Windows that Won’t

for solo piano and laptop ensemble

2012

Konrad Kaczmarek

Window's That Won't – Performance Notes

This piece is for solo improvising piano and laptop ensemble and is structured in three sections. The soloist should follow the instructions in the 'conductor' software on how to navigate the three sections, and members of the ensemble receive performance instructions in real-time through the 'ensemble' software interface. The lead sheet provided with this score contains suggestions for the types of harmonic and melodic material to explore in each section, but the soloist should feel comfortable improvising beyond these guidelines.

Section I

Piano – play sparsely, using the three note melodic fragments notated in the lead sheet. Become more active as ensemble builds to chord texture. Listen to the chords that emerge from the ensemble and improvise harmonically with resulting harmonies. Use the scales on the lead sheet as a guide.

Ensemble – [**Buffer Player – shift 1**] start out gesture by selecting most recent index/note (**shift-i**) and trigger note manually using the letter keys(**a,b,c…x,y,z**) and the hammer trigger. Listen to the texture, and make smaller crescendos after each note. On cue, start building texture, and then eventually toggle on a number key [**8**] for automatic triggering. Move attack value away from 0, either by manually moving the slider up, or pressing (**shift-a**). Once you reach a sustained sound, play around with moving the attack value back to 0 for changes in timbre, and moving through previous index/notes (**left, right arrow**). The cue for the end the this section tells you to stay on the last index/note, and then switch to [**Sample Player – shift 2**] with the mouse at the top of the screen.

Section II

Piano – play melodically, either triggering counterpoint in the ensemble by manually sending notes (**Ft 11-19**) or toggling on the pitch tracker (**FT-20**). Use attack to trigger a random note from the pitch class, or an instruction.

Ensemble – With Attack still non 0, mouse-map on, and pitch interpolation set to 1500ms, follow the instructions to go to various pitches. Instruction include 'move slowly to $1 – then stay there', 'go directly to $1', 'move down' etc. End of the section cue says to slowly decrease the pulse tempo, by stepping down through the number keys until you get to [**1**], and then set the attack value to 0.

Section III

Piano – improvise with the ensemble in a rhythmic way– listen to the melodic fragments that emerge from the hocketing in the ensemble, and trade off with the soloists. Play around with ostinato patterns.

Ensemble – Your resting state for this section is with [1] toggled on. When the entire ensemble is synced together, let it sound a few times, and then re-sync (**~-key**) yourself out of the group pulse. Watch carefully for the solo section, and when you are soloing, move to a faster rhythmic pulse (**[4]-spacebar**) to play a short/gestural solo.

