Audio feedback for gesture recognition*

John Williamson[1] Roderick Murray-Smith[1,2]

[1] Department of Computing Science,
University of Glasgow,
Glasgow G12 8QQ
Scotland, UK.

[2] Hamilton Institute,
National Univ. of Ireland, Maynooth,
Co. Kildare,
Ireland

E-mail: jhw,rod@dcs.gla.ac.uk

Abstract

A general framework for producing formative audio feedback for gesture recognition is presented, including the dynamic and semantic aspects of gestures. The beliefs states are probability density functions conditioned on the trajectories of the observed variables. We describe example implementations of gesture recognition based on Hidden Markov Models and a dynamic programming recognition algorithm. Granular synthesis is used to present the audio display of the changing probabilities and observed states.

1 Introduction

Gesture-based input is becoming increasingly important because of the increasing need for devices capable of supporting complex, continuous interaction in a range of contexts. However, gesture recognition technologies are not perfect; particularly in the circumstances where they would be most useful, such as with portable devices, on the move. Disturbances to the inputs from walking or moving and inaccuracy caused by noisy sensors can cause even the best of extant recognition systems to fail. Variation in performance is also normal, as each gesture from the same person will vary, different users will have individual gesturing styles, and furthermore these are all subject to change in different contexts, such as changing activities or emotional states.

Some of the accuracy limitations of gesture recognition systems can be overcome by providing feedback to the user, making the recognition process less opaque. Many conventional systems can provide sound or vibrational summative feedback on the completion of a gesture, allowing the user to identify any potential errors. However, dynamic, continuous feedback during the gesture allows the user to respond to the system in real-time. This avoids repetition, and can help the user understand where errors or deviations are occuring in the gesture performance. It also allows users to gain insight into how the system interprets their movements.

*Technical Report TR-2002-127, Department of Computing Science, University of Glasgow, Glasgow G12 8QQ, Scotland, UK. 20th December, 2002.

In this paper we will describe methods for providing formative audio feedback. Many of the features will also be applicable to vibration or other forms of feedback.

In the context of this work, gestures will be considered to be trajectories of a vector $x(t)$ in some state-space of dimension n; this might include measurements of orientation, position or other sensed data. The space may also be augmented with variables inferred from measurements; for example estimating acceleration from position information. The space may also include more complex beliefs inferred from measurements, e.g the probabilities of a particular sequence being related to a particular gesture. By considering the entire system as a trajectory in some state space, a general framework can be applied to provide feedback on gesture performance. A mapping can be defined between the state space and an output mechanism; this can then be used to feedback information ranging from the simplest direct measurements to the semantic content of the gestures within a single framework. An example state vector could be

$$x(t) = \left[\dot{\theta}\ \dot{\psi}\ \dot{\phi}\ x\ y\ z\ v_x\ v_y\ v_z\ a_x\ a_y\ a_z\ p_1\ p_2\right], \tag{1}$$

where $\dot{\theta}, \dot{\psi}$, and $\dot{\phi}$ are changes in pitch, yaw and roll angles, other terms are positions and linear accelerations at time t, where p_1 and p_2 could be probabilities of the events 'delete' or 'accept' conditioned on all information up to time t, provided by some pattern recognition algorithm.

2 Review

A number of previous systems have included some audio feedback facilities for the gesture recogniton process. Muller-Tomfelde and Steiner (2001) describe a collaborative whiteboard system which uses gesture control. A formative audio feedback mechanism for the recognition process is briefly mentioned, in which the feedback consists of a melody which always begins in the same way, and as the gesture progresses the melody changes as the gesture is recognized.

Ghez *et al.* (2000) describes a system providing musical feedback based on joint movements. The system was intended to help rehabilitate patients who had lost proprioception in their limbs. In the described experiment, rhythmic output was produced based upon the motion of the mouse. Melodic lines are presented, whose contours roughly follow the outlines of the motion. When the patient moved in synchrony with the system, the audio was adjusted in include a second timbre. In this way, the patient was able to synchronize her motion with system. With the aid of this feedback, the patient's performance in producing rhythmic motion with her arm was found to be roughly equivalent to that of unimpaired subjects.

A second experiment was performed in which the patient's arm was instrumented with rotary potentiometer to measure joint angles at the elbow and shoulder. The angular velocity was mapped to pitch, and direction reversals (which correspond to zero-crossings in the velocity curves) caused additional audio cues to be played. The timbre of the feedback was adjusted according to how far out of synchronization the actual motion was with the intended motion. The patient was asked to move towards a series of visual targets, while her arm was unexpectedly unbalanced by the experimenter. This feedback was sufficient to allow the patient perform the task satisfactorily using the audio alone.

Direct mapping of physical motion to sound for gesture feedback is described in McQueen and Mantei (1994), where motion in two dimensions was directly mapped to various synthesizer parameters. Volume, pitch and timbre were used, with timbre and pitch as the x and y axes being the most successful. Participants were asked to perform "strokes" — very simple

gestures consisting of lines of different orientations. They were then to use the feedback to help perform specific strokes. Evaluation showed that users were able to learn the sound patterns, and that the provision of audio did improve their performance. This technique did not, however take into account the dynamics of the signal; it merely mapped physical space to an audio space.

