SHAMUS – A SENSOR-BASED INTEGRATED MOBILE PHONE INSTRUMENT

Georg Essl

Deutsche Telekom Laboratories
TU Berlin
georg.essl@telekom.de

Michael Rohs

Deutsche Telekom Laboratories
TU Berlin
michael.rohs@telekom.de

ABSTRACT

ShaMus is a sensor-based approach to turning mobile devices into musical instruments. The idea is to have the mobile device be an independent instrument on its own right. The sensors used are accelerometers and magnetometers as can now be found in some commercial mobile devices. The sound generation is also embedded on the phone itself and allows individual and untethered gestural performance through striking, shaking and sweeping gestures.

1. INTRODUCTION

ShaMus is a sensor-based approach to turning mobile devices into musical instruments. The sensors that are employed predominantly relate to motion acquisition. Specifically we use accelerometers and magnetic field sensors. These sensors do now appear in select commercially available consumer devices and hence promise a potential wide availability.

The idea is to have the mobile device be an independent instrument on its own right. This requires that the sound generation also be embedded on the phone itself.

Turning mobile devices into musical instruments has been explored by a number of researchers. Tanaka presented an accelerometer based custom-made augmented PDA that could control streaming audio [14]. Geiger designed a touch-screen based interaction paradigm with integrated synthesis on the mobile device using a port of Pure Data (PD) for Linux-enabled portal devices like iPaqs [8, 7]. CaMus is a system that uses the camera of mobile camera phones for tracking visual references to allow performance [11]. Various GPS based interactions have also been proposed [13, 15]. This system used an external computer for sound generation. A review of the general community was recently presented by Gaye and co-workers [6].

There are various uses of accelerometers and other sensors in mobile interactions. For example Rekimoto introduced the notion of of tilt for scrolling interactions [10] and Williamson and co-workers investigated the use of accelerometers and vibrotactile display for eyes-free text-messaging interactions [16].

Goal of this work is to bring interactive mobile music making close to a commodity state while maintaining the expressiveness of free arm and wrist gestures. Commodity mobile devices now offer embedded sensor capabilities, which until recently were only available through external custom devices. This allows first steps towards disseminating mobile music environments without additional hardware. Moreover we show that the sensor quality of special purpose hardware still exceeds those of now built-in sensors justifying the use of special purpose sensing to pre-figure future higher-fidelity commodity options for sensing.

For this purpose we present two sensing scenarios: In the first we use the built-in 3-axis accelerometer of Nokia 5500 mobile phones. In the latter we use a special purpose external wireless sensor package called Shake [1] to illustrate more complex interactions.

Some of these extra sensors available in the Shake unit can also already be found in commodity devices. For example Nokia 5140 mobile phones contain a magnetometer to determine compass bearing.

2. SENSORS

The Nokia 5500 is a commodity phone with a built-in 3-axis accelerometer. The Shake device [9] is a small device designed to incorporate a range of high-fidelity sensors for rapid prototyping of mobile interactions. The core unit contains a 3-axis accelerometer, a 3-axis magnetometer, a vibration motor for vibro-tactile display, a navigation switch, and capacitive sensing abilities. All analog data is sampled to 12 bit resolution at 1 kHz. The dimensions of a Shake unit are $54 \times 40 \times 16$ mm. Hence the device is small enough to fit at the back of a cell phone without adding excessive bulk.

2.1. Accelerometer Quality

Initial tests with Nokia 5500 phones showed that the quality of the sensed data is too poor for the kinds of interactions we were interested in. The measured results for the vertical axis of the phone lying on a table without motion are: min 248, max 406, mean 339.5, stdev 23.9, and range

[1] www.samh-engineering.com

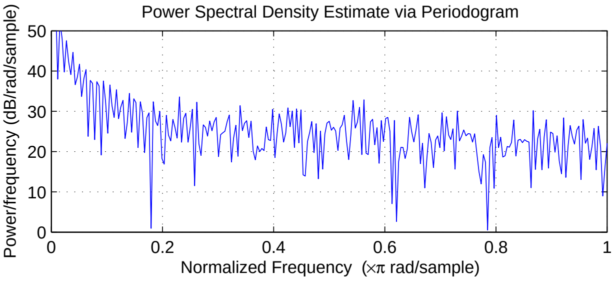

Figure 1. Spectrum of the sensor data along one axis of a Nokia 5500 accelerometer. The noise floor is flat away from DC up to sample frequency.

158 in device specific unity. We see an error of 7% relative to the static mean.

