SenSynth: a Mobile Application for Dynamic Sensor to Sound Mapping

Ryan McGee
Nokia Research Center
and
Media Arts and Technology
UCSB
ryan@mat.ucsb.edu

Daniel Ashbrook
Nokia Research Center
daniel.ashbrook@nokia.com

Sean White
Nokia Research Center
sean.white@nokia.com

ABSTRACT
SenSynth is an open-source mobile application that allows for arbitrary, dynamic mapping between several sensors and sound synthesis parameters. In addition to synthesis techniques commonly found on mobile devices, SenSynth includes a scanned synthesis source for the audification of sensor data. Using SenSynth, we present a novel instrument based on the audification of accelerometer data and introduce a new means of mobile synthesis control via a wearable magnetic ring. SenSynth also employs a global pitch quantizer so one may adjust the level of virtuosity required to play any instruments created via mapping.

Keywords
mobile music, sonification, audification, mobile sensors

1. BACKGROUND
In 2002 Hunt stated that an electronic instrument was more than just an interface coupled with a sound generator, and referred to the mapping between interface and sound as the "essence" of an instrument [6]. With today's mobile devices we see a plethora of sensors available for interaction coupled with powerful processors that can rival the sound synthesis capabilities of the laptop. Yet, we often only see instruments exploiting either the sensor or synthesis potential of these devices. For example, applications like Cosmovox[1] and AirVox[2] make innovative use of the camera and sensors on a phone, but are rather limited in synthesis abilities and do not allow one to reconfigure the sensor-to-sound mapping. Users of these applications are not free to explore the "essence" of other mappings. While commercial mobile applications such as Synthstation[3] and Nanostudio[4] bring powerful synthesis to mobile phones, they still rely on interaction via touch screen or external MIDI adapter and fail to make use of the many available mobile sensors on modern phones such as the accelerometer, gyroscope, and magnetometer.

[1]http://leisuresonic.com/cosmovox/
[2]http://www.yonac.com/AirVox/
[3]http://www.akaipro.com/synthstation
[4]http://www.blipinteractive.co.uk/

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

Also in 2002, Wessel and Wright suggested a NIME design criteria with "initial ease of use coupled with a long term potential for virtuosity. . . and clear and simple strategies for programming the relationship between gesture and musical result" [13]. Many existing mobile instruments such as the Ocarina[5] take the "hyperinstrument" approach to accomplishing initial ease of use and potential for virtuosity by modeling traditional instruments. Overholt and Roads suggested an alternate approach of designing around modern synthesis techniques rather than creating instruments with familiar interfaces – "the mimicking of these older designs does not take into account the predilection for the electronic composer to work inside the sounds." [10] Likewise, our emphasis with SenSynth was to allow the user to easily create sensor-to-sound mappings rather than to mimic interactions with traditional instruments. Overholt and Roads state that new instruments will require a "new type of virtuosity" from an "interdisciplinary composer/programmer" [10]. We must, however, still consider criteria for initial ease of use when designing new interactions, which in our case results in a simple mapping interface, the inclusion of preset instruments, and an optional pitch quantizer to ease the playing of notes in a musical scale using various sensors.

Another category of applications includes RjDj[6], Inception[7], and ZooZBeat[8]. These augmented music applications do bring an innovative use of onboard sensors as interfaces to interact with songs or preset patches, but overall do not stand as instruments themselves. Pre-smartphone systems such as Sonic City [5] and PuDAC [7] synthesized and composed musical experiences using a number of wearable sensors as a user walked through a city or engaged in exercise. Both projects viewed themselves as augmented musical experiences rather than instruments for musical performance. We believe that the ideal interface should be able to stand as both a performance instrument and augmented music application. A key feature to realize this versatility, which all of the aforementioned applications lack, is the ability to quickly and dynamically create custom sensor to sound mappings. Without this ability to remap, performers are constrained to a fixed form of interaction and the new musical interface becomes no more versatile than a traditional musical instrument. Likewise, listeners of augmented musical experiences should be able to create their own interactions instead of downloading other's presets.

