COMMON GROUND, MUSIC AND MOVEMENT DIRECTED BY A RASPBERRY PI

Jonathan Bell, Aaron Wyatt

COMMON GROUND, MUSIC AND MOVEMENT DIRECTED BY A RASPBERRY PI

Jonathan Bell
Aix Marseille Univ, CNRS, PRISM
Perception, Representations, Image, Sound, Music
Marseille, France
*bell*jonathan50@gmail.com

Aaron Wyatt
Sir Zelman Cowen School of Music
Monash University,
Melbourne, Australia.
Aaron.Wyatt@monash.edu

ABSTRACT

This paper describes *Common Ground*, a piece for six dancing singers and electronics, in which the coordination between performers is ensured by a RaspberryPi-embedded node.js web application. The singers received synchronised scores in the browser of their phone, which they wore in head-mounted display in order to free their hands and enhance their scenic presence. After a description of the artistic project, the elaboration of the score is examined under the categories of movement notation (how trajectories are embedded in musical notation), spectral composition (microtonal tuning between synthesised sounds and human voices), algorithmic processes (how the recent *bell* coding language facilitates processes for which Max patching is ill-suited). The article finally describes the Raspberry implementation, outlining potential ameliorations of the current system, including dns support and unnecessary dependance on a dedicated router.

1. INTRODUCTION

Common Ground, by the Spanish artist Keke Vilabelda, is an immersive installation with large paintings, videos, and three tons of salt covering the ground. Commissioned by the Grau Projekt art gallery in Melbourne [1] , it reflects on the common features of landscapes (salt lakes) situated at antipodes of one another (Spain Australia). The initial idea of the musical piece of the same name was to take advantage of this beautiful immersive space, and use it as set design for the performance of six female voices accompanied by electronics [2] (see Fig. 1). The poems chosen for the piece, by the English poet Robert Bell, take the sea as source inspiration - the horizon, natural elements, treated as points of departure for meditation upon everyday life.

From the beginning, the visual aspect of the piece revealed itself to be of primary importance, which is why the performative part had to integrate movements/dance, costumes, and find a way for the score to be part of this ecosystem without disturbing it. Indeed after more than ten pieces written for the Smartvox system, with choirs and ensembles of various sizes, the main issue in concert/performance situation concerns more the theatrical restitution of the piece than the music itself. The system allows singers to move freely on stage and around the audience whilst singing with confidence, which gives the work an interesting immersive feeling. However, the way the system has so far been visually presented need to improved. Although the fact that singers wear headphones while singing should arguably be the most questionable source of interference between the singer and his audience, it is in fact their visual presence which is the most problematic when the singer has to break eye contact with his/her audience in order to watch the score [3] . Moreover, the presence of the smartphone itself as an object part of the performance seemed most problematic, which encouraged for the search of different solutions.

Figure 1. The *Common Ground* installation by Keke Vilabelda at *Grau Projekt*, Melbourne.

2. HMD

Following *Mit allen augen*, a piece in which singers and instrumentalists wore head-mounted displays and walked freely around the audience, *Common Ground* carries on with similar concerns, trying to take this idea further by adding a precisely determined choreography. Placed above the head in HMD, smartphones are still rather cumbersome from a theatrical perspective, but SmartVox will probably take advantage in a few years of lighter solutions, such as Vufine glasses (see Fig. 2, left) which proved to be the

[1] https://www.kekevilabelda.com/common-ground

[2] A captation of the performance is available here : https://youtu.be/ZrLgbBw4xfU

[3] See for instance *SmartVox in India:* https://youtu.be/7_FMqLg9vHM

Figure 2. Experimentation with various lowcost Head-Mounted Display (HMD) solutions.

most discreet, allowing for a mirror display of the score (i.e. of the smartphone's screen) in the corner of one of the lenses of the performer's glasses. Although one-eyed, the display is comfortable and wide enough, unfortunately its hdmi connection too often interfered with the audio output of the phone, making it unreliable in a concert/performance situation. Furthermore, its relatively high cost made it inappropriate for large score distribution [4]. Solutions such as QLPP 90 ° FOV AR headset (see Fig. 2, center), evocative of Microsoft Hololens imitations, showed interesting results as they allow for holographic display of the score, but the curvature of their glass requires calibration depending on the performer's phone size and their pupillary distance, again inconvenient for efficiency purposes, because of the often limited time for rehearsals in the performance space.

