

INTERACTIVE GRANULAR SYNTHESIS OF HAPTIC CONTACT SOUNDS

STEPHEN BARRASS AND MATT ADCOCK

CSIRO Mathematics and Information Sciences, GPO Box 664, Canberra, Australia, 2601
 Stephen.Barrass@csiro.au, Matt.Adcock@csiro.au

The goal of this project is to automatically synthesise contact sounds that can be easily added to existing and future haptic applications without the need for extensive sound design, sampling or programming. Contact sounds are caused by interactions with objects such as scraping a surface. This continuous information about the surface is much richer than simply triggering a sound sample.

In this paper we investigate an “ecological” granular synthesis algorithm that has been used to simulate the sounds of objects made of different materials breaking, bouncing and being scraped. Granular synthesis algorithms can generate a wide range of complex sounds from a pool of sampled grains. We implemented granular synthesis on a Windows PC using the OpenAL API and hardware acceleration on the soundcard to achieve 1000 grains per second. The algorithm was integrated into the Reachin API and rendered from the 1kHz haptics thread with latency between haptics and sound below the 2ms threshold of perception. We developed an initial simple mapping of haptic interaction parameters to the excitement of grains based on tests with four distinct types of haptic surfaces. We found we could hear information such as the force, rate and timing of scraping gestures on the surface as well as properties of the surface such as stiffness, friction, roughness and bumpiness.

INTRODUCTION

The haptic workbench is a human-computer interface that allows the user to see and feel 3d virtual objects in a hand-sized virtual reality environment. The workbench consists of a Phantom haptic device collocated in a stereo-visual graphic interface under a mirror (Fig. 1).

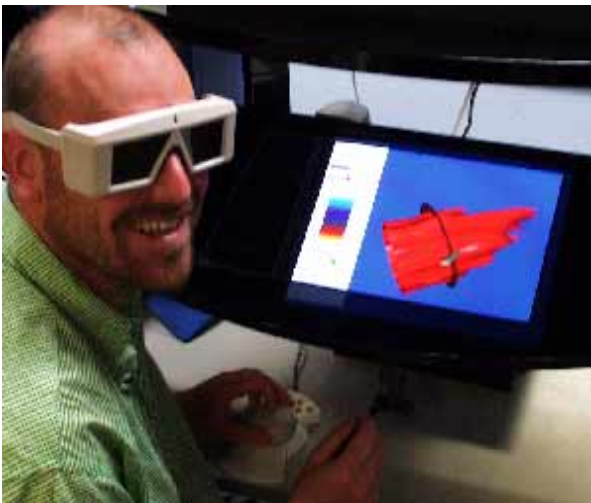


Figure 1: Haptic Workbench.

Over the past few years we have been researching haptic interfaces and developing applications in areas such as mining exploration, surgical training, and collaborative sculpting [5,8,13]. These applications are built on the Reachin application programming interface (API) [11] which has a haptic-visual scene-graph and supports sound through the OpenAL sound API [9]. We have previously added sound to our haptic applications

in a conventional way by triggering samples from special events – for example in a catheter trainer the verbal feedback “try a steeper angle” or “success” informs the trainee about their performance with the needle. Although the sounds are readily accessible in the API not many applications actually have sounds in them. From our experience we believe this is because it takes considerable effort and expertise to develop a sample based auditory interface specifically for each application. Furthermore the triggering of sampled sounds provides only very basic categorical or Boolean information that does not provide sufficient advantages over the non-auditory interface.

In this paper we describe our work on the automatic synthesis of sounds to provide information about continuous interactions with surfaces of objects in the haptic workbench. The goal of the work is to make it quick and easy to add effective sounds to existing or future haptic applications, without the need for specialist sound design or user interface design expertise.

In the following sections we describe previous work on contact sounds, our own work on granular synthesis of contact sounds, an experiment on interactive granular synthesis using a Windows PC system, details of the integration of granular contact sounds with the haptics system, an initial mapping from haptic interaction parameters to granular synthesis parameters with observations on different types of surfaces, and finally directions for future work.

1 BACKGROUND

The first examples of synthesized contact sounds in computer user interfaces were Gaver's Auditory Icons for the Apple desktop GUI [3]. These sounds were designed to provide information:

- about previous and possible interactions,
- to indicate ongoing processes and modes,
- for navigation,
- and to support collaboration.

