USING GESTURE DATA TO GENERATE REAL-TIME GRAPHIC NOTATION: A CASE STUDY

Gil Dori
Phonos Foundation
Barcelona, Spain
gil.dori@asu.edu

ABSTRACT

The paper describes the concepts and compositional process of my recent real-time graphic score work *Arcos*, for cello and augmented violin bow. The work's graphic notation is generated directly from gesture data of various bowing techniques. A Myo armband was used to record cello bowing data, and the augmented bow's own position tracking module was used to record its motion data. After processing the data, gestures were visualized on the screen, as a form of real-time graphic notation based on imitation. This notational approach—a low-level symbolic representation of gestures—allows for an immediate, intuitive interpretation on the spot, and provides an instantaneous connection between notation and action. My work also offers a distinct perspective on notation for electronic instruments in the context of real-time action-based scores.

1. INTRODUCTION

1.1. Real-Time Action-Based Notation

One approach to digital scores with notation that is "created or transformed during an actual musical performance" [1, p.1], i.e. real-time scores, focuses on representation of performance actions. This so-called action-based notation, albeit not restricted to real-time scores, helps to facilitate sight-reading and to provide the performer with easily available information for interpretation on the spot. These are crucial features for the practice of real-time scores. Gerhard Winkler, a real-time notation pioneer, writes:

> Different parts of the score have to be reduced to a number of elements, which can be learned and 'trained' in advance, and which can be seized with 'one glance' immediately during a performance. On the other hand the used signs have to be precisely [sic.] enough to avoid that the musicians shift into 'improvisation' [2, p. 2].

Furthermore, he emphasizes the importance of creating an environment in which performers and computers can communicate and influence one another in a "complex, non-linear way", [2, p. 1].

Winkler maintains that combining graphic notation and standard, common-practice notation is the most efficient way to create real-time scores. For example, his work *KOMA* (1995-96), for string quartet and interactive live electronics, features a score that changes in real-time according to decisions the leader of the quartet makes at the moment of performing. The score combines different types of notation, such as common-practice notation to indicate pitch; graphic notation to indicate glissandi; and text to indicate microtonal changes in cents [2, pp. 2-3].

Seth Shafer continues Winkler's approach with *Terraformation* (2016-17), for viola and computer. Its real-time score combines a five-line staff for common-practice notation; tablature for fingering positions; and color gradients for bow position and left-hand finger pressure [3, p. 4]. By applying physiological models of performance actions, Shafer designed a score that facilitates the "cognitive translation from graphic representation to bodily action" [3, p. 3]. The algorithm that drives the score takes into account these physical action models when selecting what musical material to present at a given moment. This action-based procedure guarantees to produce idiomatic material that is easy to sight-read, yet still allows for wide sonic variety and depth [3, p. 6]. However, while Shafer intends to balance the cognitive strain on the performer, the score interpretation process appears to be time-consuming and unintuitive. In his paper about *Terraformation,* Shafer relays violist Mike Capone's experience interpreting the piece. This process involves several stages, from gathering information from the different parts of the score to executing the performance actions [4].

On the other hand, graphic notation by itself could provide sufficient information to the performer, especially when used to indicate simple actions. Ryan Ross Smith, for example, developed an idiosyncratic graphic action-notation language which employs radial motion to specify points of attack (and sometimes sustain) in time. Albeit its apparent simplicity, this notational approach allows for complex rhythmical procedures, from hocket to polytemporality. Smith also applies ideas of imitation for action notation. For example, the scores of his works *Study no. 15b* (2013), for cello or upright bass, and *Study no. 15b.1* (2013), for violin or viola, show the fingerboard of the instrument, with circles that indicate where to press the strings [5, 6].

Particularly interesting to this paper is Smith's *Study no. 50* (2015), for a percussion ensemble, in which each performer plays on seven wooden planks. The real-time score shows attack cursors that travel along arcs between seven nodes. One node is assigned to each plank, and the attack cursor indicates when to play which plank. The score is projected downwards from above the ensemble, so that the nodes are positioned directly over the planks. Projecting the score in such fashion "creates a direct correspondence between the notation and the instrument" [7, p. 6]. The performers, then, can interpret the score instantaneously by imitating the motion they see. It is, however, a one-way communication between computer and performers. The algorithm that drives *Study no. 50* generates the notation in real-time automatically and randomly, without any external influences nor an overall compositional structure, as Smith strives for an infinite "consistently-inconsistent flow of events" [7, p. 6].

