A Sonification Method using Human Body Movements

Felix Albu
Departments of Electronics
Valahia University of Targoviste
Targoviste, Romania
email: felix.albu@valahia.ro

Mihaela Nicolau, Felix Pirvan, Daniela Hagiescu
R&D Department
Advanced Slisys SRL
Bucharest, Romania
email: aslisys.121@gmail.com

***Abstract*—In this paper, we propose a new method to generate piano music starting from body movements recorded with a webcam. The joint coordinates found by using the convolutional pose machine method are used to calculate the pitch and velocity of the produced consonant or dissonant chords.**

Keywords- Sonification; generative music; convolutional pose machine; computer vision.

I. Introduction

In principle, distinct information or data sets in various fields, such as geology, medical research, or financial markets, can be perceived using the ears by sonification instead of studying large tables or graphics [1]. The principle underlying all sonication techniques is the arbitrary mapping of input data in the auditory domain.

One of the first sonification systems was the Unité Polyagogique Informatique CEMAMu (UPIC) system [6] that generates complex sounds by writing on a screen with a digital pen. Similar systems have been implemented in the AudioSculpt [7] and SPEAR software [8]. These applications reshape modified images in sounds. Other software solutions are EyesWeb [9], and Max/MSP/Jitter [10]. The available technology has allowed the study of sounds without being limited to a system based on symbols, such as the Western music notation [1].

Among first attempts to generate sounds from body movement were those made by using sensors attached to the body or extracting movement from video recordings [1]-[5]. A relevant information can be obtained from the spatial or temporal content of the movement. It is known that the movement extraction using camcorders or webcams are less precise than those using body-attached sensors. On the other hand, this approach is much cheaper, simpler, less obtrusive and offers the ability to make recordings in various locations. Recent approaches of using motion detection to control the real-time sound generation are the Motion Composer [12] or Point Motion Control [13] devices. In [14], an application that translates the perceived movement of the scenery such as passing trains, into music is presented. The continuously changing landscape view outside of the train window was captured with a camera and translated into Musical Instrument Digital Interface (MIDI) events that are replayed instantaneously. A sonification method that incorporates both spatial information and color distribution properties of captured video frames was implemented in [14]. Unfortunately, this method cannot be adapted for body movements because the background is typically static.

One of the most promising techniques for creating music based on the movement of the human body uses "motiongrams" [5]. A "motiongram" is a visual representation of movement based on the difference between successive frames. The visual similarity between motiongrams and spectrograms is exploited by transforming motiongrams into sounds through an inverse Fast Fourier Transform (FFT) [5]. Therefore, an image is treated as the spectrogram with frequency information on the Y axis and time on the X axis as the basis for synthesizing a sound file. Unfortunately, the method is too complex, involving multiple FFT processing [5].

Another promising sonification approach based on heavily numerically complex optical flow computation has been presented in [15]. A musical note is played when a local peak in the optical flow magnitude is higher than a threshold. The pitch corresponds to the location and flow direction of the peak and the velocity (or intensity) of the note corresponds to the magnitude [15]. A fish bowl was filmed and consonant chords were generated when fish were near one another and moved in approximately the same direction [15]. In a proposed alternative for static images the Hue, Saturation and Value (HSV) color space was used.

In this paper, we propose a novel and computationally simpler method based on computer vision techniques that uses the body joint coordinates found by Convolutional Pose Machines (CPM) from [16] and a modified sonification method. To the best of our knowledge, the CPM method has not been previously used for sonification of captured body movements. Another original part of this work is the approach to use computed joint coordinates, avoiding outliers and generate aesthetically pleasing piano sound starting from a layout of pitches using low-level harmonic notions proposed in [15]. Also, the proposed method does not use the HSV color space or optical flow computation methods.

The rest of this paper includes Section II that describes the proposed method, while the acknowledgement, conclusions and future work close the article.

II. THE PROPOSED METHOD

The scheme of the proposed method is shown in Figure 1.

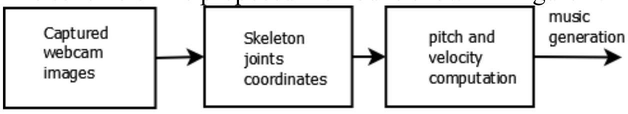

Figure 1. The scheme of the proposed method

A. *Skeleton joints coordinates computation*

The skeleton joints coordinates are found by using the CPM [16] on captured webcam images (see Figure 2). The image from Figure 2 has a width of 491 pixels and a height of 368 pixels.