The Speedial [3], urMus [4], and Momu [2] projects are probably the most similar to SenSynth in terms of ideology. Momu is a programming toolkit to aid in the rapid mapping between mobile sensors and audio. Speedial and

[5]http://ocarina.smule.com/
[6]http://rjdj.me/
[7]http://inceptiontheapp.com/
[8]http://www.zoozmobile.com/zoozbeat.htm

urMus are mobile applications that allow for rapid sensor to sound mapping in a manner very similar to our own. However, SenSynth provides a GUI front end to all instrument parameters so that sliders and buttons may be used to control parameters that are not mapped to any sensor. urMus allows the scripting of GUI's that may be uploaded to the phone, but our approach augments a variety of provided synthesis interfaces with customizable sensor control. We believe this approach to be easier to use for the electronic music layman. This paper will also focus on novel interactions enabled with SenSynth.

Our goal was to make a mobile interface with an emphasis on arbitrary mapping between an array of sensor inputs and several sound synthesis techniques. We did not set out to model any familiar modes of interaction nor did we attempt to define any modes of interaction for existing synthesis techniques. While included preset modes of interaction and a global pitch quantizer provide instant usability, those desiring more control are free to design their own instruments or abandon the automatic tuning. SenSynth redefines itself as an instrument through user-created mappings and compositions created by the user's motions.

2. SENSYNTH INTERFACE

SenSynth allows the user to arbitrarily map a mobile phone's sensor output values to parameters for sound synthesis and effects. For example, the accelerometer might be connected to control the modulation depth of a FM synthesizer, so that when when the user shakes the phone, more partials appear in the FM sound. SenSynth is a mobile platform for sonic experimentation and the results of much mappings may be used equally for musical performance and sonification. The ease of use of the GUI enables rapid experimentation with both the mappings and the instrument itself. The key to SenSynth is that every onscreen button or slider can be controlled via any sensor on the phone (Figure 3).

2.1 Implementation

SenSynth currently runs on a Nokia N9/950 under MeeGo. Programming was accomplished in C++ using the QT Quick package which includes the QT Mobility API for mobile sensor communication and audio callbacks as well as QML for scripting dynamic graphical user interfaces. The sound synthesis library used is custom but greatly inspired by the CREATE Signal Library (CSL)[9].

2.2 GUI

SenSynth initially presents the user with a simple mixer page to activate and adjust the gain of each available sound source. Each source has its own page in which all parameters can be adjusted via touch sliders and buttons. There is an FX page with on/off switches and the appropriate sliders for each audio effect.

The mapping page allows the user to dynamically map any sensor on the phone to control any source or effect parameter. The user is presented with a list of all available sensors and once a selection is made all available sources and effects are displayed. After making a source or effect selection, a list of parameters that can be controlled by the chosen sensor are displayed. While mappings are arbitrary for the most part, there are some limitations. For example, the proximity sensor can only output a boolean on/off, so it is not able to control the frequency of an oscillator. A single sensor can control any number of sound parameters for a one-to-many mapping. If a user tries to assign multiple sensors to a single parameter the last sensor will gain all control and the previous mappings to that parameter will be removed. Figure 1 illustrates the selection of a sensor and Figure 2 shows several completed mappings.

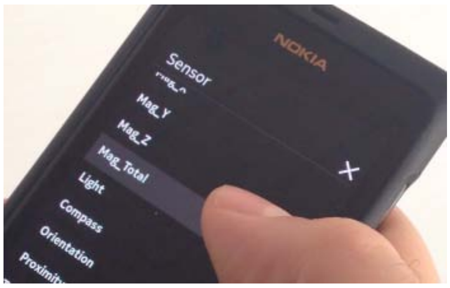


Figure 1: Selecting a Sensor

Figure 2: An Example Instrument

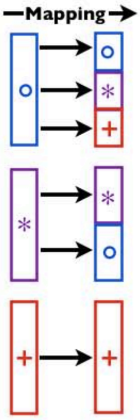
Once a mapping is created it is displayed in a list with two buttons: one to remove the mapping and one to invert the sensor control. For example, if the gyroscope is mapped to control the frequency of an oscillator, tilting the phone upwards (increasing the gyroscope's pitch) will cause an increase in the frequency of an oscillator by default, but applying the invert button will cause a decrease in the oscillator's frequency when the phone is tilted upwards. Series of several mappings can be named and saved as presets on the Load / Save page.

