Audio Engineering Society

Convention Paper

Presented at the 110th Convention
2001 May 12–15 Amsterdam, The Netherlands

Segmentation of Musical Signals Using Hidden Markov Models.

Jean-Julien Aucouturier, Mark Sandler
Department of Electronic Engineering, King's College
London, U.K.

ABSTRACT

In this paper, we present a segmentation algorithm for acoustic musical signals, using a hidden Markov model. Through unsupervised learning, we discover regions in the music that present steady statistical properties: textures. We investigate different front-ends for the system, and compare their performances. We then show that the obtained segmentation often translates a structure explained by musicology: chorus and verse, different instrumental sections, etc. Finally, we discuss the necessity of the HMM and conclude that an efficient segmentation of music is more than a static clustering and should make use of the dynamics of the data.

1. INTRODUCTION

In this paper, we present an algorithm that segments a piece of music into a succession of abstract textures over time.

Like a MIDI score, a musical stream can be described as the superposition of different sources –i.e. instruments- emitting sound at the same time, each with its own timbre. In this work, "texture" is defined as the composite "polyphonic timbre" resulting from instruments playing together. (Identically, a texture can be viewed as the "monophonic timbre" of a composite meta-instrument). Figure 1 shows an example of instruments merging into such textures.

This process is similar to audio segmentation. Segmenting acoustic data refers to identifying and labeling its different sections of interest. For instance, if we process music, we would like to highlight the alternation of chorus and verse, the beginning of a solo, a sudden change of orchestration, etc. Tzanetakis in [1] stresses the importance of segmentation for Audio Information Retrieval, where it is better to consider a song as a collection of distinct regions than as a whole with mixed statistics. By discarding any pitch and harmonic information, and focusing only on textures, our algorithm leads directly to such a labeling of the relevant sections in music.

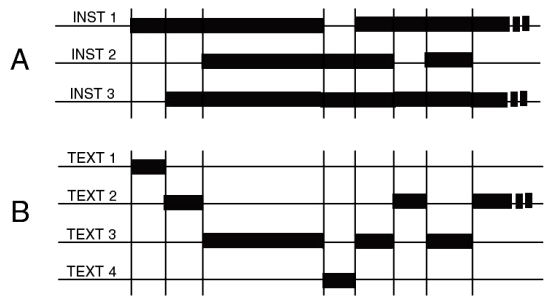

Figure 1: Comparison of the classic MIDI track representation of music (A) and the texture representation (B).

To uncover the different textures occurring in a musical piece, we investigate the use of Hidden Markov Models (HMM). Over the past 20 years, HMMs have been applied with great success to many pattern recognition applications, such as speech recognition, and see

growing interest in the music analysis community ([2], [3]). In our case, we believe that each state of an appropriately trained HMM can account for a specific texture.

The paper is organized as follows. In section 2, we discuss the most relevant signal processing front-end for the model. We examine three methods of spectral envelope estimation: cepstrum, linear prediction and discrete cepstrum, and suggest two ways to encode the resulting feature vector. Their qualities are compared.

In section 3, we present an efficient algorithm to uncover the succession of textures using a HMM. We compare the performance of the different front-ends suggested in section 2. We also address the main limitation of the system: its fixed topology.

Finally, in section 4, we justify the use of a HMM compared to more static clustering schemes, such as k-means clustering. We show that we often have to take account of the dynamic evolution of the features to achieve a good segmentation.

2- FRONT END

2.1- Requirements

It is of great importance to select an appropriate set of features as input to the hidden Markov model. To allow an efficient segmentation of the audio data, the features should meet the 2 following criteria:

a) The ideal feature set will be a perceptually realistic measure of the similarity of timbres. Similar textures must be represented by close "points" in the multi-dimensional feature space, and, the other way round, close points should represent similar timbres.

b) At the same time, since we don't want to segment the different notes or events within a single texture, the feature set should be relatively independent of pitch.

To assess the quality of a feature set given this compromise between timbre and pitch, we use two specially recorded audio samples. Sample1 is a 5-second clarinet glissando, comparable to the one opening G.Gershwin's "Rapsody in Blue". It is an example of a constant texture, with varying pitch. Sample2 is 5-second C4 note generated by MIDI, in which the instrument playing the note changes each 500ms: clarinet-bassoon-trumpet-organ-violin-piano-etc. It is an example of a varying texture with constant pitch. A quality measure of the feature set is obtained by comparing the standard deviation of the features computed on both samples: a good front end would show great variance on sample2, and little on sample1.

2.2- A good candidate: the Spectral Envelope.

