CLUSTERING MUSIC RECORDINGS BY THEIR KEYS

Yuxiang Liu[1,2] **Ye Wang**[2] **Arun Shenoy** **Wei-Ho Tsai**[3] **Lianhong Cai**[1]

[1] Department of Computer Science & Technology, Tsinghua University, Beijing, China

[2] School of Computing, National University of Singapore, Singapore

[3] Department of Electronic Engineering, National Taipei University of Technology, Taipei, Taiwan

liuyuxiang06@mails.tsinghua.edu.cn, wangye@comp.nus.edu.sg,

arun@arunshenoy.com, whtsai@ntut.edu.tw, clh-dcs@tsinghua.edu.cn

ABSTRACT

Music key, a high level feature of musical audio, is an effective tool for structural analysis of musical works. This paper presents a novel unsupervised approach for clustering music recordings by their keys. Based on chroma-based features extracted from acoustic signals, an inter-recording distance metric which characterizes diversity of pitch distribution together with harmonic center of music pieces, is introduced to measure dissimilarities among musical features. Then, recordings are divided into categories via unsupervised clustering, where the best number of clusters can be determined automatically by minimizing estimated Rand Index. Any existing technique for key detection can then be employed to identify key assignment for each cluster. Empirical evaluation on a dataset of 91 pop songs illustrates an average cluster purity of 57.3% and a Rand Index of close to 50%, thus highlighting the possibility of integration with existing key identification techniques to improve accuracy, based on strong cross-correlation data available from this framework for input dataset.

1. INTRODUCTION

Musical key which specifies the tonal center (also called tonic), describes the hierarchical pitch relationship in a composition. Tonic refers to the most stable pitch in a music piece, upon which all other pitches are referenced and scale implies pitch set which occur in a passage and interval between them. Therefore, key is extremely important for music representation and conveys semantic information about a composition. Automatic key estimation can be applied to many problems in content-based analysis of music, such as structure analysis and emotion detection, and also in music retrieval & recommendation systems.

Although considerable work can be found in the literature addressing the problem of estimating music key from audio signal automatically, it is still a challenging task. Major difficulties lie in the fact that key is a high level feature and difficult to extract from audio signals based on complexities of polyphonic audio analysis. Krumhansl [9] proposed a Maximum key-profile correlation(MKC) method that compares spectrum of music piece with key profiles which are derived by probe tone rating task and represent perceived stability of each chroma within the context of a particular key. The key that provides the maximum correlation with the music piece is considered as the solution[13]. Some modifications of MKC are involved, such as [19] which introduces Bayesian probabilistic model to infer key profiles from a pattern of notes. In [2], Chew described a mathematical model called Spiral Array, where pitch class, chord and key are spatially aligned to points in 3D space using a knowledge-based approach. The distance from tonal center of the music piece to the key representation acts as a likely indicator, and key estimation is performed through finding the nearest neighbor of the music piece. İzmirli[6] calculated similarity of tonal evolution among music pieces via dynamic time warping. Martens *et. al.*[12] compared decision tree methods with distance-based approaches. However, supervised classification approaches need a large scale of annotated data to train a model and new samples can't be reliably detected. Shenoy and Wang[16] adopted a rule-based method that combines higher level musical knowledge with lower level audio features. Based on a chromagram representation for each beat spaced frame of audio, triads are detected and matched against templates of key patterns, to identify the key with the highest ranking. Zhu and Kankanhalli[23] attempted to extract precise pitch profile features for key finding algorithms, considering the interference of pitch mistuning and percussive noise. Gómez and Herrera [4] extracted kinds of descriptors which can represent tonal content of a music piece and introduced machine learning models into key estimation. HMM based models have also been used to identify keys[11,14] and to detect key changes[1] by modeling timing and context information during a music performance. Most of the techniques discussed above suffer from the fact that complexities in polyphonic audio analysis make it rather difficult to accurately determine the actual template or position for each key.