3 Audio feedback

Audio is an obvious modality for presenting gesture information, given the contexts in which gesture recogntion is likely to be useful. It can be used to present high-dimensional, dynamic data, and is suitable for use when the eyes may well be occupied – either with a other visual displays, or with other tasks such as walking when using mobile devices.

There are numerous novel electronic instruments (Jorda (2001), Mulder (1994)) which have been developed to transduce movements into sounds. Although these have some relation to formative feedback for gesture recognition, it is important to distinguish musical instruments, which aim to produce some aesthetically pleasing output, from gesture feedback systems, which aim to display useful information to assist the user, related to the dynamics or the semantics of the input. Existing musical culture must be exploited to produce meaningful feedback, so concepts such as dissonance, rhythm, harmony and melodic contours can be utilized to present the information; but the aim is not to produce music.

3.1 Dynamics related feedback

A simple model for enhancing user performance with gesture recognition systems is to simply augment the user's proprioception with audio. This involves sonifiying the aspects of the gesture state space corresponding to physical movements; e.g. acceleration or velocity. This can help the user intuitively understand how movements they are making are perceived by the device and potentially increase stability and repeatability of the movements in the relevant dimensions.

One example is to focus on the zero crossing points in an input variable. Specifically, zero crossings in acceleration curves correspond to changes of direction of the forces applied by the user; sonifying these corresponds to the actual movements the user's body is making. By producing notes at these points, these changes of direction can be sonified. Dividing the curve into segments delimited by zero crossings, then mapping the maximum amplitude of the curve in the previous segment to the amplitude of the note causes large changes in force to be emphasized, and high-frequency noise is reduced.

An enhancement is to play a pair of notes at each crossing, descending if the crossing is positive-to-negative and ascending if negative-to-positive, and mapping the pitch difference between these notes to the previous maximum amplitude of the curve. This produces a melody whose contour reflects the contour of the acceleration. Making each pitch change relative to the last produces a continuous contour. Figure 1 shows this process. Different dimensions of the input (x and y or yaw, pitch and roll) can be distinguished by using different timbres for the notes.

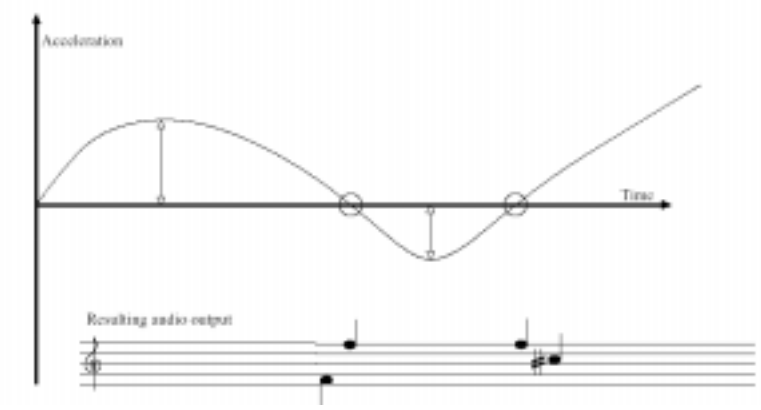


Figure 1: Zero crossings in a curve, and corresponding sounds generated

3.2 Sonification of probabilistic belief revision

Probabilistic methods for recognition allow the dynamic estimation of the system's beliefs about the meaning of the current movement input. Extending the sonification to include elements of the space representing probability densities for gestures, the system's 'understanding' of the recognition process can be communicated to the user in real time. The user can then respond, adjusting their behaviour if they realise that the system is interpreting their movements other than intended.

The feedback model requires that the gesture recognition model should be able to dynamically update the probability associated with each gesture with each new sampled input, in order to pass this time-series to an audio display for real-time presentation of the changing probabilities. Systems capable of dealing with context information can calculate probabilities of gestures given a both model and the current system context; this produces some new output probability which can then be sonified in this framework.

Many existing sonifications of continuous data, such as those presented in (Ghez *et al.* 2000, Brown *et al.* 2002), use sequences of discrete notes to represent data. Gesture trajectories, and their associated probabilities, are continuous in both space and time; thus we suggest that it may be more effective for the audio output to be continuous as well.

3.2.1 Dissonance feedback

A relatively straightforward sonification of the recognition process can be achieved by using a distinct pitch for each gesture model. The amplitude of each of these pitches, which can be pure sinusoidal tones or any other waveform, is then mapped to the current probability of the current trajectories being generated by that particular model.

Frequencies of the tones are selected so that they are sufficiently closely spaced that they are unpleasant when sounding simultaneously. This causes the sound to be dissonant and unpleasant to human ears when ambiguity is high and a number of tones are present. As a gesture is performed, the feedback will initially sound dissonant and confused, but as the gesture progresses, one or two tones will come to dominate until a single tone remains if the gesture is unambiguously identified. At this point the user can stop performing the gesture, as they know that the system has sufficient information to correctly interpret their movement.

