Virtual Air Guitar*

MATTI KARJALAINEN,[1] *AES Fellow,* **TEEMU MÄKI-PATOLA,[2] AKI KANERVA,[2] AND**
(matti.karjalainen@tkk.fi) (tmakipat@tml.hut.fi) (aki.kanerva@iki.fi)

ANTTI HUOVILAINEN[1]
(ajhuovil@acoustics.hut.fi)

[1]*Helsinki University of Technology, Laboratory of Acoustics and Audio Signal Processing, Espoo, Finland*
[2]*Helsinki University of Technology, Telecommunications Software and Multimedia Laboratory, Espoo, Finland*

A combination of handheld controllers and a guitar synthesizer is called "virtual air guitar" (VAG). The name refers to playing an "air" guitar, that is, just acting the playing with music playback, and the term virtual refers to making a playable synthetic instrument. Sensing of the left-to-right-hand distance is used for pitch control, the right-hand movements are used for plucking, and in advanced versions of the VAG the finger positions of both hands can be used for other features of sound production. Three different hand gesture controllers are discussed. The sound synthesis algorithm simulates the electric guitar, augmented with sound effects such as tube amplifier distortion, as well as intelligent mapping from playing gestures to synthesis parameters. The realization of the virtual instrument is described, and sound demonstrations are available on a Web site.

0 INTRODUCTION

Synthesis of musical instruments combined with novel interface technology make interesting new virtual instruments. For sound synthesis, physical modeling techniques are particularly attractive due to their flexible parametric control abilities; see, for example, [1], [2]. A good introduction to existing gesturally controlled and augmented musical interfaces can be found, for example, in Paradiso [3] and Wanderley and Battier [4]; see also [5]–[7]. Many prototypes have been created (for some examples see [8]–[10]) and even a few commercial controllers exist, such as [11], [12], and Yamaha's Miburi system of the mid-1990s.

Novel interaction techniques, such as magnetic tracking, acceleration sensors, and computer vision, let us extract expressive control data from the user. Thanks to the increase in both processing power and research attention, computer vision in particular is becoming more and more popular in user interfaces [13]–[15]. One of its advantages is the lack of wires and control devices, allowing the performer complete freedom of expression.

In this engineering report we describe the development of a virtual instrument that we call "virtual air guitar" (VAG). The idea was to combine "air guitar playing," that is, acting the playing of an imaginary instrument along with music playback [16], [17], and a guitar synthesizer including sound effects, thus making a virtual instrument, playable with sensors that follow hand movements.

There were two main motivations to the present study: (1) to investigate and realize expressiveness of synthetic guitar playing with hand-following sensors and (2) to design an easy-to-play attraction for a science center exhibition.

The entire VAG is composed of several subsystems: (1) a guitar synthesizer with sound effects and audio reproduction, (2) a user interface consisting of handheld sensors (as well as possible foot controllers), and (3) software to map user interface signals to expressive playing of the synthetic guitar. While some of the hardware and software components are available off the shelf, we wanted to explore many of the subsystems more freely with new innovations.

The guitar synthesizer is realized as an extended Karplus–Strong type synthesis algorithm [18], [19]. For the electric guitar it is fairly easy to calibrate the algorithm to yield a desired sustain and spectral properties. For good attack a wavetable can be sampled through inverse filtering of a real played sound by the calibrated string model. For good acoustic guitar sound the calibration of the model is more involved because a body model is also required.

Audio effects are needed for realistic electric guitar sounds [20]. Reverberation through room simulation also

*Manuscript received 2006 February 24; revised 2006 July 13, August 28, and September 6.

is useful for the acoustic guitar. The electric version of the VAG typically includes a distortion unit, a delay unit, and a reverberation unit as software algorithms integrated with the guitar synthesizer.

For the user interface we have experimented with three approaches: (1) using data gloves and position tracking in a cave-like virtual room with multiwall stereo-vision video projection and multichannel sound reproduction, (2) using small-sized handheld control devices, and (3) applying hand tracking by video image analysis. The two main control parameters required in all cases are the left-to-right-hand distance (or left-hand position relative to the player's body) for pitch control and the right-hand movement sensing for plucking of the strings.

For the handheld control devices the pitch control is realized by measuring the propagation delay of high audio frequency pulses from a small loudspeaker to an electret microphone, both inside handheld control sticks. The right-hand plucking movement is measured by an acceleration sensor. Hand tracking by video analysis tracks the user hands through a common Web camera (webcam). The locations of the user's hands are extracted from the animation frames and fed into a gesture extractor to detect when certain playing gestures take place.

In addition to pitch and plucking controls, it is desirable to have control over other usual guitar-playing techniques such as vibrato, slide, muting, pulloff, hammer-on, and selection of strings and chords. The virtual room and webcam interfaces support also these techniques through gesture extraction and suitable control mappings. A musical intelligence layer is used to map simple hand movements to techniques mimicking qualified musicians. The implementation details of all these approaches are described here.

Both the webcam and the control stick versions were placed on display at the Heureka Science Centre in Finland in 2005 March. The webcam version became the most popular attraction of the music-related exhibition, being played over 60 000 times during the one year of the exhibition. It has also aroused international media attention, including numerous television shows, radio programs, popular magazine articles, and online articles.

1 THE GUITAR MODEL

In this section we describe the synthetic guitar model, particularly for a Stratocaster type electric guitar, as well as sound effects processing.

1.1 Electric Guitar Synthesizer

Physics-based models for real-time synthesis of the guitar have been studied actively; see, for example, [18], [19], [21]–[29]. Many popular models are based on the extended Karplus–Strong (EKS) concept, see [18], [19] in particular. It is a computationally efficient digital waveguide algorithm for modeling the guitar string as a single-delay loop filter structure with parametric control of the fundamental frequency and losses in the filter loop.

Fig. 1 depicts the basics of this model structure. It is analyzed in detail in [19]. Therefore we present here only a brief overview. A wavetable (often a set of selectable wavetables) stores an excitation waveform that may be a synthetic signal or extracted from a real recorded guitar pluck by inverse filtering with the EKS filter structure [25]. A timbre control filter, shaping the damping versus boosting of high frequencies, follows the wavetable. The signal is further passed through a pluck position control. It implements a comb filter effect that in a real string results from the standing-wave structure of vibration and depends on the point of the string that is plucked. The signal is then fed through a single-delay feedback loop, which is the most fundamental part of the string model, to simulate the exponentially decaying vibration of the plucked string. Finally the signal is integrated so that the output is approximately proportional to the force acting to move the bridge, when the excitation has been an impulsive acceleration signal at the plucking point [19].

In the electric guitar, which is the case discussed in this engineering report, the signal is captured by a magnetic pickup [30]. For example, in a Stratocaster type solid-body model there is a selector for three pickups: bridge, middle, and neck pickups. The magnetic pickup is sensitive to the string velocity in the vertical direction (normal to the top plate of the guitar), and it introduces another comb filtering effect in a way similar to the plucking point filtering. Fig. 2 depicts a filter model for the pickup [19]. Delay is the time of wave propagation from the pickup to the bridge and back. Block LP is an optional filter to adjust the comb filtering details. Block $P_o(z)$ implements the spectral shaping characteristics due to electrical *RLC*-type low-pass filtering and the finite effective width of the pickup sensing of the string vibration. Filter $I(z)$ is an integrator to yield the string velocity.

Fig. 3 shows a full synthesis model for one dual-polarization guitar string [19]. The model is developed primarily for the acoustic guitar, but is applicable to the

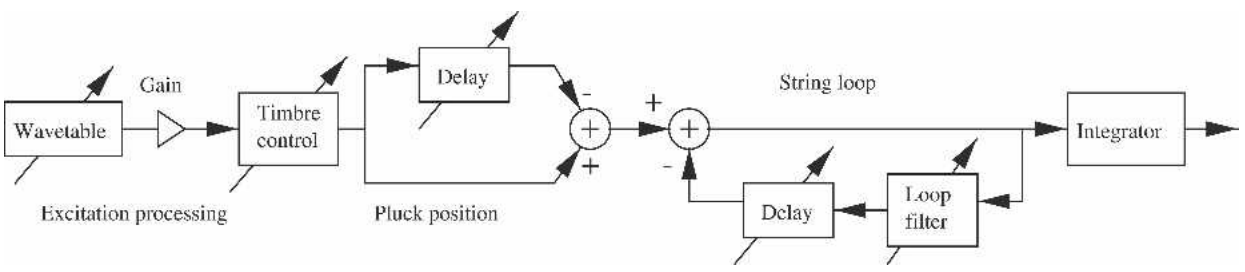

Fig. 1. Basic structure of single-delay loop filter and excitation mechanism for implementing extended Karplus–Strong string model.

electric one when properly interpreted. The wavetables store excitation waveforms that capture the plucking and the body effects. While in an acoustic guitar the body response can be commuted to the plucking wavetable [31], [32], in the solid-body electric guitar the body effect is of minor importance only. Pluck shaping and plucking point filters are as discussed before. Sympathetic couplings between strings play a lesser role in the solid-body electric guitar, so they can be omitted.

The magnetic pickups in the electric guitar are sensitive to the vertical movements of the string only [30]. Yet a dual-polarization string model is useful for simulating the slight beating effects found in the partials of string vibration. When the string is plucked, there is a combination of vibration components of both polarizations [33]. The polarization state will change, however, in the course of time so that the pickup output exhibits some beating in the envelope of each partial. This phenomenon can be simulated by summing dual-polarization model components [34], [19], although this may not always be a physically correct way of realizing the beating. Since the beating in the electric guitar sound is typically quite weak, it is not a critical, although a desirable feature in sound synthesis.

1.2 Calibration of Guitar Model

In this engineering report we discuss the modeling and synthesis of a solid-body guitar, a case study applied to a hand-made copy of the Fender Stratocaster. A fairly accurate synthesis model was achieved using the procedure described hereafter. Each string is calibrated separately in the following way.

