Review of sound synthesis and effects processing for interactive mobile applications

Jyri Pakarinen, Henri Penttinen, Vesa Välimäki, Jussi Pekonen*

Dept. of Signal Processing and Acoustics, Helsinki University of Technology, P.O. Box 3000, FI-02015 TKK, Espoo, Finland. Tel: +358-9-451-6027, Fax: +358-9-460-224, email: jyri.pakarinen@tkk.fi (Corresponding author)*

Jarno Seppänen

Nokia Research Center, Helsinki, Finland

Fredéric Bevilacqua, Olivier Warusfel

IRCAM, Paris, France

Gualtiero Volpe

UGDIST, University of Genova, Italy

Abstract

Several interesting mobile applications using sound synthesis and audio effects processing have emerged in the last few years. As the processing power and sensor arsenal of mobile devices has constantly increased, many of these applications are able to turn the mobile device into a new musical instrument with fascinating new sonic properties. This report discusses the technical possibilities given by modern mobile platforms and reviews the state-of-the-art applications for sound synthesis and effects processing in the mobile context.

Keywords: mobile communication, acoustic signal processing, music.

1. Introduction

Sound synthesis and effects processing are widely used in the current mobile device industry. Virtually all modern mobile phones contain at least a simple synthesizer for sonifying MIDI ringtones, while more sophisticated synthesis and effect applications are constantly being introduced in the market. Also, modern multi-modal interfaces can improve the usability of these applications, making the synthesizers more intuitive to control.

The aim of this report is to review those sound synthesis and audio effect techniques that are suitable for current mobile devices, as well as to study those state-of-the-art mobile audio applications that relate to sound synthesis.

The following subsection lists some of the state-of-the-art mobile applications, while Section 2 discusses the properties of modern mobile device platforms. Various sound synthesis techniques suitable for mobile use are reviewed in Section 3. Section 4 discusses several audio effects processing applications for mobile usage, while gestural control issues are presented in Section 5. Section 6 presents a novel virtual air guitar application as a design example. Finally, conclusions are drawn in Section 7.

1.1. State-of-the-art applications

Several interactive virtual instrument applications can already be found on mobile devices. Virtual instruments, such as the piano[1], PocketGuitar[2], or the IR-909 drum machine[3] are available for the Apple iPhone[4]. The iPhone's multi-touch screen allows the player to use familiar playing gestures in controlling the instruments. For example, the player can fret the strings on the PocketGuitar by positioning his fingers on the touch-screen as he or she would on a real guitar.

The Nintendo DS mobile gaming console[5] also offers virtual musical instruments in the Jam Sessions videogame[6], where also a guitar application, strummable by the console's control pen, can be found. Also, the electronic synthesizer manufacturer Korg has recently introduced a virtual analog synthesizer DS-10[7], exclusively for the Nintendo DS. Also the popular Guitar Hero videogame[8] has been released as a mobile phone application[9]. Instead of using an external guitar controller, the user plays the game by pressing the numeric buttons on the phone in the correct time instants.

Regarding sound synthesis tools and platforms, Pure Data (PD) (Puckette, 1996) has been ported to mobile devices that support the Linux environment such as the iPaqs (Geiger 2003, 2006). It utilizes the touch-screen capabilities for controlling sound synthesis parameters. In addition, the Synthesis Toolkit (STK) (Cook and Scavone 1999) has recently been ported to the Symbian OS as MobileSTK (Essl and Rohs 2006). It is the first fully parametric synthesis environment available on mobile phones.

2. Mobile platform

Mobile phones have become powerful tools. Those hi-end products which, in addition to making phone calls, are capable of doing many of the tasks computers do, are today called smart phones. Smart phones typically contain a

[1] http://moocowmusic.com/Pianist/
[2] http://code.google.com/p/pocketguitar/
[3] http://www.cratekings.com/iphone-ir-909-drum-machine-and-iphonesynth/
[4] http://www.apple.com/iphone/
[5] http://www.nintendo.com/ds
[6] http://www.youtube.com/watch?v=GyaEzMGiANE
[7] http://www.youtube.com/watch?v=rorBOzwR3Tc
[8] http://www.guitarhero.com/
[9] http://www.guitarheromobile.com/
(all above URLs retrieved on 2008-09-14).

microphone, keypads, one or two loudspeakers, a two to five mega-pixel video/still camera, a 3D accelerometer, Bluetooth, wireless local-area network (WLAN) capability, GPS, color LCD display with possible touch pad capabilities and a processor with 200-400 MHz CPU (Central Processing Unit)[10]. Many of these technologies have become viable during recent years because they have matured and the prices have dropped steadily. Figure 1 displays two state-of-the-art mobile devices, the Apple iPhone and Nokia N95 8GB. The main differences with these devices and laptop and tabletop computers are the amount of memory, computation speed, and power consumption. These differences will be highlighted throughout the report. Here we present the main and typical features of today's mobile phones that are relevant in interactive mobile applications.

Figure 1: Two examples of modern mobile phones with gestural input possibility: the Nokia N95 8GB (left) and the Apple iPhone (right).

2.1. Microphone

A mobile phone naturally has a microphone. Its fidelity is optimized for close range recording, less than twenty centimeters. However, loud sound sources and background noises can be captured from longer distances.

---

[10] Some examples of such devices:
Nokia Products, http://europe.nokia.com/products
Apple - iPhone, www.apple.com/iphone/
Sony-Ericsson, http://www.sonyericsson.com/
(all above URLs retrieved on 2008-09-15).

For musical and interactive applications, live sampling and playback can be utilized as was done for example in the Cellphone Quartet in C major, op. 24 (Wang et al., 2008). Through signal analysis, such as estimation of the background noise level, the microphone can be used for context-aware applications.

2.2. Three-axis accelerometer

A significant number of current mobile devices have integrated three-dimensional accelerometers, much for the same reason that digital cameras have them: to automatically rotate photos between portrait and landscape orientations. A 3D accelerometer measures the acceleration that results from forces acting on the phone, in three dimensions, relative to the device itself. As a direct consequence, the earth's gravity (g) is always present in the accelerometer readings, superimposed to the movements of the device.

The current accelerometers have a −2 g to 2 g measurement range and an 8-bit resolution for each axis. The sampling frequency is approximately 30 Hz. As such, the accelerometer data stream is quite limited but nevertheless sufficient for a number of uses, e.g., in terms of activity recognition.

Accelerometers can be utilized in nearly every music application, by designing gesture controls – a mature field of research (Camurri and Volpe 2003; Paradiso 1997; Wanderley and Depalle 2004). On the other hand, accelerometer signals are highly useful in nonintrusive context-aware applications, because they can provide valuable activity information without much CPU or memory usage (Karantonis et al. 2006). Accelerometer data has also been used in a custom-made augmented PDA device that controlled streaming audio (Tanaka 2004).

