AUTOMATIC LOCATION AND MEASUREMENT OF ORNAMENTS IN AUDIO RECORDINGS

Michael Casey and Tim Crawford

Centre for Cognition, Computation and Culture
Goldsmiths College
University of London

ABSTRACT

This paper reports on our first experiments in using the feature extraction tools of the MPEG-7 international standard for multimedia content description on a novel problem, the automatic identification and analysis of score-based performance features in audio recordings of music. Our test material consists of recordings of two pieces of 17th- and 18th-century lute music in which our aim is to recognise and isolate performance features such as trills and chord-spreadings. Using the audio tools from the MPEG-7 standard facilitates interoperability and allows us to share both score and audio metadata. As well as using low-level audio MIR techniques within this MPEG-7 context, the work has potential importance as an 'ornamentation filter' for MIR systems. It may also form a useful component in methods for instrumental performer identification.

1. INTRODUCTION

A perennial challenge in Music Information Retrieval is the reconciliation of the audio and symbolic domains of music representation, [4]. While it can be argued that audio represents everything that most audiences might reasonably be expected to perceive about a piece of music, it lacks significant information that is explicitly present in a symbolic representation. The task of recovering information like formal and harmonic structure, voice-leading and even note-detail from any but the simplest of audio textures is still the subject of much research activity. [1, 2, 8, 9, 11, 12, 13, 14, 15, 18, 19].

One way in which score and audio representations of a piece of music differ is in the matter of those low-level performance gestures that are generally classed under the heading of 'ornamentation'. We use the term here somewhat broadly to include all 'extra' note events; this embraces some classes of the temporal redistribution of notes, in particular the spreading of chords which are basically notated as a single event or a set of simultaneously-starting notes but are performed as a succession of discrete note onsets.

Scores usually indicate ornaments by the use of symbols from a quite extensive lexicon of signs which has accumulated and evolved over several centuries,[10]. A standard subset of at most a few dozen is more-or-less universally accepted in today's Common Music Notation, and each has a roughly standard conventional interpretation.

Intermediate representations such as MIDI files may, or may not, include 'realisations' of these ornament signs; they may be separated from the 'score' notes in the file, or they might simply be included in the general 'mix' of notes. Ornament realisations in MIDI files might be historically appropriate to the music or merely fanciful or instinctive, just as those in audio recordings might be.

Often ornamentation gestures are added to music even when they are not prescribed explicitly in a score. This is true for genres other than classical instrumental music; for example, opera, jazz, blues, soul and all kinds of popular and traditional music have their own lexica of ornaments which can be added spontaneously, even when devices such as tempo deviation are not used, [22, 16].

We would like to automate the process of recognising performance gestures directly from audio for various reasons. Firstly, the analysis of performance on audio recordings is rapidly gaining ground as an important area of musicological research, bearing strongly as it does on how people understand and 'feel' music; see [5]. Most research in this field has to operate at the broad-brush level of tempo-deviation comparisons, or by manual examination of spectographic images; see [6]; see also [20]. As far as we can see, very little if any work has been done on tools that can locate, identify and measure detailed ornamental gestures within an audio recording. Secondly, an audio performance of a piece of music with ornamentation naturally contains significantly more 'notes' than the score representation; for MIR or analysis techniques that depend on matching score and audio, this strongly raises the chances of false or confused matches. An 'ornamentation-filter' might prove to be a useful pre-processing tool for MIR systems. Thirdly, a further interesting MIR challenge is whether it is possible to use the output of an ornament-recognition tool to aid in the problem of identifying instrumental performers directly from audio where timbral features are more-or-less invariant between players (such as in piano or organ recordings, for example).

This paper reports our first steps in developing ornament detection and recognition applications using MPEG-7 feature extraction tools. At this early stage we have concentrated on a specialised repertory that crucially depends on ornamentation for its effect, 18th-century lute music. The music involved is a very small part of a considerable repertory of lute music being acquired and studied in the ECOLM corpus-building project, funded by the UK Arts and Humanities Research Board, [7]. Though audio does not form part of the scope of that project, it may provide us with suitably marked-up encodings which represent the scores down to a sufficient level of detail.

