



Audio Engineering Society Convention Paper

Presented at the 122nd Convention
2007 May 5–8 Vienna, Austria

The papers at this Convention have been selected on the basis of a submitted abstract and extended precis that have been peer reviewed by at least two qualified anonymous reviewers. This convention paper has been reproduced from the author's advance manuscript, without editing, corrections, or consideration by the Review Board. The AES takes no responsibility for the contents. Additional papers may be obtained by sending request and remittance to Audio Engineering Society, 60 East 42nd Street, New York, New York 10165-2520, USA; also see www.aes.org. All rights reserved. Reproduction of this paper, or any portion thereof, is not permitted without direct permission from the Journal of the Audio Engineering Society.

Using Multiple Feature Extraction with Statistical Models to Categorize Music by Genre

Ben Fields¹

¹Goldsmiths College, University of London, New Cross, London, United Kingdom

Correspondence should be addressed to Ben Fields (b.fields@ieee.org)

ABSTRACT

In recent years, large capacity portable personal music players have become widespread in their use and popularity. Coupled with the exponentially increasing processing power of personal computers and embedded devices, the way people consume and listen to music is ever changing. To facilitate the categorization of these personal music libraries, a system is employed using MPEG-7 feature vectors as well as Mel-Frequency Cepstral Coefficients classified through multiple trained Hidden Markov Models and other statistical methods. The output of these models is then compared and a genre choice is made based on which model gives the best fit. Results from these tests are analyzed and ways to improve the performance of a genre sorting system are discussed.

1. INTRODUCTION

This paper describes a robust automated genre categorization system, derived from extracted audio descriptors, that expands on existing systems[1],[2],[3],[4]. The audio descriptors come primarily from the MPEG-7 standard, though Mel-Frequency Cepstral Coefficients (MFCC) are used as well[5]. The system takes the audio signal classification system present within the MPEG-7 standard and expands upon it in a number of ways. Unless otherwise stated, all

MPEG-7 functions use the All-XM Matlab toolkit implementations[6].

2. THE SYSTEM

Within the MPEG-7 standard, the SoundModelDS classifier system is centered on a single audio feature: audio spectrum over time. In an effort to make a system robust enough to maintain or improve its accuracy with large and musically diverse datasets[7], more features are extracted while making a classifi-

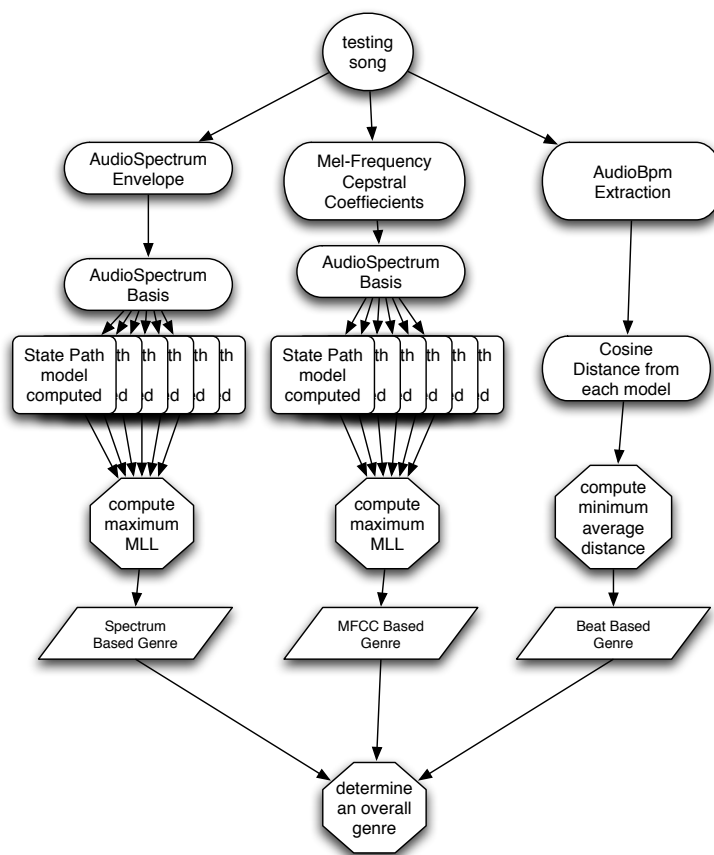


Fig. 1: An overview of all three testing decision chains and how they connect with each other.

cation decision. These features are each used independently to determine a genre. After each feature chain has made a decision about a song's genre, a final genre is selected based on the derived prevalent genre. If a clear genre is not derived, the confidence measure (as described in section 2.4) produced by each feature chain is used to assign the final genre of the test song.