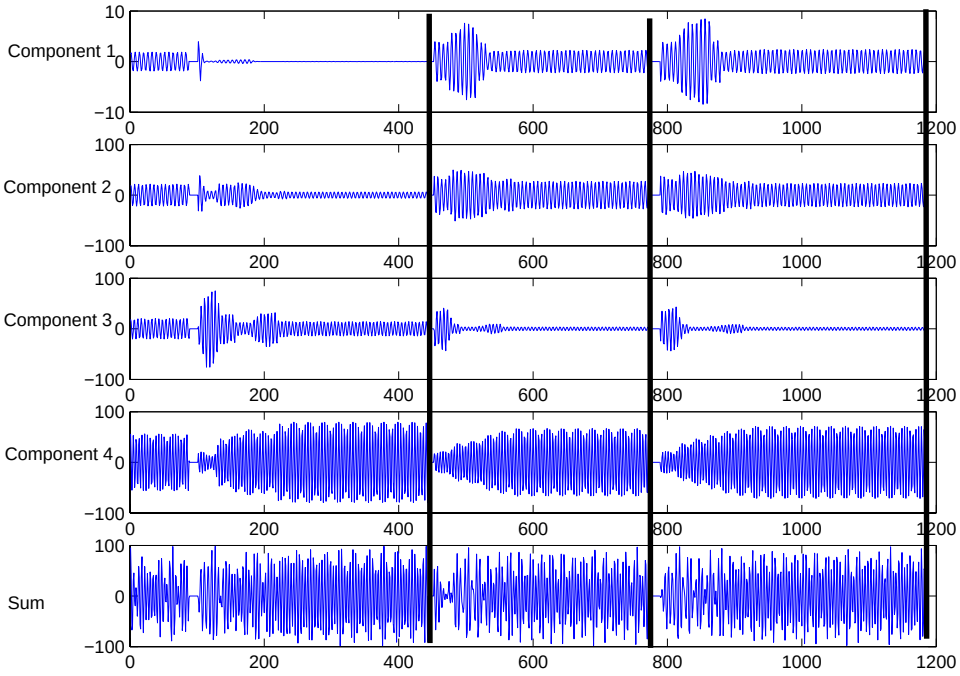


Figure 2: Output waveform for probability series in Figure 3. Each gesture has an associated sine wave of some frequency, the amplitude of which is mapped to the probablity of that gesture

Conversely, if the system is unable to distinguish the gestures during performance, the user will be aware of the potential mis-recognition and can then reproduce the gesture. Figure 2 shows the output waveform for the probability time series shown in Figure 3.

The 'tightness' of the feedback can be adjusted by adding a nonlinear mapping from the output probabilities to the amplitudes. This can be used to compensate for perceptual non-linearities; for instance it may be difficult to respond to a large number of simultaneous pitches. Essentially this corresponds to a reweighting of the probabilities; lower probabilities can be penalized (by some function, such $p(\text{gesture})^k, \quad k > 1$) or a threshold can be applied to remove lower probability elements entirely; or alternatively only the top n probability components can be used.

This can be used to produce feedback which resolves much more quickly to a single tone for "sharper", potentially less confusing feedback, which may feel more responsive. The choice of mapping is then part of the feedback design.

3.3 Granular synthesis

3.3.1 Overview

Granular synthesis (Xenakis (1971), Roads (1978), Truax (1988)) is a probabilistic sound generation method, based on drawing short (10–200ms) packets of sound, called "grains" or "granules", from source waveforms. A large number of such packets are continuously drawn from various sources, shaped so as to avoid discontinuities and summed.

In asynchronous granular synthesis, the grains are drawn according to some probability

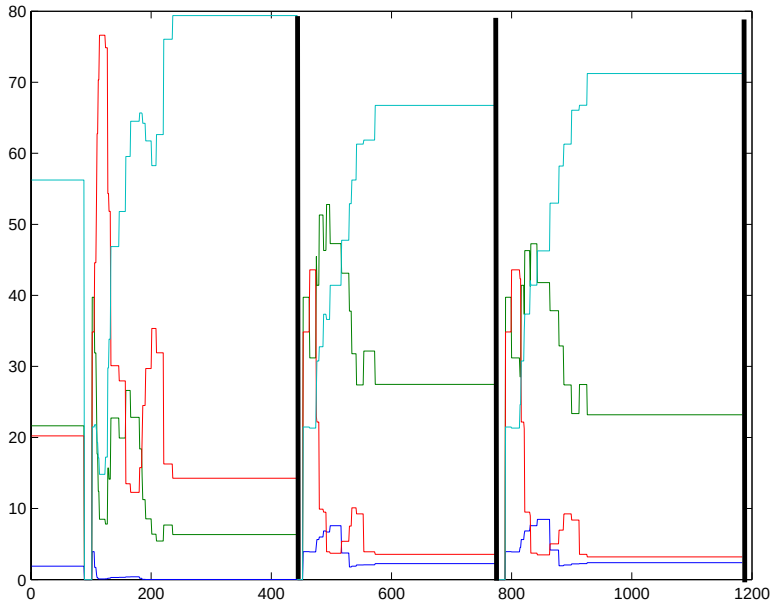


Figure 3: Probability series for a gesture alphabet of 4 symbols, with 3 gesture performances. Black lines indicate recognition of gesture. The stable period before recognition is the quiescent period used to indicate gesture termination.

distribution giving the probability of the next grain being selected from one of the potential sources. Figure 4 shows the basic process. This gives a smooth continuous texture, the properties of which are modified by changing the probabilites associated with each grain source.