Two contrasting pieces by Silvius Leopold Weiss (1687-1750) were examined for this paper, each of them existing in commercial recordings.

The first is his best-known work, the 'Tombeau pour le Comte de Logy' (1721). A Tombeau is a doleful work which commemorates the death of a famous person, somewhat in the manner of a musical funeral oration, in this case Count Jan Antonin Losy (or Logy) (c1650-1721), said to be the finest aristocratic lute player of his age. Its marking is 'Adagio' and it is taken very slowly indeed by each of our players. At the opening of the piece is a series of repeated chords marked to be spread rather than played in a single stroke. The second piece is a Passacaille in D major. This is composed on a falling ground bass, which is repeated 12 times and over which the composer weaves a varied texture of chordal and melodic variations. Chord-spreading does not form an explicit feature of the notation of this piece. Both pieces contain performance markings such as trills or appoggiaturas (the same sign is used for both), left-hand slurs and (at the beginning of the Tombeau only) explicit chord-spreading. However, in fact, all performances contain examples of chords that are spread despite not being explicitly so marked in the score. See Figure 1 and Figure 3 for tablature and scores of the opening phrases of these works.

For both pieces, we prepared special MIDI files of the scores; these contain, as well as the sounding notes indicated in the tablature, the ornament and chord-spreading signs represented as Meta Text events. This provided a symbolic reference point to which the recorded performances can be related.

For the *Tombeau* we looked at three commercial CD recordings played on typical baroque lutes of the period. In the case of the Passacaille, we were able to use five recordings in all, two on baroque lute and three on other instruments: modern classical guitar, piano and lute- harpsichord. The lute- harpsichord was a gut-strung harpsichord specifically designed to imitate the sound of the lute; JS Bach, a friend of SL Weiss, is said to have taken a special interest in the instrument's development.

2. APPROACH

Our corpus consisted of uncompressed commercial recordings of performances of the two above-mentioned works by SL Weiss; *Tombeau* and *Passacaille*.

The works are polyphonic and were written for the 18th-century 13-course baroque lute, an instrument played in similar fashion to the guitar, but with a series of diatonically-tuned bass strings extending an octave below the six normal playing strings. All but the two highest-sounding strings are arranged in pairs (known as 'courses'); the eight lowest-pitched courses (A1-A2) are tuned in octaves.

Each piece consists of multiple sections that may be optionally repeated by the performer. There were three lute performances of each piece by different performers; there were also further performances of the Passacaille in arrangements for another three instruments: modern classical guitar, modern piano and lute-harpsichord (an 18th-century hybrid gut-strung harpsichord specifically designed for keyboard players to imitate the lute).

The scores in which the music has come down to us are written in the special form of notation known as tablature, which, as well as the notes to be played, indicates certain other performance features by means of special signs at the appropriate point. Just as with harpsichord, organ or piano music, most of these ornamental markings involve the addition of notes which are extra to the main notes. In these cases the ornament signs can be thought of as a short-hand to instruct the performer to insert additional notes from a note pattern 'codebook'. Other cases, such as chord-spreading or vibrato signs involve the application of procedures to the notes, rather than adding note patterns. This codebook varies significantly between periods, composers, performers, instruments and musical styles. Thus, we have in general little prior knowledge about how a given ornament will be performed. Even if concrete rules for a given period are derived from historical sources, these sources tend to disagree in detail, and in any case there is no guarantee that a performer would feel obliged to follow any particular set of rules consistently. Furthermore, even when following prescribed rules, a performer has the liberty to add ornamentation even when not explicitly notated in the score.

We therefore chose to limit our investigations to a small subset of the possible performance options open to a player. These all involve the occurrence of note-events extra to those notated in the score (or MIDI file). In the case of appoggiaturas, trills, mordents, turns, etc., the extra notes tend to be manifested as very short events clustered around the point where the 'main' note is expected to be. In the case of chord-spreading, the chord (a single multi-note event in the score) actually is performed as a sequence of note-onsets (each generating a sustained tone) clustered around the point where the chord might be expected to sound if it were a single event. The spreading of chords, especially in slow music, is crucial in suggesting metrical or harmonic stress and expressive intent; for this reason it is often very precisely notated in 19th-century piano music, yet one always expects a performer to spread all chords to some extent (indeed it is almost technically impossible not to do so).

