Crossole: A Gestural Interface for Composition, Improvisation and Performance using Kinect

Sertan Şentürk
Music Technology Group,
Universitat Pompeu Fabra
Roc Boronat, 138
Barcelona, Spain 08018
sertan.senturk@upf.edu

Sang Won Lee
Georgia Tech Center for
Music Technology
840 McMillan St.
Atlanta, GA 30332-0456
sangwonlee717@gmail.com

Avinash Sastry
Khush Inc.
358 Angier Ave NE
Atlanta, GA 30312
avinash@khu.sh

Anosh Daruwalla
Georgia Tech Center for
Music Technology
840 McMillan St.
Atlanta, GA 30332-0456
adaruwalla3@gatech.edu

Gil Weinberg
Georgia Tech Center for
Music Technology
840 McMillan St.
Atlanta, GA 30332-0456
gilw@gatech.edu

ABSTRACT

Meaning *crossword* of sound, *Crossole* is a musical meta-instrument where the music is visualized as a set of virtual blocks that resemble a *crossword* puzzle. In *Crossole*, the chord progressions are visually presented as a set of virtual blocks. With the aid of the Kinect sensing technology, a performer controls music by manipulating the *crossword* blocks using hand movements. The performer can build chords in the high level, traverse over the blocks, step into the low level to control the chord arpeggiations note by note, loop a chord progression or map gestures to various processing algorithms to enhance the timbral scenery.

Keywords

Kinect, meta-instrument, chord progression, body gesture

1. INTRODUCTION

Crossole is a meta-instrument that uses body gestures to make music in different depths and modes. It is directly influenced by the meta-instrument concept, which implements one-to-many mapping between a musician's gestures and the sound so that a musician may perform music in a high level instead of playing note by note.

One of the earliest examples of a meta-instrument is the Radio Baton [5], designed and developed by Max Mathews. The system consists of two batons and a base unit termed *the antenna*. The capacitance between each baton and the antenna is measured continuously to estimate the position of the tips. The position data for each baton is used to drive the parameters controlling an oscillator.

The relationship between music and movement is one that has been explored in extensive detail, particularly in forms of contemporary dance. Weschler states that the advantage of using computer systems in dance comes mainly from their ability to bridge different forms of interaction and media [10]. The EyesWeb [1] is a particularly interesting system in this regard, designed to track the body movements and use this information to generate music and visuals based upon the affect or the emotional content in the gestures.

Michael Waisvisz's *Hands* [9] was one of the first meta-instruments to use hand gestures, along with a small keypads under the fingers, to control the sounds produced by synthesizers. In *BioMuse*, Tanaka created a biosignal musical interface based on gestures using sensors to detect arm gestures [8]. A close relative to *Crossole* is the Air Worm [2], which comprised a digital theremin to produce MIDI or audio output. Users were allowed to control the tempo and loudness of the performance by varying the position of their hands. Finally, representation of the musical structure with either virtual or tangible blocks has been previously represented demonstrated in several devices such as Sony's Block Jam [6] and the Tenori-On [7].

Crossole also draws from the well-established tradition of hacking and circuit-bending among instrument builders, going back to the works of Reed Ghazala [3]. At the time of its development, *Crossole* was one of the earliest applications for musical performance and composition using the Kinect, and relied completely on reverse-engineered open source drivers and frameworks [4].

2. CONCEPTUAL DESIGN

Crossole consists of three modes of operation, which are visualized in three different views: **1.** the "crossword view", where the chord structures are set and played (Section 2.1); **2.** the "*grid* view", where the arpeggiations of the chords are arranged (Section 2.2); **3.** the "*timbral* view", where the user is able to make timbral manipulations on top of a recorded "chord path" (Section 2.3). The interaction between the user and *Crossole* is intended in a way that the interface can be manipulated simply by intuitive body gestures and visual feedback given by the system (Section 3).

2.1 Crossword View

The visual screen of *Crossole* employs a novel representation of musical structure: a chord is symbolized by a colored square (or a "Block") and a piece of music can be displayed by a set of blocks, which resembles a crossword puzzle (Figure 1). To construct a chord, the player instantiates a block. Next, the player associates a base note (e.g. C, D, E) and a chord type (e.g. major, minor, diminished) to the block. Finally the chord block must be placed adjacent to an ex-

NIME'12, May 21 – 23, 2012, University of Michigan, Ann Arbor.