The solution therefore adopted for *Common Ground* was simply a headset constituted of a double mirror (see Fig. 2, right) for a large and comfortable display slightly above the head of the performer, leaving free the lower field of view, which was appreciated as the performers need to move - sometimes rapidly - in the performance space.

Cat Hope [1] and Christian Klickenberg [2] have both used animated notation in several of their operas, giving evidence that these new forms of notation re-shape the roles traditionally assigned to conductor, singers and instrumentalists : the beating of the time by the conductor is not the main point of reference for singers, instruments, lighting etc… The scrolling of time with a cursor (as, for instance, in the Decibel ScorePlayer) allows for many processes to be automated, giving the music director a different role. Graphic notation also allows for more freedom of interpretation from the perspective of the performer.

However, those solutions still imply the presence of many screens on stage. An interesting field of research in the domain of AR scores [3][4] consists in using Hololens for display of 3D holographic structures (see Fig. 3), an area which soon hopefully interest a larger community of composers and performers.

3. SCORE ELABORATION

3.1 *bach* INScore

The notation in *Common Ground* is conceived as a mixture of singing and movement information. The movement is represented as a graph representing singers are the stage

[4] Performances using this system often require large ensemble, up to 80 performers in *Le temps des nuages* nuages, 12 instruments and 12 voices in *Mit Allen Augen*, or 30 singers in SmartVox.

Figure 3. The Elision ensemble performing David Kim-Boyle's new work at TENOR 2019. Photographed by Cat Hope.

below the musical stave, and was controlled by spatial information stored in *bach*, so that the choreography could be written in the musical score directly. The movements are then sent from *bach* to INScore via OSC. INScore [5] is an environment for the design of augmented interactive music scores, opened to unconventional uses of music notation and representation, including realtime symbolic notation capabilities. It can be controlled in real-time using Open Sound Control [OSC] messages as well as using an OSC based scripting language, that allows designing scores in a modular and incremental way. INScore supports extended music scores, combining symbolic notation with arbitrary graphic objects. All the elements of a score (including purely graphical elements) have a temporal dimension (date, duration and tempo) and can be manipulated both in the graphic and time space. They can be synchronized in a master/slave relationship i.e. any object can be placed in the time space of another object, which may be viewed as "time synchronisation in the graphic space". As a result, a large number of operations can be performed in the time domain and in particular, moving a cursor on a score is simply achieved using the synchronization mechanism and by moving this cursor in the time space. Time in INScore is both event driven and continuous [6] , which makes it possible to design interactive and dynamic scores. The system is widely open to network uses [5]: it allows to use both local and remote resources (via HTTP), it provides a forwarding mechanism that allows scores to be distributed in real time over a local network. INScore has built-in solutions for monitoring the position and the speed of cursors in an efficient way.

bach supports by default spatial information in its 9th 'spat' slot. Inspired by Ircam Spat, this slot allows to store position information (x, y), as well as azimut indicating in which direction the source is facing [5]. Designing a movement of one of the singers therefore consisted in sending coordinate changes or interpolations (x, y and azimut) with a given duration (the length of the note, movements be-

[5] See https://www.youtube.com/watch?v=sTgTv9yqZhI for demonstration.

ing marked in blue, with empty noteheads) [6] . A compositional constraint consisted in avoiding overlapping between dance and singing : the singer never had to move and dance at the same time, rather he/she alternates between singing and moving.

3.2 Spectral Composition - Synthesis

The first pieces composed with SmartVox used mixtures of recorded and synthesised sounds [7] , but most recent works sound synthesis almost exclusively, because it allows for more precise control over sound as well as harmony. The first piece of the cycle [8] mainly consisted of an exploration of basic wave shaping techniques [9] (hence the overall metallic sonority of the piece). *Shir Hassirim* [10] , a piece written around the same time, was more focussed on FM synthesis [11] : through spectral analysis, the goal was to replicate in the choir phenomena like modulation increase. [12]

In the piece *Mit Allen Augen* [13] , all the electronics were generated from the PRISM laboratory synthesizer [7], whose perceptive model aims for the emulation of sometimes inconceivable sounds, such as a liquid rain of metal. [14] This material was analysed in *bach* to extract pitch material [15] then ready to use for orchestration (for 12 voices and 12 instruments).

