5Q

Howcast. (2013, September 08). How to Play Basic Rock Drum Beats | Drumming. Retrieved from https://www.youtube.com/watch?v=CvsFEsXakwo

PasadenaPCO. (2018, May 11). Concerto for Bass Drum and Orchestra - Gabriel Prokofiev. Retrieved from https://www.youtube.com/watch?v=E3ZSLMKcdzQ

Music, T. W. (2014, November 06). Five Minute Drum Lessons - How To Play Concert Bass Drum. Retrieved from https://www.youtube.com/watch?v=2-MXOI457EY&t=

Appendix:

[1] object_detection function in project_main.py
[2] tracking function in project_main.py
[3] contact_made function in project_main.py
[4] contact_made_2 in project_main.py
[5] contact_made_stereo function in project_main.py
[6] create_audio function in project_main.py
[7] write_video_audio function in project_main.py