2.3 Sensors and Sound Parameters

Figure 3 shows all available sensors and sound parameters as well as their data types. The colored boxes indicate which sensor information can control which sound parameters. For example, it is possible for floating-point outputs to control integer and boolean parameters. So, if the gyroscope roll is mapped to control the waveshape of an oscillator the floating-point output will be quantized to 5 values, one for each of the 5 possible waveshapes. As the user twists the phone, the waveshapes will change as the values fall into the quantized ranges. For floating-point to boolean control, "true" is triggered whenever the value surpasses a fixed threshold and outputs "false" otherwise. For instance, one could record into the sampler by twisting the phone one way and stop recording by twisting the opposite direction.

[9] http://fastlabinc.com/CSL/

Sensor	Data
Gyroscope	• Roll°, Pitch°, Yaw°
Accelerometer	• X°, Y°, Z°, Total°
Magnetometer	• X°, Y°, Z°, Total°
GPS	• Distance°, Speed°
Compass	• Degrees from North°
Light	• Ambient light*
Orientation	• Top up, top down, left up, right up, face up, face down *
Proximity	• Something near the phone (i.e. phone in pocket) +
Tap	• X, Y, and Z direction +



Module	Parameters
Wavetable Synthesizer	• waveshape*, amp°, freq°, ADSR°, AM°, FM°, pan°
Scanning Synthesizer	• scanned data°, amp°, freq°, AM°, FM°, pan°
Granulator	• density°, average grain size°, grain bandwidth°
Sampler	• record start/stop+, looping+, reverse+, ADSR°, AM°, FM°, pan°
Karplus	• string pluck+, drum hit+, snare hit+, amp°, freq°, pan°
Delay	• on/off+, time°, feedback°
Reverb	• on/off+, decay°
Pitch Shifter	• on/off+, pitch ratio°
EQ	• on/off+, low, mid, and high gain°
Tuner	• on/off+, scale*

° = Continuous floating-point   * = Discrete integer   + = Boolean

Figure 3: Available Sensors, Sound Parameters, and Mapping Options in SenSynth

3. INTERACTIONS

Here we present two examples of interactions with SenSynth. The first is an instrument that derives its sound waveform from the audification of accelerometer data, and the second explores the simple concept of using a wearable magnetized ring to interact with sound via the magnetometer.

3.1 The Kinetic Synthesizer

In addition to the common oscillators found in other mobile sound synthesis applications, SenSynth includes 3 scanned synthesis sources for the direct sonification of sensor data. Scanned synthesis reads periodic motion-generated data at haptic rates (sub 15Hz) and literally scans the data as an audio waveform at a rate fast enough to enter our range of hearing. Thus, one may directly manipulate the spectrum of a sound by human movements [12]. Our application of scanned synthesis may also be classified as a form of audification– the act of reading arbitrary data as an audio waveform [8]. Using scanned synthesis with the accelerometer and gyroscope we have constructed a preset instrument called the "kinetic synthesizer."

All 3 scanned sources are mapped to read the accelerometer x, y, and z axis respectively. Dynamic amplitude variations within each of the three summed accelerometer waveforms results in vector synthesis, a technique in which multiple wavetables are rapidly crossfaded with each other [11]. We then map the gyroscope's roll, pitch, and yaw to the frequency (directly related to the scan rate) of each scanned source. We term the resulting combination of motion-produced scanned and vector synthesis the "kinetic synthesizer" (Figure 4). Creating this instrument can be accomplished in seconds using SenSynth's mapping interface.