In this paper, we propose a novel, unsupervised method to cluster music recordings into several categories without performing actual key identification. We believe that

This work was performed in National University of Singapore

elimination of training process, along with not having a dependency on template patterns of musical keys, makes this unsupervised framework applicable across a broad range of musical styles and can be fairly easily ported to other forms of clustering by changing input feature set. The performance of existing key identification approaches in the literature, though high, is observed to drop as dataset increases or musical recordings do not satisfy set criteria, like the presence of a strong beat and time signature of 4/4. Cluster information can thus serve as a valuable input to increase accuracy of key identification, on the basis of strong cross correlation of songs in a specific cluster, and the perceived stability of cluster purity over a large data set, demonstrated later in the evaluation section.

The rest of paper is organized as follows: the overview of the proposed approach is introduced in Section 2. The details of chroma-based extraction and inter-recording dissimilarity measurement are presented in Section 3. Section 4 describes the approach for cluster generation and number of clusters estimation. Experimental results are discussed in Section 5. Finally, conclusions are drawn in Section 6.

2. METHOD OVERVIEW

2.1. Problem Formulation

Given a dataset of N musical recordings, each one performed in one of P different keys, where P, the actual number of keys in the specific dataset, is unknown. Our aim is to produce a partitioning of the N recordings into M clusters such that $M=P$, and each cluster consists exclusively of recordings associated with the same key. Existing key identification algorithms could then potentially yield an improved performance from this cluster information. This schematic for the proposed framework is shown in Figure1 below.

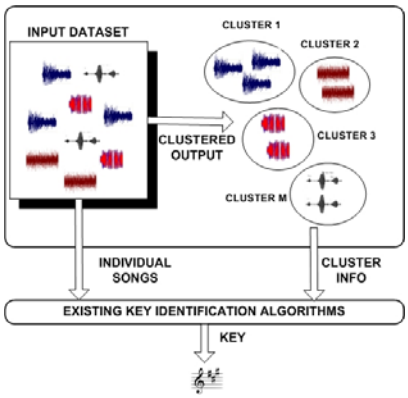


Figure 1. Clustering music recordings by their keys

2.2. System Configuration

As illustrated in Figure 2, the proposed clustering system consists of four major components: chroma-based feature extraction, computation of inter-recording dissimilarities, cluster generation and estimation of cluster number.

In the phase of feature extraction, pitch is estimated from the audio signal by spectral analysis technique and mapped into a chromagram. The dissimilarity computation based on chromagram is designed to produce small values for dissimilarities between recordings associated with the same key and large values for dissimilarities between recordings associated with different keys. Then, clusters are generated in a bottom-up agglomerative manner, followed by an automatic cluster number estimation algorithm.

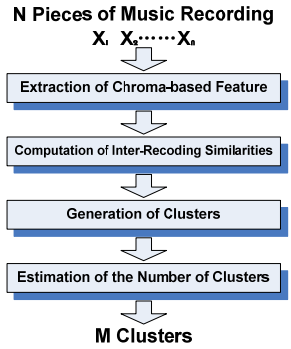


Figure 2. Framework for music clustering by keys

3. DISSIMILARITY MEASURE

Inter-recording dissimilarity computation is the most critical task in our work. The dissimilarity between recordings serves as a distance metric which imparts position and distribution of samples in the space. A good distance metric helps to gather similar samples together and make them easy to cluster. This section provides a viable approach to measure dissimilarity between music recordings, where spectrum divergence as well as harmonic center is taken into account.

3.1. Chroma-based Feature Extraction

Chroma-based[17] feature is a musical representation of audio data, where spectrum is reduced to 12 bins, corresponding to pitch classes in music theory. Intensity of frequencies is analyzed from audio and projected onto chroma scale. Two pitches separated by an integral number of octaves are mapped to the same element in a chroma vector. The output feature for each frame is a 12-dimensional vector, called chromagram, which stores the distribution of the energy for each of twelve semitones.

In our current system, input audio data is divided into half-overlapping 32ms long frames with Hanning widow. In each frame, spectrum is obtained via FFT. Then energy of pitches sharing the same pitch class are summed up and assigned to the corresponding bin in chroma vector.