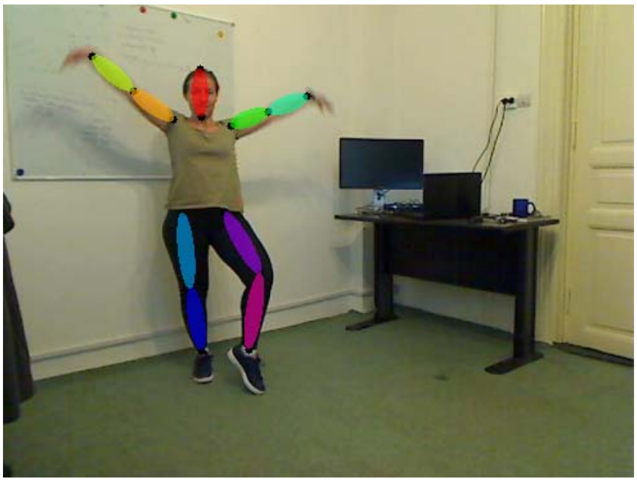

Figure 2. A captured image with overimposed computed joints coordinates.

The convolutional pose machine is a human pose detector. The algorithm produces the coordinates for the following 14 human skeleton joints: head, upper neck, shoulders, elbows, wrists, hips, knees and ankles. An example of skeleton joints is shown in Figure 3. The CPM uses two convolutional neural networks, one to detect the persons present in an image and the other to detect person's skeleton joints. The networks were trained on several public datasets [16]. Each network is composed of a sequence of several stages. Each stage produces belief maps that are supervised within each stage, thus addressing the vanishing gradients problem, inherent to deep neural networks. Each stage is composed of a sequence of convolution and pooling layers. The convolutions capture local features of the size of the convolution kernel (5x5, 9x9, and 11x11 kernels are used), while the pooling layers downscale the image by a factor of 2. The effect of pooling is that the subsequent convolution will operate on a less-detailed version of the image, capturing features on a bigger scale.

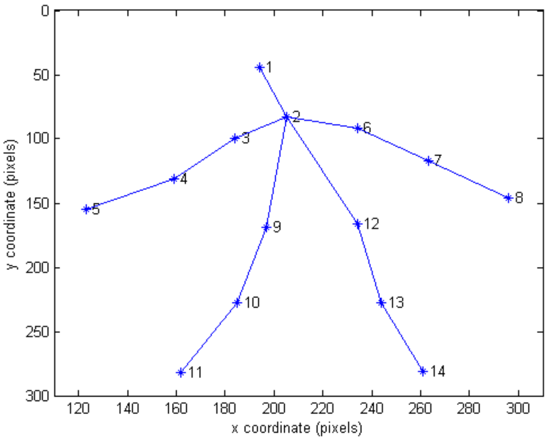

Figure 3. The computed skeleton joints.

The CPM method achieved state of the art accuracy on all primary benchmarks [16] and does not need special expensive equipment like Kinect devices. It has been reported that the CPM has some failure cases when multiple people are in close proximity [16]. However, this is not our case, since only one person is expected to dance in front of the camera. Therefore, the proposed framework is salient enough for the proposed model. More details about the CPM method can be found in [16].

Figure 4 shows the normalized vectors of y-coordinates evolution in time for various body joints. The blue curve shows the head coordinates, the red curve shows the left shoulder coordinates, and the green curve shows the right knee coordinates.

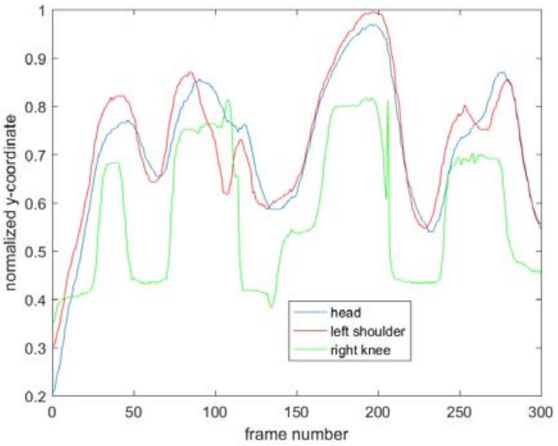

Figure 4. Examples of vectors obtained from the coordinates of the various body joints.

As expected, there is a rather close similarity between the normalized vectors of head and shoulder due to the physical constraints of human body joint movements. It is also obvious that the correlation between the knee coordinates and head/shoulder coordinates is much lower.

