# MUSIC STRUCTURAL SEGMENTATION BY COMBINING HARMONIC AND TIMBRAL INFORMATION

**Ruofeng Chen**
Georgia Tech Center for Music Technology
Georgia Institute of Technology
ruofengchen@gatech.edu

**Ming Li**
Institute of Acoustics
Chinese Academy of Sciences
liming@mail.ioa.ac.cn

## ABSTRACT

We propose a novel model for music structural segmentation aiming at combining harmonic and timbral information. We use two-level clustering with splitting initialization and random turbulence to produce segment labels using chroma and MFCC separately as feature. We construct a score matrix to combine segment labels from both aspects. Finally Non-negative Matrix Factorization and Maximum Likelihood are applied to extract the final segment labels. By comparing sparseness, our method is capable of automatically determining the number of segment types in a given song. The pairwise F-measure of our algorithm can reach 0.63 without rules of music knowledge, running on 180 Beatles songs. We show our model can be easily associated with more sophisticated structural segmentation algorithms and extended to probabilistic models.

## 1. INTRODUCTION

Identifying music structural segmentation is one of the most important and difficult problems in music information retrieval (MIR). Its goal is to automatically locate the musically repetitive parts within a piece of music (e.g. verse, bridge and chorus in popular music). It has applications such as music thumbnail, segment-based editing and segment-based navigation. It may also facilitate other MIR tasks like beat tracking and chord detection.

There are some noteworthy existing systems, which inspire our proposed model. Foote [1] proposed self-similarity matrix for structure representation. Levy et al [2] proposed a two-level model for structural segmentation problem. In the lower level, they introduced Hidden Markov Models (HMMs) to quantize audio feature vectors into discrete states; in the upper level, they formed histograms by counting the HMM states in local windows and designed a clustering algorithm to quantize histogram vectors into segment labels. Weiss at el [3] showed the potential of Non-negative Matrix Factorization (NMF) in the structural segmentation problem. Notably, they make use of sparseness constraint to automatically determine the number of segment types in a song. Kaiser et al [4] exploited NMF on self-similarity matrix and clustering to differentiate segment types.

Undoubtedly, music structure is perceived based on many sources of information, among which harmony and timbre are primary players. Some existing systems use multiple features as starting points, listed in [5], but few found a good model to combine them. As is shown in [4], combining harmonic and timbral information works even worse than using timbral information alone. In this paper, we focus on building a model to combine the two sources to reach higher segmentation performance.

Our model is comprised of two parts. The first part is a two-level clustering algorithm, which produces segment labels using either harmonic or timbral information. The second part is a novel algorithm to bring segment labels from two different aspects together into a score matrix, and exploiting NMF to extract segment labels and sparseness to automatically determine the number of segment types in a given song. We call the score matrix and NMF based algorithm SM-NMF for short.

In Section 2 we describe our two-level clustering algorithm. In Section 3 we describe the SM-NMF algorithm. In Section 4 we present our experimental results and explain for them. In Section 5 we introduce possible extension of our model in future research. For convenience, we define the following symbols that will be used in the paper. $n$: number of frames in a song, each corresponds to a state label. $m$: dimension of feature vectors. $N$: number of windows in a song, each corresponds to a segment label. $k$: number of segment types.

This work was performed while interning at Institute of Acoustics, Chinese Academy of Sciences.

## 2. TWO-LEVEL CLUSTERING

Our two-level clustering algorithm is shown in Figure 1. From the feature extraction module we get frame-based feature vectors used for the lower-level feature clustering module, which quantizes feature vectors into states. The histogram module counts states in windows and forms histogram vectors used for the higher-level histogram clustering module, which quantizes histogram vectors into segment labels. This algorithm is similar to [2], except that we substitute HMMs with another clustering and no constraint is imposed.