2.1. Ornament Classifier Construction

2.1.1. MPEG-7 Standard Features

We used standardised features from the MPEG-7 International Standard for constructing the ornament classifier, [23]. Both Matlab and C++ reference software implementations are available from *musicstructure.com*. From the audio low-level descriptor framework (LLD) we used *AudioSpectrumEnvelopeD*, *AudioSpectrumBasisD* and *AudioSpectrumProjectionD*. The following spectral attributes were used in all the experiments:

Attribute	Value
loEdge	62.5 Hz
hiEdge	8000 Hz
resolution	8 bands per octave
sampleRate	44100 Hz
windowLength	30 ms
window	Hamming
hop	10 ms
FFT Length	2048

Table 1. MPEG-7 spectral parameters used for ornament detection.

AudioSpectrumEnvelopeD was calculated by transformation of a Short-Time Fourier Transform (STFT) Power Spectrum:

$$\mathbf{X} = |FT|^2_{\frac{N}{2}+1}\mathbf{T} \tag{1}$$

The linear transform matrix, $\mathbf{T}$, partitions the STFT values into $\frac{1}{8}$th octave logarithmically-spaced frequency bands in the range $62.5 - 8000Hz$ such that the sum of powers in each spectral band is preserved under transformation. In practice, the low-frequency resolution of the resulting log frequency spectrum is dependent upon careful multi-resolution implementation of the linear transform, detailed discussion of which is outside the scope of this paper.

The primary feature used was *AudioSpectrumProjectionD* and was calculated by converting *AudioSpectrumEnvelopeD* to a decibel scale and applying a decorrelating linear transform matrix, $\mathbf{V}$, consisting of K cepstral basis functions. In MPEG-7, the decorrelating basis is called *AudioSpectrumBasisD*:

$$\mathbf{Y} = 10\log_{10}(\mathbf{X})\,\mathbf{V} \tag{2}$$

Equations 1 and 2 describe a feature that is similar to the widely-used Mel-frequency Cepstral Coefficients (MFCC) feature. If we choose $\mathbf{T}$ to be the linear-to-Mel frequency map and $\mathbf{V}$ to be the discrete cosine transform (DCT) then *AudioSpectrumProjectionD* will be equivalent to MFCC. Most MFCC-based studies use the first few cepstrum coefficients, $K \leq 20$, which relate to wide-band spectral features such as formants.

It is our view that this feature ranking disregards narrow-band pitch-based information in the spectrum which is contained in the higher order coefficients. To admit sensitivity to pitch content, a large number, $K > 20$, of MFCC coefficients would need to be used. This is disadvantageous because higher dimensional features perform poorly in classification tasks. This negative property of larger feature vectors is known as the *curse of dimensionality* in machine learning and pattern recognition literature, [24].

To achieve a balance between feature compactness and sensitivity, MPEG-7 uses SVD or ICA analysis of the cepstral feature space, [25] [26]. Thus *AudioSpectrumBasisD* and the corresponding *AudioSpectrumProjectionD* coefficients were derived using the SVD:

$$[\mathbf{Y}, \mathbf{S}, \mathbf{V}] = \text{SVD}\left(10\log_{10}(\mathbf{XT})\right), \tag{3}$$

and the resulting basis functions, $\mathbf{V}$, and coefficients, $\mathbf{Y}$, were truncated to 20 dimensions yielding optimally compact cepstral features.

2.1.2. Hidden Markov Models

Our approach models neither the instrument nor the perfomer explicitly. Instead, a hidden Markov model (HMM) of each performance was inferred from the *AudioSpectrumProjectionD* data using Bayesian inference. Models were initialised using a fully connected 40-state model and model parameters were inferred using maximum *a-posteriori* (MAP) estimation, model trimming and parameter extinction to force low-probability parameters toward zero, [29].

Given an HMM, the Viterbi algorithm estimates the optimal state sequence $\{\alpha_i\}_{i=1}^{N_\alpha} \in \{1, 2, \cdots, S\}$, for an HMM with S states. The HMM effectively quantizes the *AudioSpectrumProjectionD* feature vectors to a $1d$ time series of integers. This technique is consistent with other discrete-quantization approaches to audio-based music information retrieval, [27][25], and forms a part of the MPEG-7 international standard under the descriptors *SoundModelDS* and *SoundModelStatePathD*.