Windows that Won't

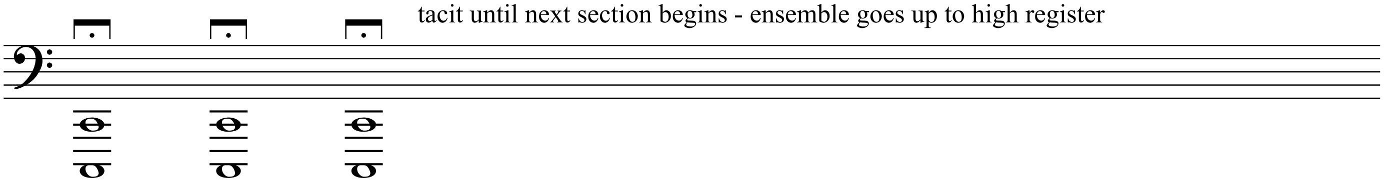
example lead sheet

konrad kaczmarek

Section I

ad lib on written scale for ≈ 45" before preceding to next 3-note phrase

tacit until next section begins - ensemble goes up to high register

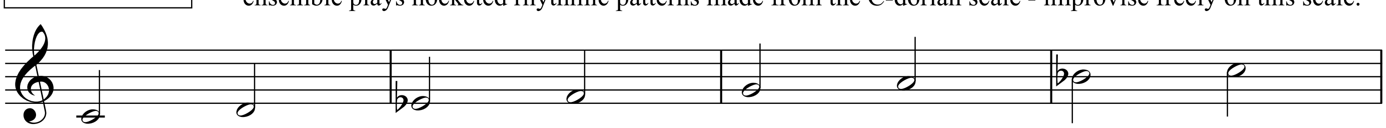


Section II

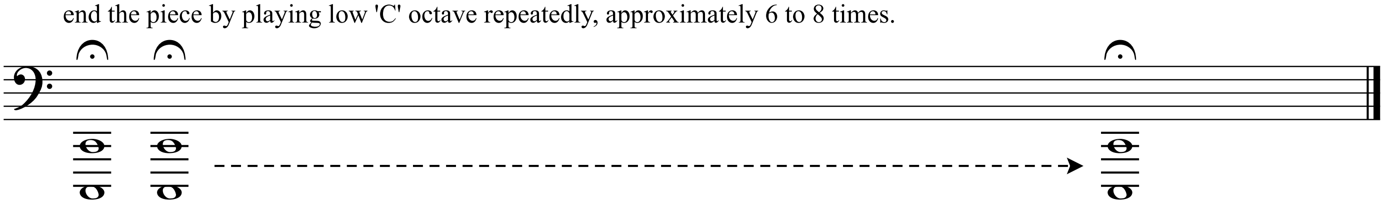
ensemble plays simple, slow-moving melodies based on either of these scales.

Section III

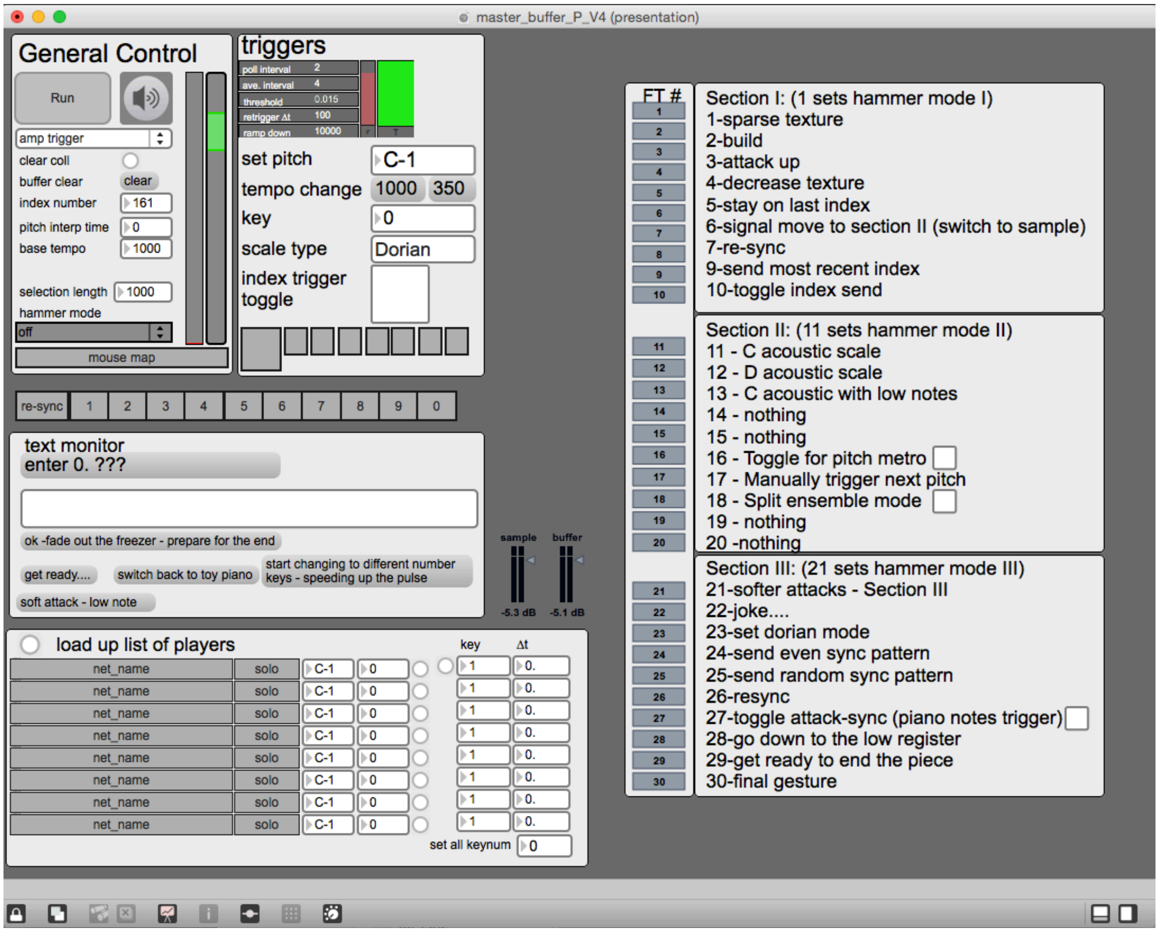
ensemble plays hocketed rhythmic patterns made from the C-dorian scale - improvise freely on this scale.