In *Common Ground*, sound synthesis primarily came from the Synthesis Tool Kit [8], accessed through the PerColate library in Max. The physical modelling objects, and brass in particular was of interest as it really captured complex timbral features of brass instruments elsewhere often discussed by Jean-claude Risset and David Wessel [9]. The second source of sound synthesis in *Common Ground* can be described as a subtractive model, in which dense FM spectra are filtered by FFT filters, so as to obtain clearly identifiable pitch content. [16]

The precise overlapping of voices and electronics is made possible in performance thanks to the recent improvement of smartphones' capacities. On the composer's side, a precise dialogue between electronics and the score is highly facilitated by *Max for Live* [17] .

3.3 Audioscore - Display

Audio scores have been a key concept for the research undertaken by one of the authors [10]. Since a survey by Bhagwati on this topic [11], audio scores seem to enjoy increasing popularity (Stanford audioscore). Our claim is that, when applied to vocal ensemble writing in particular, they allow unprecedented accuracy in the realm of spectral

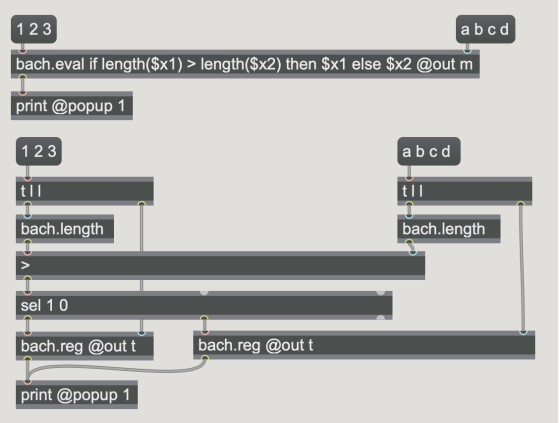

Figure 4. A comparison between an llll manipulation process described through a snippet of bell code (in the bach.eval object box, above) and the corresponding implementation within the standard graphical dataflow paradigm of Max (below): The first version is evidently easier to read.

composition, when the singers need to match the harmonic content of the tape accurately.

Working with various ensembles and receiving each time important feedback has confirmed that visual and auditory information need to be anticipated as much as possible : pauses in a singer's separate part always provides aurally what comes next. Also visually, *Common Ground* adopted a 2-systems-per-page mechanism in which the system that is not playing always anticipates what comes next. [18]

3.4 Aspects of the bell language

3.4.1 Introduction: Vocaloid

Common Ground also marks an evolution in the way vocal generation is used in SmartVox, which hitherto consisted in storing path to samples of vocal speech inside each note or each melisma through the slot storing system in *bach*. [19] This method however presented significant drawbacks : although it made the end result slightly more expressive or convincing, it was extremely time consuming to make phoneme change correspond to note change whilst designing each vocal line. Also reading samples direct to disk often introduced delay which made the result imprecise.

The new method consists in using a vocal synthesizer (AlterEgo Plogue, a free equivalent of the japanese Vocaloid synthesizer[12][13]), using *bach*'s slot storing system to control the synthetiser via midi. This slightly more robot-sounding solution has the great advantage to be far more malleable algorithmically, since a new note can trigger a phoneme change. With the discovery of the new *bell* language [14] in *bach* [15], the method opened the door to promising experiments in the realm of algorithmic composition.

[6] See the part of Soprano 1: https://youtu.be/FcF4oNxJweg
[7] See *SmartVox* for instance: https://youtu.be/JZsJn7EEW-A
[8] See *In Memoriam J.C. Risset*: https://youtu.be/hQtyu1dcCaI
[9] See for demonstration: https://youtu.be/v-LlClEnxf0
[10] See *Shir Hassirim*: https://youtu.be/7GpArQa6mQ4
[11] See for demonstration: https://youtu.be/D6mCgx4pSxs
[12] As in the following extract: https://youtu.be/7GpArQa6mQ4?t=26
[13] See *Mit Allen Augen*: https://youtu.be/ET_OBgFWx04
[14] See for demonstration: https://youtu.be/2kdIaqAhUGs
[15] See for demonstration: https://youtu.be/gZKONcOOhaE
[16] See for demonstration: https://youtu.be/zN95OkWSDHY
[17] Bach's playback notification can be redirected to ableton via M4L for precise synchronisation: see https://youtu.be/VJvY5wYl_cM
[18] See for instance the alto part: https://youtu.be/yWD6u2cPvSc. Slot No4 (the yellow one **here**) is the one that stores the sample's path.
[19] See for demonstration: https://youtu.be/s4qS2khwkT0

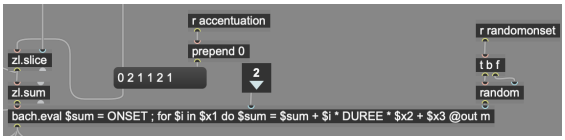

Figure 5. A loop calculates the onset of each syllable of a vocal line according to a starting onset (the variable "ONSET"), a given duration/tempo ("DUREE"), and prosodic accentuation (2 1 1 2 for long short short long).

