Integrated Modal and Granular Synthesis of Haptic Tapping and Scratching Sounds

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

Abstract

People can hear a lot of information about an object by tapping and scratching it. We have developed a TapAndScratch node that automatically synthesises the sounds of tapping and scratching interactions with haptic objects in the Reachin haptic API.

The scratching sounds are produced by a granular synthesis algorithm that is used to provide information about interactions with the surface of an object, such as variation in force, rate of scratching and duration of scratch strokes. The scratching sounds can also provide information about properties of the surface such as stiffness, friction, and bumpiness. The tapping sounds are produced by a modal synthesis algorithm that provides more global information about the material and shape of the object when it is tapped. We believe these continuously varying sounds can provide useful information and further engage the user.

1. Introduction

When you tap an object you can hear how hard it is, whether it is hollow and what it is made of. If you scratch or scrape it you can hear how rough it is, how bumpy, whether it has a regular texture. However most haptic objects don't make a sound when you handle them. It's as though you are in outer space.

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 [7,12,17]. These applications are built on the Reachin application programming interface (API) [16] which has a hapto-visual scene-graph and supports sound through the OpenAL sound API [13]. 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 sampled sounds triggered by events only give very basic categorical information that does not provide sufficient advantages over the non-auditory interface to warrant substantial effort.

We have addressed these problems by building a TapAndScratch node that automatically synthesizes sounds from user interactions with haptic objects in the Reachin haptic API. This node makes it quick and easy to add sounds to existing or future haptic applications, without the need for specialist sound design or user interface design expertise.

Synthesized sounds can provide information about the force and velocity of interaction gestures, surface properties of the object such as stiffness [3], roughness and texture [11], and global properties of the object such as shape [10,20] and material [6,18]. Consequently, these synthesized sounds can be a source of additional information that engages the user and increases the tangibility and sense of presence in the haptic experience.

In the following sections we overview previous work on haptic contact sounds, describe our work on synthesizing tapping and scratching sounds, make observations on the resulting system, and finally give directions for future work.

2. Previous Work

Actions and events in the Apple desktop GUI were used by Gaver to synthesise metaphorical sounds that he called Auditory Icons [5,6]. When a file was dragged the sound mimicked the sound of a real paper file being dragged across a real desktop. Gaver produced his Auditory Icons using modal synthesis, a source/filter algorithm and FM synthesis. Modal synthesis was used for impacts, bouncing and breaking sounds. The source/filter algorithm was used for more general interactions with objects, such as scratching and dragging. The FM algorithm was used for the sounds of ongoing processes represented by cyclic machine-like sounds.

The modal synthesis algorithm used by Gaver was adapted by van den Doel and Pai in a program called Sonic Explorer, which enabled the user to 'hit' virtual objects such as tables, chairs etc. by tapping them with a mouse in a 2D window [20]. The following parameters were considered when synthesising the sound:

- The shape of the object - represented by the frequency response.
- The location of the impact - represented by the relative amplitudes of the frequency components.
- The material of the object - represented by the decay of frequency components due to internal friction.
- The force of the impact - the amplitude of the emitted sound was proportional to the square root of the energy of the impact.

The modal algorithm was developed further by Pai, van den Doel et al. [14] when they modelled the sounds of a virtual clay pot from the impulse response of a real object. The model was measured by applying a finite approximation of an impulse force at a large number of impact points and recording the sound at each point. The modal frequencies were estimated from the average power spectrum of the recordings (corrected for background noise) using peak identification and then refined by performing a phase reconstruction.

The modal algorithm was then adapted to a haptic interface by DiFillipino and Pai [4] who used the modal algorithm to produce haptic contact sounds in their custom real-time system using a 3DOF Pantograph device connected to a DSP audio subsystem. Since the modal algorithm only models an impact they developed an 'acoustic force profile' to represent more complex interactions such as scratching. The acoustic force profile was calculated from the haptic forces and convolved with the modal impulse response to produce the final sound. Spurious impacts were removed by threshold pruning, and jitter was low pass filtered to produce a smooth model of the interaction. This system, with guaranteed latency of 0.5ms, was used to establish a 2ms lower bound on the perception of latency between the haptic and audio response to an impact event.

