Jyri Pakarinen, Tapio Puputti, and Vesa Välimäki
Helsinki University of Technology
Department of Signal
Processing and Acoustics
P.O. Box 3000
FI-02015 TKK Finland
{jyri.pakarinen, tapio.puputti,
vesa.valimaki}@tkk.fi

Virtual Slide Guitar

This article describes a system for virtual slide guitar playing. From the control point of view, it can be seen as a successor of the virtual air guitar (VAG) developed at Helsinki University of Technology (TKK) a few years ago (Karjalainen et al. 2006). The original VAG allows the user to control an electric guitar synthesizer by mimicking guitar-playing gestures. In the current work, the same gesture-control approach that was found successful in the VAG is used: a computer-vision-based system, in which a camera detects the player's hands and a computer tracks the hand movements and converts them into control data, such as pluck events and string length. Sound synthesis in the virtual slide guitar application is based on an energy-compensated time-varying digital waveguide model of a guitar with new extensions to generate contact sounds caused by the slide tube touching the strings. Video files showing the virtual slide guitar in action can be found on the accompanying Web page (www.acoustics.hut.fi/publications/papers/vsg/) and the forthcoming 2008 *Computer Music Journal Sound and Video Anthology* DVD. Although the current implementation uses a camera-based user interface, it can also be controlled by other human–machine interfaces or computer programs.

Excellent reviews on gestural control of music synthesis have been written by Paradiso (1997) and by Wanderley and Depalle (2004). Camera-based gesture analysis has become very sophisticated owing to increased computing power and new methodologies provided by research, as exemplified by the EyesWeb platform (Camurri et al. 2000, 2005; Gorman et al. 2007). Digital waveguide modeling is mature technology, which can be applied to high-quality synthesis of various musical instruments (Smith 1992; Välimäki et al. 2006). For more information on waveguide synthesis of string instruments, see Välimäki et al. (1996), Karjalainen, Välimäki, and Tolonen (1998), and Karjalainen et al. (2006).

The roots of this study are in an early prototype of the TKK virtual air guitar, which here is called the rubber-band virtual air guitar. It was a simplified gesture-control system, in which the distance between the hands was detected and directly converted to string length. The string length was updated frequently enough to be practically instantaneously dependent on hand locations. The rubber-band VAG was entertaining for a short time, but it was not a practical musical instrument: There was a noticeable latency between a pluck and the resulting sound, pitch changes occurred too effortlessly, and it was practically impossible to maintain a constant pitch. As a consequence, a rubber-band VAG player generated glissandi up and down or tones with shaky pitch but could not play a single melody. In the advanced versions of the VAG, the distance between hands is quantized so that a region of several centimeters in the air corresponds to a constant pitch or a single chord (Karjalainen et al. 2006). Although such a restriction decreases the expressive range, it also enables the user to learn play simple melodies and chords with this invisible instrument.

The authors realized that the rubber-band VAG controller could be used for controlling slide-guitar synthesis. The term *slide guitar* or *bottleneck guitar* refers to a specific traditional playing technique on a steel-string acoustic or electric guitar. When playing the slide guitar, the musician wears a slide tube on the fretting hand. Figure 1 illustrates this.

Instead of pressing the strings against the fretboard, the player glides the tube on the strings while the picking hand plucks the strings in a regular fashion. This produces a unique, voice-like tone with stepless pitch control. Although the tube is usually slid along all six strings, single-note melodies can

Computer Music Journal, 32:3, pp. 42–54, Fall 2008

Figure 1. When playing the slide guitar, the musician wears a slide tube on the fretting hand.

be played by plucking just one string and damping the others with the picking hand. The slide tube—usually made of glass or metal—also generates a squeaking sound while moving along on the wound metal strings. In most cases, the slide guitar is tuned into an open tuning (for example the open G tuning: D2, G2, D3, G3, B3, and D4 starting from the thickest string). This allows the user to play simple chords just by sliding the tube into different positions on the guitar neck. The player usually wears the slide tube on the little or ring finger (see Figure 1), and the other fingers are free to fret the strings normally. For an example of slide-guitar playing, refer to Johnson (1990).

In slide-guitar playing, the user gets continuous auditory feedback of the hand movements, and this helps the regulation of hand movements and pitch control in the virtual slide-guitar application. The present article and the related videos (on the forthcoming DVD) show that this leads to a functional application, the virtual slide guitar (VSG).

Slide Guitar Sound Analysis

The unique, characteristic sound of the slide guitar is mostly due to the continuous change in pitch. Typically, the player plucks a string while simultaneously damping out any vibrations in the other strings. Next, the player slides the tube to a new position on the guitar neck, thus continuously changing the pitch. After reaching the target pitch, a vibrato is usually applied by sliding the tube back and forth. Owing to the excessive use of vibrato, constant-pitched tones are usually only those that are played on the open strings, namely, strings not touched by the tube. The sliding tube-string contact excites both string segments, the one between the bridge and the tube, and the one between the tube and the nut. The latter segment is often, however, damped with the free fingers of the tube hand.

