Live Writing: Writing as a Real-time Audiovisual Performance

Sang Won Lee
Computer Science and Engineering
University of Michigan
2260 Hayward Ave
Ann Arbor, MI 48109-2121
snaglee@umich.edu

Georg Essl
Electrical Engineering & Computer Science
University of Michigan
2260 Hayward Ave
Ann Arbor, MI 48109-2121
gessl@umich.edu

Mari Martinez
School of Music, Theatre & Dance
University of Michigan
1100 Baits Dr.
Ann Arbor, MI 48109-2085
faeries@umich.edu

ABSTRACT

This paper suggests a novel form of audiovisual performance-live writing- that transforms creative writing into a real-time performing art. The process of typing a poem on the fly is captured and augmented to create an audiovisual performance that establishes natural links among the components of typing gestures, the poem being written on the fly, and audiovisual artifacts. Live writing draws upon ideas from the tradition of live coding in which the process of programming is revealed to the audience in real-time. This paper discusses the motivation behind the idea, interaction schemes and a performance interface for such a performance practice. Our live writing performance system is enabled by a custom text editor, writing-sound mapping strategies of our choice, a poem-sonification, and temporal typography. We describe two live writing performances that take different approaches as they vary the degree of composition and improvisation in writing.

Author Keywords

Writing, Live Coding, Sonification, Web Audio

ACM Classification

H.5.5 [Information Interfaces and Presentation] Sound and Music Computing, H.5.2 [Information Interfaces and Presentation] User Interfaces—Screen design, J.5 [Computer Application] Arts and Humanities—Performing arts (e.g., dance, music).

1. INTRODUCTION

Writing is a rich form of communication and we live in an age when we produce large volumes of writing through digital platforms such as the World Wide Web, and mobile devices. As writing is an expressive process guided and adapted by thoughts that evolve over time, the writing process includes improvisational aspects that resemble music performances. However, traditionally there is a separation between the process of writing and how written results are presented to an audience. At the highest level, a piece of writing tends to be presented to the readers is rather in a linear fashion. Static text does not expose the temporal dynamic of the actual writing process. Rather it presents a fixed, monolithic result that is usually consumed by readers at a pace of their choosing. What motivates this work is the dynamic expressive process that is hidden in the final text in order to transform the process of writing into a real-time performing art. In this paper, we review the temporal dynamics that are normally overlooked in the writing process and draw ideas from the tradition of live coding to propel the study. The paper then discusses approaches to transform a piece of writing into an interactive performance with respect to the writing interface, input modalities and its mapping, and temporal typography, the technique of dynamic text rendering that we suggested in a previous work [14]. Lastly, we share our experiences with two performances of live writing.

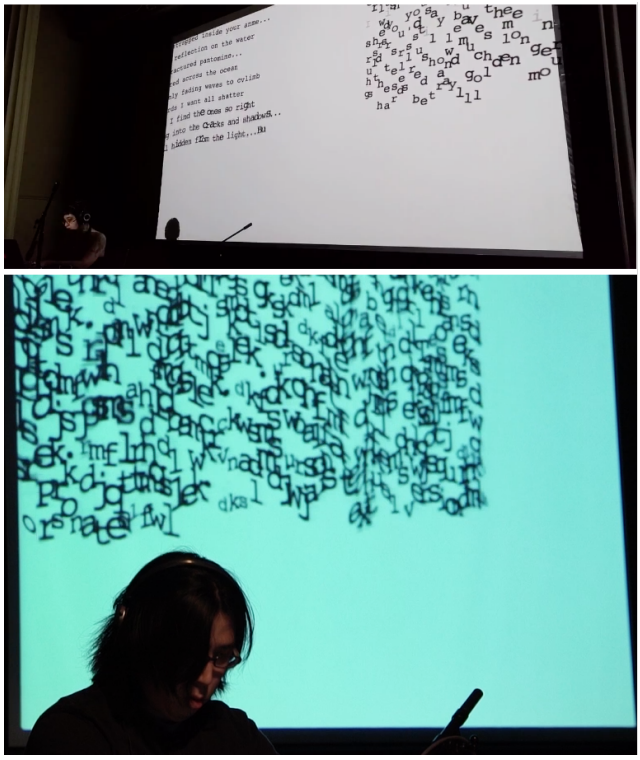

Figure 1: Footages of Live Writing Performances

2. WRITING AS A REAL-TIME PROCESS

Writing is a real-time process and so is reading. But traditionally reading and writing are disjointed, giving writing the appearance of static monoliths that are presented for a reading that can have very different characteristics than the