The modal synthesis algorithm was extended to more complex interactions of scratching and rolling by van den Doel, Kry and Pai in their FoleyAutomatic system [18]. The scratching sounds were generated by substituting the acoustic force profile with a looped wavetable filled with samples of fractal noise. The velocity of the scratching motion was mapped to the rate of looping of the wavetable so that faster scrapes sounded faster. Rolling sounds were simulated by low-pass filtering the scrape sound based on a pseudo physical analysis of rolling motions on surfaces, however they found these sounds to be less convincing than the scrapes and impacts. This extended modal synthesis algorithm was implemented for a Phantom haptic device [15] to allow the user to tap and scratch a haptic bell object.

The need to provide information about more complex scratching and scraping interactions led Barrass and Adcock to adapt an ecologically-based granular synthesis algorithm (EGS) as an alternative to modal synthesis [1]. This algorithm explicitly separates the interaction model from the object model so that the interaction forces can be heard in detail [8,9]. It has the added advantage that it can also represent surface and material properties of the object that is being interacted with. The technique requires millisecond temporal control over streams of short 3-30ms 'grains' of sound that can be sampled from real objects or synthesized. They used the 1kHz haptic rendering thread to achieve fine-grained temporal control over the grain stream with haptic-audio latency below the 2ms perceptual threshold on an off-the-shelf Windows PC and soundcard.

3. Surface Information Using Granular Synthesis

The EGS algorithm can provide auditory information about more complex user interactions with haptic objects, such as scratching and dragging. In order to make it quick and easy to apply to haptic applications we implemented the EGS as an abstract EGSSurface node within the Reachin Haptic API. This node uses the OpenAL 3D sound API via Reachin SoundSouce nodes.

Besides hearing information about the user interaction we would also like to hear information about the object that is being handled. We explored some simple mappings of the EGS algorithm to Reachin Surface nodes with different properties to explore how it might represent information such as stiffness, roughness and bumpiness:

- EGSSimpleSurface 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. We map the square of the magnitude of the normal force to grain amplitude. Pushing against stiffer surfaces requires more force and produces a louder sound. This mapping is not very intuitive, as most real objects do not make any sound if you just push against them.

- EGSFrictionalSurface is an EGSSimpleSurface that also has static and dynamic friction applied to tangential-forces. This simulates surfaces that can vary from slippery to sticky. We map the squared magnitude of tangential force to the amplitude of the grains. In this case the sounds respond to sliding contacts very immediately, and increase in loudness with increases in friction of a surface due to the larger tangential resistive forces.

- EGSRoughSurface is an EGS FrictionalSurface with random variation of the friction according to a (gaussian) probability distribution. This simulates surfaces that vary in roughness at a fine scale. We found that it was possible to hear the variations in roughness as correlated variations in the loudness of the grain stream.

- EGSBumpmapSurface is an EGSFrictionalSurface in which a greyscale bumpmap image modifies the normal force. This surface can simulate larger scale grainy textures, gratings, bumps, ridges and troughs. Again we found that even the simple mapping to grain amplitude produces information about the bumpiness of the surface as the stylus forces vary on edges and smooth sections.

In informal experiments, we found that the EGS algorithm can provide fine grained information about the force, rate and timing of scratching gestures on a haptic surface as well as properties of the surface such as stiffness, friction, roughness and bumpiness.

4. Shape Information Using Modal Synthesis

The modal synthesis algorithm, commonly used to synthesise haptic contact sounds, provides information about the force of tapping interactions as well as properties of the object such as size, shape and material [6, 20] that are not provided by the EGS algorithm. We implemented the modal synthesis algorithm as a ModalSynthObject node in the Reachin API. The algorithm was adapted from van den Doel and Pai's JASS implementation [19] and coupled to the STK Synthesis ToolKit [2] to handle the low level audio output. The code was multithreaded using locks to prevent buffers being overwritten. This algorithm runs at interactive rates on a Windows PC.

The ModalSynth node can represent rectangular or circular membranes made of different materials, or arbitrary objects specified by a set of frequencies, gains and dampings. The parameters of the node are as follows:

- Frequencies, gains and dampings – The modal frequencies of a vibrating object that can be used to specify a complex object.
- LengthX, LengthY – The length and width, in the case of a rectangular membrane.
- Radius – The radius, in the case of a circular membrane.
- DampingFactor – A constant used to characterize material, representing the internal friction of the membrane.
- SpeedOfSound – The speed of sound in the membrane determined by the membrane tension and density.

In informal experiments we found that tapping on the surface of a ModalSynthObject in different places produces different sounds that vary depending on the point of contact, the shape of the object, the material of the object and the force of the impact.