There are three independent feature-decision chains in this sorting system. The first is based heavily on the SoundModelDS system of categorization[1]. The second is a feature chain based around an extraction of MFCC from the music audio file. The third is based on the beat and tempo related information as extracted by AudioBpmD (this feature extraction method is from the MPEG-7 standard). Each of these three chains outputs two pieces of data: the

genre estimate and the confidence measure for that estimate. The confidence measure is generated for each of the feature-decision chains that use Hidden Markov Models (HMM) based on the likelihood of the selected statistical model. A similar process is used for the tempo feature chain, based upon distance from the average tempo. This decision process can be seen in Figure 1.

2.1. The AudioSpectrumEnvelope Chain

This chain, as is true with all three feature chains, has two distinct processes. The first is a training by example process and the second is a testing process. The training examples can be as broad or focused as the particular application requires though it is important that whatever the methodology used for training sample selection, the entirety of the given genre is covered so as to minimize false negatives.

For each of these songs the AudioSpectrumEnvelope MPEG-7 descriptor is obtained. This is a logarithmic representation of the frequency spectrum on multiples of the octave and is the basic feature vector used in this entire process. Then the output of all of these AudioSpectrumEnvelope descriptor instantiations is passed into the AudioSpectrumBasis descriptor, see Figure 2. This descriptor is a wrapper for a group of basis functions that are used to project the AudioSpectrumEnvelope descriptors onto a lower dimensionality to facilitate classification. This descriptor is generated through a matrix multiplication of the AudioSpectrumEnvelope and the matrix produced by the basis functions. The output retains the maximum energy of the feature vectors, while reducing its dimensionality, helping to alleviate the dimensionality curse[8]. The output from this point contains the feature vectors that are used to create the HMM that will be used to make the determination as to which of the available genres our test information will fit. The HMM used in this implementation is informative to the MPEG-7 specification for the SoundModel descriptor scheme [9] and uses the standard solutions to the 3 critical HMM problems as can be found in [10]. It uses the Baum-Welch re-estimation algorithm to optimize likelihood.

The likelihood is defined as $P(x|\omega_k)$, where ω_k is the given genre class and x is the extracted feature data. This probability is found as a reduced proportion of Bayes' rule (Equation 1).

$$P(\omega_k|x) \propto P(x|\omega_k) \quad (1)$$

This reduction from Bayes' rule to Equation 1 is achieved by taking both $P(x)$ and $P(\omega_k)$ to be equal to 1. The maximum value of $P(\omega_k|x)$ is found by estimating $P(x|\omega_k)$ through the iterative use of the Viterbi algorithm[11] with a test song's extracted feature against a genre class' HMM to find the maximum likelihood of a given path within a HMM.

2.2. The MFCC Chain

In many ways the MFCC chain is similar to the AudioSpectrumEnvelopeD based chain. The only major difference is that rather than using spectral envelopes describing the signal, a matrix of MFCC are extracted. After all the training files have their MFCC matrices extracted, the stacked matrix is sent to AudioSpectrumProjectionD and from there to the