The CPM method is able to handle non-standard poses and solve ambiguities between symmetric parts for a variety of different relative camera views [16]. However, there are few failure cases that can appear when there is a sudden change of coordinates (see the green curve from Figure 4). These outlier coordinates can be removed by using the Dynamic Time Warping (DTW) distance between consecutive normalized joint coordinates vectors [17]. The DTW algorithm is a well-known algorithm that computes the optimal alignment between two time-series. It has been used in many applications such as speech recognition [17], handwritten evaluation [18][19], etc. In the classical DTW algorithm, a two-dimensional cost matrix is formed and its elements are the minimum accumulated distances for the sequences time series. More information about the DTW algorithm can be found in [17]. In Figure 5, the histogram of the DTW distances between consecutive vectors of coordinates for a dancing performance is shown. The elements of the vector containing the DTW distances are sorted into 64 equally spaced bins between its minimum and maximum values. It can be easily seen from Figure 5 that most of the time the DTW distances between the coordinates computed from consecutive frames are rather small. This is expected, because in most dance movements, there is not a very fast variability of joints positions in time. It can be noticed that a threshold set to 0.05 can reasonably detect outliers. If the DTW distance between two consecutive vectors is higher than 0.05, the particular vector is not taken into account in order to generate music. Generally, about 5-10% of vectors are ignored and usually these vectors are generated by faulty coordinates provided by the CPM block. The DTW distance was preferred to the Euclidian distance due to its better clustering properties and robustness to outliers.

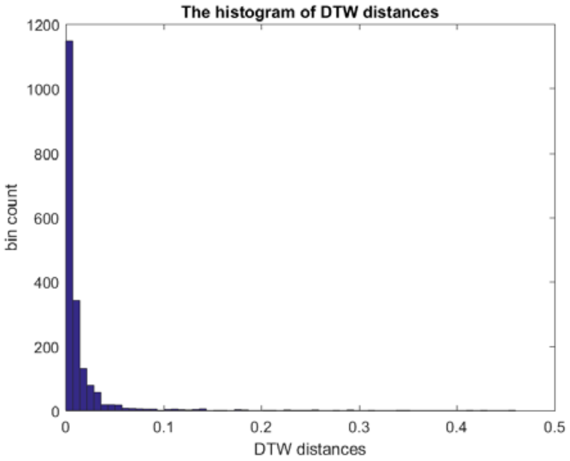

Figure 5. The histogram of DTW distances between consecutive vectors of coordinates.

B. Pitch and velocity computation

The pitch mesh pairs method from [15] is adapted for our approach. The selected normalized joints coordinates are used instead of the pixel values in the RGB or HSV color spaces as proposed in [15]. The pitch mesh consisted on vertically stacked pitch chains in parallel octaves [15]. The musical tone is generated with the "dominance" bit and two real numbers, pitch and register respectively. By increasing the range of pitch, *x*, the amount of chromaticism and dissonance is increased, while increasing the register, *y*, the broadness of the register of the generated notes is increased [15]. There is a correlation between the coordinates vectors and if they move in the same direction, the generated sound is basically consonant, while changes trigger functional changes in harmony. The chromaticism of the generated music is altered by modifying the thresholds for the normalized coordinates [15]. With a very low range of *x*, the notes may all emerge from the same tetrachord, whereas with a very high range, the piece could sound fundamentally atonal [15].

An example of generated pitch and velocity values for 50 frames is shown in Figure 6. The threshold and beat duration were set to 0.5, the pitch range was 7 and the register range was set to 3. The pitch and velocity values were scaled from -1 to 127. The size of the pitch and velocity vectors depends on the result of comparison with the threshold (e.g., it is 896 for Figure 6).

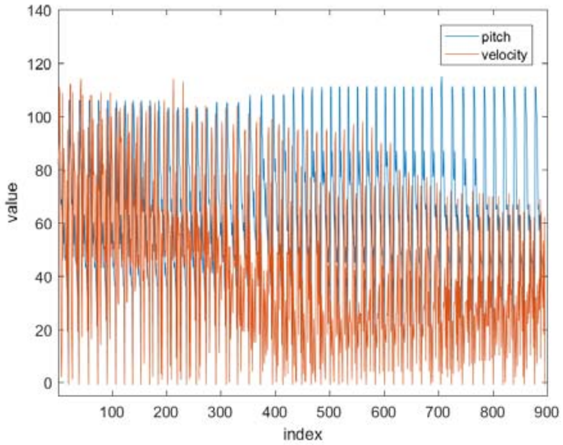

Figure 6. Computed pitch and velocity values.

The generated music contained many times two notes on the same pitch within a distance of each other. Therefore, in this case, the abruptly repeated notes were removed by retaining the note with the higher velocity. The effect of the distance on the filtered pitch and velocity parameters can be seen on Figures 7 and 8. The distance parameter was set to 3 for Figure 7 and 9 for Figure 8, respectively.

