Air Drums: A Computer Vision Based Drum Simulator

Kaan C. Fidan† KAANCFIDAN@SABANCIUNIV.EDU
İhsan Kehribar† KEHRIBAR@SU.SABANCIUNIV.EDU
M. Tuğçe Şahin† MERVETUGCE@SU.SABANCIUNIV.EDU
Serhan Coşar† SERHANCOSAR@SU.SABANCIUNIV.EDU
Devrim Ünay‡ DEVRIM.UNAY@BAHCESEHIR.EDU.TR

†Computer Vision and Pattern Analysis Laboratory, Sabanci University, Orhanli, Tuzla, İstanbul, Turkey

‡Electrical and Electronics Engineering, Bahcesehir University, İstanbul, Turkey

1. Introduction

The aim of this paper is to present a novel system which tracks the motion of a drummer and generates the corresponding drum sounds. Only a camera, some colored markers and an everyday PC are used in the development of the system. The input video sequence from the camera is processed in real-time by using local and adaptive color segmentation and Kalman filter based tracking. The Kalman filter is used to predict the ”hits” so that we can overcome the processing delays and provide a more-realistic drumming experience. We use a local and adaptive search to detect the effective points of the drum sticks, which ensures robustness to background clutter and reduces the computational burden. We developed a working demo and evaluated its performance by comparing with the output signal of an electronic drum pad. We observed that the timing errors have an average of -8.4 ms and a standard deviation of 5.4 ms, where the two extreme values were -22.9 and 3.2 ms in a real drumming experiment consisting of 121 hits.

2. Related Work

The aim of the project is to create an ”edutainment” opportunity in an easily achievable system. The resulting human-machine interface from this work can be used as a way to improve training sessions of the drummers as well as entertainment purposes. There has been some research conducted about tracking drumsticks in the search for new media for educating the next generation in specialized skills. The described system requires previously recorded videos of qualified drummers performing some basic training sets, then the videos are processed and the captured motions are parameterized for future comparison to the students' results (Tansuriyavong et al., 2006). The advantage of our system is its real-time tracking of the drumsticks in order to create a more realistic drumming experience.

Another field of research is focused on audio-visual processing and musical transcription of the drumming performances. These works exploit visual information of the drumsticks and the drums together with the audio of the performance (Gillet & Richard, 2005; McGuinness et al., 2007). To the contrary, our system simulates an imaginary drumset and generates the audio from the visual data.

3. Materials and Methods

3.1 System Overview

Initially our system was planned to work with an everyday webcam, however the nature of the drumstick motions requires processing at high frame rates. The discretization becomes too steep in low frame rates to approximate velocities and accelerations of the tips. Therefore in this work we employ an IDS uEye 1640 camera, which can provide up to 100 frames-per-second (fps) at 320x256 resolution in decent lighting conditions.

For drumstick tracking and hit detection, our system, which is fully implemented in C#, employs the computer vision algorithms presented in the OpenCV library (Bradski, 2000) through the use of the EmguCV wrapper (http://www.emgu.com/wiki). Following the hit detection step, we generate the corresponding MIDI signals that can be picked up through a virtual MIDI cable by any audio processing tool.

The algorithm workflow can be seen in Figure 1. The system only needs clicks of the user on the tips of the drumsticks to initiate the segmentation process.

3.2 Drumstick Segmentation

The segmentation is carried out in HSV space. The hue channel is thresholded within a small tolerance (all the other pixels which do not belong in the range *picked hue*