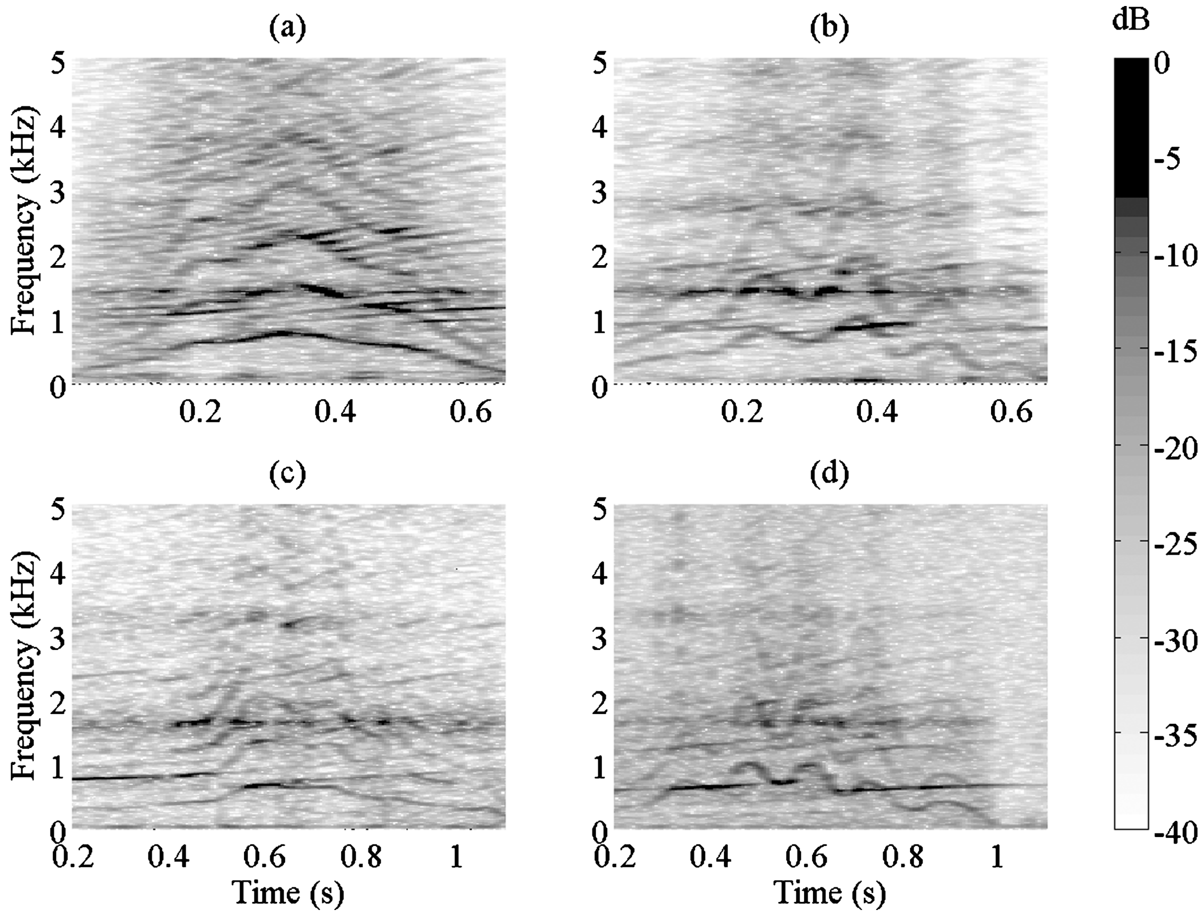
Contact Sounds

The sliding contact point between the string and the slide tube produces noise. The type of this contact sound depends not only on the tube material, but also on the construction of the strings. Wound strings (i.e., strings where another string is wrapped around a core string) tend to produce a squeaky sound, whereas plain strings produce a quiet, hiss-like noise. The contact noise type also varies with the sliding movement: Fast slides produce louder contact noise with more emphasis on the high frequencies than do slow slides.

Figure 2 shows spectrogram images of four different slide events conducted on a steel-string acoustic guitar. These slides were performed by simply moving the tube from one position to another on a single string, without any special attempt for maintaining a constant slide velocity. The recordings were made in the small anechoic chamber at Helsinki University of Technology using a microphone (AKG C 480 B) placed 1 m away from the instrument and directed at the sound hole. The signals were recorded digitally (44,100-Hz sampling rate, 16 bits) using a Yamaha 01V digital mixer and then fed into a PC laptop via a Digigram VX Pocket soundcard. In Figure 2a, the slide was performed with a brass tube on the wound sixth string. In Figure 2b, a glass tube was slid on the same sixth string. Figures 2c and 2d illustrate the slides of a brass and glass tube, respectively, on the wound fifth string. In all cases, the string was damped on both sides of the tube so that the transversal string vibrations were heavily attenuated.

Generally, the contact sound between the tube and the wound strings is caused by the same

Figure 2. Spectrogram images of four different slide events conducted on a steel-string acoustic guitar: (a) the slide was performed with a brass tube on the wound sixth string; (b) a glass tube was slid on the same sixth string; (c) and (d) illustrate the slides of a brass and glass tube, respectively, on the wound fifth string. In all cases, the string was damped on both sides of the tube so that the string vibrations were heavily attenuated. The x-axis shows time in seconds.

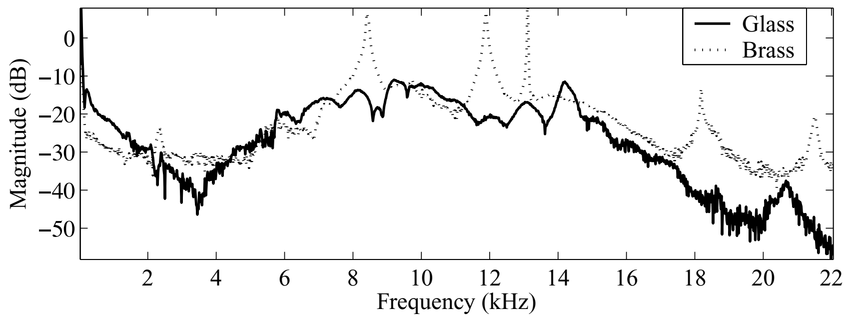


phenomenon as the handling noises created by a moving finger-string contact. These have been thoroughly analyzed in a recent study (Pakarinen, Penttinen, and Bank 2007), so only a brief discussion of the noise properties is given here. It can be seen in Figure 2 that in all cases, the contact sound has some common timbral qualities. First, the overall spectral shape of the noise is of a low-pass type, where the slide velocity controls the cutoff frequency. Second, the noise has a clear harmonic structure, where the frequency of the harmonics depends on the slide velocity. The harmonic structure of the sound can be explained by noting that the slide noise is periodic in nature owing to the windings around the string. A similar phenomenon was noted in a recent study related to a Chinese string instrument that also uses wound strings (Penttinen et al. 2006).

