IMPROVING UPON MUSICAL ANALYSES OF CONDUCTING GESTURES USING COMPUTER VISION

Teresa Marrin Nakra
Departments of Music and Interactive Multimedia
The College of New Jersey

Daniel Tilden
Department of Computer Science
Virginia Polytechnic Institute

Andrea Salgian
Department of Computer Science
The College of New Jersey

ABSTRACT

For more than ten years, researchers have been developing software-based methods for analyzing the movements of orchestral conductors. Beginning with the *Conductor's Jacket* research project, researchers have combined wearable sensors with face-on video recordings to improve tracking and understanding of the structure inherent in the gestures. Building upon more recent work employing computer vision, the current project refines the methods for tracking the hands and visualizes the data in a way that reveals more of the underlying structure in an easier-to-view fashion. Such improved methods will enable more specific understanding of the functions of conducting gestures, and allow for more concrete applications of those gestures to interactive conducting systems for public exhibits, video games, and integrating dynamic audio enhancements to live concerts.

1. INTRODUCTION

The quantitative analysis of musical conducting gestures has a range of applications for use in video games, interactive exhibits, and other computer-based musical experiences [1,2]. Since 1997, wearable sensors have been used to characterize conducting gestures from a quantitative perspective; the *Conductor's Jacket* [3] featured a combination of wearable physiological sensors with face-on video recordings to improve tracking and understanding of the expressive structure inherent in conductors' movements. While video recordings were initially taken for archival purposes, it soon became clear that they were essential for analyzing the data afterwards, to be able to reference and align the data with specific gestures or events in the score.

In more recent collaborations with computer vision researchers [6,7], a new analytical technique was created, enabling a more detailed view of aspects of conducting gestures. For example, the new visualization method enabled a finding that the height of the conductor's right hand correlated with the tempo of the performance. Expressive and structural features were also discovered and documented, including "height deltas," tiered gesture platforms, and smooth versus jagged beat shapes [6]. The current project refines the hand-tracking method in [6] and visualizes the data in a way that reveals more of the underlying structure in an easier-to-view fashion.

2. BACKGROUND

Perhaps one of the most promising areas to which computer vision has been applied is gesture recognition – tracking and responding to movements of the hands. This concept has been applied to musical conducting as far back as 1991, when a team at Waseda University built a sensor glove and camera system that tracked the motions of a conductor, allowing that person to conduct and control a MIDI-based synthetic orchestra [4].

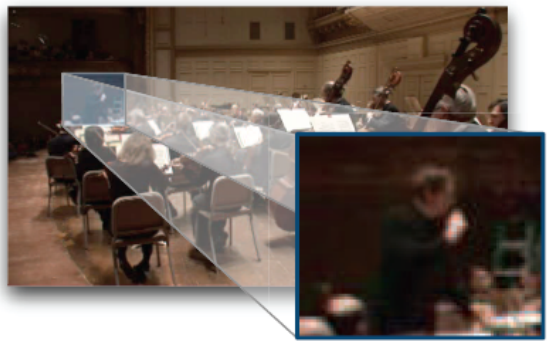

Figure 1: The 2006 video and its cropped form [6]

Since 1991, numerous other large projects involving computer vision and conducting have been undertaken. In 2008, a team at TCNJ built a Matlab system application that tracked conducting gestures from video footage of a professional conductor [6,7]. The video had been taken to document earlier *Conductor's Jacket* experiments with the Boston Symphony Orchestra. However, due to the source video's poor quality and low resolution (see Figure 1), the tracking would intermittently lose track of the hand's position [7].

Figure 2: A frame from the new video. Note that it is clearer and taken from a more useful perspective.

It was concluded that the best solution to the quality problem would be to record an entirely new video for the specific purpose of motion tracking. With that in mind, a new recording was made, featuring the conductor of the TCNJ student Orchestra. The video showed a view from the orchestra's perspective, in which the conductor's image filled the frame (see Figure 2). Using the improved video, an improved tracking program was created to track the conductor's hand movements.

Figure 3: In some frames, the tracker would lose track of the hands.

3. IMPLEMENTATION DETAILS

The new tracking system features three major enhancements: increased accuracy and robustness, added beat detection, and real-time capabilities in C++. Early problems were encountered when the tracker would mistakenly assume that an object in the foreground (i.e. part of the orchestra) was one of the conductor's hands (see Figure 3). We were able to rectify this problem by employing background subtraction to the area of the video where the orchestra was. To accomplish this, we took the average of every frame in the video sequence (see Figure 4) and subtracted it from the lower portion of the video sequence. This greatly enhanced the tracking accuracy.