Auditory Icons are designed with an "ecological" method where the sounds of objects and events in the interface are modelled on real everyday objects and events. For example the sound of a data file being dragged across the graphic desktop is modelled on the sound of a paper file being dragged across a wooden desktop. Gaver proposed a modal synthesis algorithm to generate the sounds of different materials impacting, bouncing, breaking, and spilling. The parameters of the algorithm, which consists of a weighted sum of exponentially damped sinusoids, are controlled by the following higher level perceptual correlates :

- mallet hardness which controls initial amplitudes of partials;
- material type which controls the damping of partials, for example wood has highly damped upper partials;
- and shape which controls the relations between partials.

Gaver modelled wooden and metal bars of varying lengths by analysing the spectrograms of real wooden and metal bars struck with a rubber mallet. He generalized from impacts to scraping of surfaces by proposing a Filter Bank algorithm to separate the characteristics of the interaction from the model of the object. The object is modelled by the weighted sum of a bank of one-pole filters with variable frequency and bandwidth. The interaction is modelled by an impulse response used to excite the filter bank. Impacts have impulsive spectra while scrapes are modelled by noise.

The haptic interaction with an object is used by Ruspini and Khatib to define classes of sounds that are particularly interesting in haptic interfaces [12]:

- impact sounds caused by striking an object;
- contact sounds caused by sliding in continuous contact with a surface;
- event sounds caused by dynamic interaction with an object such as slamming a door.

They investigated haptic impact sounds using a Phantom haptic device that has six degrees of freedom (6DOF) for position and rotation of the stylus, and 3DOF force feedback through motors that resist the movement of the stylus held by the user [10]. The sound

is generated by triggering enveloped samples on a soundcard. The envelope has an exponential decay

$$f(t) = Ae^{-dt} \quad (1)$$

Haptic impacts were mapped to parameters of the envelope as follows:

- change in velocity of the haptic proxy at the time of impact maps to the initial amplitude A of the sample envelope;
- material of the object maps to the damping d of the exponential sample envelope, so for example metal has low damping and wood has high damping;
- greyscale image texture can be used as a spatial parameter map to a signal processing effect such as the centre frequency of a band pass filter applied to the sample.

Multiple sounds generated by micro-impacts and bounces need to be pruned using a threshold to prevent performance degradation due to channel overloading. Ruspini and Khatib note that sampling of sounds to create soundfonts and the building of parameter textures is a tedious process, especially if there are more than a few objects in the application.

Rather than using samples, DiFilipo and Pai followed Gaver's approach and used modal synthesis for haptic contact sounds for a haptic interface using a Pantograph haptic device with 3DOF spatial input and 3DOF force feedback [2]. The parameters of the modal model were analysed from the impulse response of a real brass vase, to accurately capture acoustic effects due to shape, material and hollowness. Interaction with the virtual object is captured from the haptic forces and convolved with the modal model to compute the audio output waveform. The haptic force profile is filtered to produce an "acoustic force profile" that has a damping factor to reduce impulses caused by breaking contact with the surface, and also uses truncation to reduce jitter due to noisy input of spatial position. The haptic-acoustic system has two separate interrupt-driven processes calculating haptic forces at 1kHz and synthesising audio at 20kHz. This system guarantees 0.5ms latency between haptic and audio renderings. This system was used in an experiment where it was found that people could not perceive a latency of 2ms between the haptics and sound renderings. This figure is much lower than the 66ms found in a study on impact sounds [2].

The modal synthesis technique can accurately represent real objects but the contact sounds it produces are tightly coupled to the original object and each different object in the interface has to be uniquely modelled. The sounds of impacts, scrapes, bounces and breaking synthesised by Gaver have also been produced "with

perceptually more satisfying” results by Keller and Truax using a technique called granular synthesis [6]. This technique can produce a wide range of complex sound “textures” from a pool of simple sound “grains”. The grains are typically built from very short 3-30ms samples or synthesised tones that can be parameterised by pitch, gain and duration. The properties of the higher level sound emerge from the organisation of the grains in “streams” that overlap and vary in density and timing, as well as the mixture of different grains. Streams of grains are classified as synchronous if the grains repeat at regular intervals, and asynchronous if the spacing is irregular (Fig. 2).

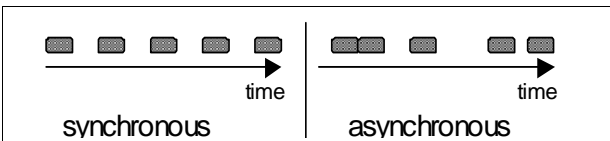


Figure 2: Classification of grain streams (after [6]).