3.2. SKL Divergence across Chroma Distributions

Kullback–Leibler divergence[10] in statistics, is a measure of difference between probability distributions. Its symmetrical version, SKL divergence, is proven to be effective for evaluation of distance between spectral envelops of audio signal[8,22]. Based on chroma vectors discussed in the previous section, we calculate expectations of chroma components and normalize them by total energy in each music piece. This spectral envelope can be interpreted as probability distribution of chroma components in a music piece. Thus, SKL divergence is utilized to measure difference between two chroma distributions with respect to musical keys.

3.3. Center of Effect

The spiral array[2] is a computational geometric representation for modeling tonality where pitches are mapped to points along a spiral in 3D coordinates. First, pitch classes are indexed by intervals of perfect fifths from a reference pitch, C for example. Then one increment in the index which stands for the interval of perfect fifth, leads to rotation of one quarter in horizontal plane as well as a height gain. And pitches with a major third apart (four increments of the index), result in vertical alignment with each other. This property satisfies the fact that interval of perfect fifth is responsible for the most consonant of the unison, while major third is the second. Then center of effect(ce) of a music piece is defined as the arithmetic mean of each pitch class weighted with its duration.

In our implementation, most of the configuration is similar to [2]. However, when calculating center of effect, we adopt energy in chromagram as the weight coefficient rather than duration and refined center of effect as:

$$ce = \sum_{i=1}^{12}\sum_{j=1}^{N}\frac{e_{ij}}{E} \times pitch_j \text{ where } E = \sum_{i=1}^{12}\sum_{j=1}^{N} e_{ij} \quad (1)$$

where $pitch_i$ denotes the coordinate of i^{th} pitch class and N is the total number of frames.

3.4. Inter-Recording Dissimilarity Measure

We utilize the linear combination of SKL divergence across chroma spectrum and Euclidean distance between center of effect to compute the overall dissimilarity between music recordings as follow:

$Div_{spectrum}$ denotes SKL divergence of the chroma spectrum envelope, while Dis_{ce} is the Euclidean distance between center of effect of two music recordings. Thus, the overall dissimilarity is

$$Dissim(i, j) = \alpha\beta Dis_{ce}(i, j) + (1-\alpha) Div_{spectrum} \quad (2)$$

where β is a normalization factor which normalizes both metrics into the same order, and α, a weighting coefficient, implies the bias between tonic and scale. They are set to 0.25 and 0.20 respectively by experience in real implementation. The linear combination of the two metrics satisfies the assumption that once one of the dissimilarity metrics increases, the overall dissimilarity will go up.

4. CLUSTER GENERATION

4.1. Agglomerative Clustering

After computing inter-recording dissimilarities, the next step is to assign the recordings deemed similar to each other to the same cluster. This is done by an hierarchical clustering method[7], which sequentially merges the recordings deemed similar to each other. The similarity is inferred from the metric described in Section 3.4. The algorithm consists of the following procedure:

```
Begin
    initialize M←N, and form clusters Ci←{Xi}, i=1, 2,..,N
    Do
        find the most similar pair of clusters, say Ci and Cj
        merge Ci and Cj
        M←M−1
    Until M = 1
End
```

Outcome of the agglomeration procedure is a cluster tree with the number of clusters ranging from 1 to N. The tree is then cut by optimizing number of cluster, which corresponds to an estimation of the number of keys actually occurring in the dataset.

4.2. Estimating Number of Clusters

According to the music theoretic basis of 24 keys (12 Major and 12 Relative Minor[1]) and the fact that not all these keys may be necessarily included in any sample data set, we limit the cluster number to be a maximum of 24 in our framework. However, in order to obtain a higher purity which is important for further processing, we can relax this limitation to a number, slightly higher than 24. Experiment shows that as long as the cluster number range is in a small neighborhood, the performance varies slightly. Additionally, we utilize a automatic cluster number estimation method, which has been used successfully to detect the number of speakers in the scenario of speaker clustering[21].