I set forth to design an intuitive graphic notation environment by recreating actions on the screen, directly from gesture data, that the performers can imitate. This approach allows to extend the amount of information the performers receive, while maintaining a simple, easy to sight-read notation. It provides an instantaneous connection between notation and action through a low-level symbolic representation of gestures. In addition, performer's actions influence the score in real-time, and every action may produce different outcomes, consequently forming a non-linear interaction between performer and computer. This interaction is further enhanced by composing for an electronic instrument. Doing so, however, necessitates special attention in designing this real-time action-based score.

1.2. Gesture and Morphology

Action-based scores for acoustic instruments rely on the inherited, physical interrelation between gesture and sound. The composer who notates actions presupposes that the physical properties of a gesture affect the sound directly. An example for this link between gesture and sound would be a guitarist who wants to produce a loud, bright sound, that encompasses a wide frequency range. To do that, the guitarist would forcibly strum all the strings in one motion, close to the bridge, with the fingernails or with a plectrum. On the contrary, to produce a soft, warm, low frequency sound, the guitarist would pluck the lowest string gently with the inner part of thumb, close to the fingerboard. Thus, as Garth Paine notes, the "physical gesture determines the amplitude, pitch, and timbre of each [sound] event" [8, p. 1]. Trevor Wishart, in his seminal book *On Sonic Art,* coins the term sound morphology to describe this link between action and sound objects, the "gestural structure" of sound, and how this interrelationship manifests in time [9].

Live electronic performance poses a unique problem in regards to morphology, as the "traditional links with physical sound-making are frequently ruptured" [10, p. 1]. Kim Cascone even characterizes live laptop performances as sound created in a non-existent place, with a fake sense of authenticity [11]. A solution to this issue lies within gestural sound control and embodied interaction with sound, using what is generally referred to as new interfaces for musical expression (NIME). However, to provide the most convincing sense of authenticity in live electronic performance, such interfaces must take into account vital gestural elements that affect the sound, such as pressure, speed, and position [12].

My composition is written for an acoustic instrument, cello, and a hybrid instrument, which combines a violin bow with a digital sound processing component. For lack of a better term, I refer to this instrument as an augmented violin bow. Composing a real-time action-based score for such an electronic instrument raises challenges not only regarding morphology in electronic music performance, but also regarding expressing electronic instrument's morphology in notation. The gesture data approach to graphic action-based notation presented in this paper offers one possible way to address these challenges.

2. COMPOSING ARCOS

2.1. Background

Arcos was composed during my residency at Phonos Foundation in 2019. The idea for a real-time score that is based on gesture data formed after premiering my piece *(bcn)621* (2018), for cello, percussion, and laptop, which is unified by bowing actions. All of the performers in this piece are using bows, playing similar and related gestures. For the laptop part, it meant creating a fitting interface for performing that will provide the same performance abilities and relationship between action and sound as the cello and the percussion do. Thus, I built an augmented violin bow, guided by notions of sound morphology in electronic music performance. Naturally, in this context of an action-based composition for an ensemble, these notions have an even greater significance. Section 2.3, about instrumentation, elaborates more about the augmented violin bow.

The composition of *(bcn)621* uses isorhythm techniques, applied to actions within blocks of time, in order to shape a perceivable structure through timed sections of actions. The parts consist of a fixed series of gestures, each within a defined time frame. The series repeats itself in the same order but in different durations. Changes in the overall duration of the sequence affect the durations of the individual time frames proportionally. For example, the cello sequence begins with playing tremolo, then resting, then scraping the bow along the strings. The total duration of the first instance of the sequence is 45 seconds, with playing tremolo for 4 seconds, resting for 3 seconds, and scraping for 6 seconds. When the sequence repeats for the third time, its overall duration is 90 seconds. Then, the same gestures take 8 seconds, 6 seconds, and 12 seconds respectively.