The noise covers the full sampled spectrum evenly as can be seen in Figure 1. By filtering and additional threshold locking, a sensibly stabilized signal can be achieved but at the cost of dynamic range corresponding to the noise floor present.

For comparison, the accelerometer data obtained from the Shake unit with digital filtering turned off on a tabletop without motion has much higher fidelity. Mean values for axis X, Y, and Z are 1003.0, 997.7, and 999.0 in units $10^{-3}g$ and a standard deviation of 0.000, 0.328, 0.090 which are errors relative to mean values of less than 0.1%. Hence we noted a factor of over 70 in error comparing the Shake accelerometers and the data from the Nokia 5500.

The Shake unit communicates with the phone via Bluetooth. The main advantage of the integrated accelerometer is that it does not require attaching extra hardware and removes the need for Bluetooth communication overhead. However it turned out that the communication overhead of the Bluetooth link was negligible. At the same time we found that the audio playback on the Nokia 5500 introduced significant latency of about 500 ms. Hence we implemented all gestures with both platforms and the implementation with the Shake unit using a Nokia 6630 camera phone as host device for audio rendering showed latency behavior appropriate for real-time performance.

2.2. Signal Conditioning

In order to clean up the signal for practical use we employ a two-step process. First, the signal is filtered for overall smoothing and to remove frequency content that is higher than the desired interaction speed. Second values are locally locked to a mean value to remove noise in the signal that is above the signal to noise ratio. For the low-pass filter we employ simple FIR averaging filters.

2.3. Shake Unit Sensor Integration via Bluetooth

The weight of the Nokia 5500 and the attached Shake unit is 170g. The sensor is attached at the lower end of the phone, which was found to be the ergonomically best position. After attaching it to the phone the magnetometer has to be recalibrated, because of electro-magnetic interference from the phone.

3. MOBILE STK INTEGRATION

Mobile STK, a Symbian OS port of the Synthesis Toolkit by Cook and Scavone [5] was integrated with software to access the sensor capacities of the Shake device and the Nokia 5500.

Nokia 5500 run the Symbian 9.1 operating system. The original MobileSTK sound synthesis library release for Symbian 8.0 [5] was modified to accommodate the API changes and merged with software reading accelerometer data of the phone. The main impact of this change was newly introduced latency from the internal handling of buffers of a minimum of 4096 bytes. The Nokia 6630 we used to originally port STK to Symbian allows for buffer sizes of 320 bytes to be queued, which leads to a $320/8000 = 0.04$s latency due to buffering. On the Nokia 5500 audio playback does not start until data of 4096 bytes has been received, leading to a latency of over 0.5s. This change is somewhat surprising as it not only impacts the type of interactive performance applications that we implemented here but also impacts real-time audio for mobile games.

While latency can theoretically be reduced by increasing the sample rate at fixed buffer sizes, we experienced that the added computational burden from computing the parametric synthesis models prevented this from being a practical option. While it is easily possible to access internal accelerometer data and use it to control streaming audio results, the current audio architecture of Nokia 5500 phones prevents this from being practical. We assume that this problem is temporary and that interactive audio playback latency will be lowered again in later architectures.

4. GESTURES AND PERFORMANCE

As a first illustration of the possibilities of using mobile phones as musical instruments we implemented a few basic gestures that are motivated by traditional musical instruments and also relate to earlier work on using existing instrument paradigms in electronic instruments [1, 2, 4] where the idea is to retain familiar gestures for matching sounds.

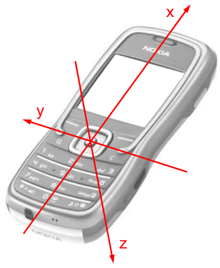

Figure 2. Orientation of tilt directions of accelerometers in a Nokia 5500 camera phone.

If one assumes the camera phone to be quasi-static, the earth's gravitation is the dominant source of continued acceleration on the sensor. By finding the direction of the vector defined by the three axes of the accelerometer one can reliably define the direction of the gravitational field, if the overall motion is otherwise only mildly accelerated. Using this property one can easily sense the tilt of the device with respect to the natural downward direction. This property has already been used to suggest navigating graphical user interface interactions like scrolling in menues via the tilt of mobile devices (see [10]).

