# Contact Interaction with Integrated Audio and Haptics

**Derek DiFilippo** and **Dinesh K. Pai**
Department of Computer Science
University of British Columbia
2366 Main Mall
Vancouver, BC V6T 1Z4, Canada
+1 604 822 6625
{difilip|pai}@cs.ubc.ca

**ABSTRACT**

We have developed a computer interface that simultaneously renders integrated auditory and haptic stimuli with very low latency. The interface includes a three degree of freedom (DOF) Pantograph haptic device that reads position input and renders force output. We synthesize audio by convolving the force profile generated by user interaction with the impulse response of the virtual surface. Thus, both auditory and haptic modes are tightly coupled. This system should aid in evaluating efficient algorithms for integrated rendering of both sensory modes. We will demonstrate the interface during the conference.

**Keywords:** User Interface, Haptics+Audio

## INTRODUCTION

When two objects collide, we have a contact event. Everyday lives are full of them: placing a cup on a table, tapping our fingers on a computer keyboard, etc. All of these contact events generate characteristic sounds and forces that communicate information about the object and the environment. An ideal computer interface for interacting with a virtual environment should be able to produce similar sounds and forces. In this paper, we present an experimental interface for displaying sounds and forces with low latency and high realism. The novelty of this interface lies in the tight integration of the auditory mode and the haptic ("touch") mode. User interaction with the simulated environment generates contact forces. These forces are rendered to the hand by a haptic force-feedback device, and to the ear as contact sounds. This is more than simple synchronization. The audio and the haptics change together when the user applies different forces to the object.

Haptic interfaces, like auditory displays, help the user modify and extract information about their environment [1]. Some applications for haptic interfaces include surgical simulators for medical training, mechanical manipulation of CAD systems, flight simulators, and computer games that allow the user to interact with virtual objects. If it is important to represent a contact event between the user and the environment, then haptic interfaces can help. It is also important to consider synthesis techniques for realistic audio if this contact occurs between two rigid objects. Our interface atomically generates both haptic and audio stimuli from contact events, rather than triggering a pre-recorded audio sample or tone when a collision is detected. This paper describes the interface hardware for user input, a simple model for calculating haptic and audio contact forces, and the algorithms for rendering these forces. We use the UBC ACtive MEasurement facility to capture sound samples and estimate audio synthesis model parameters of the objects we wish to simulate. This is described in another ICAD conference paper [2].

Figure 1: The Pantograph haptic interface.

## HARDWARE

The key piece of hardware in our system is a 3 degree of freedom Pantograph device (Figure 1). The 5-bar mechanism is based on a design by Hayward but extended to 3DOF to our specification [3]. It reads three degrees of freedom of position as user input, and renders three degrees of freedom of forces as output. The user can move the handle in the plane as well as rotate the handle. There are two large Maxon motors attached to the base of the Pantograph which apply forces on the handle via the 5-bar linkage; a small motor in the handle can exert a torque on the handle as well. The device, therefore, is complete for rigid motions in the plane, i.e., it can render the forces and torque due to any contact with a rigid body attached to the handle in a planar virtual world ("flatland").

The haptic device is controlled by a dedicated motion control board (MC8, Precision Microdynamics), hosted in a PC running Windows NT. The board has 14 bit analog to digital converters (ADCs) for reading the potentiometers attached to the base

motors, as well as quadrature decoders for reading the encoder which measures the handle rotation.
Real-time synthesis of sounds and forces is performed with a SHARC DSP (Analog Devices 21061 running at 40MHz) on the same board. Output voltages for controlling the Pantograph motors and for rendering audio are sent out through 14 bit digital to analog converters (DACs). The audio waveforms can be sent directly to a sound system; in our set up they are fed back into the soundcard of the computer for ease of capture and playback.
By using this specialized hardware, we bypass the complications that arise from balancing the needs of real-time, deterministic response and ease of access from user-level software on a widely available operating system such as NT. Our code is compiled for the DSP and has exclusive control over its resources. This allows us to precisely time our control algorithms, as well as accurately diagnose inefficiencies and bugs.