5. Integrated TapAndScratch Node

In order to allow the user to both tap and scratch haptic objects we integrated the EGSSurface and ModalSynthObject nodes to create a TapAndScratch node as shown in Figure 1.

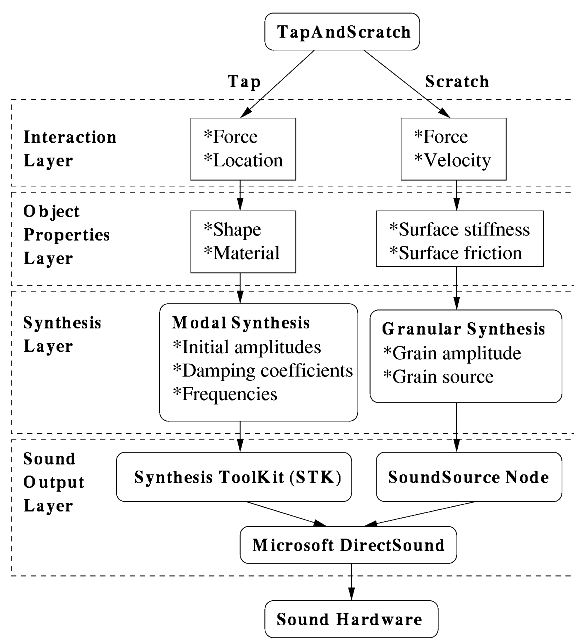

Figure 1 – The TapAndScratch Architecture.

The TapAndScratch node builds up force buffers over a number of haptic loops and then sends these values as a 'force impulse' to excite the sinusoids of the relevant pre-computed circular or rectangular membrane model. The resulting sound is fed to the soundcard via STK. The TapAndScratch node also calls the Granular Synthesis method each haptic cycle during haptic contact. The granular synthesis method scales the gain of a sound grain from the grain pool by the force and speed of the scratching motion and plays it with a Reachin SoundSource player.

6. Observations

Although the modal synthesis and granular synthesis are now working together, they are essentially two separate synthesis techniques outputting sound at the same time. This works fairly well since hitting and scratching actions with a stylus do not tend to overlap. However, the transition from hitting to scratching is not smooth due to the change in timbre from one algorithm to the other. Also, the characterisation of resonance included in the rendering of tapping has not yet been included in the scratching sounds.

During development, some small virtual scenes were constructed to demonstrate various aspects of the TapAndScratch node. We built a virtual marimba from rectangular wooden bars of different lengths which had notes that varied with the length of the bars, the force of the impact and the location of the impact. Another demonstration program was constructed from different sized circular plates to resemble a virtual drum kit.

Although the tapping sounds are quite convincing, there was a noticeable lag between the haptic force and the sound output. The modal synthesis algorithm is tightly coupled to interaction through the haptic loop but is too CPU intensive to synthesise the initial impact transient with a lag less than the 2ms perceptual threshold.

The scratching sounds are very responsive to user interaction since they are initiated from inside the haptic loop and synthesized in hardware wavetables on the soundcard. However when we started experimenting with more complex mappings the haptic feel of the surface became noticeably rougher. This effect is due to extra CPU load in the mapping slowing down the haptic frame rate. We tried moving the granular synthesis out of the haptics loop to a separate thread, as per the modal synthesis, but this does not seem to have solved the problem.

We observed that the tapping and scratching sounds also provided extra information about what is happening in a Haptic Workbench application to other people in the vicinity. This leads us to consider the possibility that these sounds could help support collaboration in collocated activities.

7. Future Work

The perceptible lag in the tapping sounds may be overcome by a smarter algorithm – for example we could use the granular synthesis for the impact transient then mix in the modal synthesis for the longer ringing part. Other approaches could include developing a more efficient or optimised algorithm that could take advantage of hardware on the soundcard.

So far we have only tried very naive mappings of haptic force to the amplitude of a grain in the granular synthesis algorithm. This mapping provides information about interaction gestures but not so much information about the properties of the surface. In future we would like to investigate mappings of surface properties to other granular synthesis parameters - for example by selecting different grain spectra based on the surface material or varying the density and overlap of grains with stiffness and damping. The development of effective and meaningful mappings will require user testing on the perception of the surface properties from the sounds. We also need to address the problem that more complex mappings slow down the haptic frame rate.