There has been a substantial amount of research on timbre and instrument separation, in most of which the analyzed acoustic data consist of short monophonic samples of a simple instrument. In such a context, it has been demonstrated that a large part of the timbre of instruments was explained by their spectral envelope. The spectral envelope of a signal is a curve in the frequency-magnitude space that "envelopes" the peaks of its short-time spectrum (STS). [4].

This echoes a classic model for sound production, in which the observed data results from a source signal passing through a linear filter. (Figure2).

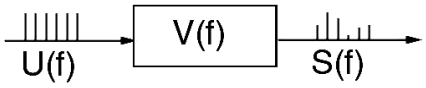

Figure 2: the source-filter model of sound production.

Although it has been designed to model speech production, it is also partially valid for musical instruments: in this case, the source signal is a periodic impulse drive responsible for the pitch, and the filter V(z) embodies the effect of -say- the resonating body of the instrument; namely, its timbre. If we identify the filter's spectral magnitude and the spectral envelope, it is clear that the envelope is independent of the pitch: the harmonic partials slide along this constant spectral "frame", which determines their amplitude.

Thus, in this simplified framework, an "evaluation" of the spectral envelope appears to be a good candidate for the front-end of our system, as it meets our 2 criteria: it is a pitch-invariant measure of timbre.

One issue raised by this paper is whether this monophonic model is portable to the polyphonic, multi-instrument context that is of interest for us. A polyphonic texture can be represented in the former framework as a sum of source-filter paths. Using the same model to describe the whole process in a global way then means that we make the following approximation:

"We can find a composite pitch excitation U_C *and a composite meta-instrument envelope* S_C *such that:*

$$\sum_i S_i \times U_i = S_C \times U_C \qquad (1)$$

where S_i *and* U_i *are the filter responses and excitations of the individual instruments.",* which seems analytically awkward. Yet, looking at polyphonic spectrums like in figure 3 and 4, we see that this approach, approximate as it may be, may well be fruitful. Envelopes can be defined, and they seem to differ sufficiently from texture to texture.

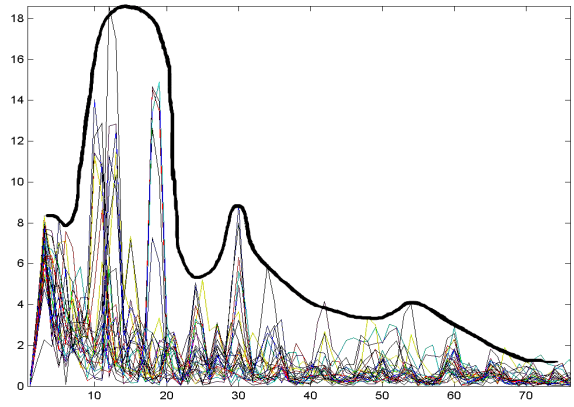

Figure 3: superposition of 50 successive STS (30 ms frames) for a polyphonic texture {guitar + bass + synthesizer}

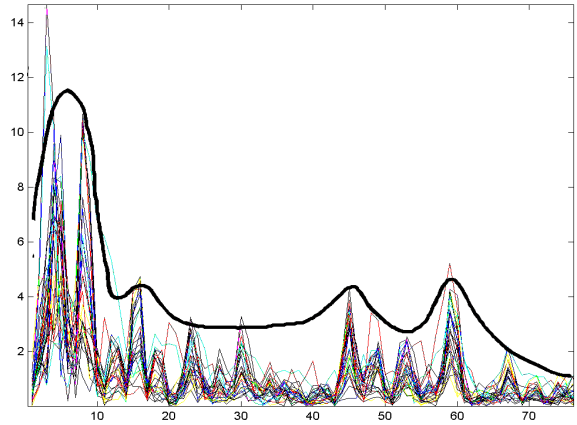

Figure 4: superposition of 50 successive STS (30ms frames) for a different polyphonic texture : {piano + bass + synthesizer}

To estimate the spectral envelope of the musical signal we analyze, we consider three different schemes:

2.3- Linear Prediction (LP)

This makes full use of the source-filter model as it identifies V(z) as an all-pole filter of order p.

$$V(z) = \frac{1}{1 + a(1) \cdot z^{-1} + ... + a(p) \cdot z^{-p}} \quad (2)$$

As described in [5], we first estimate p+1 autocorrelation values for each frame, and then derive the p filter coefficients from a set of linear equations. This involves inverting the autocorrelation matrix, which is done with the Levinson-Dublin algorithm for Toeplitz matrices. From the p coefficients, we then have access to the spectral envelope of the filter (Figure5).