Third, in addition to the rather sharp, moving harmonic resonances, there are wider, static harmonic resonances with the lowest peak near 1,500 Hz. These components are caused by the longitudinal string vibration (Pakarinen, Penttinen, and Bank 2007).

There are some differences in the contact noise depending on the tube and string type. Generally, with thicker strings, the noise is louder than with thin strings. Also, the high-frequency content is more prominent with thick strings. This is most likely due to the smaller windings of the thinner strings, so the string surface is smoother. There are

Figure 3. Averaged magnitude responses of several measurements conducted on the glass and brass tubes.



differences caused by the tube material as well: A brass tube produces a louder noise with more high-frequency content than a glass tube. Furthermore, the temporal variation in the harmonic resonance frequencies depends on the tube material: Glass tubes tend to produce more frequency modulation or jitter in the resonances than brass tubes. The authors suspect that this is caused by the mass differences between the two slides: the brass tube is clearly heavier (80 g) than the glass tube (14 g). The smaller inertia of the glass tube might allow the player's small involuntary muscle tremble to be transferred more easily to the slide tube movement, thus producing frequency modulation at the harmonic resonances. Also, the different frictional characteristics between the string and the two tubes might have an effect.

The contact noise between a slide tube and a plain (i.e., unwound) string is very much quieter when compared to a wound string. In fact, it does not have a harmonic structure, and it closely resembles white noise. It must be noted that when the slide guitar is played, the first contact between the string and the tube produces a percussive click when the tube is pressed on the string. This click is small in magnitude but still audible.

Tube Vibration

The impulse responses of different slide tubes were measured to find out if the tube vibration produces an audible effect in the contact sound. The impulse responses were measured by dropping a pen on the slide tube and recording the generated sound. The slide tube was placed in situ (i.e., on the player's finger). The same measurement setup was used as discussed earlier, except that now the signals were recorded into a Macintosh laptop via an Edirol UA-101 audio interface. Figure 3 illustrates the averaged magnitude responses of several measurements conducted on the glass and brass tubes.

In Figure 3, it can be seen that the glass tube does not possess strong resonances in the audio range. Although the spectrum of the brass tube shows a few sharp resonant peaks, they are too high in frequency (over 8 kHz) to be effectively coupled to the guitar body and radiated as audible sound. Thus, we concluded that it is the surface texture of a slide tube, rather than its vibration, that creates the audible difference between the tube types.

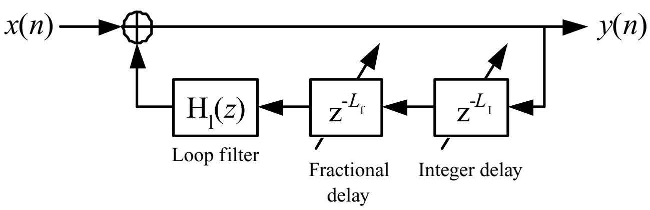
Slide Guitar Synthesis

Sound synthesis of the slide guitar is carried out using a time-varying digital waveguide string. The handling sounds emanating from the sliding contact between the string and the slide tube are synthesized with a parametric model and inserted as excitation into the waveguide.

Digital Waveguide String with Time-Varying Pitch

A single-delay loop (SDL) digital waveguide (DWG) model (Karjalainen, Välimäki, and Tolonen 1998) with time-varying pitch forms the basis of the slide-guitar synthesis engine. Figure 4 illustrates such a string model.

Figure 4. Single-delay loop digital waveguide model (Karjalainen, Välimäki, and Tolonen 1998) with time-varying pitch forms the basis of the slide-guitar synthesis engine.



The waveguide consists of a simple integer delay loop with two additional filters: one for implementing the fractional part of the delay, and the other (referred to here as the "loop filter") for simulating the vibrational losses in the string. Note that both the integer delay line length and the fractional delay filter are time-varying: The user controls the total loop delay value—and thus also the pitch—during run time.

The fractional delay filter allows for a smooth transition between pitches and also enables the correct tuning of the string. There are several techniques for implementing fractional delay filters, a thorough tutorial being found in Laakso et al. (1996). For the purpose of this work, a fifth-order Lagrange interpolator was found to work sufficiently well. It must be noted that, because the interpolation accuracy of a fractional delay filter is highest near the midpoint of the filter (e.g., near delay value 2.5 for a 6-tap filter), this fifth-order Lagrange interpolator is operated in the delay range from 2.0 to 3.0 samples, and the constant two-sample overhead delay due to the Lagrange interpolator is compensated in the integer delay line by making it two samples shorter.

For the loop filter, a one-pole lowpass filter [$H_l(z) = g(1 + a)/(1 + az^{-1})$ with cutoff parameter a and gain g] is used with approximated polynomial parameters depending on the length and type of the string, as suggested in Välimäki and Tolonen (1998).

Energy Compensation

When changing the length of a DWG string during run time, the signal energy is varied (Pakarinen et al. 2005). For example, if the DWG string is suddenly shortened to half of its original length, half of the signal samples are discarded, and approximately 50 percent of the signal energy is lost. In practice, this can be heard as an unnaturally quick decay of the string sound. While the true energetic behavior of an actual physical string with time-varying length might be controversial, one can safely assume that this type of artificial energy change owing to the DWG implementation is non-physical, and we must therefore compensate for it. Two energy-compensation methods are presented by Pakarinen et al. (2005), from which the computationally simpler energy scaling method, the zero-order energy-preserving interpolation, which adds a single time-varying scaling coefficient into the SDL, was chosen.