To evaluate the accuracy of clustering algorithm, here we follow the manner of [21], using two basic metrics: cluster purity[18] and Rand Index[5,15]. Cluster purity indicates the degree of agreement in a cluster. The purity for the m-th cluster C_m is defined as:

$$purity_m = \sum_{p=1}^{P}\left(\frac{n_{mp}}{n_{m*}}\right)^2 \quad (3)$$

where n_{mp} is the number of recordings in cluster C_m that are performed in the p-th key and n_{m*} is the number of

[1] We only consider major keys and natural minor keys here. In the rest of this paper, "minor key" refers to natural minor key except as noted.

recordings in the cluster of C_m. Deriving from Equation 3, *purity* follows $\frac{1}{n_{m*}} \le purity_m \le 1$ and is proportion to the probability that two music recordings in a cluster are in the same key. Specifically, the overall performance can be evaluated using average purity for all clusters:

$$\overline{purity} = \frac{1}{M}\sum_{m=1}^{M}(n_m \times purity_m) \quad (4)$$

The average purity is monotonically increasing with cluster number. This is based on the fact that as the number of clusters increases, average count of recordings in each cluster decreases, which leads to a higher purity. Hence, purity is not suitable for evaluating clustering performance, when the number of clusters is uncertain.

In contrast, Rand index, implying the extent of divergence of clustering result, is the number of incorrect pairs, actually performed in the same key but are placed in different clusters and vice versa.

Let n_{*p} denotes the number of recordings associated with the p-th key. Rand index can be calculated by:

$$R(M) = \sum_{m=1}^{M} n_{m*}^2 + \sum_{p=1}^{P} n_{*p}^2 - 2\sum_{m=1}^{M}\sum_{p=1}^{P} n_{mp}^2 \quad (5)$$

Rand index can also be represented as a mis-clustering rate:

$$R(M) \text{ in percentage} = \frac{\sum_{m=1}^{M} n_{m*}^2 + \sum_{p=1}^{P} n_{*p}^2 - 2\sum_{m=1}^{M}\sum_{p=1}^{P} n_{mp}^2}{\sum_{m=1}^{M} n_{m*}^2 + \sum_{p=1}^{P} n_{*p}^2} \times 100\% \quad (6)$$

It's obvious that the smaller the value of $R(M)$ is, the better the cluster performance will be. It has been proven that the approximately minimal value will be achieved, when the number of cluster is equal to the actual number of keys occurring in the dataset[21]. So our task is to search for a proper cluster number, such that Rand Index is minimized.

Recalling the Rand Index in Equation 5, the first term in the right side of the equation, $\sum_{m=1}^{M} n_{m*}^2$, can be computed based on the clustering result. Meanwhile the second term, $\sum_{p=1}^{P} n_{*p}^2$, is a constant irrelevant to clustering. The third term, $\sum_{m=1}^{M}\sum_{p=1}^{P} n_{mp}^2$ requires that the true key attribute of each recording is known in advance, which cannot be computed directly. To solve this problem, we represent it by

$$\begin{aligned}\sum_{m=1}^{M}\sum_{p=1}^{P} n_{mp}^2 &= \sum_{m=1}^{M}\sum_{p=1}^{P}[\sum_{i=1}^{N}\delta(h_i,m)\delta(o_i,p)]^2 \\ &= \sum_{m=1}^{M}\sum_{p=1}^{P}[\sum_{i=1}^{N}\delta(h_i,m)\delta(o_i,p)][\sum_{j=1}^{N}\delta(h_j,m)\delta(o_j,p)] \\ &= \sum_{m=1}^{M}\sum_{p=1}^{P}\sum_{i=1}^{N}\sum_{j=1}^{N}\delta(h_i,m)\delta(o_i,p)\delta(h_j,m)\delta(o_j,p) \\ &= \sum_{i=1}^{N}\sum_{j=1}^{N}\delta(h_i,h_j)\delta(o_i,o_j)\end{aligned} \quad (7)$$

where $\delta(\cdot)$ is Kronecker Delta function, h_i is the index of cluster where the i-th recording is located, and o_i is the true key attribute of the i-th recording. Note that h_i, $1 \leqslant i \leqslant N$, is an integer between 1 and M, if M clusters are generated. The term $\delta(o_i,o_j)$ in Equation 7 is then approximated by the similarity between Xi and Xj.

$$\delta(o_i,o_j) \leftarrow \begin{cases} 1, & \text{if } i = j \\ S(X_i,X_j), & \text{if } i \neq j \end{cases}$$

where $S(Xi, Xj)$ is a similarity measure between Xi and Xj, and $0 \leqslant S(Xi, Xj) \leqslant 1$, which derives from dissimilarity metric(Equation 2). Hence, the optimal set of cluster indices can be determined by

$$M^* = \arg\min R'(M) \quad (8)$$

where $R'(M) = \sum_{m=1}^{M} n_{m*}^2 + \Omega - 2\sum_{i=1}^{N}\sum_{j=1}^{N}\delta(h_i,h_j)S(X_i,X_j)$ is the estimated Rand Index.