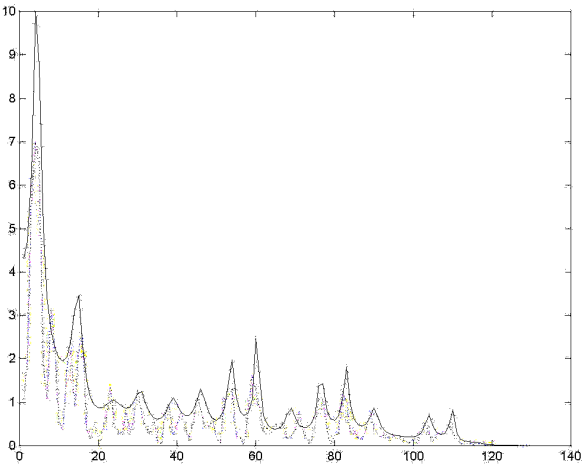

Figure 5: Spectral envelope estimated via LPC, order 15.

The linear prediction coefficients (LPC) a(i) control the position of the poles of the transfer function: this method is thus appropriate for modeling a signal with significant peaks in its Power Spectral Density (PSD). Since it assumes that the excitation is a white noise, the spiky aspect of the output's PSD is also seen on the filter's envelope estimate: it gets down very sharply in between the partials, rather than linking them smoothly.

Figure 6 compares the standard deviation of LPC on sample1 and sample2 plotted against the order of the LPC: the ratio between the two remains approximately constant with as the number of coefficients varies. Indeed, the pitch dependence comes from the "spiky" behavior, and this behavior doesn't depend on the order. By reducing the number of poles, we just reduce the number of spikes, which simply yields poorer estimates.

2.4- Mel Frequency Cepstrum (MFC)

The cepstrum is the inverse Fourier transform of the log-spectrum.

$$c_n = \frac{1}{2\pi} \times \int_{\omega=-\pi}^{\omega=+\pi} \log(S(e^{j\omega})) \cdot e^{j\omega \cdot n} d\omega \quad (3)$$

We call mel-cepstrum the cepstrum computed after a non-linear frequency warping onto the Mel frequency scale. The c_n are called MFC coefficients (MFCC). MFCCs are widely used for speech recognition, and Logan in [6] has shown that they were also justified for music analysis. Several instrument recognition systems also make use of MFCC [7], [8].

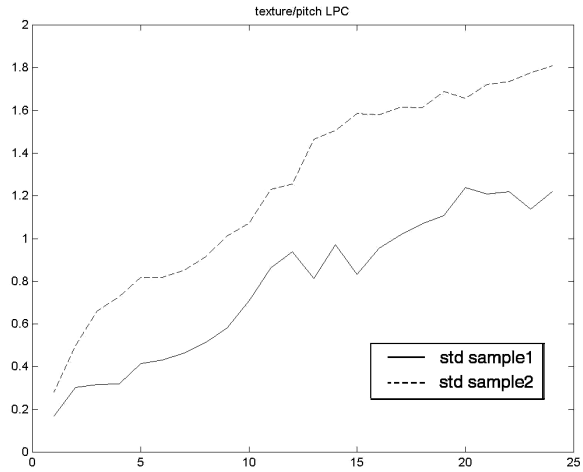

Figure 6: Comparison of the standard deviation of LPC coefficients on sample1 and sample2, plotted against the number of coefficients

If we consider the signal to be produced by the source-filter model, then taking the log-spectrum exhibits a deconvolution of the excitation and the envelope.

$$\log(V(z) \times U(z)) = \log(V(z)) + \log(U(z)) \quad (4)$$

Thus, the lower cepstral coefficients account for the slowly changing spectral shape, and the higher orders describe the fast variations of the excitation. So, to obtain an envelope measure that is independent of pitch, we should only use the first few coefficients. (Figure7). Note that the low order cepstrum can be roughly viewed as averaging the spectrum: it no longer has the property to envelope and link the peaks of the STS.

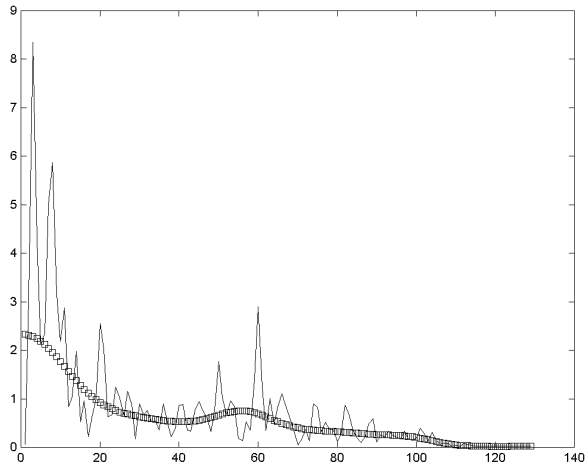

Figure 7: Spectral envelope estimated with MFCC, order 8.