end the piece by playing low 'C' octave repeatedly, approximately 6 to 8 times.

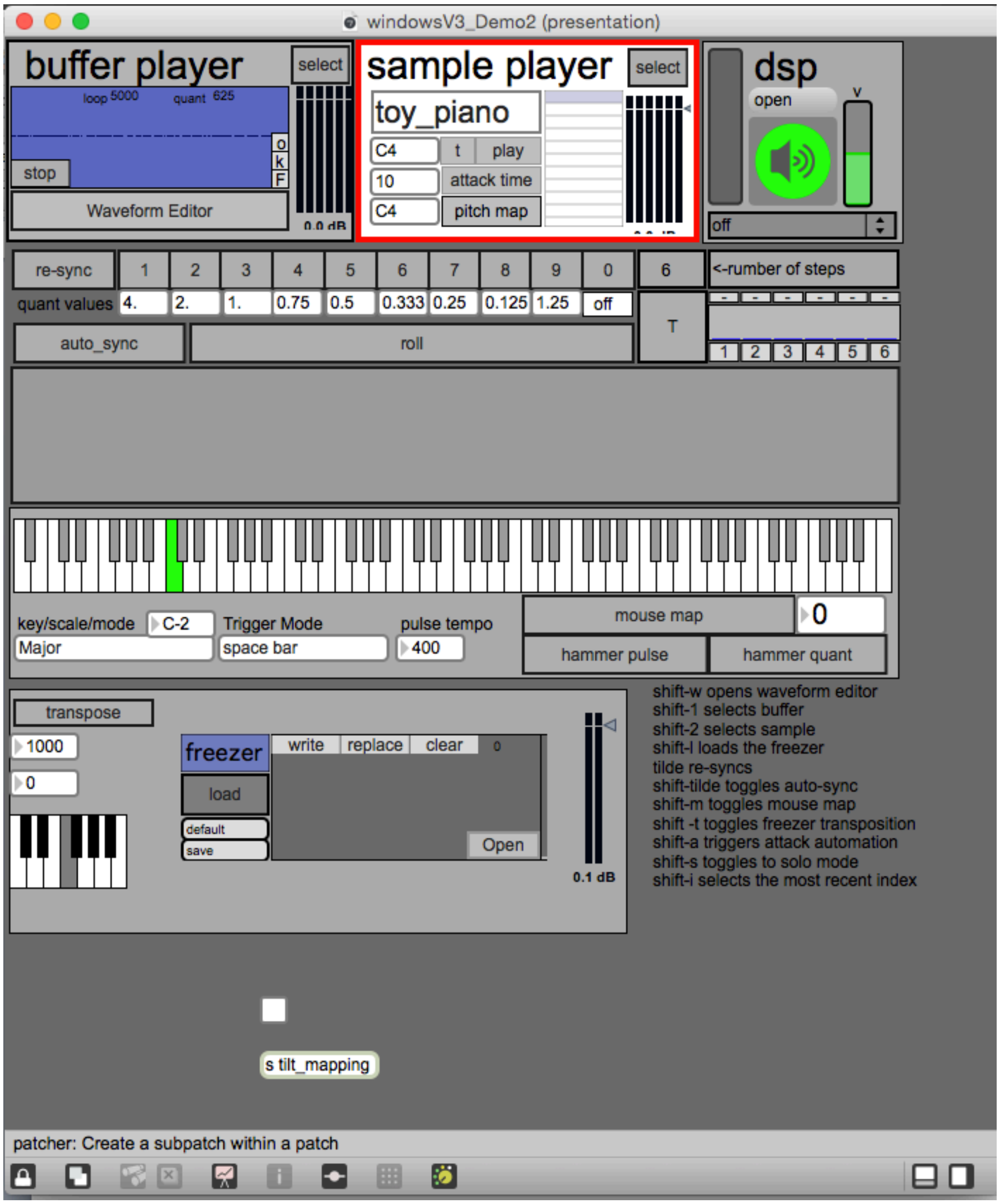


Conductor Software



The soloist uses a MIDI foot controller to trigger the program changes and ensemble instructions displayed in the list at the right of the interface. The 'Triggers' section contains settings for the envelope follower, and the "Text Monitor" section allows the soloist to send ad hoc messages. The "General Control" section contains audio status settings and other settings for the ensemble software. For more information on the conductor software, see the attached paper, "Windows That Won't: Exploring collective real-time granular textures with a laptop ensemble."

Ensemble Software



The ensemble software contains the various synthesis modules as well as an interface that provides performance instructions from the conductor software over the wireless network. Performers control the software using modified piano hammers as a physical triggering device and key-strokes on the laptop keyboard. For more information on the conductor software, see the attached paper, “Windows That Won’t: Exploring collective real-time granular textures with a laptop ensemble.”

Windows That Won't:

Exploring collective real-time granular textures with a laptop ensemble

Konrad Kaczmarek
Princeton University
kkaczmar@princeton.edu

ABSTRACT

Windows the Won't is a work for laptop ensemble and solo piano composed for the Princeton Laptop Orchestra in 2012, and subsequently adapted for the ensemble Sideband.

This paper documents key technical and compositional approaches used to generate the collective real-time granular textures of the piece. The organic and physical contribution of each member of the laptop ensemble is highlighted through the adoption of a simple and intuitive input device and the use of wireless network communication. Specifically, aspects of mapping both user input and network control data to sound file granulation are investigated over the three sections of the piece.

Keywords:
laptop ensemble, wireless network, sound file granulation