Figures 2 and 4 illustrate *piano rolls* of the state sequences obtained by the Viterbi algorithm used with each performance's HMM. The ornament *ground truth* judgements, provided by expert listeners, are also indicated by boxed regions. Comparison with the scores illustrates that steady-state sections of audio are expressed as repeated states and dynamic spectral behaviours, such as note onsets and ornaments, are characterised by the state transitions.

2.1.3. HMM State Transition Count (STC)

It is our hypothesis that HMM state transitions occur at higher rates during ornaments than during non-ornamented segments of an audio signal. To test this hypothsis we constructed an audio-based ornament classifier using the HMM state transition count (STC) calculated over a moving fixed-length window.

To detect transitions we located the non-zero first differences of the HMM state sequence. State transition locations were defined as the sample points where changes in state occurred. The state transition indicator function of the HMM state sequence, $I(\alpha_i)$, marked state transition positions with a 1 and steady-state positions with a 0:

$$I(\alpha_i) = \begin{cases} 1 & \text{if} \qquad\qquad |\alpha_i - \alpha_{i-1}| > 0 \\ 0 & \text{otherwise.} \end{cases}$$

We divided the state transition indicator array into overlapping fixed-length windowed subsequences. The window duration was varied from 100ms to 1000ms with a hop of 10ms. The sum of the elements in the windowed segments provided a moving state transition count $\{\Phi_t\}_{t=1}^{N_\alpha}$. The state-transition count was smoothed with a hamming window which was also varied between 100ms and 1000ms.

The local maxima of windowed transition counts were located at the positive-to-negative zero crossings in the first-order difference function $\Phi_t - \Phi_{t-1}$. Figure 2 and Figure 4 illustrates the transition counts in relation to ornament ground truths for a performance of both the *Tombeau* and the *Passacaille*.

Figure 1. (upper) tablature notation (lower) score transcription of *Tombeau*.

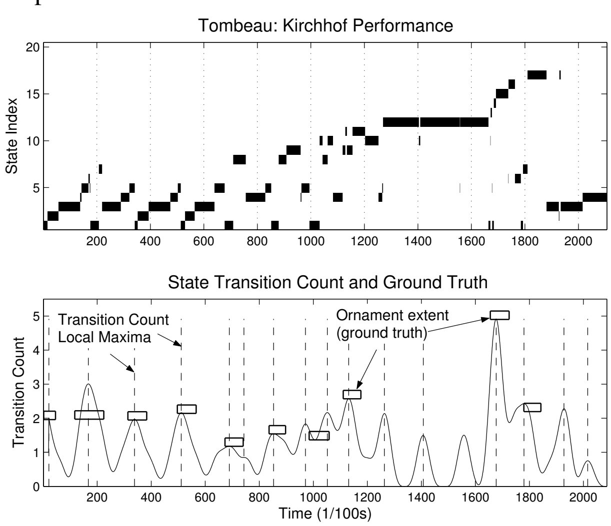

Figure 2. (upper) state *piano roll* (lower) ornament ground truth and state transition counts for a performance of *Tombeau*.

Figure 3. (upper) tablature notation (lower) score transcription of *Passacaille*.

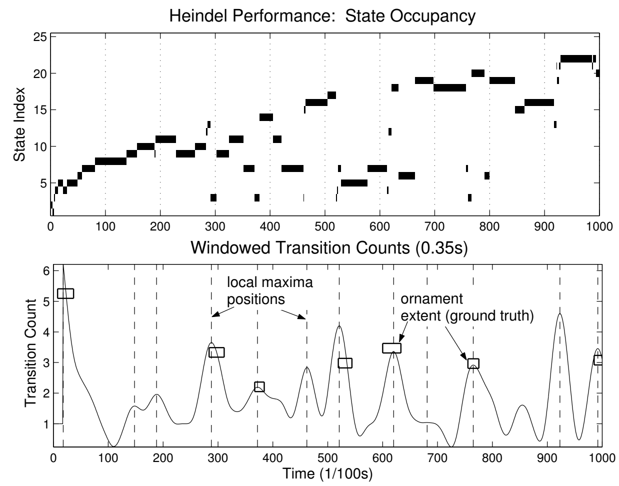

Figure 4. (upper) state *piano roll* (lower) ornament ground truth and state transition counts for a performance of the *Passacaille*.