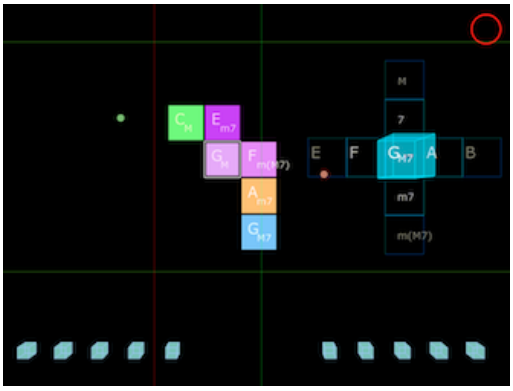

Figure 1: The *crossword* view. The chord blocks are clustered in the center of the screen. A new block is created in the right, and the cursor (white unfilled square) is located on the top *Gmajor* block in the middle. The green and red dots indicate the locations of the left and the right hands respectively.

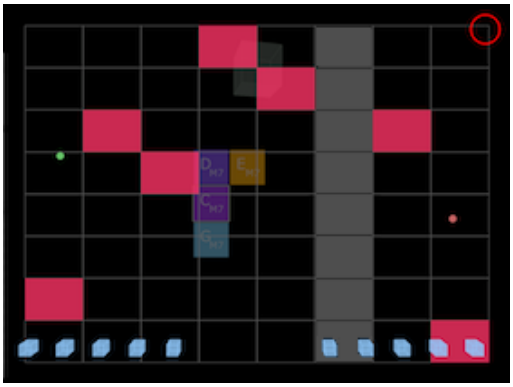

Figure 2: The *grid* view. The metronome is seen in the sixth column. The activated squares show the notes to be arpeggiated. The *crossword* view is drawn in the background showing the current chord being played.

isting block (except the first one, which can be virtually placed to anywhere on the screen).

Another important element of controlling chord progressions is the cursor, which is depicted as a white empty square. While building block sets up a base for the music structure, the cursor control determines the temporal progression of chords. The cursor is able to move once at a time in all directions (vertical, horizontal, diagonal). A player can play a chord on-the-fly by placing the cursor upon a block representing the specified chord. By controlling the timing and the direction of the gestures an unique chord sequence is generated. Nonetheless, the cursor has to be traversed precisely to achieve a smooth progression.

The two dimensional depiction of blocks encourages users to understand the musical structure and find efficient ways of constructing chord clusters. For example, if a sequence of four chords is to be repeated multiple times, it is much reasonable to place 4 blocks in a circular manner than to place dozens of blocks from left to right. With meticulous planning, the repetitions, variations and musical sections (such as A-A-B-B, verse, chorus, bridge) can be interpreted in an original, coherent and efficient structure.

2.2 Grid View

In *Crossole*, there is a separate screen to determine the sequence of the notes to arpeggiate for each chord, called the *grid* view (Figure 2). The *grid* view is a classical sequencer type interface, where the x-axis is the quantized time, the y-axis corresponds to pitches related to the chord and the current state of the sequencer is indicated by a visual metronome moving from left to right. The *grid* view enables users to draw a pitch contour by marking the cells on the *grid*. The number of notes in the *grid* is interdependent from the current block selected (by the cursor) such that there are only eight keys (four pitch classes spanned in two octaves). For example, if the cursor is on a *C major7* block, the notes in the cells in the *y*-axis will be {*C4, E4, G4, B4, C5, E5, G5, C6*}; whereas for *C major*, they will be {*C4, D4, E4, C5, C5, D5, E5, C6*}.

The *crossword* view and the *grid* view are interconnected as two different control levels. The *crossword* view follows the concept of meta-instrument where the player performs a musical instrument in a high level (chord progression). On the other hand, the *grid* view lets the *Crossole* player access the low level, where music is played note-by-note. However, it should be noted that, unlike a conventional instrument, the *grid* view is limited by the location of the cursor and by the selected chord. When the player moves the cursor and changes the chord in the *crossword* view (or "high level view"), the note sequence specified in the low level view will maintain its pitch contour, but the notes will be transposed to the new chord. This repetitive transposition in the pitch contour is very common in popular music and Western classical music such as J.S. Bach's Prelude in C Major (BWV 846). *Crossole* works effectively in this particular musical style since it takes out the necessity to focus on playing the same contour for each chord.

2.3 Timbral View

In musical pieces with a fair number of chord repetitions, *Crossole* opens another opportunity by allowing the user to record the route of the cursor and the timestamps. Whenever the cursor returns to the first block the recording has started, a closed chord progression loop is formed. From this point on, the cursor movement is automated to follow the exact recorded temporal progression, until the player chooses to discontinue the repetition.