5. EXPERIMENT

The evaluation of the framework has been carried out in two phases - effectiveness of the dissimilarity metrics, and the clustering algorithm. The test dataset consists of 91 pop songs, which include 21 out of 24 keys. The audio has been collected from CD recordings and contain the singing voice together with musical instrument accompaniment. The files are stored as 44 kHz, 16 bit, mono PCM waveform. Ground truth for the actual key information has been obtained from commercially available sheet music[1].

5.1. Dissimilarity efficiency validation

For convenience, we denote the relationship between music pieces as intra-class, relative-class, parallel-class and inter-class. Musical pieces with the same key are intra-class; share the same Major/Relative Minor combination of keys are relative-class (for example C Major and A Minor); share the same tonic but are in Major and Minor modes are parallel-class (e.g. A Major vs. A Minor), while inter-class means two pieces are in unrelated keys. The Dissimilarity evaluation is carried out using SKL Divergence of chroma spectrum and Euclidian Distance between center of effect, of two music recordings. The results are discussed below:

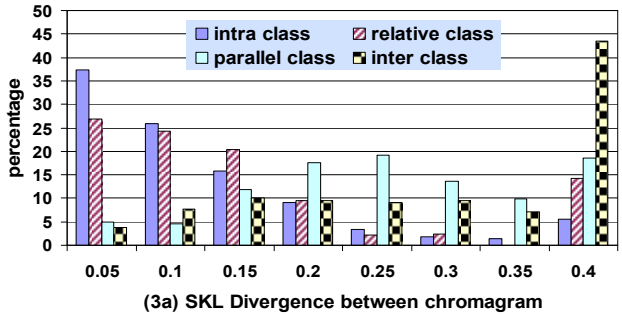
		SKL Divergence of chroma spectrum	Euclidian Distance between Center of Effect
Intra class	**average**	**0.1402**	**0.2917**
	Stat. lower bound[2]	**0.0004**	**0.0193**
	Stat. upper bound	**0.2177**	**0.4419**
Relative class	**average**	**0.1860**	**0.3344**
	Stat. lower bound	**0.0330**	**0.0675**
	Stat. upper bound	**0.4777**	**0.5224**
Parallel class	**average**	**0.2715**	**0.5160**
	Stat. lower bound	**0.0740**	**0.2317**

[1] http://www.musicnotes.com/, Commerical Sheet Music Archive
[2] Stat. lower/upper bound stands for the bound of interval, which contains 80 percentile of the total samples

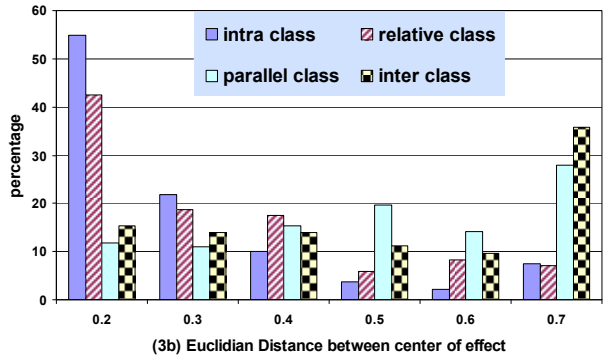
Inter class	Stat. upper bound	0.4092	0.8670
	average	0.3681	0.5637
	Stat. lower bound	0.1292	0.2087
	Stat. upper bound	1.1042	1.6324

Table 1. Dissimilarity between music with various keys

From Table 1 it is seen that average SKL Divergence for inter-class samples (0.36) and parallel-class samples (0.27) is much higher than that of the intra-class (0.14). This difference can be further demonstrated by Figure 3a, which shows the percentage distribution of SKL Divergence. It is observed that the total percentage of the intra-class distances less than 0.2 is 78%, while 80% of inter-class and parallel-class distances are greater than 0.2. A similar trend is observed for Euclidian Distance between Centre of Effect in Table 1 and Figure 3b.