**AUDIO SYNTHESIS**

We wish to simulate the audio response of everyday objects made out of wood, metal, ceramic, etc. Contact with these objects can be characterized by impulsive excitation of relatively few exponentially decaying, weakly coupled sinusoidal modes. Modal synthesis and impulse generation techniques have been developed for these types of percussive sounds [4]. We use the modal audio synthesis algorithm described in [5]. This algorithm is based on vibration dynamics and can simulate effects of shape, location of contact, material, and contact force. Model parameters are determined by solving a partial differential equation, or by fitting the model to empirical data. The sound model assumes that the surface deviation $y$ obeys a wave equation. We add a material-dependent decay coefficient to the wave equation to damp the sounds. The exponential damping factor $d = f\pi \tan\phi$ depends on the frequency $f$ and internal friction $\phi$ of the material, and causes higher frequencies to decay more rapidly. The internal friction parameter is material dependent and approximately invariant over object shape. Equation 1 represents the impulse response of a general object as a sum of damped sinusoids.

$$y(t) = \sum_{n=1}^{\infty} a_n e^{-d_n t} \sin(\omega_n t) \tag{1}$$

The sound model of an object consists of a list of amplitudes $a_n$ and complex frequencies $\Omega_n = \omega_n + id_n$. Equation 2 shows how one complex frequency is computed. At time $0$, the signal is the product of the frequency-amplitude $a_n$, and the contact force $F(0)$. At each successive time step, the signal is the sum of a decayed version of the previous signal, plus a new product of amplitude and contact force. The model responds linearly to input force $F(k)$. Once we have the model parameters, all we need to begin synthesizing sounds is a series of contact forces to plug into the right-hand side of the recursion. The output signal at time $k$ is $\mathbb{R}(\sum y_n(k))$, the sum over all computed frequencies.

$$\begin{aligned} y_n(0) &= a_n F(0) \\ y_n(k) &= e^{i\frac{\Omega_n}{Fs}} y_n(k-1) + a_n F(k) \end{aligned} \tag{2}$$

This synthesis algorithm has two benefits. First, it is linear. The computed audio is the discrete convolution of the force history with the impulse response. There is a natural relationship between the input forces and the output signal; this relationship would not be as straightforward if we used sample-based synthesis. The linearity also makes it efficient. Each complex frequency can be computed with 3 multiplications and 4 additions per sample. Second, the audio quality can degrade gracefully. If DSP time is running short we can compute fewer frequencies, resulting in a loss of detail rather than audio dropouts.

**HAPTIC FORCE SYNTHESIS**

As the user moves the Pantograph handle we need to compute the contact forces resulting from these interactions, and then render them as forces on the handle of the Pantograph by exerting torques on its base joints. These computations take place in two coordinate frames. One is the world frame of xy-coordinates, and the other is the Pantograph frame of joint angles. The simulated environment uses the world frame, but the control code only knows about joint angles. We need a forward kinematic mapping that gives the xy-position of the handle as a function of base joint angles, as well as a differential kinematic mapping that gives the base joint torques as a function of applied force to the handle.
For the forward kinematic mapping, we specify the base joint of motor 1 as the origin of the world frame. There is a simple geometric constraint that allows us to simply compute the position of the handle: the vector pointing from elbow 1 to elbow 2

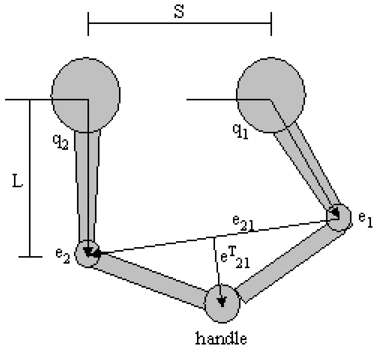


Figure 2: Pantograph kinematics.