The basic difficulty with energy compensation in time-varying strings is that when the length of the DWG string changes, an estimate for the additional loss or gain of energy would be needed. Obviously, this requires an estimate for the signal values in the lost or gained delay line segment. In the zero-order energy-preserving interpolation, a constant signal value is assumed for this delay segment, and, as noted by Pakarinen et al. (2005), the scaling operation can be expressed as

$$p_c(n) = \sqrt{1 - \Delta x}\, p(n) = g_c\, p(n) \tag{1}$$

Here, n is the time index, $p(n)$ is the signal output from the time-varying delay block, $p_c(n)$ is the energy-compensated signal, g_c is the scaling coefficient, and Δx is the delay-line variation in samples per one time step. The zero-order energy-preserving interpolation is accurate only when the string's length does not change too rapidly compared to the wavelength λ:

$$|\Delta x| \ll \lambda \tag{2}$$

Assume that a quick slide with open G tuning is performed from the second fret on the sixth string (E2 = 82.4 Hz = f_1) one octave up to the 14th fret (E3 = 164.81 Hz = f_2), and it takes time $\Delta t = 0.5$ sec. Now, the absolute value of the delay line length change per time step will be

$$|\Delta x| = \frac{1}{2\Delta t}\left(\frac{1}{f_2} - \frac{1}{f_1}\right) \tag{3}$$

which equals 0.0061 for the frequencies chosen here. For a given frequency f, the wavelength on a

Figure 5. Contact-sound excitation mechanism.

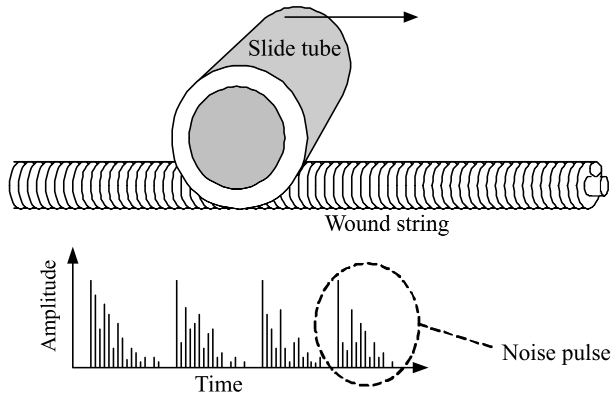
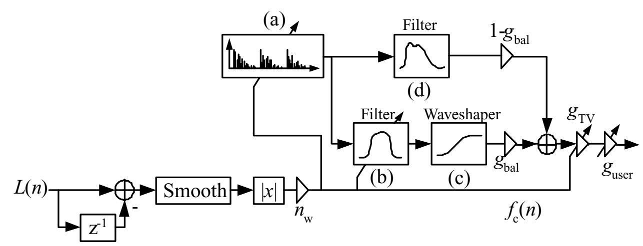


Figure 6. General structure of the contact-noise generator block.



string can be expressed as $\lambda = 2Lf_0/f$, where L is the length of the string and f_0 is the fundamental frequency of the open string. Thus, an expression for a limit frequency

$$f_{\lim} = \frac{2Lf_0}{|\Delta x|} \tag{4}$$

can be formulated, which states the upper bound for the zero-order energy-preserving interpolation. For the sixth string in open-G tuning, $L = 0.65$ m, and $f_0 = 73.4$ Hz, so the interpolation is accurate only for frequencies well below $f_{\lim} = 16$ kHz. In practice, all sustained tones in the guitar lie well below that limit, so the energy compensation method can be considered accurate enough for modeling purposes.

Contact Sound Synthesis

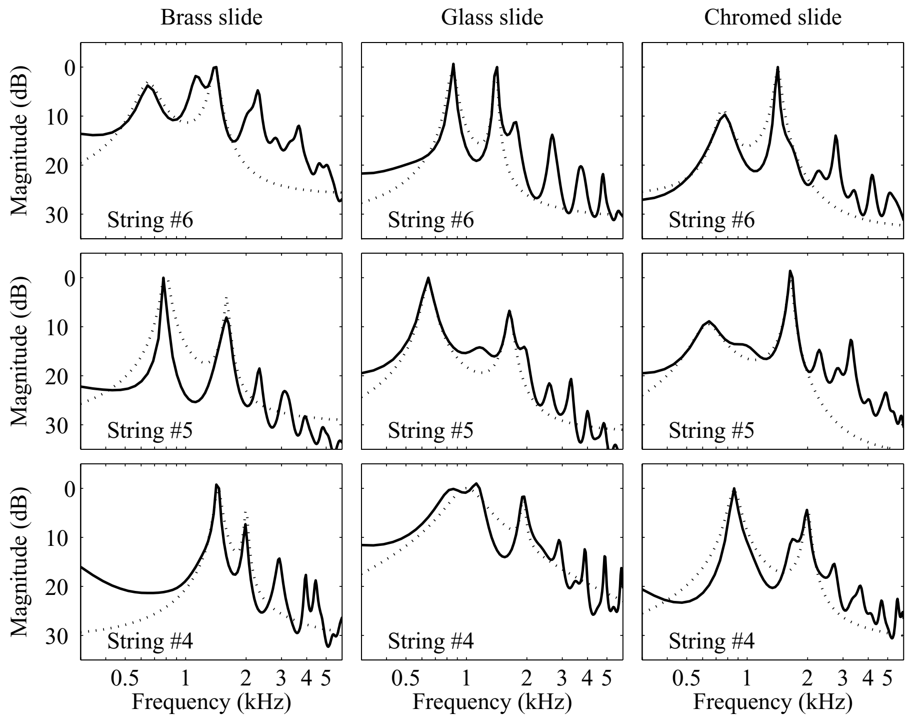
To model the handling sounds generated by the contact between the sliding tube and the string, we chose a noise pulse train as the excitation signal. This is based on the assumption that when the tube slides over a single winding, it generates a short, exponentially decaying noise burst. Although recorded pulses could be used in the synthesis, the contact-sound mechanism is so quick in practice that the exact pulse shape is not critical, as long as identical pulses are not used. This excitation mechanism is illustrated in Figure 5.

The time interval between the noise pulses is controlled by the sliding velocity; a fast slide results in a temporally dense pulse train, whereas a slow slide makes the pulses appear further apart. In some sense, the contact sound synthesizer can be seen as a periodic impact sound synthesis model rather than a friction model. Impact and friction sound synthesis models have been presented by Aramaki and Kronland-Martinet (2006), Aramaki et al. (2006), Avanzini, Serafin, and Rocchesso (2005), Cook (1997), Peltola et al. (2007), Rath and Rocchesso (2005), Fontana (2003), Rocchesso and Fontana (2003), and Rocchesso, Bresin, and Fernström (2003).