2.1.4. Ranking and Evaluation

To test the transition count hypothesis we evaluated its effectiveness in a number of binary ornament classification experiments. The time locations of transition count maxima, $\mu \in \{M\} \subset \{1 \ldots N_\alpha\}$, were ranked and sorted by descreasing transition count such that $\Phi_{\mu_i} \geq \Phi_{\mu_{i+1}}$.

To evaluate the performance of the binary ornament classifier we measured the precision and recall of ornaments with respect to a ground truth set, $\{\Omega_j\}_{j=1}^{N_\Omega}$, provided by an expert listener for each performance. Maxima locations and ornament locations were considered intersecting if the location of the maxima occurred between the start and end point (inclusive) of an ornament ground-truth judgement.

Precision was defined as the number of intersections between each candidate list of maxima locations and the ground truth ornament locations normalised by the length of the candidate list:

$$P_L = \frac{\{\Phi_i\}_{i=\mu_1}^{\mu_L} \bigcap \{\Omega_j\}_{j=1}^{N_\Omega}}{L}, L = \{1...|\{M\}|\}. \qquad (4)$$

Recall was defined in a similar manner, but the normalisation term was the total number of ground-truth ornament locations:

$$R_L = \frac{\{\Phi_i\}_{i=\mu_1}^{\mu_L} \bigcap \{\Omega_j\}_{j=1}^{N_\Omega}}{|N_\Omega|}, L = \{1..N_M\}. \quad (5)$$

Classifiers were constructed over the parameter space of transition count and smoothing window lengths, each taking values in the range 100ms to 1000ms. To determine the performance of a classifier with parameters W_c and W_s, the length of the counting window and smoothing window respectively, the precision and recall were calculated for multiple performances against each performance's ground-truth judgements. Following standard IR practice, the recall values were interpolated to standardised recall levels of 10%, 20%, ..., and the precision at each standardised recall level was averaged across the test group of performances. The mean of precision and recall at the break-even point was used as a ranking of the classifier's performance. The highest performing parameters over each group of test performances are reported below.

3. RESULTS

3.1. Tombeau

In the following experiments, the window lengths W_c and W_s were varied in the range 100ms-1000ms for each test group. In the first experiment, the classifiers were tested on three lute performances of the opening of the *Tombeau*. This sequence contained 10 chord spreads and two slurs with minor differences in the number of ornaments between performances. The results for the best performing classifier, scored according to the break-even criterion, are show in Figure 5 and Table 2. The optimal parameters for this test group were a transition count window length of 400ms and a smoothing window of 900ms performing at a precision of 77% at a standardised recall rate of 80%.

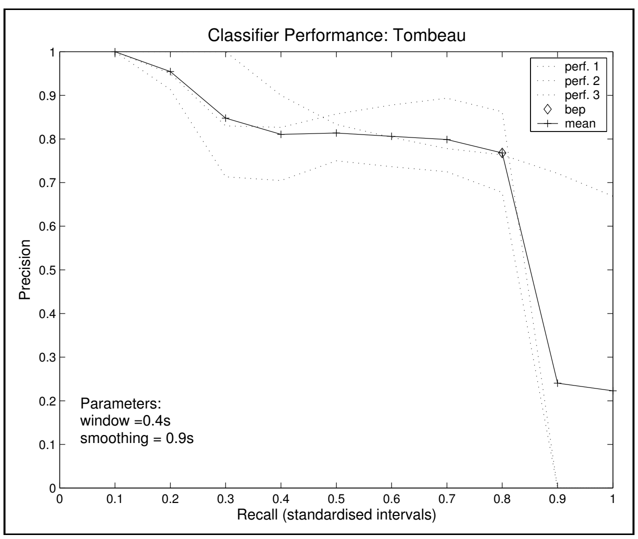

Figure 5. Precision-recall curves for three performances of Tombeau on lutes only. The break-even point is at 77% mean precision at a recall rate of 80%.