Although *(bcn)621* was successful in terms of achieving the compositional and conceptual goals it aimed for, there were practical issues that complicated rehearsing and performing the piece. I believe that the gesture-based real-time score of *Arcos* offers a solution for these issues.

The main problem was maintaining the accurate times of the sections. The temporal units are measured in seconds, indicated by a tempo mark of 60 beats per minute. The interface of the digital instrument shows a timer, cues count, and automatic on-screen instructions. The cello and percussion performers, however, had to constantly count beats in order to make sure they execute actions in the exact times and for the right durations. This need for endless counting was very stressful for the performers, and diverted their attention away from performing the piece as an ensemble in a musically sensible way. To relieve the performers, I made a visual click track, a Max patch, that shows beats and bar numbers[1]. While this patch helps immensely, and allows for a smooth performance, it brings additional technical requirements that could be cumbersome, such as mounting an auxiliary screen. In addition, monitoring the digital instrument is done visually, which presents too much visual information at once, and can be confusing during a live performance. A real-time score, then, would significantly reduce the strain on performers, as well as it would keep technical requirements to the essential minimum. It would alleviate practical difficulties, while still supporting and developing the concepts explored in *(bcn)621*.

2.2. Key Concepts

Same as with *(bcn)621,* the broad concepts of *Arcos* are unity by bowing gestures, and structure through blocks of time. The current work also presents the central concept of creating a more natural, intuitive performance environment through a low-level symbolic representation of gestures. This idea, which came as a response to the pitfalls o f *(bcn)621*, is manifested in generating the real-time graphic notation directly from gesture data.

Basing the whole notation on gesture data brings the gestures even more into the forefront of the work, further emphasizing the idea of actions as a means of unifying the composition. As with any real-time score, sight-reading becomes the prominent way of performing. The gesture data driven notation presented here offers a more immediate, intuitive approach to sight-reading.

My basic assumption is that imitation is a rudimentary form of communication, which elicits instantaneous, instinctive response. Since the graphic notation is generated directly from gesture data, the performers can imitate what they see naturally, executing performance actions intuitively and idiomatically. They can perform in a way that is already ingrained in the performance practice of their instrument, without the intervention and interpretation of high-level symbolic notation. By contrast, the score of *(bcn)621* combines common-practice notation, action-based graphic symbols, and text instructions. While this notational approach is effective in explaining on paper which gesture to play and for how long, the process of learning the score—the symbols and their meaning—and interpreting it, is time consuming and requires intense conscious effort.

The real-time aspect of *Arcos* also contributes to this concept of intuitive performance by eliminating the need for counting temporal units, and even for thinking about time all together. It lets the performers focus exclusively on performing.

1 A standard click track was not a viable option, due to the need for indicating bar numbers.

Generating the graphic notation directly from gesture data brings the work closer to what Juraj Kojs sees as the "pure action-based scores", which provide immediate, easy access to music, and "utilize images that suggest clear instructions at first sight and [that] need no further explanation" [13, p. 67]. One reservation, however, regarding the configuration of my real-time score, is that the algorithm selects which gesture to display from a closed set of given actions. This type of real-time score is referred to as live permutated, or real-time permutative score [14, 15]. Since only the order is indeterminate, but the material itself is known, the performance does not entail pure sight-reading. In a sense, however, this resonates Winkler's notion of real-time notation that can still be learned in advance.

An isorhythmic structure is applied in *Arcos* as well, with large sections, divided into blocks of time, that repeat in augmentation or diminution. When the sections repeat, the change in overall duration alters the durations of the internal units proportionally. The ratios in this case derive from data analysis of gestures, but simplified. And, as I have mentioned above, the order of the actions is not fixed, unlike the sequences of *(bcn)621.*

Since actions are no longer organized in a set order, units of rest, or inaction, have a greater importance for the perception of the structure. As these units of inaction are the only fixed segments within each section, they become the key element for comprehending the structure, functioning as markers of form.

