AUDIOVISUAL SCORES AND PARTS SYNCHRONIZED OVER THE WEB

Jonathan Bell
Aix Marseille Univ, CNRS, PRISM
Perception, Representations, Image, Sound, Music
Marseille, France.
jonathan.BELL@univ-amu.fr

ABSTRACT

Typing **smartvox.eu** into the address bar of the browser of several phones, tablets and/or computers simultaneosly is a simple way to understand what this web application is about: the different parts of the same musical score are being synchronized over the internet, through a remote server. This form of music making falls under the category of *networked music performance*, and addresses questions regarding what becomes a live performance of chamber music when musicians are distanced from each other, when sheet music is replaced by screens and headphones, or when the form of the piece is generated live by algorithms. The scores, composed in the *Bach* environment, display a scrolling playhead in proportional notation, bypassing conventional bars and beat rhythmic notation. Providing the performer with audio-scores and animated musical representation undoubtedly simplifies and speeds up the rehearsal process, but large forces (such as 80 simultaneous connections) and the constraints of a concert situation still leave numerous technical problems unsolved (mainly concerned with synchronization between devices, and server saturation), to which the present paper will attempt to partially formulate a response.

1. INTRODUCTION

This article presents the continuation of the research published in *SmartVox, a web-based distributed media player as notation tool for choral practices* [1]. SmartVox is developed within the *Soundworks* framework [2]. It is a single page web application dedicated to the delivery and synchronization of polyphonic multimedia (audio + visual) scores, themselves composed in the *Bach* environment [3] for Max/MSP. In a performance involving, for instance, thirty singers, each performer hears and sees his/her own part displayed in the browser of his/her smartphone, and the whole is synchronized through the distributed state of the web application. The application has enjoyed increasing success since its conception in 2015, which leads to two questions:

- For the performer, how does audiovisual animated notation differ from paper music?
- What is transformed from a chamber music perspective, and from an audience point of view?

Following animated notation with a cursor/timeline moving from left to right is often experienced, from a performer's perspective, as far simpler than deciphering a rhythmically complex score and following a human conductor. Dynamic notation [4] and audio-scores [5] can also facilitate the learning process of unfamiliar idioms, such as microtonality. However, the use of screens can be a potential drawback, as it prevents performers from making notes on their scores, such as breath indications, fingerings, etc... If James Bean's DENM environment attempts to solve this issue with an elaborate GUI [6], some composers judge this (extreme) sight-reading situation to be something worthwhile exploring [7][8][9]. This "slow but steady shift away from textualization in digital media" [10] only concerns the way the information is presented to the performer—not its content—, and yet the score *medium* is of such importance for the composer that it presents a genuine change of paradigm in his/her craft. With audiovisual information distributed to the players through smartphones during a performance, the interaction between audio-scores (musical material sent through earpieces to performers) and visual input (musical notation) changes the traditional relationship between composer, conductor, performer, and listener.

Networked music performances are usually thought of as way to play together from distant locations: "Networked music performance is often intended as a practice of remote performance, with musicians displaced far away at distant locations" [11], "A Network Musical Performance (NMP) occurs when a group of musicians, located at different physical locations, interact over a network to perform as they would if located in the same room" [12], or even across the globe: "Sonic and spatial aspects of networked music performance (NMP) or networked multimedia performance [...] will be explored from a particular perspective—that of remoteness or spatial distance. In NMP the performers (agents) are typically separated by distances, which in extreme cases add up to tens of thousands of kilometers." [13] With the growth of the internet at the turn of the century, the SoundWire group at Stanford CCRMA and other labs produced important research works and concerts which explored this global aspect of performance. Georg Hajdu had already achieved such a performance in 2000: "In the performance of my piece MindTrip [...], the five performers were located in different cities across the globe.