- The string is plucked about 10 mm from the bridge in the vertical direction by a hard sharp pick. This approximates an ideal pluck so that the comb filtering due to the plucking point effect is moved to high enough frequencies. The response is recorded from the bridge pickup with the tone and volume control potentiometers turned clockwise.
- The decay time constants of the 10–20 lowest partials of the recorded signal are analyzed. This can be done using procedures proposed, for example, in [25], [35]. Here we band-pass-filtered each partial and fitted a line to a dB-scaled decay curve for fret 1 and 12 fingerings. The period and depth of beating in the envelope of each partial were also approximated if prominent.
- A first-order low-pass loop filter (Fig. 1) with two control parameters, dc gain, and a high-frequency control parameter is fitted to the string model in the two cases of fingerings, for example, by the algorithm proposed in [25]. Due to relatively regular behavior in the electric guitar, the loop filter parameters were adjusted manually, instead of an automatic procedure, to give proper decay times. For other fingering positions the loop filter parameters are interpolated linearly at run time according to the fret position. In practice we have divided the loop filter into two cascaded filters, one for nominal damping values inherent in the string as a function of the fretting position and another for controlling extra damping due to right- or left-hand operations.
- Approximately correct values are set for the polarization submodel string-length difference for the proper beating rate and mixing ratio for the beating depth. Because these effects need to be approximated only roughly, the parameters are adjusted manually to yield a perceptually good result. These parameters are also interpolated according to the fretting position.
- Fifth-order Lagrange interpolation is used for loop delay control [36] to obtain click-free variations of the string length and a good approximation of a unity-magnitude response up to about 10 kHz (the sampling rate is 44.1 kHz) in order not to affect the loop gain.
- The single-coil Stratocaster pickup model used here is a low-pass plus comb filter type digital filter. It is designed to match the pickup response spectrum recorded previously. Fig. 4 compares a measured spectrum and a single-coil pickup response spectrum when the pickup used was modeled as a cascade of two fourth-order But-

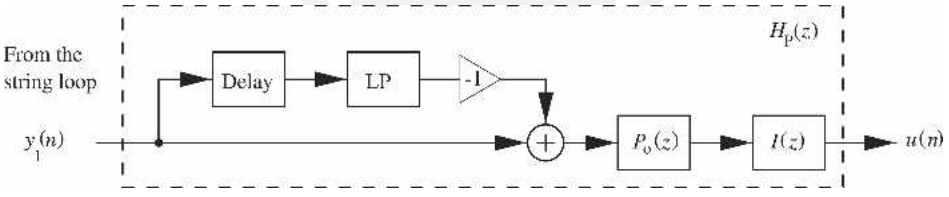

Fig. 2. Filter model for magnetic pickup in an electric guitar.

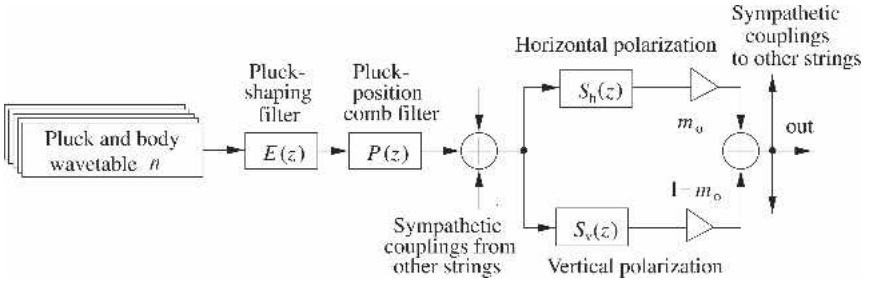

Fig. 3. Dual-polarization string model.

terworth low-pass filters with a cutoff frequency of about 10 kHz plus the comb filtering. Notice that a specific pickup response depends on many factors, such as pickup type, electric control circuitry and settings, electric loading, and magnet pole distance from the string [30].
- The calibration procedure described yields the proper sound for "ideal" plucking if an impulse is used as the excitation wavetable to the model. Better sounding attacks can be obtained if real instrument recordings are inverse filtered by the string and pickup models, and the residual signal is properly truncated and used as a wavetable. Different plucking and excitation sounds can be stored in different wavetables, and the pluck shaping filter can be used for fine-tuning the timbre.

The Stratocaster guitar synthesizer consists of six strings, each calibrated separately, with controls for pitch (string length), plucking point, pickup position, plucking wavetable selection and triggering, and plucking spectral shaping for smoother or sharper attack. The real-time synthesis model is implemented as a patch in the BlockCompiler [37], and it can be exported to different other software platforms such as pd (PureData) [38] and Mustajuuri [39].

1.3 Guitar Effects

The solid-body electric guitar without any sound effects in general is perceived quite dull in timbre. Sullivan [22] demonstrated that for a simple synthetic guitar model the addition of distortion and feedback can make relatively convincing classic distortion guitar effects.

In rock guitar playing the sound of the guitar is processed with a variety of effects to obtain the final sound [20]. In particular, the recorded sound of the guitar is typically fed through a tube amplifier. When driven at high volume, this amplifier and the loudspeaker introduce nonlinear distortion and overtones to the sound, which is pleasing to the human ear. Because the VAG is designed primarily for rock music, the system must simulate a typical rock guitar setup as well as the sound of the guitar itself.

A typical setup consists of a compressor, a preamplifier, a power amplifier, loudspeaker cabinets, an equalizer, a reverberation unit, and possibly a delay unit. While we can use any external effect devices and amplifiers with our VAG, we wanted to combine all elements in software that runs on a single computer. In the Mustajuuri audio software we have used the CAPS plugin suite [40], a collection of open-source audio plugins using the LADSPA, a common application programming interface for audio processors on the Linux platform.

In this study we paid special attention to the distortion effect produced in tube preamplifiers and developed a simulation model that runs on the BlockCompiler software. We describe this tube distortion simulation in detail next. (Recently we made a similar distortion simulator using wave digital filter modeling [41].) Other effects used in the VAG versions are not discussed here.

1.3.1 Tube Amplifier Stage

The circuit diagram for a typical tube preamplifier stage is shown in Fig. 5. The voltage V_{gk} between grid and cathode as well as the voltage V_{pk} between plate and cathode determine the plate current. The plate resistor R_p converts this current to voltage. The capacitor C_o acts as a high-pass filter to block the dc voltage from the next-stage input. The resistor R_i prevents parasitic high-frequency oscillations and limits the input current to the grid of the tube when V_{gk} approaches or exceeds zero voltage. Resis-

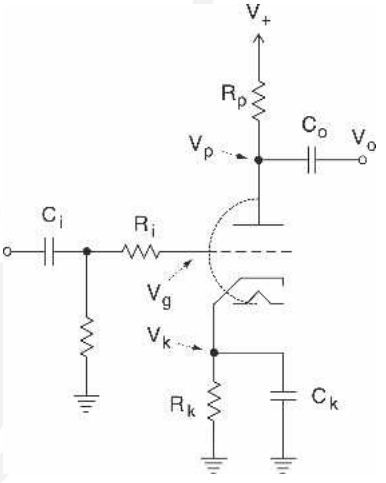

Fig. 5. Tube gain stage.

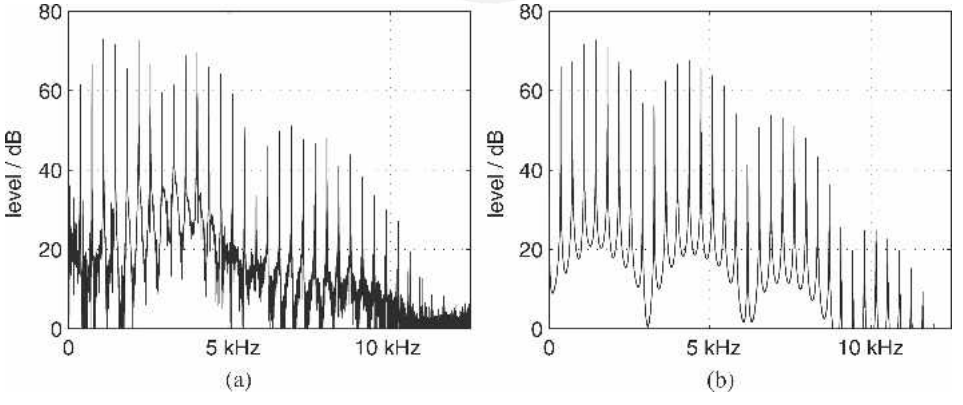

Fig. 4. Comparison of spectrum responses for string 1, fret 1, bridge pickup, plucked 10 mm from bridge. Model is excited by an impulse. (a) measured, (b) modeled.

tor R_k biases the cathode positive compared to the grid (which is normally at ground potential). The capacitor C_k acts as a low-pass filter to keep the signal current from affecting the cathode voltage. For more information about triode amplification stages, see [42], [43].

Because the V_{gk}, V_{pk} to I_p transfer function of an electron tube is nonlinear and asymmetric, the cathode voltage changes somewhat depending on the average signal level. It produces dynamically varying distortion on note attacks as the bias shifts. The authors feel that this is one of the reasons why many guitarists describe tube amplifiers as sounding more "dynamic" or "lively." This effect was therefore deemed desirable to be duplicated.

Another source of time variance in the circuit is the input capacitor C_i (the same as the output capacitor C_o of the previous stage). Because the grid current rises rapidly as the grid-to-cathode voltage approaches zero, the normally constant voltage over C_i is changed. This nonlinearity is only to one direction and causes a shift in the input dc voltage, biasing it negatively. If the biasing is severe enough, it will cause the stage to go into cutoff, making the signal cut in and out—an effect known as blocking distortion [44]. Blocking distortion can be avoided by increasing the input resistor R_i to limit the amount of grid current.

1.3.2 Tube Amplifier Stage Model

Accurate modeling of the amplifier stage would require solving a system of nonlinear differential equations, which is problematic and CPU intensive. However, if the plate load is assumed to be constant and resistive, then the plate current and voltage are coupled and a curve $V_p(V_{gk})$ can be calculated. This curve depends only on the grid-to-cathode voltage V_{gk}, the power supply voltage V_+, the plate resistor R_p, and of course the tube type. If V_+ and R_p are held constant, the curve can be measured directly by shorting the cathode in Fig. 5 to ground and varying V_{gk}. A curve for tube 12AX7 with $V_+ = 250$ V and $R_p = 100$ kΩ is shown in Fig. 6.