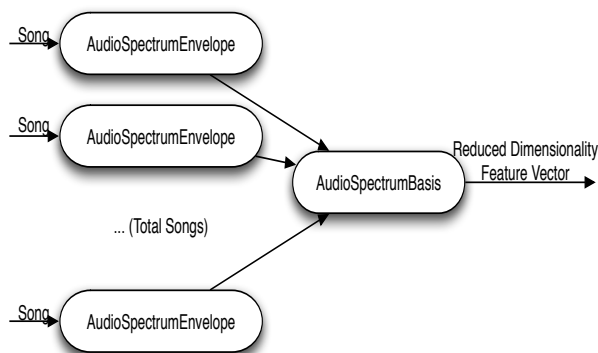


Fig. 2: The dimensional reduction process with the AudioSpectrumEnvelope feature

AudioSpectrumBasisD. This follows the same signal flow as the SoundModelDS. As seen in the SoundModelDS, the output of Audio-SpectrumBasisD is used to create and train an HMM. This training process can be seen in Figure 1. The HMM is then used by the testing process, along with the song to be tested, in the same testing process seen in the ASE Chain. The only difference being that the data, both in the genre class model and the test song, are derived through the extracted matrices of MFCC.

2.3. The Beat Chain

The third genre decision chain is based around beats per minute and other related information produced by the AudioBpmD descriptor. The creation of this model is a statistically simpler process than the model creation of the two prior chains. The model is composed of three $2 \times N$ matrices, where N is equal to the number of songs used in each model. These matrices store the values of each of the scalar values that are produced as output by the MPEG-7 v2 AudioBpmD descriptor[9], BPM, correlation and reliability. The correlation and reliability are both byproducts of the filterbank into combfilter methodology employed by the tempo detection algorithm [12]. The first row contains values calculated at the beginning of each audio file; the second row contains values calculated from the midsection of each audio file. These three matrices are then stored as the beat model for a given genre. The testing phase of the beat chain begins with the extraction of the six scalars using the AudioBpmD. Each pair of

scalars (BPM, correlation and reliability) can then be thought of as a vector in a two dimensional space. Similarly, each pair of scalars in the model matrices can be considered in the same way. Then a distance measure is taken between the test vector and each vector in the corresponding training matrix. A cosine distance measure is used here over simple Euclidean distance as the cosine distance has been found to yield better results in music similarity tasks [13]. Once the distance measures have been taken they are normalized and averaged together to yield an overall score of the test song against the genre model. This score is taken for each genre model and the minimum score (smallest average distance) across all the models is taken to be the genre from the perspective of the beat chain. This process puts equal emphasis on the tempo itself, as well as the reliability and correlation of the tempo. This accounts for the diverse range of tempo across different genres. Where some genres may see large difference in tempo, these same genres may show a high degree of independence and correlation in the reliability of that tempo or the correlation measure of the tempo.

2.4. Confidence Measure

The confidence measure, as defined in Equation 2, is used as a means to measure how strongly the chosen genre class matches the test song.

$$C = 100 \frac{\lambda_g}{\sum_1^n \lambda_i} \quad (2)$$

Where λ_g is the selected genre's normalized MaximumLogLikelihood and λ_i is genre i 's normalized MaximumLogLikelihood. In both of these cases the normalization occurs by Equation 3.

$$\lambda_i = l_i - \min([l_1, \dots, l_n]) \quad (3)$$

Where l_i is the MaximumLogLikelihood that a song fits in the HMM representing genre class i . This is useful in the event that there is no agreement across the three feature chains in selecting an overall genre class for a given test song, as can be seen in Figure 3.

3. EXPERIMENT

The experiment is designed to examine the systems structure and usefulness in a number of ways. The broadest way this is done is through the overall accuracy and the accuracy measures of each genre within

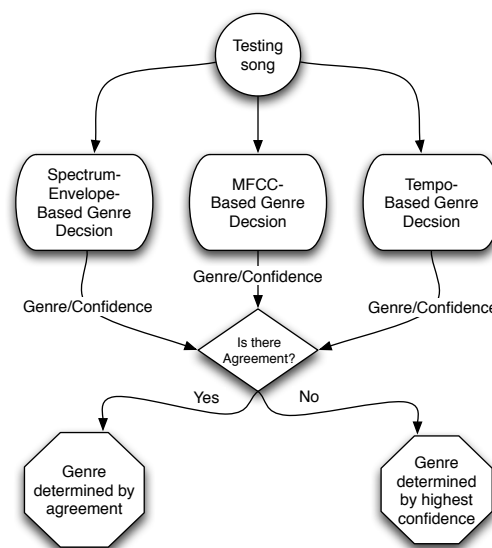


Fig. 3: overview of testing process with confidence indicators

the test set. In order to gain further insight about the system and how it responds to the data set, the errors in labeling are broken down by genre to look for emerging correlated properties between genres based on the feature dimensionality examined in the genre assignment process.