Figure 8 compares the texture/pitch standard deviation for MFCCs: Logically, the low order cepstrum shows a good ratio, being rather independent of pitch. As the order increases, the "sharp" variations of the spectrum are increasingly taken into account, and the ratio progressively falls under 1. The limit point is around 10 coefficients.

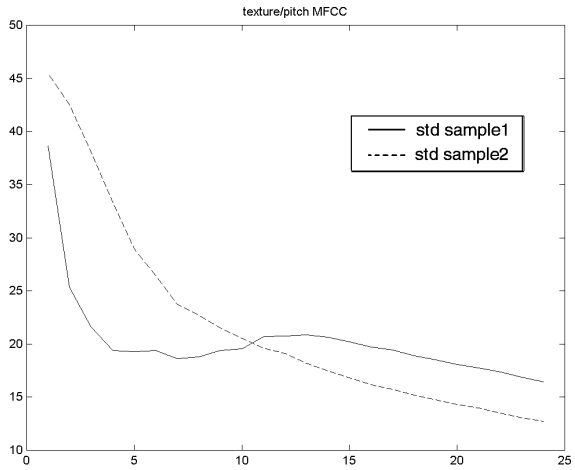

Figure 8: Comparison of the standard deviation of MFCC on sample1 and sample2, plotted against the number of coefficients

2.5- Discrete Cepstrum (DC)

Both previous methods are spectral estimation techniques, and we examined how they stood up for the problem of spectral *envelope* estimation. This third method, though, is specific to envelopes. It was introduced by Galas and Rodet in [9], who suggested to estimate the cepstral coefficients directly by interpolating the spectrum.

Since we want the envelope estimate to smoothly link the partials, this suggests that we already know the envelope estimate's value at the partials frequencies: it is the value S_k of the spectrum at these same frequencies f_k.

If we do a careful peak-picking on the spectrum to measure the frequencies of the partials, the (log-amplitude) envelope estimate A(f) can then be interpolated from the corresponding values of the spectrum. In practice, this is done via the minimization of the frequency-domain least square criterion:

$$\varepsilon = \sum_k \left\| 20 \cdot \log(s_k) - A(f_k) \right\| \qquad (5)$$

where the log-amplitude envelope estimate, $A(f), \forall f$ is parameterized in terms of p cepstral coefficients (if order is p):

$$A(f) = c_0 + 2 \cdot \sum_{i=1}^{p} c_i . \cos(2\pi f \cdot i) \qquad (6)$$

Cappe in [10] shows that this method suffers from ill-conditioning problems, and derives a modified criterion which constraints the envelope to be smooth. The resulting envelope estimate seems to be a good compromise between "good fit to the partials" and "pitch independence". (Figure9)

Figure 10 shows the texture/pitch standard deviation for Discrete Cepstrum Estimation. We notice the same behavior than with MFCCs (low-order/high order), only here the variances are globally very small. The estimate is so smooth than it levels the differences between the spectrums.

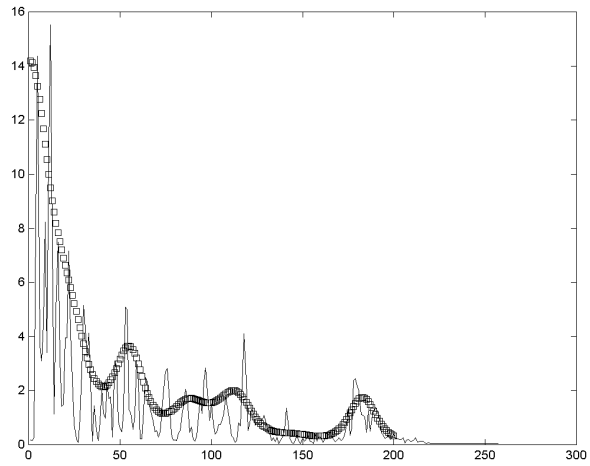

Figure 9: Spectral envelope estimated with Discrete Cepstrum, order 8.

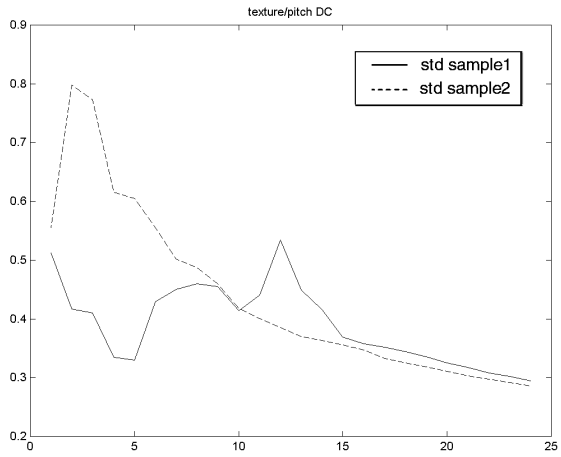

Figure 10: Comparison of the standard deviation of Discrete Cepstrum on sample1 and sample2, plotted against the number of coefficients.