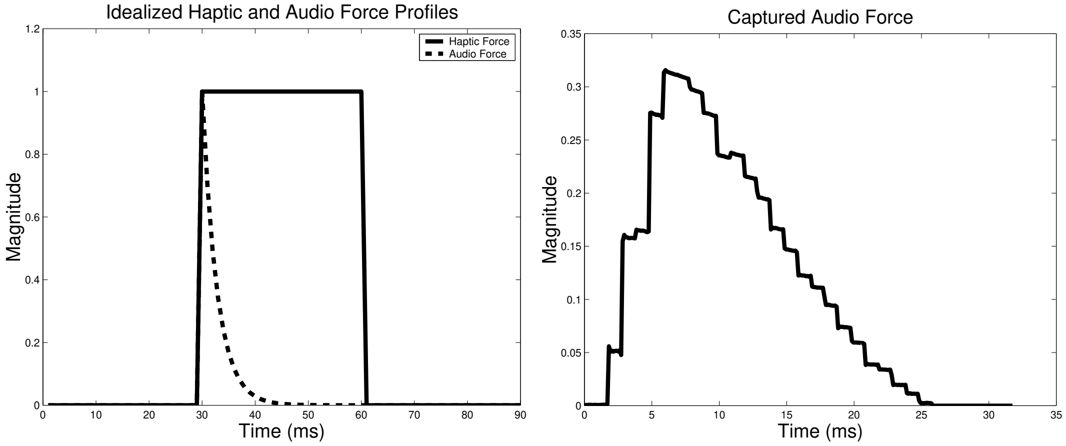


Figure 3: (a) Idealized haptic and audio force profiles. (b) A captured audio force profile.

($e_{21} = e_2 - e_1$) in the world frame is always perpendicular to vector pointing to the handle from the midpoint of $e_{21}$. If $q_1$ and $q_2$ are the base joint angles, then the elbows become $e_1 = (L\cos(q_1), L\sin(q_1))$ and $e_2 = (L\cos(q_2) + S, L\sin(q_2))$ where $L$ is the length of the proximal arms, and $S$ is the separation between the two base joints. Setting $e_{21}^{\perp}$ as the vector pointing from the midpoint of $e_{21}$ to the handle $h$, we have $h(q) = e_1 + 0.5e_{21} + e_{21}^{\perp}$. This expression for $h$ in terms of joint angles $q$ has a simple geometric interpretation, as shown in Figure 2.

Once we have the handle coordinates, and compute a contact force $F$, we need to transform this force into base joint torques $\tau$ for rendering. The Jacobian $J = \partial h(q)/\partial q$ of the forward kinematic mapping relates forces to torques by $J^T F = \tau$. The details of constructing the Jacobian for the Pantograph are quite general and are covered in basic robotics texts [6]. In our particular implementation, we can avoid the expense of computing the partials of $h(q)$ by exploiting the structure of the Jacobian. Details are removed here for the sake of brevity.

In this paper, we only consider interactions normal to the surface of a plane. A spring/damper combination constrains the user to the surface. If the normal displacement past the surface is $x_n$, and the current normal velocity is $v_n$, then the haptic constraint force is $F = Kx_n + Dv_n$ where $K$ and $D$ are spring and damping constants. The algorithms and techniques we develop for the plane should be easily incorporated with more sophisticated applications that manage their own collision detection between complex geometries.

## AUDIO FORCE SYNTHESIS

Naively using the raw haptic forces to synthesize audio produces horrible results. There are two main properties of our synthesized haptic forces that cause trouble. This section will describe how we filter out these two properties from the haptic force. The filtered result is the *audio force* that we convolve with the stored impulse response in equation 2. The two main properties of the haptic force that we wish to filter are as follows: (1) a spurious impulse results when the user breaks contact with the surface and the haptic force discontinuously drops to zero, and (2) high frequency position jitter.

Figure 3(a) plots an idealized haptic contact force. At 30ms the user comes into contact with the surface and stays in contact for another 30ms. Convolving this square wave profile with the impulse response of the surface will produce a spurious second "hit" when the user breaks contact. We introduce an attenuation constant $\beta$ to allow the audio force to smoothly move to zero during sustained contact. If $t$ is the elapsed time since contact, then the current audio force is the current haptic force attenuated by $\beta^t$. We have found that setting $\beta = 0.85$ (halflife of 5ms) produces good results.