The general structure of the contact-noise generator block is illustrated in Figure 6. The input variable $L(n)$ denotes the relative string length controlled by the user. Here, n is the time index. Because the contact noise depends on the sliding velocity, a time difference is taken from the input signal. If the control rate of the signal $L(n)$ is different from the sound-synthesis sampling rate—as is often the case—a separate smoothing block is required after the differentiator. The smoothing block changes the sampling rate of $L(n)$ to be equal to the synthesis sampling rate and uses polynomial interpolation to smooth the control signal. Furthermore, because the contact noise is independent of the direction of the slide (up/down on the string), the absolute value of the control signal is taken. The scaling coefficient n_w denotes the number of windings on the string. The signal f_c in the output of this scaling block can therefore be seen as the noise-pulse-firing rate.

The basis of the synthetic contact sound for wound strings is produced in the noise-pulse-train generator [block (a) of Figure 6]. It produces exponentially decaying noise pulses at the given firing rate. In addition, the type of the string determines the decay time and duration of an individual pulse. For enhancing the harmonic structure of the contact noise on wound strings, the lowest time-varying harmonic is emphasized

Figure 7. Contact-sound filter magnitude responses (dotted lines) in comparison with contact-sound spectral estimates, obtained using a linear-prediction filter of order 100.



by filtering the noise pulse train with a second-order resonator [block (b)], where the firing rate controls the resonator's center frequency. The higher harmonics are produced by distorting the resonator's output with a suitable nonlinear function [block (c)]. A scaled hyperbolic tangent function is used for this. Hence, the number of higher harmonics can be controlled by changing the scaling of this nonlinear function. This approach for generating harmonics is similar to waveshaping synthesis (Arfib 1979; Le Brun 1979).

A fourth-order IIR filter [block (d)] is used for simulating the static longitudinal string modes and the general spectral shape of the contact noise. Obviously, as the noise characteristics depend on the tube material and string type, different filter parameters should be used for different slide tube and string configurations. In Figure 6, the scaling coefficient g_{bal} controls the ratio between the time-varying and static contact sound components. Finally, the total amplitude of the synthetic contact noise is controlled by the slide velocity $f_c(n)$ via a scaling coefficient g_{TV}. The value $g_{\mathrm{TV}} = f_c(n)/100$ was found to work well. The user can also control the overall volume of the contact sound via an external parameter, g_{user}. For unwound strings, the contact-sound synthesis block is simplified by replacing the noise-burst generator [block (a) in Figure 6] with a white-noise generator and omitting blocks (b), (c), and (d). Thus, the synthesized contact sound for unwound strings is just white noise scaled according to the sliding speed $f_c(n)$.

Figure 7 illustrates the fourth-order contact-sound filter magnitude responses (dotted lines) in comparison with contact-sound spectral estimates, obtained using a linear-prediction filter of order 100. Each row in Figure 7 represents a different string (the sixth, fifth, and fourth strings from top to bottom), while each column represents a different slide tube type (brass, glass, and chrome tubes from left to right). Table 1 lists the pole and zero frequencies and radii for the contact sound filters used in Figure 7.

In conclusion, the slide-guitar synthesis model for a single string is illustrated in Figure 8. It resembles the ordinary time-varying SDL DWG string, but

Table 1. Pole and Zero Frequencies and Radii for the Fourth-Order IIR Contact Sound Filter

		Brass Tube		*Glass Tube*		*Chrome Tube*	
		F (Hz)	*R*	*F (Hz)*	*R*	*F (Hz)*	*R*
6th string		0	0.9485	0	0.9272	±696	0.9608
		0	0.8510	0	0.8222	±1422	0.8042
	zeroes	±1400	0.9079	±1400	0.9608	—	—
	poles	±643	0.9894	±850	0.9957	±748	0.9929
		±1400	0.9922	±1400	0.9984	±1422	0.9937
5th string		0	0.9406	0	0.9646	0	0.9686
		0	0.8105	0	0.7902	0	0.7752
	zeroes	±1600	0.9478	±1640	0.9217	±1640	0.8042
	poles	±793	0.9957	±644	0.9957	±622	0.9859
		±1600	0.9948	±1640	0.9922	±1640	0.9937
4th string		0	0.8727	0	0.9887	0	0.9644
		0	0.7269	0	0.0543	0	0.6564
	zeroes	±2000	0.9687	±1920	0.9826	±2000	0.9217
	poles	±1449	0.9930	±980	0.9720	±859	0.9929
		±2000	0.9948	±1920	0.9948	±2000	0.9922

The corresponding magnitude responses are illustrated in Figure 7.

it has the additional energy compensation and contact-sound generator blocks.

Real-Time Implementation

Because the VSG provides only an auditory feedback of the continuous pitch, the latency between the user's action and the resulting sound should be much smaller than in the VAG. For this reason, a high frame rate (120 frames per second) infrared (IR) camera is used for detecting the user's hand locations. The camera operates by lighting the target with IR-LEDs and sensing the reflected IR light. Therefore, for successful recognition, users must have IR-reflecting material in their hands. A real slide tube coated with IR reflecting fabric is used for detecting the user's fretting hand. Using a real slide tube instead of a glove makes the VSG more intuitive for the user. For recognition of the picking hand, a small ring of IR reflecting fabric is worn on the index finger.