2.6- Encoding

So far, we've shown how a vector of LPC, MFCC or discrete cepstrum coefficients can describe the spectral envelope of a STS. In an attempt to improve the texture/pitch ratio of our feature set, we investigate an alternative encoding of the estimated envelope:

For computational respects, directly storing a sampled representation of the continuous envelope estimate itself is intractable, or it would imply a drastic down sampling. We can however describe the continuous envelope estimate by computing its moments, as if it was a probability distribution: we get a new feature vector consisting of the curve's centroid (first order moment), standard deviation (second order moment), skewness (third order), kurtosis (fourth order), and so on.

We notice that the standard deviation values with the moment encoding are generally higher than with the coefficient encoding. Figure11 compares the texture/pitch standard deviation ratio for the three methods of estimation (MFC, LP, DC) with the moment encoding. The ratio for MFC and DC is <1, which translates a poor

discrimination power. For low order (<10), the "LP + moment" front-end has a ratio comparable with LP coefficients.

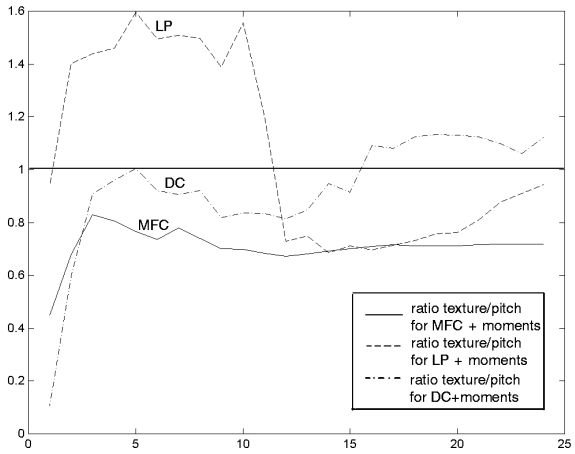

Figure 11: Comparison of the standard deviations ratio for all estimation methods, encoded with 10 moments, plotted against the order of the estimation.

2.7- A note on temporal features

Not all research on timbre makes use of spectral features. Martin, in [11], has built an automatic monophonic sound source recognition scheme using only temporal characterization calculated from a log-lag correlogram. However, the precise evaluations - and even definition - of parameters such as rise-time, decay, or vibrato, in the context of multi-instrument polyphonic data is not obvious. To account for the temporal dynamic of the data, we've considered appending delta coefficient to our feature set (i.e. first order difference between the spectral features of consecutive frames), but we didn't observe any improvement of the performance.

3- THE SEGMENTATION ALGORITHM

3.1- Principle

We divide the audio stream, sampled at 10KHz, into 50% overlapping, 30 ms frames, on which it is assumed to be stationary. For each frame, a feature vector is computed from the STS. The whole tune is thus turned into a trajectory of points in the high dimension feature space. We are then faced with an unsupervised learning problem: we want to infer a meaningful probability distribution over this space. Pattern recognition theory provides us with a common model to mimic the probability density of a sequence of observations: hidden Markov models (HMM).

It is useful to view a HMM as a generative model: it generates a sequence by following a first-order Markov chain of states, changing state once every time unit. Each time t that a new state is entered, an observation vector O_t is generated with some probability distribution (figure12). By specifying the transition probabilities between states and the observation generation model for each state, we can attempt to capture the underlying dynamics of a large set of observation vectors.

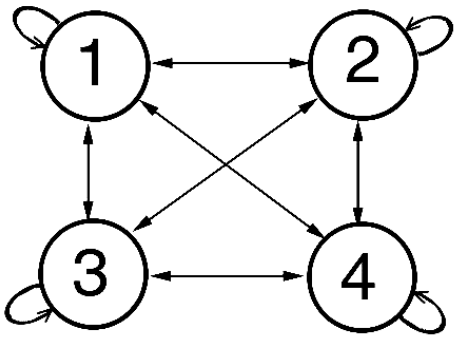

Figure 12: A 4-state ergodic HMM. Each state can generate a vector in the feature space according to some probability distribution

We model the song by a single N-state ergodic HMM. The observation generation process for each state is modeled by a mixture of Gaussian distributions over the feature space :

$$p_i(O_t) = p(O_t / S_i) = \sum_{m=1}^{M} c_{im} \cdot \mathrm{N}(O_t, \mu_{im}, \Gamma_{im}) \quad (7)$$

where O_t is the observation vector generated at time t by state S_i, N is a Gaussian pdf with mean μ_{im}, covariance matrix Γ_{im}, and c_{im} is a mixture coefficient.