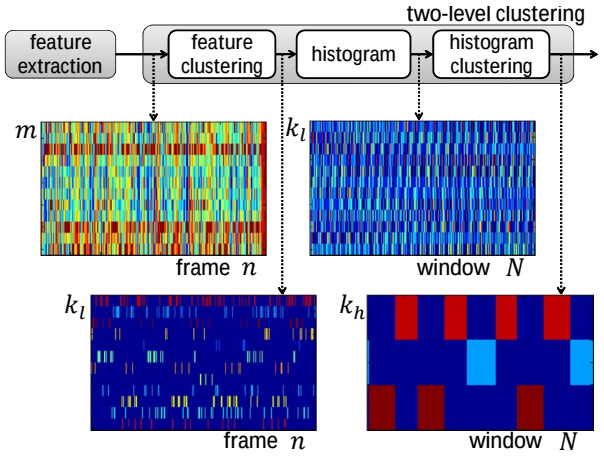


**Figure 1**. The flowchart and illustration of intermediate results of two-level clustering. The lower two graphs' colors only illustrate different labels for better looking.

### 2.1 Feature Extraction

We extract two types of vector features separately from audio files. Chroma is a 12-dimension representation indicating the power within each of the 12 pitch classes. So chroma has a close relationship with the harmonic characteristics of music. See [6] for algorithm of extracting chroma. Mel-frequency Cepstrum Coefficients (MFCC) is usually a 13-dimension representation describing the spectral envelope. It is easy to calculate and potential to reveal timbral similarity in feature space. See [7] for algorithm of extracting MFCC.

We divide the whole song with fixed frame length of $L_f$ ms and hop size of $L_{fh}$ ms, then calculate feature for each frame.

### 2.2 Clustering Algorithm

Clustering is a process of gathering points in the feature space to a fixed number of clusters so that hopefully neighboring points would have the same cluster label. K-means is one of the most straightforward algorithms to perform clustering [8]. Firstly, a fixed number of $k$ cluster centers $\mu_1, \mu_2 \ldots \mu_k$ are initialized, often randomly. Then two steps alternate iteratively: a) assign each point $x_j$ to its closest cluster center; b) recalculate each cluster center, until the objective function

$$G(x, \mu) = \sum_{i=1}^{k} \sum_{x_j \in C_i} DistanceMeasure(x_j, \mu_i)$$

converges, where $C_i$ is the set of feature vectors assigned to the $i$th cluster. Note that k-means is the coordinate descent of $G(x, \mu)$ so only local minimum is guaranteed.

In our experiments, we find that using uniform distribution to randomly initialize cluster centers sometimes converges to unreasonable local minima, so we apply an "initial guess by splitting" method described in [9] instead. If the target number of clusters is not power of 2, we split the cluster with largest variance until we achieve the right number. We find that using this technique most unreasonable results are avoided.

We have described one level of clustering. Now we move to two-level clustering. Firstly, we perform clustering on either chroma or MFCC into $k_l$ clusters, using Euclidean distance as distance measure, to obtain a state label for each frame, which can be interpreted as harmonic unit or timbre unit. Then we slide a window with length of $L_w$ frames and hop size of $L_{wh}$ frames throughout the whole song, and count the occurrence of every state. Now we have an array of histogram vectors, which are further normalized to be probabilistic. We perform clustering on histogram vectors into $k_h$ clusters, using symmetric Kullback-Leiber (KL) divergence [10] as distance measure.

$$KL(P||Q) = \frac{1}{L_w} \sum_{i=1}^{k_l} P_i \log \frac{2P_i}{P_i + Q_i} + Q_i \log \frac{2Q_i}{P_i + Q_i}$$

Symmetric KL divergence describes how dissimilar $P$ and $Q$ are to the assumed actual distribution $(P + Q)/2$. The resulting labels indicate segment types.

To further reduce $G(x, \mu)$, we insert a random turbulence module between splitting initialization and two-step iteration, for both levels of clustering. To do this, we add a vector with tiny norm and random direction to each cluster center. Make sure the shifted centers satisfy probability constraints for KL divergence. Then we perform clustering for $T$ times to get $T$ slightly different solutions. We can pick out the solution with lowest $G(x, \mu)$.