They were linked by the Quintet.net server running locally in Münster, connected via a 128 kbit dial-in ISDN connection to the internet" [10]. Networked performances, therefore, refer in no small part to the possibility to play together from distant locations. Some recent developments of NPM and distributed systems, however, show great interest in connecting terminals located in the same room: "In this book, the use of wireless technologies enables, instead, to connect musicians located in the same room, in a large indoor space or in outdoor spaces. We may, thus, distinguish between indoor local, outdoor local or remote NMP" [11]. In the work presented below, performances took place in the same room, but with performers distanced from each other – around the audience – for an immersive listening experience. From a performer's point of view, mutual listening is modified, or sometimes even impeded, in these kinds of concerts, and the codes of chamber music have to be conceived of in a different way, hence the reliance on a local area network.

2. MAJOR UPDATES

2.1 Go web

When transferring a file from one machine to another which is located in the same room, web technologies often reveal the paradox that it is often easier to use the internet (e.g. WeTransfer), rather than finding a local solution. The latest technical improvement of SmartVox therefore consisted of hosting the application on the web (in July 2017). Since its conception in March 2015, the SmartVox app required the physical presence of a server in the room where the performance took place. The server consequently required node.js (server-side javascript), and the application [1] . This practical limitation prompted remote web hosting, which made it possible for performers to rehearse the piece together without the physical presence of the server on a computer. Since then, performers can rehearse physically in the same room – but also remotely –, and all that is required from them is a tablet (or phone) with access to the internet, in order to type into their browser the domain name of the piece to be played (e.g. nuages.smartvox.eu). This feature considerably simplifies the task of the performer, and lets us foresee great potential in terms of dissemination of the application. Whilst increasingly used in rehearsals, local servers are still obviously considered to be the best practice in a concert situation. With a reliable internet connection and relatively recent devices, the synchronization between different devices seems almost equivalent to those realized with a local server (see Section 4, Table 2 and 3). The measurements of Section 4 will therefore seek to determine how several factors (the distance of the server, its architecture and configuration, the quality of the network) impact on the loading and synchronization of the different files (the scores), which are prerequisites to a successful musical performance.

[1] The SmartVox web application, open source, is available here: https://github.com/belljonathan50/SmartVox0.1

2.2 Algorithmic composition/open form

The second major improvement of the application consisted of generating the form of the piece in real time [2] , but with fixed events on a local level. The code below shows a basic algorithm permanently used by the server accessible through **smartvox.eu**, choosing randomly which section will come next.

```
let openform = function () {
let timesArray = [0, 84, 123, 173, 262, 292, 362, 403, 517, 535];
let sectionDice = Math.random();
let sectionNumber = Math.floor(timesArray.length*sectionDice - 1 );
console.log(`chosen section is ${sectionNumber +1}`);
console.log(`its seek value is ${timesArray[sectionNumber]}`);
console.log(`its end value is ${timesArray[sectionNumber +1]}`);
let startTime = timesArray[sectionNumber];
let endTime = timesArray[sectionNumber+1];
let duration = endTime - startTime;
console.log(`its duration is ${duration}`);
let thisTime = timesArray[sectionNumber];
experience.sharedParams.update('seek', thisTime);
experience.sharedParams.update('transport', 'Start');
function myFunction() {openform();}}
setTimeout(myFunction, duration*1000);
```

Sandeep Bhagwati defines four categories of real-time music scores [7]:

- Permutational, where sections can be performed in a different order each time.
- Parametric, where more parameters are left free to the performer.
- Auto-reflexive, where the performer's actions have an incidence on the unfolding of the piece.
- Co-creative, where audience and/or conductor can interact with the display of the notation.