Recall	0.100	0.200	0.300	0.400	0.500
Precision	1.000	1.000	1.000	0.900	0.833
	1.000	0.913	0.713	0.705	0.750
	1.000	0.950	0.830	0.827	0.857
Mean	1.000	0.954	0.848	0.810	0.813
Recall	0.600	0.700	0.800	0.900	1.000
Precision	0.804	0.778	0.764	0.721	0.669
	0.737	0.725	0.678	0.000	0.000
	0.878	0.893	0.862	0.000	0.000
Mean	0.806	0.799	0.768	0.240	0.223

Table 2. Summary of precision-recall results for experiment 1 for three performances of the *Tombeau*.

3.2. Passacaille

The second experiment tested classifier performance on three performances of the opening phrases of the *Passacaille*, two on lutes and one on lute-harpsichord. A larger number of ornaments with more varied expression occured in these performances than for the *Tombeau* experiment. Ornaments included vibrato, slurs, trills, mordents, chord spreads, and appoggiaturas. The number of ground truth ornaments were 13, 19 and 20 respectively for each of the performances thus giving a total relevant set of 52 targets.

The best performing classifier in this experiment consisted of window parameters $W_c = 350$ms and $W_s = 500$ms. The precision was 68% at a recall rate of 70%, see Figure 6 and Table 3.

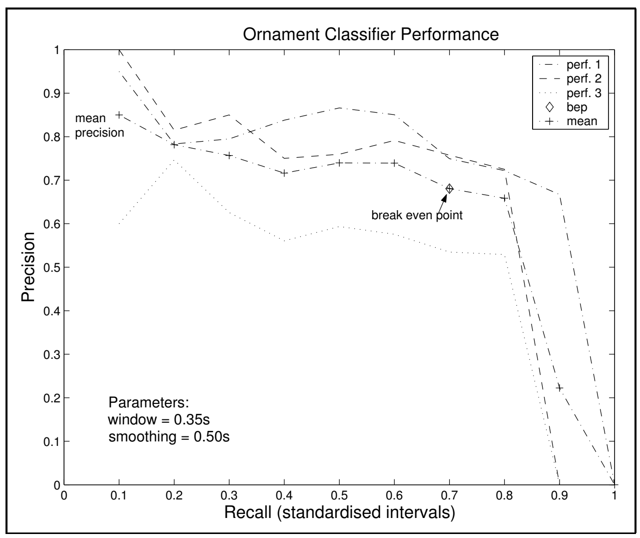

Figure 6. Precision-recall curves for three performances of the *Passacaille*. The break-even point is 68% mean precision at 70% recall.

The third experiment consisted of a more varied set of performances than the second; three lutes, lute harpsichord and modern acoustic guitar. 84 ground truth ornaments were identified as targets in this experiment. The parameters for the best performing classifier were $W_c = 300$ms and $W_s = 500$ms. The mean precision at the

Recall	0.100	0.200	0.300	0.400	0.500
Precision	0.950	0.783	0.795	0.838	0.866
	1.000	0.815	0.850	0.750	0.760
	0.600	0.747	0.626	0.561	0.593
Mean	0.850	0.782	0.757	0.716	0.740
Recall	0.600	0.700	0.800	0.900	1.000
Precision	0.851	0.750	0.722	0.667	0.000
	0.791	0.758	0.725	0.000	0.000
	0.576	0.535	0.529	0.000	0.000
Mean	0.739	0.681	0.659	0.222	0.000

Table 3. Summary of results of ornament classification for three lute and lute-harpsichord performances of the *Passacaille*.

break-even point was 60% at a recall rate of 60%, see Figure 7 and Table 4.

