Structural Segmentation of Musical Audio by Constrained Clustering

ARTICLE *in* IEEE TRANSACTIONS ON AUDIO SPEECH AND LANGUAGE PROCESSING · MARCH 2008

Impact Factor: 2.48 · DOI: 10.1109/TASL.2007.910781 · Source: IEEE Xplore

2 AUTHORS:

Mark Levy
Mendeley Ltd.
13 PUBLICATIONS **284** CITATIONS
SEE PROFILE

Mark Brian Sandler
Queen Mary, University of London
294 PUBLICATIONS **3,082** CITATIONS
SEE PROFILE

Structural Segmentation of Musical Audio by Constrained Clustering

Mark Levy, *Student Member, IEEE,* Mark Sandler, *Senior Member, IEEE*

***Abstract*— We describe a method of segmenting musical audio into structural sections based on a hierarchical labelling of spectral features. Frames of audio are first labelled as belonging to one of a number of discrete states using a Hidden Markov Model trained on the features. Histograms of neighbouring frames are then clustered into segment-types representing distinct distributions of states, using a clustering algorithm in which temporal continuity is expressed as a set of constraints modelled by a Hidden Markov Random Field. We give experimental results which show that in many cases the resulting segmentations correspond well to conventional notions of musical form. We show further how the constrained clustering approach can easily be extended to include prior musical knowledge, input from other machine approaches, or semi-supervision.**

***Index Terms*— audio, segmentation, music, clustering.**

I. Introduction

THIS paper describes an approach to extracting the overall structure of a piece of music directly from an audio recording. Knowledge of this structure has various useful practical applications, for example: music summarisation, such as the automatic selection of short representative audio 'thumbnails' for use in browsing large music collections; automatic section-by-section alignment of audio tracks to aid retrieval algorithms; the development of features for use in audio editing or synchronization to video, such as 'jump to start of next section', 'double-click to select current phrase', etc. It also has more general significance for the continued development of techniques in music information retrieval, in particular to support the goal of learning the semantics of musical audio signals: many relevant semantic concepts ('lively', 'serene', 'vocal', etc.) apply more naturally to sections of a piece rather than to entire tracks.

A. Background

The task under consideration here is that of finding high-level structure in musical audio. The kind of structure we have in mind is close to what musicians and musicologists refer to as 'musical form'. Form is, in principle, a description of the particular structure exhibited by a given piece of music, but in practice common musical forms become established by imitation, and we refer to pieces as being 'written in' a particular form because most pieces composed in a given genre conform more or less to one of a handful of widely-recognised structural patterns. This is particularly true for conservative genres of vocal music, such as most popular music (rock, pop, country, reggae, Hip-Hop, etc.), where the lyrics also tend to follow established or straightforward verse forms. Perhaps the best-known form, common to many different musical genres, is Verse-Chorus form, which frequently consists of the following sequence of sections: intro-verse-chorus-verse-chorus-bridge-verse-chorus-outro, where the verse sections set the verse of the lyrics and the chorus its refrain, and the bridge is a contrasting, sometimes instrumental, section. In general the tempo and metre of a Verse-Chorus song are constant, and each section has a fixed length, often of eight or sixteen bars. Classic examples include The Beatles' 'All you need is love' and 'Penny Lane', The Beach Boys' 'California girls', or Jimi Hendrix's 'Foxey lady'.

It is possible to approach automatic structural segmentation by treating observations of this kind as strong assumptions about the music under consideration, specifically by assuming that it will correspond exactly to one of a small number of canonical forms. In this case the segmentation task can be reduced to beat-tracking to establish tempo and metre, followed either by brute-force matching of audio features to possible structural templates [1] or by a search across certain elements of structure constrained by strong heuristics [2], [3]. In general, however, musicians, producers and composers take a playful approach when choosing a structure for their compositions, and while structures usually conform loosely to standard forms, they frequently diverge from their canonical patterns. Of the Beatles 210 recorded songs, for example, only 23 begin with the intro-verse-chorus-verse-chorus sequence, some of those in fact dispense with the intro, and they continue in several different ways [4]. Similarly while sections in most music are four, eight or sixteen bars long, choruses amongst the Beatles' songs range from two to nineteen bars, including most of the possible intervening lengths [5]. Although there are many artists whose music is more predictable than that of the Beatles, relying on strong assumptions about the structure being sought is likely to be unsafe in general. In the test set of 60 songs described in section V, for example, only one (Sinead O'Connor's cover of 'Nothing compares with you') contains exactly the intro-verse-chorus-verse-chorus-break-verse-chorus-outro sequence of sections, and its section lengths are not regular.

Ideally we would like to base our estimation of high-level structure on the entire range of evidence available to a human listener: the progression of harmonies, the location

Manuscript received December 15, 2006. This work was supported by EPSRC grants GR/S84750/01 (Hierarchical Segmentation and Semantic Markup of Musical Signals) and EP/E017614/1 (Online Music Recognition And Searching).

M. Levy and M. Sandler are with the Centre for Digital Music, Department of Electronic Engineering, Queen Mary, University of London, London E1 4NS, U.K. (e-mail: mark.levy@elec.qmul.ac.uk; mark.sandler@elec.qmul.ac.uk)

and nature of melodic cadences, changes of instrumentation and production effects between sections, the presence of telltale drum fills, the relationship of music to lyrics, etc. In practice, many of these features are not yet readily available for automatic analysis. Existing methods therefore rely either on local strong changes of timbre to indicate possible section boundaries [6], [7], or repetition of sequences of features to indicate the reocurrence of sections of a particular type [1], [2], [3], [8], or a combination of the two [9], [10]. Such methods formalise partial expectations about the overall form of the music: we expect, for example, that in most songs the chorus (if there is one) will contrast in timbre with the verse, but that it will be based on the same melody and sequence of harmonies each time it occurs.