Additionally, a distribution can be defined over the time axis of the source waveform, giving the probability of a grain being drawn from any point in the wave. For example, in Figure 5 a narrow Gaussian distribution is shown. By applying this distribution to the sources, adjusting the mean of the distribution causes the sound to smoothly move through the waveform. The mean can be moved at any speed with smooth results, allowing flexible time stretching without pitch distortion. The width of the distribution can also be adjusted to allow varying degrees of temporal blurring.

3.3.2 Application to gesture feedback

Clearly, this sound generation method is an ideal feedback machanism for probabilistic recognition methods. Given some state space, a number of densities can be placed in the space, and the trajectories in that space can then be sonified. More formally, each density provides a mapping from our n-dimensional state-space to one of the m output series. The number m of output series, and the level of interaction among their densities is an arbitrary design feature.

Further interpretations of the significance of different regions of state-space can be added by increasing the number of belief terms, without necessarily affecting the existing densities.

An illustration is given in Figure 6 which shows how a mixture of Gaussians could be used to map regions of a two-dimensional state-space to sound. In this case there are six

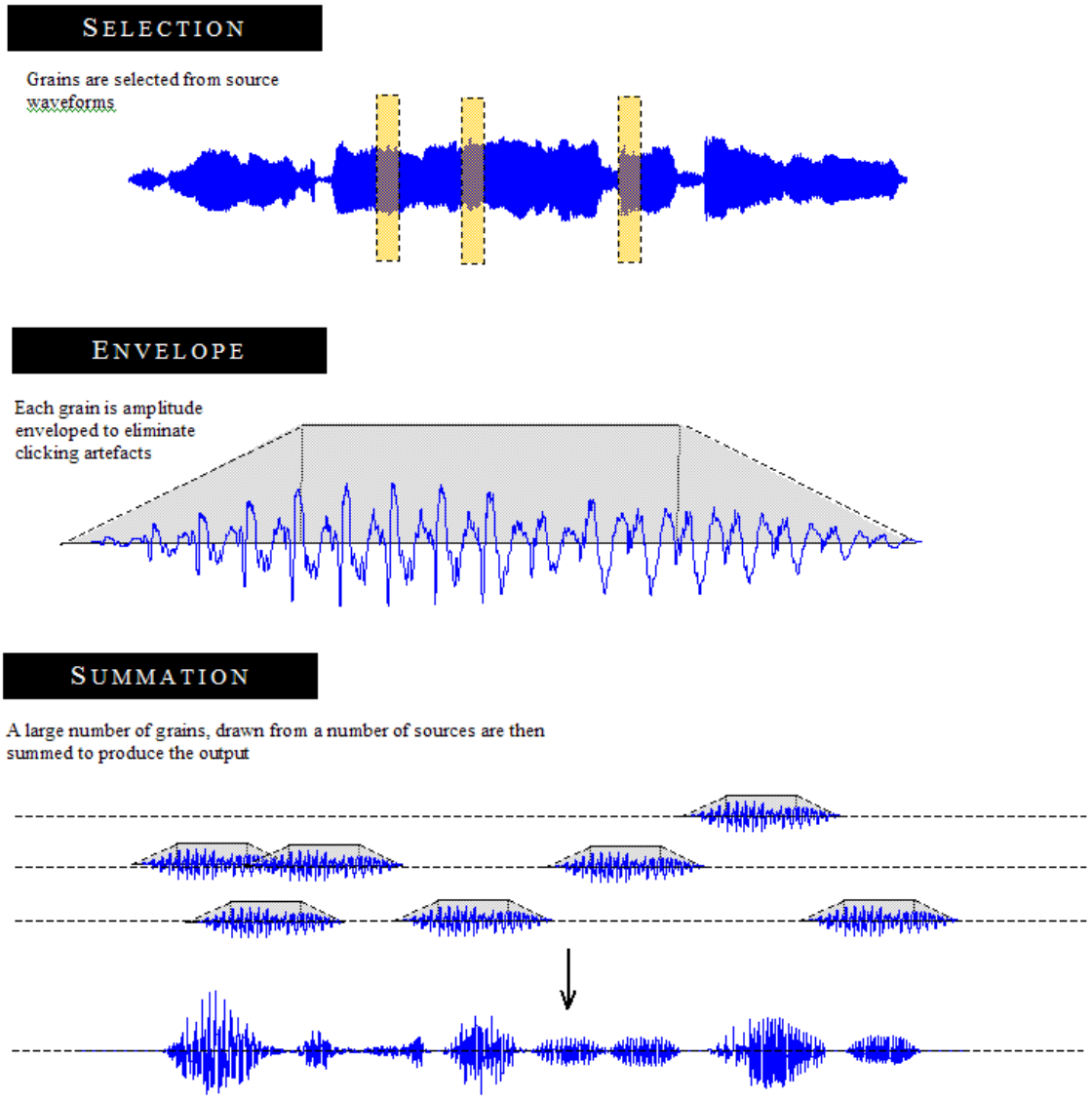


Figure 4: Simple granular synthesis process. A much greater number of grains would be used in real output.