According to this taxonomy, *And the Sea* (the piece whose score is accessible through smartvox.eu) is permutational, as is, for instance, Pierre Boulez's 3rd Sonata, 2nd movement (*Formant 2 – trope*), in which the four sections (*Commentaire, Glose, Texte* and *Parenthèse*), can be performed in different orders. As Freeman observes, these new forms of algorithmic notation closely relate to the aesthetic of the *open-form*, and even constitute a revival of the essential questions addressed in the 1960s by Umberto Eco, Alexander Calder, and Earle Brown, among others. As Freeman states: "Real-time music notation systems draw from a broad spectrum of algorithmic composition environments that produce music notation. They are also influenced by an open-form aesthetic in which a musical score is read differently in each performance of a composition" [8]. However the generation of the material can obviously go beyond the mere permutation: "I outline a new form of computer-assisted composition, in which the author, in the classical sense, recedes and his artifact, the score – dynamically generated from algorithms – exists only in the moment of its creation" [10]. The idea of an ephemeral piece that permanently generates itself is in itself extremely attractive, but, by putting the performer in an unpredictable situation, its benefits can also be called into question: "While this immanence has often been perceived as a force for the emancipation of performers and spectators, it can also give rise to unaccountability" [7].

[2] To see the generation in action, go to **smartvox.eu**, choose a different instrument (e.g. piano, flute, cello) on each tab (or each device), and press the play button to unlock the video. After few seconds, the videos start wandering semi-randomly along the timeline of the video.

2.3 Client-side synchronization

An recent update in SmartVox consisted of implementing a client-side synchronization algorithm, exposed in Section 4.3, allowing for unprecedented temporal accuracy between players/singers.

3. CHALLENGES IN PRODUCTION

Production and research have different goals. A public performance in a concert hall demands reliable technologies, whilst the development of a notational environment such as the one presently described can only improve by testing its limits. The use of smartphones in rehearsals, workshops, or in pedagogical contexts is generally accepted with enthusiasm by musicians. This distributed system, however, still presents several risks in performance (see Section 4.1), and demands that its technical limitations be overcome in order to succeed in forthcoming productions.

3.1 And the Sea

And the Sea, commissioned by the SKAM [3] collective, was written for voice, cello, flute, and piano. SmartVox was originally dedicated to vocal ensembles, sending an audio-score as well as visual information; for this piece however, the instrumentalists only received visual notational information. In spite of the three major updates discussed in Section 2, on the day of the performance, the piece had to run locally (the devices were not connected to the internet, but to a LAN – Local Area Network), and was played from the beginning to the end [4] , i.e. not in its algorithmic form (unlike the smartvox.eu website, where the timeline/form is constantly being generated once the player unlocks his video, pressing the play button). All the rehearsals until the day before the concert were nevertheless practiced and synchronized through the algorithmic-score website (**smartvox.eu**). The animated notation was also sent to the performers in the linear (non-algorithmic) version [5] , and the performers never expressed the need or desire for a printed version of the score. The system proved to be helpful and easy to use for musicians; they could read their score independently without having to rely on a conductor, and could be placed very far away from each other: the singer was freely walking around the church (Figure 1) during the whole performance, the piano was on the altar, the cellist in the middle of the audience, and the flautist was in the organ loft. The animated notation also helped especially for the synchronization to the 8-channel tape of electronics. The placements of the speakers, finally, was also greatly simplified by the setup, since it did not require any sound card, nor the use of a mixing desk, but only a few cables and four phones, connected to two mini-loudspeakers each, and placed all around the audience.

[3] Stuttgarter Kollektiv für Aktuelle Musik: http://skam.io/

[4] The trailer of a piece performed with this score is available here: https://youtu.be/prcXUbhd-ZY

[5] The parts were available as youtube links. The piano part, for instance, can be accessed at the following address: https://www.youtube.com/watch?v=QByxPXItxHs

Figure 1. *And the Sea*, SmartVox used in performance with SKAM.

3.2 SmartVox, the piece

The SmartVox piece/production, for 5 soloists, choir, and electronics, was premiered in Nantes in March 2017 [6] . Involving a local choir each time, this project has a participative aspect that makes it financially viable. SmartVox will therefore be sung in several French cities in 2018-19: Metz, Rouen, and Caen. In spite of its focus on web technologies, the piece relates to ancestral traditions, first because of its sacred text [7] , and secondly because of its polychoral construction: several choirs are placed in different locations, around the church. One of the aims was therefore to highlight the creative act that involves “reading early music today” [14], or any form of interpretation of ancient texts. The use of audiovisual notation for this piece was also justified by its microtonal language, because of the confusion that the notation of such intervals may cause to some singers. This use of notation as an aid for unfamiliar idioms relates to the work of G. Hajdu [4], who proposes that dynamic notation can provide solutions to the learning of non-standard (e.g. microtonal) musical practice. In this piece, the composition workflow consisted of analyzing the frequencies contained in a recorded or synthesized sound [8] , in order to subsequently compose melodic lines within this harmonic grid. [9]