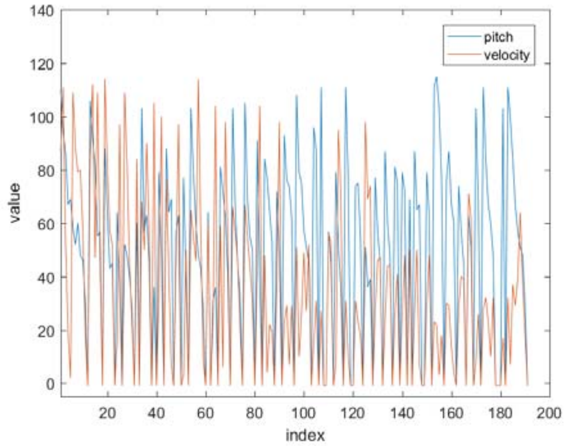

Figure 7. Filtered pitch and velocity for a distance of 3.

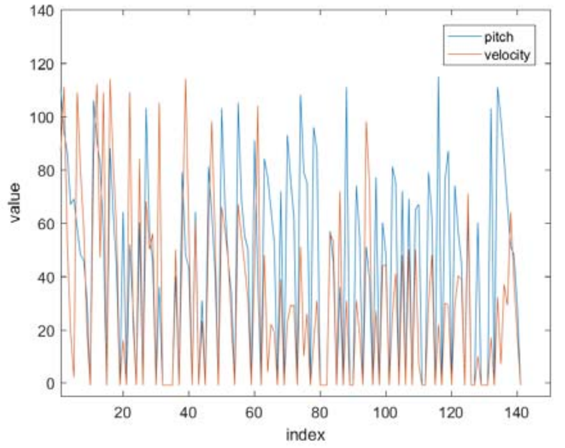

Figure 8. Filtered pitch and velocity for a distance of 9.

The size of the pitch and velocity vectors is reduced a lot (e.g., it is 191 for Figure 7 and 141 for Figure 8). Although the filtered parameters for a distance of 9 seems to be shrinked version of those obtained using a distance of 3, we've found by listening the generated music that a distance of 3 gives slightly more aesthetically pleasing sounds. These filtered parameters were used to generate the piano music in Matlab by employing the Microsoft MIDI Mapper [20] as the midi output device.

The complexity of using the human pose detector based on CPM is much smaller than that of computing the optical flow on successive frames. Also, our proposed sonification approach uses only 14 joints coordinates per frame and simple mathematical operations and comparisons for music generation. The complexity of the methods proposed in [5] [14] and [15] is at least two orders of magnitude higher because they use very numerically intensive and complex operations on frames with full or low resolution.

The pitch and velocity parameters computed using the method from [15] are shown in Figure 9.

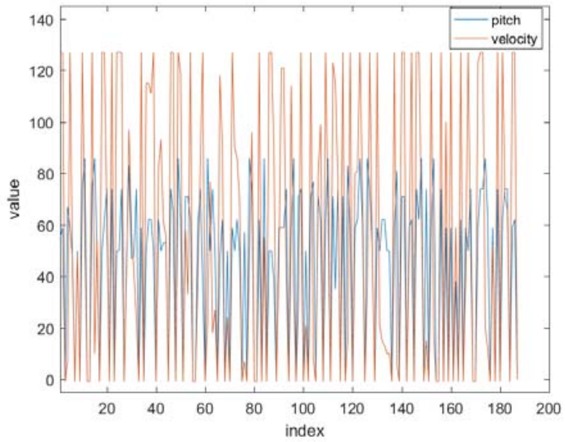

Figure 9. The pitch and velocity parameters using the method from [15]

It can be noticed that the average pitch value of our method is higher than that of [15]. Also, the average velocity value of the proposed method is smaller than that of [15]. The difference from the parameters shown in Figure 7 and Figure 9 can be explained by the fact that the optical flow computation method can get more movement information over the all the body, not only that of the joints tracked by our method. Our generated music sounds differently than that obtained by the method of [15]. However, different notes and piano music feeling can be obtained by varying the threshold, pitch range, register range and beat duration parameters of the proposed skeleton joints coordinates based method.

III. Conclusion and Future Work

A piano music generating method from dance movement using a human pose detector, dynamic time warping and pitch pair mesh approaches is presented. The proposed technique is simple to implement, does not need special equipment and sensors and has the potential to generate aestethically pleasing sounds. Future work will be focused on optimization of the parameters of the proposed method in order to open new perspectives of soundtrack generated from body movements.

