CONCERT-MASTER: SYNTHESIS OF MUSIC FOR THE LAZY USING CLASSIFICATION OF HAND GESTURES

Anirud Thyagharajan

Electrical Engineering Department,
Indian Institute of Technology Kharagpur
WB, India - 721302

SL Happy

Electrical Engineering Department,
Indian Institute of Technology Kharagpur
WB, India - 721302

ABSTRACT

Gesture detection is an active area of research in the fields of Computer Vision, Machine Learning and Artificial Intelligence. It involves classifying certain positions of the human body and using them to actuate activities. This paper proposes a gesture based music synthesis system for the blind, to enable them to generate music notes and/or select songs solely by moving their hands. The proposed system consists of a camera (preferably webcam) which detects the user's hands – contour and fingertips, and can be operated in two modes, (1) playing a specific tone of a specific octave, and (2) playing a song from a list of them. The detection of the hands are done using skin detection in the Y-Cb-Cr space followed by a contour analysis of the same. Other modes also include segmentation in the HSV space, and adaptive thresholding using histogram backprojection. The system is programmed in *Python* and has a friendly command line interface for operating various modes.

Index Terms— Gesture, Hand Detection, Music, Y-Cb-Cr

I. INTRODUCTION

The gesture is an important part of human communication, and it is used often - even unconsciously - as a means of expression and interaction with the world, having a strong impact on how humans perceive and interpret themselves. There is an important distinction between gesture and movement. Gesture presupposes an intention, a meaning and the movement is the physic action itself (e.g. a set of arm movements waving composes the gesture of saying goodbye). Nowadays, different methods can be used to capture human movements using, for instance, video cameras, body wearable sensors or external sensors, such as infra-red Motion Capture (MOCAP) systems. All these processes allow capturing and gathering signal data, from where relevant gesture features can be extracted for further analysis and processing (e.g. estimating amplitude, periodicity, rhythm, diversity, etc., of a gesture or movement). Such features can then be used as inputs for real-time algorithmic music composition systems, paving the way for novel expressive and artistic works, where humans and machines interact in a more semantically and artistically meaningful dialog. The motivation behind this particular system, is the possibility of analyzing human gesture at a higher level. Hence, the musical output can present more abstract and complex relations with the human gestures, rather than the direct mapping of human movements.

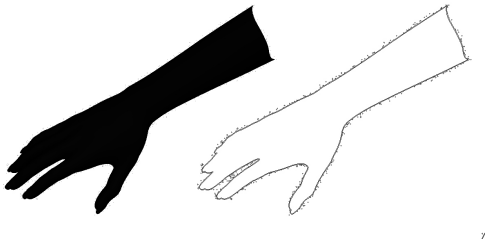

Fig. 1. Hand Detection serves as the first key to gesture recognition

This paper describes the system Concert Master, a modular system that permits the capture and analysis of human gestures using non-invasive methods (uses the ordinary Laptop camera in a well lit room). The system supports multiple modes, of which we will talk about in the sections to come. The camera captures frames and sends it to a software analyser, which will make sense of the frames depending upon the operating mode. In this paper, we employ 3 techniques to detect the hand, using (1) HSV Thresholding, (2) HSV Thresholding followed by a histogram backprojection and (3) Y-Cb-Cr thresholding. After the gesture or motion of the user is detected, this analyser will trigger a corresponding musical response to the gesture made. The novelty in this system is regarding the simplicity of the hardware and the granularity of selection of music pieces.

The paper is organised as follows. Section I presents a brief introduction to the research area that the system targets and the associated proposed system. Section II presents the prior art in the area and the pathflow of methodology associated. Coming to section III, it talks about the system design and the implementation of the system including the modes of operation. Section IV conveys the preliminary

results and observations, while Section V concludes with applications and future work proposed.