2.3. Camera

Most of the current mobile devices contain one or two integrated cameras. For example, the Nokia N95 has a 5 megapixel camera at the back (away from the user) and a 0.1 megapixel camera at the front (towards the user). The back camera is used for taking photos while the front camera is for video calls. Often the back camera is behind a lens shield, which must be opened mechanically by the user, and which automatically launches the camera application in the device.

The cameras are capable of still and live video recording. Recent devices have integrated flash and image processing capabilities for digital zoom, exposure, and white balance. The images can be captured in raw RGB (Red Green Blue) and JPEG (Joint Photographic Experts Group) formats. The CaMus system (Rohs et. al., 2006) used mobile camera data for sound synthesis control. This is discussed more thoroughly in Section 5.

The cameras are potentially useful for context sensitive applications as well, provided that sufficient feature extraction and recognition is implemented at the client device. The front camera is potentially more useful because it does not need the user opening the camera shield. For example, with the front camera, one could detect whether the mobile device is in a pocket or bag or whether it is in daylight.

Many devices feature also an integrated light brightness sensor near the display. The sensor measures the ambient brightness in the front of the phone and adjusts the display brightness accordingly. Such a sensor lends itself naturally also to context-aware applications.

2.4. Touch screen

Touch-screen technology provides a flexible means of input data. There are three basic systems[11] that are used to recognize a person's touch: resistive, capacitive, and surface acoustic wave. One of the main practical differences is that a capacitive screen does not react to a stylus whereas the other techniques do. A capacitive screen is controlled with by using a bare finger. Multi-touch screens enable even more complex gestures. The aforementioned PocketGuitar and Jam Sessions are good examples of musical applications that exploit the touch screen.

2.5. Location acquisition

The location of modern mobile devices can be estimated using multiple technologies: GPS (Global Positioning System), GSM/3G network cell identification, and WLAN neighborhood discovery. Each of the technologies have their strengths and weaknesses, and none is fully able to replace the others.

Indoor location cannot be obtained with GPS, but it can be estimated using WLAN neighborhood signals. All recent mobile devices have WLAN radios built in for wireless networking purposes, up to ranges of a hundred meters. The WLAN neighborhood provides a useful indication of the indoor location, because WLAN access points are often positioned statically inside buildings.

Location- and context-aware services are primary applications of location data. However, for most practical context-sensitive applications, GPS locations may be unnecessarily precise, especially considering the battery life with GPS reception enabled. Further, GPS location is only available in limited scenarios, mostly only when the user has intentionally obtained the GPS fix, e.g., by launching a navigation software. A much less intrusive, however also less precise location information can be computed from the mobile cell tower identifiers.

GPS based interactions have recently also been utilized in artistic applications (Strachan et al. 2005; Tanaka et al. 2007). However, by default they do not use onboard sonification, but use an external computer for sound generation.

2.6. Bluetooth wireless communications

Bluetooth radio technology is a standard component of mobile devices today. It is a wireless communication protocol designed for connecting devices and accessories in short ranges, up to 10 meters. Bluetooth[12] v1.1 and v1.2 are both

---

[11] How do touch-screen monitors know where you're touching? http://electronics.howstuffworks.com/question716.htm, retrieved 2008-09-15.

[12] Bluetooth v1.1, IEEE Standard 802.15.1-2002, Bluetooth v1.2, IEEE Standard 802.15.1-2005.

IEEE standards. The specifications of the latter are controlled by the Bluetooth special industry group (SIG)[13].

Bluetooth neighborhood provides a useful indication of the social situation for context-aware applications. This is because the Bluetooth devices, especially mobile devices, are personal devices that usually indicate who is present at the same space.

2.7. Example platform: EyesWeb Mobile

In order to develop applications exploiting gestural control of sound synthesis and effects from mobile devices enabling tools and platforms are needed. In this direction the EyesWeb XMI platform for eXtended Multimodal Interaction (Camurri et al., 2007) has been recently extended with a new component, EyesWeb Mobile, explicitly devoted to provide an interface to EyesWeb XMI from mobile devices.

EyesWeb Mobile is an application for both desktop computers and mobile devices running Windows Mobile operating system. In its current implementation, EyesWeb Mobile is a user interface for the remote control of applications running on EyesWeb XMI servers.

The EyesWeb Mobile client supports transmission to the server of the sensorial inputs available on the mobile device it runs on (e.g., webcam, audio input, accelerometers, GPS, etc.). It can also exploit EyesWeb XMI to perform some processing of such data on the mobile device itself.

EyesWeb Mobile has been recently used to remotely control from a mobile device the interactive music installation Mappe per Affetti Erranti (Camurri et al., 2008), a first example of active listening paradigm where users, in a social context, can navigate and mould music content through their movement and gesture at multiple levels: from navigation in a physical space to explore the polyphonic structure of a music piece up to affective, emotional spaces to explore different expressive performances of the same music piece.

Figure 2 shows EyesWeb Mobile running on a mobile device (DELL Axim X51). The server is running on the notebook on the background. In the simple example, the notebook is connected to a webcam and the video stream is being sent to EyesWeb Mobile via a standard WLAN connection.

[13] Bluetooth special industry group, https://www.bluetooth.org/, retrieved 2008-09-15

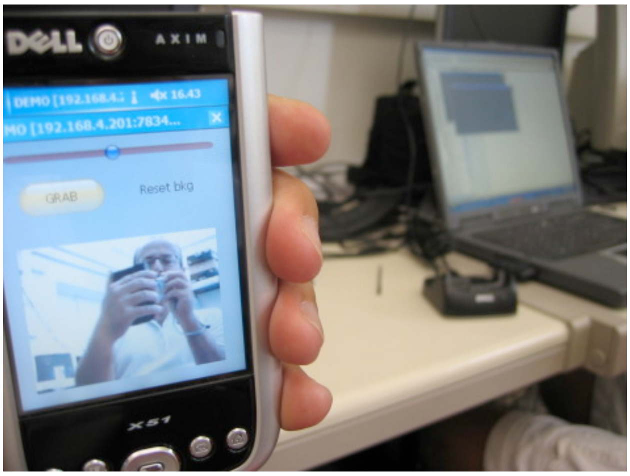

Figure 2: EyesWeb Mobile running on a DELL Axim X51. Images are captured by a webcam on the notebook on the background and streamed to the mobile via a standard WLAN connection.

3. Sound synthesis

Digital sound synthesis aims to create new sounds by artificially generating waveforms or by modifying pre-stored sound signals using computational algorithms. This section discusses sound synthesis techniques that are not computationally extremely demanding. Therefore, these synthesis methods could be used in current mobile applications.

3.1 Physics-based methods

Physics-based synthesis methods create sounds by simulating the behavior of the sounding object, i.e. the object producing the sound. This allows the synthesis control signals and parameters to be chosen so that they have a strong correspondence to actual physical quantities. This, in turn, often leads to creation of synthesizers, which are intuitive and relatively easy to control. The caveat, however, is that since the models are trying to simulate real physical entities, their computational complexity might be overwhelming for current mobile applications. Some computationally light physics-based sound synthesis methods are discussed in the following. For a more thorough review on physics-based discrete-time sound synthesis techniques, refer to (Välimäki *et al.* 2006).