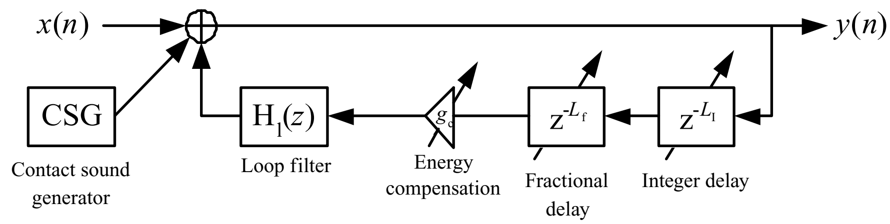
Technical Details

The implementation works on a 2.66-GHz Intel Pentium 4 CPU with 1 GB of RAM and a SoundMax Integrated Digital Audio soundcard. Both the sound-synthesis part and the camera interface operate in the Windows XP environment. The sound synthesis uses Pure Data (PD) (Puckette 1996) version 0.38.4-extended-RC8. The sampling frequency for the synthesis algorithm is 44.1 kHz, except for the string waveguide loop, which runs at 22.05 kHz, as suggested by Välimäki et al. (1996). A Naturalpoint TrackIR 4:Pro USB IR camera is used for gesture recognition. Its output is a 355 × 290 binary matrix, where the reflected areas are seen as "blobs."

Camera API

For the camera API (Application Programming Interface), Naturalpoint's OptiTrack SDK version 1.0.030 was used. The API was modified in the Visual Studio environment to include gesture-recognition features. The added features consist of the distinction between the two blobs (i.e., the slide hand and the plucking hand), calculation of the distance between them, recognition of the plucking and pull-off gestures, and transmission of the control data to PD through Open Sound Control (OSC) messages. Furthermore, an algorithm was added to keep track of the virtual string location,

Figure 8. Slide-guitar synthesis model for a single string.



that is, an imaginary line representing the virtual string. This is similar to the work presented by Karjalainen et al. (2006). The line is drawn through the tube and the averaged location of the plucking hand, so that the virtual string slowly follows the player's movements. The API detects the direction of the plucking hand movement, and when the virtual string is crossed, a pluck event and a direction parameter is sent. Also, a minimum-velocity limit is defined for the plucking gesture to avoid false plucks.

For more realistic playing, a pull-off-feature has been added to the system. This means that the API switches the string length to maximum whenever the slide hand is opened. When the slide hand is closed, the string length is again set according to the distance between the user's hands. Thus, the user can lift the slide tube off the virtual strings, pluck open strings, and then press the tube on the strings again. This is idiomatic for slide-guitar playing. Opening the slide hand makes the tube finger point to the camera such that the slide tube vanishes from the IR camera's view. When the tube is missing, the coordinates where the tube was last seen are used for setting the string's location. In this way, open strings can also be plucked.

Because the slide tube and the fabric ring have quite different shapes, it is easy for the system to distinguish between them. In practice, this is done by selecting the more square-like blob as the ring and the longitudinal blob as the tube. This allows the instrument to be played by left-handed people as well.

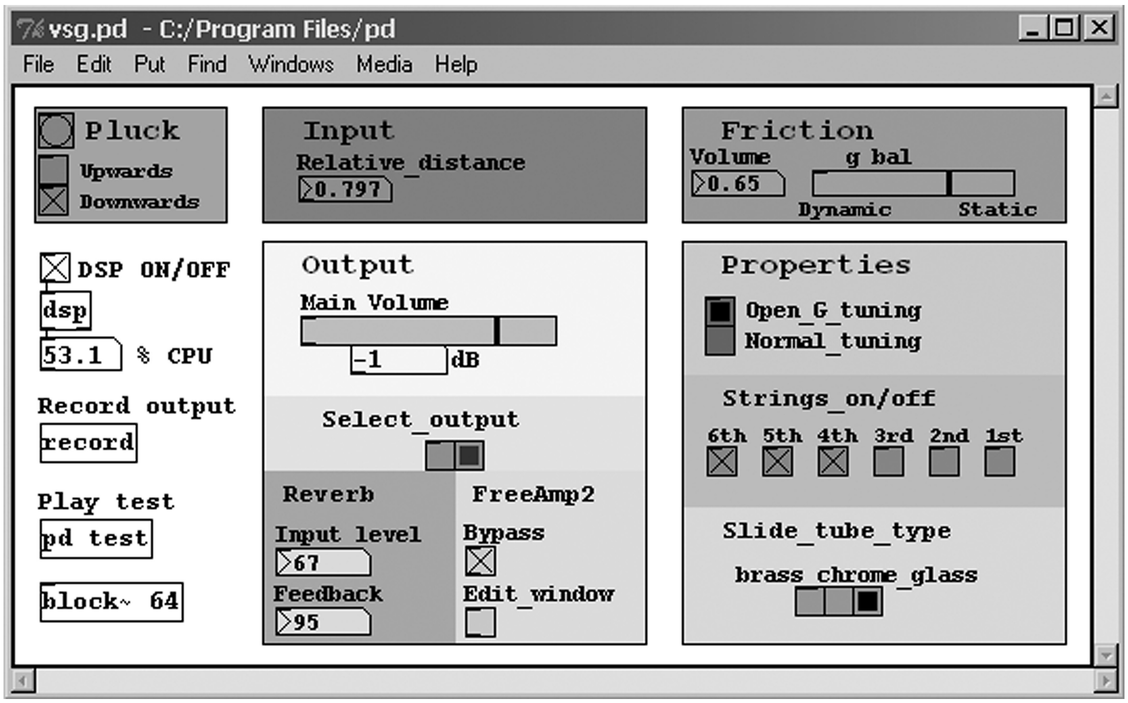
System Calibration and PD Implementation

The system is calibrated so that the distance of 250 pixels corresponds to 48 cm when played approximately 2 m away from the camera. The distance is constrained in such a way that moving the hands further apart than 250 pixels does not make the strings any longer but will map them as open strings. Similarly, the minimum distance between the user's hands is constrained to 62.5 pixels (12 cm when played 2 m away), thus leading to a playing pitch range of an octave and a minor third for each string (from open string to the 15th fret). Because the plucking hand is not normally positioned at the bridge of the guitar but near the sound hole, an offset of 17 cm is added to the distance to obtain the total length of the strings (65 cm for open strings). Next, the distance between the hands is normalized by dividing it with the open string length. This results in a relative string length [$L(n)$ in Figure 6] between 0.446 and 1.0. Because typical use of a slide tube makes every string have the same playing length, this normalized string length is used as a control signal for each of the synthesized strings.