In our experiments, we notice in most cases the solutions with lowest $G(x, \mu)$ do not necessarily correspond to good results (see Table 2 for results). Therefore, to further improve the performance, we have to keep all $T$ solutions for further analysis.

## 3. COMBINING HARMONIC AND TIMBRAL INFORMATION

In this section, we describe how to combine harmonic and timbral information, i.e. the two-level clustering results from chroma and MFCC, to produce better segmentation results. For convenience, we name the segment labels produced by chroma as chroma solution. Similarly we have MFCC solution. We name the segment labels produced by the SM-NMF algorithm described below as final solution.

To motivate our idea, we show the typical results from chroma and MFCC respectively in Figure 2. Although both features produce fair results (pairwise F-measure 0.61 and 0.62), they are from completely different perspectives. For example, the chroma solution fails to distinguish verse and verse(instrumental) because the underlying harmonic patterns are exactly the same, but is good at distinguishing verse and bridge because of different harmonic patterns; the MFCC solution separates verse(instrumental) successfully because the timbre in this segment is very different from others, but cannot distinguish verse and bridge for their similar timbres.

Therefore, we set up the following rule for combination: two windows should have identical segment labels only if the two windows are both harmonically and timbrally similar. However, we cannot simply mix a chroma solution and an MFCC solution because segments from two aspects usually do not have common boundaries. There will often be lots of fragments in the outcoming results. In order to obtain a result with the same level of detail as chroma or MFCC solution, we make use of all $T$ chroma solutions and $T$ MFCC solutions to smooth the boundaries. Now we describe how to bring all $2T$ solutions into one final solution.

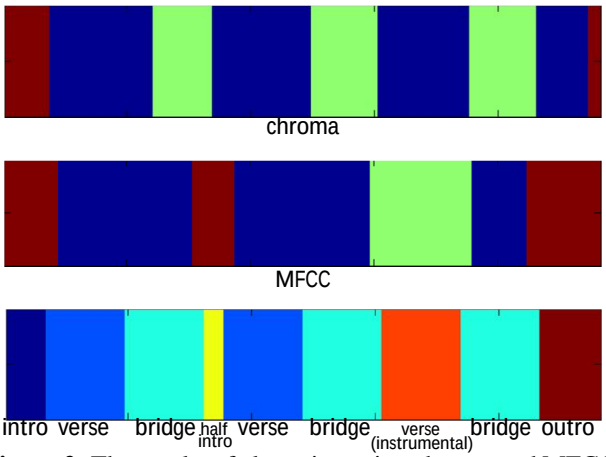


**Figure 2**. The results of clustering using chroma and MFCC respectively, along with the ground truth, of "In My Life".

### 3.1 Score Matrix

By analyzing $T$ different chroma or MFCC solutions from clustering with random turbulence, we find that typically some pairs of windows always have identical labels. These windows are lying within steady regions of a song. By contrast, some pairs occasionally have identical labels. Then either of them is lying within boundary regions (for example the short transition between segments with complicated instrumentation changes). Therefore, counting the times two windows having identical labels can reveal the steady regions and boundary regions in a song. We can construct a score matrix to describe how likely it is for two windows to have identical labels. This idea can be directly extended to a score matrix describing how likely it is for two windows to have identical labels in *both* chroma solution and MFCC solution.

To implement this, initialize an $N \times N$ matrix with all zeros. Perform two-level clustering using chroma and MFCC as feature separately, with splitting initialization and random turbulence, for $T$ times. Then investigate all the $T^2$ chroma-MFCC solution pairs: If the $i$th and $j$th windows in both chroma solution and MFCC solution have identical labels, the corresponding element in the score matrix increases by one. Finally, normalize all the elements by dividing by $T^2$.

The resulting score matrix serves the same purpose of visualizing music structure as Foote's self-similarity matrix, but the score matrix is much more well-structured and smooth. See Figure 3 for a graphical example.