Digital waveguide (DWG) modeling (Smith 1992) is best suited for simulating sounding objects, which produce harmonic sounds, such as string- or wind instruments. In practice, DWGs are implemented using delay lines with dissipative feedback, so that an input signal circulates within the delay line and gradually attenuates. An early string model, the Karplus-Strong algorithm (Karplus and Strong 1983), can be seen as a first implementation of a simple DWG string. This straightforward algorithm requires only a few operations per sample and is generally well suited for mobile applications, although low notes require long delay lines which often can not be implemented in mobile devices.

On the other hand, poor mobile loudspeaker performance for low notes most likely restricts the frequencies anyway. For more information on waveguide synthesis of string instruments, see papers by Välimäki et al. (1996) and Karjalainen et al. (1998).

Source-filter models are based on the idea that a sounding object consists of a source that feeds acoustic energy into the system and a filter or resonator that colors the sound of the source. Although source-filter models do not necessarily need to represent any physical system (consider, for example subtractive synthesis, discussed in Section 3.2) they can be seen as a physics-based modeling scheme for some cases, such as the human vocal-tract (Klatt 1980) or string instrument body (Karjalainen and Smith 1996). In many cases, source-filter models offer a computationally and conceptually simple sound synthesis method, although the mapping between synthesis parameters and physical quantities might be vague. Thus, source-filter models are a good candidate for mobile sound synthesis.

In modal synthesis (Adrien 1991), the synthesizer is designed by describing the vibrational properties of the sounding object in the frequency domain. After the most important eigenfrequencies have been listed, the vibrating system can be simulated using e.g. a parallel resonator bank. An input matrix, giving the relation between the excitation location and the excited modes, is also often defined. Modal synthesis is especially suitable for synthesizing inharmonic sounds such as bells or gongs, since the modal frequencies can be chosen freely. For spectrally simple sounds (e.g. 10 modes or less) modal synthesis suits also mobile applications.

Mass-spring networks (Cadoz et al. 1983; Florens and Cadoz 1991) consider the sounding object as a collection of point-like masses, connected together with a set of idealized springs. Mass-spring models are particularly well suited for sonifying objects which contain separate interacting sub-particles, such as shakers. However, since the system is defined using local interactions between elementary particles, imposing global rules for the behavior of the entire object (such as tension modulation nonlinearity in strings) might be difficult.

3.2 Abstract methods

3.2.1 Sampling and wavetable synthesis

An intuitive sound synthesis method is to play back digital recordings, sample wavetables, from the memory. This synthesis technique is called sampling. The length of each sample can be arbitrarily long, limited only by the memory capacity (Roads, 1995). Figure 3 illustrates the block diagram of a typical sampling synthesizer.

Since most musical sound waves are repetitive, an efficient synthesis method is to store the values of a single period of a tone into memory. It is called a wavetable. In order to reproduce the same tone, the stored wavetable is read in a loop, again and again. A sound synthesis technique implementing these procedures is called wavetable synthesis (Roads 1995). Although the wavetables are usually small in size, many different wavetables can consume much memory. Therefore data reduction must be considered. Most commonly the data compression is implemented by differential coding, where the difference between adjacent samples is stored (Maher 2005).

To produce tones of different pitch, the sample increment for the table look-up must be changed. Since the fundamental frequency can be arbitrary, the sample increment is not always an integer. The best solution to the non-integer sample increment is to interpolate the wavetable value at the obtained position. Interpolation can be implemented efficiently with fractional delay filters (Laakso et al., 1996).

In order to produce time-varying timbres, some modifications to the wavetable synthesis technique can be applied. In wavetable crossfading, the synthesizer plays two wavetables simultaneously adjusting their gain over the course of an event instead of scanning only one wavetable. In wavetable stacking, a set of wavetables are mixed with their corresponding envelope functions (Roads, 1995). Additionally, a combination of sampling and wavetables can be utilized. Sampling may be used for the attack while wavetable synthesis is used in the tone's decay phase (Yuen and Horner, 1997). Wavetable synthesis with good sound quality is obtained by finding wavetable spectra and the associated amplitude envelopes which provide a close fit to an original time-varying spectrum. This can be done with a genetic algorithm or with principal component analysis methods (Horner et al., 1993; Beauchamp and Horner, 1995), or by grouping the harmonics of the signal into separate wavetables (Horner and Ayers, 1998).

Scanned synthesis is a related technique that can be thought as an extension of wavetable synthesis (Verplank et al., 1998). It involves a dynamic wavetable, from which the audio signal is read, and usually a haptic sensor, which controls slow variations of the dynamic wavetable. For example, the wavetable can in this case be a two-dimensional array, which contains modeled vibrations of a membrane that is excited based on sensor data. The scanning can take place on a circular path on the two-dimensional array. This method combines user's gestures and synthesis in a meaningful way and can be very useful for mobile applications.

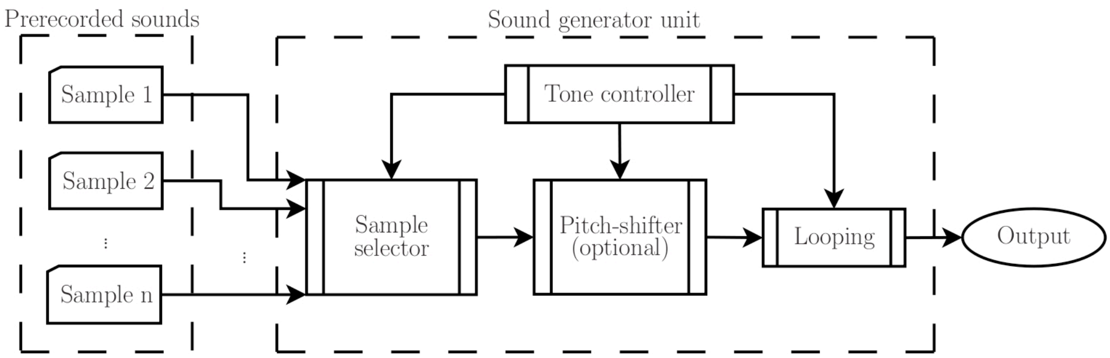

Figure 3: Block-diagram of a sampling synthesizer (adopted from (Pekonen 2007)).

3.2.2 Additive synthesis

Additive synthesis, as its name suggests, is based on summation of sinusoidal components to generate a spectrally more complex waveform (Roads, 1995). In addition, the generator may add colored (filtered) noise to the resulting signal (Serra and Smith, 1990). In mobile applications, additive synthesis can provide an efficient algorithm for timbres with only a very few spectral components, such as organ sounds. For more complicated sounds, inverse FFT-based sound generation is commonly used to alleviate the computational load (Chamberlin, 1985; Rodet and Depalle, 1992). However, the real-time computation of FFT and inverse FFT is currently a large task for a mobile audio processor, but it may become attractive in the future.