A grain stream can be characterised by the amount of time grains are active, which is called the *fill factor*, calculated by multiplying the average number of grain onsets per unit time by the average duration of the grains.

$$fill_factor = onsets \times duration \quad (2)$$

Parallel streams of grains are classified as phase-synchronous if grains across streams are triggered in step, and phase-asynchronous if grains across streams are independent (Fig. 3). The sum of parallel streams is called the coincidence stream.

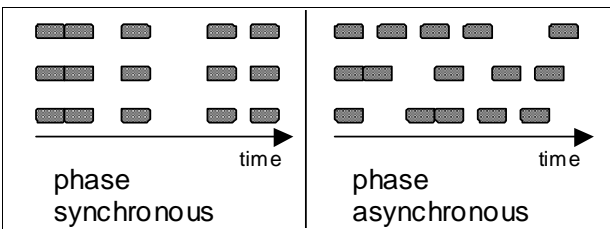


Figure 3: Classification of parallel streams (after [6]).

In their “ecologically based” granular synthesis (EGS) Keller and Truax separate meso-level control functions from micro-level parameterisation of the grains. The control function is analysed from the temporal patterning of the event to be simulated – for example, a scrape function is modelled on the temporal gesture of someone scraping a surface. The pool of grains is drawn from samples of real scraping sounds that have a timbre that establishes the material of the surface being scraped. The parameters of the EGS algorithm are shown in Table 1.

Parameter	Description	Computation
Onset i	Beginning of grain	Temporal distribution
End i	End of grain	
Dur i	Grain duration	Filling factor
Env i	Grain amplitude envelope (window)	Spectral profile
Sample i	Grain sound file	Sample pool
Channel i	Source sound file channel	Stereo placement
Onset($i+1$)	Beginning of next grain	
Del i	Delay between end i and onset($i+1$)	Δ Onset $i -$ Dur i
Δ Onset i	Delay between consecutive grains	Onset($i+1$) – Onset i
Overlap(t)	Number of active grains at time t	Σ grains(t)
Δ Phase i	Phase offset for grain i between stream n and contiguous stream	Onset $i(n) -$ Onset $i(n+1)$
Onset s	Beginning of stream	Attack synchronicity
Sample s	Sample pool for stream	Sound class
Path s	Structure for spatial placement of stream	Spatial localisation
Energy(t)	Energy at time t	Excite(t) – Damp(t)
Damp(t)	Damping coefficient	
Excite(t)	Energy input coefficient	

Table 1: Parameters of EGS (after [7]).

2 INTERACTIVE GRANULAR SYNTHESIS

Granular synthesis is suited to computer processing due to the large number of parameters and small time scales involved. However the large amount of processing required has long meant that granular synthesis could not be computed at interactive rates. Recent advances in processing power have reached the point where interactive rates are now possible – for example we used an SGI Onyx2 with eight parallel processors to synthesise ten spatialised grain streams in an interactive interface for sonification of well-log data [1]. These advances have flowed-on into personal computer systems and there are now many examples of PC-based interactive granular synthesis instruments.

These developments lead to the possibility that we can implement EGS at interactive rates on a Windows PC. We are interested in the ecological algorithm because the separation of meso-level control function from micro-level grain parameterisation corresponds with the different timescales of the graphic and haptic threads.

To test this idea we implemented a granular synthesis engine on a Windows PC and characterised the interactive ranges for EGS parameters. The system is a dual 1.5GHz Intel Pentium III with a Creative Labs Soundblaster Audigy soundcard running under Windows 2000. The granular synthesis engine is implemented with the Open Audio Library (OpenAL) API designed for specifying 3D sound sources that move around a listening position. The Windows implementation of OpenAL takes advantage of

hardware acceleration on the soundcard to provide 32 sound sources at interactive rates on our system. Each sound source plays a wave-table (memory buffer) on the soundcard. We loaded a wave-table with a 10ms down-ramp sound grain (Fig. 4) that has a sharp leading edge that can be readily detected in measurements.

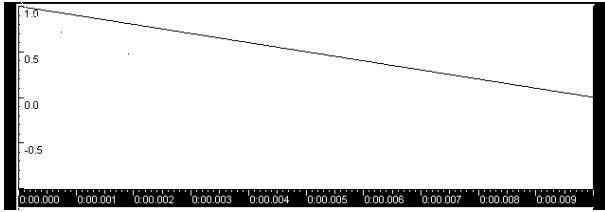


Figure 4: Sound grain with 10ms down-ramp envelope.