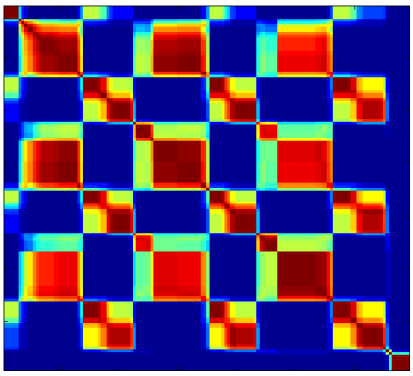

**Figure 3**. The score matrix of "Help!". The same song's self-similarity matrix is shown in [4].

### 3.2 Non-Negative Matrix Factorization

We can view the score matrix as an array of column vectors. Each vector corresponds to a window. Suppose we have a set of vector templates. Vectors in the steady regions of a song may be directly found in the set, while vectors in the boundary regions may be approximated by linear combination of vector templates. This observation pushes us to Non-negative Matrix Factorization (NMF) [11].

The $N \times N$ score matrix is approximately factorized into product of a $N \times k$ matrix $\mathbf{W}$ and a $k \times N$ matrix $\mathbf{H}$. The $j$th column of $\mathbf{W}$ can be viewed as the vector template for the $j$th segment type. The $j$th column of $\mathbf{H}$ describes the intensities of the $k$ segment types for the $j$th window. An example in shown in Figure 4.

We implement NMF using the multiplicative update rules [11]. Similar to clustering, NMF can only guarantee a local minimum of the sum of errors between the score matrix and $\mathbf{W} \times \mathbf{H}$. So we run NMF for several times with uniformly distributed random initialization and pick out the factorization result with lowest sum of errors.

After we obtain $H$, we apply Maximum Likelihood by assigning the segment label associated with the largest energy to each window. Note that in [4], clustering was used for the same purpose. In our experiment, we find that clustering and Maximum Likelihood produce almost the same performance. We choose Maximum Likelihood because it's simpler and more consistent.

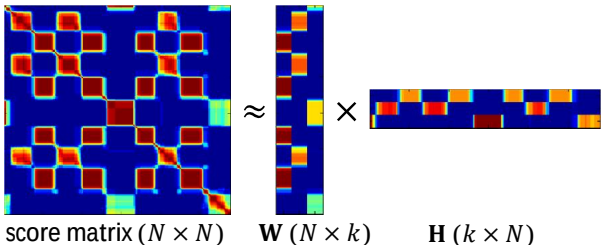


**Figure 4**. The score matrix is approximately factorized into the product of $W$ and $H$, from "Drive My Car".

### 3.3 Automatic Determination of the Number of Segment Types

Automatically determining of the number of segment types in a song is hard for two-level clustering, because clustering is a process of hard decision and all information about a window is its associated cluster label. However, using NMF, we have the matrix $\mathbf{H}$ whose columns involve intensities of all segment types. An example is shown in Figure 5. Intuitively one will agree $k = 3$ is the optimal number of segment types because the $\mathbf{H}$ with $k = 3$ is the most "resolute" one with least windows having much energy spread into multiple segment types. So we want a measure to quantify how much energy of a column is concentrated in as few components as possible. Sparseness [12] is a good measure which can satisfy the need.

$$sparseness(\mathbf{h}) = \frac{\sqrt{k} - (\sum |\mathbf{h_i}|)/\sqrt{\sum \mathbf{h_i}^2}}{\sqrt{k} - 1}$$

where $\mathbf{h}$ is a column of $\mathbf{H}$. The sparseness listed in Figure 5 is the average sparseness of all $N$ columns. We hope the columns of $\mathbf{H}$ to be as sparse as possible, so we factorize the score matrix with different $k$, then we pick out the $\mathbf{H}$ with largest average sparseness.

To summarize, we show the whole process of our model in Figure 6.