B. Approach

We work with an even weaker and more general model of musical structure, but one which tries to take into account its intrinsically hierarchical nature: in virtually all music, notes are grouped into beats, beats into bars, bars into phrases, phrases into sections, etc. We assume that we are able to label each beat of the music (frame of audio) as belonging to a particular state, such that all beats in the same state sound similar in some respect. For the sake of concreteness each possible state might, for example, correspond to a particular chord. We require only that the states quantise the full extent of some feature space observed in the track under consideration. We assume further that each segment-type (verse, chorus, etc.) is characterised by a particular distribution of states, because each segment of a particular type contains roughly similar music. We estimate the local distribution of states at each point in the track by counting states within a small histogram window. The characteristic state distributions, and the segment-type assignments for each beat expressing the overall segmentation, can then be found by clustering the histograms with a suitable algorithm: each cluster corresponds to a particular segment-type. The clustering algorithm is parameterised only to enforce our expectation that segments will rarely be shorter than some given duration. We do not currently attempt to attach semantic descriptions ('verse', 'chorus', etc.) automatically to the segment-types.

All existing methods for structural segmentation, including our own, are more likely to succeed on recordings made with modern production techniques, in particular when copy and paste has been used to clone multiple segments of the same type, and when transitions between segments of different types are marked by heavy block changes in instrumentation and production effects. Our approach has the advantage, however, that it can still find structure even if there is no repetition at segment-level (i.e. the piece is what musicians call 'through-composed'), or if there are no evident 'hard' boundaries between segments. In addition, even when we fail to recover a segmentation similar to a human listener's assessment of the form of the music, our method has the merit that, by design, output segments are guaranteed to contain audio that is consistent in the feature (strictly feature-model) space, as discussed further in Section III. This is a virtuous property if the purpose of doing the structural segmentation is as a preparatory step before feature modelling, for example if we intend to learn or classify semantic properties of the track in question.

The organisation of the remainder of this paper is as follows: Section II shows how we achieve the low-level labelling in which each beat of the music is assigned to a particular state; Section III presents evidence that, in real musical audio, segment-types are reasonably well characterised by state distribution, given a suitable choice of feature-type and labelling algorithm; Section IV introduces the constrained clustering algorithm which we use to segment the state sequence; Section V describes our experiments in segmenting a varied set of audio tracks, and introduces suitable evaluation measures; Section VI gives experimental results; Section VII discusses related work, and Section VIII gives our conclusions.

II. Low-level state labelling

Our method of low-level state labelling is based on the AudioSpectrumEnvelope, AudioSpectrumProjection and SoundModel descriptors of the MPEG-7 standard [11]. The underlying audio feature is the AudioSpectrumEnvelope, a power spectrum with the frequency domain divided into logarithmically spaced sub-bands between 62.5Hz and 16kHz, with two additional bands for the power below 62.5Hz and above 16kHz, with values converted to a decibel scale. The use of logarithmic scaling is intended to imitate approximately the response of the human ear. We extract AudioSpectrumEnvelope features with bands at $\frac{1}{8}$th-octave spacing from audio mixed to mono and downsampled to 11025Hz. We use a hop size equal to the beat-length of the music (typically 300-400ms), as estimated by a beat-tracking algorithm [12], and a window of three times the hop. We choose a hop of one beat as a natural resolution for structural segmentation, although this is not essential for our method and we can use a default resolution if beat-tracking fails. The spectrum for each window is normalised by its $L2$-norm, representing the overall power, and the dimensionality of the features is reduced by applying Principal Component Analysis to the entire sequence of features over the track, retaining the first 20 principal components. The norm of each spectrum is appended, itself normalised by its largest value over the entire track to give values in the range $[0, 1]$. This yields 21-dimensional AudioSpectrumProjection feature vectors, in which the first 20 dimensions represent the spectral shape, and the final dimension the relative power, in each window.

We train a Hidden Markov Model (HMM) with a fairly large number of states on the entire sequence of AudioSpectrumProjection feature vectors for the track, with a single Gaussian output distribution for each state, and a single covariance matrix tied across all states. We then Viterbi-decode the features using the trained model, to give the most likely sequence of state assignments for each beat of the music. Fig. 1 shows the resulting sequences of beat-level labels for two tracks, using a 40-state HMM.

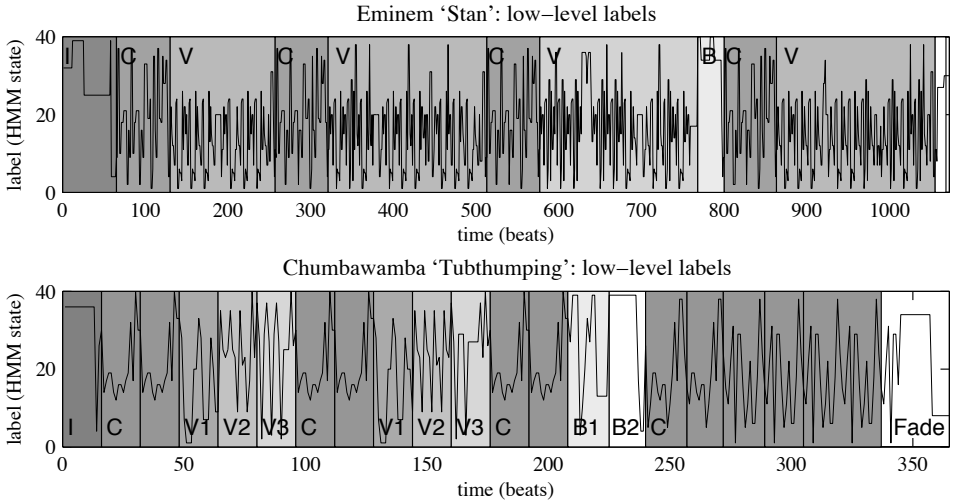

Fig. 1. Sequence of low-level labels against manual segmentation for two tracks. Note how segments of the same type (verses, choruses, breaks, etc., shown in same background shade and by initial letter) contain similar sequences of states.

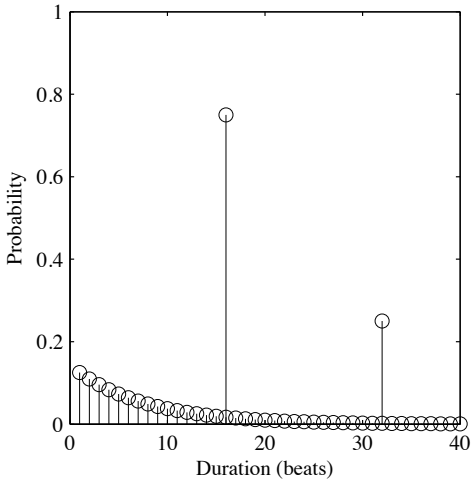

Fig. 3. Geometric duration distribution of an HMM state vs segment duration distribution for a typical chorus, which lasts for 4 or sometimes 8 bars.