At present the EGS and modal algorithms are architecturally integrated, but independently synthesize separate audio streams in response to the same interaction. In future we would like to research and develop new hybrid algorithms that integrate the surface and shape properties in a single coherent sound that morphs from impacts to scrapes, rather than mixes.

8. Summary

We have developed a new TapAndScratch node that automatically synthesises the sounds of user interactions with haptic objects in the Reachin Haptic API. This node can be used to quickly and easily add haptic sounds to existing or future applications without the need to sample and program individual sounds for every event and object as is required in conventional sample based auditory interfaces.

The TapAndScratch node integrates two different sound synthesis algorithms for scratching and tapping interactions with haptic objects. The scratching sounds are produced using a granular synthesis algorithm that provides detailed information about user interactions with the surface of the object, such as force, rate of scratching and duration of the interaction. The scratching sounds also provide information about the surface properties such as stiffness, friction, and bumpiness. The impact sounds are produced by a modal synthesis algorithm that provides global information about the material and shape of the object when it is tapped. We believe these continuously varying sounds can provide useful information, engage the user, increase the tangibility of the haptic experience, and support collaboration.

9. References

[1] S. Barrass and M. Adcock, "Interactive Granular Synthesis of Haptic Contact Sounds", in *Proc. 2002 AES 22nd International Conference on Virtual, Synthetic and Entertainment Audio*, Espoo, Finland, 2002, July 15-17.

[2] P.R. Cook and G. Scavone, "Synthesis ToolKit (STK), Version 2.1", in *Proceedings of the International Computer Music Conference*, Beijing, China, 1999.

[3] D.E. DiFranco, G.L.Beauregard, and M.A. Srinivasan, "The Effect of Auditory Cues on the Haptic Perception of Stiffness in Virtual Environments", in *Proc. 1997 of the ASME Dynamic Systems and Control Division*, Ed. G. Rizzoni, ASME, 1997, pp 17-22.

[4] 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.

[5] W. Gaver, "The SonicFinder: an Interface that uses Auditory Icons," *Human Computer Interaction*, no. 4, 1989, pp 67-94.

[6] W. Gaver, "Synthesizing Auditory Icons," in *Proc. 1993 of Int. Conf. on Computer Human Interaction*, Amsterdam, Netherlands, 1993, April 24-29.

[7] C. Gunn, "Virtual Artisan's Workbench," in *Proc. 2000 of the Advanced Simulation Technology and Training Conference* Sydney, NSW, 2000, February 28-March 2.

[8] D. Keller and B. Truax, "Ecologically-based Granular Synthesis," in *Proc. 1998 of Int. Computer Music Conference*, Ann Arbor, USA, 1998, October 1-6.

[9] 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.

[10] A. Kunkler-Peck and M.T. Turvey, "Hearing Shape", in *Journal of Experimental Psychology: Human Perception and Performance*, Vol. 26, February 2000, pp 279-294.

[11] M.R. McGee, P.D. Gray and S.A. Brewster, "Feeling Rough: Multimodal Perception of Virtual Roughness", in *Proc. 2001 of Eurohaptics*, Birmingham, UK, 2001.

[12] 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.

[13] OpenAL API www.openal.org/about/

[14] D.K. Pai, K. van den Doel, D. L. James, J. Lang, J. E. Lloyd, J. L. Richmond and S. H. Yau. "Scanning Physical Interaction Behavior of 3D Objects", in *Computer Graphics (ACM SIGGRAPH 01 Conference Proceedings)*, Los Angeles, USA, 2001, August 12-17.

[15] Phantom Haptics Device www.sensable.com/haptics/

[16] Reachin API www.reachin.se/products/reachinapi/

[17] D. Stevenson, D. 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.

[18] K. van den Doel, P.G. Kry and D.K. Pai, "FoleyAutomatic: Physically-based Sound Effects for Interactive Simulation and Animation", in *Computer Graphics (ACM SIGGRAPH 01 Conference Proceedings)*, Los Angeles, USA, 2001, August 12-17.

[19] K. van den Doel and D.K. Pai, "JASS: A Java Audio Synthesis System For Programmers", in *Proceedings of the 2001 International Conference on Auditory Display*, Espoo, Finland, 2001, July 29-August 1.

[20] K. van den Doel and D.K. Pai, "The Sounds of Physical Shapes", in *Presence*, Vol. 7, No. 4, August 1998, pp 382-395.