Haptic instabilities and signal noise generate sustained low amplitude, high frequency jitter in the position readings. These high frequencies are passed into the haptic force profile by the linear spring constant. Without filtering, this noise becomes audible as excessive crackling and popping when the user contacts the surface. This low amplitude noise can be removed by truncation. Typically, we remove the 8 lowest order bits. This dramatically improves the audio signal. Figure 3(b) plots a typical audio force profile. The signal is not perfectly constant during each millisecond interval because it was captured as the input signal to a soundcard.

## REAL-TIME SIMULATION

The basic control structure for our real-time synthesis and simulation is interrupt-driven. There is a haptic interrupt service routine that generates haptic and audio forces, and an audio interrupt that convolves the audio force with the impulse response of the modelled object. Using these two separate interrupts, we can synthesize the audio signal at a much higher rate than we generate haptic feedback. This section will describe the two interrupt routines shown in Figure 4.

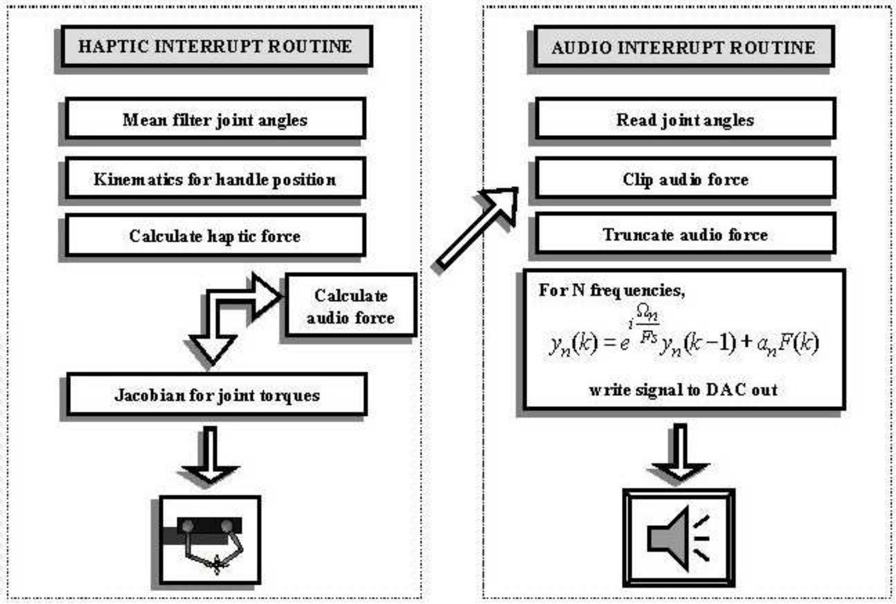


Figure 4: Flow of control for real-time synthesis and simulation.

The audio interrupts and all DAC/ADC latches are synchronized to trigger at the audio control rate. The audio interrupt routine reads the Pantograph joint angles from the ADCs and stores them in an array that contains a history of joint angle readings. Converting the DAC input to equivalent a floating point number requires 1 comparison, 2 multiplications and 2 additions. The current audio force is clipped to lie between 0.0 and 1.0, and then truncated to remove low amplitude noise. This requires 2 comparisons and 2 multiplications. A discrete convolution step using this filtered audio force $F(k)$ produces the output audio signal $y_n(k)$. This signal is placed in the DAC out. Computing the audio signal requires 3 multiplications and 3 additions per complex frequency. If the DSP is short on cycles, we can decrease the number of active frequencies.

Haptic interrupts trigger at a fraction of the audio control rate. The current joint angles are the mean of the array of joint angles captured during the audio interrupt routine. From these filtered values, we use the forward kinematics of the Pantograph to compute the position of the handle. Since we only consider interactions normal to the plane, determining contact between the handle and the plane takes a sign check. If there is contact, we compute the resulting haptic force using a spring/damper model. Finding the current Jacobian takes 22 multiplications, 4 trigonometric calls, and 2 square roots. The Jacobian of the Pantograph translates the haptic force into motor torques. The voltages to generate these torques are written to the DACs. If there has been contact for $t$ milliseconds, then $\beta^t$ times the haptic force becomes the current audio force. The haptic interrupt writes the current audio force to a global variable shared with the audio interrupt routine.