III. Segment-type state distributions

As Fig. 1 illustrates, segments of the same type frequently contain similar, though not identical, sequences of low-level labels. It is tempting to speculate that by reducing the number of states we may be able to use an HMM to label segment-types directly, as proposed (though without any evaluation of the resulting segmentations) in [13], [14]. State sequences from an HMM with five states are shown in Fig. 2, illustrating the weakness of this approach: segment-types are clearly not well modelled by individual HMM states. The reason for this is that the geometric distribution on state-occupancy implicitly defined by the HMM is a very poor model for segment lengths in real music, as shown in Fig. 3.

We can demonstrate directly, however, that manual annotations of the form of real music can be expressed as sets of characteristic distributions of low-level states. We suppose that for a particular track we have a human segmentation $\{y_i^*\}$ into K segment-types, where $y_i^* \in \{1, ..., K\}$ is the segment-type label for the i-th beat. We extract a corresponding sequence of low-level states using an M-state HMM, and estimate the local state distributions $\{x_i\}$ at each beat of the sequence by counting neighbouring states within a small histogram window. We find the characteristic reference distributions for each segment-type $\{m_k^*\}$ by counting states over the relevant beats $\{i : y_i^* = k\}$. We can then evaluate how well the manual segmentation $\{y_i^*\}$ is expressed by the reference distributions $\{m_k^*\}$ by treating them as cluster centroids in the space of local distributions $\{x_i\}$. We assign segment-type labels to each beat $\{\tilde{y}_i\}$ according to the closest reference distribution

$$\tilde{y}_i = \text{argmin}_k \text{d}(x_i, m_k^*) \quad (1)$$

where $\text{d}(.,.)$ is a suitable distance measure. The human segmentation will have been well expressed by the reference distributions if the $\{\tilde{y}_i\}$ reconstruct the $\{y_i^*\}$ well.

We ran this evaluation over a test set of 60 varied tracks containing 650 manually-annotated segments (see Section V for details), using Euclidean distance as the distortion measure in (1), with various different histogram lengths $h = 3, 7, 13$ and numbers of HMM states $M = 40, 80$. We show mean pairwise f-values (a standard metric for quality of clustering, described in detail in Section V) in Table 1. The results are best with $M = 80$, with the size of the histogram window making little difference. Typically the overall stucture of $\{y_i^*\}$ is well preserved in $\{\tilde{y}_i\}$, as illustrated in Fig. 4, with $\tilde{y}_i = y_i^*$ on average for over 90% of the frames in each segment over the test set. The top panel shows the segment-type labels $\{y_i^*\}$ according to the human reference segmentation, and the second panel the labels $\{\tilde{y}_i\}$ from (1), based on the distribution of HMM states found in each segment type. The third and fourth panels for comparison show segment-type labels assigned by simple clustering algorithms applied to histograms of HMM states.

We conclude that it is reasonable to express structural segmentations of music as a set of reference distributions of low-level states, so that each segment-type corresponds to a cluster in the space of state histograms. The results in

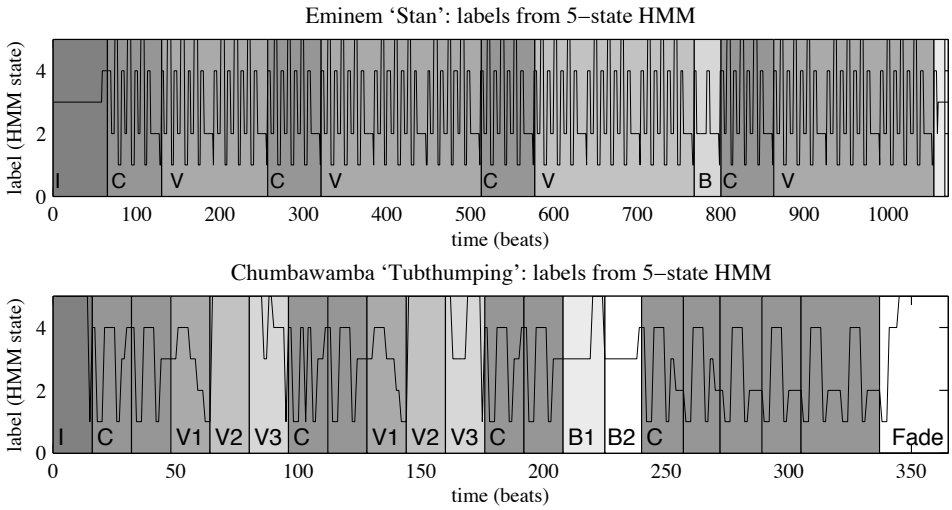

Fig. 2. Sequence of labels from a 5-state HMM against manual segmentation for two tracks. Note the lack of correspondence between HMM states and segment-types (shown as background shades and by initial letter).

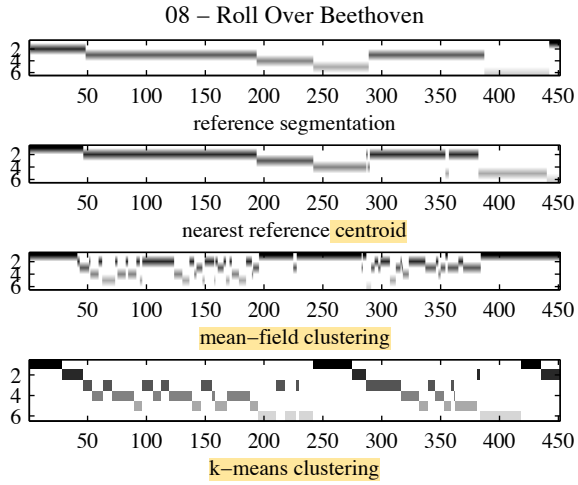

Fig. 4. Reference centroids vs simple clustering algorithms (clusters have been renumbered to aid comparison).