We use the classic Baum-Welsh algorithm [12] to train the HMM on the sequence of feature vectors for the song. Through maximum likelihood, the states' output distributions and transition probabilities converge to best explain the succession of the features. Frames that present the same statistical properties are accounted for by the same state, which embodies the corresponding texture. We then use Viterbi decoding [12] to uncover the most probable hidden state sequence that has generated the observed succession of frames. Each frame of the input signal is thus associated with a state Si of the model, i.e. with a texture Ti, and we obtain the desired segmentation.

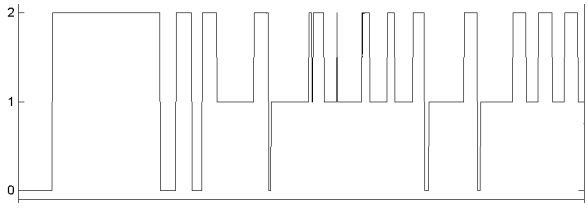

Figure 13: Segmentation of Bourvil's song. State 0 is {silence}, state 1 is {voice + accordion + accompaniment} and state 2 is {accordion + accompaniment}

Figure 13 shows the results of such an analysis on 20 seconds of music, a 60's French song by Bourvil, modeled by a 3-state ergodic HMM. Its instrumentation consists of a male singer accompanied by an accordion, and a discrete rhythmic section. We see that the segmentation is very acute (indeed, the theoretical precision is one 30ms frame): we notice the background accompaniment at the end of every sung phrase, sometimes even between the sung words. The accordion introduction appears very clearly, as well as the periodicities of the verse. This bird's eye view representation can be

used to analyze the long-term structure of the song, and gives information that is not always available on a musical score.

3.2 Comparison of the feature sets

In Section 2, we established 6 different sets of features (3 methods of estimation, 2 encoding). The segmentation algorithm was tested on 20 songs of various genres, from folk to rock, pop, blues and orchestral music.

The segmentation is evaluated by recovering the audio frames associated with each texture (which is easy through the indexing of the feature frames), and listening to them. The better the segmentation, the more coherent the different textures sound: following Bourvil's example, we ideally recover all the "sung voice" frames in one single file, all the "accordion" frames in another file, and lastly the initial frames of silence, without any "leakage" from one file to another.

	COEFFICIENTS	MOMENTS
MFC	Low order (8-10): ++ High Order (>10): - (non relevant sub-divisions of some textures according to pitch)	-- Irrelevant and noisy segmentation
LP	Low/High order(>8): + (more noisy than MFC)	Low/High order: +
DC	Numerical Problems due to very small variance -> ?	-- Irrelevant and noisy.

Figure 14: Comparison of the performance of the different front-ends.

The results are presented in figure 14.

It appears that the best feature set for our system is a vector of about 10 MFCCs. Adding more MFC coefficients doesn't improve the performance, as it causes unwanted pitch dependence, and fewer coefficients simply yield a poorer envelope estimate.

LPC also performs quite well, relatively independently of the number of coefficients, provided it is >8 (again, fewer coefficients yield a poorer envelope estimate).

Discrete Cepstrum Coefficients could not be tested on sufficient data. The small variance of the set nearly always causes numerical problems during Baum-Welsh re-estimation, as Gaussian distributions collapse on individual points, and log-likelihood grows to infinity. The moment representation solves this problem, but engenders very poor results. We thus would like to further investigate this DC coefficient front-end, which have shown promising results in the field of sound synthesis, and could still prove a seductive alternative to the popular MFCCs.

Our alternative moment encoding, although it has the nice property to increase the variance of the feature set, yields very poor results for both cepstrums, and is only comparable to the coefficient representation as regards linear prediction, without reducing the dimensionality of the data.

3.3 The topology issue.

One issue raised by Logan in [3] is the choice of the right number of states for the model: with classic Baum-Welsh re-estimation, the topology of the model is fixed a priori. We have shown that the previous tune was well modeled and segmented with a 3-state HMM. If we now train a HMM with more states, the resulting segmentation is sometimes less exploitable, as one texture will be shared by two or more states. Similarly, with an insufficient number of states, some states may account for several textures. Logan suggests that this can be improved by HMM structure learning techniques such as [13] or [14].

However, in such cases where the number of states is too high, we often observe a considerable oscillation amidst the states sharing the same textures. Since we work on very small frames, and since in music, textures often last for a few seconds, the diagonal of model's transition matrix is normally very dominant. If we notice, like in figure 15, that a couple (i, j) and (j, i) of non-diagonal terms is abnormally high, we may consider clustering the two corresponding states. This is done by building a new composite state with a mixture of the two clustered Gaussian distribution. The new model with N-1 states is then retrained by Baum-Welsh.
This crude technique partly solves the problem, but there is still a need for a better learning algorithm or a higher-level output layer to infer the right number of states.