**RESULTS/EVALUATION**

We have coded a simple environment using the basic control structure just described. The user scrapes the Pantograph handle across a sinusoidally modulated surface profile. We convolve the resulting audio force with the impulse response of a brass vase acquired from the University of British Columbia ACtive MEasurement facility.

This simple environment has been informally tested in our laboratory. Haptic interrupts trigger at 1kHz and audio interrupts at 20kHz. Thus, there is a 1ms latency for changes in force and audio. The auditory and haptic stimuli are perceptually simultaneous. Figure 5 plots a captured audio force and the convolution of this force with the measured impulse response of the vase. We compute 20 modes for each audio sample. The user interaction in this example was five single strikes of increasing force normal to the surface, then tangential motion across the surface. The middle two bursts are slower scrapes back and forth, and the final two bursts are faster scrapes.

**FUTURE WORK**

This section will consider two items for future work. We discuss adding texture and friction to our model, as well possible opportunities for perceptual studies of integrated audio and haptics. Both of these items have been previously addressed in the haptics community – our future work will use this knowledge to improve our haptic and audio integration.

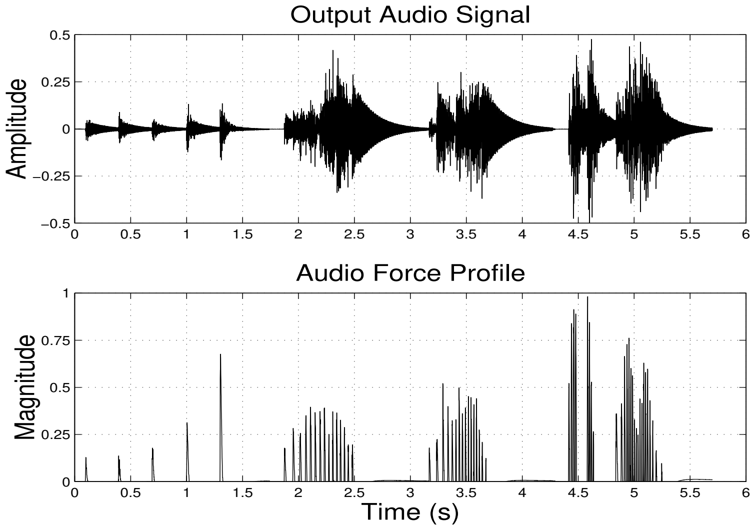


Figure 5: Output audio signal and audio force magnitude. We subsampled the data when making the figure, thus the peak of the third contact is not accurately represented.

Our current simulation generates audio forces from haptic forces normal to a locally flat patch. Large scale surface features can consist of a collection of polygons which use our flat patch algorithms after collision detection. An obvious extension would be to incorporate small scale local surface features into our model. This would include tangential forces generated by surface textures and surface friction. Stochastic models for haptic textures based on filtered noise and fractional Brownian motion are well known in the haptics community [7, 8]. For example, perturbing the normal force by sampling from a Gaussian distribution generates haptic sandpaper. Roughness is positively correlated with the variance of the distribution.

Hayward and Armstrong have developed a stick-slip friction model that produces realistic viscoelastic (creeping, sticking, slipping) behaviour [9]. Synthesized signals of this type are available at <http://www.cim.mcgill.ca/ haptic/squeaks.html>. This discrete, online, friction model has been applied to the real-time physical modelling of a violin [10]. Computing friction force as a function of handle displacements only requires adding a comparison and a few multiplications to the haptic interrupt routine. A good test for adding friction to our audio synthesis routine would be to simulate rubbing the rim of a wine glass and forcing it to resonate.