Acknowledgment

The results presented in this work concerns the research carried out for the "SENTIR" research project, co-financed through the European Regional Development Fund, POC–A.1–A1.2.1–D–2015 grant, research, development and innovation supporting economic competitiveness and the development of businesses.

References

[1] T. Hermann, A. Hunt, and J. G. Neuhoff, The Sonification Handbook, Logos Verlag, Berlin, 2011.

[2] M. M. Wanderley, B. W. Vines, N. Middleton, C. McKay, and W. Hatch, “The musical significance of clarinetists’ ancillary gestures: an exploration of the field,” Journal of New Music Research, vol. 34, no. 1, Feb. 2007, pp. 97–113, doi: 10.1080/09298210500124208.

[3] A. R. Jensenius, “Action-Sound: Developing methods and tools to study music-related body movement,” PhD dissertation, University of Oslo, 2007.

[4] R. M. Winters, A. Savard, V. Verfaille, and M. M. Wanderley, “A sonification tool for the analysis of large databases of expressive gesture,” The International Journal of Multimedia and its Applications, vol. 4, No. 6, Dec. 2012, pp. 13-26, doi: 10.5121/ijma.2012.4602.

[5] A. R. Jensenius and R. I. Godøy, “Sonifying the shape of human body motion using motiongrams,” Empirical Musicology Review, 7(3), Aug. 2013, pp. 73-83, doi:10.18061/emr.v8i2.

[6] G. Marino, M. H. Serra, and J. M. Raczinski, “The upic system: Origins and innovations,” Perspectives of New Music, Vol. 31, No. 1, Jan. 1993, pp. 258-269, doi: 10.2307/833053.

[7] AudioSculpt software [Online]. Available from http://anasynth.ircam.fr/home/english/software/audiosculpt 2017.11.06

[8] Spear software [Online]. Available from http://www.klingbeil.com/spear/ 2017.11.06

[9] A. Camurri et al., ”Eyesweb: Toward gesture and affect recognition in interactive dance and music systems,” Computer music Journal, vol. 24, No. 1, Mar. 2000, pp. 57-69, doi: 10.1162/014892600559182.

[10] M. Wright, R. Dudas, S. Khoury, R. Wang, and D. Zicarelli, "Supporting the Sound Description Interchange Format in the Max/MSP Environment", Proc. of the Int. Computer Music Conference (ICMC), Oct. 1999, pp. 1-4, doi: 10.1.1.30.6737.

[11] A. R. Jensenius, “Some video abstraction techniques for displaying body movement in analysis and performance,” Leonardo, Vol. 46, No. 1, Jan. 2013, pp. 53-60, , doi: 10.2307/23468117.

[12] Motioncomposer device [Online]. Available from http://motioncomposer.de/ 2017.11.06

[13] Pointmotioncontrol software [Online]. Available from http://www.pointmotioncontrol.com/ 2017.11.06

[14] T. Pohle and P. Knees, “Real-Time Synaesthetic Sonification of Traveling Landscapes” Proc. of 5th Int’l Mobile Music Workshop (MMW), May 2008, pp. 1-3, doi:10.1145/1459359.1459592.

[15] A. M. Taylor and J. Altosaar, "Sonification of Fish Movement Using Pitch Mesh Pairs" Proc. of the Int. Conf. on New Interfaces for Musical Expression (NIME), Jun. 2015, pp. 28-29.

[16] S. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh. “Convolutional Pose Machines” Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), IEEE Press, Jun. 2016, pp. 1-9, doi: 10.1109/CVPR.2016.511.

[17] C. S. Myers and L. R. Rabiner, "A Comparative Study of Several Dynamic Time-Warping Algorithms for Connected-Word Recognition," Bell System Technical Journal, vol. 60, No. 7, Sep. 1981, pp. 1389–1409, doi:10.1002/j.1538-7305.1981.tb00272.

[18] F. Albu, D. Hagiescu, M. A. Puica, and L. Vladutu, "Intelligent tutor for first grade children’s handwriting application", Proc. of 9th International Technology, Education and Development Conference (INTED), Mar. 2015, pp. 3708–3717, doi: 10.13140/RG.2.1.2591.7607.

[19] F. Albu, D. Hagiescu, and M. A. Puica, “Quality evaluation approaches of the first grade children’s handwriting”, Proc. of the 10th International Scientific Conference on eLearning and software for Education (ELSE), Apr. 2014, pp. 17-23, doi: 10.12753/2066-026X-17-055.

[20] Microsoft Windows MIDI Mapper Help [Online]. https://support.microsoft.com/en-us/help/84817/using-the-midi-mapper 2017.11.06.

ISBN: 978-1-61208-611-8