1. BACKGROUND AND STRUCTURE

Windows that Won't is scored for solo piano and eight performers, each equipped with a laptop running custom software, an audio interface, physical input for control, and 6-channel hemisphere speaker array with subwoofer[1]. The soloist performs a structured improvisation in three sections, which correspond to changes in the types of sounds and textures created by the ensemble, as well as changes in how they interact with the software using the input device. During the first section of the piece, the pianist plays sparse melodic fragments while the ensemble alternates between discrete hocketed textures and dense granular clouds generated from audio sampled from the soloist in real-time. In the second section the ensemble switches to a toy piano sampled granular instrument to generate a sustained and slowly moving counterpoint to the more active piano solo. Finally, the third section involves quantized rhythmic patterns that move in and out of phase with each other across the ensemble as the soloist engages in a more exaggerated rhythmic style of playing.

Motivation for the piece came from a desire to incorporate the sounds and models of interaction that I had been developing in several live electronics pieces for solo instruments[2] into a piece for multi-player laptop ensemble. First, this process involved creating a program that would distribute the audio processing among the laptop performers in a musically meaningful way, taking advantage of the unique spatial-acoustic characteristics of the ensemble and allowing it to function collectively as an instrument[3] (see figure 1). Secondly, it required devising a way to collaboratively control the audio processing, creating a cohesive sound that would also highlight the unique contribution of each member of the ensemble. Finally, I wanted to use a physical input device that could provide a dramatic visual component to the performance and provide the audience with a clear indication of how each individual player contributes to the sound as a whole.

