Music Moves

Wim Looman
Electrical and Computer Engineering
University of Canterbury
Christchurch, New Zealand
wgl18@uclive.ac.nz

Richard Green
Computer Science and Software Engineering
University of Canterbury
Christchurch, New Zealand
richard.green@canterbury.ac.nz

***Abstract*—This report details the design and implementation of an "air drum" interactive music program. Various methods for implementing this on consumer hardware utilising a simple webcam are explored. The results from this, while not entirely working, are definitely promising. The marked hand tracking worked effectively and the neural net based gesture recognition was able to generate a surprisingly high proportion of matches. Unfortunately the initial plan of having a MIDI output that could be connected to an actual music program such as GarageBand was sadly not achieved.**

I. Introduction

The goal of this project is to create an interactive music program that can generate music based by analysing the movement of a human body or hand position. This will be capable of running in real time on a consumer pc utilising a single webcam.

A lot of different methods for producing movement from sound have been researched. Some of these involve using the hands to "shape" the music, for example having a 2d pointer (which could easily be generated from a hand tracking algorithm) that affects the music based on a set of simple gestures [1]. Others attempt to provide a more traditional musical interface, albeit without the actual instrument. This can be a simple interface such as a DJ's deck [2], or a more complex interface like a full piano [3] or guitar [4].

Whilst the first approach will probably lead to a more novel category of research, it would also require a deep understanding of music theory. Therefore the second approach was the one chosen for investigation.

One instrument that hasn't been widely studied for video based playing is the air drum. It would have to be wondered why; the drum simultaneously provides both a complex interface, the 3d position of the imaginary sticks, and a simple interface, only around 3-10 different drums/cymbals to hit depending on the drum set.

To enable the user to play the air drum just via a normal webcam a robust algorithm for turning their motions into sound will be needed. One possibility would be to write an algorithm that looks for differences in the hand positions that indicate the *strike* of the swing. However, this would be a rather complex algorithm and be difficult to derive manually.

Instead a segmented approach was decided upon, the traditional unix-like piping approach; see Figure 1. The individual frames of the camera will first enter a hand-tracking algorithm. This will output the current (x, y) co-ordinate of each hand.

These (x, y) co-ordinate sets will then enter a buffer. This allows the next part of the pipe to compare the last set of frames, necessary for detecting the *strike* of a swing.

Each co-ordinate set in the buffer will be input into a neural network, this will compare all the values and decide if one of the drums was just struck. This will almost necessarily introduce at least a one frame delay in the output, part of the optimization of the neural network will be testing out different delays.

The strikes from the neural network will then pass into the output stage, this will either output the sound directly or convert the strikes into the appropriate signals to be passed along to the sound controller.

A. Hand Tracking

There are a few possibilities for the hand tracking, with or without marks. With marks we could have the user wearing a pair of brightly coloured gloves, or holding a pair of brightly coloured drumsticks. Without marks we could attempt an implementation of the Kalman filter to track the front/tops of the users fisted hands.

B. Neural Network

The neural network would have a large input layer, four nodes per frame of the video. To provide context it is assumed that at least a second of video will be required, at say 30 frames a second this means the input layer would be at least 120 nodes. The actual length of the context will be part of the optimisation of the neural network.

C. Sound Output

II. Implementation

A. Hand Tracking

To make this simpler the mark based tracking was chosen, specifically a pair of brightly coloured gloves. This will limit the subjects clothing in that it cannot contain the same colour as the gloves. However it greatly simplifies the hand tracking to just following the specific colour of the gloves.

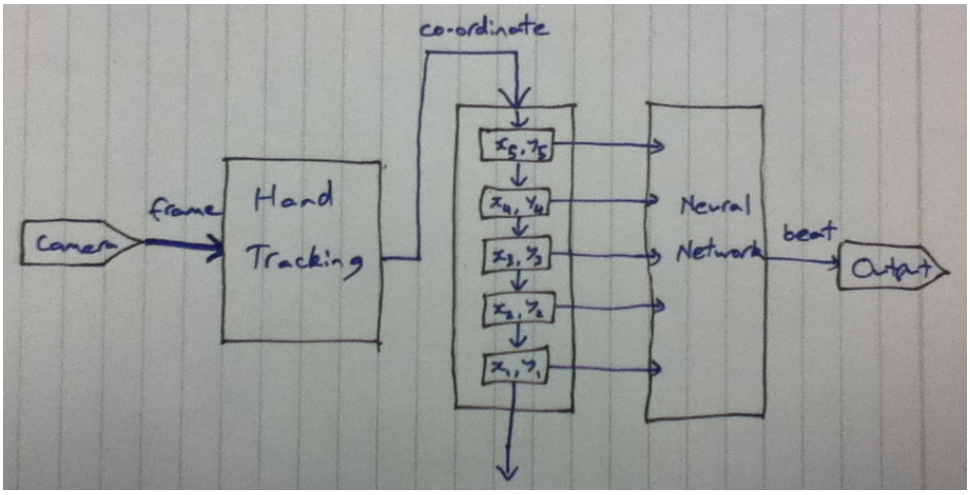

Figure 1. The pipeline of the approach (draft image, will be converted to svg at some point).