NIME'16, July 11-15, 2016, Griffith University, Brisbane, Australia.

original writing had at the time of creation. If one writes in front of the public by sharing the screen of a computer (or hand-writing), one can already be providing a convincing performance. In fact, reading aloud a piece of writing is a common artistic expression of public reading, such as live poetry, authors' book reading events or verse drama. As the performative aspects of (verbal) live poetry was studied in [19], the oral presentation of a poem has the additional dimensions of expressivity for a reader such as accent, tone, and pace. What this paper argues is that live writing is a textual version of live poetry where the text is used as the medium.

The pace and rhythm of typing can be a powerful dimension for expressive writing. Various kinds of writer's cognitive and emotional states (such as contemplation, hesitation, confidence, or agitation) can emerge during typing, thanks to the temporal patterns of their keystrokes, for instance, pauses, bursts, or cursor highlights. In addition, corrective steps are ways to reflect changes in a writer's thought process. For example, if some of the text is deleted, it reveals the dynamic and often non-persistent nature of emergent ideas. Hence, deleted text builds a powerful message not only showing the initial, aborted thoughts but also showing the oscillation in the thought process. Some ideas that have been considered during writing may be completely absent from the a final product yet dramatically change the nuance of the final text.

Once a writer understands the real-time nature of writing, temporal patterns become part of a writer's expressive vocabulary and performance technique. Writers can deliberately take advantage of the additional expressive dimension to structure the writing as a real-time performing art. Deploying such expression in the process of writing is similar to choreographed visuals and organized sound in an audiovisual performance. This way a writer can "compose" a live writing music performance and rehearse it to deliver the piece as it is composed. On the other hand, the process of writing can be similar to a musical improvisation, where the writing itself can be written by a performer on the fly. We believe that for a performer both approaches have different potentials and challenges, and these are discussed below as the paper analyzes two performances of live writing.

Hoping to transform everyday written communication into a real-time performance, the authors here have developed another Live Writing system[1] that can record all the keystrokes and asynchronously replay the writing in real time [13]. While this web-based writing platform does not provide any other functionalities than replaying the user interactions, it supports existing written communication such as email and blogging with these real-time replay functions so that anyone can perform a writing. It also aims to support the asynchronous replay of a live coding performance with symbolic information (as opposed to audio/screen recording). Sharing the common goal of transforming writing into a real-time experience, this paper focuses on the synchronous communication between a writer and the audience in live writing as an audiovisual performance.

3. LIVE CODING PEOPLE'S MIND

The notion of live writing is directly inspired by the emerging field of live coding [6] in computer music and creative coding. Live coding is an audiovisual performance where a programmer writes a program on stage in front of an audience. The outcome of the program is generative music and perhaps visuals. In live coding music performances, the programming language can be viewed as a musical instrument [2]. It is typical to project program code text in the performance space for the audience to understand the process of music making. This highly visible nature of the coding process makes explicit the performance aspect of code writing. This principle is well captured in the following statement of *TOPLAP*[2] (the live coding community) manifesto: "Obscurantism is dangerous. Show us your screens."

In a broader sense, live writing shares a common artistic aesthetic with live coding in that both practices reveal the whole creative process to the audience and are "composed" to some extent on the fly. They are also similar in how their textual interactions (keystrokes) serve as the main gesture of creation as well as their indirect mapping strategy (as opposed to one-to-one gesture-sound mapping in a traditional musical instrument). The written text can also be interpreted by a machine to generate audiovisual outcome (one by the interpreter, the other by natural language processing techniques). Lastly, as with live coding, the core of live writing performance is having the writing projected on the performance space.