When the PD implementation receives an OSC message containing a pluck event, an excitation signal is inserted into each waveguide string. The excitation signal is a short noise burst simulating a string pluck. There is also a slight delay (20 msec) between different string excitations for creating a more realistic "strumming" feel. The order in which the strings are plucked depends on the plucking direction.

Figure 9 illustrates the PD implementation of the user interface. The overall latency of PD is about 20 msec using ASIO (Audio Stream In/Out) sound drivers. For additional effects, VST (Virtual Studio Technology) plug-ins can be used with PD, but they will naturally add to the computational load of the system. The PD implementation can produce the sound as it is, through a reverb effect (modified from PD audio examples), or through a FreeAmp2 VST plug-in (available online at frettedsynth.asseca.com).

Figure 9. PD implementation of the user interface.



The camera software can be set to show the blob positions on screen in real time. This is not required for playing, but it helps the user to stay in the camera's view. The camera API uses roughly 10 percent of CPU power without the display and 20–40 percent with the display turned on. Because PD uses up to 80 percent of CPU power when playing all six strings, the current VSG implementation can run all six strings in real time without a noticeable drop in performance, provided that the blob-tracking display is turned off. Selecting fewer strings, switching contact sound synthesis off, or dropping the API frame rate to half, the display can be viewed while playing. Using the FreeAmp2 plug-in, only three strings can be played simultaneously.

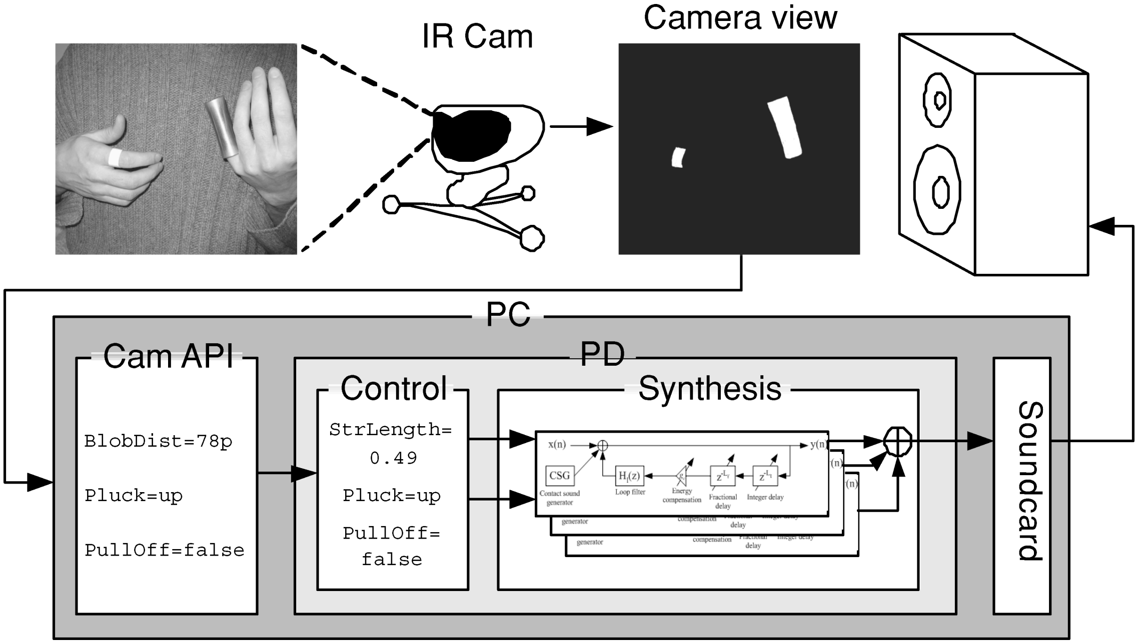
PD allows sub-programs (called sub-patches) to be nested inside the parent patch, and it allows the use of external patches (called abstractions) located in separate files. The whole program consists of three main parts: the main patch (with control options and user interface), string synthesis, and contact-sound synthesis. Sub-patches and abstractions are switched off when they are not needed to save computing time. For example, reducing the contact-sound volume to zero switches all contact-sound computation off.

Because the waveguide loop runs at half the sampling rate, anti-aliasing filters [e.g., two-tap averaging filters $H(z) = (1 + z^{-1})/2$] are required at its input and output. The string-synthesis abstraction converts the relative distance to frequency and delay-line length, calculates the loop-filter parameters depending on string length, as suggested in Välimäki and Tolonen (1998), sends the pluck and contact-sound excitation to the delay line, and implements energy scaling and anti-aliasing filtering. The contact-sound synthesis abstraction receives string parameters through the string-synthesis abstraction and generates contact noise according to the slide-tube type, string properties, and hand movements.

Virtual Slide Guitar

The virtual slide-guitar system is illustrated in Figure 10. The camera API recognizes the playing

Figure 10. Virtual slide-guitar system.



gestures and sends the plucking and pull-off events, as well as the distance between the hands, to the synthesis control block in PD. The synthesis block consists of the DWG models illustrated in Figure 8. At its simplest, the VSG is easy to play and needs no calibration. The user simply puts the slide tube and reflecting ring on and starts to play. For more demanding users, the VSG provides extra options, such as altering the tuning of the instrument, selecting the slide-tube material, setting the contact-sound volume and balance between static and dynamic components, or selecting an output effect.

Generally, the VSG is not as easy to play as the VAG, because there are more freedom and options for the player. The VAG offers the user only a few chords or notes to play and might thus at first sound "better" to the audience, but this severely reduces the expressive range. The additional VSG features add versatility for playing, but a short training session is recommended to get the most out of this virtual instrument. The tube–string contact sound gives the user direct feedback of the slide-tube movement, and the pitch of the string serves as a cue for the tube position. Thus, visual feedback is not necessarily needed to know where the slide tube is situated on the imaginary guitar neck.