As soon as the playback starts, the hand gestures are freed from controlling the chord progression, and a new sound control interface named *timbral* view is also introduced (Figure 3). In the *timbral* view, the player can create new soundscapes by applying various sound effects (e.g. filters, phasors, delays) using the free-form hand gestures. In the meantime, the player is still able to switch to the *grid* view and change the pitch contour.

3. GESTURES AND VISUAL FEEDBACK

The gestural controls of *Crossole* is composed of four major parts: placing chord blocks, controlling the cursor, drawing a pitch contour and manipulating the timbre. The *Crossole* player has to manage the first three types of gestures in concurrent manner. Finally the *timbral* control is employed when a recorded path is being re-traversed. The gestures are designed to apply these operations while fulfilling ease of use and clarity of interaction.

We use the position of both hands as the primary controlling variables. A red and a green dot are drawn for the left and right hands respectively to indicate the projection of the hand positions. Moreover, two horizontal and two vertical lines are drawn in the *crossword* view to indicate the thresholds to pass so that the cursor is moved in the desired

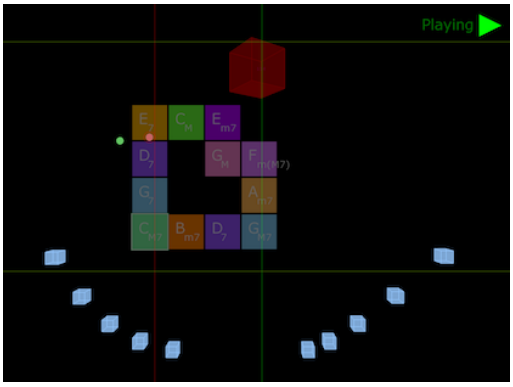

Figure 3: The *timbral* view. The icon in the top right indicates a loop is being played in the background and the trajectory of the small cubes in the bottom hints the locations of the hands.

direction. Such visual cues are integral to easily execute the actions controlled by the hand gestures.

In the *crossword* view, the familiar concepts of drag-and-drop and swiping are used for block creation. Similar to the drag-and-drop gestures of a mouse, the depth position of the right hand is treated as the left button of mouse. A block is instantiated by dragging the *meta-block* (rotating 3D cube on the top of the screen) to an arbitrary location. The base note and the chord type are selected by swiping through two perpendicular menus and placing the desired base note and chord type onto the the cube (Figure 1). To enhance the visual feedback, the cube is also rotated in the same direction with the swiped menu. Once the player chooses the chord, the block can be dragged and dropped next to the existing chord block structure. The system automatically places the block to the closest position possible provided drop point is neighbored by a block. Finally, the base notes and the chord types are printed on the top, and each block is color coded according to the base note and the chord type.

For the cursor movement, the swiping gesture is utilized again. To allow the player to build a block and move the cursor simultaneously, the right hand and the left hand positions are assigned to block-creation and cursor movement respectively. Swiping the cursor in the latitudinal and longitudinal dimensions concurrently is also possible, which enables a diagonal swiping gesture as well.

In the *grid* view, the right hand is used to draw the pitch contour. When the circle representing the right hand enters to a cell from left, the note associated to that cell is activated in the corresponding time slot of the sequencer, and any active note in the same column is removed. Similarly, entering to an "active" cell from right deactivates it. Nonetheless, entering to a cell either from the bottom or the top does not change its state. In this view, the projection of the right hand is mapped slightly beyond the borders the *grid* view, thus the hands can be positioned without any unintentional state changes.

During the performance, the *Crossole* user has to frequently switch back-and-forth between the *crossword* view and the *grid* view. The distance between the user and the camera is used to switch between these two views so that no other additional controller is needed. When the player stands at a distance from the sensor, *Crossole* is in the *crossword* view so that the player can control blocks and move the cursor. If the player gets closer to the camera than a certain threshold, the system will display the *grid* view. Therefore the player can easily switch the level of control by "stepping into" the low level control (*grid* view) and "stepping out" to the high level control (*crossword* view).