Live coding can be a special case of live writing where the writing outcome is code text that runs live. It also implies that a live writing performance can vary drastically from live coding, as the performance is not confined to a rigid syntax that is meant to allow algorithmic consequences. In turn, live writing is not powered by the expressivity of live algorithms written in a programming language. Hence, audiovisual outcomes must be carefully designed and developed in advance of the performance, making it more similar to a composition with other digital musical instruments. The style of music can mainly left to a composer and it should be tightly coupled with the text to be written, for example, the style of music is not necessarily generative music or an algorithmic composition.

The general challenge for a musician of computer music is often to bridge the gap between the audience and the performer using a laptop on a table. Similarly, live writing allows the audience to experience what is normally hidden from and puts them immediately behind the screen with the performer. A piece of writing that shares the emotional state of a performer with the audience in a human readable form can resonate in readers' minds directly with strong messages coded in the text. With the metaphor of live coding, the code text written in live writing performance utilizes the expressivity of the language we speak (as opposed to the language that a machine speaks) and the target object that the code influences will be the minds of readers (as opposed to a computer program that generates music). In this sense, one could consider live writing to be a metaphorical form of live coding the audience's mind.

4. WRITING PERFORMANCE INTERFACE

Writing by itself does not generate music in general. Composers and developers put efforts into providing a writing environment that facilitates a computer music performance. In this section, we focus on the writing software, input modalities of the textual music-making system, and audiovisual outcomes, the integration of which transforms the writing activity into a real-time performance.

4.1 Text Editor as part of Composition

The typical writing interface can vary from a complex word-processor to a simple text editor with a cursor. One core aspect of the editor in live writing he editing environment in live writing must be visible to the audience. It may be argued that one could create an interface for a writer

[1] `http://www.echobin.com`

[2] http://www.toplap.org/

that is separated from the visualized projection so as to have more creative flexibility and expressivity, though at the same time hiding the real performance interface. We believe, however, such separation would preclude the performance being an act of live writing. Indeed, such an arrangement means interaction that goes beyond the writing activity. The disjunction between the visualization and the performer hidden behind the screen obscures the audience's understanding on the main idea of live writing. In addition, the synchronized presentation of the performer's writing to the audience expands the traditional what-you-see-is-what-you-get (WYSIWYG) quality of a text editor in the temporal dimension.

The choice of editor is a significant factor. Not only is it the medium through which a performer delivers the aesthetic of the composition but it is also the environment that renders the visual representation. In the live coding context, a musician chooses a coding environment based on the performer's preference and the programming language used; quite often the text editor is augmented with the custom-developed visualization integrated in to the editor for better audience communication [11, 15, 18, 21]. Although visualization is not essential to live writing, it attracts live writing musicians to tightly connect the poem written and the music via the visuals in order to make the performance more expressive, responsive, and "live". Given the visualization and music vary by performance, engaging with a custom text-editor is part of the composition process through which musicians realize live writing with, text, sound and visuals in sync.

4.2 Gesture-Sound Mapping in Live Writing

In order to augment a piece of writing in a musical way, a composer needs to carefully design a music-making system that takes the process of writing as a main input to the system. In the process of plain writing, diverse input signals can be captured to generate and trigger sounds at different levels ranging from the laptop's native physical inputs to the semantic meaning of the content written. The variety of choice in input modalities enables a composer to develop a compound strategy of gesture-sound mapping. This section reviews the available input modalities in live writing and discusses our mapping strategies to utilize each modality. Note that our discussion of gesture-sound mapping in live writing is limited to the case of typography media and exclude handwriting/drawing using pen-based musical instruments [10].