A small error still remains because V_k changes with the input level. This can be safely ignored as the variation in V_k (hundreds of millivolts) is very small compared to V_p variations (approximately 200 V peak to peak). Note that this does not mean that V_k variations can be completely ignored as the input level is of a similar order of magnitude.

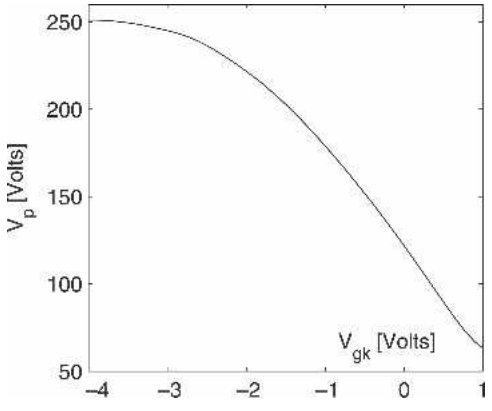

Fig. 6. Plate voltage V_p as a function of grid-to-cathode voltage V_{gk} for tube 12AX7 with $V_+ = 250$ V, $R_p = 100$ kΩ.

The V_{gk} to I_{gk} curve can be measured in a way similar to the plate voltage curve. It was found to be approximately exponential in shape. Fitting a curve to the measured data gave

$$I_{gk} = e^{7.75V_{gk}-10.3}. \tag{1}$$

In simulation the V_i to V_{pk} relationship is more useful than the V_{gk} to V_{pk} relationship. If the input impedance is assumed to be constant and resistive, this means solving the implicit equation

$$\frac{V_i - V_g}{R_i} = I_{gk}(V_{gk}) \tag{2}$$

and then computing

$$V_p = V_{pk}(V_{gk}). \tag{3}$$

These can be combined into a single function F_{tube} (V_{gk}, R_i) for use in a digital model. This function can be precalculated to avoid solving an implicit equation at run time.

1.3.3 Digital Tube Stage Model

With the further assumption that the input dc blocking capacitor C_i can be replaced by a normal first-order high-pass filter (a reasonable assumption since in amplifier design blocking distortion is usually avoided as much as possible), Fig. 7 now shows a digital model of the tube amplifier stage.

The LPF_{in} block models the low-pass filtering caused by R_i and tube Miller capacitance [45]. The cutoff frequency is $1/[(2\pi R_i C_{pg}(1 + G_{tube})]$, where C_{pg} is the plate-to-grid parasitic capacitance (typically around 1.7 pF [46]) and G_{tube} is the gain of the tube stage (around 60).

HPF_o is the interstage dc blocking filter. Its cutoff frequency depends on the output capacitor value and the next-stage input resistance.

The gain block R_k/R_p provides the biasing for the tube, whereas LPF_k models the low-pass filtering caused by C_k. A unit delay is inserted between the output of LPF_k and the tube input to make the model realizable. The extraneous phase shift caused by the unit delay is not a problem as the LPF_k cutoff frequency is very low (tens to hundreds of hertz) and gain at high frequencies is low.

1.3.4 Complete Amplifier Model

The model for a complete guitar amplifier consists of three gain stages followed by an equalizer and a cabinet

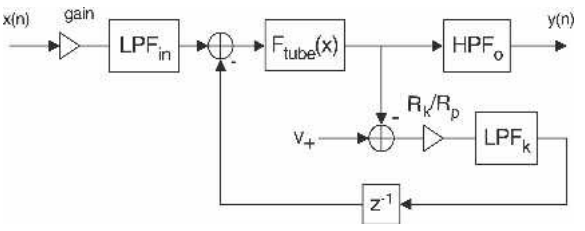

Fig. 7. Digital model of tube amplifier stage.

simulator. The power amplifier is not simulated. Table 1 shows the parameters that are common for all stages, and Table 2 shows individual stage parameters. These parameters were derived from an AX-84 high-octane guitar amplifier preamplifier circuit [47].

Guitar amplifier equalizers are in general purely passive and the controls have complex interactions. However, the resulting frequency response is usually more or less a V-shape curve. A parametric midcut equalizer can be used to get acceptable results.

Guitar amplifiers are practically always used with dedicated guitar loudspeakers. Their response is often far from flat and the response shape strongly colors the signal. Certain amplifiers and music styles are often associated with specific loudspeakers and cabinets (for example, the 4 × 12-in Marshall stack is almost de facto standard in some rock styles). Furthermore, the fast rolloff in the response above 5 kHz attenuates the harsh sounding high frequencies that will result from distortion. Fig. 8 shows the frequency response of a typical guitar loudspeaker.

A simple solution for loudspeaker simulation is to convolve the signal with a recorded impulse response. While producing very good results, this technique is computationally too expensive to be feasible. Therefore the loudspeaker is often simulated with an IIR filter bank that mimics the peaks and notches. To replicate the response accurately, a large number of filters is required, which is again computationally intensive. Furthermore, matching a set of IIR filters to a certain response is not trivial. We have therefore opted to use a minimum-phase reconstruction of a recorded impulse response. This has the advantage that any loudspeaker and cabinet combination can be simulated easily. The method loses time and phase information, but because loudspeakers are often close-miked, it is unlikely to be a problem. Informal listening has shown that using a 200-tap minimum-phase FIR filter (at a 44.1-kHz sampling rate) is indistinguishable from the original close-miked impulse response for typical guitar loudspeakers.

Table 1. Parameters common for all stages.

V_+	250 V
R_p	100 kΩ
HPF_o	31 Hz

Table 2. Parameters for amplifier stage.

Stage	R_i/(kΩ)	LPF_{in}/(Hz)	LPF_k/(Hz)	R_k/(Ω)
1	68	22 570	86	2700
2	250	6531	132	1500
3	250	6531	194	820

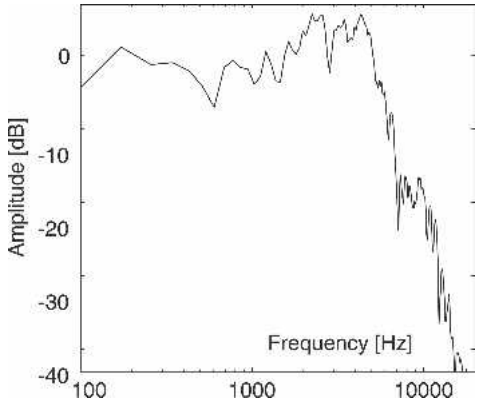

Fig. 8. Frequency response of Celestion Vintage 30 loudspeaker in 4 × 12-in cabinet.

2 USER INTERFACES

Playing air guitar is like playing rock guitar, only without an actual instrument or musical skills. The goal of the VAG was to replicate the mood of the experience but also to control the guitar sounds instead of just acting along background music. The intensity of the user's physical gestures was to be transferred to intensity in the sound and yet produce a pleasant rock guitar sound experience. It follows, then, that players do not want an intricate and expressive instrument that requires years to learn to play, but something that allows them to rock on with their favourite music. Thus the VAG is more of an entertainment device than a professional musical instrument.

The air guitar is by definition a nonphysical instrument. Thus its interfaces try to minimize the obstruction the controller sets upon the moving of the performer. This is best attained in the webcam interface that tracks the movement of just the user's empty hands in the air.

The physical parameters of the extended Karplus–Strong sound synthesis model are not intuitive even for a real guitarist, and much less so for an air guitarist. The gestures of the user need to be mapped to the control parameters of the sound model in a way that feels intuitive to the user. Most people will not even read short directions for an installation in a large exhibition. Thus our aim was that the user could discover the playing techniques based solely on his mental picture of rock guitar playing.

We experimented with several levels of support for the playing. Fig. 9 shows alternative paths how the input data

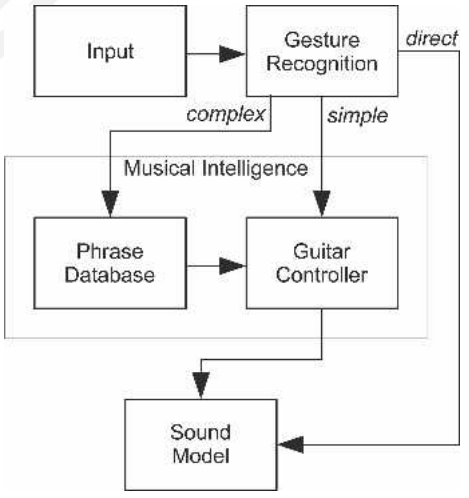

Fig. 9. Three levels of mapping of user input to sound model parameters in VAG user interface.

from the user are mapped to control data for the sound model. First the user's hand movements (and finger movements, depending on the implementation) are tracked by the input module. It feeds these data into a gesture recognition module, which detects the supported playing gestures from it.

At the simplest, the recognized gestures control the sound module directly, for example, by continuously altering the pitch according to the distance between the user's hands and plucking the string when the user makes a plucking motion. On a more intelligent mapping different levels of musical intelligence can be applied. The gestures can control the sound model through a guitar controller or with an additional level triggering musical phrases from a phrase database. The database contains songs as scripts, telling which actions are performed on the guitar, in which order, and at what time to pass them on to the guitar controller. In both cases the controller receives guitar playing actions and converts them into control data that the sound model understands. The controller keeps track of the state of the guitar. It knows which strings are playing, on which fret they are pressed, and so on. Some actions, such as a pulloff, depend of the current state of the guitar. When the controller is told to do a pulloff, it knows which string is playing and on which fret, and it triggers the same string more silently from a lower fret.

2.1 Input Hardware Options

We developed three different interfaces for the VAG. Each uses quite different input hardware. Two were designed for the science center exhibition. Initially the goal was to test which interface is best suited for the exhibition and install only that, but in the end both were installed into the same booth so visitors could jam together. For testing and developing different interaction approaches we used the virtual reality system that accompanies data gloves and a magnetic motion tracker. The virtual reality system offers the most expressive user input, but it is obviously difficult to port outside the laboratory. The three hardware options are discussed in Subsections 2.2–2.4.