In addition, the visual information contributes to the sensation of time as an indicator of structure. Seeing how long a gesture appears on the screen, as opposed to only hearing a sound being played, may help with grasping the blocks of time of which the sections consist, and the temporal relationships between them. The combination of visual and auditory information possibly allows for perceiving more complex temporal relationships. The algorithm itself facilitates the use of unconventional ratios that are hard to implement on a paper score. On the other hand, to make these relationships perceivable, they cannot be too complex. Hence, I decided to simplify the ratios I derived from the gesture data.

2.3. Instrumentation

The piece is written for cello and augmented violin bow. Since the work focuses on bowing gestures, the cello part is made exclusively for the right hand, playing with a bow. The sole left hand instruction is to stop all the strings lightly—so they are neither fully pressed nor completely muffled—to produce sound which could be described as "hollow" or "airy". The cellist, however, could do that by placing a rubber band over the top part of the neck of the cello, thus, not using the left hand at all. The cello may be substituted by a double bass. In any case, the acoustic instrument should occupy the lower range of frequencies, in order to complement the augmented bow, which produces higher frequencies.

The augmented violin bow consists of a tangible part (a violin bow) and a digital part (a Max patch). As mentioned above, the purpose of the augmented bow is to afford a physically engaging electronic performance, focused on bowing gestures. I wanted to express the bowing idea not only in terms of movement, but also in terms of sound. Therefore, the performance is done with the bow, and its own sound is amplified and processed. The bow's motion and position data is mapped to the sound processing units (figure 1).

The construction of the bow is fairly simple: I mounted a contact microphone on the bow frog, and an LED diode on the bow stick. The contact microphone amplifies the bow's vibrations, i.e. the sound of the bow itself, and the LED is used for tracking the position of the bow, based on the Jitter jit.findbound method[2]. The Jitter algorithm follows the LED light as a point in space, and outputs two-dimensional coordinate positions of this point. Since the input is from a webcam, it is crucial to find the right position in the performance space, and to make sure no other light interferes with the LED light input. The amplified sound of the bow is processed in real-time, according to movement detection, brightness level, and position tracking data of the LED light. The audio signal processing includes a flanger effect and a spectral pitch shift effect. The augmented bow, then, is both the sound generator and the gestural sound controller, making it a standalone instrument with clear morphology.

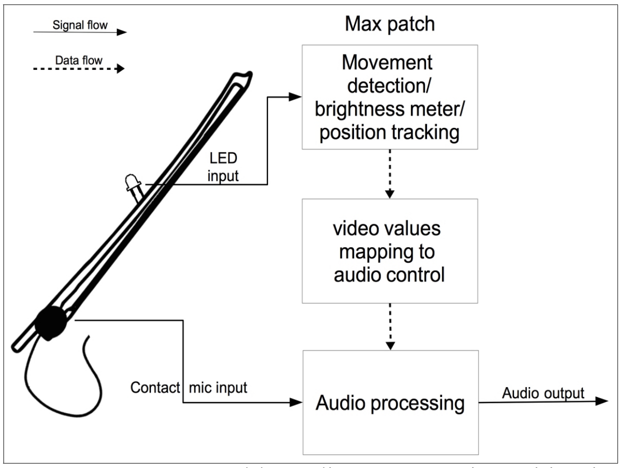

Figure 1. Augmented bow diagram. X axis position is mapped to pitch shift and flanger delay gain, Y axis position is mapped to flanger modulation rate, and distance from the center is mapped to flanger delay time. Movement activates the audio output, and brightness controls the volume.

2.4. Data Acquisition, Processing, and Mapping

Clearly, a vital step in creating a score based on gesture data is to obtain the data and make it usable. Collecting motion data from the two instruments was done by registering Myo armband[3] data of cello bowing gestures, and by capturing videos of the augmented bow position tracking, as it appears on the Max patch Jitter window.

Cellist Leo Morello lent a hand, quite literally, with the cello gesture data acquisition. In a long rehearsal session, Morello performed various bowing actions, while wearing a Myo armband on his right hand. The actions included normal bowing, scraping the bow along the strings, circular bowing, ricochet, arpeggiation, applying different amounts of bow pressure, and bowing on different parts of the instrument. The Myo output of each gesture was saved onto a separate text file. Each file, then, contains accelerometer, gyroscope, quaternion, and EMG data of specific gestures[4]. It is important to note that the Myo device was used only for recording data, and it is not meant to be used during the actual live performance.