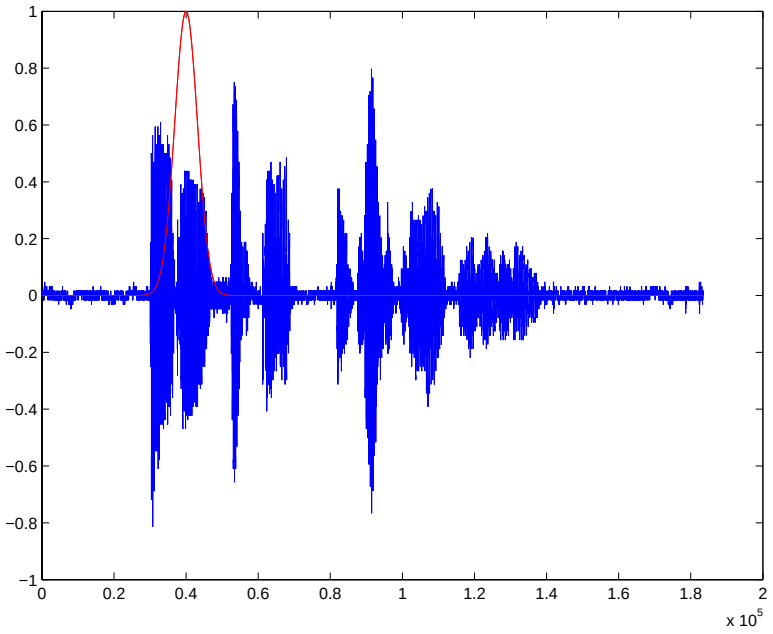


Figure 5: Gaussian probability distribution over time in waveform being translated to produce timestretching

mixtures corresponding to six belief variables. A similar example of a number of realisations of gestures is shown in Figure 7 where a simple Gaussian distribution is placed around a prototype gesture. More realistic estimates of variance could be derived from methods such as *Functional Data Analysis* as described in (Ramsay and Silverman 1997), or mixtures of Gaussian Processes (Shi *et al.* 2002).

In the gesture example, each model can be assigned a source waveform, and the model's output probabilites directly map to the probability of drawing a grain from the appropriate source. The design issue is then reduced to creating a suitable probability model, and selecting appropriate waveforms as sources. In the implemented system, grains are generated such that around 100–1000 are always active; as one grain finishes, a new one is drawn from to the current distribution.

The previously described dissonance model can be exactly reproduced by simply creating each source with the same timbre and different pitches. Of course, much more interesting differences in timbre can also be used; harmonic or inharmonic, bright or dull, metallic or wooden, brass or strings, and so on.

The granular time stretching technique is a very natural extension of this method, which can be used to provide an indication of a user's progression through a gesture. Mapping the arc length of the current gesture to the mean of the distribution across time inside each source produces sounds which smoothly progress as the gesture continues. For example, speech or short segments of music can be used, making it simple to produce feedback which is both relevant and interesting for the user.

The width of the distribution can be mapped to any uncertainty about the position in the gesture; wider distributions giving a more blurred sound. Alternative distributions, such as a mixture of Gaussians, can be used to represent more complex density functions on position, if