The three directions of tilt for the built-in accelerometers are depicted in Figure 2. The arrow depicts the direction of tilt in which the accelerometer value for this axis will increase. For example tilting the device forward will increase the x-axis accelerometer reading as this vector gets more aligned with the direction of the gravitational field. The same basic principle applies to the 3-axis accelerometer inside the Shake unit. But because the device does not have a natural orientation (i.e. no display that should face the user), the core orientation of the axis is a matter of choice.

In all our implementations we tried both the built-in Nokia 5500 accelerometer data as well as data from a Shake unit. For the purpose of the viability for the implemented gestures, both platforms are appropriate, despite the drastically lower dynamic range in the signal from the Nokia 5500. We found no significant overhead in the Bluetooth communication from the Shake unit to mobile phones.

4.1. Striking Gestures

First we implemented a strike gesture to emulate impulsive striking instruments like pianos, drums, glockenspiels and the like. We do not have an absolute position relative to a virtual impact plane in space, rather we have an estimate of the tilt angle of the device relative to the gravitational field to get an angular notion of impact. To this end we define any horizontal plane independent of elevation to be the impact plane and use the crossing angle with that plane (0 degrees) in the x-axis of the phone as the impact moment. More precisely, an impact is registered when the phone had a tilt angle larger than 0 degree previously and crosses through the 0 degree in negative direction. A measure of amplitude is derived from the local gradient at the crossing defined by $|\alpha_{n-1} - \alpha_n|$ where n is the discrete time of impact and α_n is the tilt angle at time n. Hence virtual impact that hit the horizontal plane with larger angular velocity will be played with louder amplitude.

This basic gesture already allows stable single-tone performance with sensible expressivity. When using Shake accelerometers with a Nokia 6630 there is no noticeable latency between impact moment and sound playback.

In order to allow for discrete pitch play, compass readings from the magnetometer of the Shake unit are used to segment the plane into angular sectors (see Figure 3). The compass algorithm built into the Shake units uses the accelerometer readings to convert the magnetometer data to a planar angular reading after calibration to the earth's magnetic field at the geographic location. But as the accelerometers do experience additional forces during a strike action, this compensation algorithm gets disturbed leading to a variation in displayed compass angle of around 20 degrees. For this reason we chose a angular segment of 45 degrees, yielding 8 discrete positions to fit the first four notes of a C-major scale starting at 261.6 Hz. The positions are fixed relative to the earth's magnetic field, hence the performer can learn what positions corresponds to what sounds and can play pitched notes reliably and accurately. This way small virtual pianos, bar percussion instruments and dulcimers can be simulated preserving familiar gestures.

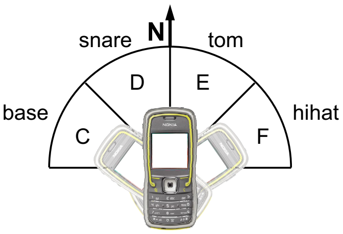

Figure 3. Spatial play for strike gestures using compass bearings. Each 45 degree segment is assigned a specific function, for example pitch.

4.2. Shaking Gestures

We implemented a shake gesture familiar from percussive rattle instruments such as rattles, tambourines and the like. This implementation is inspired by the PhiSEM controllers by Cook [2]. The basic observation here is that acceleration directly relates to the force exerted on the rattle via Newton's law $F = ma$. We make the simplifying assumption that we are always quasi-stationary. This means that the last vector describes the direction of the earth's gravitational field and that a change to that direction constitutes the induced acceleration by the motion. Hence we can compute the magnitude of the local acceleration vector as the normed difference between old and new accelerator vector data

$$|a| = \sqrt{(x_n - x_{n-1})^2 + (y_n - y_{n-1})^2 + (z_n - z_{n-1})^2}$$

where x, y, z are accelerometer readings at discrete times n. This magnitude is the scaled to directly drive physically informed particle models [1].

4.3. Sweeping Gestures

Finally we wanted to allow a gesture that more closely matched the energy infusion typical for sustained sounds. A violin bow action would be an example of such a gesture. In order to implement the violin bow action metaphor we implemented a sweeping gesture.

While the precise dynamics of violin bowing is complex, bow velocity is an important determinant of the resulting sustained sound [3]. Hence the goal is to find a way to read steady velocity from gestures to use as drivers for sustained sounds.

Due to the inherent drift introduced by the noise of accelerometer data when integrated to get velocity we decided to use magnetometer data to arrive at velocity.