In order to make the cello data available for visualization—recreating the gestures on the screen with a virtual bow—the next step was to calculate the tilt angle, angular velocity, and angle of rotation around the imaginary center from the accelerometer, gyroscope, and quaternion values[5]. A complementary filter was applied in order to integrate accelerometer and gyroscope values. With the final angles calculated, it was possible to map three-axis values to the position of the virtual bow, with X and Y mapped to the two-dimensional position of the bow on the screen, and Z to the size of the bow (giving the impression of moving towards and away from the viewer). To indicate pressure on the bow, EMG values were extracted from separate text files containing data of different degrees of pressure (heavy, medium, light, and gradual change). Figure 2 demonstrates the clear difference in EMG values between light pressure on the bow and heavy pressure on the bow.

The computer decides which gesture to show, and in the relevant cases, what amount of bow pressure to assign to it. Every gesture is represented with a number of files (between two to four), to account for variations within the action, such as intensity or placement of the bow. Each gesture, then, can be shown in a variety of ways and bow pressure levels. For the piece, I chose to use six gestures: normal bowing (side to side motion), circular bowing, scraping along the strings, arpeggiation/tremolo, tapping with the bow (hit once), and ricochet (hit and let the bow bounce).

The process of acquiring and using the augmented bow data was much simpler. Since a Jitter window shows the LED light mounted on the bow as a point in space, videos of each action were recorded directly from the Max patch (figure 3). Following each video, I programmed animations that illustrate the movements of this point in space, that is the motion of the bow. While the cello part of the score shows actions as they are, the augmented bow part of the score shows actions as they appear in the Max patch to the performer monitoring the bow's movements. When the computer decides which augmented bow gesture to show, the relevant animation is created in real-time. The animation of each gesture incor-

2 Jitter Tutorial 25: Tracking the Position of a Color in a Movie, https://docs.cycling74.com/max6/dynamic/c74_docs.html#jitterchapter25

3 A Myo armband is a wearable inertial measurement unit (IMU) device, built with eight medical-grade sensors. It provides motion, rotation, and orientation estimation of the device in space, via a three-axis gyroscope, a three-axis accelerometer, and a three-axis magnetometer, as well as muscle electrical activity via electromyography (EMG) [16].

4 Registering the Myo data was done in Max, using the Myo for Max externals by Jules Francoise: https://github.com/JulesFrancoise/myo-for-max

5 Quaternion values were transformed into Euler angles with the jit.quat2euler Jitter object.

porates random elements, such as direction or pace, for the purpose of variation and interest. Here, too, I chose six gestures, some are closely related to the cello ones: normal bowing, drumming with the bow, circular bowing, tapping on the bow, plucking its hair, and “bow crunch” (holding the bow and making crushing motions with the fingers).

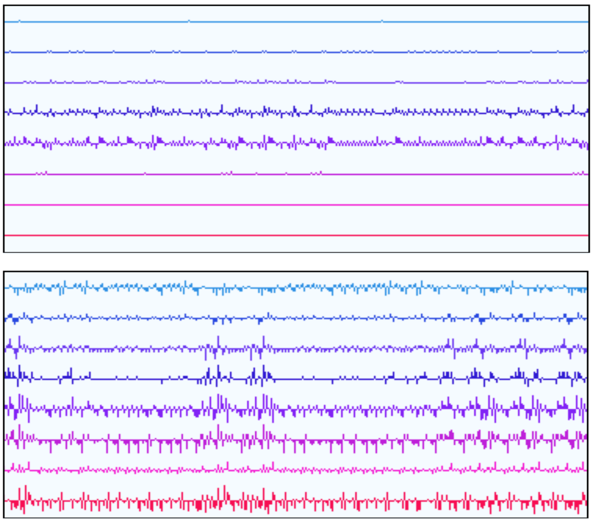

Figure 2. EMG values of bow pressure: light (top) and heavy (bottom).