The piece also required devising a means of conducting the ensemble that could adapt to the flexible and improvisatory nature of the solo part. Building on previous work developing a piece for laptop ensemble called *alskdjalskdjalskdj*[4], I adopted a means of communicating both text-based performance instructions and direct control data from a single conductor machine to each member of the ensemble over the wireless network. Subsequent revisions of the piece for the laptop ensemble Sideband incorporated the networking tool LANdini for more accurate timing over the wireless network[5]. This structure facilitated tightly coordinated automation within the ensemble, allowing the piece to explore the boundaries between hyper-mechanical and a more natural and diffused mode of playing.

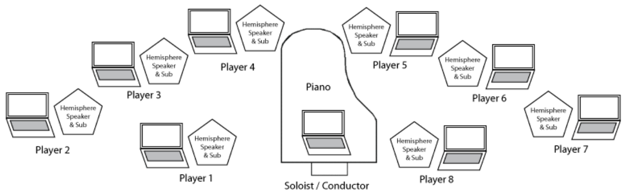


Figure 1: Performers surrounding the soloist on stage provide an immersive sonic environment capable of 48 discrete channels of audio output.

2. SOFTWARE

2.1 The Conductor Program

The soloist runs a conductor program, created with Max, that sends messages to the ensemble over a wireless network while also providing a visual road-map of the structure of the piece (see figure 2). Messages sent to the ensemble, which are triggered by a MIDI foot pedal, include traditional performance instructions in the form of text that appears on each player's screen, as well as direct control messages used to update or reconfigure aspects of the ensemble's software. Examples of this type of direct control data include changes in pulse tempo, volume, scale type, granular density and duration, as well as overall section changes and mapping modes. Messages can be sent to individual players, a defined sub-set of the group, or the entire ensemble. The conductor patch also performs an analysis of the soloist's audio signal, and sends the resulting pitch tracking, volume envelope, and attack detection data out to the ensemble. Finally, in addition to the wireless control data, audio is routed from the conductor patch to each member of the ensemble for real-time processing using a multi-channel audio interface and an 8-channel XLR snake.

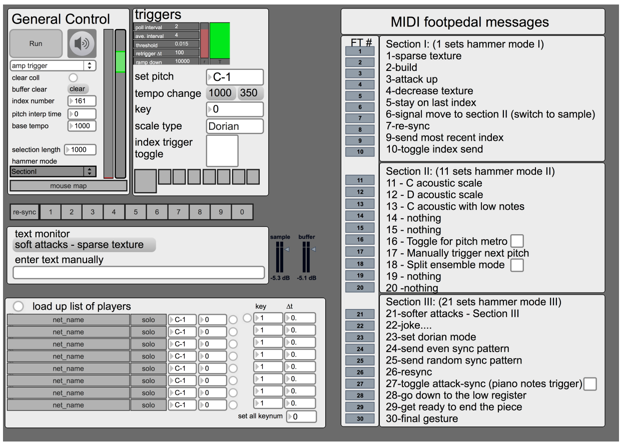


Figure 2: Conductor Program

2.2 The Ensemble Program

Each member of the ensemble performs the piece using a standard PLOrk configuration that includes laptop running custom software, an audio interface, physical input for control, and a 6-channel hemisphere speaker array [2]. The software is a patch written in Max that applies various granulation techniques to two types of audio data[6][7][8]. Buffer-based playback reads from a single, performance-length buffer that contains audio of the soloist that is recorded in real-time, while sample-based playback reads from banks of pre-recorded samples. The two granular synthesis modules also differ in the approach to extracting grains from the source material, one relying on an event-based input while the other utilizing a feature-based input[9]. The grain scheduler and general grain structure are the same in the two devices, and are determined by direct user input, preset configurations, and control data that is distributed to the software in real-time.