The grain was allocated to a sound source and played 1000 times consecutively inside a timed loop. The time ranged from 1260ms to 1280 ms, consistently less than 1.3ms per grain. The audio output is a series of 999 square-ish shapes at regular intervals, followed by a final 10ms down-ramp (Fig. 5).

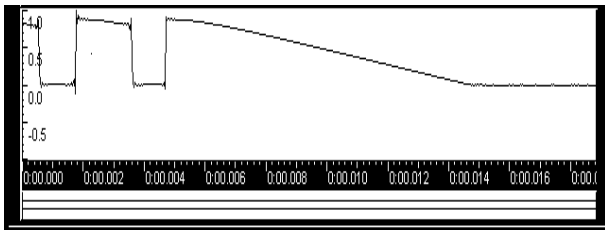


Figure 5: Tail end of a stream of consecutive interrupted 10ms down-ramps.

The square-ish shapes are down-ramps that have been interrupted by the start of the next grain. The delay between onsets of consecutive interrupted grains is a measure of the total system latency to synthesise a grain, which is typically 2ms. The delay between the end of an interrupted grain and the start of the next is the hardware latency between resetting the wave-table index and beginning of synthesis of the next grain, which is typically less than 1ms.

In order to measure the phase offset between streams (ΔPhase_i), we loaded the 10ms down-ramp into all 32 sound sources, and played them simultaneously using the OpenAL `alSourcePlayv()`. The audio output shows a series of overlapping triangular grains that is the sum of the 32 parallel streams (Fig. 6).

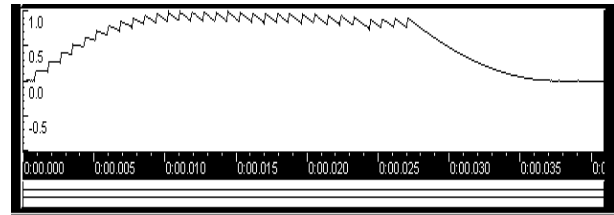


Figure 6: 32 parallel overlapping 10ms down-ramps.

The delay between onsets of grains is 1ms and each down-ramp plays to full duration without being interrupted. The maximum duration of a grain is limited by the number of sound sources that can be allocated to it divided by the delay between onsets in parallel streams

$$Dur_{\max} = (\text{SoundSources} - 1) / \Delta\text{phase} \approx 31\text{ms} \quad (3)$$

When we synthesised an ongoing stream for a period of several seconds using this granular synthesis engine we heard a regular pulsing. Examination of the output audio (Fig. 7) shows a regular 15ms period of silence every 100 ms which produces the 10Hz modulation we could hear. After some investigation we found that the duration of this silent period is proportional to the number of SoundSources that are active – with about 0.5ms per SoundSource.

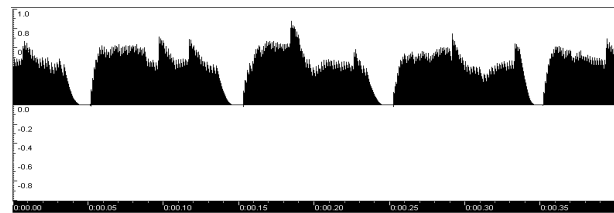


Figure 7: Stream interruptions.

We have not yet identified the source of this interruption but to work around it we reduced the number of SoundSources to 11 so that the final 10ms in the series overlaps and covers the silent 5ms interval between series (Fig. 8).

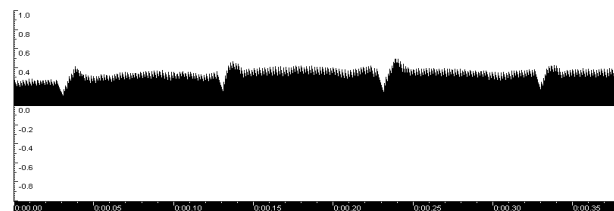


Figure 8: Interactive Synchronous Grain Stream.

The interactive ranges for EGS parameters are shown in Table 2. We have added two extra system-dependent parameters called `SoundSourceMax` and `SoundSources` which affect the maximum duration for a grain (Dur_i)

and the maximum number of overlapping grains ($\text{Overlap}(t)$).