2.2 Magnetic Tracker and Data Gloves

In addition to offering expressive input with several degrees of freedom, the virtual reality system promotes quick testing of new interaction methods. We can easily experiment new mappings and combinations of musical intelligence and gesture recognition with it.

The virtual reality system that we use is a cave-like virtual room, called EVE [48]. Fig. 10 shows a snapshot of playing the VAG in the virtual room. The virtual reality interface hardware for the VAG consists of two 5DT data gloves combined with a MotionStar magnetic sensor attached to both gloves. The location and orientation of the data gloves (magnetic sensors) are read at a rate of 100 Hz by a MotionStar magnetic tracker. The location accuracy of the tracker is on the order of one centimeter; the orientation accuracy a few degrees. The 5DT data gloves are simple and inexpensive. They measure the finger flexure through the bending of optical fibers and return one integer value for each finger. The control gestures of the virtual room version are described in Section 3.1.

The input data from the tracker and the gloves are passed to an SGI Onyx 2 infinite reality main frame running the virtual room. The Onyx 2 has four graphics pipelines, two of which are used to produce stereographic three-dimensional graphics on the three walls and the floor of EVE. The graphics are viewed through Chrystal Eyes shutter glasses to produce a realistic three-dimensional imagery of the virtual world. In the case of the air guitar there is currently no other visualization than a moody background scenery. We have also planned an interactive crowd simulation of an audience behavior.

2.3 Optical Tracking of Hand Movement

Using optical tracking for control parameter extraction consists of capturing video data of the user and processing the data on the fly to extract the desired features. Every frame of the video is a single image of the user at a specific moment in time. Computer vision algorithms are used to analyze the individual images to extract information such as the locations of the user's hands. The webcam interface contains also a foot pedal for changing between different play modes.

For a motion-intensive optical tracking system it is desirable to use a camera with a high update rate. This ensures an accurate control flow and prevents aliasing effects such as missing fast movements. Also, with low update rates the light is accumulated for a longer time, radically blurring fast movements. For instance, a hand may become a large blurry sweep, making the image analysis much more difficult.

Most Web cameras capture only 30 frames per second, although some new cameras offer 60 frames per second. Of course special cameras offer much higher rates. However, for the continuation of the webcam air guitar support we wished to build the system from parts that even a

Fig. 10. Soulful playing of VAG in virtual room, EVE.

normal home computer user could obtain at an affordable price.

We ended up using Apple's iSight webcam. It has good image quality but a relatively high latency. However, our user tests showed that the latency was not noticed by non-expert users. For example, the IBot Pro webcam (Orange) has a shorter latency but also contains much more noise in the picture. The iSight camera captures 30 frames per second with a 640*480 pixel resolution. The USB webcam ToUcam PRO II (Philips) takes up to 60 frames per second. Because of problems with Linux camera drivers we had to use a FireWire webcam in the current version.

We ended up making the optical hand tracking simple, since the approach was also found to be robust. The hands of the user are extracted from the raw pixel data based on their color (see Fig. 11). The user wears orange gardening gloves. The locations of the gloves are extracted by finding the two largest blobs of orange color from each camera frame in an YUV color space, which separates the color information from the intensity. Thus one can search for a color, independent of its intensity, which makes finding the gloves as independent of lighting as possible.

Camera frames are searched for color values inside a predefined range for orange. Pixels falling in between threshold values are marked, and each connected area forms a blob object. The approach is similar to filling bound areas with a certain color in computer painting programs. Then the X/Y locations of the centers of these areas (blobs) are evaluated by counting the center of mass of each blob. The two biggest blobs of orange color are treated as the user's hands. Thus even lots of falsely classified pixels do not harm the process. The center of the leftmost big blob is treated as the location of the left hand, and similarly that of the right one for the right hand (see Fig. 12).

Colored gloves were used instead of stickers or hand-held colored props because they are visible no matter how the hand is turned. In order to keep the image analysis as

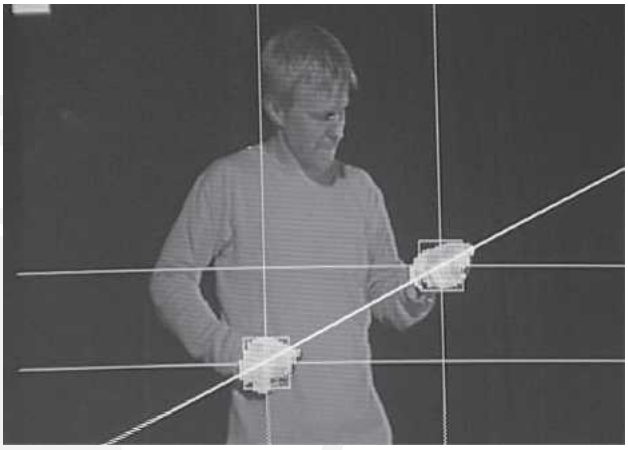

Fig. 12. Camera tracking of hand positions (orange gloves), as seen by computer.

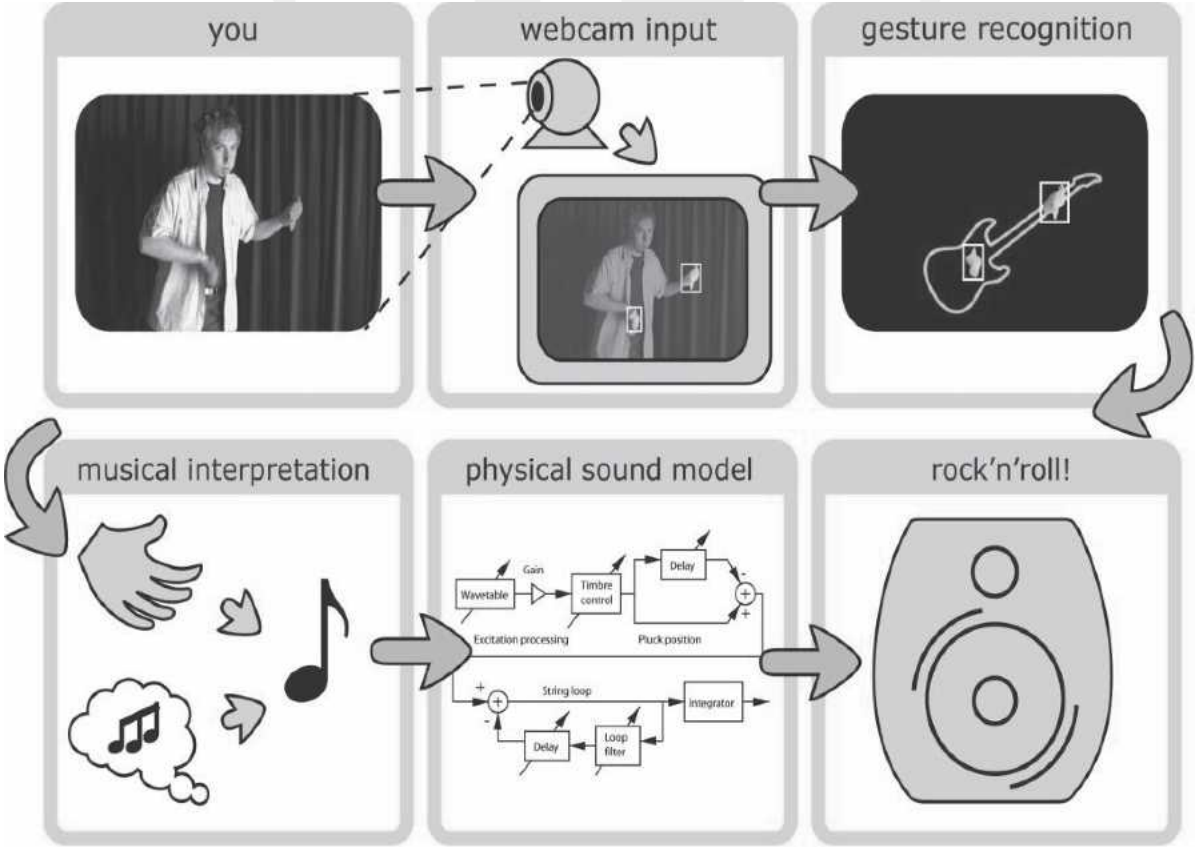

Fig. 11. While playing webcam version, user wears colored gloves. Gloves are found from webcam signal based on color. Hand locations are fed into a gesture recognition module that detects different playing gestures. Triggered gestures control a physical electric guitar sound model through a context-aware musical intelligence.

fast as possible we did not use an existing computer vision library such as PureData [38] or EyesWeb [49]. For a more complex image analysis task the tools these libraries offer would certainly be of use, but we did not really need them, and we wanted to keep the extra libraries and processing overheads as low as possible. Thus we implemented the capture of camera data and the simple blob tracking ourselves. As an end result the image analysis, gesture extraction, Mustajuuri DSP software running the sound effects, guitar control plugin, and the six physical string models producing the initial guitar sound run fast enough on a single 2.8-GHz Pentium 4 computer.

2.4 Special Control Devices

While advanced but expensive data gloves provide the highest control over hand gestures, and video tracking of hands is the closest to nothing at hands, specialized control sticks allow for a variety of low-cost possibilities to hand and finger controls. In this study we developed simple sticks for obtaining the two basic control parameters—pitch and pluck—see Fig. 13.

2.4.1 Acoustic Positioning of Player's Hands

There are two conceptually different approaches to the wireless measurement of the distance of two small sensors: electromagnetics or acoustic waves. Electromagnetics is successfully applied, for example, in the musical instrument Theremin, where the distance is measured by sensing the capacitance between an antenna and the hand of the player. Another approach with electromagnetics would be to measure the phase difference between sent and received pulses. With these approaches the measurement could be done without acoustic interference, no matter how loud a band would surround the player of the VAG. A drawback is the need of RF electronics in the implementation.

Acoustic measurement of the distance based on the propagation delay of sound has the advantage that it can be implemented with merely a loudspeaker and a microphone (and possible amplifiers) connected to a computer, leaving the signal processing to DSP. This is attractive since the VAG in any case has an audio interface and DSP available. The use of sound, however, leaves the measurement to be done in a noisy environment, especially if there are, for example, percussionists next to the VAG.