3.2.3 Subtractive synthesis

The term 'subtractive synthesis' is often used in computer music to describe techniques that are essentially source-filter models (Roads, 1996). This process is called subtractive synthesis, since the source signal is usually a broadband signal or a harmonically rich waveform, which is then modified with a filter to obtain the desired sound. Digital subtractive synthesis is nowadays called virtual analog synthesis, when reference is made to computational methods that imitate the sound generation principles of analog synthesizers of the 1960s and 1970s.

Subtractive synthesis is more demanding to implement using digital signal processing techniques than is generally understood. One problem is aliasing caused by the sampling of analog waveforms that have sharp corners, such as the square wave or the sawtooth wave. The spectrum of such waveforms continues infinitely high in frequency, and the signals are thus not bandlimited. Several algorithms have been proposed to generate discrete-time versions of analog waveforms so that aliasing is completely eliminated (Winham and Steiglitz, 1970; Moorer, 1976) or is sufficiently suppressed (Stilson and Smith, 1996; Välimäki and Huovilainen 2007). Another difficulty is that analog filters do not obey the linear theory exactly: at high signal levels they generate nonlinear distortion. This does not naturally occur in discrete-time signal processing, but it must be implemented, for example, by using a nonlinear function (Rossum, 1992; Huovilainen, 2004) or by describing a circuit model with nonlinear differential equations, which are then solved using numerical methods (Civolani and Fontana, 2008).

3.2.4 FM synthesis and other methods

Frequency Modulation (FM) was not applied to audio frequencies and sound synthesis purposes until late 1960s (Chowning, 1973). In FM synthesis, the instantaneous phase of a sound signal is varied with a modulator signal, i.e., the frequencies of the original waveform oscillate around their nominal values along the modulator signal. A related modulation technique called phase modulation (PM) is a special case of FM, or other way round. FM and PM synthesis techniques offer a computationally efficient way of generating a wide variety of musical sounds and are therefore attractive for mobile use.

In a simple FM synthesizer, the amplitude ratios of the newly generated signal components vary unevenly according to Bessel functions when the modulation index parameter is varied. This problem can be overcome by using feedback FM (Tomisawa, 1981). In the simplest implementation of the feedback FM synthesizer, the frequency of a single oscillator is modulated according to its own output. In two-oscillator feedback FM synthesizer, the feedback is used to drive the modulator oscillator.

A recently introduced variation called adaptive FM synthesis can bring about FM synthesis-like effects to arbitrary audio signals (Lazzarini et al., 2008). The modulator is assisted with a pitch detector. The modulation is implemented by varying the length of a delay line in which the input signal propagates. When modulation is turned off, the output signal will be identical to the original signal. Familiar sounding FM synthesis effects are obtained with non-zero values of the modulation index.

A novel synthesis method, reminiscent of FM synthesis, is the logical synthesizer introduced by Kleimola (2008). This synthesis method applies bitwise logical operations (OR, AND, XOR) between two signals, and thus efficiently generates synthetic sounds with wide spectra. Another exotic synthesis technique uses circle maps, nonlinear algorithms that efficiently create both harmonic- and noise-like sounds (Essl 2006).

4. Effects processing

The sound produced by electric and acoustic instruments is sometimes considered quite dull and dry. Therefore the sound is usually processed with additional sound effects, which brings liveliness to the plain instrument sound. There are numerous different effects designed for creating different kinds of expressions. However, perhaps the most commonly used effects are chorus, flanger, phaser, reverb, and distortion.

4.1 Chorus, flanging, and phasing

Several typical effects processing algorithms can be implemented with a common structure: a copy of the input signal is processed and mixed with the input signal. The chorus effect creates an illusion of multiple simultaneous sounds (Dattorro, 1997). A simplified implementation of chorus is called doubling, where the original sound and its delayed copy are mixed together. This structure is called an FIR (finite impulse response) comb filter. When more than one delayed copy is added, each with independent, possibly time-varying delay, a more realistic chorus effect is obtained.

Another popular effect algorithm, the flanging effect, is essentially similar to doubling, but the delay-line length varies over time, for example by using a sinusoidal low-frequency oscillator (Dattorro, 1997). This leads to a filter structure similar to the chorus, and these two effects are usually implemented with the same filter by changing the filter coefficients. Despite the computationally efficient filter structure, the chorus and flanging effects require a rather long delay line, which is problematic in memory limited systems. For this reason, it may only be feasible in a mobile system to use the flanging effect with a short delay line.

In the phasing effect, time-varying notches in the spectrum are created by mixing allpass filtered input signals with the original signal, leading to a slightly different sounding effect than chorus and flanging. The digital phaser can be implemented by using second-order allpass filters in cascade (Smith, 1982). Each allpass filter creates one notch, so the desired number of notches determines the number of required state variables. The phasing effect is more complicated in terms of the number of operations than flanging and chorus, but it does not require a large delay-line memory.

4.2 Nonlinear processing

Probably the most widely used nonlinear audio processing technique is that of dynamic range compression (DRC). Basically, DRC algorithms aim to attenuate loud signal levels, while keeping low signal levels unaffected. This results in an audio signal with reduced dynamic range when compared to the original. Since mobile audio devices are usually equipped with relatively low-quality loudspeakers, applying DRC can be desirable, since the result often sounds stronger or more coherent. This can considerably improve the intelligibility of the signal if the listening environment is noisy, as might well be the case with mobile devices.

However, when high-quality loudspeakers are used, DRC techniques do not fit very well with certain type of signals. If heavy DRC is applied for acoustic instrument music, for example, the important musical nuances will be lost. Increasing amounts of DRC have been used in the production contemporary pop- and rock music, leading to a situation called "loudness war"[14]. Digital algorithms for obtaining DRC are discussed e.g. in the book (Zölzer, 2002).

If the instantaneous signal gain is changed too rapidly, new frequency components are created in the signal spectrum. This phenomenon, called nonlinear distortion, can be desirable as a special effect for example in the electric guitar. In the simplest case, nonlinear distortion can be obtained by applying a nonlinear function (such as hyperbolic tangent) to the signal. This approach is called waveshaping (Le Brun 1979; Arfib 1979). The nonlinearity can also be read from a pre-stored lookup-table (Kramer, 1989), if physical memory requirements do not restrict this. Also more sophisticated dynamic modeling techniques can be used for simulating real guitar tube amplifiers. For an extensive review on digital guitar tube amplifier modeling techniques, see (Pakarinen and Yeh 2009). A simple distortion effect is implemented in the virtual mobile air guitar, discussed in Section 6.

Exciter and enhancer algorithms aim to add artificial brightness or clarity to the sound signal. Instead of simply boosting the high-frequency-content, these effects apply a mild nonlinear distortion, possibly combined with equalization

---

[14] http://en.wikipedia.org/wiki/Loudness_war

and phase-shifting. For a more thorough discussion on exciters and enhancers, see (Zölzer, 2002).