The tracking will involve two stages, target acquisition & learning and the actual tracking. During acquisition and learning the user will start with their hands behind their back, then hold them in a few positions in front of the camera. This will produce a series of images which will be automatically processed to attempt to compensate for the lighting conditions. Once an accurate enough colour sample has been achieved the actual tracking will be started.

B. Neural Network

To simplify the creation of the neural network it was decided that this should be a separate program. Writing the network in Erlang would be much easier than using C, and by having a separate program the training would be simpler to setup.

III. Results

A. Hand tracking

The hand tracking worked successfully, using the bright red gloves provided a very distinct edge in the filtered image. This could then simply have the centroid calculated for the hand position. An example is shown in Figure 2.

Figure 2. The hand tracking (draft image, real image will come soon).

B. Neural Network

The neural network was also quite successful, with an 87% pass rate on the test cases.

The network was tested with a large variety of different parameters, via a series of automated tests. The number of input frames was varied between 10 and 55 with a step of 15, corresponding to around 0.3 to 2 secs at the nominal output rate of 30 fps that the hand tracking algorithm was able to sustain. The number of hidden layers was also varied between 1 and 2, in case the required learning pattern was of a higher order.

IV. Conclusion

References

[1] K. Jensen, "Aspects of the multiple musical gestures," *Journal on data semantics*, vol. 3902, pp. 140–148, Jan. 2006.

[2] B. Yetton and R. Green, "Virtualmix - virtual dj interface using computer vision," 2010, produced for COSC428.

[3] R. I. Godøy, E. Haga, and A. R. Jensenius, "Playing 'air instruments': Mimicry of sound-producing gestures by novices and experts," *Journal on data semantics*, vol. 3881, pp. 256–267, Jan. 2006.

[4] Y. Pan and R. Green, "A mark-less air guitar game through computer vision," 2010, produced for COSC428.

[5] B. Chong and R. Green, "Music moves," 2010, produced for COSC428.

[6] L. Tarabella, "Handel, a free-hands gesture recognition system," in *Proceedings of the 2004 Second Internaional Symposium Computer Music Modeling and Retrieval*. University Esbjerg, Denmark, 2004.

[7] T. Winkler, "Making motion musical: Gesture mapping strategies for interactive computer music," *Computer*, 1995.

[8] G. Castellano, R. Bresin, A. Camurri, and G. Volpe, "User-centered control of audio and visual expressive feedback by full-body movements," in *Proceedings of the 2nd international conference on Affective Computing and Intelligent Interaction*, ser. ACII '07. Berlin, Heidelberg: Springer-Verlag, 2007, pp. 501–510. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-74889-2_44

[9] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," *ACM Comput. Surv.*, vol. 38, December 2006. [Online]. Available: http://doi.acm.org/10.1145/1177352.1177355

[10] D. Popa, G. Simion, V. Gui, and M. Otesteanu, "Real time trajectory based hand gesture recognition," *WSEAS Trans. Info. Sci. and App.*, vol. 5, pp. 532–546, April 2008. [Online]. Available: http://portal.acm.org/citation.cfm?id=1481952.1481972

[11] R. Behringer, "Gesture interaction for electronic music performance," in *Proceedings of the 12th international conference on Human-computer interaction: intelligent multimodal interaction environments*, ser. HCI'07. Berlin, Heidelberg: Springer-Verlag, 2007, pp. 564–572. [Online]. Available: http://portal.acm.org/citation.cfm?id=1769590.1769654

[12] A. Salgian, M. Pfirrmann, and T. Nakra, "Follow the beat? understanding conducting gestures from video," in *Advances in Visual Computing*, ser. Lecture Notes in Computer Science, G. Bebis, R. Boyle, B. Parvin, D. Koracin, N. Paragios, S.-M. Tanveer, T. Ju, Z. Liu, S. Coquillart, C. Cruz-Neira, T. Müller, and T. Malzbender, Eds. Springer Berlin / Heidelberg, 2007, vol. 4841, pp. 414–423. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-76858-6_41

[13] K. Bowyer, C. Kranenburg, and S. Dougherty, "Edge detector evaluation using empirical roc curves," *Computer Vision and Image Understanding*, vol. 84, no. 1, pp. 77 – 103, 2001. [Online]. Available: http://www.sciencedirect.com/science/article/B6WCX-4582BV1-6/2/ad72aaf94049253f5333f721b9afdb69