3.3 Le temps des nuages

This piece, premiered in January 2018 [10] , sets poems by French philosopher Michel Onfray. It used *SmartVox* on a much larger scale than in previous attempts: five singers (the *De Caelis* ensemble), five percussionists (the *Links* ensemble), four channels of electronics, and 74 junior high-school students were placed around the audience. The technical challenge here was to handle eighty connections simultaneously. For rehearsal purposes, each separate part was accessible through the address nuages.smartvox.eu. The size of the concert hall (600 seats) and the number of con-

[6] A live recording and animated full score of the piece is available here: https://youtu.be/8R4Twc1A7Ks?t=1

[7] The piece is based on the old testament, in Hebrew and in its French translation by André Chouraqui, often de-constructed using algorithmic processes.

[8] An example of a capture in *Audiosculpt* [15] shows the spectrogram of a synthesized sound: https://youtu.be/8OlkZa7cTl4

[9] The same electronic sound is then used as a harmonic canvas: https://youtu.be/Xh1Vxe_lQ-U?t=66

[10] A recording of the piece is available at the following address: https://youtu.be/SyFdR2HiF00

nexions required three powerful wifi antennas in order to irradiate the whole room (where the singers stood). Node.js had previously experienced difficulty when too many clients requested heavy files in a short period of time. On this occasion, *nginx* (see Section 4.3) was successfully tested as *reverse proxy*, in order to manage load balancing, cache static content (the videos) and manage port contention between clients.

4. TECHNICAL CONSIDERATIONS

The above-mentioned pieces need to overcome a major difficulty: the devices of the participants are always different from one another. Although the performances of mobile devices improve very rapidly, unexpected behavior can always occur. Section 4.1 lists the causes to the problems faced by SmartVox since 2015, mainly concerned with synchronization and network overload. Section 4.2 measures the delay between phones in different situations. Section 4.3 exposes a solution which highly improved synchronisation across devices, developed by Benjamin Matuszewski in January 2018.

4.1 Description of frequently faced problems

4.1.1 Standby/sleep mode

In the former version, most accidents (in concerts and in rehearsal) occurred when the performers' device switched to sleep mode. A standby on the client-side was in most cases likely to cause synchronization problems, or even sometimes interruption of the WebSocket connection, in which case the only possibility that remains is to reload the page. A standby on the server side evidently interrupted the connection between clients. Sleep mode was the most problematic behavior of smartphones for this application, since the '*start*' message was only received by the phones once the sleeping period was over, hence causing long delays between parts. [11]

4.1.2 Other Breakdown factors

At its previous state, the app was already reliable in rehearsals: if one device got out of sync or disconnected itself, it could be updated on the next 'start' message of the conductor in most cases. For concerts, based on the results discussed in Section 4.2.1, if all the devices started exactly together, there was no need to adjust timing, since all devices could keep in time with each other. Whilst this way of performing music has been the object of great interest from nonprofessional singers, in performance situations, with more than twenty singers on stage, a single user interaction was often likely to disturb the beginning of the piece, which often only run smoothly (e.g. with all the singers in sync) only three or four minutes after the piece had started. Among these user interactions can be listed:

- Idle mode: switching between applications may disturb the clock synchronization and/or pause the video.

[11] iOs devices seemed able, unlike Androids, to receive a '*play*' message while sleeping; an iPhone on standby could start the video approximately at the time the 'start' message was received.

Figure 2. Time differences between several devices were measured with snapshot pictures.

- Putting on/taking off headphones often causes the video to pause.
- When playing a video in full screen, phones' media players usually propose a 'done' button, which enables the user to escape to the fullscreen mode; this action causes the video to pause.