3.1. The Dataset

To facilitate full experimental trials, a significantly larger set of digital music test files is required. To that end, a means of selecting songs that are representational of a wide number of genres is needed. Since the nature of this system relies on statistical independence between genres, the specific songs chosen will clearly have a profound effect on the accuracy rates produced by the classifier. Further, it is important to establish an agreed upon ground truth genre for each song used and that that genre assignment be as objective as possible, given the inherent cultural subjectivity of genre and genre classification.

A natural fit for these requirements is the direct digital music retail download service. The music retail industry must make decisions and assign music into genre categories as part of its business, so the assignment of a genre is made as new material is acquired.

In a physical store this is done by placing media in different areas in the store (i.e. which box does the record go in?) and by analog this is seen through the use of metadata tags describing genre in the digital download music retailers. As a means for choosing a select representation from each genre, the 40 most popular songs, as determined by sales (in download purchases) within each genre on March 13, 2006, will be used. For these tests, Apples iTunes Music Store was used as the source of these lists, though there are many other digital download services available at the time of this writing that would have served just as well. A complete list of all songs used in the trial and their genre assignments, including the breakdown of training songs and testing songs, are available[14]. A simple random algorithm was used to divide each of these groups of 40 songs into sub groups of 15 songs for training and 25 songs for testing. The largest trial involved the use of 10 genres, with less genres being used in subsequent trials, as is described below.

3.2. Trial Runs

All ten genres selected test songs are used to create a model for each of the three feature-decision chains used in the system. These models were then used for a series of four test runs, each using a different subsection of the test songs. The purpose of the series of tests is to examine the effect of limiting the number of genre classes on the overall accuracy and the trends within the error spread of a given genre. For the first run, all ten genre-models and the associated 250 test songs were used in the trial (25 songs per genre). In the second and third trials, the two worst performing genres of the previous trial are removed from the data set. So, in the second run there are 200 test songs evenly distributed across 8 genres. Similarly, in the third trial in additional two genres are removed from the data set, leaving 150 songs across the remaining 6 genre classes.

4. RESULTS AND ANALYSIS

4.1. Ten Genre Trial

The ten genre-class trial was the first and largest of the trials using the data from the iTunes Music Store. As noted above, there were 15 songs used to train each genre model, for a total of 150 songs used in the training process. On the testing side, there were 25 songs used per genre for a total of 250 songs

used in testing, giving a grand total of 400 songs for the entire trial. As would be expected, these trials took a non-trivial amount of time to process. All software was built and ran in MATLAB 7 (R14) on an Apple G4 1Ghz Powerbook with 1GB of RAM. In that environment each model took at most 60 minutes to run, for a total run time of approximately 8 hours. The models were only built once and were used for all the subsequent trials. The testing process was a bit faster, though still lengthy, with each song taking a bit over a minute to process for a total testing run time of about 5 hours.

Accuracy rates on the full data set were less than stellar, with an overall accuracy rate of 37.75%. This accuracy rate was slightly better than the best performing of the three feature-decision chains, the spectral envelope chain, which had an accuracy rate of 37.35%. The accuracy of the MFCC chain came next with an accuracy of 32.1%. The tempo-based chain did the poorest with an accuracy of 13.7%. The overall accuracy of the entire system is broken down by genre-class in table 1.

4.2. Eight Genre Trial

As can be seen in Table 1, the two lowest accuracy genre classes in the ten genre class trial were electronic and rock, with accuracies of 12% and 20% respectively. So, for the trial with eight genres, these two genre models and their associated test files were removed from the data set and the test was run again. The elimination of these two models helps the overall accuracy considerably, increasing it to 51%. The full breakdown of system output versus expected output appears in Table 2.