Figure 3. Video recording of the Jitter window, capturing the motion of the LED diode mounted on the augmented bow.

2.5. Score Design

The score has two levels of function: control function, which is done in Max, and graphic display function, which is done in Processing. The Processing program receives instructions from Max (via Open Sound Control protocol[6]) of which gesture to display and when. While the “when to display gestures?” is governed by fixed timed sections, the “which gesture to display?” depends on the gesture that is currently being played by the augmented bow. Thus, the control function includes a gesture identification module[7]. Once a gesture is identified, a first order Markov chain determines which action to display at a given block of time. The Markov chain receives the current gesture played on the augmented bow, and outputs the next one to display for both instruments. Each identified gesture correlates to different probabilities of which would be the next gesture to display. These probabilities were determined according to natural playing positions and ease of transition between playing actions. For example, if the detected gesture is normal bowing, then there is a 5% chance for the same gesture to be selected next, 30% for drumming with the bow, 20% for circular bowing, 15% for “bow crunch”, 15% for tapping on the bow, and 15% for plucking its hair. If circular bowing is detected, the same gesture gets 5% to be displayed next, normal bowing gets 15%, drumming with the bow gets 30%, “bow crunch” gets 15%, tapping on the bow gets 10%, and plucking the hair gets 25%. Finally, the Markov chain's output is sent to Processing at given points in time, according to the fixed formal sections.

The code in Processing comprises different modules for each gesture. When it receives a gesture command from Max, the corresponding module is activated, and the relevant gesture is displayed. The graphic representation of the gestures is generated in real-time. In the case of the cello part, the gesture data from the corresponding text file is also being sent from Max in real-time. Figure 4 shows the overall process of the score's algorithm:

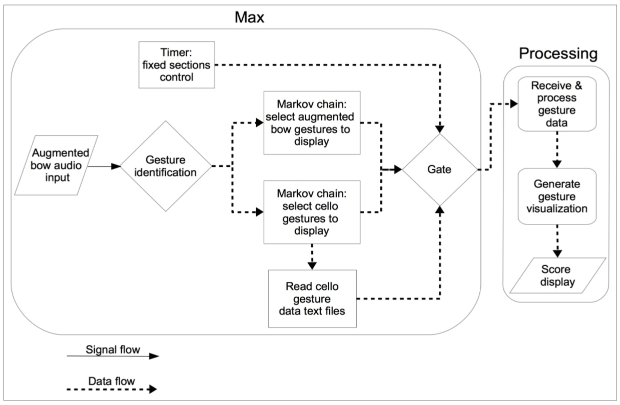

Figure 4. Overall score algorithm flowchart.

The screen itself is divided into two halves, with the cello part on the left side and the augmented bow part on the right side (figure 5). The performers are instructed to position themselves on stage accordingly. They also must sit in such a way that still allows the people in the audience to see the performance actions (i.e. not with the back to the audience).

The graphic notation, in the form of gestures that the performers imitate, looks differently for each instrument. The cello part shows the front of the instrument, including the lower part of the fingerboard and the upper part of the tailpiece. A virtual bow moves on the screen, driven by the gesture data, instructing the cellist what to play, or more correctly, how to play. Pressure on the bow is indicated by the color of the virtual bow: the darker the color, the heavier the pressure (figure 6). The augmented bow part shows a white dot, representing the LED mounted on the bow. The movements of the dot instruct the actions to the performer. Various amounts of delay were added to the white dot, depending on the gesture, for aesthetic pur-

6 The program uses oscP5 library for Processing by Andreas Schlegel: http://www.sojamo.de/libraries/oscp5/

7 While I tried visual gesture identification methods, such as with the MuBu Hierarchical Hidden Markov Models objects [17], the best results were achieved by using audio descriptors, such as a combination of loudness and spectral skewness [18].

poses (figure 7). I also decided to alter the point of view of the augmented bow's part circular bowing animation, showing the circle from above instead of from the performer's actual point of view, which is from the side (figure 8). The change was made in order to show the circular gesture more clearly, since the side point of view could be confusing. This clearer point of view also allows for greater variations in the motion, and it is more aesthetically pleasing than the sideways point of view.