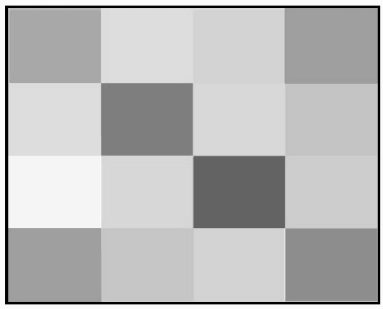

Figure 15: Transition Probability Matrix showing that states 1 and 4 are abnormally linked.

4. DISCUSSION OF THE NECESSITY OF THE HMM

The evaluation of our algorithm on different genres of music allows us to sketch the following classification, and to discuss the relevance of the HMM as a segmentation tool:

1) In the simplest case of music with elementary instrumentation, such as pop music, jazz trios, etc., we observe that, most of the time, the textures' distribution are very well separated in our feature space. Figure 16 shows an example of such a situation on a folk song with voice, guitar and harmonica. The timbres are very definite and separate from one another.

A very accurate fit is therefore not needed, and the HMM works ideally with a single full-covariance Gaussian distribution for each state.

Moreover, the temporal structure of the song brings very little information that is really necessary to the segmentation: actually, the textures are nearly "already" segmented. In the most extreme case - stripped-down folk music, say- we may even say that the "hidden variables" are not hidden anymore, but rather directly observable. The whole process is then nothing more than a static clustering in the feature space, and there is no need for a HMM. Logan in [3] shows that K-means clustering performs adequately on Beatles' songs to find repeated phrases. We established similar results for the segmentation of Delta blues and Bob Dylan tunes.

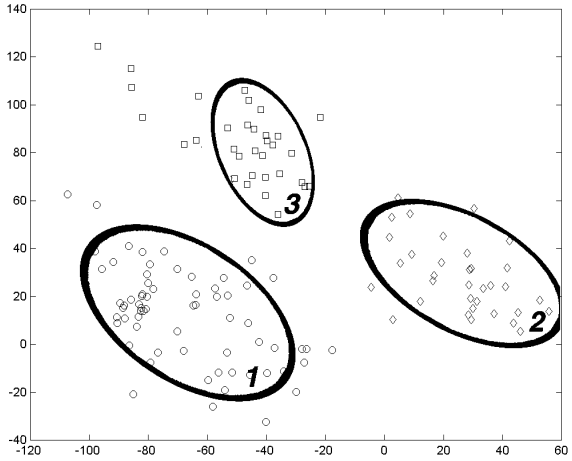

Figure 16: 2D Projection of the feature vectors of a folk tune, onto the first 2 principal components. Circles (1) represent frames of voice, diamonds (2) guitar and squares (3) harmonica.

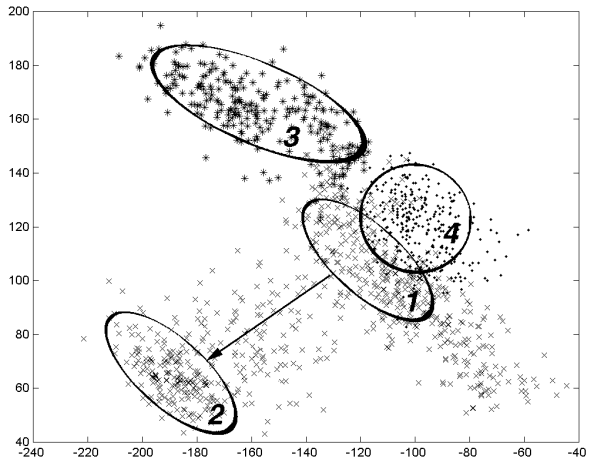

Figure 17: 2D projection of the features vectors of an extract of orchestral music. Crosses (1,2 and around) correspond to clarinet frames, stars (3) to violin and dots (4) to a large composite brass section.

2) In the intermediate case, such as orchestral music, the timbres are more intricate, and therefore all the approximations done for envelope estimation don't stand up as well as before:

- For a given texture, the envelope is not independent of pitch.
- For a given texture, even with constant pitch, the envelope does evolve with time.

The static discrimination of the textures is no longer accurate, and we have to take account of the dynamics of the data.

Figure 17 clearly demonstrates this. The analyzed extract is from an orchestral piece involving clarinet, violins and a large brass section. Collections of points number 1 and 2 correspond to clarinet frames separated by one octave (E3 and E4, respectively). They are more distant from one another than 1 and 4, which yet represent different textures. In this case, a static clustering wrongly associates 1 and 4 and separates 1 from 2. The algorithm really needs to keep rack of the dynamics of the data to see that 1 and 2 proceed of the same texture.