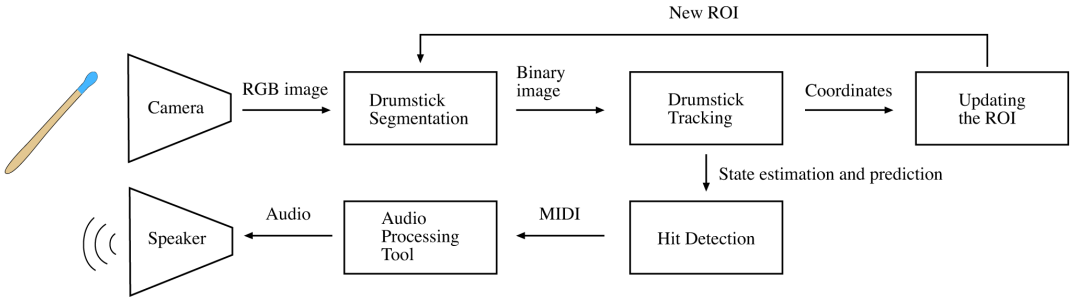

Figure 1. Algorithm Block Diagram

$\pm$*tolerance* are suppressed, chosen pixels are marked with 255). The tolerance for the saturation and the value is tuned by the user and processed in the same manner. The saturation and the value tolerance covers the shadows and bright spots on the colored marker. The intersection between the hue, saturation and the value thresholded images contains only the colored marker. The *gravity center* of the binary image is calculated.

3.3 Drumstick Tracking

We used *Kalman filter* for tracking. The main motives are:

✦Reducing the effects of instantaneous measurement ripples caused by lighting variations on the drumstick.

✦Predicting the ”hits” before they happen for the sound to be generated at the moment of the actual hit.

In our case, Kalman filter works with state vectors which contain the position, the velocity and the acceleration on X and Y coordinates. The measurement is given as the position of the gravity center. Kalman filter uses this information to approximate the velocity and the acceleration, correct the current measurement using noise models and predict the next state (Kalman, 1960).

3.4 Updating the Region Of Interest

The ROI is the window where the segmentation is done in each cycle. It is crucial for reducing the computational weight and increasing the number of processed frames, hence increasing the number of data points. Once the gravity center is defined, a window is formed around it. If the predicted position of the tip gets out of the window, the window gets larger in proportion with the velocity for not losing the tip. When the tip is stable, the window turns to its initial size. If the tip is lost (i.e falls outside the window), the ROI is canceled and whole image is processed until it is found again.

The Fig.2 shows a captured image from the working software. The drumstick is modeled with a boardmarker and it is accelerating downwards at the moment. In the binary image, red point marks the current gravity center and the blue point, the predicted position. The white frame is the ROI, which is expanding to include the next state prediction.

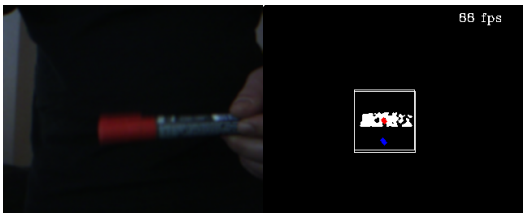

Figure 2. Segmentation and tracking of the drumstick tip.

3.5 Hit Detection

The real-time goal brings the difficulty of predicting the hit without the proper information. Hence, the hit detection algorithm includes a series of assumptions:

✦ *Vertical acceleration should be negative and larger than a small threshold.* The downward acceleration tells that the drumstick is gaining speed and getting close to a “hit”. The downward acceleration peaks for a very small amount of time before the hit due to the nature of the specific motion.

✦ *The predicted position should be in a drum zone.* The sound is generated corresponding to different ”drum zones” and thus, where the hit will land is important.

✦ *There cannot be 2 consecutive hits with the same drumstick in 80 ms.* The downward acceleration may appear in several consecutive frames corresponding to the same hitting motion and cause multiple detections. It is assumed that even if the user is able to perform at such speed, the camera would not be able to gather enough data points to correctly detect the hit.

✦ *Volume of the hit is proportional to the current velocity.* In the real world, the volume is proportional to vibration

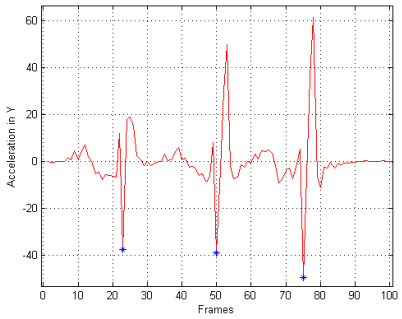

Figure 3. Vertical acceleration of the drumstick tip with the detected "hit" marked by a star.