Absolute angular position again can be used to derive meaningful data here. We derive discrete position over time $\Delta\beta = \frac{\beta_n - \beta_{n-1}}{h}$, where β_n is the planar compass heading angle at time n chosen with respect to some absolute 0 degree and h being the discrete time step size, to arrive at discrete angular velocity. Unlike the integration of accelerometer data, this operation does not accumulate over time hence is insensitive to noise in the sensor signal over time. The tilt angle of the accelerometer in the x-axis is used as bowing pressure using the analogy of leaning into the string for added pressure. In principle further tilt axis, like the y-axis could be used as bowing tilt, but our currently available synthesis engine (default STK) did not include a bow width model. However such models are available [12].

5. CONCLUSIONS

Shamus demonstrates a first integrated sensor-based mobile performance system that is completely untethered while allowing a wide range of motion based gestures. Mobile STK was integrated with sensor reading of integrated accelerometer sensors for Symbian phones, if such sensors are available or alternatively using a Shake unit for the same purpose. The Shake sensor unit also contains magnetometer readings which we use to for position information in the plane and for velocity estimations in striking and sweeping gesture algorithms.

We only demonstrated a basic system here, leaving a wide array of possible gestures or synthesis approaches open for future work. Also, while we implemented a basic networked performance for the related CaMus system, which uses optical tracking for motion sensing, full networked support for sensor-based mobile device performance is also still to be developed.

6. REFERENCES

[1] P. R. Cook. Physically Informed Sonic Modeling (PhISM): Synthesis of Percussive Sounds. *Computer Music Journal*, 21(3):38–49, 1997.

[2] P. R. Cook. Principles for Designing Computer Music COntrollers. In *ACM CHI Workshop in New Interfaces for Musical Expression*, Seattle, 2001.

[3] L. Cremer. *The Physics of the Violin*. MIT Press, Cambridge, MA, 1984.

[4] G. Essl and S. O'Modhrain. Scrubber: An Interface for Friction-induced Sounds. In *Proceedings of the Conference for New Interfaces for Musical Expression*, pages 70–75, Vancouver, Canada, 2005.

[5] G. Essl and M. Rohs. Mobile STK for Symbian OS. In *Proc. International Computer Music Conference*, pages 278–281, New Orleans, Nov. 2006.

[6] L. Gaye, L. E. Holmquist, F. Behrendt, and A. Tanaka. Mobile music technology: Report on an emerging community. In *NIME '06: Proceedings of the 2006 conference on New Interfaces for Musical Expression*, pages 22–25, June 2006.

[7] G. Geiger. PDa: Real Time Signal Processing and Sound Generation on Handheld Devices. In *Proceedings of the International Computer Music Conference*, Singapure, 2003.

[8] G. Geiger. Using the Touch Screen as a Controller for Portable Computer Music Instruments. In *Proceedings of the International Conference on New Interfaces for Musical Expression (NIME)*, Paris, France, 2006.

[9] S. Hughes. *Shake – Sensing Hardware Accessory for Kinaesthetic Expression Model SK6*. SAMH Engineering Services, Blackrock, Ireland, 2006.

[10] J. Rekimoto. Tilting operations for small screen interfaces. In *Proceedings of the 9th annual ACM symposium on User interface software and technology*, pages 167–168. ACM Press, 1996.

[11] M. Rohs, G. Essl, and M. Roth. CaMus: Live Music Performance using Camera Phones and Visual Grid Tracking. In *Proceedings of the 6th International Conference on New Instruments for Musical Expression (NIME)*, pages 31–36, June 2006.

[12] S. Serafin and J. O. Smith. A Multirate, Finite-Width, Bow-String Interaction Model. In *Proceedings of the Conference on Digital Audio Effects (DAFX)*, Verona, Italy, December 7-9 2000.

[13] S. Strachan, P. Eslambolchilar, R. Murray-Smith, S. Hughes, and S. O'Modhrain. GpsTunes: Controlling Navigation via Audio Feedback. In *Proceedings of the 7th International Conference on Human Computer Interaction with Mobile Devices & Services*, Salzburg, Austria, September 19-22 2005.

[14] A. Tanaka. Mobile Music Making. In *NIME '04: Proceedings of the 2004 conference on New Interfaces for Musical Expression*, pages 154–156, June 2004.

[15] A. Tanaka, G. Valadon, and C. Berger. Social Mobile Music Navigation using the Compass. In *Proceedings of the International Mobile Music Workshop*, Amsterdam, May 6-8 2007.

[16] J. Williamson, R. Murray-Smith, and S. Hughes. Shoogle: Excitatory Multimodal Interactions on Mobile Devices. In *Proceedings of CHI*, 2007.