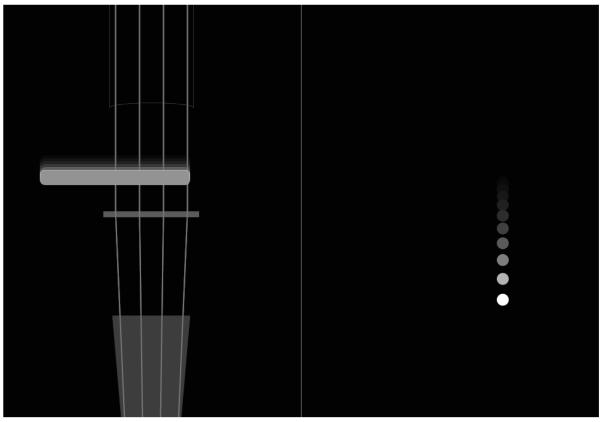

Figure 5. Screen layout: cello part on the left, augmented bow part on the right.

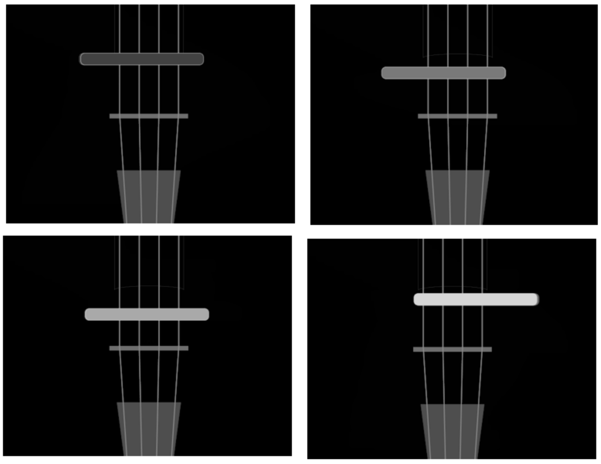

Figure 6. Cello bow pressure range from heavy (top left) to light (bottom right).

Figure 7. Augmented bow part: recreating the LED diode motion, as it appears in the Jitter window, with added delay.

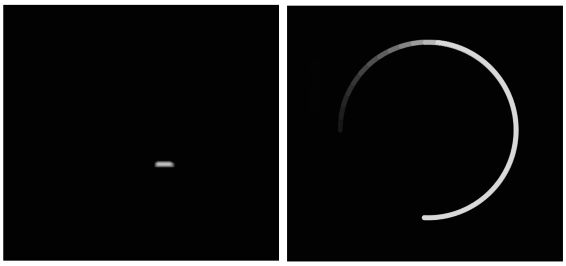

Figure 8. Augmented bow circular motion as captured by video, showing the sideways perspective (left), and as recreated in the score, showing the gesture from above (right).

3. CONCLUSIONS

At the time of writing this paper *Arcos* has yet to be performed. I am unable, then, to determine whether the gesture data approach I took for creating its real-time score was effective in a live performance situation. Only from my own perspective, as the augmented bow performer, I can conclude that interpreting graphic notation that is generated from gesture data is much clearer, and does feel more natural and fluent compared to interpreting text notation. While it is still necessary to learn this type of graphic notation designed for the augmented bow, and to practice interpreting it, I consider it an integral part of the performance practice of this instrument. And, although on a higher level of symbolic representation than the cello part, the bow's notation is still derived directly from the sound morphology of the instrument. It forms a tight relation, or even a sort of an ecosystem, between action, sound, and notation.

Furthermore, this gesture-based type of graphic notation discussed in this paper is especially useful for indicating complicated gestures and different variations within each gesture. However, creating such notation is a lot more energy and time consuming than providing textual instructions.

In addition, the setup of one computer that runs both a Max patch and a Processing program is CPU intensive, and may be too risky in a live performance setting. So far, however, it proved stable in testing. In my opinion, it is also preferable to networking devices, which could be even riskier in live performance.