4.3 Spatial effects

Integrating 3D audio reproduction is an important factor for creating convincing interactive environments. Our spatial auditory perception contributes to the localization of objects in direction and distance, the discrimination between concurrent audio signals and self-representation in the environment. In the context of interactive applications, the introduction of auditory cues associated to the different components of a virtual scene together with auditory feedback associated to the user interaction enhances the sense of immersion and presence (Hendrix, 1996; Larsson et al., 2002).

4.3.1. Sound localisation

One of the primary goals of spatial audio rendering is to reconstruct to the ears of the listeners the desired sensation of incoming direction of the source signal (azimuth and elevation). Among the different 3D audio formats studied in audio research, binaural techniques are best suited for headphone reproduction and thus for mobile phones. They produce a two-channel output from a monophonic signal by applying a pair of filters, known as Head Related Transfer Functions (HRTFs) and resulting from direction-dependent scattering of incoming waves due to the ear/head/torso (Wightman & Kistler, 2005). They are generally obtained through direct measurement on human heads and convey all the perceptual cues involved in directional localisation: interaural time delay (ITD) and interaural level differences (ILD) both determinant for sound localisation in the horizontal plane, and spectral cues which are determinant for localizing in the vertical plane.

However, binaural rendering requires high computer resources, i.e. typically 1.5 MIPS per source with a sampling rate of 16kHz (Huopaniemi et al. 1996, Jot et al. 1998). An interesting feature of binaural techniques is that they can easily afford audio format compatibility through the paradigm of virtual loudspeakers, i.e. where each signal of a given multichannel format is rendered using the HRTF filters corresponding to the direction of the loudspeaker on which it should be fed (e.g. 5.1 setup).

4.3.2. Distance and room rendering

Sound rendering of spatial sound scenes mainly involves the simulation of *Doppler effect* associated to fast moving sources, the *directivity* of sound objects, the *occlusion/obstruction* effects linked to partition walls and the *reverberation* which will be determinant for monitoring the auditory perceptual distance of sound events and the identification of the environment (size and materials of the room).

Doppler effect implies the implementation of pitch shifting, while directivity and occlusion/obstruction can be easily rendered through gain attenuation and/or first order low-pass filters. A common approach for providing reverberation in a real-time, is based on parametric models (Gardner, 1997, Blesser 2001). Although they cannot provide a simulation of real acoustic environments as accurate as physical modeling (Min 2000, Tsingos et al. 2001, Lokki et al. 2001), they can efficiently model the main statistical properties of late reverberation in enclosures in both the frequency and time domains (i.e. density of acoustic modes and reflections). Feedback delay networks (FDNs) or waveguide networks are the most commonly used implementations (Stautner and Puckette 1982, Jot et al. 1991, Rochesso and Smith 1997). In FDN, inputs and outputs of a small number of delay units (typically 4 to 16) are connected together through a feedback matrix. The modal and echo densities of the reverberation are controlled by adjusting the delay lengths, while the exponential decay characteristics (reverberation time vs. frequency) are controlled by associating a frequency-dependent attenuation to each delay unit. It is possible to propose a series of presets mimicking the characteristics of typical enclosures of various sizes (e.g. bathroom, lecture hall, concert halls, churches, etc.). FDN also allow for special audio effects such as intinite reverberation time (Jot, 1999).

4.3.3. Rendering pipeline optimization

Typical situations encountered in interactive mobile phone applications (e.g. games, spatialised chat) require the processing of a large number of sources, which may rapidly become over the capabilities of common audio dedicated hardware. Several contributions, building on auditory perceptual properties have been proposed to make audio signal processing pipelines more efficient (Fouad et al., 1997). The general approach consists in structuring the sound scene by sorting and selecting the sound components according to their relative importance, discarding sound sources that will be masked. Further optimization can be obtained by clustering and pre-mixing neighbouring sources before sending them to the spatial processing (Tsingos et al. 2004). Several approaches have also been proposed to directly process coded audio signals yielding faster implementations than a full decode-process recode cycle (Touimi et al. 2004).

5. Design of gestural control

The use of mobile devices as powerful gestural interfaces for music is still in its infancy. Nevertheless, one can expect a rapid increase of music applications where mobile devices act as musical tangible interfaces. Also, research and experimental artistic activities have produced pioneering works on the use of mobile devices as musical interfaces.

Commercial musical applications on mobile devices have been limited to straightforward cases of touch input using either keypad or touchpad. Among research works on that area, Geiger (2006) proposed a complete set of touch screen control for a virtual guitar, drums, or the Theremin.

The recognition of gesture considered as motion of the mobile itself, using embedded accelerometers, is rapidly emerging. Simple movements such as shaking are already available commercially for advancing or randomly selecting song/sound (e.g. Sony Ericsson W910i). More advanced research was reported on gesture recognition using Bayesian Network (Choi et al. 2005, Cho et al. 2006), Hidden Markov Models (HMM) or Finite State Machines (FSM) implemented on mobile devices (Mäntyjärvi et al. 2004, Pylvänäinen 2005). Generally, these systems can recognize letters and other abstract shapes drawn in space with the mobile. Strachan (2007) developed similar recognition schema for a gesture controlled MP3 player. In particular, Strachan used a statistical model to recognize shapes of basic handling of mobile devices from accelerometer data (Strachan 2007). He derived dynamic movement primitives to process the data and operates filtering and developed a physical modeling scheme to facilitate the control of continuous parameters such as the volume. The interaction, for example, is modeled following a paradigm of a "ball in bowl". Essl et al. also developed a basic gesture recognition system to differentiate gestures such as striking, shaking, and sweeping, using both accelerometers and magnetometers, and used the recognition results to control various sounds (Essl and Rohs 2007).

Generally, these music applications do not support full expressive control of sound, but can rather be considered as a "gestural remote control". Interestingly, these works demonstrate that fairly complex gesture recognition schema can nowadays be implemented on mobile devices. Thus, we foresee that such approaches will certainly grow in the near future since such paradigms have already be proven to be efficient for music control running on a standard computer system (Bevilacqua et al. 2007).

Considering mobile devices as complete musical instruments, expressive control has been experimented in the context of "Mobile Phone Orchestra" (Wang et al. 2008). Simple mapping from gesture to sound have been applied in this case: both triggering of sound events and continuous control from accelerometer data were used to control various synthesis engines in mobile devices.

As described previously, most mobile applications have been taking advantage of embedded accelerometers. However, others sensors can also be effectively used as discussed in (Essl and Rohs 2007). In the CaMus system (Rohs et. al., 2006) the camera was used to track the distance and orientation of the phone from a sheet of paper to allow control of synthesis parameters on a laptop. CaMus2 (Rohs and Essl, 2007) extended this to allow multiple mobile phones to communicate with each other and with a PC via an ad hoc Bluetooth network. Using the mobile camera viewing a paper with a printed structure, they were able to compute the spatial position and orientation of the phone, which was then used to control a commercial sequencing software.