(3a) SKL Divergence between chromagram



(3b) Euclidian Distance between center of effect

Figure 3. Proportion of dissimilarity metric in various intervals

These results corroborate significant confidence in distinguishing intra-class samples from parallel-class and inter-class samples. However, relative-class distances are observed to be much more difficult to distinguish as the SKL Divergence and Euclidian Distance, are both observed to be fairly close to the intra-distance. This can be explained by the music theoretic knowledge that relative keys share the same scale and similar harmonic structure. Thus, chroma features of relative-class pieces have similar distribution. In addition, modulations between relative modes are common in tonal music, which makes it harder to identity whether a song is primarily structured around a Major or its Relative Minor key (for example, a song with the verse sections in C Major and the Chorus sections in A Minor).

5.2. Clustering performance evaluation

The accuracy of clustering results is evaluated via cluster purity and Rand Index. Figure 4 reveals cluster purity and Rand Index as well as their correlation with the number of clusters. It can be observed that the cluster purity is always above 50%. On the other hand, the Rand index is relatively stable around 50% and reaches minima of 48% when the number of clusters is 21 (the actual number of keys occurring in the test dataset as per ground truth). This follows the discussion in Section 4.2 that Rand Index will reach its minimal value, when the number of cluster is equal to the actual number of keys occurring in the data.

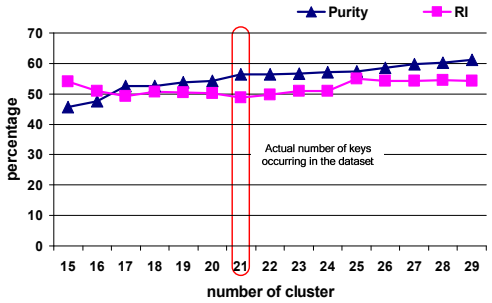


Figure 4. Clustering accuracy evaluation

The number of clusters predicted by our system for the test dataset is observed to be 24 based on a minimum of estimated Rand Index(computed by Equation 8), while the cluster purity is observed to be 57.2%. It can be seen that this is fairly close to the actual number of clusters - 21.

On further analysis of the clustering result, we find that quite a few errors are caused because of the Major/ Relative Minor ambiguity. A straightforward approach to reduce the confusion here would be to merge such keys into key groups[12], which implies that the signature for each cluster is a combination of 2 keys - the Major and its Relative Minor. In our experiments, the cluster purity has been observed to be as high as 70% with this change. Furthermore, errors are also caused because the algorithm is sometimes unable to distinguish the tonic from the dominant (perfect fifth interval). The overlap of harmonic components makes it difficult to identify the chroma component the harmonic belongs to. Besides, perfect fifth is a basic element when construction of triads is concerned in harmony. Similar errors were also observed in [3,12].

6. CONCLUSION

In this paper, a framework has been presented to cluster a higher level feature of music, the key, in an unsupervised way, discarding prior training information and music theoretic based rules. To the best of our knowledge this is the first attempt in this direction and hence there is no strong basis for evaluation against existing techniques. An empirical evaluation shows that accuracy of the existing

rule-based key estimation approach[16] suffers a decrease from over 80% to around 60% as the dataset has been scaled from 30 to 91 songs. From the discussion above, it is observed that the clustering performance is still stable as the data set grows because the dependency on specific higher level musical knowledge is not present. This gives us sufficient confidence that clustering, if involved as a preprocessing component in key detection tasks, will contribute in improving the accuracy for key estimation in large music databases. Future work will focus on integrating the clustering framework with key detection techniques to evaluate performance and scalability. Although the clustering framework is not yet sufficient to output the actual key assignment for each music recording or cluster, we believe it could provide useful information for further structure analysis of musical work, in addition to music retrieval & recommendation systems, and emotion recognition systems.