II. METHODOLOGY

The basis of the work is based on hand detection. Hand detection in the absence of other body parts resolves to skin detection; and hence the primal investigation in this direction occurs in image processing methods for skin detection. [1] talks about pixel based skin color detection techniques. Primitive methods suggest the use of HSV thresholding; with the HSV space calculated as follows:

$$R' = R/255,\ G' = G/255,\ B' = B/255 \tag{1}$$

$$C_{max} = max(R', G', B') \tag{2}$$

$$C_{min} = min(R', G', B')$$

$$\Delta = C_{max} - C_{min} \tag{3}$$

$$H = \begin{cases} 0 & \Delta = 0 \\ 60 * \frac{G'-B'}{\Delta} mod 6 & C_{max} = R' \\ 60 * (\frac{B'-R'}{\Delta} + 2) & C_{max} = G' \\ 60 * (\frac{R'-G'}{\Delta} + 4) & C_{max} = B' \end{cases} \tag{4}$$

$$S = \begin{cases} 0 & C_{max} = 0 \\ \frac{\Delta}{C_{max}} & C_{max} \neq 0 \end{cases} \tag{5}$$

$$V = C_{max} \tag{6}$$

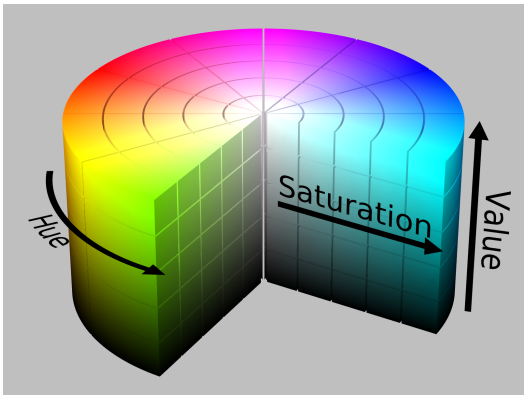

Fig. 2. The HSV Cylinder Model

Thus, using the above equations, the HSV space is generated from the RGB color space. After doing so, the image is thresholded as follows. Samples are thresholded with respect to minimum and maximum conditions: H(0, 14), S(66, 154) and V(110, 238). However, as is palpable from our experiments and previous research work, the HSV space was not resilient to changes in illumination, and hence skin detection in such cases was hampered severely.

As a result of this, some steps were taken in the experiments. The proposed system considers a 10 second initial learning phase when it requests the user to place his/her hand in certain squares of the window. The system thereby attempts to estimate the concentration of pixels in the hands, and thereby tries to learn the thresholds adaptively using a histogram backprojection method. This method can be seen in action in Fig. 1.

Though this method turns out to be much better than the naive HSV thresholding method, this method is also not entirely illumination invariant. [2] proposed a method of skin detection by transforming the image space to the Y-Cb-Cr space. Y-Cb-Cr is a family of color spaces used as a part of the color image pipeline in video and digital photography systems. It is used to separate out a luma signal (Y) that can be stored with high resolution or transmitted at high bandwidth, and two chroma components (Cb and Cr) that can be bandwidth-reduced, subsampled, compressed, or otherwise treated separately for improved system efficiency.

The components can be calculated as follows. As in (1), the analog R', G' and B' components are calculated.

$$Y = 16 + (65.481R' + 128.553G' + 24.966B')$$

$$C_B = 128 + (-37.797R' - 74.203G' + 112B')$$

$$C_R = 128 + (112R - 93.786G' - 18.214B')$$

The above Y, C_B and C_R values are measured in the 8 bit format. Thus, [2] noted that thresholding in this space yielded even better results, as it turned out to be illumination invariant to a good extent, discounting any drastic changes (either very dark or very light). However, due to thresholding with Chroma components, the detection could be marred by having objects of similar color, and hence this method fails in such scenarios.

The proposed system has options for all the three methods listed above, and hence can be switched based on the application/environmental conditions. The software section will discuss more about switching between these modes.

III. SYSTEM DESIGN

The system was built on a x64 machine running Ubuntu 14.04 in *Python*. The hardware module is just a webcam, which in this case was the default laptop webcamera. The software module had the following dependencies:

- Python 2.7
- OpenCV (2.4.x)
- Pyknon [1] (for allowing to create musical notes)
- Numpy (for matrix/array processing)
- Pygame (for playing the music)

The software was developed with a user friendly interface, allowing the users to provide options and arguments. The list of options include:

- `--deb`: Used for employing the Debugging mode.
- `--num`: Used for specifying the number of hands.