4.2.1 Keystroke as a Musical/Linguistic Note

The basic gesture of live writing is typing. The keystroke is a physical action that can be used as an input gesture of a digital musical instrument. Fiebrink et al proposed [9] mapping strategies to transform the "qwerty" keyboard into an instrument. Their intent was not to preserve writing but to use the keyboard as an input device. A similar approach can be applied to use each key as a direct trigger of certain sounds and map each letter to a different musical property of the sound. As a typical keyboard today allows binary interaction (key on/off) per key, the expressivity is limited to mapping each key to different musical properties (e.g., pitch) and the duration of keypress. This limit can be overcome if the system utilizes a pressure-sensitive qwerty keyboard [7] or a developer augments existing keyboards with additional sensors, approaches similar to those that have been attempted on the touchscreens of mobile phones [8].

The major difference of using a qwerty keyboard in live writing from previous approaches is that a performer presses keys under the set of syntax rules that are defined by the language in which it is written; the outcome of typed letters must be semantically meaningful. For example, if a key is mapped to generate a pitched note, playing a musical melody will generate random text that looks like gibberish. Therefore, if each keystroke triggers a certain sound, the generated sequence of sounds is the outcome determined by the combination of key-sound mapping and the written text. Word choices determines the composed melody.

While it seems that generated sound is a random outcome determined by the mapping and the content, a performer can organize sound in musical way by arranging words and sentences. For example, if a writer repeats exactly the same (or similar) sentences twice, the inherent rhythm of typing these two sentences repeats the same notes. Furthermore, applying a certain poetic form can be a way to organize sound. Typing a sonnet with a consistent rhyme scheme will sound more regular than typing a long narrative poem. Lastly, the typing rhythm over the writing structure (or even at the word level) can be arranged to musically, which requires a performer to practice typing the poem differently from naturally typing. This way, a temporal structure in writing process can emerge from the typing gestures as opposed to just writing constantly.

Another interesting aspect of keystroke-triggered sound is that the content(or the tunes) is dominated by the language it is written in. Based on the language, a musician can develop a different virtuosity because every language should have a specific characteristic that makes the sequence of sound more regular. For example, for the sentence structure of Korean or Japanese, which is a subject - object - verb (SOV) language, sentences can be written in a way that all sentence uses one verb conjugation form and they end with the same sequence of letters (e.g., Korean: -nida, Japanese: -desu, -masu) Therefore, if a musician performs in such a language, each sentence can have a same four letters at the end, giving the end of sentence a certain musical cue, which is exactly the case in our first performance. Due to the complex interdependency among language choice, creative writing, mapping choice, poetic elements, and the execution of rhythmic typing by a performer, live writing challenges a composer and performer to develop their virtuosity in a truly interdisciplinary way in order to turn a sequence of keystrokes into organized sound.

4.2.2 Live Capture of Typing Sound

As noted above, due to the limits of a keyboard, keystrokes usually fail to capture the intensity of keystroke gestures, an intensity often linked to the dynamics in musical instruments. In turn, we find the typing sound captured from a microphone can be a good way to express the dynamics of the performer's gestures, similar to the case of using the typewriter as an acoustic instrument [1, 20]. The dynamically changing typing sound, amplified directly through the main speaker, reflects the performer's intention in expressing the words and the sentences at the moment with their semantic meaning. In addition, the amplification of keystroke dynamics not only serves as the textual version of poetic feet in the writing process but also provides rhythmic and percussive components in the music. Note that the typing sound is synchronized with the visuals (letters on screen) and the keystroke-triggered sound discussed in 4.2.1. These actions effectively clarify the idea of live writing to an audience.

One practical challenge of amplifying of keystrokes at a performance is that the microphone shall be placed in such a way as to capture the sound but is then prone to audio feedback; after all, the typing sound is very soft compared

to the instrumental sound. Such risk can be reduced by carefully placing a directional microphone with respect to the location of main speakers (usually behind the speakers at the side of the stage). Further dangers for feedback can be controlled by providing a headset for the performer to allow for monitoring typing sound with the direct control on the microphone input.