On an experimental level, as reviewed in (Gaye et al. 2006), note that a community has emerged on mobile music technology, generally with a focus on collaborative systems and social issues. In several cases, the use of GPS information is used to map geographical information to sound/music selection. Nevertheless, several of these works also uses sensors' input on mobile devices or small computer systems. For example SonicCity (Gaye 2003) utilized several sensors (e.g. light, microphone, accelerometers, IR proximity sensor) with a wearable computer to create a sonic environment that responds to the urban environment. For example, basic motion such as start, stop and the starting user pace are calculated from the accelerometers and determines the tempo of generated music. Note that the measurement of the walking or running pace for the selection of a matching song in a playlist were also reported in (Elliott and Tomlinson 2006, http://synchstep.com/) and (Biehl et al. 2006).

6. Use case: virtual air guitar on a mobile phone

As an example of all the topics discussed in this report we designed a virtual air guitar for a mobile phone. This implementation is loosely based on previously introduced virtual air guitars (Karjalainen et al. 2006; Pakarinen et al. 2008). The synthetic instrument is played by moving the mobile phone rhythmically. Each time a fast gesture is detected the song moves forward to the next note. Hence, the player controls the tempo of the song. The block diagram of the application is shown in Figure 4. It consists of a gesture recognition block, gesture mapping, a sound synthesizer, and a distortion model.

The gesture recognition is based on the analysis of the 3D accelerometer data. In practice, the acceleration of the three axes are squared and summed and a threshold is set for onset detection. In the case of a strong change in the acceleration the next note in the song is played. This simple gesture mapping provides practical and natural control for the player. The sound is produced with a synthesizer with a table of 2048 fixed-point sinusoid values. The perfect fifth chord is generated with the sinusoidal synthesizer, and the output is heavily distorted with a nonlinear distortion model (Doidic et al. 1998). Although the output from the synthesizer has (ideally) only two frequency components, the saturating nonlinear distortion effect creates sum and difference components that at the end produce a sound that resembles a distorted electric guitar. Aliasing, physical-modeling, vibrato, and other issues have been discarded in this version. Naturally, complexity can be added to all the stages of the application. However, this interactive virtual instrument functions as a design example of a mobile application were only the rudimentary components are implemented while still maintaining all the desired functionality and design goals.

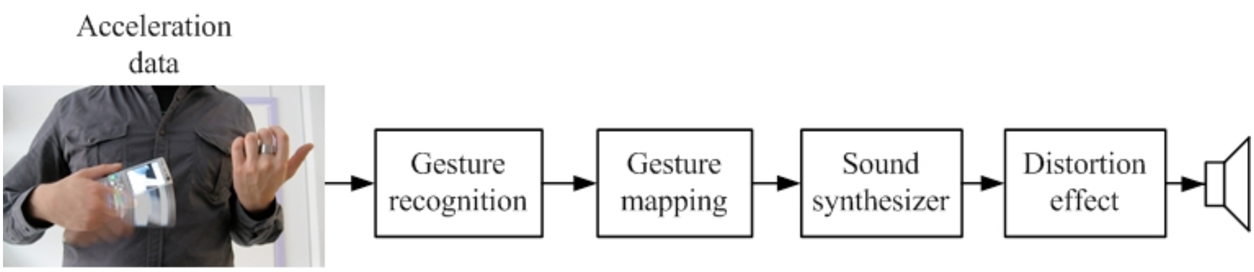

Figure 4: Block diagram of the mobile virtual air guitar.

7. Conclusions

This report reviewed sound synthesis and effects processing techniques suitable for mobile devices, and discussed the related state-of-the-art applications. Gestural control issues of the related mobile applications were addressed, and a mobile virtual air guitar was introduced as a use case.

8. Acknowledgements

This research has been funded by the European Commission 7th Framework Programme SAME project (no. 215749).

9. References

Adrien, J-M. 1991. "The missing link: modal synthesis" *Representations of Musical Signals* ed. G. De Poli *et al* (Cambridge: MIT Press) pp. 269–297

Arfib, D. 1979. "Digital Synthesis of Complex Spectra by Means of Multiplication of Nonlinear Distorted Sine Waves." *Journal of the Audio Engineering Society* 27(10):757–768.

Bevilacqua, F., Guédy, F. Schnell,N., Fléty, E., and Leroy, N, 2007. " Wireless sensor interface and gesture-follower for music pedagogy", Proc. of the International Conference of New Interfaces for Musical Expression (NIME 07), pages 124-129.

Biehl, J. T., Adamczyk, P. D., and Bailey, B. P. 2006. "Djogger: a mobile dynamic music device", in *CHI '06: CHI '06 extended abstracts on Human factors in computing systems*, pages 556–561, New York, NY, USA. ACM Press.

Blesser, B. 2001. "An interdisciplinary integration of reverberation". *J. of the Audio Engineering Society,* 49(10):867–903.

Cadoz, C., A. Luciani, and J-L. Florens. 1983. "Responsive input devices and sound synthesis by simulation of instrumental mechanisms: the CORDIS system" *Computer Music Journal* 8:60–73.

Camurri, A. and G. Volpe, (eds.) 2003. "Gesture-Based Communication in Human-Computer Interaction", Springer LNAI2915, 2003.

Camurri A., Coletta P., Demurtas M., Peri M., Ricci A., Sagoleo R., Simonetti M., Varni G., and Volpe G. 2007. "A Platform for Real-Time Multimodal Processing", in *Proceedings Intl Conference Sound and Music Computing 2007 (SMC2007),* Lefkada, Greece, July 2007.

Camurri A., Canepa C., Coletta P., Mazzarino B., Volpe G. 2008. "Mappe per Affetti Erranti: a Multimodal System for Social Active Listening and Expressive Performance", in *Proceedings 2008 Intl. Conference on New Interfaces for Musical Expression (NIME-08),* Genova, Italy, June 2008.

Chamberlin, H., 1985. *Musical Applications of Microprocessors*, Hayden Books, 2nd edition.

Cho, S.-J. Oh, J. K. Bang, W.-C. Chang, W. Choi, E. Jing, Y. Cho, J. and Kim, D. Y. 2004. "Magic wand: A hand-drawn gesture input device in 3-d space with inertial sensors". In *IWFHR '04: Proceedings of the Ninth International Workshop on Frontiers in Handwriting Recognition*, pages 106–111, Washington, DC, USA. IEEE Computer Society.

Cho, S.-J., Choi, E., Bang, W.-C., Yang J., Sohn, J., Kim, D.Y., Lee, Y.-B., and Kim, S. 2006. "Two-stage Recognition of Raw Acceleration Signals for 3D-Gesture-Understanding Cell Phones", In *Tenth International Workshop on Frontiers in Handwriting Recognition* 2006.

Choi, E. S., Bang, W. C., Cho, S. J. Yang, J. Kim, D. Y. and Kim, S. R. 2005. “Beatbox music phone: gesture- based interactive mobile phone using a tri-axis accelerometer”. In *Proceedings of the International Conference on Industrial Technology*, p 97–102.