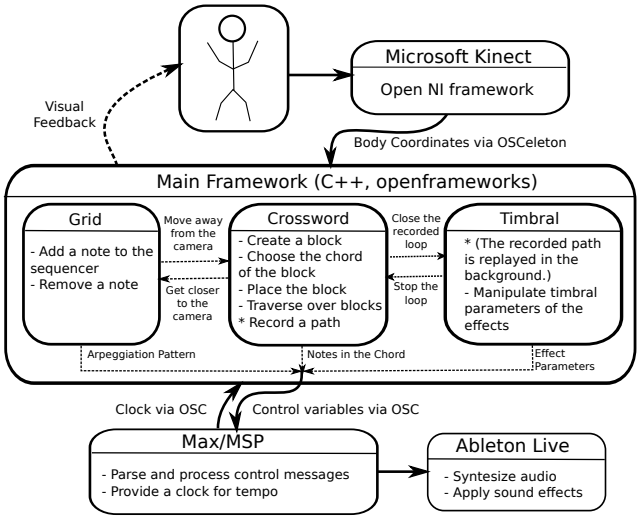

Figure 4: The block diagram of *Crossole*.

To start recording a loop, the user simply has to "press" the record button on the top right of the *crossword* view. In the *timbral* view, the x, y and z coordinates of both hands are mapped such that a six-dimensional control mechanism can be applied to control the parameters of the sound effects. In this mode, the ten small cubes at the bottom of the screen visualizes trajectories of hand movements to help the user and the audience to track the interaction between gestures and the instrument (Figure 2).

The *grid* view and and *timbral* view are always drawn on top of a dimmed *crossword* view so that the *Crossole* user, the other musicians, and the audience in the performance can effortlessly track the current block played and anticipate the upcoming chord progression.

4. IMPLEMENTATION

Our system utilizes Microsoft Kinect, a commercially available and affordable 3D-sensing device, to track the body gestures. Kinect makes it possible to sense the full-body pose (including the depth) without the use of markers or extra devices. At the time of the development, there were no official development kits[1]. Nonetheless, a number of developer communities were able to hack the input/output data and distributed non-commercial libraries. We used OpenNI [2] and OSCeleton[3] to produce OSC messages of pose data. This message includes three-dimensional coordinates of the seventeen points of a body, such as hands, elbows, knees.

These OSC messages are received by the main framework, which is coded in C++. The main framework maps the three-dimensional coordinates into control variables and visualizes the current state of *Crossole*. In each mode of operation, the coordinates and gestures are mapped into different control variables such as the notes to play, the sequence to play the notes and timbral parameters. The chords and the notes forming them are kept in a text file (or a "chordbook"), which is read in the start of the software. The user has the freedom to edit the chordbook, define customized chords and associate any MIDI note to them. For visualization, openframeworks[4] is used. Visualization is mostly

[1]Recently, the Microsoft Kinect SDK has been published: `http://www.microsoft.com/en-us/kinectforwindows/`.
[2]`http://www.openni.org/`
[3]`https://github.com/Sensebloom/OSCeleton`
[4]`http://www.openframeworks.cc/`

based on OSC messages from OSCeleton which includes the 3D coordinates of the two hands and torso, and provides the user an intuitive and efficient means of visual feedback.

Next, the processed control variables are concatenated into an OSC message and sent to Max/MSP. The Max/MSP patch parses the message and processes the variables into musical actions such as the note sequences in the arpeggios or the effect parameters. It also provides a reliable clock for the tempo of the music and for the sequencer visualization in the *grid* view. Finally, the parsed data is sent to Ableton Live, and the musical actions are synthesized.

5. PERFORMANCE

Crossole was performed in Listening Machines 2011, the annual concert series hosted by Georgia Tech Center for Music Technology in 30 April 2011, at the Stubbins Gallery in Georgia Institute of Technology, Atlanta, Georgia[5]. One of the authors (Lee) composed a piece for the concert and played the virtual instrument. Another author (Daruwalla) improvised with an electric guitar on top the chord progressions. Before the performance, the chordbook was edited to hold only the chords used during the performance.

The piece was composed of five sections. The first part (0:00 - 3:15) is to demonstrate the basic concepts; switching between the *crossword* view and the *grid* view, block creation, moving the cursor and drawing the pitch contour of the arpeggios. The second section (3:15 - 5:25) is a solo for *Crossole*, where the performer forms the backbone of chord progression and moves the cursor simultaneously. At the third section (5:25 - 7:15), the *Crossole* player makes slight variations in the same chord progression, while the cursor route is being recorded. This part focuses on the guitar solo with the accompaniment of *Crossole*, which exposes the audience to a contrast and harmony between a conventional and a electronic meta- instrument. In the fourth section (7:15 - 9:05), the cursor is automated and maintains the exact same chord route of the preceding section. This section exploits the free-form gestures to produce continuous sound effects with a variety of timbres on the top of recorded chord progression. At the last section (9:05 - 11:25), both performeners engage in collaborative improvisation.