One frequent problem encountered in rehearsal (and in concerts...) consisted of a pause of the performer's media player. As a workaround solution to this issue, in the algorithmic piece *And the sea* for instance, the '*seek*' update is always paired with a '*start*' message. [12] This message allows the user to cancel the 'pause' behavior exposed above.

4.2 Measurements of timing accuracy

As a conductor working regularly with this application, the main difficulty so far has been to cope with temporal issues, i.e. when the phones are not exactly in sync with each other [16]. Extensive research has been made in the domain of "Synchronization for Distributed Audio Rendering over Heterogeneous Devices, in HTML5" [17], which shows that extreme timeliness can be achieved. In the present study, measurements were realized in order to understand where dysfunction might come from, so as to improve timing accuracy in the context of rehearsals and/or concerts. The application being constantly used with a variety of devices, the measurements were made with different types of smartphones: iPhone 5, iPhone 4s, Oppo r5, Huawei C8815, Miui 8.5, HTC 802w. Figure 2 shows a typical snapshot picture taken while the playhead is crossing the staff from left to right.

4.2.1 Drift

A hypothesis has been put forward that the media-player integrated into the browser's phone may experience latency while reading the mp4 file, and subsequently cause a delay after some time. To measure this, 5 devices (p_1, p_2, p_3, p_4,

[12] In the coding example Section 2.2, the '*start*' message corresponds to experience.sharedParams.update('transport', 'Start');

	t_1	t_2	t_2-t_1	Drift
p_1	1”8	14”10	12”3	0”02
p_2	1”95	14”4	12”45	0”13
p_3	1”85	14”15	12”3	0”02
p_4	2”03	14”35	12”32	0 (reference)
p_5	2”00	14”3	12”3	0”02

Table 1. Temporal drift from p_4

	t_1	t_2	t_3	t_4	t_5	t_6
p_1	1”45	1”85	1”5	2”7	2”2	3”95
p_2	1”5	2”1	1”3	2”8	2”35	4”1
p_3	1”45	2”0	1”2	2”75	2”25	4”05
p_4	1”6	2”1	1”25	2”8	2”35	4”15
p_5	1”30	2”10	1”3	2”9	2”4	4”15
sum	0”35	0”35	0”35	0”25	0”3	0”3

Table 2. Local server with sync module.

p_5) were photographed simultaneously twice, while reading and displaying a seven-minute file displaying a score with a timeline. The times displayed on p_1 were 1”8, and 6’14”10 (the 6 minutes are not displayed on the table for clarity). p4 was chosen as the reference from which the drift should be calculated. The results of the experience (see Table 1) showed that the drift from p_4, being lesser than 100 milliseconds, can be considered null for our purpose.

4.2.2 Local server

As stated in Section 2.1, one of the main recent improvements consisted of hosting remotely the server that was initially used locally. The following measurements will try to determine how much a distant server impacts on the synchronization of the setup. The sum row adds up the differences between the mean value [13] and the other devices' values (see Table 2).

4.2.3 Distant server

The same experience with a remote server (i.e. accessing the application over the internet) reveals slightly greater sum values (see $\mathbf{t_5}$ and $\mathbf{t_6}$ in Table 3), and therefore, less precise time accuracy.

[13] For instance in Table 2, the mean value for the sixth measurement t_6 is 4”1, the value displayed by the second phone p_2.

	t_1	t_2	t_3	t_4	$\mathbf{t_5}$	$\mathbf{t_6}$
p_1	4”1	4”55	3”35	3”30	7”60	5”0
p_2	4”1	4”6	3”35	3”25	7”55	5”0
p_3	4”0	4”45	3”25	3”20	3”90	5”9
p_4	4”15	4”6	3”35	3”25	7”65	5”05
p_5	4”05	4”5	3”20	3”35	7”35	5”0
sum	0”2	0”25	0”25	0”2	**3”95**	**0”95**

Table 3. Distant server – with sync module.

	t_1	t_2	t_3	t_4	t_5	t_6
p_1	4”65	10”8	6”9	12”9	7”9	9”3
$\mathbf{p_2}$	0”05	3”75	2”85	4”55	2”05	2”15
p_3	4”75	5”75	6”8	12”85	7”95	9”25
p_4	4”5	10”75	6”7	12”82	7’80	9”15

Table 4. Distant server – different networks – $\mathbf{p_2}$ failed loading the page correctly.