Another useful technique to utilize typing sound is to connect the intensity of the sound to control certain properties of the audiovisuals. For instance, the sound of typing can be used to control the loudness of other audio signals (sample or synthesized sound). This maps the dynamics of typing intensity to the intensity of another voice. Or one can simply use a convolution to process the live input sound with other audio sample voices. The risk of audio feedback still remains in these example mappings. On the other hand, the typing sound can be used only to express dynamic visualization, which removes the audio feedback problem. For example, the loudness of typing sound can be connected to changing the font size of each letter typed. (See the first example of Figure 2.) The built-in microphone on a laptop can be useful for the visualization purpose due to its proximity to the keys and its omnidirectional property. Using the built-in microphone for visualization is discussed further in 4.3.

4.2.3 Sonification of a Poem

Poetry is a form of expression in which time, dynamic, and expression play particularly important roles. This makes a poem a particularly appealing form of written expression to be considered for live writing. While a keypress is an instant gesture that adds a letter to the editor and triggers a sound, the accumulation of letters forms words, sentences, and eventually an artistic expression in a poetic form. The sonification - composed specifically in response to the real-time progression of the poem - dominates the ambiance of the piece and further conveys the meaning of the composition to the audience. The mapping between the poem and the composition hence goes beyond a simple one-to-one mapping of the gesture-based sound (keystrokes) previously discussed. Rather, the music is generated based on the content and is close to a sonification piece of which data is the writing. Such a connection in mapping is similar to that of code and the sound outcome in live coding; it may be a subtle idea for an audience to understand how input gestures (content) cause the music. Using writing gestures in composition is explored in [23] and the work shares the idea of live text being used at multiple levels (from one-to-one character mapping to subtle word recognition).

Just like written text can be pre-composed (not improvised), the sonification of writing can be pre-composed so that it is triggered to play when the composer wants. The various structures of a poem that indicate the progress, such as letter counts, lines, stanazas, and pages, are useful tools to trigger (or gradually change) the sonification so that the performer still has control over the progression of the music structure (as opposed to linearly played tape music in the background from the beginning). In addition, such writing structure progress can be used to schedule a change in the mapping strategies of aforementioned input gesture (typing sound, keystrokes) for a composed music structure. In this case, the sonification algorithm can be composed/programmed offline in advance or as a pre-composed snippet of an audio sample to be played as the piece proceeds with additional inputs from typing.

In contrast, a musician can develop an online algorithm that sonifies interactively based what is available to the editor on the fly. The algorithm must still be tightly connected to the central idea of the composition and should not be treated as a global sonification algorithm that can turn any piece of writing into a piece of music. The algorithm can take various mapping strategies to analyze at the syntactic, lexical, and/or semantic level. The algorithm can be as simple as a detection of a set of reserved keywords, triggering certain musical events whenever such words (or letters) are typed. On the other hand, it can be an intelligent algorithm that generate music based on machine learning. For example, the authors in [12] attempted to develop a stochastic algorithm that took into account various features of Chinese poetry at an acoustic level. Typically, existing techniques of natural language processing (e.g., sentiment analysis) can be grafted onto the algorithmic composition. Lastly, the algorithm need not depend on the linguistic features of a poem. Rather, the poem can be seen as time-series data that vary over time. In the sonification piece *Code That Sings Itself*, the algorithm analyzes the style of programming to sonify how elegantly the code was written [3]. Another metric that we developed in our composition was the pace of typing which represents certain writing patterns like of burst, pauses or corrective steps and reflects the state of writer's state of mind.

4.2.4 Mouse, Cursor and Viewport

As another important user interactivity in writing is the mouse control (or something equivalent e.g., trackpad), it is natural to include mouse as a secondary control to complement the keyboard. Similar to one-to-one mapping in keystroke, a musician can map mouse interaction directly to control parameters in music. Doing so gives the musician a separate musical instrument that they can play with. Such a direct mapping approach is weak, however, because it is far from the live writing activity.