TABLE I
QUALITY OF SEGMENT-TYPE LABELLING USING REFERENCE DISTRIBUTIONS

		$M = 40$	$M = 80$
$h = 3$	pairwise f-value	48.1%	83.5%
$h = 7$	pairwise f-value	82.3%	88.4%
$h = 13$	pairwise f-value	85.3%	89.5%

Table 1 also serve as a rough upper bound on the quality of segmentation we can achieve by this histogram clustering approach. We have repeated this experiment using different features as input to our low-level labelling, in particular with 12-dimensional chroma (Pitch Class Profile) which has been reported as successful in other segmentation tasks [3], [8], [15], but the results are less good.

We observe that, using HMM states as low-level labels, the characteristic distributions $m_k = (m_{k1}, ..., m_{kM})$ found by clustering histograms will correspond to the component weights for Gaussian mixtures modelling the features z for each segment-type:

$$P(z|k) = \sum_{j=1}^{M} m_{kj} P(z|j) = \sum_{j=1}^{M} m_{kj} \mathcal{N}[z, \mu_j, \Sigma] \quad (2)$$

where μ_j is the mean of the j-th HMM state, and Σ is the tied covariance.

Although human structural segmentations can be well expressed as cluster centroids, it does not follow that the reference distributions can easily be found with any simple clustering algorithm. A clustering approach to segmentation was first proposed in [16], using algorithms designed specifically to cluster histograms due to [17], [18]. As Fig. 4 illustrates, these methods can do somewhat better than basic k-means clustering, but the clusters found still fail to respect the temporal continuity that we expect from segments.

IV. CONSTRAINED CLUSTERING

We represent temporal continuity as a set of 'must-link' constraints in a clustering setting. Formally we consider a set of observations $X = (x_1, ..., x_n)$ (our histograms of states), and a corresponding set of hidden labels $Y = (y_1, ..., y_n)$ (the segment-type for each histogram). Each hidden variable y_i encodes the cluster label of the point x_i and takes its value from a set of cluster indices $S = (1, ..., K)$ (the possible segment-types). A hidden set of generative model parameters consists of cluster prototypes $\Theta = (m_1, ..., m_k)$. Observations are constrained to have come from the same cluster according to a set of observable 'must-link' constraints $C = (c_{12}, c_{13}, ..., c_{n-1n})$ where $c_{ij} = 1$ indicates that x_i and x_j belong to the same cluster, while $c_{ij} = 0$ indicates that the pair (x_i, x_j) is unconstrained.

We define the set of neighbours N_i of y_i to be the points to which x_i is must-linked: $N_i = \{y_j : c_{ij} = 1\}$. This defines a Markov Random Field over the hidden variables Y:

$$P(y_i|y_{S\setminus i}) = P(y_i|N_i) \quad (3)$$

According to the Hammersley-Clifford theorem [19], the prior probability of a particular sequence of labels Y can be expressed as a Gibbs distribution

$$P(Y) = \frac{1}{Z}\exp(-v(Y)) = \frac{1}{Z}\exp\left(-\sum_{N_i \in N} v_{N_i}(Y)\right) \quad (4)$$

where Z is a normalising term (the 'partition function') over the set of all possible label configurations, N is the set of all neighbourhoods, and where the overall potential function $v(Y)$ can be decomposed into a sum of potential functions for each neighbourhood $v_{N_i}(Y)$. Because the potentials are based on the pairwise constraints C, we can further decompose $P(Y)$ as

$$P(Y) = \frac{1}{Z}\exp\left(-\sum_{i,j}\phi(i,j)c_{ij}(1-\delta_{y_i,y_j})\right) \quad (5)$$

where δ is the Kronecker delta and $\phi(i,j)$ represents a penalty incurred if the constraint c_{ij} is violated.

To enforce temporal continuity on cluster assignments, we create constraints between temporally neighbouring labels according to a neighbourhood size d_{ML}, i.e. we set

$$c_{ij} = \begin{cases} 1 & \text{if } 1 \leq |i-j| \leq d_{ML} \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

We assume that observations are generated using the model parameters Θ based on the labels Y, but independent of the constraints. Each observation x_i is assumed to be dependent only on its corresponding label y_i and distributed according to

$$P(x_i|y_i = k, \Theta) = \frac{1}{Z_\Theta}\exp\left(-\frac{1}{\sigma^2}\mathrm{d}(x_i, m_k)\right) \quad (7)$$

where $\mathrm{d}(.,.)$ is a Jensen-Shannon divergence to capture similarity between histograms, σ represents the lengthscale of the clusters, and Z_Θ is a normalising term.

We define the constraint penalty function simply as a fixed cost

$$\phi(i,j) = \lambda \quad (8)$$

For clustering by Expectation-Maximisation we require the posterior distribution for each individual label $p(y_i = k|X, \Theta)$. Direct calculation of this is not computationally feasible and we are therefore required to use an estimation method such as Iterated Conditional Modes (ICM) or a mean-field approximation [20], [21].

A. Expectation-Maximisation with constraints

1) Initialisation: We initialise the cluster parameters $\theta^{(0)} = \{m_k\}$ by clustering the observations with conventional k-means, setting m_k to the centroid of the kth cluster. We note the maximum-likelihood sequence of labels according to the centroids (i.e. neglecting the constraints) $\hat{y}_i$.

2) E-step: Given the current parameters $\theta^{(t)}$, we use ICM to update the sequence of labels $\hat{y}_i$. We choose a random ordering of the labels and update them sequentially, so that each label individually has maximum likelihood given the current values of the others

$$\hat{y}_i = \mathrm{argmax}_k[q_i(k)] \quad (9)$$

where

$$\begin{aligned} q_i(k) &= p(y_i = k|\hat{y}_{S\setminus i}, X, \theta^{(t)}) \\ &\propto p(x_i|y_i = k, \theta^{(t)})p(y_i = k|\hat{y}_{S\setminus i}) \end{aligned} \quad (10)$$

by Bayes' theorem and the conditional independence of the x_i. Inserting (5), (7) and (8) into (10)

$$q_i(k) = \frac{1}{Z}\exp\left(-\frac{1}{\sigma^2}d(x_i, m_k) - \sum_{j\neq i}\lambda c_{ij}(1-\delta_{k,\hat{y}_j})\right) \quad (11)$$

where Z is a normaliser such that $\sum_k q_i(k) = 1$. We repeat in a new random order until the labels $\hat{y}_i$ do not change. Under ICM, q converges rapidly to a local maximum of $P(Y|X, \Theta)$ [22].