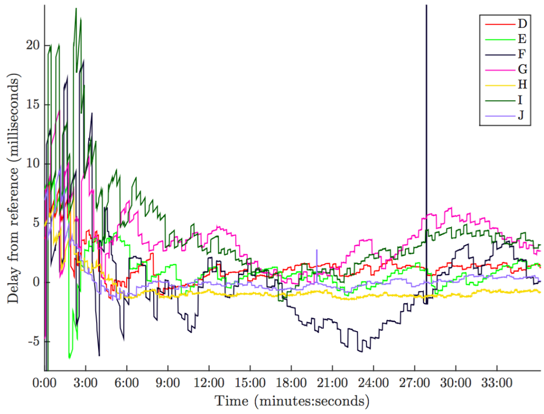

Figure 3. Synchronization of 7 heterogeneous devices. (Extract from J. P. Lambert [17]). The process still improves after 4-5 minutes.

4.2.4 Distant server – different networks (3G, 4G, Wifi...)

Each device is connected to the internet differently, i.e. only one of them is connected via local wifi. Phone N°2 is a recent iPhone (5s), the browser is Safari. The measurements were taken without reloading the page, highlighting a constant dysfunction of phone N°2: the intermittence of the 4G network on this phone may have contributed to the page load deficiency (see Table 4, $\mathbf{p_2}$). Once the page reloaded, the behavior was normal again.

4.2.5 Sync Module

The latest version of SmartVox has implemented a *sync module* [17] [18], which provides an elaborate solution that permanently computes an estimation of the reference time, in order to sync all the connected devices to the same shared clock. According to this reference time, the server can set-up a *look-ahead* scheduler, delaying messages (of e.g. 2 seconds), in order to leave enough time for all devices to receive this message at exactly the same time. Figure 3 shows that the synchronization of 7 heterogeneous devices gradually improves over time. A comparison be-

	t_1	t_2	t_3	t_4	t_5
p_1	6”15	5”85	5”45	9”2	1”4
p_2	6”4	5”9	5”4	9”45	1”4
p_3	6”3	5”9	5”55	9”05	1”4
p_4	6”5	5”85	5”9	9”55	1”4
p_5	6”25	6”4	5”75	11”05	1”75
sum	0”5	0”6	0”8	3”5	0”45

Table 5. Local server – without sync module.

tween table 2 and Table 5 demonstrates that the synchronization improves when the *sync* module is activated.

4.3 Synchronization update

The measurements of section 4.2.2 (performed on a local server) showed a satisfying synchronisation between devices. This confirmed the assumption that the important delay experienced between parts in rehearsals were most of the time due to user interactions or 'sleep mode' exposed in section 4.1. As a remedy to this issue, a solution was found by Benjamin Matuszewski in order to update dynamically the client's timeline, whenever he/she gets out of sync: every tickPeriod (for instance every second), on the client-side, the local time (syncTime) and the local timeline or transport (videoCurrentTime) are compared to the server's global time (triggerSyncTime) and global timeline or transport (transportTime). In the case presented here, if the difference (jit) exceeds 0.5, the local (i.e client-side) timeline is changed.

```
onUpdateTime(transportTime, triggerSyncTime) {
  if (!this.isReady)
    return;
  const syncTime = this.syncScheduler.currentTime;
  if (triggerSyncTime > syncTime) {
    this.syncScheduler.defer(() => {
      const videoCurrentTime = this.$video.currentTime;
      const jit = Math.abs(transportTime - videoCurrentTime);
      if (jit > 0.5) {
        this.$video.currentTime = transportTime;
      }
    }, triggerSyncTime);
  }
}
```

This new release of the application was used for two productions in 2018 (*Le Temps des Nuages* in January, and *Smartvox* in April), and gave promising musical results, with far greater clarity in the polyphony, and in homorhythmic responses between groups of singers.