The buffer-based player extracts and sequences grains according to three input parameters: attack index, attack offset, and random deviation. As the soloist plays, the conductor program continually updates an index of attack times, which correspond to individual notes or phrases recorded in the buffer. By incrementing or decrementing the attack index value in the waveform viewer window, the user can navigate the buffer in a sequential, temporally non-linear way (see figure 3). Similarly, a separate trigger causes the program to jump to the most recent detected attack, corresponding to the last note played on the piano. The attack-offset parameter shifts the grains away from the attack of the note to more sustained areas of the sound. The random deviation parameter allows animated clusters of grains to be generated around the given static index point by sequencing grains over a range that can include both the note attack and sustain zones, as well as surrounding notes [7]. The synthesis engine is capable of playing up to 30 voices simultaneously, and grains are scheduled manually using an input trigger or automated using a variable pulsetrain signal.

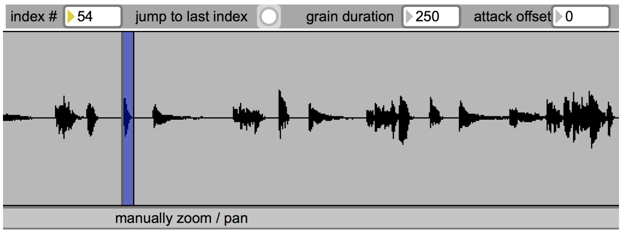


Figure 3: Waveform viewer with index, grain duration and attack offset parameters

The sample-based player uses a 30-voice playback engine to granulate banks of pre-recorded samples. Based on the concept of data-driven concatenative sound synthesis [10], the synthesis module determines which sample to extract a grain based on a given bank number and a floating-point note value (60.0 = middle C). Individual grain rate of playback is adjusted according to the input note value and a predetermined pitch mapping. Each sample bank can have its own mapping, which is saved with the patch. The sample player uses a variable three-stage linear envelope, and ramp-up and ramp-down times can be set as absolute times or as percentages of the grain duration. As with the buffer-based player, individual grains of the sample player can be manually triggered or automated using the grain scheduler. Additional synthesis parameters common to both buffer- and sample-based modules include grain size, rate of playback, grain volume (including a random volume distribution over a given range), window shape, and multi-channel output mapping.

Figure 4: Piano action hammer with piezo element

Players control both granulation algorithms using hammers from a piano action fitted with piezo sensors. The piezo element is attached to the hammer head molding, the wooden part of the hammer underneath the felt head (see Figure 4), and is routed directly to one of the inputs of the audio interface. Players are instructed to strike the metal frame of the hemisphere speaker, generating data used for both discrete and velocity-scaled triggering (see Figure 5). Attack detection and velocity sensitivity are calculated in the software, and the patch includes an interface that allows the performer to either manually set the soft to hard attack threshold or enter them in by example using a learn mode. The felt head of the hammer provides the performer with a natural and tactile response when striking the rigid metal frame, allowing them to focus more intuitively on the sounds they are generating. The felt also helps to mask any acoustic sound that the hammer would otherwise make. While the hammer was initially chosen for the symbolic significance that it provided the piece, it ultimately proved to be an excellent input device in terms of physical handing, durability, and dynamic range. The specific way in which this simple physical input is mapped to different synthesis parameters changes depending on the section of the piece or a specific sub-section mapping mode (see Section III).

Figure 5: Hammer input device with hemisphere speaker

3. MAPPING USER INPUT TO SOUND FILE GRANULATION

3.1 Section I

In the opening section of the piece, each member of the ensemble contributes a single stream of grains to an overall granular texture according to two mapping modes. The first mode relies predominantly on performer physical input to trigger grains, generating a sparse and asynchronous granular texture[7]. The second mode automates grains into tightly coordinated streams, in a texture that approaches collaborative quasi-synchronous sound file granulation[7], with density and phase uniquely determined by each performer.

In the first mapping mode, the intensity of the hammer attack determines the grain size, density, phase, and sample offset. The range in the hammer velocity maps linearly to a relatively large grain size (500ms to 1000ms) and a sparse grain density (0.25 to 1 grain per second). During this section, individual grains have a three-stage linear fixed envelope with a steep attack time (20ms), which ensures that the envelope does not mask the note's attack. Each hammer strike moves the buffer-based player to the most recent indexed attack point in the audio buffer, corresponding to the last note that the soloist played. The hammer strike also manually re-syncs the granular pulse, generating a discrete note on impact. Individual player's granular density is relatively low, resulting in inter-onset values of 1 to 4 seconds in a diffused and pointillistic texture within the ensemble as a whole. The collective granular density thus varies from 8 to 2 grains per second with an un-coordinated phase distribution. Performers are instructed to navigate the texture intuitively in response to the soloist's part, generating a constantly evolving and shifting hocketed rhythm.