[1] https://github.com/kroger/pyknon

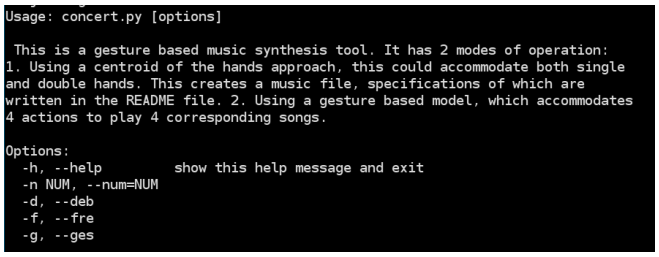

Fig. 3. The CLI Documentation

- `--fre`: Used along with the above option for free music synthesis.
- `--ges`: Used for a 4-class gesture for music selection, disjoint with the above 2 methods.

The software is open sourced at this link [2].

III-A. Feature Processing

As mentioned earlier, the system accepts input from a computer webcamera. The following points illustrate the processing associated:

- *Free hand movement*: Under this feature, the RGB frame is converted to a Y-Cb-Cr frame. The resultant binarized, thresholded image is sent for a contour analysis, following which the 2 largest contours are identified. Now, depending upon the user's choice, the program detects one or two of the user's hands and computes a unique identifier for music synthesis and pipes it to the music synthesis process. To guide the user, the program also draws a grid showing the pitch and octave variations in the x and y directions respectively. This mode computes the centroids of the contours and sends it to the music synthesis process.
  The music synthesis process here evaluates the octave and pitch based on and thus concatenates the entire gesture and creates a song, and plays it.

- *Gesture selection*: Under this feature, the RGB frame is converted to a Y-Cb-Cr frame. The resultant binarized, thresholded image is sent for a contour analysis, following which the largest contour is identified. After this, a convex hull is fitted over the identified contour, along with the convexity defects. Thus, this gives us the information about the vertices of the convex polygon fitted, and the defects, which are the points on the contour which lie farthest from the fitted polygon. The main idea of this is to find the junctions between the fingers, with which multiple gestures can be idenitified and classified.
  However, the convexity defects contain extraneous points than the desired junctions, and hence need filtering, based on the distance from the convex hull and the defect, and the angle subtended between two consecutive hull vertices at the defect. The distance should be greater than an adaptively learnt threshold, and the angle should be greater than 80°.
  After having detected the number of junctions in the gesture implied, the gesture is selected over a majority voting function over 10 epochs, and a corresponding song is played. Currently, only 4 gestures are made, but due to its geometric rule based nature, it is natural that it can be extended to custom gestures as well.

[2]https://github.com/anirudt/concert-master

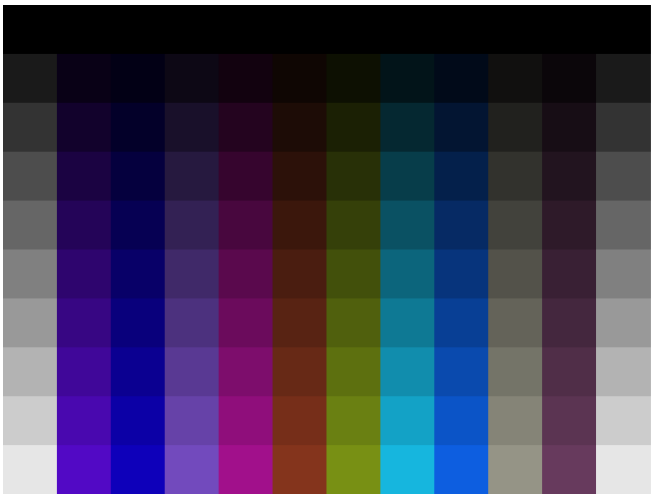

Fig. 4. Wallpaper grid for reference

III-B. Music Generation

The music generation is done using a Python library called Pyknon. The documentation of this library can be found in a book written by the writer of this library. The music module is also differentially drafted for each of the above operational modes:

- *Free hand movement*: The music module receives the `x` and `y` coordinates of the hand centroid. The music module divides the image captured into several blocks in the X and Y direction. After this, each block represents a certain (X, Y), which corresponds to a certain (P, O), where P is pitch and O is octave. This constitutes one single tone. Using this concept, the musical objects are sampled over every frame, and the musical tones thus generated are concatenated together to form one musical piece. In the end, this piece is synchronously played using `pygame mixers`.

- *Gesture selection*: The music module receives the gesture ID, and the `pygame mixer` plays the corresponding song from the playlist.

IV. RESULTS

The free mode of the system generates raw unprocessed notes. This can be viewed by the user as in Fig. 5, where

the scatterplot of the user's movement can be tracked; the darker red dots are plotted when the user initially starts, while the lighter ones depict the movement of the hand's centroid as it moves through time. This dynamic plot is generated to guide the users through the music generation process; users can learn from feedback how to move their hands to create customized musical pieces. Also, Fig 6 & 7 depict the different gestures in the Gesture selection mode of the system.

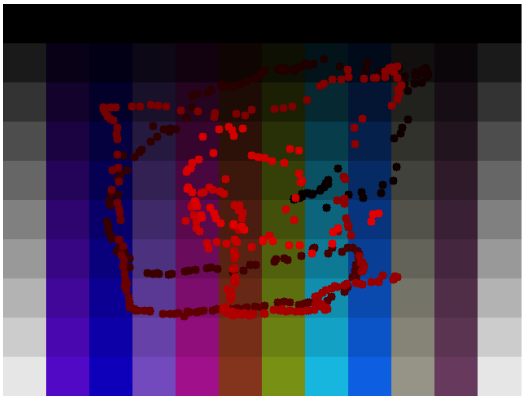

Fig. 5. Scatter pattern of the hand centroid over the musical grid

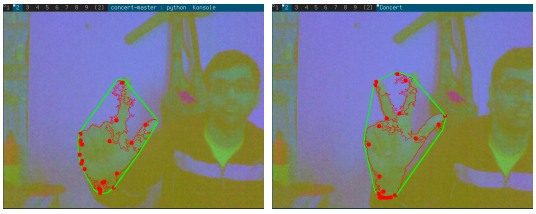

Fig. 6. Gesture 1 & Gesture 2

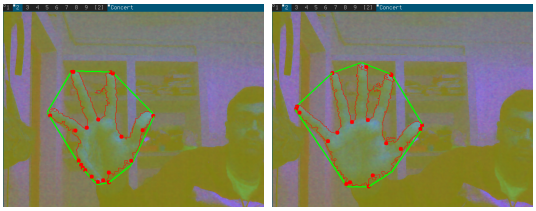

Fig. 7. Gesture 3 & Gesture 4

V. CONCLUSION

Though there have been systems like [3] which do offer musical generation, such systems may have heavy hardware requirements and computational requirements for running machine learning modules. Such requirements may not be available all the time, and hence simplistic models like the proposed system could be used for small scale productive musical generation. As discussed earlier, the proposed system's algorithms could be easily extended to include more number of gestures using rule based learning.

VI. REFERENCES

[1] Vladimir Vezhnevets, Vassili Sazonov, and Alla Andreeva, "A survey on pixel-based skin color detection techniques," in *Proc. Graphicon*. Moscow, Russia, 2003, vol. 3, pp. 85–92.

[2] Douglas Chai and King N. Ngan, "Face segmentation using skin-color map in videophone applications," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 9, no. 4, pp. 551–564, 1999.

[3] André Baltazar, Lg Martins, and Js Cardoso, "ZATLAB: A Gesture Analysis System to Music Interaction," *6th International Conference on Digital Arts (ARTECH 2012)*, 2012.