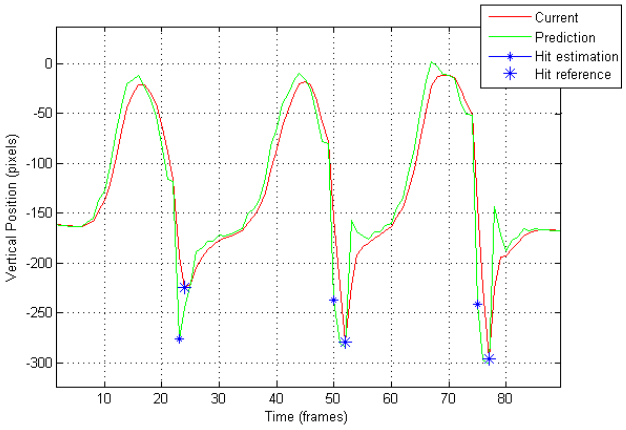

Figure 4. Vertical Position vs. Frames

amplitude of the drumhead caused by the reaction force. The reaction force can be estimated from the acceleration which stops the drumstick with Newton's $F = m.a$. We assume that the current velocity is a measure of how high the reverse acceleration will be when the drumstick stops.

4. Results

Our system currently faces difficulties at robustly tracking the tips of actual drumsticks due to their very fast movements which cause motion blurs at current acquisition settings. We use shorter sticks (i.e. board markers) that have smaller radius of circular motion, hence easier to spot the tips in rigid form. Figures 3 and 4 depict a recording of 3 hits and shows the hit detection algorithm's results.

We have created an experimental setup where we processed the visual data from a drum practice routine consisting of 121 hits on electronic drum pads and compared the resulting MIDI signals. We observed that the timing errors had an average of -8.4 ms and distributed with a standard deviation of 5.4 ms (Fig. 5).

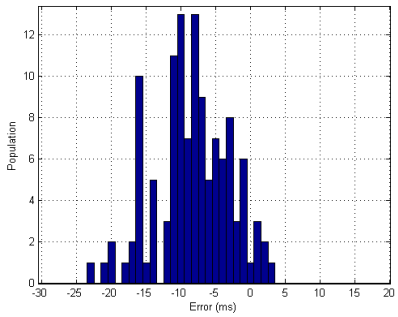

Figure 5. Error distribution of the proposed method evaluated on a recording of a drum practice routine.

5. Conclusion

The results show that we are able to compensate the latencies coming up from the sound generation, and provide the desired experience. The system works well in decent acquisition conditions (i.e uniform and direct lighting, high frame rates) and can be used to play and record drum sequences as if they were played on a MIDI keyboard controller.

As future work, we will try to track 2 other colored markers on actual drumsticks and use them to estimate the position of the lost tip. We will also create a user-friendly calibration routine for mapping the units to the metric system. The calibration will allow the system to be more robust to variations in distance between the user and the camera. Up-to-date information and the demo videos can be found at http://www.airdrums.info.

References

Bradski, G. (2000). The opencv library. *Dr. Dobb's Journal of Software Tools*.

Gillet, O., & Richard, G. (2005). Automatic transcription of drum sequences using audiovisual features. *IEEE International Conference on Acoustics, Speech, and Signal Processing*.

Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Transactions of the ASME - Journal of Basic Engineering*, *82*, 35–45.

McGuinness, K., Gillet, O., O'Connor, N. E., & Richard, G. (2007). Visual analysis for drum sequence transcription. *EURASIP*, 312–316.

Tansuriyavong, S., Nagai, H., Nakahira, K. T., & Fukumura, Y. (2006). Development of multimedia contents for specialized skill education. *Current Developments in Technology-Assisted Education* (pp. 371–375).