5. GOING FURTHER

5.1 Dialoghi Spezzati

This piece, for twelve voices, twelve channels of electronics and *organetto*, was composed for the Mucem museum in Marseille and was performed with SmartVox. Since the application is essentially a multichannel video player, this piece explored the possibility of syncing live singers (each singer being guided by his/her audiovisual score) with filmed singers (displayed and accessed through the application, like the scores of the singers). An interesting dialogue could be perceived between live performers and recorded performers, displayed on screens. A natural continuation of this idea would be the implementation of WebRTC, adding visual and audio input and output to each terminal of the web application, to create a dialogue with remote performers.

5.2 Pedagogy in Classroom

SmartVox was tested this year (2017) on a weekly basis with 2nd year musicology students, in Aix-Marseille University. For this course, about Renaissance music, the application was particularly useful because it is mainly concerned with polyphony: each student could read and hear his own part on his device (phone, tablet, or laptop), with the full score [14] projected on the board of the classroom. The students were therefore able to sight-read up to eight-part complex polyphonies [15] with very little difficulty. [16]

5.3 Smartphones used as an instrument

The role of SmartVox is to turn the participants' devices into scores, but phones are often conceived as an orchestra of musical instruments rather than a notational tool [2][19]. With a similar architecture (a distributed web application), a wide range of user interactions can be imagined, mapping the user's gesture (such as screen-click, compass turn, accelerometer motion...) to a sample or a musical parameter. These types of musical experiments are strongly evocative of video games, and let us envisage playful forms of interactions with audiences. [17]

6. CONCLUSION

The present study concerns the realms of *networked musical performance* and *computer-aided performance*. The rapid evolution of smartphones, tablets and web technologies lets us hope that the technical problems listed above will soon be overcome. Musically, however, these limitations have strongly shaped the music I have written in recent years. In 2007, I realized experiments with a string quartet and four click tracks (a primitive form of wired networked performance), where the focus was put on extreme timeliness between players and electronics placed around the audience. [18] Years later, having accepted the temporal issues that can appear when working with web technologies, the focus was put on harmonic coherence, and tolerating a minimum of delay between parts, rather than on rhythmic exactitude. The present measurements have nevertheless shown that a more precise time writing can be achieved, thus allowing many different kinds of music or performative situation to be imagined.

Acknowledgments

I am very grateful to Norbert Schnell, who directed the COSIMA ANR research fund, and Benjamin Matuszewski, web developer at IRCAM, who coded SmartVox in the SoundWorks framework [2][20].

[14] For each piece, experience has shown that making a full score (conductor's score) available among all available parts is now considered best practice, since it is often very useful for performers.

[15] The scores are available on the internet, at the following addresses: tallis.smartvox.eu, josquin.smartvox.eu, dunstaple.smartvox.eu, dufay.smartvox.eu, avignon.smartvox.eu, canon.smartvox.eu; in order to conduct all the performers connected to the same score, the conductor should add the suffix /conductor on his device, e.g. canon.smartvox.eu/conductor

[16] The pieces studied in class were then recorded by the students in a church, with SmartVox, the result be heard here: https://www.youtube.com/watch?v=bofWvTCNNKI&t

[17] *Your Smartest choice*, by Huihui Cheng and Benjamin Matuszewski, was created at the Eclat festival 2017 in Stuttgart. https://github.com/ircam-cosima/your-smartest-choice. A demo is available here: https://www.youtube.com/watch?v=WKvhUJEE90k

[18] An extract of the piece is available here: https://www.youtube.com/watch?v=gOGMo_uwnlo

7. REFERENCES

[1] J. Bell and B. Matuszewski, "SmartVox. A web-based distributed media player as notation tool for choral practices," in *Proceedings of the 3rd International Conference on Technologies for Music Notation and Representation (TENOR)*, A Coruña, Spain, 2017.