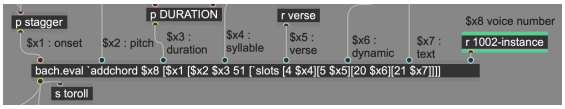

Figure 6. The *bell* language is mainly exposed in Max through the *bach.eval* object. $x1, $x2... correspond to the different inlets of the object. *bach.eval* makes the construction of lisp-inherited parenthesis structures much easier than with the data-flow *bach.wrap* system.

3.4.2 Bach Evaluation Language for llll - an overview of the bell language

The *bell* language in *bach* arose from an observation that the Max patching environment can be cumbersome when formalising algorithmic compositional processes: "It has been clear since the beginning of *bach* that non trivial tasks - in algorithmic composition - require the implementation of potentially complex algorithms and processes, something that the graphical, data-flow programming paradigm of Max [...], is notoriously not well-suited to."[14]. Single line snippets code in bell often require many objects and cables in Max (see Fig. 4).

Indeed, while the Max GUI can be extremely intuitive and efficient for many DSP processes, its data-flow paradigm can make message formatting efficient in Max (and hence in *bach*). As exemplified in Fig. 5, bach.eval allows for a single line of code to centralize message formatting, which would have formerly required dozens of objects, themselves most often bringing order or priority issues.

The implementation of variables in the *bell* language constitutes another major improvement of *bach*. The ability to name variables in Max (such as ONSET, or DUREE, as in the loop expressed in Fig. 5) and assign them a value helps again centralising information within simple equations, which the message-driven send-receive Max functionality would have made more prompt to error.

3.4.3 Algorythmic composition with bell

Although in germ in *Common Ground*, a more systematic approach to algorithmic polyphony generation was used in a *Deliciae* [20] , a piece composed just after *Common Ground*, while discovering the new *bell* language (see Fig. 6) [14] [16].

The polyphony of European tradition obeyed extremely strict rules throughout Europe during the Renaissance. Many of those rules discussed in the treatises of the time served

[20] A video of the performance is available at: https://youtu.be/zxnznD0Gzo0

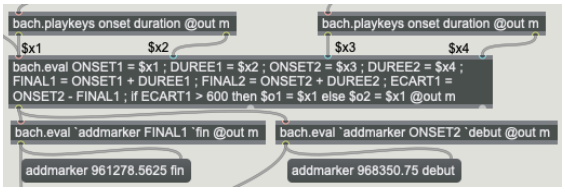

Figure 7. The following script adds markers only when two notes are separated by more than 600ms.

as source of inspiration for polyphony generation with the tools exposed above. The first obvious parallel consists in treating each voice as equal, unlike for instance when writing for an instrumental ensemble of a modern orchestra. This is why most polyphonic passages in *Common Ground* and *Deliciae* were generated inside a poly~ in Max, with each instance (i.e. each voice) receiving the same information regarding text, prosody, and harmonic material, but only differing by vocal range (sopranos for instance cannot sing below middle C and so forth).

Contrast in Renaissance polyphony often consist in alternation between homophonic passages and contrapuntal ones, which inspired most parameters available to tweak for a given verse: when the variables RANDUR, RANDOMONSET, DECAL, and STAGGER are set to 0, the algorithm will generate a homophony [21] (singers articulate and move from one pitch to the next at the same time). If only RANDUR increases, voices will start at the same time, but their duration will differ between each other. If only RANDOMONSET increases, they will all have the same duration but start at different times. If only DECAL increase, voice will enter at regular intervals from the bottom-up (and inversely if DECAL is negative). STAGGER, finally, imitates a behaviour typical of the renaissance where two groups of voices are staggered or delayed by a given value.

3.4.4 Automatic cueing system

Since the beginning of SmartVox (see [17], Fig. 4), cueing the singers with what comes next appeared one of the main advantages of the system.