Chowning, J. M., 1973. The synthesis of complex audio spectra by means of frequency modulation, *Journal of the Audio Engineering Society*, 21(7):526–534.

Civolani, M., Fontana, F., 2008. A nonlinear digital model of the EMS VCS3 voltage-controlled filter, in: *Proc. Int. Conference on Digital Audio Effects*, Espoo, Finland, pp. 35–42.

Cook, P. and G. Scavone. 1999. “The Synthesis ToolKit (STK)”. In *Proceedings of the International Computer Music Conference*, Beijing, 1999.

Dattorro, J., 1997. Effect design, part 2: Delay line modulation and chorus, *Journal of the Audio Engineering Society*, 45(10):764–788.

Doidic, M., M. Mecca, M. Ryle, and C. Senffner. 1997. “Tube modeling programmable digital guitar amplification system.” U. S. Patent No. 5,789,689. Filed Jan. 17, 1997, issued Aug. 4, 1998.

Elliott, G. T. and B. Tomlinson, B. 2006. “Personalsoundtrack: context-aware playlists that adapt to user pace”, in *CHI Extended Abstracts*, p 736–741.

Essl, G. 2006. “Circle maps as a simple oscillators for complex behavior: II. experiments.” In *Proceedings of the International Conference on Digital Audio Effects (DAFx)*, Montreal, Canada, September 18-20, 2006.

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

Essl, G., Rohs, M. 2007. "ShaMus - A Sensor-Based Integrated Mobile Phone Instrument" in *Proceedings of the International Computer Music Conference (ICMC)*, Copenhagen.

Essl, G., Rohs, M. 2007. "The Design Space of Sensing-Based Interaction for Mobile Music Performance" In *Proceedings of the 3rd International Workshop on Pervasive Mobile Interaction Devices (PERMID),* Toronto, Ontario, Canada, May 13.

Fouad, H., Hahn, J., and Ballas, J. 1997. Perceptually based scheduling algorithms for real-time synthesis of complex sonic environments. In *Proc. of the 1997 International Conference on Auditory Display*, Xerox Palo Alto Research Center, Palo Alto, USA

Gardner W. G. 1997. “Reverberation algorithms”. In *Applications of Signal Processing to Audio and Acoustics*, ed. M. Kahrs, K. Brandenburg.

Gaye, L. 2003. “Sonic city: The urban environment as a musical interface”, *in Proceedings of New Interfaces for Musical Expression (NIME’03)*, pages 22–24.

Gaye, L., Holmquist, L. E., Behrendt, F., and Tanaka, A. 2006. ”Mobile music technology: Report on an emerging community”, *in Proceedings of New Interfaces for Musical Expression*, pages 22–25.

Geiger, G. 2003. “PDa: Real Time Signal Processing and Sound Generation on Handheld Devices”. In *Proceedings of the International Computer Music Conference*, Singapore, 2003.

Geiger, G. 2006. “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.

Hendrix, C., and Barfield, W. 1996. “The sense of presence within auditory virtual environments”. *Presence : Teleoperators and Virtual Environments* 5, 290-301.

Horner, A., Ayers, L., 1998. Modeling acoustic wind instruments with contiguous group synthesis. Journal of the Audio Engineering Society, 46(10):868–879.

Horner, A., Beauchamp, J., Haken, L. 1993. Methods for multiple wavetable synthesis of musical instrument tones, *Journal of the Audio Engineering Society*, 41(5):336–356.

Huopaniemi, J. and M. Karjalainen 1996. “HRTF filter design based on auditory criteria”. *Proc. Nordic Acoustical Meeting (NAM'96),* Helsinki.

Huovilainen, A., 2004. Non-linear digital implementation of the Moog ladder filter, in: *Proc. International Conference on Digital Audio Effects*, Naples, Italy, pages 61–64.

Jang, J. 2003. “Signal Processing of the Accelerometer for Gesture Awareness on Handheld Devices”, *Proceedings of the 2003 IEEE lnternational Workshop on Robot and Human Interactive Communication.*

Jot, J.-M., Wardle, S. and Larcher, V. 1998, “Approaches to binaural synthesis”. In *Proceedings of the 105th Audio Engineering Society Convention*, preprint 4861.

Jot J-M. and Chaigne A., 1991, “Digital delay networks for designing artificial reverberators”. *Proc. 90th Conv. Audio Eng. Soc*. (preprint no. 3030).

Jot J-M. 1999, “Real-time spatial processing of sounds for music, multimedia and interactive human-computer interfaces”. *Multimedia Systems*, 7(1) :55–69.

Karantonis, D. M., M.R. Narayanan, M. Mathie, N.H. Lovell, and B.G. Celler, 2006. “Implementation of a Real-Time Human Movement Classifier Using a Triaxial Accelerometer for Ambulatory Monitoring”. *IEEE Trans. Inform. Tech. in Biomed.* 10(1), Jan. 2006, pp. 156–167.

Karjalainen, M., T. Mäki-Patola, A. Kanerva, and A. Huovilainen. 2006. “Virtual Air Guitar.” *Journal of the Audio Engineering Society* 54(10):964–980.

Karjalainen, M. and J. O. Smith, 1996 “Body modeling techniques for string instrument synthesis.” *Proc. Int. Computer Music Conf. (Hong Kong, 1996)* pp. 232–239

Karjalainen, M., V. Välimäki, and T. Tolonen. 1998. “Plucked-String Models: From the Karplus-Strong Algorithm to Digital Waveguides and Beyond.” *Computer Music Journal* 22(3):17–32.

Karplus, K. and A. Strong .1983. “Digital synthesis of plucked-string and drum timbres.” *Computer Music Journal* 7:43–55

Klatt, D. 1980 “Software for a cascade/parallel formant synthesizer” *J. Acoust. Soc. Am.* 67:971–995

Kleimola, J. 2008. “Audio synthesis by bitwise logical modulation”. *Proc. Intl. Conf. on Digital Audio Effects (DAFx-08)*, pp. 67-70. Espoo, Finland, Sept. 1-4.

Kramer, G. 1989. “Digital signal processor for providing timbral change in arbitrary audio and dynamically controlled stored digital audio signals.” U. S. Patent No. 4,991,218. Filed Aug. 24, 1989, issued Feb. 5, 1991.

Laakso, T. I., Välimäki, V., Karjalainen, M., and Laine, U. K., 1996. Splitting the unit delay – Tools for fractional delay filter design, *IEEE Signal Processing Magazine*, 13(1):30–60.

Larsson, P., Västfjäll, D. and Kleiner, M. 2002. “Better presence and performance in virtual environments by improved binaural sound rendering”. In *Proceedings of the 22nd Audio Engineering Society Conference on Virtual, Synthetic and Entertainment Audio*, 31–38.

Lazzarini, V., Timoney, J. and Lysaght, T., 2008. The generation of natural-synthetic spectra by means of adaptive frequency modulation, *Computer Music Journal*, 32(2):9–22.