Similarly, Figure 17 proves the necessity of considering mixtures of Gaussians: if the clarinet is modeled by a single Gaussian distribution, it may get trapped in a local minima by accounting for only collection 1 or collection 2.

This interestingly echoes music cognition: in order to understand and segment acoustic phenomena as a succession of textures, the brain dynamically links a multitude of short events ("frames"), which, if considered out of their context, couldn't always be separated. This is another way of saying that temporal cues are as important as spectral cues in auditory events perception.

3) In the "worst" case of contemporary music, we may even reach the "anti-clustering" situation, where the quasi-uniform feature set has structure only by the virtue of its pattern through time. As an example, all of our front-ends have failed on segmenting pieces from Gyorgy Ligeti.

The segmentation becomes a very complex problem of unsupervised learning, as it has to account simultaneously for different scales. Just as we use a HMM to model the succession of different instrumental sections in music, we could model explicitly the dynamics of the spectral features that define a single timbre.

In fact, the whole process could probably be integrated into a larger architecture, where the different levels of analysis cooperate. We are currently investigating a possible framework for this: the Hierarchical HMM [15]. It generalizes the standard HMM by allowing hidden states to represent multi-level stochastic process themselves: each state/texture of a large HMM is itself a "smaller" HMM.

5. CONCLUSION

In this work, we describe an algorithm for music segmentation, which uses an unsupervised learning approach with a single hidden Markov model. To select a proper front-end to the system, we've implicitly done a number of approximations: we've applied a monophonic model to a polyphonic signal; we've also assumed that the spectral envelope was constant for a given texture and did not depend on pitch. Had these approximations stood up, the segmentation in our feature space would have been a simple static clustering problem. However, "real world" music is a complex dynamic process, and we've shown the necessity –as well as the difficulties- of a hidden structure model such as a HMM.

Note still that, in practice, K-means clustering yields very good results when the music is not too "polyphonic", which still encompasses a lot of commercial music. This is a powerful result, since the computational cost is very low compared to the HMM algorithm.

Future work will investigate a suitable implementation of Discrete Cepstrum, and a better assessment of the topology problem, possibly by using a multi-scale point of view such as [15].

6. ACKNOWLEDGMENTS.

We would like to thank the members of the Audio & Music Processing Group at King's College, London for useful discussions, and especially Dr Mike Davis, who provided invaluable insight into stochastic models. Jean-Julien Aucouturier would also like to thank people at the Ecole Superieure d'Electricite for making this whole project possible.

7. REFERENCES:

[1] Tzanetakis, G. & Cook, P., "Audio Information Retrieval (AIR) tools", in *Proc. Music IR 2000*

[2] Raphael, C., "Automatic Segmentation of Acoustic Musical Signals Using Hidden Markov Models", in *IEEE transactions on Pattern Analysis and Machine Intelligence, vol.21, NO4, April 1999*

[3] Logan, B. & Chu, S., "Music summarization using key phrases" in *Proc.ICASSP, 2000.*

[4] Schwarz, D. & Rodet, X.,"Spectral Estimation and Representation for Sound Analysis-Synthesis". in *Proc. ICMC, 1999.*

[5] Rabiner, L.R. & Juang, B.H., "Fundamentals of speech recognition". *Prentice-Hall 1993*

[6] Logan, B., "Mel Frequency Cepstral Coefficients for music modeling", in *Proc. Music Information Retrieval 2000*.

[7] Eronen, A. & Klapuri A., "Musical Instrument Recognition using cepstral coefficients and temporal features", in *Proc. ICASSP 2000*

[8] Dubnov, S. & Rodet, X., "Timbre recognition with combined Stationary and Temporal Features".

[9] Galas,T. & Rodet, X., "An improved cepstral method for deconvolution of source-filter systems with discrete cepstra : application to musical sounds" in *Proc. of International Computer Music Conference, Glasgow.*

[10] Cappe,O and al, "Regularized Estimation of Cepstrum Envelope from Discrete Frequency Points", in *Proc. EUROSPEECH 95*

[11] Martin, K., "Towards Automatic Sound Source Recognition: Identifying Musical Instruments", in *Proc. NATO Computational Hearing Advanced Study Institute, Italy 1998*

[12] Rabiner, L., "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition", in *Proc. IEEE, vol. 77, no. 2, 1989*

[13] Brand, M. "Structure Learning in conditional probability models via an entropic prior and parameter extinction", *MERL Technical Reports, 97.*

[14] Wolfertstetter, F. & Ruske, G., "Structured Markov Models for speech recognition", in *Proc. ICASSP 1995, vol.1, 544-47*

[15] Fine,S. and al, "The Hierarchical Hidden Markov Model : Analysis and Applications", in *Machine Learning, 32(1), July 98.*