3) M-step: We make the standard update to the parameters using the $q_i(k)$ found in the E-step

$$m_k^{t+1} = \frac{\sum_i q_i(k)x_i}{\sum_i q_i(k)} \quad (12)$$

B. Extensions to the clustering framework

It is straightforward to extend this clustering framework to include pairwise 'cannot-link' constraints, allowing us to incorporate information about sections of the track which should be labelled as belonging to different segment-types, for example to take advantage of candidate boundaries supplied by a suitable detection function, or directly by the user in a semi-supervised context.

V. Experiments

We tested our segmentation method on a set of 60 tracks of rock, pop, Hip-Hop and jazz from the past five decades. Half the tracks are by the Beatles, while the others are from a variety of well-known artists from the 1980s to the present day, including Björk, Britney Spears, Eminem, Madonna, Michael Jackson, Nirvana, Prince and The Clash. A full list, together with the reference structural segmentations used for evaluation, is available online[1]. In all cases our reference segmentations attempt to capture the formal structure described in Section I-A. Tracks from three of the Beatles' albums were chosen because their music is unusually well-studied: our reference segmentations are based on Alan Pollack's authoritative formal analyses [23]. The reference segmentations for 14 of the remaining tracks are expert annotations originally prepared for the MPEG-7 working group, while the remainder were prepared by the first author, who is a trained musicologist and an experienced professional musician.

[1] http://www.elec.qmul.ac.uk/digitalmusic/downloads

For each track we computed a sequence of low-level labels using the method described in Section II with an 80-state HMM, and then histogrammed the sequence over a sliding window of length 7 beats. We clustered the histograms to give structural segmentations for each track by three methods: simple k-means, the mean-field histogram clustering algorithm which was the best-performing method in [16], and the constrained clustering algorithm described in Section IV-A. In each case we set the number of clusters to $K = 6$, reflecting the typical number of segment-types in the reference segmentations, and for constrained clustering we set the neighbourhood size to $d_{ML} = 16$, reflecting a standard musical phrase-length of 16 beats.

A. Evaluation metrics for segmentation

A number of non-standard evaluation methods have been proposed for music segmentation, reflecting, firstly, the difficulty of jointly evaluating the accuracy of segment boundaries and the quality of the segment-type labels, and, secondly, a desire to allow a range of segmentations of quite different granularities to be recorded as equally good. The metrics found in the literature include Mutual Information between sequences of segment-type labels for each frame [16], a segment-wise Hamming distance [16], and string edit distance between 'normalised' sequences of labels for each segment [8], [9]. We prefer to use standard measures that can be summarised over a test set (unlike MI or Hamming distance, whose possible values are data-dependent) and give values in relation to a relatively objective formal groundtruth (unlike the 'normalised' edit distance which gives equally good scores to plausible segmentations on any timescale - even our sequence of low-level labels scores well with this metric).

We evaluate segment labels with pairwise f-value, one of the standard metrics for quality of clustering. This compares pairs of beats which are labelled with the same segment-type (i.e. assigned to the same cluster) in the machine output with those in the reference segmentation. Let P_m be the set of similarly-labelled pairs of beats in a track according to the machine, and P_h be the set of similarly-labelled pairs in the human reference segmentation. Then

$$\text{pairwise precision} = \frac{|P_m \cap P_h|}{|P_m|}$$

$$\text{pairwise recall} = \frac{|P_m \cap P_h|}{|P_h|}$$

$$\text{pairwise f} - \text{value} = \frac{2 * \text{precision} * \text{recall}}{\text{precision} + \text{recall}}$$

We evaluate segment boundaries separately, regarding an output boundary as correctly retrieving a reference boundary if it falls within some fixed small distance away from it, where each reference boundary[2] can be retrieved by at most one output boundary: we can then measure retrieval performance with standard precision, recall and f-value.

[2]Some of our reference segmentations annotate segments of double length as repeated segments of the same type. We ignore the resulting 'internal' boundaries in our evaluation.

TABLE II
SEGMENT-TYPE LABELLING PERFORMANCE.

		Recent	Beatles	Overall
k-means	pairwise f-value	45.7%	42.5%	44.1%
mean-field	pairwise f-value	56.0%	53.8%	54.9%
constrained	pairwise f-value	60.5%	60.4%	60.3%
reference	pairwise f-value	86.0%	90.7%	88.4%

TABLE III
BOUNDARY RETRIEVAL PERFORMANCE.

		Recent	Beatles	Overall
k-means	precision	34.1%	28.2%	31.1%
	recall	84.7%	77.0%	80.9%
	f-value	46.9%	40.5%	43.7%
mean-field	precision	41.1%	32.0%	36.6%
	recall	63.8%	69.1%	66.5%
	f-value	47.1%	42.4%	44.8%
constrained	precision	73.6%	56.0%	64.8%
	recall	59.3%	54.1%	56.7%
	f-value	64.0%	54.0%	59.0%

VI. RESULTS

We summarise the performance of the various clustering methods in Tables II and III, and in Figs. 5 and 6 we show example output for a few of the tracks which were relatively well segmented by constrained clustering. The boundary retrieval evaluation shown is based on a threshold of 3s, for comparison with the boundary-finding method reported in [24]. The results shown for constrained clustering were obtained with a cluster lengthscale of $\sigma = 1$ and an experimentally-determined constraint penalty $\lambda = 0.02$, although we find that the precise values are not significant.

A. Discussion

The results in Table III show an increase of over 30% in the overall f-value for boundary retrieval compared to the mean-field histogram clustering algorithm which was the best-performing method in [16]. The higher recall values for mean-field clustering merely indicate that a larger number of boundaries, mostly spurious, are being output. Although short of state of the art performance for boundary-finding alone, this result is encouraging in a method that does no explicit boundary-finding.