Figure 4: The average of every frame in the sequence, used to calculate the areas that comprise the 'background'.

Secondly, beat detection was added. This was done using *MEAPsoft* [8], a program that automatically segments and rearranges portions of music recordings. One of its functions outputs detected beats to a text file (see Figure 5). Using this data, we were able to parse and overlay the beats, graphed as red lines, on the tracked output video (see Figure 6).

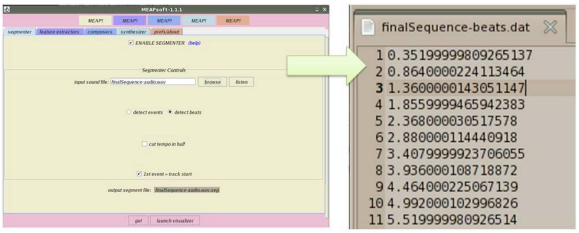

Figure 5: Using MEAPsoft, we were able to generate a series of time markers denoting the beats.

Finally, we ported the program from MATLAB to C++. While MATLAB was a convenient environment in which to develop the program, it was too slow for production; tracking a 3-minute video sequence took approximately five hours. C++ was selected, because it runs in real-time

and has several image processing libraries available. We also decided to use the OpenCV image-processing library [9], because it functions as open source under the BSD license, and it also supplies a wide variety of highly optimized image processing and matrix operations, making it ideal for porting from MATLAB. Implementing the C++ version proved to be reasonably straightforward; most of the algorithm could be ported directly from MATLAB.

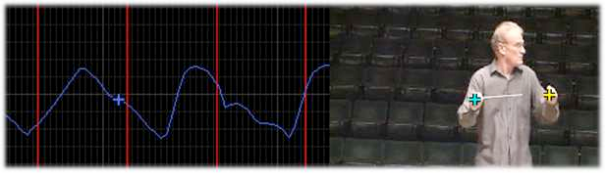

Figure 6: The beats (each represented by red lines), overlaid onto the plot of the hands.

4. SYSTEM FEATURES

Features of the tracking program include skin detection, blob detection, limb identification, and the alignment of gestures with musical features. Our first step was to identify which pixels in any given frame belonged to skin. To do this, we utilized a series of RGB color thresholds, taking into account the ratio of each color component's intensity to the other colors. (While this method worked well for this particular sequence, it is not generally reliable for variable lighting or skin tone conditions.)

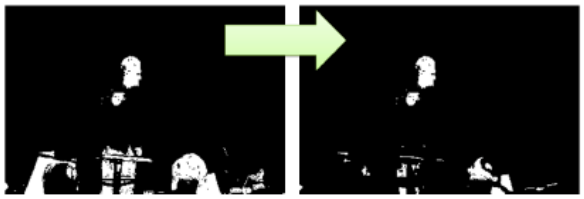

Figure 7: Skin detection allowed us to subtract some of the complicating elements in the image.

After the skin was detected, the algorithm identified the positions of the conductor's left and right hands. Further testing indicated that keeping track of the position of the subject's head minimized the risk of the program "seeing" the face as a hand. The program then made sense of the "skin" pixels by using "blob detection"; each contiguous area of skin pixels was determined to be a "blob". We used a technique that employed the *flood fill* function in OpenCV. However, this technique only provided a bounding rectangle for each blob. Using the center of the estimated rectangle, as opposed to the true centroid of a detected blob, caused the tracking to be too erratic. (For example, the conductor's baton could greatly increase the size of a blob's bounding box, causing it to move significantly.) This problem was remedied by calculating the true centroid of the blob, taking the average values of the x and y coordinates of the pixels belonging to the blob.

Next, to perform limb identification, the algorithm checked to see if there were exactly three limbs detected. If so, it assumed that the largest blob was the head, the left smaller blob was the left hand, and the remaining blob the right hand. Jump detection was not used; even though this caused some hand-crossing problems, it made the algorithm more robust overall, and much less prone to error. If the program did not locate exactly three limbs, it looked to previous frames for ideas, with which it made a "best guess" of the wayward limb's position. If there were too many candidate blobs detected, the program examined the previous frame's recorded blob positions and compared their distances to previous positions.

Figure 8: The output of the tracking algorithm. The head and hand positions are shown as colored squares.

It then assumed that the candidates with the smallest distances were the limbs. Once the program located the limbs, it then used linear projection to determine where the position of that limb would be if it continued its motion from the previous two frames. Based on testing, this projection was limited to a specified distance, after which the position will simply be frozen in place. Without this limit, limbs had a tendency to project at high speeds out of the frame, which was problematic.