To identify appropriate moments for page turns and cue-

[21] See parameters tweaks on the right hand side for demonstration here: https://youtu.be/OKkiySEagm0. ONSET is in milliseconds and correspond to the position in the timeline where the generation is happening : as exemplified in the video 747618 ms correspond to 12'28". The term DUREE (French for duration) represents the duration of notes : the tempo speeds up when durations diminished

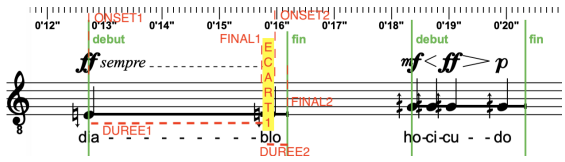

Figure 8. The first note (with lyrics "dia") has a duration that lasts until the beginning of the following note, (with lyrics "blo"). The distance between the two (ECART1, highlighted in yellow) is almost null.

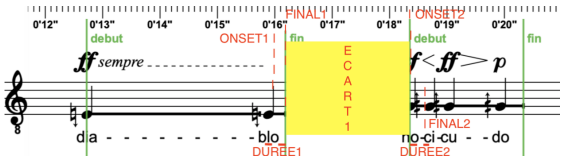

Figure 9. The two notes (with lyrics "blo" and "ho" respectively) are separated by a silence longer than 600 ms (ECART1 lasts a bit more than two seconds), therefore two markers are generated.

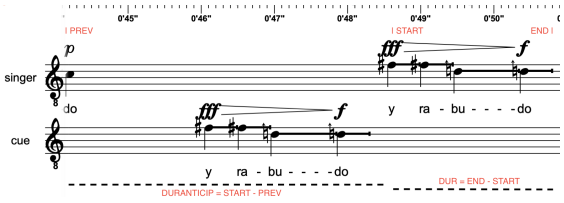

Figure 10. When the pause is long (or very long....) the cue needs to be provided as late as possible i.e. just before the singer's entrance. The corresponding onset value is 0'46" because START*2 - END = 48,5*2 - 51 = 46

ing the singers accordingly, the first step consisted in identifying the start and end of each phrase (see Fig. 7): with iterations on each note of the score two by two, we evaluate if the distance between two notes is superior to 600 ms: in the first case it isn't (see Fig. 8, the two notes are close to one another) and nothing happens. On the following iteration however, the gap between two notes is wider than 600ms (see Fig. 9), so the messages "addmarker fin" and "addmarker debut" are sent to the end of the phrase and to the beginning of the next phrase respectively.

When a performer has nothing to sing, this precious time is systematically used in the score to provide cues feeding the perfomer's headphone with what is coming next: using the markers previously generated to retrieve their onsets, if the pause is longer than the phrase to sing, (i.e. if the DURANTICIP is greater than DUR (see Fig. 10, and the "then" stance of the "if" statement in the code below), then the cue will need to start at the onset corresponding to the difference between entrance of the singer (START) and the end of his phrase (END), with a 300ms break between the two. If on the other hand, the pause is shorter than the phrase to sing (see Fig. 11, and the "else" stance of the if statement below), then the cue needs to start as soon as possible, i.e. as soon as the singers has finished to previous phrase (PREV):

```
START = $x2:($x1 1) ;
END = $x2:($x1+1 1) ;
PREV = $x2:($x1-1 1);
DUR = END - START ;
DURANTICIP = START - PREV ;
if DURANTICIP > DUR then ';
'tocue 'paste 2* START - (END + 300) 2
else '; 'tocue 'paste (PREV + 60) 2
```

Finally, onset information from the 'end' markers (the ones named 'fin', as in Fig. 8 at 0'16"200"') are used for

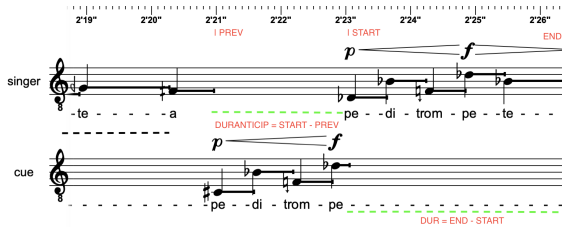

Figure 11. When the pause is short, the cue needs to be provided as soon as possible i.e. just after the previous singer's phrase (see the PREV variable).

display information : the domain to be displayed on the playing staff and on the preview staff (i.e. the staff-line that is coming next, as for page turns) of the *bach.roll*.

```
'addmarker $x2:$x1
['play [$x2:$x1 ($x2:($x1+1)+ 200)]
'preview [$x2:($x1+1) ($x2:($x1+2)+ 200)]]
```