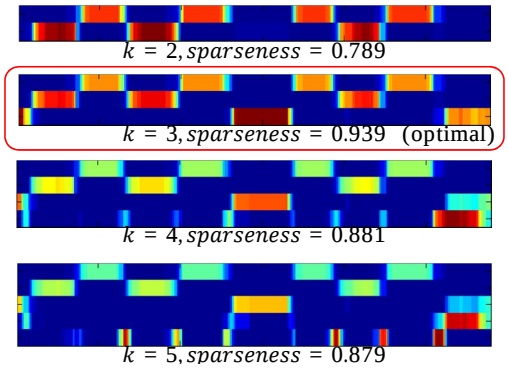


**Figure 5**. Obtaining $H$ with different $k$, we can use the result with largest average sparseness.

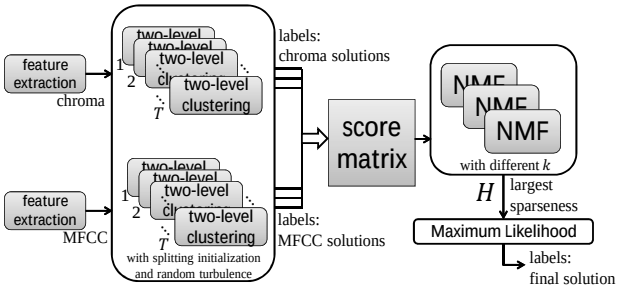


**Figure 6**. The complete flowchart of our proposed model. See Figure 1 for detail of two-level clustering.

## 4. EVALUATION

### 4.1 Parameters Configuration

We describe how to set up parameters (shown in Table 1) for two-level clustering. $L_f$ and $L_{fh}$ are set by assuming the audio signal is stationary for all frequency components in this short time duration. $k_l$ should be set a large number according to [2]. In our experiment, we see $k_l = 64$ works best. $L_w$ and $L_{wh}$ are not affecting the performance (pairwise F-measure) much, except that too small $L_w$ might make a very short segment longer than its actual length.

$k_h$ can be viewed as the number of types of harmonically similar segment or timbrally similar segment. $k_h = 3$ is a reasonable number, because a typical song has about 3 harmonic patterns (such as intro, verse and bridge) and also about 3 timbral patterns (such as intro, verse/bridge and instrument solo).

| | |
|---|---|
| frame length $L_f$ | 100 ms |
| frame hop size $L_{fh}$ | 50 ms |
| # of states $k_l$ | 64 |
| slide window length $L_w$ | 10 s |
| slide window hop size $L_{wh}$ | 1 s |
| # of segment types $k_h$ | 3 |
| # of loops $T$ | 7 |

**Table 1**. Parameters used in two-level clustering.

### 4.2 Overall Results

Our database comprises 180 Beatles' songs, consistent with the available ground truth annotations in Isophonics [1]. All songs are in the wav format of 16kHz/16bit/mono. We evaluate pairwise F-measure (PFM) [2] of our algorithms on the whole database referencing Isophonics annotations. Clustering processes with chroma and MFCC share the same set of parameters in Table 1.

Table 2 shows the PFM of two-level clustering with splitting initialization and without random turbulence, and the PFM of running two-level clustering with splitting initialization and random turbulence for $T$ times and minimization with regard to $G(x, \mu)$. $k$ cannot be automatically determined in clustering, so we fix $k = 3$, which can produce highest PFM in our experiments.

| | chroma | | MFCC | |
|---|---|---|---|---|
| random | no | yes | no | yes |
| PFM | 0.58 | 0.58 | 0.58 | 0.60 |

**Table 2**. PFM of two-level clustering with and without random turbulence.

We note that minimizing with regard to $G(x, \mu)$ can reduce $G(x, \mu)$ dramatically, but not necessarily improve PFM. So the relationship between PFM and $G(x, \mu)$ is not straightforward.