Parameter	Description	Limit
SoundSourceMax	Maximum sound sources for this system	32
SoundSources	Active sound sources	11
Dur i	Grain duration	≤ 10 ms
$\Delta\text{Onset } i$	Delay between consecutive grains	Integer multiples of 1ms
Del i	Delay between end i and onset(i+1)	≥ 1 ms
Overlap(t)	Number of active grains at time t	≤ 10
$\Delta\text{Phase } i$	Phase offset between streams	≥ 1 ms
Path s	Spatial placement	3D

Table 2: Characterisation for interactive parameters.

The coincidence stream from the granular synthesis engine is a synchronous grain stream with

- minimum $\Delta\text{Offset } i = 1$ ms,
- maximum Dur i of 10ms per grain,
- maximum $\text{Overlap}(t)$ of 10 grains
- fill factor = 10000.

3 HAPTO-AUDIO LATENCY

The Windows PC system has the processing power for EGS at interactive rates. But this is only half the story. Integrating the sound with haptics places additional demands on the system for perceptual synchronisation. To measure the latency between haptics and audio renderings we attached a Phantom Desktop haptics device [10] to the PC (Fig. 9).



Figure 9: Phantom Desktop haptics device.

The haptic forces are rendered at 1kHz to maintain the perception of smooth haptic contact with a surface. We hypothesised that the low overhead to activate grains could make it possible to render them from the haptics thread, which would allow tight coupling of haptics and audio without the need for a real-time interrupt driven

system. Down-ramp sound-grains were loaded into the soundcard and triggered from the haptics loop. The resulting high-pitched buzz was rather unpleasant but felt very synchronous with the touch of the stylus onto the surface and did not cause any noticeable jerkiness or deterioration in the haptic smoothness of the surface.

We measured the latency between the outputs from the haptic and audio subsystems by connecting an oscilloscope with one channel on the electrical contacts of a stepper motor on the Phantom and the other on the audio output from the soundcard. When the haptic and audio are triggered by touching a surface with the stylus, the signal on the stepper motor leads the audio down-ramp by less than 1ms.

The capability to play the sound without interrupting the haptic rendering, combined with millisecond haptic-audio latency that is below the 2ms threshold for perception, provides a solid footing for granular synthesis of contact sounds for haptic workbench applications running on a Windows PC.

4 REACHIN API

Applications for the haptic workbench can be built using the Reachin multi-modal scene-graph API. The scene-graph maintains synchronisation of haptic and visual renderings using a high speed 1kHz haptics thread and a 30Hz visual thread. Scene-graph nodes can be programmed in C++ from an extensible library of shape nodes, surface property nodes, simulation and scripting nodes, and control nodes for different haptic and tracking devices. The available surface nodes include :

- SimpleSurface that has stiffness and damping on forces normal to the surface. This simulates surfaces of different elasticity or sponginess that can vary from soft to hard.
- FrictionalSurface is a SimpleSurface that also has static and dynamic friction applied to tangential-forces. This simulates surfaces that can vary from slippery to sticky.
- RoughSurface is a FrictionalSurface with random variation of the friction according to a (gaussian) probability distribution. This simulates surfaces that vary in roughness on a fine scale.
- BumpmapSurface - a FrictionalSurface in which a greyscale bumpmap image is applied to the normal force. This surface can simulate larger scale grainy textures, gratings, bumps, ridges and troughs.

The stylus contact with a surface has two components – the proxy and the finger. The proxy is a small sphere

constrained to remain outside all surfaces in the scene, while the finger is the actual position of the haptic device. The finger is less constrained and may move below a surface. The Surface nodes have an *evaluate()* member function called from the haptics thread to calculate the force to apply to the haptics device and the movement along the surface for the proxy. This function takes input parameters for the 3D position of the finger, the position on the surface of the proxy, and the change in position of the proxy. The force vector has a w component normal to the surface at the point of contact, and orthogonal u and v components tangential to the surface.