After receiving a cue from the soloist, each member of the ensemble strikes the hammer with a single hard attack, initiating the second mapping mode. This mode generates a repeating stream of grains with varying degree of overlap, resulting in a continuous and sustained sound. Grains are fixed at 100ms long with a Gaussian envelope, granular density varies from 12.5 to 20 grains per second according to hammer intensity, and a variable attack-offset value is applied to the attack index point. This offset shifts the grains away from the attack of each detected note towards the sustained part of the sound. As the patch steps through the six output channels with every new grain, the second mode's continuous granular stream results in a dense and phase-rich acoustic sound that is harmonically derived from the piano solo part. Subsequent hammer strikes move the attack index to one of the three most recent values, and temporarily zero the attack-offset value, centering the grains directly over the note attack to create a short burst of timbral variation. In addition to a sustained granular texture, this section of the piece generates chords within the ensemble, as performers harmonically navigate the recent history of notes performed by the soloist. Again on cue, a single hard attack resets the program back to the first mapping mode.

3.2 Section II

In the second section of the piece the performers generate granular glissandi using prerecorded toy piano samples. The sample-based player generates grains that last up to 500ms and use a three-stage linear envelope with a fixed ramp-up time to mask the sample's attack. Throughout this section a pulsetrain signal automatically triggers the sample player, generating grains at a rate of 20 to 25 per second, resulting in a smooth and continuous texture. The glissandi are coordinated by the conductor patch, which sets target pitch and ramp time parameters, and then starts a visual countdown that cues each player when to strike their hammer, initiating the glissando. While the target note value ramps smoothly from one value to the next, the sample granulator triggers discrete and overlapping grains as it travels through the glissando. Subtle variations in timing and granular density, which are mapped to the hammer intensity in each player, create an interesting cumulative effect in the ensemble as a whole. Each hammer attack also momentarily shortens the grain envelope ramp-up time, revealing more of the sample's attack to provide a timbral variation during the first part of the glissando. As the section continues, the ensemble breaks into sub-groups, each with their own sequence of pitches and coordinated cues, generating a 2- and then 3- part counterpoint to the improvising soloist.

3.3 Section III

Finally, the third section uses granular pulses that are coordinated across the entire ensemble, capable of producing precisely quantized rhythmic patterns. The section begins with an abrupt reduction in granular density from 25 to approximately 4 grains per second, creating a sense of deceleration in the texture as well as perceptual shift from continuous pitch to discrete pulse or rhythm. Although the performers trigger the same bank of toy piano samples, the decreased granular density coupled with a steep three-stage linear envelope creates a pointallistic texture reminiscent of the opening section. Hammer intensity is mapped to both grain density and pitch. As opposed to the free mapping in the opening section, both parameters are now quantized, limiting pitches to a set scale and granular densities to integer multiples of a given base rate. More specifically, soft hammer attacks produce low pedal notes that repeat every 4 seconds (whole note), while the hardest attack produces notes in the range of 72.0 to 102.0 that repeat every .125 seconds (32nd note). The performers can expressively navigate the quantized steps between both extremes, producing melodic fragments with varying pitch and rhythmic value.

The conductor patch is able to override each performer's hammer-derived parameters of rate, phase, and pitch, creating tightly coordinated ensemble-wide granular textures. To achieve this, the conductor patch sends out a single synchronized message that contains information in a list format for the entire ensemble. The software running on each player's laptop receives the message and parses out the data intended for that specific machine (each laptop broadcasts its name to the conductor patch). The message also contains a player-specific control delay time, which allows phase-accurate timing within the ensemble. For example, setting all of the granular density to the whole note, and delay times to integer multiples of an 8th note pulse, the conductor patch can generate an even 8th note pattern that moves around the ensemble. More complex rhythmic patterns can be created as each performer can be set to any quantized grain density and arbitrary delay time. Instructions such as "Let evenly quantized state persist for several seconds, and then strike your hammer to interrupt the rhythm" are given to the ensemble. Finally, the conductor patch's attack detection can function as a trigger for each individual player. The piece ends with chord clusters generated in the ensemble that are synchronized to the soloist's note attacks.