Each time the cursor hits one of these markers, the domain display of both 'playing' and 'preview' staves are updated, provoking at the same time an alternation up and down between the position of those staves, so that the passive (or 'preview') roll looks like an anticipation of the active (or 'playing') one, resulting on a 2-staves display with constant preview. [22]

4. A RASPBERRY PI HARDWARE EMBEDDED SYSTEM SOLUTION FOR LOCAL NMPS

In search of a light plug-and-play dedicated system to be sent over the post, the Raspberry Pi quickly appeared as the best option to host SmartVox on an embedded system. Node.js runs on Raspbian, and SmartVox proved to be very stable on a Raspberry Pi 3, so, once installed, the only two steps for a *0-conf* deliverable hardware were:

- Setting up a static address for a dedicated router (e.g. tp-link...).
- Starting SmartVox at boot using linux 'systemd' service.

Starting a script at boot can be done on Raspbian with a file containing the following in the etc/systemd/system:

```
[Unit]
Description=My service
[Service]
ExecStart=/home/pi/Desktop/hello.sh
[Install]
WantedBy=multi-user.target
```

With the hello.sh script containing the following to launch the server:

```
#!/bin/bash
cd /home/pi/Desktop/risset
npm run start
exec bash
```

[22] See for instance the tenor part: https://youtu.be/NLpI_OpFcTs

This low-cost system now allows the sending of ready-to-use scores. Once the system is power-supplied, all the performers need to do is to join the dedicated Wi-Fi, and type the static IP address of the server on their smartphone/tablet (i.e. for the performers: 192.168.0.100:8000, and for the conductor: 192.168.0.100:8000/conductor). In January 2019, the system was rented to the Caen French conservatoire via BabelScores, [23] thus proposing a rental of performing scores (separate parts) of a new kind.

To make configuration even easier in the future, a lightweight DNS server, like dnsmasq, could be installed and configured on the Raspberry Pi to allow performers to enter a friendlier, more human readable address to access the nodejs server. Additionally, new ways of updating the scores on the device could be explored to both simplify the process and to limit the amount of data that needs to be sent out to it. Currently, the most straight-forward way to update the device is to send out a new disk image to be written to the SD card at the other end. Piping the image through xz substantially reduces the image file size, particularly if the filesystem on the card contains a large amount of free space. This can be done using the following terminal commands to first create the image on one end and to then write it on the other:

```
sudo dd if=/dev/disk3 bs=4m | xz > common2.iso.xz
xzcat common2.iso.xz | sudo dd of=/dev/disk3 bs=4m
```

(where /dev/disk3 is replaced by the SD card device name).

To make this process easier for the end user, it could also be possible to have a web server on the device configured to accept the upload of update packages. These would then only need to contain newer versions of resource files (like the videos used for the scores), so that the entire system doesn't need to be refreshed for minor changes.

5. POSSIBLE AVENUES FOR A FUTURE EXTENSION OF THE WORK

The encounter with the *bell* language offered the author [24] new perspectives in the realm of algorithmic composition. The linkage with the *ConTimbre* library in particular offers promising results in the instrumental domain, with at the same time an intuitive control over the generated result as well as a possibly more speculative approach related to machine learning [25] . A future extension of *Common Ground* with therefore include instruments as well as voices.

6. CONCLUSIONS

This article presents an overview of the evolution of the SmartVox project, with an emphasis on the artistic project *Common Ground*, its more systematic use of algorithmic processes for composition thanks to the *bell* language in *bach*, as well as the Raspberry Pi implementation of the piece/server.

Acknowledgments

We would like to thank Cat Hope for having made this work possible in Melbourne.

7. REFERENCES

[1] C. Hope, A. Wyatt, and D. Thorpe, "Scoring an animated notation opera – the decibel score player and the role of the digital copyist in 'speechless'," in *Proceedings of the International Conference on Technologies for Music Notation and Representation – TENOR'18*, S. Bhagwati and J. Bresson, Eds. Montreal, Canada: Concordia University, 2018, pp. 193–200.

[2] C. Klinkenberg, "A combination of graphic notation and microtonal pitch notation in the video score of the opera "the cross of the engaged"," in *Proceedings of the International Conference on Technologies for Music Notation and Representation – TENOR'18*, S. Bhagwati and J. Bresson, Eds. Montreal, Canada: Concordia University, 2018, pp. 186–192.