As well as the surface nodes the Reachin API has a SoundSource node that provides access to the OpenAL API. The SoundSource encapsulates a sound sample with fields for 3D position, pitch, gain, velocity, looping, and triggering. This node is used to trigger samples from events in either the haptic or graphic threads. For example, a door bell sound can be attached to a button and triggered whenever the button is pushed. Although the SoundSource node makes it easy to add sampled sounds into haptic applications it is not often taken advantage of because of the need to find suitable sampled sounds and because these types of sounds can typically only provide very simple types of Boolean or categorical information. We have augmented the various types of Surfaces in such a way that they automatically synthesise continuous contact sounds from haptic interactions. The augmented surfaces are denoted by the prefix ‘EGS’ to make, for example, EGSBumpmapSurface. These EGSSurfaces each include a version of the granular synthesis engine developed in previous sections. Each engine’s parallel grain streams have been implemented as an array of SoundSource nodes. The EGSSurface plays by cycling through its SoundSource nodes and sending successive ‘play’ instructions to each one in turn to build a synchronous grain stream at the 1kHz haptic frame-rate.

Each of the EGSSurface SoundSource nodes can be initialised with a grain file from a grain pool. We have built a small pool of grains by sampling a metal nail-punch scraping across a smooth ceramic plate, a textured paper pad, a matte plastic surface, a grainy wooden tabletop, and a rough metal lamp stand. A short 10ms portion was cut from the consistent middle part of the sample to capture the micro-level spectral characteristics and reduce meso-level effects of the scraping gesture. The spectral effects due to the shape of the grain were reduced by enveloping it within a 10ms hamming window (Fig. 10).

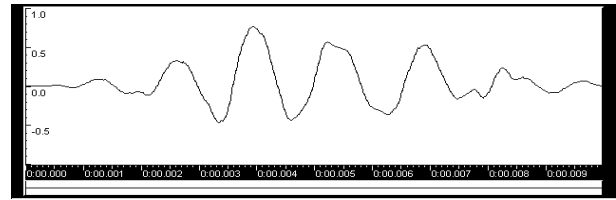


Figure 10: Plastic scrape with hamming envelope.

The grain stream similar to Fig. 8 but with the plastic scrape grain instead of the down-ramp is shown in Fig. 11. In this case the pulsing caused by the gap between grains is much less noticeable due to the hamming envelope.

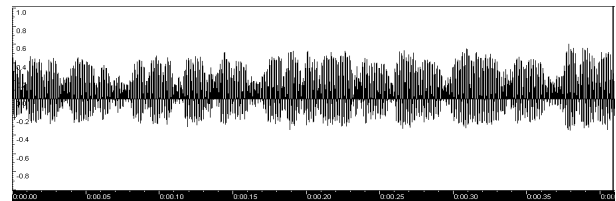


Figure 11. Grain stream from plastic scrape grains.

5 MAPPING HAPTIC CONTACT SOUNDS

There are many possible mappings from the haptic variables in the surface contact function to the parameters of the contact sound grain stream. In our initial round of investigations we have been testing and observing the interactive responsiveness of the sounds to the haptic contact. Previous work with modal models mapped force to the initial amplitude of partials so we followed this lead by mapping the magnitude of the normal force to the energy of a grain:

$$\begin{aligned} \text{energy}(i) &= \text{excite}(i) - \text{damp}(i) \\ \text{excite}(i) &\propto f_w^2(i) \end{aligned} \quad (4)$$

We applied this mapping to a SimpleSurface and found the sound was very responsive to haptic contacts. We used the squared magnitude because this parabolic function seems to allow better perceptions of difference from the sounds than the magnitude function.

Next we applied the mapping to a FrictionalSurface that has tangential resistance. We altered the mapping by substituting tangential force for the normal force so that sounds would be generated by movements across the surface.

$$\text{energy}(i) \propto (f_u^2(i) + f_v^2(i)) \quad (5)$$

In this case the sounds again respond immediately to the contact, and increase in loudness with increases in friction of a surface due to the larger tangential resistive forces. When we applied the mapping to a RoughSurface we found that it was possible to hear the

variations in roughness as variations in the grain stream. The mapping also seems effective for representing the interaction with grosser level features in a BumpmapSurface where we can easily hear the edges of troughs and bumps.

We noticed that the contact sounds do have a noticeable effect on the haptic rendering of the three more complex surface types. We are not yet sure if this is due to the additional CPU load, but we expect to determine the cause in further investigations.

Our results so far indicate that the granular synthesis is very responsive to haptic contacts and can convey auditory information about haptic interaction such as the rate or timing of gestures as well as properties of surfaces such as roughness, bumpiness and edges. The EGSSurface template can be quickly and easily retrofitted to surfaces in existing haptic applications and used to augment future applications, without the need for sound design, sampling or programming of extra events in the interface.

6 FUTURE WORK

So far we have only tried a very simple mapping from haptic force to the excitement of a grain, but this really ‘only scratches the surface’.