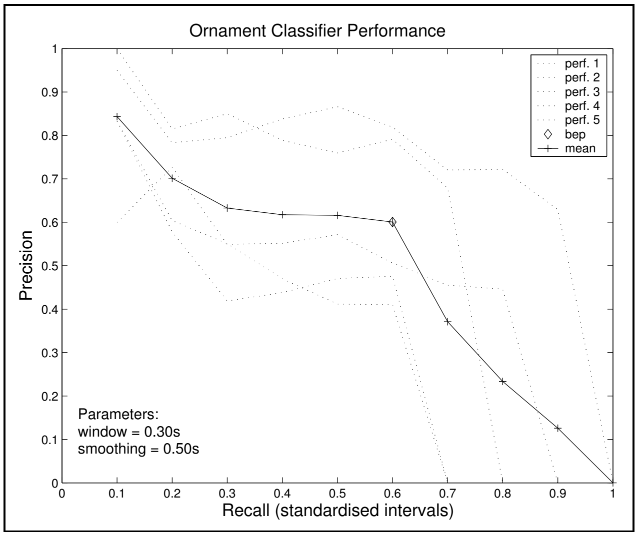

Figure 7. Precision-recall curves for five performances of the *Passacaille* tested on a range of instruments; three lutes, guitar and lute-harpsichord. The break-even point was 60% mean precision at 60% recall.

3.3. Discussion

Our results indicate that the best performing classifiers perform at a reasonable precision-recall rate and therefore somehwat support our hypothesis that windowed HMM state transition counts are an estimator of ornamented regions in audio recordings of solo polyphonic instrument performances. The performance does, however, appear to diminish with the addition of more varied ornamentation and instrumentation. This will impact the scalability of the classifiers to significantly larger corpora.

For a more scalable classifier, the fixed window parameterisation applied over all performances is too restrictive so further work will be carried out to evaluate window parameters that adapt to each performance. For example, the state transition count feature is insensitive to the surrounding context; in a performance with many short-duration notes in the notated score, ornaments might not be easily discriminated from the highly active background. In the future we will explore the relationship between the heights and the widths of the STC peaks thus seeking to adapt to the mean density of non-ornamental onsets in each performance.

Recall	0.100	0.200	0.300	0.400	0.500
Precision	0.950	0.783	0.795	0.838	0.866
	0.833	0.578	0.419	0.438	0.471
	0.833	0.604	0.551	0.470	0.412
	1.000	0.815	0.850	0.789	0.760
	0.600	0.727	0.549	0.552	0.571
Mean	0.843	0.701	0.633	0.617	0.616
Recall	0.600	0.700	0.800	0.900	1.000
Precision	0.819	0.720	0.722	0.629	0.000
	0.476	0.000	0.000	0.000	0.000
	0.410	0.000	0.000	0.000	0.000
	0.791	0.679	0.000	0.000	0.000
	0.506	0.455	0.446	0.000	0.000
Mean	0.600	0.371	0.234	0.126	0.000

Table 4. Summary of results of ornament classification for five performances of the *Passacaille* on lutes, guitar and lute-harpsichord.

4. CONCLUSIONS

We developed an HMM-based ornament classifier that automatically locates ornaments in audio recordings and evaluated performance against human-judged ground truths for a set of recordings. We hypothesised that windowed HMM state transition counts could be used for labelling regions of audio corresponding to ornaments and provided some experimental evidence to support this claim.

The utility of the methods presented herein will be explored in further work, including classification of individual ornament species and ornament 'filtering' for audio-based information retrieval using score-based queries.

5. REFERENCES

[1] Abdallah, S. and Plumbley, M., "An independent component analysis approach to automatic music transcription ", in *Proc. of the 114th Convention of Audio Engineering Society*, Amsterdam, March 2003.

[2] Bello, J.P., "Towards the automated analysis of simple polyphonic music: A knowledge-based approach", Ph.D. thesis, University of London, 2003.

[3] Brown, J.C. and Smaragdis, P., "Independent Component Analysis for Automatic Note Extraction from Musical Trills " , *J. Acoust. Soc. Am.* 2004. in press.

[4] Byrd, D. and Crawford, T. "Problems of music information retrieval in the real world", *Information Processing and Management*, 38(2): 249-272, 2002.

[5] The Centre for the History and Analysis of Recorded Music (CHARM) at Georgetown (US) and Southampton (UK) Universities: http://www.georgetown.edu/departments/AMT/music/bowen/CHARM.html

[6] UK Charm's "Analytical techniques" page at: http://www.soton.ac.uk/ musicbox/charm5.html

[7] http://www.ecolm.org

[8] Fujishima, T., "Realtime Chord Recognition of Musical Sound: a System Using Common Lisp Music", *Proceedings of the 1999 International Computer Music Conference*, pp. 464-467, Beijing, China

[9] Goto, M. and Hayamizu, S., "A real-time music scene description system: detecting melody and bass lines in audio signals". In *Proc. IJCAI Workshop on CASA*, pages 31-40, 1999.