Then we evaluate our proposed SM-NMF algorithm. $k$ is automatically selected from $\{3, 4, 5\}$ according to the largest sparseness in the corresponding $\mathbf{H}$. We note that although many songs have more than 5 segment types according to annotations, such as the one shown in Figure 2, intro and half-intro are both harmonically and timbrally identical so it is impossible to discriminate them using only harmonic and timbral information. Therefore it is normal that the automatically determined $k$ is smaller than the actual number of segment types in annotations. In Table 3, besides SM-NMF, we also show the results using NMF with fixed $k$ for comparison. We see that SM-NMF produces better results than two-level clustering (Table 2) and sparseness is a good measure for the number of segment types.

| | fix $k$ | | | automatically |
|---|---|---|---|---|
| | $k = 3$ | $k = 4$ | $k = 5$ | determine $k$ |
| PFM | 0.62 | 0.62 | 0.61 | 0.63 |

**Table 3**. PFM of SM-NMF.

In Table 4 we show the results of a different way to form score matrix – by counting how many times two windows have identical labels using only one type of feature. The results indicate it is combining harmonic and timbral information that actually makes the main contribution to the performance of SM-NMF.

| | only chroma | only MFCC | both |
|---|---|---|---|
| PFM | 0.59 | 0.61 | 0.63 |

**Table 4**. Forming score matrix with either harmonic or timbral information versus both information.

Finally, in Table 5, we compare our results with other state-of-the-art methods, which use the same Isophonics annotation, listed in [3]. To be more informative, we also list pairwise precision rate (PPR) and pairwise recall rate (PRR).

| System | PFM | PPR | PRR |
|---|---|---|---|
| Mauch et al [13] | 0.66 | 0.61 | 0.77 |
| SM-NMF | 0.63 | 0.61 | 0.69 |
| Weiss et al [3] | 0.60 | 0.58 | 0.68 |
| Levy et al [2] | 0.54 | 0.58 | 0.53 |

**Table 5**. Segmentation performance of SM-NMF and other state-of-the-art methods on the Beatles data set.

Our algorithm does not involve any post-processes based on music knowledge such as eliminating too short segments or restricting segment length to multiples of 4 beats [13]. These rules can help reduce fragments, so we can expect our algorithm to produce higher PRR, and thus higher PFM, if we consider them.

### 4.3 Case Study

We study an example shown in Figure 7. In the chroma solution, we see that a verse is oversegmented into three segments (blue, red, green). We see in the score matrix that the red-labeled segment is tolerated in larger boxes but the green-labeled segment is not. This is because the red-labeled segment is a correctable mistake produced by some unstable clustering results, while the green-labeled segment is an uncorrectable mistake produced by the interference from heavy drumming. In the MFCC solution, we see that the first and second verse are given different label from the third and fourth verse. This is produced by the differences in background choir, by which MFCC solution is confident that they have two distinct timbres. So we see in the score matrix the upper left four large boxes are completely separated from the lower right four large boxes. The final solution will hide all correctable mistakes but display all uncorrectable mistakes. Therefore, SM-NMF performs well when the front-end structural segmentation algorithm (two-level clustering for this paper) makes as few uncorrectable mistakes as possible.

[1] www.isophonics.net

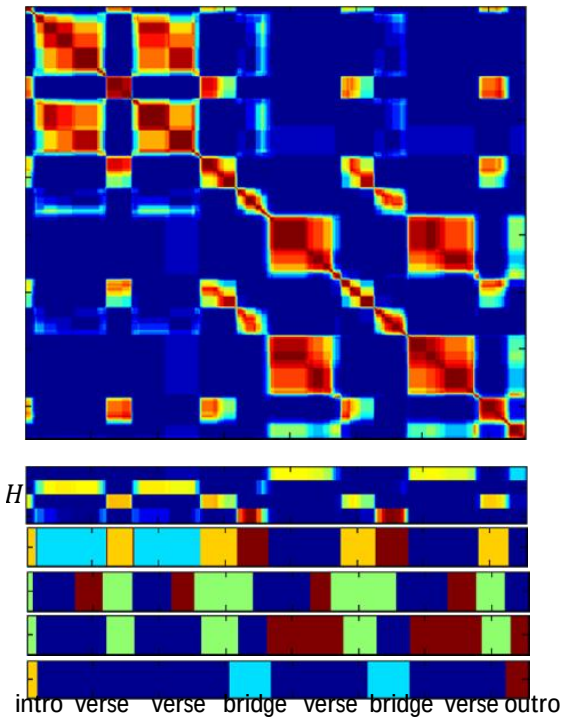


**Figure 7**. Example:"You Won't See Me". The last four labels are respectively final solution, chroma solution, MFCC solution and ground truth.