7. ACKNOWLEDGEMENT

This work was supported by Singaporean Ministry of Education grant with the Workfare Bonus Scheme No. R-252-000-267-112 and National Science Foundation of China (No. 60433030).

8. REFERENCES

[1] Chai, W. and Vercoe, B., Detection of key change in classical piano music, *Proc. of the International Symposium on Music Information Retrieval*, 2005

[2] Chew, E., The spiral array: an algorithm for determining key boundaries, *Music and Artificial Intelligence*, Vol. 2445, 2002

[3] Chuan, C. H. and Chew, E., Fuzzy Analysis in pitch-class determination for polyphonic audio key finding, *Proc. of the International Symposium on Music Information Retrieval,* 2005

[4] Gómez, E. and Herrera, P., Estimating the tonality of polyphonic audio files cognitive versus machine learning modelling strategies, *Proc. of the International Symposium on Music Information Retrieval*, 2004

[5] Hubert, L. and Arabie, P., Comparing partitions, *Journal of Classification*, Vol. 2, Issue 1, 1985

[6] İzmirli, O., Tonal similarity from audio using a template based attractor model, *Proc. of the International Symposium on Music Information Retrieval*, 2005

[7] Kaufman, L. and Rousseuw, P. J., Finding Groups in Data: An Introduction to Cluster Analysis, John Wiley and Sons, New York, 1990

[8] Klabbers, E. and Veldhuis, R., Reducing audible spectral discontinuities, *IEEE Trans. on Speech and Audio Processing,* Vol. 9, Issue 1, 2001.

[9] Krumhansl, C., *Cognitive Foundations of Musical Pitch*, Oxford University Press, New York, 1990

[10] Kullback, S. and Leibler, R. A., On information and sufficiency, *Annals of Mathematical Statistics* Vol. 22, Issue 1, 1951

[11] Lee, K., Slaney M., Acoustic chord transcription and key extraction from audio using key-dependent HMMs trained on synthesized audio, *IEEE Trans. on Audio, Speech, and Language Processing*, Vol. 16, No. 2, 2008

[12] Martens, G., De Meyer, H., etc. Tree-based versus distance-based key recognition in musical audio, *Soft Computing- a Fusion of Foundations, Methodologies and Applications archive*, Vol. 9, Issue 8, 2005

[13] Pauws, S., Extracting the key from music, *Intelligent Algorithms in Ambient and Biomedical Computing*, Springer, Netherlands, 2006

[14] Peeters, G., Musical key estimation of audio signal based on HMM modeling of chroma vectors, *Proc. of DAFX*, McGill, Montreal, Canada, 2006.

[15] Rand, W. M., Objective criteria for the evaluation of clustering methods, *Journal of the American Statistical Association*, Vol. 66, No. 336, 1971.

[16] Shenoy, A. and Wang, Y., Key, chord, and rhythm tracking of popular music recordings", *Computer Music Journal*, Vol. 29, No. 3, 2005

[17] Shepard, R., "Circularity in judgments of relative pitch", *The Journal of the Acoustical Society of America*, Vol. 36, Issue 12, 1964

[18] Solomonoff, A., Mielke, A., etc. Clustering speakers by their voices, *Proc. of the IEEE International Conf. on Acoustics, Speech, and Signal Processing*, 1998.

[19] Temperley, D., A Bayesian approach to key-finding, *Proc. of the Second International Conf. on Music and Artificial Intelligence*, 2002

[20] Tsai, W. H., Rodgers, D. and Wang, H. M., Blind clustering of popular music recordings based on singer voice characteristics, *Computer Music Journal*, Vol. 28, No. 3: 2004

[21] Tsai, W. H., and Wang, H. M., Speaker Clustering Based on Minimum Rand Index, *Proc. of the IEEE International Conf. on Acoustics, Speech, and Signal Processing,* 2007

[22] Veldhuis, R. and Klabbers, E., On the computation of the Kullback–Leibler measure for spectral distances, *IEEE Trans. on Speech and Audio Processing,* Vol. 11, No. 1, 2003.

[23] Zhu, Y. W. and Kankanhalli, M. S., Precise pitch profile feature extraction from musical audio for key detection, *IEEE Trans. on Multimedia*, Vol. 8, No. 3, 2006.