Figs. 5 and 6 compare the segmentations found by constrained clustering with our reference segmentations, and with segmentations produced by simple k-means and by mean-field histogram clustering of HMM state histograms. As the figures illustrate, the use of constraints can be very effective in enforcing realistic expectations for segment lengths, as controlled by the neighbourhood size d_{ML} in (6). We set $d_{ML} = 16$, on the basis that 16 beats, i.e. 4 bars of four-time, is reasonable as a minimum expected segment length. The exact value is unlikely to be significant, as can be seen from the good results achieved for Björk's cover of 'It's oh so

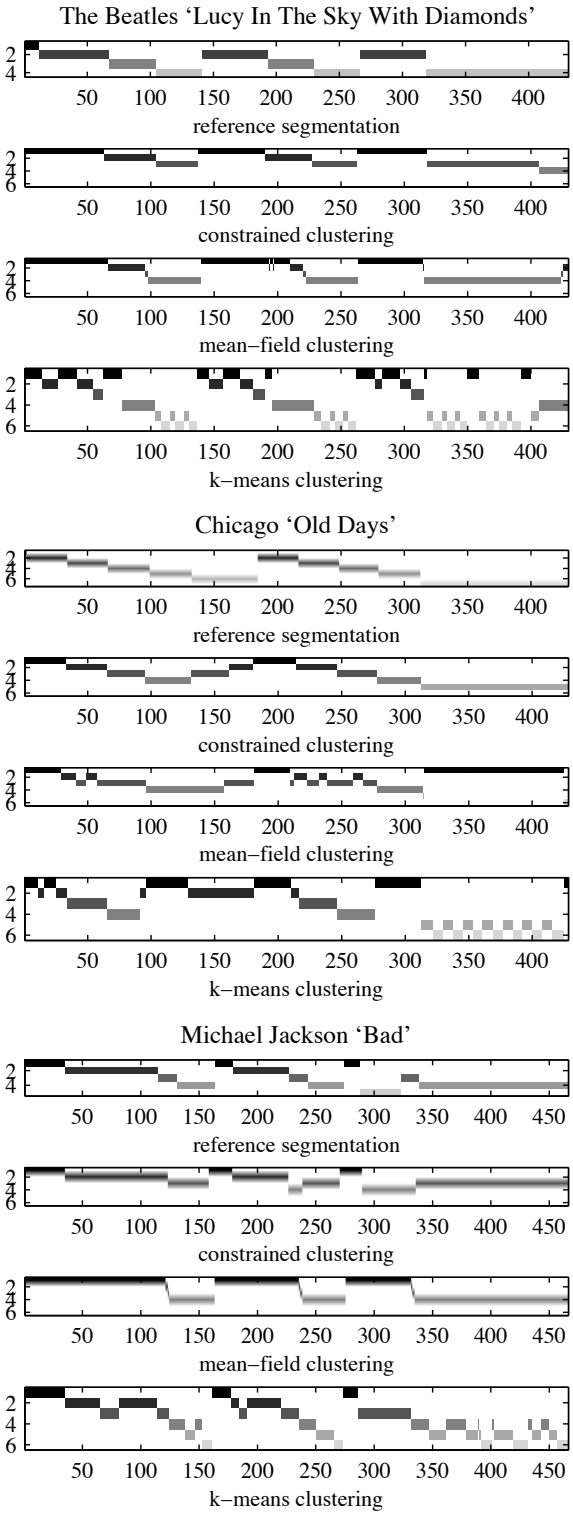

Fig. 5. Machine vs reference segmentations (clusters have been renumbered to aid comparison).

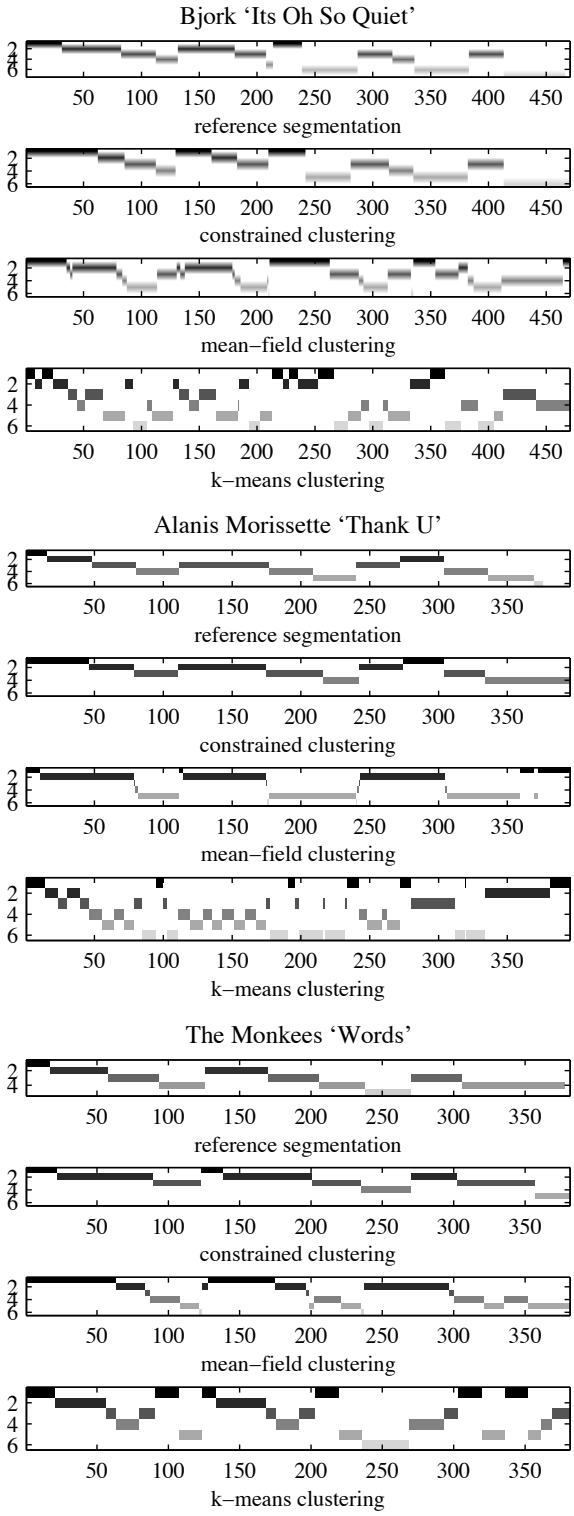

Fig. 6. More machine vs reference segmentations (clusters have been renumbered to aid comparison).