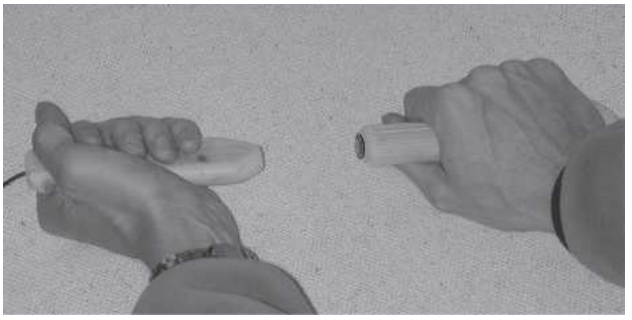

Fig. 13. Prototype control sticks for playing VAG. Right-hand stick includes an acceleration sensor as well as a small headphone driver to send distance measurement pulse. Left-hand stick receives pulse by an electret microphone.

In order to keep electronic and audio equipment minimal and simple, the measurement of the propagation delay can be done using audio frequencies. To achieve an adequate signal-to-noise ratio the sound pressure level needs to be high enough, thus making the measurement sound signal potentially audible. This is prevented with the use of only the highest audio frequencies. If the common 44.1-kHz sampling rate is used, the applicable frequency range in the measurement is limited to about 18–20 kHz. The lower limit is for keeping a relatively strong signal, which is needed, inaudible, and the higher limit is due to the antialiasing filters of the analog-to-digital and digital-to-analog converters.

The desired operating range for the distance measurement is 0.1–1.0 m, which corresponds to 0.3–3.0 ms of time at the speed of sound. With the sampling frequency of 44.1 kHz the range translates to about 13–130 samples. While one sample interval corresponds to 7.5 mm and the minimum guitar fret spacing is about 10 mm, subsample accuracy of the distance measurement is highly desirable.

2.4.2 Time-Delay Estimation

Time-delay estimation (TDE) is a common signal processing task with many proposed algorithms [50]. For example, the maximum-likelihood estimate of the delay between a signal and its delayed replica embedded in noise is the argument that maximizes a properly weighted cross-correlation function. In a digital implementation, if the ratio of the sampling frequency to the signal frequency is high enough, this yields an accuracy of ±0.5 sampling interval. Subsample accuracy can be obtained with simple interpolation, for example, by parabolic fit. In our case, however, the measurement signal is almost critically sampled and the resulting sampled cross-correlation function does not in general attain its peak in the vicinity of the underlying continuous function, and a relatively complex interpolation algorithm would have been required. The phenomenon is described in [51].

2.4.3 Envelope Follower Algorithm

To avoid the computational cost of cross correlation, we devised a simpler algorithm, which seeks the peak of the received signal envelope of a modulated pulse. The block diagram of the envelope follower type of TDE is illustrated in Fig. 14. A bandpass-filtered impulse is sent as a measurement signal, and the received signal is bandpass filtered again, rectified (absolute value), and low-pass fil-

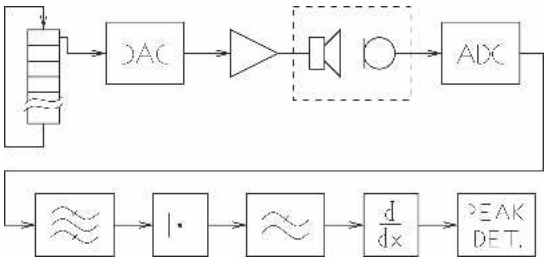

Fig. 14. Diagram of time-delay estimation using envelope follower algorithm.

tered, yielding a smooth pulse envelope. The peak of the envelope is found by searching for the first zero crossing of its derivative after the maximum point. The estimate is further refined for subsample accuracy by interpolating the zero-crossing position by linear interpolation; see Fig. 15.

Through experiments we found that a small off-the-shelve dynamic headphone loudspeaker and an electret microphone connected to a computer soundcard yield a good enough signal-to-noise ratio for our purposes. If only speech and typical background noise of a public space are considered, the strongest interfering signals in the 18–20-kHz range are the fricatives *f, s, sh,* and so on, when spoken close to the microphone. Then the worst situation in practice where the system should be operable occurs when the loudspeaker–microphone distance is 1 m, the microphone is off the main axis of the loudspeaker, and someone is pronouncing a loud *f* phoneme at a distance of 0.7 m from the microphone.

In the prototype control sticks shown in Fig. 13 we used an electret microphone (Sennheiser KE 4-211) as an acoustic receiver, which is sensitive enough and almost omnidirectional at 18–20 kHz. The measurement pulse is transmitted through a small headphone driver, which has a good response up to 20 kHz and is almost omnidirectional (within 5 dB for angles of ±60°). We found that many small earplug-type headphones have drivers that can be used as transmitters for this purpose to yield high enough received levels. By the interpolated peak approximation we have obtained about ±1-mm accuracy (repeatability) in the estimation of the left-to-right-hand distance.

2.4.4 Sensing of Right-Hand Movements

Sensing of the right-hand movements is done simply by acceleration sensors, which are available as microchips. We have applied the VTI Technologies chip SCA320-CDCV1G [52], which senses the acceleration perpendicular to the plane of the chip and delivers 0.15 V/g when driven by a +5-V supply voltage.

The acceleration signal is first bandpass filtered to the frequency range of 0.3–20 Hz. An example signal obtained for a rhythmic plucking movement of the right-hand stick is plotted in Fig. 16. Peaks of the envelope above a given threshold (dotted line) are used to send a pluck control event to the guitar synthesizer with an amplitude proportional to the height of the peak.

Simple strategies for the damping/muting of a string are 1) no damping of previous vibration during a new pluck, 2) fast muting of a previous note while activating a new one, and 3) damping when a negative acceleration peak is found. More advanced damping control can be realized by extra sensors (finger controls) in the control sticks.

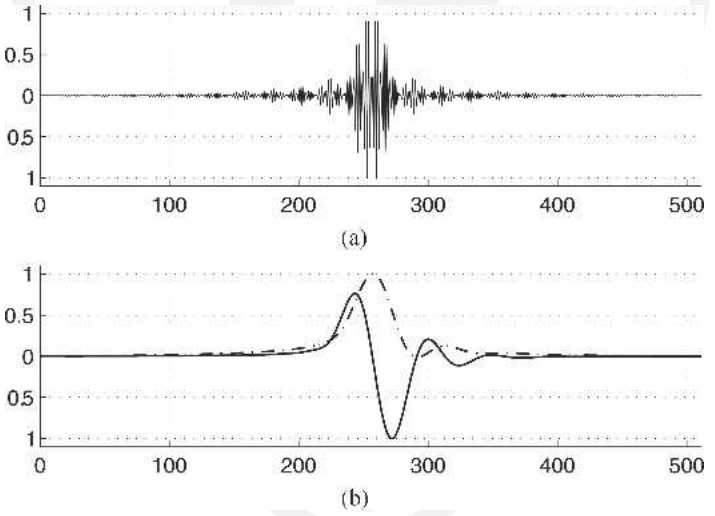

Fig. 15. (a) Bandpass pulse used as measurement signal. (b) Received envelope (—·—) and envelope derivative (——).

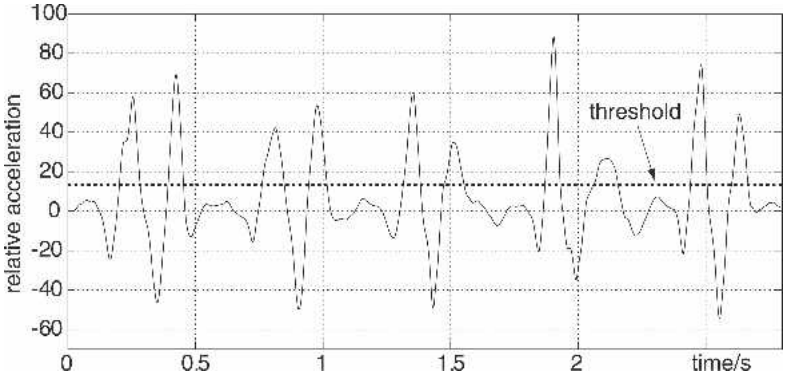

Fig. 16. Output of acceleration sensor for rhythmic plucking movement (——) and threshold for pluck event detection (....).

3 PLAYING SUPPORT SOFTWARE

A real guitar, electric or acoustic, offers several different playing techniques. Basically all of the techniques alter only a few sound parameters—pitch, amplitude, damping, and excitation. Our guitar string sound model offers the same parameters for real-time control. The excitation type, be it plectrum or finger, can be altered by changing excitation sample data sent to the guitar sound model. The other parameters are sent to the sound model through a guitar controller plugin using a guitar meta language which we developed.

Three different input hardware approaches offer different amounts of input parameters to be mapped into the guitar control parameters. The computer vision interface gives just the X/Y location of both hands. Hand velocity and acceleration can be extracted from the consecutive samples of the location data. However, there is no orientation information of the hands or finger flexure information that the virtual room interface offers. The control stick interface extracts just the distance between the user's hands and the acceleration of the user's right hand.

The fewer parameters the input hardware offers, the more control parameters of the sound model need to be generated automatically. The virtual reality system offers enough input parameters for implementing expressive free control, but the computer vision interface needs some intelligence for producing similarly expressive playing techniques. Expanding the control was implemented in two ways: by gesture extraction combined with a musical intelligence module and by a guitar meta language controlling the sound model through a guitar controller. The gestures are interpreted in the current musical context by the musical intelligence module. It translates the input events into guitar events and sends them to a module that keeps track of the guitar state. This guitar controller translates the guitar events into control parameters understood by the sound model.

3.1 Gesture Recognition

No matter which input hardware is used, the raw input data are sent to the gesture recognition module for gesture extraction (see Fig. 9). The module then attempts to recognize meaningful air guitar gestures, which are then used for controlling the sound model through a phrase database or through a context-aware musical intelligence module.