[10] "Ornaments ", 11. Index to ornaments and table of signs, *New Grove Dictionary of Music and Musicians*, New York and London, 2000. Online at www.grovemusic.com

[11] Hainsworth, S. and Macloed, M. "The Automated Music Transcription Problem" http://www-sigproc.eng.cam.ac.uk/ swh21/ontrans.pdf

[12] Leman, M. and Carreras, F., "The Self-Organization of Stable Perceptual Maps in a Realistic Musical Environment", *Proceedings of JIM96, Journees d'Informatique Musicale* 16-18 May 1996, Caen, France.

[13] Martin, K. D., "Automatic Transcription of Simple Polyphonic Music: Robust Front End Processing." *M.I.T. Media Lab Perceptual Computing Technical Report* 399, November 1996, presented at the "Third Joint Meeting of the Acoustical Societies of America and Japan", December, 1996. http://xenia.media.mit.edu/ kdm/research/papers/kdm-TR399.pdf

[14] Martin, K. D., "A Blackboard System for Automatic Transcription of Simple Polyphonic Music". M.I.T. Media Lab Perceptual Computing Technical Report 385, July 1996. http://xenia.media.mit.edu/ kdm/research/papers/kdm-TR385.pdf

[15] Martin, K. D. and Scheirer, E. D., "Automatic Transcription of Simple Polyphonic Music: Integrating Musical Knowledge". Presented at *SMPC*, 1997.

[16] Mitchell, D.W. "A comparison of embellishments in performances of bebop with those in the music of Chopin", MM, Music: University of Arizona, 1992.

[17] G. Monti and M. Sandler, "Automatic polyphonic piano note extraction using fuzzy logic in a blackboard system ", in *Proc. of DAFx-02*, Hamburg, Germany, 2002.

[18] Pickens, J., Bello, J.P., Crawford, T., Dovey, M., Monti, G., Sandler, M. and D. Byrd, "Polyphonic score retrieval using polyphonic audio queries: A harmonic modelling approach ", in *The Journal of New Music Research*, vol. 32, no. 2, June 2003.

[19] Eric Scheirer: web page on audio transcription: http://web.media.mit.edu/ eds/transcription.html

[20] *Sforzando*, Windows software program designed for analysing tempo and dynamic change in recordings at: http://www-personal.usyd.edu.au/ rjohnson/sforzando/

[21] Smaragdis, P., and Brown, J.C., "Non-Negative Matrix Factorization for Polyphonic Music", *2003 IEEE Workshop on Applications of signal Processing to Audio and Acoustics*, October 19-22, 2003, New Paltz, NY.

[22] Williams, J. K., "A method for the computer-aided analysis of jazz melody in the small dimensions". *Annual review of jazz studies*, USA Vol. III (1985) 41-70.

[23] ISO 15938-4(2002) *Multimedia Content Description Interface* Part 4 (Audio). International Standards Organization 2002.

[24] Duda, R., Hart, D. and Stork, D. *Pattern Classification*, 2nd Ed. New York: Wiley, 2002.

[25] Casey, M., "MPEG-7 Sound Recognition Tools", *IEEE Transactions on Circuits and Systems: Video Technology*. IEEE, June 2001.

[26] Casey, M., "General Audio Information Retrieval", in Innocenti, P. (Ed.), *MMIR: Multimedia Information Retrieval*. AIDA, 2004.

[27] Foote,J., "A Similarity Measure for Automatic Audio Classification." In *Proc. AAAI 1997 Spring Symposium on Intelligent Integration and Use of Text, Image, Video, and Audio Corpora*. Stanford, March 1997.

[28] Hu, N., Dannenberg, R. B. and Tzanetakis G. "Polyphonic Audio Matching and Alignment for Music Retrieval" *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics* October 19-22, 2003, New Paltz, NY.

[29] Brand, M., "Pattern discovery via entropy minimization", in *Proc. Artificial Intelligence and Statistics #7.* 1998.