The musical pitch of each source can be controlled by the gyroscope's 3 dimensions of tilt. With the pitch quantizer active one can easily construct harmonic chords and triads since all frequencies will be quantized to match the nearest note in the selected musical scale. Without the quantizer the gyroscope gives a subtle vibrato effect when trying to hold the phone in a certain plane and shaking it to produce the waveforms, and these effects can be stretched to FM-like sounds with the combination of rapid shaking and tilting of the phone. The experience is reminiscent of Hunt's accidental theremin discovery in which it was determined that necessitating movement to make sound results in a more interesting instrument [6]. Playing the instrument, users immediately become aware of the connection between their physical vibrations and the resulting sound. As O'Modhrain shows [9], such haptic feedback incorporated within a computer-based instrument can improve players' abilities to learn new behaviors.

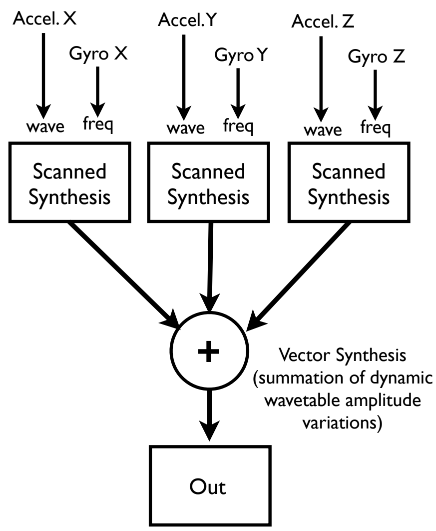


Figure 4: The Kinetic Synthesizer

3.2 Magnetic Ring Control

Using a simple neodymium magnet shaped as a wedding ring (and optionally painted with black epoxy as in our case) we are able to obtain a high degree of control (down to about 1mm of hand movement) when mapping the phone's magnetometer to various sound parameters. Such magnetic rings have also been successfully used for selection of audio and visual targets [1]. Of course, our initial test with the ring was to use our hand motion to control the frequency of an oscillator much like a theremin, but the most powerful results were obtained by using the ring to control expressive parameters such as amplitude modulation (tremolo), panning, granulator density, grain size, EQ levels, sampler playback, delay feedback, and reverberation amount. For example, moving a hand towards the phone could simultaneously increase the granulator density, reverberation amount, and cause a note to trigger from one of the oscillators. To produce a greater magnetic effect one may simply attach additional small neodymium magnets to the ring, increas-

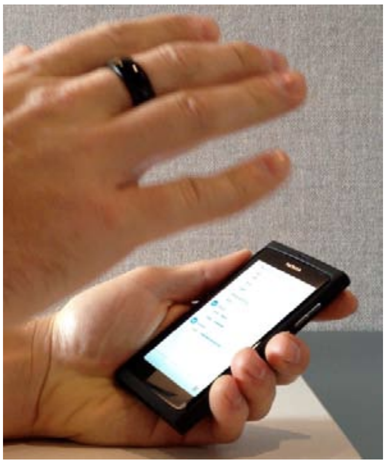


Figure 5: Interaction Using the Magnetic Ring

ing the magnetic field and allowing one to interact from a greater distance. The overall experience using the ring is similar to a theremin except that the user is free to select the axis of interaction and work with a much larger sound palette.

One interesting situation is when the user wants to interact with a button or slider on the touch screen with the ring-wearing hand which will also cause changes in any sound parameter controlled by the magnetometer. Initially this was seen as a downfall of the ring interaction, but it can serve to produce synchronous cues as to when the user/performer is interacting with the touch screen. For instance, let's say a performer wishes to record a new sample via the internal microphone, and in doing so he or she taps the record button on the screen with a finger from the ring hand. Let's also assume they have mapped the magnetometer to control the playback rate of the currently looped sample. The change in magnetic field as the finger moves closer to touch the screen would correspond to a change in the playback rate of the sampler preceding the new recording. In terms of composition, this could be used as a cue to indicate the entrance of a new sample in the song.