The control-stick version of the guitar is the simplest for control gesture extraction. It recognizes a pluck as an acceleration value that exceeds a certain threshold, and the distance between the sticks controls the pitch directly by altering the string lengths inversely proportional to the pitch.

In the webcam version a pluck is detected when the right hand passes through the imaginary guitar centerline (see Fig. 12), with a speed above a specified threshold to avoid random triggering. The centerline spans from the location of the left hand through a smoothed (low-pass filtered) location of the right hand. As a result the guitar moves with the player and allows to play even near the feet or behind the neck (as long as the gloves are visible). Yet because of the smoothing the guitar does not shake along with the plucks of the right hand. The low-pass filtering is simply a 1.5-second average of the right hand's location. If both hands move left or right at the same time (the user moves his/her whole body by stepping left or right), the average location of the right hand is immediately shifted accordingly.

The distance between the hands is mapped into fretting positions on the current scale by the musical intelligence module. The guitar calibrates its scale (size) based on the width of the user's grip at the beginning of the playing and can thus be played equally well by adults and children of all sizes. In the case of a noninformatic grip in the beginning, the calibration is also fine-tuned while playing and spatially expanded if the user's reach gets bigger than estimated at the beginning of the playing.

The gesture extractor understands also a few other guitar playing gestures. A semitonewide vibrato of 5.5 Hz is switched on when the left hand is shaking along the imaginary guitar neck. The vibrato is detected when the velocity of the user's left hand has changed direction along the guitar neck twice or more inside a short time threshold. Thus it takes several input samples to detect it. The vibrato is switched off when the shaking of the hand stops.

Moving the left hand (altering the distance between the hands) along the guitar neck translates into a fret-based slide. Thus the beginning of a vibrato may produce a slide in both directions before turning on the actual vibrato. This produces a proper start for the synthetic vibrato, which starts several animation frames late.

When there is only one large orange blob visible in the camera frame, all strings are muted. Thus the player can mute the strings, for instance, by placing his hand behind his back.

Except for muting, the virtual reality version of the guitar supports the same gestures as the webcam version. The virtual reality version contains also a continuous (nonfretted) pitch mode, where the automatic vibrato is switched off and the actual hand location of the player controls the pitch directly and makes the vibrato through shaking of the closed left hand. If the left hand is open, the pitch of the currently playing note is frozen, that is, not affected by altering the distance between the hands. Then the pitch of the plucked note is kept until the next plucking event. The jumps in pitch values (delay loop lengths) are always interpolated in the sound model to prevent clicks in the sound.

The virtual reality version supports also several additional control gestures. The user has control of hammer-on, pulloff, slide, and bend playing techniques. In the mode that supports these the pitch is again quantized to fret positions. A hammer-on is initiated if the middle finger of the left hand is bent while the right hand triggers a pluck. If the user now bends the ring finger of his left hand as the sound is still playing, a hammer-on is triggered. The hammer-on is executed as a new more silent pluck one fret line higher on the string used for producing the current sound.

Similarly to a hammer-on, a pulloff is triggered in the opposite bending order. Both the ring and the middle finger of the left hand are bent while the string is plucked. As the ring finger is then lifted, a pulloff is initiated. The pulloff triggers a more silent pluck one fret line lower on the same string. A new hammer-on can be triggered after a pulloff by bending the ring finger again.

In this playing mode a slide is done in the same spirit as the hammer-on and the pulloff. When the user bends the middle finger of his left hand, the slide mode is activated. If he closes also the ring finger, a hammer-on is triggered, but if he instead quickly slides the hand closer or further, a slide is done. The slide sends an event on each passing of a virtual fret line, generating a silent pluck after rescaling the pitch.

Bending is done by twisting the left hand around the imaginary neck of the air guitar. The forward direction vector of the knuckles is used for deciding how much to bend. Normally the vector points upward or away from the user. When the vector rotates more than a specified threshold in a predefined amount of time, the bend is activated. The orientation of the vector is stored as the bend begins. The amount of bending is then scaled to one semitone if the vector turns fully upward.

Bendings, hammer-ons, and pulloffs are controllable only on the virtual reality version of the VAG as the other input methods do not offer finger-state or hand-orientation information.

3.2 Musical Intelligence

In the case of the science center exhibit it was important to create an entertainment device instead of an accurate replica of a real guitar. After all, playing the air guitar should not require any skill with a real guitar. To this end we created a layer of musical intelligence that knows the musical context and makes educated guesses as to what the user wants to play. Technically the musical intelligence interprets the hand position data received from the input device within a certain musical context and generates meta language events to be sent to the guitar controller.

Since the concept of air guitar playing is directly related to a certain music style—indeed, to individual artists and musical pieces—the musical context is known beforehand. The user will want to play in the style of Jimi Hendrix, Pete Townshend, and so on. Therefore the musical intelligence module can be programmed to push the resulting music into the direction of this context. The challenge is to retain a feeling of control for the user, while still making sure the output does not sound chaotic.

To further narrow down the musical context, we came up with two different playing modes, which give the user a certain mindset of what to expect and how to play.

3.2.1 Chord Mode

The chord mode is initially active when the user starts playing the webcam version, as it is the simpler one of the two modes. The imaginary guitar neck (the maximum distance between the hands) is divided into four parts, each of which corresponds to a certain "power chord." Power chords are a common technique in rock guitar playing, usually played on the bottom three strings, producing a thick and heavy sound.

When the user makes a plucking motion with the right hand, the position of the left hand determines which chord will sound. The four chords available are those found in the introduction riff to the song "Smoke on the Water" by Deep Purple, an air guitar classic. Experiences at the science center indicate that getting this particular sequence is just challenging enough for users to learn it within the playing time, yet they find it very satisfying when they complete the challenge.

3.2.2 Solo Mode

The solo mode is more difficult to learn, but it has much more control over the sound—no two playing sessions are the same. Instead of the four chords in the chord mode, the guitar neck is divided into six parts, representing notes in an inverted pentatonic E minor scale, which is commonly used in rock guitar solos. A real guitarist would play this scale without having to move the left hand at all, just moving to a higher or lower string as necessary. Most nonguitarists do not know this, however, and mapping the left-hand position to pitch is more natural for users. The solo mode interprets this position and plays the guitar sound model as if a real guitarist were fretting one of the three top strings.

In addition to more notes, the solo mode also allows for embellishments. When the left hand moves along the guitar neck, a fret slide is played. This common technique adds expression (and fun) to the playing. Sliding is also intelligent in that when the left hand is stopped, the application makes sure the note it ends in belongs to the pentatonic scale. Users can also add vibrato to notes by quickly waving the left hand back and forth along the guitar neck. The vibrato is also intelligent in that its frequency and amplitude are predetermined and not actually read from the user, making it more controlled. Often the users do not even notice that they do not control the actual pitch and amplitude of the vibrato. This suggests that it reacts in the way they are musically accustomed to.

Finally we added a way of recognizing the intensity of playing. When both hands are close to each other, as is common in air guitar, it is possible for two or even three strings to be plucked at the same time. The two notes then produce a surprising and very satisfying distortion, as two different harmonic spectra are driven through the amplifier. Since the notes plucked are always on the pentatonic scale, "ugly" distortion from incompatible notes is not heard. The end result of this is that when the user starts playing very hard and fast, as is often the case with an air guitar, the VAG responds by increasing the intensity of the sound. This last feature in particular has made users want to play again just to reach that cool distortion. Presumably the effort put into the challenge is rewarded with a satisfying result.

3.3 Guitar Meta Language

The implementation of musical intelligence described in the previous section is a complex mapping from input data to sound model parameters. For the musical intelligence to work, there should be an abstract, musical representation of what a real guitarist would do when playing. However, neither musical notation nor sound model control data are very useful for this purpose. On the one hand, musical notation lacks in expressive information and the mechanical completeness required for computer processing, because the musicians are expected to interpret the score and add their own expression to it. And on the other hand, it would not be prudent to write down every change to each of the sound model's parameters, as this would result in hundreds, even thousands of lines of data for a simple passage.

Therefore an abstraction between these two is required, a language for representing guitar playing techniques that can be converted into sound model parameters. For this purpose we developed a meta language that describes rock guitar playing in musical concepts, and is therefore not tied to any sound model. Because the scope of the language is very specific, it can be made very efficient and intelligent.

The guitar control language defines events with parameters, much like the MIDI language. Each event is assigned a time code for when it is supposed to be run, and the event has some parameters that define what it does and how. Unlike MIDI, however, the air guitar language is designed just for this specific instrument, and the events it contains are directly related to actions of a real guitarist playing a real guitar. For example, there is a "pluck" event that corresponds to plucking a string on the guitar, and its parameters define which string is plucked, which fret is selected from the fretboard (that is, the pitch), how strong the string is plucked, and when it is muted.

With the meta language sitting in between the input device and the sound model, it is possible to work on a level of musical concepts, making it easy to control the musical style involved. For example, the sound model parameter for controlling pitch is the length of the delay line, which is not a musical concept. But with the meta language it is possible to ask the system to fret the third fret on the first string and pluck it. This makes it easy to restrict playing to the pentatonic scale, as it is possible to simply define the frets to be used. In addition, the events generated by the meta language are a somewhat accurate representation of what a real guitarist would be playing, making it easy to provide the physical restrictions that make guitar playing sound realistic.

The meta language also makes the system highly modular, as both the input device and the sound model can be changed easily. The input device can, for example, be a magnetic tracker, a MIDI keyboard, or a sequencer. All that needs to be done is to make it generate meta language events. Likewise, the sound model can also be anything, such as a physical model or a MIDI synthesizer, as long as there is a module that implements the meta language events for that particular sound model.

3.3.1 Meta Language Description

To compile a list of required events, we first needed to find out what real guitarists do when they play their instrument. In [26] Erkut et al. discuss some of the expressive parameters of the classical guitar. Furthermore, in [53] Rossing discusses what a guitar actually sounds like from a physical perspective. Table 3 lists the most important guitar-playing techniques in rock music.