5. CONCLUSIONS

The video created by this tracking system is available online in AVI format; it can be accessed at this address:

http://www.tcnj.edu/~nakra/TCNJconductvis.avi

We feel that the visual layout of the vertical and horizontal movements of the two hands, overlaid with information about the beat occurrences, enables a kind

of visual analysis of conducting that is new and informative. Several musical features are revealed and aligned in a real-time format, resulting in improved possibility for musical analyses. Also, the ability to see the scrolling recent history allows for backwards comparison over several beats.

Movement tracking also helps analyze other data by giving quantitative ground truth about other factors such as heart rate. From the tracked video, one can also draw several conclusions. Firstly, it seems that the x-position of the hand is not impacted much by the rhythm of the music; the graphs of that axis are quite erratic compared to the even, steady beats visible along the Y-axis of the left hand. It is also interesting to note when the right hand mirrors the beat, and when it does not. Furthermore, from the overlaid beat plots, one can clearly note that there is a strong correlation between the beat gestures in the right hand and the beats of the audio.

6. FUTURE WORK

The current algorithm does not generalize to different lighting conditions, due mainly to the method used for skin detection. While the algorithm works very well for the current situation, it currently cannot be generalized. The implementation of a different skin detection algorithm would address this problem.

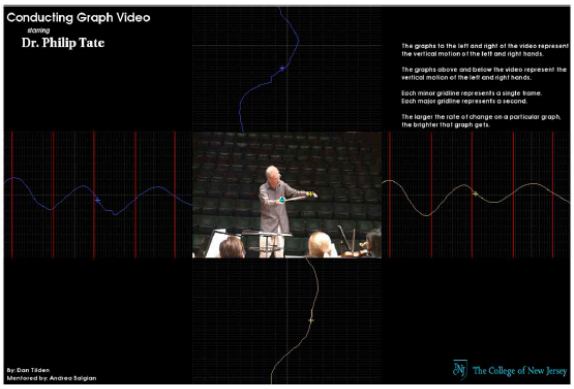

Figure 9: A frame from our video sequence, with beat lines shown in red.

We would also like to enhance the real-time analysis abilities of this system. The current tracking algorithm is not designed to track footage from a real-time source (such as a webcam or a firewire camera), but this can be easily achieved using functions from the OpenCV library.

We hope that this tool will be useful to future researchers to enable more quantitative and specific analyses and to improve on the performance of interactive systems for the public.

7. ACKNOWLEDGEMENTS

The authors would like to thank Dr. Philip Tate, Assistant Professor of Music at The College of New Jersey, for allowing us to videotape him while conducting. We would also like to thank the student Orchestra at The College of New Jersey, for allowing us to sit in on their rehearsals. Finally, we would like to thank the Mentored Research program at TCNJ for enabling this student-faculty collaboration.

8. REFERENCES

[1] Ilmonen, T. "The Virtual Orchestra performance," in CHI'00: Proceedings of the SIGCHI Conf. on Human Factors in Computing Systems, 2000. ACM Press.

[2] Lee, E., Nakra, T.M., and J. Borchers. "You're the Conductor: a Realistic Interactive Conducting System for Children," In NIME'04: Proceedings of the 2004 Conference on New Interfaces for Musical Expression, pp. 68–73, Singapore, 2004.

[3] Marrin, T. and Picard, R. "The Conductor's Jacket: a Device for Recording Expressive Musical Gestures," International Computer Music Conference, Ann Arbor, MI, 1998, pp. 215-219.

[4] Morita, H., Hashimoto, S. and Ohteru, S. "A Computer Music System that Follows a Human Conductor," Computer, 24(7): 44–53, 1991.

[5] Nakra, T.M. "Inside the Conductor's Jacket: Analysis, Interpretation and Musical Synthesis of Expressive Gesture. PhD thesis, MIT, 2000.

[6] Nakra, T.M., A. Salgian, and M. Pfirrmann. "Musical Analysis of Conducting Gestures Using Methods from Computer Vision." International Computer Music Conference, Montreal, 2009.

[7] Salgian, A., Pfirrmann, M. and T.M. Nakra. "Follow the Beat? Understanding Conducting Gestures from Video." 3rd Int. Symposium on Visual Computing, Lecture Notes in Computer Science, Springer, 2007.

[8] MEAPsoft – Computers Doing Strange Things with Audio: www.meapsoft.com

[9] Open Computer Vision Library on Sourceforge: http://sourceforge.net/projects/opencvlibrary/