Previous studies on the influence of auditory stimuli on haptic perception have used audio samples or tones triggered by contact events. The study by Miner, et al, used pitched tones and attack envelopes to simulate hard and soft sounds [11]. They found that "the auditory stimulus did not significantly influence the haptic perception". The study by DiFranco, et al, triggered audio samples of contact events they recorded by hand [12]. They found that "sound cues that are typically associated with tapping harder surfaces were generally perceived as stiffer". Both of these studies focus on the perception of hardness, which is a function of force on the users hand. These studies do not explore the perception of surface roughness, which in addition to being a function of force, is also a non-trivial function of speed [13]. Because our approach more closely couples the auditory stimulus with the underlying physical process of collision, the simple sinusoidal grating described above produces auditory textures that vary as a function of both force and speed. In addition, we believe our interface could be used for exploring the perceptual limits of synchronization between auditory and haptic events. As our contact simulation matures, we hope that integrated audio and haptics will be useful for perceptual studies of multimodal display.

## CONCLUSION

Designing compelling simulated environments is the high-level goal of this research. The representation and rendering of contact events comprises an essential component of any such simulation. Atomically representing the contact event as something that produces both sound and force helps integrate auditory and haptic stimuli. We believe that this atomicity is a natural way to think of simulating contact. The preliminary results from the interface we have implemented for rendering contact events encourages us to continue research in this direction.

## REFERENCES

1 Srinivasan, M.A., Basdogan C., Haptics in Virtual Environments: Taxonomy, Research Status, and Challenges. *Comput. & Graphics 4*, 21 1997, 393–404.

2 Richmond, J.L., Pai, D.K. Robotic Measurement and Modeling of Contact Sounds. *International Conference on Auditory Display*, Atlanta GA, April 2000.

3 Ramstein, C. Hayward, V. The Pantograph: a large workspace haptic device for a multi-modal Human-computer interaction. *Conference on Human Factors in Computing Systems ACM/SIGCHI*, Boston MA, April 1994.

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

5 van den Doel, K., Pai, D.K. The Sounds of Physical Shapes. *Presence 7:4, 1998*, The MIT Press, 382–395. Available as <http://www.cs.ubc.ca/spider/pai/papers/vanPai96.ps.gz>.

6 Murray, R.M., Li, Z., Sastry, S.S. *A Mathematical Introduction to Robotic Manipulation.* CRC Press, Ann Arbor MI, 1994.

7 Siira J., Pai D.K. Haptic Textures – A Stochastic Approach. *IEEE International Conference on Robotics and Automation*, Minneapolis MN, April 1996.

8 Fritz, J.P., Barner K.E. Stochastic models for haptic texture. *Proc. SPIE International Symposium on Intelligent Systems and Advanced Manufacturing*, Boston MA, November 1996.

9 Hayward, V., Armstrong, B. A new computational model of friction applied to haptic rendering. *Experimental Robotics VI*, Lecture Notes in Control and Information Sciences, Springer-Verlag, Vol. 250 403–412.

10 Serafin, S., Vergez, C., Rodet, X. Friction and Application to Real-time Physical Modeling of a Violin. *International Computer Music Conference*, Beijing, October 1999.

11 Miner, N., Gillespie, B., Caudell, T. Examining the Influence of Audio and Visual Stimuli on a Haptic Display. *Proceedings of the 1996 IMAGE Conference*, Phoenix AZ, June 1996. Available as <http://users.aol.com/nadine505/papers/img96.htm>.

12 DiFranco, D.E., Beauregard, G.L., Srinivasan, M.A. The Effect of Auditory Cues on the Haptic Perception of Stiffness in Virtual Environments. *Proc. ASME Dynamic Systems and Control Division*, 1997.

13 Lederman, S.J., Klatzky, R.L., Hamilton, C.L., Ramsay, G.I. Perceiving Surface Roughness via a Rigid Probe: Effects of Exploration Speed and Mode of Touch. *The Electronic Journal of Haptics Research* , Vol. 1, No. 1, October 1999. Available as <http://www.haptics-e.org/Vol_01/he-v1n1.pdf>