The mouse can play an important role in live writing with the metaphor of a cursor in the writing environment. Typically, controlling a mouse is done through the cursor and the viewport control (by scrolling wheels). Therefore it is more transparent for the audience if the mouse control in live writing is used directly as a cursor. The cursor determines the location of where the next change occurs and it influences the audience's line of sight. In addition, highlighting certain text draws the audience's attention to a specific part of the poem. A performer can utilize the mouse to set the focal point in the poem. Similarly, the viewport of the editor determines the content that is visible to the audience by zooming in/out and scrolling.The visibility controlled with the mouse is useful for limiting the input data of sonification (discussed in 4.2.3). For example, if a musician zooms in to magnify or to highlight a specific word in a poem, the sonification algorithm can take the only word on screen instead of taking the whole poem as a sonification input, making the sonified result sound more responsive, interactive, and transparent.

4.3 Temporal Typography on Live Writing

The progress of writing should be revealed to the audience by sharing with them the writer's screen. By default then, the visualization of live writing is of the poem projected on the screen. Similar to visualizations of live coding performances, the visualization of a poem in live writing engages the audience in the performance in synchronization with typing and sound.

One of the visualization approaches that we suggest in this work is temporal typography [14]. We presented this visualization technique that turns plain text into a highly interactive and semantically meaningful medium. This is particularly effective in the context of live writing since the

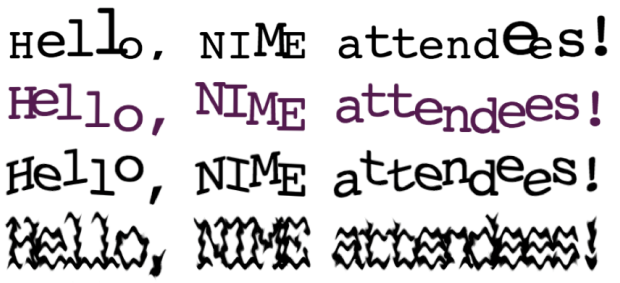

Figure 2: Various Examples of Temporal Typography. From Top to Bottom 1) mic input to font size, 2) keystroke density to rotating speed, 3) audio signal mapped to distort text, 4) sine wave convoluted text.

visual artifact is the text itself that needs to be read by the spectators. It allows developers to take any real-time input (from sensors or audio signals) and change the visual properties (shape, position) of the text rendered. (See Figure 2 for example.) For example, the algorithm can take input from the microphone where the typing sound is captured and mapped to the letter shapes (*sensor -> audio -> visual*). Alternatively, the temporal typography can be written to take input from the mouse control, which changes the viewport of the editor, and directly influence the sonification algorithm by scoping the input parameters (*sensor -> visual -> audio*). Lastly, the built-in microphone can be useful for visualization purposes. It should be noted that the mic captures both typing sounds and the music being played in the concert hall. This compound audio input actually makes the visuals highly responsive and interactive since the temporal typography responds to both a performer's play and the music. For the technical details and suggested examples, see [14] and the following link for the interactive demo of temporal typography.

`http://sangwonlee.com/temporal-typography/`

5. LIVE WRITING PERFORMANCES

The authors have carried out three live writing pieces as public performances (See Figure 1). This paper discusses the first two compositions. The first piece is *Live Writing: Gloomy Streets*, and it premiered at the University of Michigan's Performing Arts and Technology Showcase 2015. The writing environment and music was composed and performed by one of the authors (Lee) and the poem was co-written by Pain -a multi-instrumental musician, composer, and lyricist. The performance videos are available online at:

- Performance : `https://youtu.be/Ng1YSxiIXq0`
- Screen recording : `https://youtu.be/boDGeiBfasI`

Live Writing: Gloomy Streets was built upon the text editor realized on the web browser, utilizing Web Audio API [22], WAAX [5] WebGL [17] and Three.js [4]. The typing environment itself was part of the composition which contains visualization techniques, mapping strategies, and structured sonification. All these techniques were carefully designed to evolve as the poem was written. The poem (and music) was composed in three different parts and demonstrated the variety of strategies mentioned in 4.2. The piece begins with a simple mapping, pure amplification of the typing sound, with no expressive visualization of the text nor any other sound artifacts. In this part, the audience clearly understands that the performer is writing a poem synchronized with the text that appears on screen with the typing sound amplified. The second and third stanza introduce new visualization techniques that make text dynamically respond to the typing sound and gradually introduce synthesized voice that are scheduled to progress by the line breaks.