Le Brun, M. 1979. “Digital Waveshaping Synthesis.” *Journal of the Audio Engineering Society* 27(4):250–266.

Lokki, T., Pulkki, V. 2001. “Evaluation of geometry-based parametric auralization”. In *AES 22nd International Conference on Virtual, Synthetic and Entertainment Audio.*

Maher, R. C., 2005. Wavetable synthesis strategies for mobile devices, *Journal of the Audio Engineering Society*, 53(3):205–212.

Mäntyjärvi, J. Kela, J. Korpipää, P. and Kallio, S. 2004. “Enabling fast and effortless customisation in accelerometer based gesture interaction”, *in MUM ’04: Proceedings of the 3rd international conference on Mobile and ubiquitous multimedia*, pages 25–31, New York, NY, USA. ACM Press.

Min, P. and Funkhouser, T. 2000. “Priority-driven acoustic modeling for virtual environments”. *Computer Graphics Forum, Proc. of Eurographic* 2000, 19(3).

Moorer, J. A., 1976. The synthesis of complex audio spectra by means of discrete summation formulas, *Journal of the Audio Engineering Society*, 24(9):717–727.

Pakarinen, J., Puputti, T., and Välimäki, V. 2008. “Virtual Slide Guitar”. *Computer Music J*. 32(3):42-54.

Pakarinen, J. and D. T. Yeh. 2009. “A Review of Digital Techniques for Modeling Vacuum-Tube Guitar Amplifiers.” *Computer Music Journal*, 33(2):1-16. Summer 2009. Accepted for publication.

Paradiso, J.A. 1997. “Electronic Music: New Ways to Play.” *IEEE Spectrum* 34(12):18–30.

Pekonen, J. 2007. “Computationally Efficient Music Synthesis - Methods and Sound Design,” Master's thesis, Helsinki University of Technology.

Puckette, M. 1996. "Pure data: another integrated computer music environment". In *Proc. Int. Computer Music Conf.*, 1996, pp. 269–272.

Pylvänäinen, T. 2005. “Accelerometer based gesture recognition using continuous HMMS”, *Pattern Recognition and Image Analysis*, pages 639–646.

Roads, C., 1995. *The Computer Music Tutorial*, The MIT Press.

Rocchesso, D. and Smith, J.O. 1997. “Circulant and elliptic feedback delay networks for artificial reverberation”. *IEEE trans. Speech & Audio Processing*, 5(1).

Rodet, X., Depalle, P., 1992. Spectral envelopes and inverse FFT synthesis, in: *Proc. 93rd AES Convention*, San Francisco, CA, USA.

Rohs, M., G. Essl, and M. Roth. 2006. “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.

Rohs, M. and G. Essl, 2007. “CaMus2: collaborative music performance with mobile camera phones”. In *Proceedings of the international conference on Advances in computer entertainment technology*, June 13-15, 2007, Salzburg, Austria

Rossum, D., 1992. Making digital filters sound “analog”, in: *Proc. International Computer Music Conference*, San Jose, CA, USA, pages 30–33.

Serra, X., Smith, J. O., 1990. Spectral modeling synthesis: A sound analysis/synthesis system based on a deterministic plus stochastic decomposition, *Computer Music Journal*, 14(4):12-24.

Smith, J. O., 1982. An allpass approach to digital phasing and flanging. *Technical Report STANM-21*, Center for Computer Research in Music and Acoustics, Stanford University, CA, USA.

Smith, J. O. 1992. “Physical Modeling using Digital Waveguides.” *Computer Music Journal* 16(4):74–91.

Stautner, J. and Puckette, M. 1982. “Designing multi-channel reverberators”. *Computer Music J.*, 6(1)

Stilson, T., Smith, J. O., 1996. Alias-free digital synthesis of classic analog waveforms, in: *Proc. International Computer Music Conference*, Hong Kong, China, pages 332–335.

Strachan, S. 2007. “Multimodal, Embodied and Location-Aware Interaction”, PhD thesis, Hamilton Institute, National University of Ireland.

Strachan, S., P. Eslambolchilar, R. Murray-Smith, S. Hughes, and S. O'Modhrain. 2005. “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.

Tanaka, A. 2004. “Mobile Music Making”. In *Proceedings of the 2004 conference on New Interfaces for Musical Expression (NIME ’04),* pages 154–156, June 2004.

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

Tomisawa, N., 1981. Tone production method for an electronic musical instrument, *U.S. Patent 4,249,447*.

Touimi, A.B., Emerit, M., Pernaux, J.-M. 2004. “Efficient method for multiple compressed audio streams spatialization”. *Proc. of ACM 3rd International. Conference On Mobile and Ubiquitous multimedia.*

Tsingos, N., Funkhouser, T., Ngan, A., Carlbom, I. 2001. “Modeling acoustics in virtual environments using the uniform theory of diffraction”. *ACM Computer Graphics, SIGGRAPH’01 Proceedings*, pages 545–552.

Tsingos, N., Gallo, E., Drettakis, G., 2004. “Perceptual audio rendering of complex virtual environments”. *Proc. of ACM SIGGRAPH* 2004 pages 249–258.

Välimäki, V., J. Huopaniemi, M. Karjalainen, and Z. Jánosy. 1996. “Physical Modeling of Plucked String Instruments with Application to Real-Time Sound Synthesis.” *Journal of the Audio Engineering Society* 44(5):331–353.

Välimäki, V., Huovilainen, A., 2007. Antialiasing oscillators in subtractive synthesis, *IEEE Signal Processing Magazine*, 24(2):116–125.

Välimäki, V., J. Pakarinen, C. Erkut, and M. Karjalainen. 2006. ”Discrete-Time Modelling of Musical Instruments.” *Reports on Progress in Physics* 69(1):1-78.

Verplank, B., Mathews, M., Shaw, R., 2000. Scanned synthesis, in: *Proc. International Computer Music Conference*, Berlin, Germany, pp. 368-371.

Wanderley, M., and P. Depalle. 2004. “Gestural Control of Sound Synthesis.” *Proceedings of the IEEE* 92(4):632-644.

Wang, G., G. Essl, and H. Penttinen, “MoPho: Do Mobile Phones Dream of Electric Orchestras?”, *International Computer Music Conference (ICMC 2008),* Belfast, August 24 - 29, 2008.

Wightman, F., and Kistler, D. 2005. "Measurement and validation of human HRTFs for use in hearing research". *Acustica, Acta-Acustica* 91, 429–439.

Winham, G., Steiglitz, K., 1970. Input generators for digital sound synthesis, *Journal of the Acoustical Society of America*, 47(2, part 2):665–666.

Yuen, J., Horner, A., 1997. Hybrid sampling-wavetable synthesis with genetic algorithms, *Journal of the Audio Engineering Society*, 45(5):316–330.

Zölzer, U. (ed.), 2002. "DAFx – Digital Audio Effects." John Wiley Sons Ltd. ISBN: 0-471-49078-4.