4. CONCLUSIONS

Sound file granulation is a data-intensive process, which often necessitates the automation of multiple synthesis parameters in order to create interesting and continually evolving sonic textures[7]. This piece strove to replace the automation of control data with the physical input of a performing ensemble through the use a simple physical input as a velocity-sensitive trigger and a flexible means of data mapping. Similarly, each individual performer's multi-channel output coupled with the orientation of the ensemble around the soloist contributed to a complex and constantly shifting acoustic image that provided an additional level of engagement for the improvising soloist.

Figure 6: Sideband rehearsing Windows that Won't

Each section of the piece utilized different audio source material for sound file granulation and used different mapping strategies for the physical input, providing a contrasting accompaniment for the soloist and an evolving model of interaction within the ensemble. By altering the time scales of various granular parameters, such as individual grain size and density, the piece was able to encompass a continuum of synthesis techniques ranging from micro- and macro-montage through to more traditional granular manipulations of texture and timbre. This continuum also corresponded to perceptual shifts in attending to rhythm and pulse through to sonic surface and density, and from melody and chord through to the subtle timbral detail of the individual note on the micro-time scale.

Future revisions to the piece will include adding wireless streaming audio, as well as the ability to incorporate recordings of previous performances into the program. This would give an ensemble the ability to perform the piece in a meaningful way without the soloist, whose part would become reanimated by the ensemble. Retaining multiple solo performances could also provide the piece with a history of not only the recorded audio, but of the various types of audio analysis, which could be utilized in a from of adaptive concatenative sound synthesis turning a single performance into the instrument itself[11]. Finally, I would like to incorporate a distributed means of conducting the piece, which would allow members of the ensemble to influence the structure of the piece in a more direct way. This might include performance instructions or global control data sent out to the ensemble directly by individual performers.

5. REFERENCES

[1] Smallwood, S., Trueman, D., Wang, G., Cook, P. "Composing for Laptop Orchestra," Computer Music Journal Spring 2008, Vol. 32, No. 1: 9–25.

[2] http://konradkaczmarek.com/material/meta_gest_2012.pdf

[3] Smallwood, S., Cook, P., Trueman, D., McIntyre, L. "Don't Forget the Loudspeaker: A History of Hemispherical Speakers at Princeton, Plus a DIY Guide", Proceedings of the New Interfaces for Music Expression Conference, Pittsburgh, 2009.

[4] Kaczmarek, Konrad. Alskdjalskdjalskjd: documenting the process involved in writing a timing-accurate networked piece for a laptop ensemble. Proceedings of the Symposium on Laptop Ensembles and Orchestras, Louisiana State University, 2012.

[5] Narveson, Jascha, LANdini: a networking utility for wireless LAN-based laptop ensembles, Sound And Music Computing Conference, Stockholm, 2013.

[6] Wolek, Nathan. *A Granular Toolkit for Cycling74's Max/MSP*. SEAMUS, University of Iowa, 2002.

[7] Roads, C. Microsound. Cambridge: MIT Press, 2001.

[8] Opie, T. Creation of a real-time granular synthesis instrument for live performance. Master's thesis, QUT, Brisbane, Australia, 2003.

[9] Bencina, Ross. "Implementing Real-Time Granular Synthesis," in Greenbaum & Barzel (eds.), Audio Anecdotes III, 2006.

[10] Schwartz, Diemo. *Data Driven Concatenative Sound Synthesis.* PhD Dissertation, IRCAM, University of Paris, 2004.

[11] Sturm, Bob. Adaptive Concatenative Sound Synthesis and its Application to Micromontage Composition. Computer Music Journal Winter 2006, Vol 30, No. 4: 46-66.