On the second page, the keystroke was mapped to a sample file and each key was mapped to a different playback speed, eventually changing the pitch of the sound. At the same time the writing on this page was written in Korean using the same English keyboard (not in Korean writing system). While the content of the poem on the second page mattered to the writer, the actual outcome was a random sequence of letters, that were neither English nor Korean. While un-readable for the audience, the execution of the second page created an interesting rhythm and meter of typing the poem. This was due to the language-specific characteristic of the language discussed earlier and the carefully written poem that was composed of a set of sentences that varied slightly from sentence to sentence.The intention behind this page was to convey the idea that typing a poem can be arranged in a musical way and by listening to it, the audience would notice that the performer was not typing arbitrary text. Non-interpretable text blocks the audience's linguistic perception of the text and helps their ears open to live writing music expressed via keystrokes.

The mouse cursor controlled the viewport of the writing environment (zoom in/out, perspective angle change) implemented in 3D space. At the same time, the changing of the perspective was directly mapped to play a synthesized voice. Page breaks were triggered by typing a command key which was a cue to control the structure. In order to gradually shape the piece as the content evolved, the program linearly interpolated various parameters that changed mapping and processed the sound. The interpolation modules were triggered at certain positions of the cursor given the poem was fixed (not improvised) so the poem's structural elements such as line breaks, stanza boundaries, page breaks, were used to trigger the musical cues. Started with static letters typed on a clean slate, eventually, words became physically real, and they shook and moved along with the music. Towards the end of the piece, the algorithm gradually distorted keystroke sounds and visuals, and slowly deconstructed the poem with a composed soundscape playing in the background.

As it is a preliminary stage of live writing, the current mapping from text/typing gestures to music may seem arbitrary. However, it is not yet clear for us that live writing practice can concretize a mapping for general writing gesture in music, especially given its multimodal - textual, visual and musical - outputs (the expressivity of text is highly complex and indefinite). While further discussion is necessary, we consider that the mapping in this piece was part of our exploratory process of composition [16].

The second live writing performance was *Live Writing: Reflections*. The piece was presented at University of Michigan's annual concert of Mobile Phone Ensemble. Everything was exactly the same as the first performance except that the piece (or poem) was written/performed by another author (Martinez). The performance video is available at :

- Performance : `https://youtu.be/1WRn2LNV9yw`

In *Live Writing: Gloomy Streets*, the visualization and the sonification were composed in consideration of an existing poem. In *Live Writing: Reflections*, the performer wrote a poem given the existing structure of the music, the software and the visuals. In addition, in the first piece, the performer wrote an existing poem as notated at the performance, while the performer of *Live Writing: Reflections*,

who had an intended piece prior to the concert, adopted a malleable approach of structured improvisation. Of her experience, the artist had the following to say:

> *There is something fantastically clumsy in the way that I compose music, words are crossed out, some words are streaming consciousness, and words that I want to take back after I have written them. (In live writing) All of the words come into this beautiful world on the screen, whether I want them to or not, and leave me vulnerable. It is this brutal honesty that comes out of utilizing live writing, that I love.*
>
> *(...) I tried to get the audience to read what was in my head, and what I was actually experiencing at the moment. My composition was based on this idea of seeing inside of my head, and trying to blur the line between what I was writing and what they may be thinking.*
>
> *(...) As a performer, I have to also respond to what is happening on the screen, if I have typed something wrong, or the wrong word has come out, I have to respond and improvise to try to get back to what I indented to say, or I need to trust the moment and allow a stream of consciousness to happen. Either way, I have to let the piece turn itself into what it wants to become. This makes each performance unique, it makes each performance feel alive, and it makes working with live writing a challenge as well as an amazing, invigorating instrument.*

Live writing differs from live coding music in that the composition and development of the software are not performed live. Here then, we see that the performer clearly understood the live nature of artistic expression of the piece and embraced it for musical expression and audience communication.