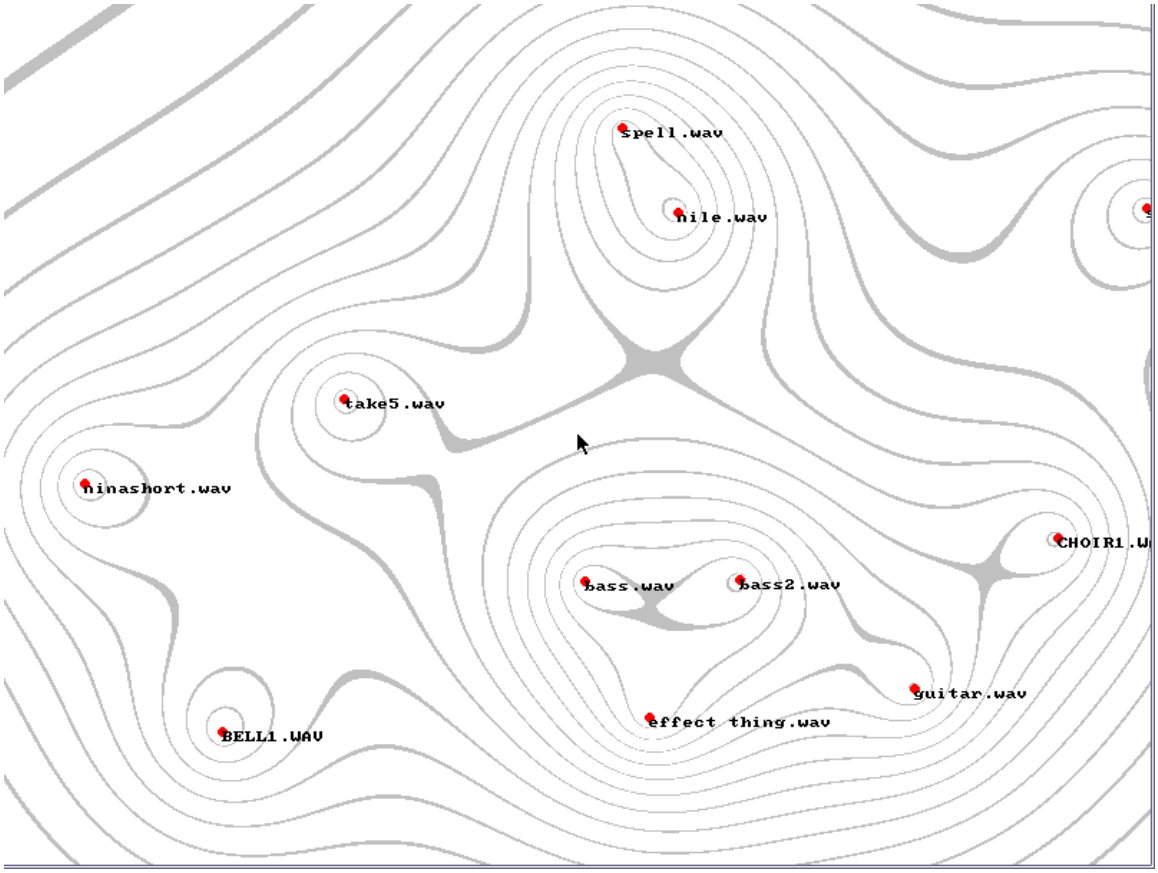


Figure 6: Mixture of Gaussian densities in a two dimensional state space as illustration of the basic concept. Each Gaussian is associated with an audio waveform, and could represent the p.d.f. of some variable.

a simpler distribution is insufficient.

4 Recognition algorithms

There a number of possible recognition models suitable for the feedback methods described. The only requirements are that the recognition system be able to produce the probability of a gesture, conditioned on the data received up to any given time step.

4.1 Hidden Markov Models

HMMs (Rabiner and Juang (1993)) have been extensively used for gesture recognition (Wilson (2000), Yang and Xu (1994), Lee and Xu (1996), Morimoto *et al.* (1996) Wilson and Bobick (1999)). Discrete or continuous HMMs are suitable for such use, though only a discrete space, discrete time HMM has been tested with the feedback framework. In this setup, vector quantization is applied to filtered velocity estimates from the input devices. The HMM's are trained using the Baum-Welch algorithm from a number of prototype gestures performed by the user.

During the recognition stage, each HMM is updated with the input symbols as they are generated, and the Viterbi algorithm can be efficiently applied to calculate the new probability of the current symbol sequence being generated by that model. This then directly maps to the probabilistic feedback model. Once the gesture is completed, either via a quiescent period or a segmentation via a physical button, the models are reset.

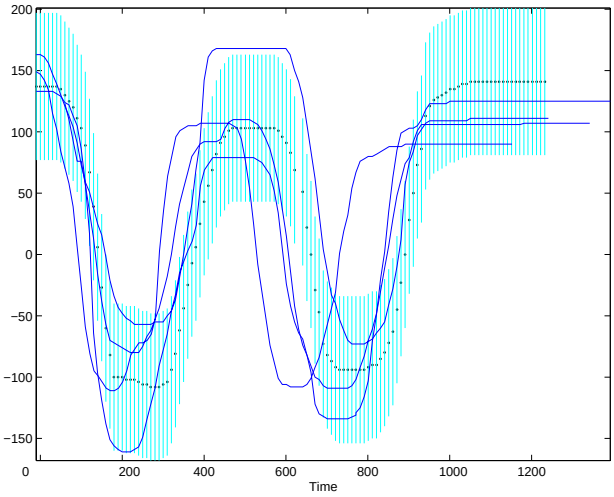
HMMs provide a compact, efficient and well-established gesture recognition model. They naturally fit in with the audio feedback techniques described, and in tests provided good performance. They can, however, be prone to poor training in some cases, when the Baum-Welch algorithm gets stuck in a local minimum. Simulated annealing, in conjuction with Baum-Welch, was implemented, which improved matters somewhat, but is generally very time-consuming except for small training sets. The poor training can lead to model dominance, where one model is consistently more probable than others for a large range of gestures. This can cause the feedback to become skewed or indistinct.

4.2 Dynamic programming – sequence matching

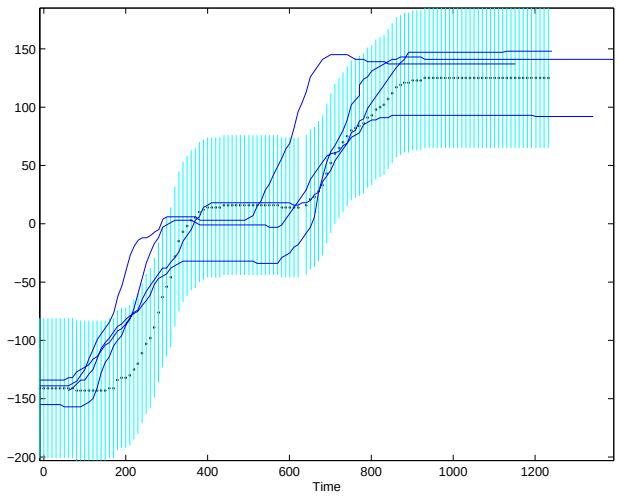
By again considering the quantized velocity vectors, a gesture can be represented as a sequence of symbols. Matching can then be perfomed using a dynamic programming algorithm to calculate the difference between any observed sequence and a stored prototype. (see Kojima and Oka (1995) for a dynamic programming approach for gesture recognition) The algorithm used (see Figure 8) is robust with respect to insertions, deletions and substitutions. Training is acheived by simply storing (possibly multiple) quantized sequences for each training gesture.

The distance between an observed gesture and a prototype can be efficiently computed at each time step. This can be used to estimate a probability by normalizing the differences across the gesture models. More effectively, a Gaussian function of the distances can be used, which penalizes large distances more heavily.