These playing techniques are what the guitar sound model should be made to perform. Some of the techniques are redundant, though, and can be collated. For example, the measured hammer-on and grace note hammer-on techniques only differ in the time it takes to hit the second note on the fretboard, even if the result is very different in musical terms. But the sound model does not care about the musical terms, and so we can compress both techniques into one event. Or more accurately, we can separate the pluck into one event, and the hammering of a fret into another, and simply call them at the appropriate times. Table 4 shows a list of all events in the guitar control language and the parameters that can be given to the events.

In addition to the playing techniques, the language must also be able to define timing. Not all of the playing techniques can be tracked accurately with all the input mechanisms, and so some can only be triggered automatically by the musical phrase. For example, bending the strings is difficult to monitor by optical tracking due to a lack of resolution and three-dimensional depth. The phrase can still contain a bend, though it is simply played automatically at a specified time after a trigger has been received.

Timing in the control language is expressed in a notation of measures, beats, and ticks, a system commonly used by sequencer software. Each event has a time code such as 12:2:480, which would mark the note to play at 480 ticks after the second beat of the 12th measure.

The VAG application uses mostly individual events with carefully selected time codes to translate the user's gestures into sound model parameters. Still it is possible to use the meta language to write long phrases, or even entire compositions. This was not within the scope of this project, but is a topic worth exploring.

Initially we planned to have a phrase-playing mode in the guitar. The user would play a certain rock song by triggering events from the meta language description of the song. Every note would be triggered by the user, thus making him define the tempo. Bends and other techniques would be applied as written to the note or chord played. We made a few phrases and experimented with this to find out that the idea did not really work. Playing a song recognizably required knowing the song really well. Even someone who knows the song was easily lost in plucking after which it was hard to catch on again. So obviously this was not going to work for a science center exhibition as there are no such songs that everyone would know by heart. Thus this playing mode was dropped from the implementation.

With events and timing it is possible to represent most guitar-playing techniques on a musically conceptual level. It must be noted, however, that not everything that is possible to perform on a guitar can be expressed this way. Some ways of producing sound with an electric guitar could hardly be expressed in musical terms, even though they are considered music. For example, parts of musical pieces by Jimi Hendrix could not be written in this meta language, let alone in traditional musical notation. Another approach to this would be to build a model of the physical context of guitar playing, modeling the guitarist's hands, fingers, and plectrum, and creating rules on how they can move in physical reality.

4 SUMMARY AND CONCLUSIONS

In this engineering report we described the development of a playable virtual instrument called the virtual air guitar (VAG). It consists of handheld controllers and a guitar synthesizer with sound effects. Three different user interfaces for controlling the instrument have been tried: data gloves used in a virtual room, optical tracking of hand movements, and special control sticks using acoustic and acceleration sensing of hand movements. The control parameters are mapped to synthesis control parameters in various ways—from direct control of pitch and plucking to advanced control based on artificial intelligence principles.

We may conclude that the webcam user interface shows most potential for very low-cost applications, such as a VAG computer game. This is because of the fact that webcam cameras are frequently included in new computer hardware, so the application needs only software and very inexpensive colored gloves. The limited number of control parameters in this approach can be compensated for by intelligent algorithms of synthesis control. The control sticks can quite easily add control functions for different fingers and thus increase expression potential. Advanced data gloves naturally allow for maximum control of the synthesizer parameters.

Two versions of the VAG, the webcam and the control stick, have been applied as demonstration systems in a music-related exhibition in the Heureka Science Centre, Vantaa, Finland. Tens of thousands of visitors, most of whom never had really played electric guitar before, have enjoyed for a few minutes the feeling of being a guitar

Table 4. Events in guitar control language.

Event	Parameters
Pluck	String to pluck, fret to select, time to mute, strength
Hammer-on	Fret to select, strength
Pulloff	Fret to select after pulloff, strength
Bend	Fret to bend to, time to bend in
Slide	Fret to slide to, time to slide in
Slide on	Direction, speed
Slide off	None
Palm mute on	None
Palm mute off	None
Vibrato on	Speed, amplitude
Vibrato off	None
Pick scrape	Speed
Mute	String to mute

Table 3. Most important guitar playing techniques in rock music.

Technique	Description	Results
Pluck string	Plucking a string	Sound
Place fret	Pushing a string toward the fretboard at a specific fret	(No sound)
Vibrato	Wobbling a string up and down with the finger that is pressing it at the fret	Pitch change in a sine wave (only upward)
Grace note bend	Pushing the string upward, takes a short time	Note that has a bend up at the beginning
Measured bend	Pushing the string upward, takes longer and has a boundary between two notes	Note, smooth change to another higher note
Slide	Sliding fingers along the fretboard, usually going to a specified fret from a few frets above or below	Smooth pitch change in steps related to frets
Grace note hammer-on	Hitting a finger on the fretboard (usually preceded by a pluck), (for right-hand tapping also), takes a short time	Grace note upward, also damped sound
Measured hammer-on	Hitting a finger on the fretboard (usually preceded by a pluck)	Two rapid notes, the second higher, also damped sound
Grace note pulloff	Pulling a finger off the fretboard and replucking the string with it while keeping a second finger on a lower fret, takes a short time	Grace note downward, also damped sound
Measured pulloff	Pulling a finger off the fretboard and replucking the string with it while keeping a second finger on a lower fret, takes longer and has a boundary	Two rapid notes, the second lower, also damped sound
Mute	Lifting a finger from the fretboard but still touching the string	Cutting off the sound from a string
Palm mute	Performing a pluck while holding the side of the right hand on top of the string	In conjunction with pluck, produces short rhythmic sound
Pick scrape	Pulling the plectrum along the strings toward the end of the neck	Screeching, falling sound
Harmonics	Touching a string over a fret bar	Bell-like sound

hero rock star. The exceptional popularity and media success of the VAG shows the potential of such entertaining devices. Much experience has been collected through the development and usage of this virtual musical instrument. This information can be applied to the design of future virtual instruments for fun but also toward more professional musical instruments.

Sound demonstrations and information related to the VAG are available at http://airguitar.tml.hut.fi/.

5 ACKNOWLEDGMENT

This work was supported by the IST/ALMA project (algorithms for the modeling of acoustic interactions, IST-2001-33059) and the Academy of Finland projects SA 104934 and SA 53537. The authors are grateful for the interest of the Heureka Science Centre in the development of the virtual instrument and for the help of VTI Technologies with acceleration sensors. Special thanks are due to Hannu Pulakka, who converted the tube distortion algorithm to BlockCompiler, and Pekka Jänis, who contributed to the control stick experiments.

6 REFERENCES

[1] V. Välimäki, J. Pakarinen, C. Erkut, and M. Karjalainen, "Discrete-Time Modelling of Musical Instruments," *Rep. on Progr. in Phys.,* vol. 69, pp. 1–78 (2006).

[2] J. O. Smith, "Physical Audio Signal Processing," http://ccrma.stanford.edu/~jos/pasp/ (2006 June 10 ed.).

[3] J. Paradiso, "Electronic Music Interfaces: New Ways to Play," *IEEE Spectrum,* vol. 34, no. 12, pp. 18–30 (1997); expanded as an online article, http://web.media.mit.edu/~joep/SpectrumWeb/SpectrumX.html (1998, visited 2006 July).

[4] M. Wanderley and M. Battier, Eds., *Trends in Gestural Control of Music* (IRCAM, Centre Pompidou, Paris, France, 2000).

[5] R. Rowe, *Interactive Music Systems—Machine Listening and Composing* (MIT Press, Cambridge, MA, 1992).

[6] T. Winkler, http://www.brown.edu/Departments/Music/sites/winkler//papers/Making_Motion_Musical_1995.pdf.

[7] M. Waiswicz, "THE HANDS: A Set of Remote MIDI Controllers," in *Proc. 1985 Int. Computer Music Conf.,* B. Truax, Ed. (Computer Music Assoc., San Francisco, CA, 1985).

[8] A. Hunt, "Radical User Interfaces for Real-Time Musical Control," Ph.D. dissertation, University of York, York, UK (1999).

[9] S. Jordà, M. Kaltenbrunner, G. Geiger, and R. Bengina, "The Reactable," in *Proc. Int. Computer Music Conf. (ICMC'05*), (Barcelona, Spain, 2005).

[10] T. Mäki-Patola, A. Kanerva, J. Laitinen, and T. Takala, "Experiments with Virtual Reality Instruments," in *Proc. Int. Conf. on New Interfaces for Musical Expression (NIME'05),* (Vancouver, B.C., Canada, 2005).

[11] I-Cube website, http://infusionsystems.com/catalog/index.php (visited 2006 July).

[12] Theremin info pages, http://www.theremin.info/ (visited 2006 July).

[13] W. T. Freeman, D. Anderson, P. Beardsley, C. Dodge, H. Kage, K. Kyuma, Y. Miyake, M. Roth, K. Tanaka, C. Weissman, and W. Yerazunis, "Computer Vision for Interactive Computer Graphics," *IEEE Computer Graphics Appl.* (1998).

[14] W. T. Freeman, P. A. Beardsley, H. Kage, K. Tanake, K. Kyuma, and C. D. Weissman, "Computer Vision for Computer Interaction," *SIGGRAPH Computer Graphics Newsl.,* vol. 33, no. 4 (1999).

[15] T. B. Moeslund, "Computer Vision-Based Human Motion Capture—A Survey," LIA Rep. 99-02, Aalborg University, Lab. of Computer Vision and Media Technology (1999).

[16] http://en.wikipedia.org/wiki/Air_guitar.

[17] http://www.omvf.net/2005/ilmakitara.php.

[18] D. A. Jaffe and J. O. Smith, "Extensions of the Karplus–Strong Plucked-String Algorithm," *Computer Music J.,* vol. 7, no. 2, pp. 76–87 (1983), reprinted in *The Music Machine,* C. Roads, Ed. (MIT Press, Cambridge, MA, 1989).

[19] M. Karjalainen, V. Välimäki, and T. Tolonen, "Plucked-String Models: from the Karplus–Strong Algorithm to Digital Waveguides and Beyond," *Computer Music J.,* vol. 22, no. 3, pp. 17–32 (1998 Fall).

[20] U. Zölzer, Ed., *DAFX—Digital Audio Effects* (Wiley, New York, 2002).