TABLE IV
EFFECT OF NEIGHBOURHOOD SIZE ON CONSTRAINED CLUSTERING PERFORMANCE.

		Recent	Beatles	Overall
$d_{ML} = 8$	label pairwise f-value	56.7%	58.8%	54.7%
	boundary f-value	66.2%	58.1%	62.2%
$d_{ML} = 16$	label pairwise f-value	60.5%	60.4%	60.3%
	boundary f-value	64.0%	54.0%	59.0%

quiet' shown in Fig. 6, despite the fact that this piece contains two quite different alternating tempi, one of which is in triple-time. Table IV gives comparative results over the test set with $d_{ML} = 8$ beats. The small improvement in boundary retrieval f-value, largely due to more short segments - and hence more boundaries - being produced, is less significant than the fall in labelling performance. A small value for d_{ML} can nonetheless produce good segmentations when there genuinely are many short sections in the form, as illustrated in Fig. 7. There might therefore be some advantage in using the method we developed in [25] to estimate an underlying base phrase-length for the track in question, and using this to calculate an appropriate value for d_{ML}, although this remains for future work.

The other parameters of the constrained clustering algorithm, the cluster lengthscale σ and the constraint penalty λ, were loosely optimised over our data set, as were the number of HMM states and the length of the histogram window, as discussed in Section III. Although we found that output segmentations were only mildly sensitive to variation in these parameters, different values might perform better on different collections of music, or given a different choice of underlying audio feature.

It is instructive to try to identify why the constrained clustering approach performs less well on some tracks. A few pieces in our test set contain a good deal of musical variation, for example changes in the overall instrumentation, in segments of the same formal type, for example between the verses of a simple Jazz ballad like Norah Jones's 'Lonestar'. This tends to lead to over-segmentation in the output, even when the reference centroids still resolve the segment-types well. In many other songs boundaries are misplaced, or entire extra segments are created, when there is a long vocal pickup before the next formal boundary. Finally in some cases, often the earlier, more 'acoustic' tracks, there is very little repeated structure evident in the HMM state sequence, despite obvious repetition in the music. In all these cases the poor performance is probably due to the timbral nature of the features we use: in future work we hope to reduce these problems with the use of well-tuned chroma features.

VII. RELATED WORK

Most previous research has been based on analysis of a self-similarity matrix comparing all pairs of frames for a given track [26], and some recent results using this approach appear promising as a way of finding structural boundaries, or of identifying a single 'most-repeated' segment (typically the chorus of a song). In [24] a novelty measure using kernel correlation along the diagonal of a similarity matrix based on MFCCs and other standard spectral features achieved boundary retrieval within a threshold of 3s with an f-value of 75%, over a set of 54 Beatles tracks. Excellent results for finding choruses were reported in [3] on a collection of 100 J-pop songs, using a search for diagonal line segments in a similarity matrix based on chroma features. Our clustering approach also gives good results for chorus-finding [27].

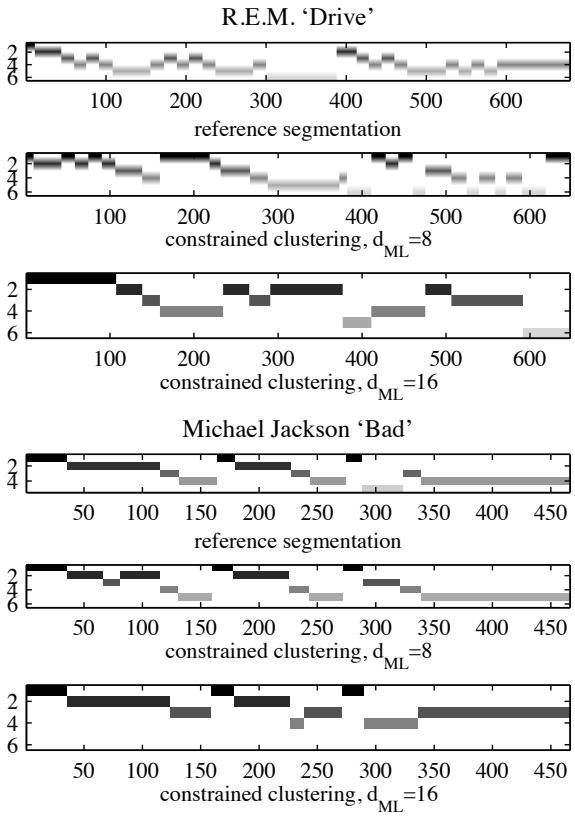

Fig. 7. Example segmentations varying the neighbourhood size for must-link constraints, d_{ML} (clusters have been renumbered to aid comparison).

Full-scale structural segmentation, however, remains a difficult problem. The modest results reported in [28], [9] suggest that it is not straightforward to extend the search for repeated sections in a similarity matrix to a full segmentation, unless very strong limiting assumptions are made about the form of the music in question, as in [1], [2]. Our work builds on the low-level HMM state labelling of [11], and the histogram clustering introduced in [16], using constraints to enable clusters to model segments effectively. We have also explored an alternative approach using explicit discrete segment duration distributions, based on statistics of segments in human reference segmentations, in a Hidden semi-Markov Model [25]. A Bayesian framework with a variety of parameterised segment-length distributions was introduced in [29].

VIII. CONCLUSIONS

This paper attempts to extract high-level musical structure directly from audio, using a very general approach to extend the frame-level labelling of existing MPEG-7 descriptors to the level of musical form. This leads to a clustering framework in

which musical and other prior information can be modelled as simple constraints. Our experiments over a varied test set of popular music demonstrate that, in principle, a clustering approach can capture musical structure effectively as a set of cluster centroids, where each centroid summarises a Gaussian Mixture Model in the feature space corresponding to a particular segment-type (verse, chorus, etc.). We have introduced a constrained clustering algorithm which improves labelling performance by 10% and boundary retrieval by 30% over clustering methods without constraints. There is still much room for improvement in performance, however, and a natural focus for future work is to incorporate explicit boundary detection as a set of cannot-link constraints within the same clustering framework. Last but not least, we hope to encourage further progress in this area by making our reference segmentations available, and by pointing to straightforward and standard evaluation metrics that allow ready comparison between different methods.