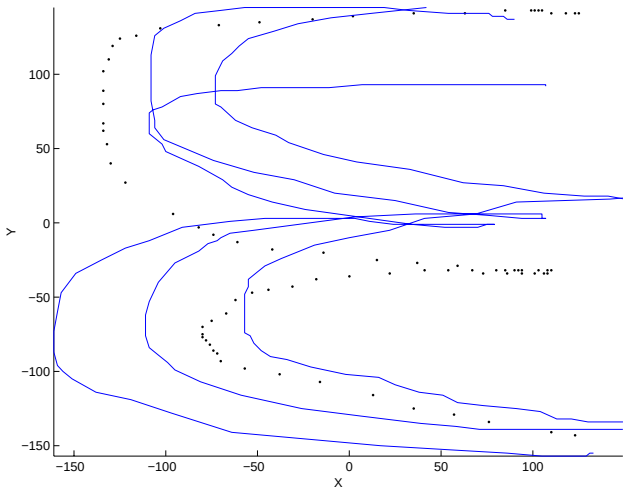
This recognition approach is very time efficient, although it can have a fairly high space overhead if many training examples of gestures are created. It generally has very good performance, especially on small training sets. In tests, in consistently outperformed the HMM for



(a) x trajectory



(b) y trajectory



(c) x & y

Figure 7: A simple two dimensional gesture, showing the independent x and y plots as well as the two-dimensional shape. Prototype gestures are shown, along with a shaded area indicating one standard deviation of a Gaussian distribution around the prototype, as a crude representation of variance in the system.

The algorithm implemented uses the following recurrence relation to initialize a matrix of values A. A has size $N \times M$ where N and M are the length of the sequence vectors X and Y respectively.

$$a_{ij} = \begin{cases} i, & j = 0 \\ j, & i = 0 \\ a_{(i-1)(j-1)}, & i,j > 0 \text{ and } x_{i-1} = y_{j-1} \\ 1 + \big(\min(a_{(i-1)(j-1)}, & \\ \quad a_{(i-1)j}, a_{i(j-1)}\big), & i,j > 0 \text{ and } x_{i-1} \neq y_{j-1} \end{cases}$$

The difference between the two sequences is then $a_{(n-1)(m-1)}$.

Figure 8: Sequence matching algorithm

gestures trained with fewer than three protoypes. Because it is not subject to training issues, the feedback does not suffer from the skewing or present in the HMM approach.

5 Implementation details

We have developed a general software architecture for developing and testing gesture recognition systems. This system, known as SIGIL, is used as the implementation platform fo the algorithms described. The system allows recognition and sonification elements to be treated as separate entities. HMM and dynamic programming recognition models were implemented as elements, and granular and dissonance sonification models were also implemented. Similarly, a zero-crossing dynamics sonification element was constructed.

The system is implemented in Java and provides a visual programming interface for the construction of various configurations of the fundamental elements of a recognition and sonification systems. A screenshot of the system is shown in Figure 9.

6 Conclusions

We have presented a framework for sonifying trajectories in a suitable state space, which is directly relevant to the audio enhancement of gesture recognition systems. An approach based on the direct sonification of salient features of the dynamics of a gesture device has been described. The granular synthesis approach provides a elegant and flexible way of sonifying elements of a state space. It can produce an audio display of any probability density function, which means that, in combination with appropriate recognition models, it can display changing beliefs in real time. This can be used to improve the quality and efficiency of interaction with gesture-controlled systems.