In future we would like to investigate mappings of other properties of haptic surfaces to other grain parameters, for example we could select a wood or metal sample from the grain pool based on a characterization of surface properties such as stiffness and damping. The geometric properties of an object, such as shape and hollowness, could also be rendered using hardware accelerated resonance, reverb, and filters on the soundcard. In order to obtain more natural behaviour for interaction with the sounds we would also like to explore physically-based mappings – for example a simple model of sandpaper where surface friction is related to the density of sound grains. Also our experience with haptic applications suggests that there are many other kinds of information that could be sonified with the granular synthesis algorithm.

There is a need to evaluate these mappings with users in order to iterate towards perceptually meaningful representations for different types of information.

7 SUMMARY

Contact sounds are generated by the interactions with objects such as scraping. People can hear information about the interaction gesture as well as the surface in the contact sound. Haptic interfaces are designed for manual interaction with 3D objects but typically these interfaces are silent (just like the movies used to be!).

One reason is the amount of effort and expertise required to design and program an auditory interface. Another reason is that conventional sample-based sounds do not convey the continuous information that is required. We have implemented an ecological granular synthesis algorithm to provide sounds that convey information about interactions with surfaces of objects in the interface, such as the rate and timing of scraping actions and the roughness or bumpiness of the surface. The granular synthesis algorithm can produce a wide range of sounds from a small pool of grains. We implemented the algorithm on a Windows PC using the OpenAL API and hardware acceleration on the wave-table soundcard. We integrated the algorithm into the Reachin API and rendered grains from the haptics thread with latency that is below the threshold of perception. We tested the sounds on four types of haptic surfaces – EGSSimpleSurface, EGSFrictionalSurface, EGSRoughSurface and EGSBumpmapSurface and observed that the sounds are highly responsive to interactions with the surface and vary depending on the properties of the surface. In future we intend to explore and evaluate a much wider range of mapping schemes.

REFERENCES

- [1] S. Barrass and B. Zehner, “Responsive Sonification of Well-logs,” in *Proc. 2000 of Int. Conf. On Auditory Display*, (Atlanta, USA, 2000, April 2-5).
- [2] D. DiFilippo and D. K. Pai, “The AHI: An Audio and Haptic Interface for Contact Interactions,” in *Proc. 2000 Annual ACM Symposium on User Interface Software and Technology*, (San Diego, USA, 2000, November 5-8).
- [3] W. Gaver, “The SonicFinder: an Interface that uses Auditory Icons,” *Human Computer Interaction*, no. 4, pp 67-94 (1989).
- [4] W. Gaver, “Synthesizing Auditory Icons,” in *Proc. 1993 of Int. Conf. on Computer Human Interaction*, (Amsterdam, Netherlands, 1993, April 24-29).
- [5] C. Gunn, “Virtual Artisan's Workbench,” in *Proc. 2000 of the Advanced Simulation Technology and Training Conference* (Sydney, NSW, 2000, Feb 28-March 2).
- [6] D. Keller and B. Truax, “Ecologically-based Granular Synthesis,” in *Proc. 1998 of Int. Computer Music Conference*, (Ann Arbor, USA, 1998, October 1-6).
- [7] D. Keller and J. Berger, “Everyday Sounds:

- Synthesis Parameters and Perceptual Correlates,” in *Proc. 2001 of the Brazilian Symposium of Computer Music*, (Fortaleza, Brazil, 2001, July 31-August 3).
- [8] J. McLaughlin and B. Orenstein, “Haptic Rendering of 3D Seismic Data,” in *Proc. 1997 of the PHANTOM Users Group*, (Dedham, USA, 1997, October 19-22)
- [9] OpenAL API
www.openal.org/about/
- [10] Phantom Haptics device
www.sensable.com/haptics/haptics.html
- [11] Reachin API
www.reachin.se/products/reachinapi/
- [12] D. Ruspini, and O. Khatib, “Acoustic Cues for Haptic Rendering Systems,” in *Proc. 1998 of the PHANTOM Users Group*, (Dedham, USA, 1998, October 3-6)
- [13] D. Stevenson, K. Smith, P. Veldkamp, J. McLaughlin, R. O'Hagan, D. Smith, and C. Gunn, “Virtual environments for Industrial Applications,” in *Proc. of IFIP Conf on Human-Computer Interaction*, (Sydney, Australia, 1997, Jul 14-18).