3.3 Pitch Quantizer

SenSynth includes a global pitch quantizer to ease the playing of musical notes in key. Without quantization, the sensitivity makes it extremely difficult to play precise musical notes and chords using the gyroscope, magnetometer, etcetera. Like an effect, the quantizer can be switched on or off via mapping to a sensor and the musical scale parameter of the quantizer can changed similarly. Source frequency values from sensors are simply quantized to match the nearest note in the selected scale. For example, if the A Major scale is selected and the gyroscope's roll position is outputting 442 Hz to control the frequency of an oscillator, then that value would become 440 Hz (A).

4. CONCLUSION

While many related mobile applications provide means for interactive music, or simply rely on established paradigms of musical interaction, SenSynth provides a less constrained approach to mobile synthesis interaction and musical performance. We have shown an instrument based on the audification of accelerometer data and introduced the idea of control via a wearable magnetic ring. New users should find the application instantly fun using the in-tune presets and hopefully become inspired to program new instruments with the simple on-screen mapping interface. Fans of more virtuosic instruments should be pleased with the several degrees of control possible and non-trivialness once the pitch quantizer is disabled. Thus, SenSynth provides a versatile mobile platform for sonic experimentation with larger range of user appeal and more expressiveness than comparable applications. The code and QT project are open-source and will be available online soon[10]. In the future we intend to support newer mobile platforms and consider more social experiences such as online sharing of instruments directly from the phone and the ability to control other phones running SenSynth on a network.

5. REFERENCES

[1] D. Ashbrook, P. Baudisch, and S. White. Nenya: Subtle and Eyes-Free Mobile Input with a Magnetically-Tracked Finger Ring. In *Annual Conference on Human Factors in Computing Systems*, pages 2043–2046, 2011.

[2] N. J. Bryan, J. Herrera, J. Oh, and G. Wang. MoMu : A Mobile Music Toolkit. In *New Interfaces for Musical Expression*, pages 174–177, 2010.

[3] G. Essl. SpeedDial : Rapid and On-The-Fly Mapping of Mobile Phone Instruments. In *New Interfaces for Musical Expression*, pages 270–273, 2009.

[4] G. Essl and A. Müller. Designing Mobile Musical Instruments and Environments with urMus. In *New Interfaces for Musical Expression*, pages 76–81, 2010.

[5] L. Gaye, R. Maze, and L. E. Holmquist. Sonic City : The Urban Environment as a Musical Interface. In *New Interfaces for Musical Expression*, pages 109–115, 2003.

[6] A. Hunt, M. M. Wanderley, S. S. West, and M. Paradis. The importance of parameter mapping in electronic instrument design. In *New Interfaces for Musical Expression*, pages 1–6, 2002.

[7] R. Jacobs, M. Feldmeier, and J. Paradiso. A Mobile Music Environment Using a PD Compiler and Wireless Sensors. In *New Interfaces for Musical Expression*, pages 193–196, 2008.

[8] G. Kramer. *Auditory Display: Sonification, Audification, and Auditory Interfaces.* Addison-Wesley, 1994.

[9] M. S. O'Modhrain. *Playing by Feel: Incorporating Haptic Feedback into Computer-Based Musical Instruments.* PhD thesis, Stanford University, 2000.

[10] D. Overholt, C. Roads, and J. Thompson. On Musical Gestures and New Performance Interfaces for Electronic Music. In *International Gesture Workshop*, April 2003.

[11] C. Roads. *The Computer Music Tutorial.* MIT Press, 1996.

[12] B. Verplank, M. Mathews, and R. Shaw. Scanned Synthesis. In *International Computer Music Conference*, 2000.

[13] D. Wessel, M. Wright, and J. Schott. Intimate Musical Control of Computers with a Variety of Controllers and Gesture Mapping Metaphors. In *New Interfaces for Musical Expression*, pages 192–194, 2002.

[10] https://github.com/ryanmichaelmcgee/SenSynth