[21] K. Karplus and A. Strong, "Digital Synthesis of Plucked-String and Drum Timbres," *Computer Music J.,* vol. 7, no. 2, pp. 43–55 (1983); reprinted in *The Music Machine,* C. Roads, Ed. (MIT Press, Cambridge, MA, 1989).

[22] C. S. Sullivan, "Extending the Karplus–Strong Algorithm to Synthesize Electric Guitar Timbres with Distortion and Feedback," *Computer Music J.,* vol. 14, no. 3, pp. 26–37 (1990).

[23] V. Välimäki, J. Huopaniemi, M. Karjalainen, and Z. Jánosy, "Physical Modeling of Plucked String Instruments with Application to Real-Time Sound Synthesis," *J. Audio Eng. Soc.,* vol. 44, pp. 331–353 (1996 May).

[24] M. Karjalainen and J. O. Smith, "Body Modeling Techniques for String Instrument Synthesis," in *Proc. Int. Computer Music Conf. (ICMC'96),* (Hong Kong, 1996 Aug. 19–24), pp. 232–239.

[25] V. Välimäki and T. Tolonen, "Development and Calibration of a Guitar Synthesizer," *J. Audio Eng. Soc.,* vol. 46, pp. 766–778 (1998 Sept.).

[26] C. Erkut, V. Välimäki, M. Karjalainen, and M. Laurson, "Extraction of Physical and Expressive Parameters for Model-Based Sound Synthesis of the Classical Guitar," presented at the 108th Convention of the Audio Engineering Society, *J. Audio Eng. Soc. (Abstracts),* vol. 48, p. 354 (2000 Apr.), convention paper 5114.

[27] L. Trautmann and R. Rabenstein, *Digital Sound Synthesis by Physical Modeling Using the Functional Transformation Method* (Kluwer Academic/Plenum, New York, 2003).

[28] Z. Jánosy, M. Karjalainen, and V. Välimäki, "Intelligent Synthesis Control with Applications to a Physical Model of the Acoustic Guitar," in *Proc Int. Computer Music Conf. (ICMC'94),* (Aarhus, Denmark, 1994 Sept.), pp. 402–406.

[29] M. Laurson, C. Erkut, V. Välimäki, and M. Kuus-

kankare, "Methods for Modeling Realistic Playing in Acoustic Guitar Synthesis," *Computer Music J.,* vol. 25, no. 3, pp. 38–49 (2001 Fall).

[30] T. Jungmann, "Theoretical and Practical Studies on the Behavior of Electric Guitar Pick-Ups," Helsinki University of Technology, Helsinki, Finland (1994 Nov.); also M.Sc. thesis, Fachhochschule Kempten, Kempten, Germany. Electronic version available at http://www.acoustics.hut.fi/publications/files/theses/jungmann_mst.pdf.

[31] J. O. Smith, "Efficient Synthesis of Stringed Musical Instruments," in *Proc. Int. Computer Music Conf. (ICMC'93),* (Tokyo, Japan, 1993 Sept.), pp. 64–71.

[32] M. Karjalainen, V. Välimäki, and Z. Jánosy, "Towards High-Quality Sound Synthesis of the Guitar and String Instruments," in *Proc. Int. Computer Music Conf (ICMC'93),* (Tokyo, Japan, 1993 Sept.), pp. 56–63.

[33] N. H. Fletcher and T. D. Rossing, *The Physics of Musical Instruments,* 2nd ed. (Springer, New York, 1998).

[34] M. Karjalainen, J. Backman, and J. Pölkki, "Analysis, Modeling, and Real-Time Sound Synthesis of the Kantele, a Traditional Finnish String Instrument," in *Proc. IEEE Int. Conf. on Acoustics Speech and Signal Processing* (Minneapolis, MN, 1993), pp. 229–232.

[35] P. A. A. Esquef, M. Karjalainen, and V. Välimäki, "Frequency-Zooming ARMA Modeling for Analysis of Noisy String Instrument Tones," *EURASIP J. Appl. Signal Process.* (Special Issue on Digital Audio for Multimedia Communications), vol. 2003, pp. 953–967 (2003 Sept.).

[36] T. Laakso, V. Välimäki, M. Karjalainen, and U. K. Laine, "Splitting the Unit Delay—Tools for Fractional Delay Filter Design," *IEEE Signal Process. Mag.,* vol. 13, pp. 30–60 (1996).

[37] M. Karjalainen, "BlockCompiler: Efficient Simulation of Acoustic and Audio Systems," presented at the 114th Convention of the Audio Engineering Society, *J. Audio Eng. Soc. (Abstracts),* vol. 51, p. 416 (2003 May), convention paper 5756.

[38] PureData software, http://www.crca.ucsd.edu/~msp/software.html.

[39] http://www.tml.tkk.fi/~tilmonen/mustajuuri/.

[40] http://quitte.de/dsp/caps.html.

[41] M. Karjalainen and J. Pakarinen, "Wave Digital Simulation of a Vacuum-Tube Amplifier," in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP'06),* (Toulouse, France, 2006).

[42] R. Aiken, "Designing Common-Cathode Amplifiers" (2003), http://www.aikenamps.com/CommonCathode.htm.

[43] http://www.normankoren.com/Audio/Tubemodspice_article.html.

[44] R. Aiken, "What Is Blocking Distortion" (1999), http://www.aikenamps.com/BlockingDistortion.html.

[45] R. Aiken, "What Is Miller Capacitance" (1999), http://www.aikenamps.com/MillerCapacitance.html.

[46] General Electric, "12AX7 Twin Triode Datasheet" (1953), http://www.mif.pg.gda.pl/homepages/frank/sheets/093/1/12AX7.pdf.

[47] "AX84 High Octane Amplifier Revision 4" (2001), http://annex.ax84.com/media/ax84_m161.pdf.

[48] Information of the EVE virtual room, http://eve.hut.fi/ (visited 2006 July).

[49] Homepage of the EyesWeb software, http://www.eyesweb.org/ (visited 2006 July).

[50] G. Jacovitti and G. Scarano, "Discrete Time Techniques for Time Delay Estimation," *IEEE Trans. Signal Process.,* vol. 41 (1993 Feb.).

[51] X. Lai and H. Torp, "Interpolation Methods for Time-Delay Estimation Using Cross-Correlation Method for Blood Velocity Measurement," *IEEE Trans. Ultrasonics, Ferroelectrics and Freq. Contr.,* vol. 46 (1999 Mar.).

[52] Homepage of VTI, http://www.vti.fi/productsen/accelerometers.html.

[53] T. D. Rossing, *The Science of Sound,* 2nd ed. (Addison-Wesley, Reading, MA, 1990), pp. 201–213.

[54] L. R. Rabiner, "A Tutorial of Hidden Markov Models and Selected Applications in Speech Recognition," *Proc. IEEE,* vol. 77, pp. 257–286 (1989 Feb.).

[55] T. Mäki-Patola and P. Hämäläinen, "Latency Tolerance for Gesture Controlled Continuous Sound Instrument Without Tactile Feedback," in *Proc. Int. Computer Music Conf.* (Miami, FL, 2004 Oct.).

[56] T. Mäki-Patola and P. Hämäläinen, "Effect of Latency on Playing Accuracy of Two Continuous Sound Instruments Without Tactile Feedback," in *Proc. Digital Audio Effects Conf. (DAFx'04),* (Naples, Italy, 2004 Oct.).

THE AUTHORS

M. Karjalainen

T. Mäki-Patola

A. Kanerva

Matti Karjalainen was born in Hankasalmi, Finland, in 1946. He received M.Sc. and Dr.Sc. (Tech.) degrees in electrical engineering from the Tampere University of Technology, Finland, in 1970 and 1978, respectively.

Since 1980 he has been a professor of acoustics and audio signal processing in the Department of Electrical and Communications Engineering at the Helsinki University of Technology, Finland. His main interest in audio technology is audio signal processing, such as DSP for sound reproduction and auralization, music DSP and sound synthesis, as well as perceptually based signal processing. His research activities also cover speech processing, perceptual auditory modeling and spatial hearing, DSP hardware, software, and programming environments; as well as various branches of acoustics, including musical acoustics and modeling of musical instruments.

Professor Karjalainen has written 350 scientific and engineering papers and has contributed to organizing several conferences and workshops, including serving as the papers chair of the AES 16th Conference and as the technical chair of the AES 22nd Conference.

He is an AES fellow and has been awarded the AES Silver Medal. He is a member of the Institute of Electrical and Electronics Engineers, the Acoustical Society of America, the European Acoustics Association, the International Speech Communication Association, and several Finnish scientific and engineering societies.

●

Teemu Mäki-Patola was born in the small town of Jämsä, Finland, in 1977. He received an M.Sc. degree in physics from the Helsinki University of Technology, Finland, in 2002. He is currently working on his doctoral thesis, "Future of Rock'n'Roll—Controlling Sound with Novel Interface Technology in Interactive Digital Media," at the Helsinki University of Technology.

His interests in audio technology focuses on novel user interface technology, human perception, and designing new ways to control current sound synthesis methods. He is also interested and involved in computer game design, computer graphics, and artificial intelligence. He enjoys playing the piano, oil painting, snowboarding, and scuba diving.

●

Aki Kanerva was born in 1980 in Vantaa, Finland. His studies at the Telecommunications Software and Multimedia Laboratory of the Helsinki University of Technology include contents production and usability design. He is writing his M.Sc. (Tech.) thesis on the "Virtual Air Guitar."

From 2004 to 2005 he worked as a research assistant in the ALgorithms for the Modeling of Acoustic Interactions (ALMA) project. Since then he has copublished three scientific articles. His interests include interface design, musical intelligence, and multimedia entertainment. He plays the piano and guitar, composes music, and takes casual photographs.

●

Antti Huovilainen was born in Espoo, Finland, in 1978. He is currently finishing his M.Sc. (Tech.) degree in electrical engineering at the Helsinki University of Technology, Finland.

His interests include modeling musical circuits for digital implementation, particularly guitar amplifiers and analog synthesizers. He has written several articles on virtual analog synthesis techniques. His hobbies include reading and playing the guitar.