[2] N. Schnell and S. Robaszkiewicz, "Soundworks – A playground for artists and developers to create collaborative mobile web performances," in *Proceedings of the Web Audio Conference (WAC)*, Paris, France, 2015.

[3] A. Agostini and D. Ghisi, "bach: An environment for computer-aided composition in Max," in *Proceedings of the International Computer Music Conference (ICMC)*, Ljubljana, Slovenia, 2012.

[4] G. Hajdu, "Dynamic Notation – A Solution to the Conundrum of Non-Standard Music Practice," in *Proceedings of the International Conference on Technologies for Music Notation and Representation (TENOR'15)*, Paris, France, 2015, pp. 241–248.

[5] 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/

[6] J. Bean, "denm (dynamic environmental notation for music): Introducing a Performance-Centric Musical Interface," in *Proceedings of the International Conference on Technologies for Music Notation and Representation (TENOR'15)*, Paris, France, 2015, pp. 74–80.

[7] S. Bhagwati, "Vexations of ephemerality," in *Proceedings of the International Conference on Technologies for Music Notation and Representation (TENOR'17)*, Coruña, Spain, 2017, pp. 161–166.

[8] J. Freeman, "Extreme Sight-Reading, Mediated Expression, and Audience Participation: Real-Time Music Notation in Live Performance," *Computer Music Journal*, vol. 32, no. 3, pp. 25–41, 2008.

[9] G. Hajdu, "Playing performers. ideas about mediated network music performance," in *Proceedings of the Music in the Global Village Conference*, Budapest, Hungary, 2007, pp. 41–42.

[10] ——, "Disposable music," *Computer Music Journal*, vol. 40, no. 1, pp. 25–34, 2016.

[11] L. Gabrielli and S. Squartini, *Wireless Networked Music Performance*. Springer, 2016.

[12] J. Lazzaro and J. Wawrzynek, "A case for network musical performance," in *Proceedings of the International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV)*, Port Jefferson, New York, USA, 2001, pp. 157–166.

[13] G. Hajdu, "Embodiment and disembodiment in networked music performance," in *Body, Sound and Space in Music and Beyond: Multimodal Explorations*, C. Wöllner, Ed. Routledge, 2017.

[14] J. Ferrando, "Reading early music today: Between reenactment and new technologies," in *Proceedings of the Computer Music Multidisciplinary Research conference (CMMR)*, Porto, Portugal, 2017.

[15] N. Bogaards and A. Röbel, "An Interface for Analysis-Driven Sound Processing," in *Audio Engineering Society Convention 119*, New York, USA, 2005.

[16] C. Chafe, "Living with Net Lag," in *Proceedings of the 43rd Conference of the Audio Engineering Society (AES)*, Pohang, Korea, 2011.

[17] J. P. Lambert, S. Robaszkiewicz, and N. Schnell, "Synchronisation for Distributed Audio Rendering over Heterogeneous Devices, in HTML5," in *Proceedings of the Web Audio Conference (WAC)*, Atlanta, USA, 2016, pp. 6–11.

[18] N. Schnell, V. Saiz, K. Barkati, and S. Goldszmidt, "Of Time Engines and Masters An API for Scheduling and Synchronizing the Generation and Playback of Event Sequences and Media Streams for the Web Audio API," in *Proceedings of the Web Audio Conference (WAC)*, Paris, France, 2015.

[19] J. Oh, J. Herrera, N. J. Bryan, L. Dahl, and G. Wang, "Evolving The Mobile Phone Orchestra," in *Proceedings of the International Conference on New Interfaces for Musical Expression (NIME)*, Sydney, Australia, 2010, pp. 82–87.

[20] D. Poirier-Quinot, B. Matuszewski, N. Schnell, and O. Warusfel, "Nü Soundworks : using spectators smartphones as a distributed network of speakers and sensors during live performances," in *Proceedings of the Web Audio Conference (WAC)*, London, UK, 2017.