## 5. SUMMARY AND FUTURE WORKS

We have described a novel model for music structural segmentation, to bring the results of two-level clustering using chroma and MFCC separately into one final solution, aiming at combining harmonic and timbral information. We use splitting initialization and random turbulence to produce slightly different chroma and MFCC solutions from two-level clustering. Then we construct a score matrix to exhibit the pairwise relation between chroma solutions and MFCC solutions. We apply NMF and Maximum Likelihood to reveal music structure and sparseness to automatically determine the number of segment types in a given song. The PFM of our proposed SM-NMF method outperforms two-level clustering using single feature.

There is lots of space for improvement. We have shown in Section 4.3 that one obstacle in SM-NMF method is the *reliability* of solutions of the front-end algorithm. The two-level clustering can be replaced by any structural segmentation algorithm as long as random turbulence is included to produce slightly different solutions. We note that the *reliability* is not equivalent to the value of PFM, because for example we cannot expect MFCC alone to identify harmonically different segments or discriminate intro and half-intro. We need ground truth directly related to harmonically similar segments or timbrally similar segments.

Besides, NMF might produce better results with some constraints exploiting symmetry and sparsity. The score matrix is a flexible representation, which might be associated with probabilistic models. For example, if we view the score matrix as a "term frequency-inverse document frequency (tf-idf)" matrix, we might make use of Probabilistic Latent Semantic Analysis [14] to give a more elegant algorithm. We might also introduce constraints such as segment length and inter-segment transition probabilities to produce more musically meaningful results.

## 6. REFERENCES

[1] J. Foote: "Visualizing Music and Audio using Self-Similarity", *ACM Multimedia*, pp. 77–80, 1999.

[2] M. Levy, M. Sandler: "Structural Segmentation of Musical Audio by Constrained Clustering", *IEEE Transactions on Audio, Speech, and Language Processing*, 16(2):318326, 2008.

[3] R.J.Weiss, J.P.Bello: "Identifying Repeated Patterns in Music Using Sparse Convolutive Non-Negative Matrix Factorization", *ISMIR*, Utrecht, Netherlands, 2010.

[4] F.Kaiser, T.Sikora: "Music Structure Discovery in Popular Music Using Non-Negative Matrix Factorization", *ISMIR*, Utrecht, Netherlands, 2010.

[5] J.Paulus, M.Muller, A.Klapuri: "Audio-Based Music Structure Analysis", *ISMIR*, Utrecht, Netherlands, 2010.

[6] M. Goto: "A Chorus-Section Detecting Method for Musical Audio Signals", *In Proc. ICASSP*,V-437-440, 2003.

[7] B. Logon: "Mel Frequency Cepstral Coefficients for Music Modeling", *ISMIR*, 2000.

[8] C.M. Bishop: "Pattern Recognition and Machine Learning", *Springer*, 2006.

[9] Y. Linde, A. Buzo, R.M. Gray: "An Algorithm for Vector Quantizer Design", *IEEE Transactions on Communications*, Vol.Com-28, No.1, January, 1980.

[10] S. Abdallah, M. Sandler, C. Rhodes, M. Casey: "Using Duration Models to Reduce Fragmentation in Audio Segmentation", *Mach Learn*, 65:485-515, 2006.

[11] D. D. Lee, H. S. Seung: "Algorithms for Non-negative Matrix Factorization", *Advances in Neural Information Processing Systems*, 2001.

[12] P. O. Hoyer: "Non-negative Matrix Factorization with Sparseness Constraints", *Journal of Machine Learning Research 5*, 1457-1469, 2004.

[13] M. Mauch, K. C. Noland, and S. Dixon: "Using Musical Structure to Enhance Automatic Chord Transcription", *Proc. ISMIR*, pages 231236, 2009.

[14] T. Hofmann: "Probabilistic Latent Semantic Analysis", *Uncertainty in Artificial Intelligence*, 1999.