6. CONCLUSION

In this paper, we have introduced live writing, a new performance practice to transform the asynchronous written communication into real-time performing art. We reviewed the meaning of live writing independently and in comparison to live coding. The paper addressed the interaction scheme in a writing environment to develop the strategies of gesture-sound mapping in live writing. The paper suggested temporal typography as a visualization technique for live writing. Lastly, we presented the two live writing performances that took different approaches in the composition of the poem.

7. ACKNOWLEDGEMENTS

We gratefully acknowledge the helpful reviews by anonymous reviewers.

8. REFERENCES

[1] L. Anderson. The typewriter. Composition, 1950.

[2] A. Blackwell and N. Collins. The programming language as a musical instrument. *Proceedings of PPIG05 (Psychology of Programming Interest Group)*, 2005.

[3] S. Brueckner. Code that sings itself. `http://www.sophiabrueckner.com/singing.html`, 2011.

[4] R. Cabello. Three. js. `https://github.com/mrdoob/three.js`, 2010.

[5] H. Choi and J. Berger. Waax: Web audio api extension. In *Proceedings of New Interfaces for Musical Expression*, pages 499–502, 2013.

[6] N. Collins, A. McLean, J. Rohrhuber, and A. Ward. Live coding in laptop performance. *Organised Sound*, 8(03):321–330, 2003.

[7] P. H. Dietz, B. Eidelson, J. Westhues, and S. Bathiche. A practical pressure sensitive computer keyboard. In *Proceedings of the 22nd annual ACM symposium on User interface software and technology*, pages 55–58. ACM, 2009.

[8] G. Essl, M. Rohs, and S. Kratz. Use the force (or something)-pressure and pressure-like input for mobile music performance. In *Proceedings of the International Conference on New Interfaces for Musical Expression (NIME)*, 2010.

[9] R. Fiebrink, G. Wang, and P. R. Cook. Don't forget the laptop: using native input capabilities for expressive musical control. In *In Proceedings of the 7th international conference on New interfaces for musical expression*, pages 164–167. ACM, 2007.

[10] N.-W. Gong, M. Laibowitz, and J. A. Paradiso. Musicgrip : A writing instrument for music control. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 74–77, Pittsburgh, PA, United States, 2009.

[11] D. Griffiths. Fluxus. In A. Blackwell, A. McLean, J. Noble, and J. Rohrhuber, editors, *Collaboration and learning through live coding, Report from Dagstuhl Seminar 13382*, pages 149–150. 2013.

[12] C.-F. Huang, H.-P. Lu, and J. Ren. Algorithmic approach to sonification of classical chinese poetry. *Multimedia Tools and Applications*, 61(2):489–518, 2012.

[13] S. W. Lee and G. Essl. Live writing: Asynchronous playback of live coding and writing. In *Proceedings of International Conference on Live Coding*, Leeds, United Kingdom, 2015.

[14] S. W. Lee and G. Essl. Web-based temporal typography for musical expression and performance. In E. Berdahl and J. Allison, editors, *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 65–69, Baton Rouge, Louisiana, USA, May 31 – June 3 2015. Louisiana State University.

[15] S. W. Lee and J. Freeman. Real-time music notation in mixed laptop–acoustic ensembles. *Computer Music Journal*, 37(4):24–36, 2013.

[16] T. Magnusson. Designing constraints: Composing and performing with digital musical systems. *Computer Music Journal*, 34(4):62–73, 2010.

[17] C. Marrin. Webgl specification. *Khronos WebGL Working Group*, 2011.

[18] A. McLean, D. Griffiths, N. Collins, and G. Wiggins. Visualisation of live code. *Proceedings of Electronic Visualisation and the Arts*, 2010.

[19] J. Novak. *Live poetry: an integrated approach to poetry in performance*, volume 153. Rodopi, 2011.

[20] S. Reich. Typing music from the cave(opera). Composition, 1995.

[21] C. Roberts, G. Wakefield, and M. Wright. The web browser as synthesizer and interface. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 313–318, 2013.

[22] C. Rogers. Web audio api. 2012.

[23] S. Waite. Reimagining the computer keyboard as a musical interface. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, 2015.