ACKNOWLEDGMENT

The authors would like to thank Samer Abdallah, Michael Casey and Christophe Rhodes for their many helpful insights, Matthew Davies for the use of his beat-tracking code, and Katy Noland for her help in creating reference segmentations.

REFERENCES

[1] N. Maddage, X. Changsheng, M. Kankanhalli, and X. Shao, "Content-based music structure analysis with applications to music semantics understanding," in *6th ACM SIGMM International Workshop on Multimedia Information Retrieval*, October 2004.

[2] L. Lu, M. Wang, and H. Zhang, "Repeating pattern discovery and structure analysis from acoustic music data," in *6th ACM SIGMM International Workshop on Multimedia Information Retrieval*, October 2004.

[3] M. Goto, "A chorus-section detecting method for musical audio signals," in *Proc. ICASSP*, vol. V, 2003, pp. 437–440.

[4] K. G. Johansson, "The harmonic language of the beatles," *Swedish Musicological Society STM-Online*, vol. 2, 1999.

[5] A. F. Moore, *Rock: The Primary Text*. Open University Press, 1993.

[6] J. Foote, "Automatic audio segmentation using a measure of audio novelty," in *Proceedings of IEEE International Conference on Multimedia and Expo*, vol. I, 2000, pp. 452–455.

[7] O. Gillet, S. Essid, and G. Richard, "On the correlation of automatic audio and video segmentations of music videos," *IEEE Trans. Circuits Syst. Video Techn.*, vol. 17, no. 3, pp. 347–355, 2007.

[8] W. Chai and B. Vercoe, "Music thumbnailing via structural analysis," in *Proc. ACM Multimedia*, 2003, pp. 223–226.

[9] J. Paulus and A. Klapuri, "Music structure analysis by finding repeated parts," in *Proc. of the 1st ACM Audio and Music Computing Multimedia Workshop*, 2006.

[10] Y. Shiu, H. Jeong, and C. J. Kuo, "Similarity matrix processing for music structure analysis," in *Proc. of the 1st ACM Audio and Music Computing Multimedia Workshop*, 2006.

[11] M. Casey, "General sound classification and similarity in mpeg-7," *Organised Sound*, vol. 6, no. 2, p. 153..164, 2001.

[12] M. E. P. Davies and M. D. Plumbley, "Beat tracking with a two state model," in *Proc. ICASSP*, 2005.

[13] J.-J. Aucouturier and M. Sandler, "Segmentation of musical signals using hidden markov models," in *Proc. AES 110th Convention*, 2001.

[14] G. Peeters, "Deriving musical structures from signal analysis for music audio summary generation: "sequence" and "state" approach," in *CMMR 2003 (LNCS2771) Lecture Notes in Computer Science*. Springer-Verlag, 2004, pp. 142–165.

[15] M. Bartsch and G. Wakefield, "Audio thumbnailing of popular music using chroma-based representations," *IEEE Trans. on Multimedia*, vol. 7, no. 1, 2005.

[16] S. Abdallah, K. Noland, M. Sandler, M. Casey, and C. Rhodes, "Theory and evaluation of a bayesian music structure extractor," in *Proc. ISMIR*, 2005.

[17] T. Hofmann and J. M. Buhmann, "Pairwise data clustering by deterministic annealing," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, 1997.

[18] J. Puzicha, T. Hofmann, and J. M. Buhmann, "Histogram clustering for unsupervised image segmentation," *Proceedings of CVPR '99*, 1999.

[19] J. M. Hammersley and P. Clifford, "Markov fields on finite graphs and lattices," 1971.

[20] S. Basu, M. Bilenko, and R. J. Mooney, "A probabilistic framework for semi-supervised clustering," in *Proc. ACM KDD*, 2004.

[21] T. Lange, M. H. C. Law, A. K. Jain, and J. M. Buhmann, "Learning with constrained and unlabelled data," in *Proc. IEEE CVPR*, 2005.

[22] J. Besag, "On the statistical analysis of dirty pictures," *Journal of the Royal Statistical Society, Series B*, vol. 48, pp. 25–37, 1986.

[23] A. Pollack, "Notes on... [the complete beatles recordings]." [Online]. Available: `http://www.icce.rug.nl/ soundscapes`

[24] B. Ong and P. Herrera, "Semantic segmentation of music audio contents," in *Proc. International Computer Music Conference*, 2005.

[25] M. Levy and M. Sandler, "New methods in structural segmentation of musical audio," in *Proc. European Signal Processing Conference*, 2006.

[26] J. Foote, "Visualizing music and audio using self-similarity," in *ACM Multimedia (1)*, 1999, pp. 77–80.

[27] M. Levy, M. Sandler, and M. Casey, "Extraction of high-level musical structure from audio data and its application to thumbnail generation," in *Proc. ICASSP*, 2006.

[28] W. Chai, "Automated analysis of musical structure," Ph.D. dissertation, MIT, 2005.

[29] S. Abdallah, M. Sandler, C. Rhodes, and M. Casey, "Using duration models to reduce fragmentation in audio segmentation," *Machine Learning*, 2006.

Mark Levy (born 1963) is a Research Assistant in the Centre for Digital Music at Queen Mary, University of London. He studied mathematics and music at Cambridge University, followed by musicology at King's College London, and more recently computer science at Birkbeck, University of London. His current research interests include automatic description of musical audio, mapping the space of musical emotion through data mining social tags, and developing practical software systems for music recommendation and playlist generation. His SoundBite playlist generation plugin for iTunes was released earlier this year by the Centre for Digital Music.

Before arriving at Queen Mary, Mark worked as a commercial software developer and was a guest Lecturer in music at Southampton University. He is also well known as a professional performer on the viola da gamba, having made numerous recordings for most of the major labels and given concerts throughout Europe, and he can often be heard on BBC radio and television, and on the soundtracks of movies including The Governess, A Knight's Tale and Titus.

Mark Sandler (born 1955) is Professor of Signal Processing at Queen Mary, University of London, and Director of the Centre for Digital Music. Mark received the BSc and PhD degrees from University of Essex, UK, in 1978 and 1984, respectively.

Mark has published over 300 papers in journals and conferences. He is a Senior Member of IEEE, a Fellow of IEE and a Fellow of the Audio Engineering Society. He is a two-times recipient of the IEE A.H.Reeves Premium Prize.