The performance was successful and received an overall positive response from audience. The interface, gestural control and corresponding visual feedback helped the audience to understand the instrument. Informal conversations with audience members indicated that they were easily able to understand the concepts of the instrument such as drawing analogies between the swiping gestures of our system and the visual interface in the *Minorty Report*[6], or explaining the *grid* view as "stepping into a sequencer."

Crossole is one of the first applications using the Kinect for musical performance and composition. The performance video drew a significant attention from the Kinect community and was shared by the major Kinect hack websites [7].

6. DISCUSSION AND CONCLUSION

The high-level capabilities of the system greatly reduces the number of gestures compared to conventional instruments, which typically exploit one-to-one mapping between the notes and the gestures. The intuitive and easy-to-use interface may also attract novice users. Moreover, the user and the audience only needs a basic note and chord knowledge to understand the musical structure and arpeggiations.

Another strength of *Crossole* comes from its flexibility in the mappings between the note symbols and sounds. The user can take advantage of the control variables given by the system by applying any arbitrary sound effect and controlling them by gestures with countless possible mappings. Another aspect that have not been fully experimented is the mappings in the chordbook. Chordbook is simply utilized as an abstraction between some symbols and associated numbers, and the user is allowed to define personalized patterns that can be mapped to any MIDI number. Even so, *Crossole* is designed with repetitive loop-based music in mind, and is better deemed as an accompanying instrument.

Nevertheless, *Crossole* incorporates some engaging factors faced during playing a conventional musical instrument, such as the real-time pressure and the gestural control. The system still requires considerable amount of practice to coordinate the block placements and make a "tight" traversal timing. Additionally, the system provides a space for finesse due to the interdependent switchings between the pitch contour and the cursor-block control. Under the light of these observations, we believe that the interface achieves a relatively low-floor and a "mid"-ceiling [11] within its limited musicality. We believe that such virtual instruments based on gesture control will be particularly effective to engage audiences and develop musical interests in novices.

7. ACKNOWLEDGMENTS

We like to thank the MSMT students at Georgia Tech Center for Music Technology (GTCMT) for their valuable feedback during the development of *Crossole*.

8. REFERENCES

[1] A. Camurri, S. Hashimoto, M. Ricchetti, A. Ricci, K. Suzuki, R. Trocca, and G. Volpe. Eyesweb: Toward gesture and affect recognition in interactive dance and music systems. *Computer Music Journal*, 24(1):57–69, 2000.

[2] S. Dixon, W. Goebl, and G. Widmer. The "air worm": An interface for real-time manipulation of expressive music performance. *Proc. ICMC'05*, pages 614–617, 2005.

[3] R. Ghazala. *Circuit-Bending: Build your own alien instruments*. Wiley Pub., 2005.

[4] J. Giles. Inside the race to hack the Kinect. *The New Scientist*, 208(2789):22–23, 2010.

[5] M. Mathews. The radio baton and conductor program, or: Pitch, the most important and least expressive part of music. *Computer Music Journal*, 15(4):37–46, 1991.

[6] H. Newton-Dunn, H. Nakano, and J. Gibson. Block jam: a tangible interface for interactive music. *Journal of New Music Research*, 32(4):383–393, 2003.

[7] Y. Nishibori and T. Iwai. Tenori-on. In *Proceedings of the 2006 Conference on New Interfaces for Musical Expression*, pages 172–175. IRCAM-Centre Pompidou, 2006.

[8] A. Tanaka. Musical performance practice on sensor-based instruments. *Trends in Gestural Control of Music*, pages 389–405, 2000.

[9] M. Waisvisz. The hands, a set of remote midi-controllers. In *Proceedings of the International Computer Music Conference, San Francisco, CA: International Computer Music Association*, pages 86–89, 1985.

[10] R. Wechsler. O body swayed to music (and vice versa): roles for the computer in dance. *Leonardo*, pages 385–389, 1997.

[11] D. Wessel and M. Wright. Problems and prospects for intimate musical control of computers. *Computer Music Journal*, 26(3):11–22, 2002.

[5]The performance video is available online at `http://www.sangwonlee.com/works/crossole/`. The time stamps in the video are provided throughout this section to support the narrative.

[6]`http://youtu.be/NwVBzx0LMNQ`

[7]e.g. `http://tinyurl.com/br7rwxz`