[3] D. Kim-Boyle, "3d notations and the immersive score," *Leonardo Music Journal*, vol. 29, pp. 39–41, 2019.

[4] D. Kim-Boyle and B. Carey, "Immersive scores on the hololens," in *Proceedings of the International Conference on Technologies for Music Notation and Representation – TENOR'19*, C. Hope, L. Vickery, and N. Grant, Eds. Melbourne, Australia: Monash University, 2019, pp. 1–6.

[5] D. Fober, Y. Orlarey, and S. Letz, "Towards dynamic and animated music notation using inscore," in *Proceedings of the Linux Audio Conference — LAC 2017*, V. Ciciliato, Y. Orlarey, and L. Pottier, Eds. Saint Etienne: CIEREC, 2017, pp. 43–51. [Online]. Available: inscore-lac2017-final.pdf

[6] ——, "Inscore time model," in *Proceedings of the International Computer Music Conference*, 2017, pp. 64–68. [Online]. Available: inscore-icmc17-final.pdf

[7] M. Aramaki, C. Gondre, R. Kronland-Martinet, T. Voinier, and S. Ystad, "Imagine the sounds: anintuitive control of an impact sound synthesizer," in *Auditory display*, ser. Lecture Notes in Computer Sciences, Ystad, Aramaki, Kronland-Martinet, and Jensen, Eds. Springer Berlin / Heidelberg, 2010, pp. 408–422. [Online]. Available: https://hal.archives-ouvertes.fr/hal-00462274

[8] P. Cook and G. Scavone, "The synthesis toolkit (stk)," 08 2000.

[9] J.-C. Risset and D. L. Wessel, "5 - exploration of timbre by analysis and synthesis," in *The Psychology of Music (Second Edition)*, second edition ed., ser. Cognition and Perception, D. Deutsch, Ed. San

[23] Babelscores (https://www.babelscores.com/) currently supports actively supporting the SmartVox project: http://1uh2.mj.am/nl2/1uh2/lgi4u.html. The first piece performed in Caen with the Babelbox is available at the following address : https://youtu.be/wUyw0KQa5Wo

[24] To clarify, Bell is author of the article, albeit mere user of the coding language of the same name.

[25] See the following for demonstration : https://youtu.be/ByeIyRLnX-w

Diego: Academic Press, 1999, pp. 113 – 169. [Online]. Available: http://www.sciencedirect.com/science/article/pii/B9780122135644500068

[10] J. Bell, "Audio-scores, a resource for composition and computer-aided performance," Ph.D. dissertation, Guildhall School of Music and Drama, 2016. [Online]. Available: http://openaccess.city.ac.uk/17285/

[11] S. Bhagwati, "Elaborate audio scores: Concepts, affordances and tools," in *Proceedings of the International Conference on Technologies for Music Notation and Representation – TENOR'18*, S. Bhagwati and J. Bresson, Eds. Montreal, Canada: Concordia University, 2018, pp. 24–32.

[12] H. Kenmochi, "Vocaloid and hatsune miku phenomenon in japan," in *InterSinging*, 2010.

[13] K. Y. Lam, "The hatsune miku phenomenon: More than a virtual j-pop diva," *The Journal of Popular Culture*, vol. 49, no. 5, pp. 1107–1124, 2016. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/jpcu.12455

[14] J.-L. Giavitto and A. Agostini, "Bell, a textual language for the bach library," in *ICMC 2019 - International Computer Music Conference*, New York, United States, Jun. 2019. [Online]. Available: https://hal.archives-ouvertes.fr/hal-02348176

[15] A. Agostini and D. Ghisi, "Bach: an environment for computer-aided composition in max," in *Proceedings of the 38th International Computer Music Conference (ICMC)*, Ljubljana, Slovenia, 2012.

[16] A. Agostini, D. Ghisi, and J.-L. Giavitto, "Programming in style with bach," in *14th International Symposium on Computer Music Multidisciplinary Research*. Marseille, France: Mitsuko Aramaki, Richard Kronland-Martinet , Sølvi Ystad, Oct. 2019.

[17] J. Bell and B. Matuszewski, "SMARTVOX - A Web-Based Distributed Media Player as Notation Tool For Choral Practices," in *TENOR 2017*, Coruña, Spain, May 2017. [Online]. Available: https://hal.archives-ouvertes.fr/hal-01660184