Switching between the slide tube types in the VSG results in different contact sounds, but it is difficult to distinguish the tube material by listening to the synthesized sound. This might be because the perceptually most important contact-sound material cue, the frequency-dependent decay rate (Klatzky, Pai, and Krotkov 2000) of the tube is missing from the synthesized sound.

Conclusions and Future Research

In this article, a real-time virtual slide guitar synthesizer has been presented. Energy-compensated time-varying digital waveguides are used for simulating the string vibration. The contact noise between the strings and the slide tube is analyzed from recordings, and a new parametric model for synthesizing it has been introduced. The contact-sound synthesizer consists of a noise-pulse generator whose output is fed into a time-varying resonator and a distorting nonlinearity. By controlling the noise-pulse firing rate, the resonator's center frequency, and the overall dynamics with the sliding velocity, a realistic time-varying harmonic structure is obtained in the resulting synthetic noise. The overall spectral shape of the contact noise is set with a fourth-order IIR filter.

The slide-guitar synthesizer is operated using an optical gesture-recognition user interface, similar to the suggestion by Karjalainen et al. (2006). However, instead of a Web-camera, a high-speed infrared video camera is used for attaining a lower latency between the user's gesture and the resulting sound. This IR-based camera system could also be used for gestural control of other latency-critical real-time applications. The real-time virtual slide guitar model has been realized in PD.

In the current implementation, the longitudinal string vibrations are simulated with fixed filters for computational simplicity. This prevents the modeling of the longitudinal mode spectrum's dependency on the sliding location (Pakarinen, Penttinen, and Bank 2007). The result of this shortcoming is that the spectrum of the contact noise is less dynamic than in reality. In a more sophisticated implementation, this effect could be simulated either by varying the contact-sound filter in time or adding a separate time-varying digital waveguide for modeling the longitudinal modes. Furthermore, the gesture-based user interface could be extended so that the user could, for example, play arpeggios by plucking individual strings. Also, the computational load of the control part could be relieved by operating the camera software on another computer and transmitting the control data over the network. These upgrades are left for future work.

Acknowledgments

The work of J. Pakarinen has been supported by the GETA graduate school, the Cost287-ConGAS action, and the Emil Aaltonen Foundation. The authors are grateful to Professors Marcelo Wanderley and Gary Scavone for helpful discussions.

References

Aramaki, M., and R. Kronland-Martinet. 2006. "Analysis-Synthesis of Impact Sounds by Real-Time Dynamic Filtering." *IEEE Transactions on Audio, Speech and Language Processing* 14(2):695–705.

Aramaki, M., et al. 2006. "A Percussive Sound Synthesizer Based on Physical and Perceptual Attributes." *Computer Music Journal* 30(2):32–41.

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.

Avanzini, F., S. Serafin, and D. Rocchesso. 2005. "Interactive Simulation of Rigid Body Interaction with Friction-induced Sound Generation." *IEEE Transactions on Speech and Audio Processing* 13(5):1073–1081.

Camurri, A., et al. 2000. "EyesWeb—Toward Gesture and Affect Recognition in Interactive Dance and Music Systems." *Computer Music Journal* 24(1):57–69.

Camurri, A., et al. 2005. "Communicating Expressiveness and Affect in Multimodal Interactive Systems." *IEEE Multimedia* 12(1):43–53.

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

Fontana, F. 2003. "Physics-Based Models for the Acoustic Representation of Space in Virtual Environments." Ph.D. dissertation, University of Verona, Italy. Available online at http://profs.sci.univr.it/~fontana/paper/20.pdf.

Gorman, M., et al. 2007. "A Camera-Based Music-Making Tool for Physical Rehabilitation." *Computer Music Journal* 31(2):39–53.

Johnson, R. 1990. "The Complete Recordings." Audio compact disc. New York: Legacy Recordings C2K46222.

Karjalainen, M., et al. 2006. "Virtual Air Guitar." *Journal of the Audio Engineering Society* 54(10):964–980.

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.

Klatzky, R. L., D. K. Pai, and E. P. Krotkov. 2000. "Perception of Material from Contact Sounds." *Presence: Teleoperators and Virtual Environment* 9(4):399–410.

Laakso, T. I., et al. 1996. "Splitting the Unit Delay—Tools for Fractional Delay Filter Design." *IEEE Signal Processing Magazine* 13(1):30–60.

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

Pakarinen, J., H. Penttinen, and B. Bank. 2007. "Analysis of the Handling Noises on Wound Strings." *Journal of the Acoustical Society of America* 122(6):EL197–EL202.

Pakarinen, J., et al. 2005. "Energy Behavior in Time-Varying Fractional Delay Filters for Physical Modeling of Musical Instruments." *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, Vol. 2. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, pp. 1–4.

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

Peltola, L., et al. 2007. "Synthesis of Hand Clapping Sounds." *IEEE Transactions on Audio, Speech, and Language Processing* 15(3):1021–1029.

Penttinen, H., et al. 2006. "Model-Based Sound Synthesis of the Guqin." *Journal of the Acoustical Society of America* 120(6):4052–4063.

Puckette, M. 1996. "Pure Data." *Proceedings of the 1996 International Computer Music Conference.* San Francisco, California: International Computer Music Association, pp. 269–272.

Rath, M., and D. Rocchesso. 2005. "Continuous Sonic Feedback from a Rolling Ball." *IEEE Multimedia* 12(2):60–69.

Rocchesso, D., R. Bresin, and M. Fernström. 2003. "Sounding Objects." *IEEE Multimedia* 10(2):42–52.

Rocchesso, D., and F. Fontana, eds. 2003. *The Sounding Object*. Florence, Italy: Edizioni di Mondo Estremo. Available online at www.soundobject.org.

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

Välimäki, V., et al. 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., and T. Tolonen. 1998. "Development and Calibration of a Guitar Synthesizer." *Journal of the Audio Engineering Society* 46(9):766–778.

Välimäki, V., et al. 2006. "Discrete-Time Modelling of Musical Instruments." *Reports on Progress in Physics* 69(1):1–78.

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