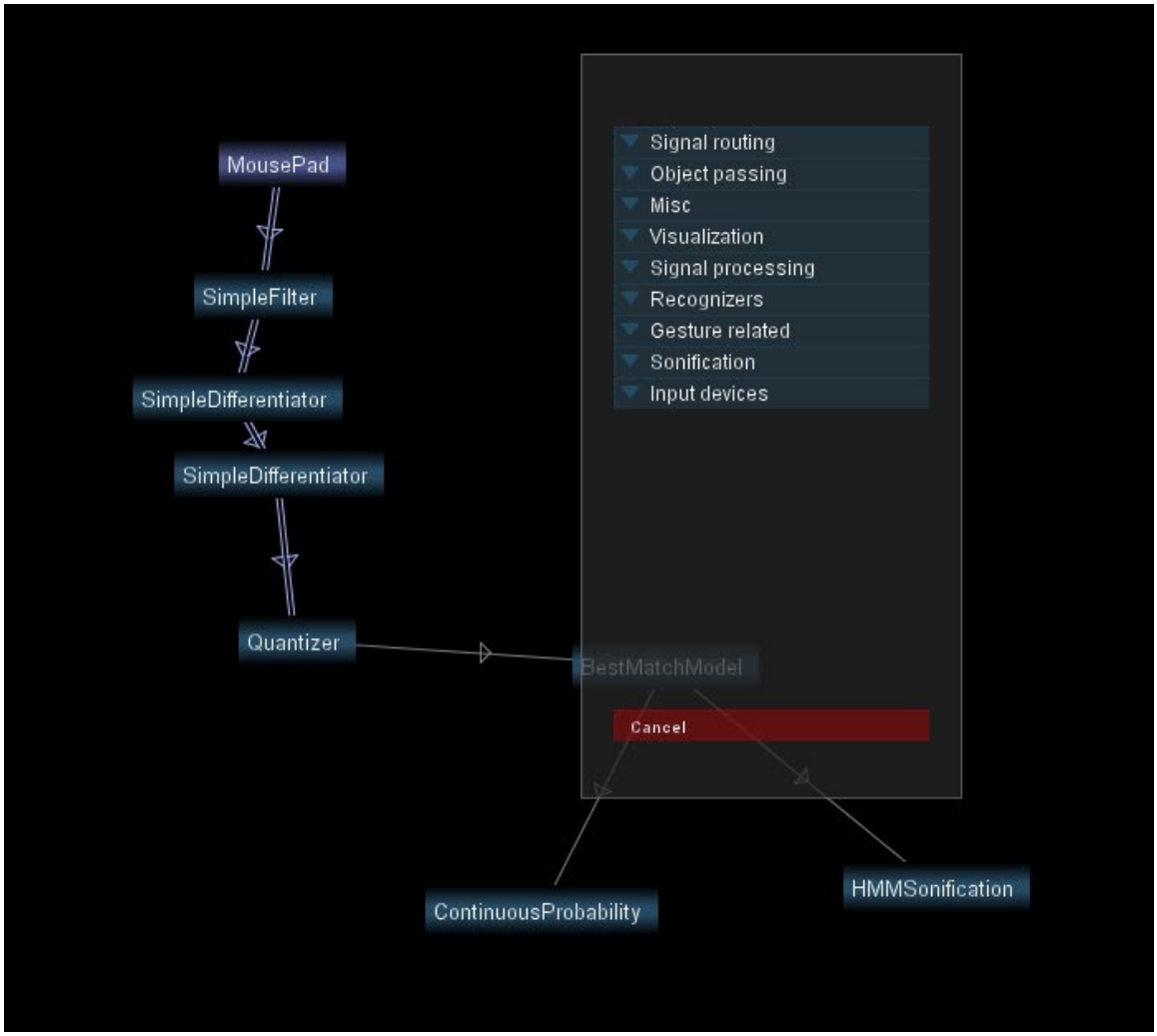


Figure 9: The SIGIL system showing a sonification/recognition setup

References

Brown, L., S.A. Brewster, R. Ramloll, B. Reidel, and W Yu (2002). Browsing modes for exploring sonified line graphs.. In: *Proceedings of British HCI*. Vol. 2. pp. 2–5.

Ghez, C., T. Rikakis, R. L. DuBois and P. R. Cook (2000). An auditory display system for aiding interjoint coordination. In: *ICAD'2000*.

Jorda, S. (2001). New musical interfaces and new music-making paradigms. In: *New Instruments for Musical Expression Workshop*.

Kojima, S. and N. Oka (1995). Efficient gesture recognition algorithm based of continuous dynamic programming. In: *Proc. of RWC Symposium*. pp. 47–48.

Lee, C. and Y. Xu (1996). Online, interactive learning of gestures for human/robot interfaces. In: *IEEE Int. Conf. on Robotics and Automation*. pp. 2982–2987.

McQueen, C. and M. Mantei (1994). Audio strokes: Using sound as a continuous feedback mechanism. In: *UIST'94*.

Morimoto, C., Y. Yacoob and L. Davis (1996). Recognition of head gestures using hidden markov models. In: *International Conference on Pattern Recognition, Vienna*. pp. 461–465.

Mulder, A. (1994). Virtual musical instruments: Accessing the sound synthesis universe as a performer. In: *International Symposium on Computer Music*.

Muller-Tomfelde, C. and S Steiner (2001). Audio-enhanced collaboration at an interactive electronic whiteboard. In: *ICAD'2001*.

Rabiner, L. and B. H. Juang (1993). *Fundamentals of Speech Recongition*. Prentice Hall.

Ramsay, J. O. and B. W. Silverman (1997). *Functional Data Analysis*. Springer-Verlag.

Roads, C. (1978). Granular synthesis of sounds. *Computer Music Journal* **2**(2), 61–68.

Shi, J. Q., R. Murray-Smith and D. M. Titterington (2002). Hierarchical Gaussian process mixtures for regression. Technical Report TR-2002-107. University of Glasgow, Scotland, UK.

Truax, B. (1988). Real-time granular synthesis with a digital signal processor. *Computer Music Journal* **12**(2), 14–26.

Wilson, A. D. (2000). Adaptive Models for Recognition of Human Gesture. PhD thesis. MIT.

Wilson, Andrew D. and Aaron F. Bobick (1999). Parametric hidden markov models for gesture recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **21**(9), 884–900.

Xenakis, I. (1971). *Formalized Music: Thought and mathematics in composition*. Indiana University Press.

Yang, J. and Y. Xu (1994). Hidden markov models for gesture recognition. Technical Report CMU-RI-TR-94-10. Carnegie Mellon University.

Acknowledgements

Both authors are grateful for support from EPSRC grant *Modern statistical approaches to off-equilibrium modelling for nonlinear system control* GR/M76379/01, and *Audioclouds: three-dimensional auditory and gestural interfaces for mobile and wearable computers* GR/R98105/01.

RM-S acknowledges the support of the *Multi-Agent Control* Research Training Network – EC TMR grant HPRN-CT-1999-00107.