While this paper concerns a specific composition, it illustrates a possible way to address real-time notation for NIME—using the interface's own data for notation—which other composers can utilize. Additionally, it could offer a direction for future work concerning interpretation by imitation. This attitude towards action notation applied in *Arcos* may prove relevant to real-time scores in general. Research into works that use similar notational procedures could shed more light on this subject as a tool for real-time notation, and maybe even contribute to our overall understanding of the process of sight-reading real-time scores. Such future research might also set the foundations for outlining a conceivable sub-category of real-time action-based notation.

Acknowledgments

Many thanks to David Dalmazzo and Rafael Ramirez for helping with acquiring and processing data, to Leo Morello for recording cello gestures, and to Nir Bitton for advising on the score design.

4. REFERENCES

[1] A . Clay, and J. Freeman, "Preface: Virtual Scores and Real-Time Playing," *Contemporary Music Review*, vol. 29, no. 1, pp 1, 2010.

[2] G. E. Winkler, "The real-time score: A missing link in computer-music performance," in *Proc. of the Sound and Music Computing Conf.*, Paris, 2004, pp. 9–14.

[3] S. Shafer, "Performer Action Modeling n Real-Time Notation," in *Proc. Int. Conf. on Technologies for Music Notation and Representation – TENOR'17*, Universidade da Coruña, 2017, pp. 117-123.

[4] S. Shafer, "Terraformation," in *Proc. Int. Conf. on Technologies for Music Notation and Representation – TENOR'17*, Universidade da Coruña, 2017, pp. 215-227.

[5] R. R. Smith, *Study no. 15b*, 2013. Accessed on: May 12, 2020. [Online]. Available: http://ryanrosssmith.com/study15b.html

[6] R. R. Smith, *Study no. 15b.1*, 2013. Accessed on: May 12, 2020. [Online]. Available: http://ryanrosssmith.com/study15b1.html

[7] R. R., Smith, "[Study no. 50][Notational Becoming] [Speculations]," in *Proc. Int. Conf. on Technologies for Music Notation and Representation – TENOR'16*, Cambridge, 2016, pp. 98-104.

[8] G. Paine, "Gesture and Morphology in Laptop Music *Performance*," in *The Oxford Handbook of Computer Music and Digital Sound Culture* (Ed: R.T. Dean). Oxford University Press, 2009.

[9] T. Wishart, *On Sonic Art*. Routledge, 1996.

[10] D. Smalley, "Spectromorphology: Explaining Sound-Shapes," *Organised Sound*, vol. 2, no. 2, pp. 107-26, 1997.

[11] K. Cascone, "Laptop Music-Counterfeiting Aura in the Age of Infinite Reproduction," *Parachute,* pp. 52-59, 2002.

[12] G. Paine, "Towards Unified Design Guidelines for New Interfaces for Musical Expression," *Organised Sound* , vol.14, no. 2, pp. 142-155, 2009.

[13] *J. Kojs, "Notating Action-Based Music," Leonardo Music Journal*, vol. 21, no. 1, pp. 65–72, 2011.

[14] S. Shafer, "Performance Practice of Real-Time Notation," in *Proc. Int. Conf. on Technologies for Music Notation and Representation – TENOR 2016*, Cambridge, 2016, pp. 65-70.

[15] L. Vickery, "The Evolution of Notational Innovations from the Mobile Score to the Screen Score," *Organised Sound*, vol. 17, no. 2, pp. 128–136, 2011.

[16] D. Dalmazzo, and R. Ramírez, "Bowing Gestures Classification in Violin Performance: A Machine Learning Approach," *Frontiers in Psychology,* vol. 10, no. 344, 2019, doi:10.3389/fpsyg.2019.00344

[17] J. Francoise, N. Schnell, R. Borghesi, and F. Bevilacqua, "Probabilistic Models for Designing Motion and Sound Relationships," *in Proc. Int. Conference on New Interfaces for Musical Expression*, London, 2014, pp. 287-292.

[18] M. Malt, E. Jourdan, "Real-Time Uses of Low Level Sound Descriptors as Event Detection Functions Using the Max/MSP Zsa.Descriptors Library," in *Proc. of the 12th Brazilian Symposium on Computer Music,* Recife, 2009, pp. 45-65.