As with the ten-class trial, the overall genre was slightly better than any single feature-decision chain. The accuracy order of the three chains remained the same, with the spectral envelope chains accuracy at 50.5%, the MFCC chains accuracy at 48% and the tempo-based chains accuracy at 17%. Even though the system as a whole did show improvement, it is interesting to note that two genres that have the lowest accuracies in this trial, Pop and R & B/Soul, both actually decreased in accuracy from the ten-genre run.

4.3. Six Genre Trial

This trial is of the six genre classes that have scored the most accurate on the prior tests. As seen in table

Table 1: Actual genre output by the system versus expected genre output for the large dataset, containing ten genre classes.

output/stated	A	B	C	E	F	H	J	P	RB	R
alternative	40	20	0	16	12	0	0	8	4	52
blues	4	52	0	4	24	4	28	0	4	4
classical	0	0	52	0	0	0	16	0	0	0
electronic	0	0	0	12	0	0	0	0	4	0
folk	0	8	4	4	48	4	8	0	12	8
hip-hop/rap	8	0	0	16	4	48	0	12	16	4
jazz	0	8	24	0	0	0	40	4	0	0
pop	20	4	16	12	12	4	4	32	28	0
R & B/Soul	8	4	4	24	0	36	4	36	32	12
rock	20	4	0	12	0	4	0	8	0	20

Table 2: Actual genre output by the system as a percentage versus expected genre output for the second dataset, containing eight genre classes.

output/stated	A	B	C	F	H	J	P	RB
alternative	52	20	0	12	4	0	28	4
blues	12	56	0	24	4	24	0	8
classical	0	0	56	0	0	16	0	0
folk	0	8	0	52	4	12	0	12
hip-hop/rap	8	0	0	4	48	0	16	16
jazz	0	8	24	0	0	40	0	4
pop	20	4	16	8	8	4	28	32
R & B/Soul	8	4	4	0	32	4	28	24

2, the two least accurate genre classes from this trial are Pop and R & B/soul. As such these two genres are removed from the dataset in the third trial in the series. This trial has 150 songs across the six remaining genre classes. The overall accuracy of this trial increased substantially with an overall accuracy of 77.3%. The genre-by-genre accuracy and error rates appear in table 3.

Interestingly, in this third trial, the spectral envelope chain had a higher overall accuracy by itself, 84%, than the overall system accuracy. This may be due to the smaller improvement seen in the MFCC chain, which correctly assign genre to 67.3% of the test data set. The tempo-based chain was again last, scoring correctly only 22.7% of the time.

4.4. Summary of Results

As the number of genre classes was decreased, the accuracy of the system improved from 38% with the ten genres dataset (table 1) to 77% with the subset of six genres (table 3). This relationship, as well as the relative performance of all three feature chains, can be seen in Figure 4. Though the first iteration of the trial lacked the accuracy seen in [1],[3], this may be due to the inconsistent and arbitrary nature of commercial genre assignment seen in the dataset, as the results improve significantly in the subsequent optimized iterations.

As can be seen in Figure 4 in the larger two trials the system showed improvement over any one feature chain by itself. Those from the AudioSpectrumEnvelope based feature-decision chain can be directly

Table 3: Actual genre output by the system as a percentage versus expected genre output for the third dataset, containing six genre classes.

output/stated	A	B	C	F	H	J
alternative	56	8	0	8	8	0
blues	4	56	4	28	4	32
classical	0	0	68	0	0	16
folk	4	28	8	60	12	12
hip-hop/rap	36	4	4	4	80	0
jazz	0	4	16	0	0	40

compared to the results in [1]. The performance of the chain (or any other single chain) compared to the entire system show that accuracy rates for each genre are more evenly distributed in the overall system.

5. ANALYSIS AND CONCLUSION

5.1. Overall Performance

On the whole, the hybrid song sorting system performed well, though with clear limitations. The most prevalent of these limitations (at least on the given test data) is one of genre overlap. This is caused by the training examples of each genre not being sufficiently dissimilar from other genres training material, along the dimensions of the feature chains in the system. The best way to see this visually is in the BPM scatter plot (Figure 5), which has the most pronounced overlap, rendering tempo chain only marginally helpful in improving the accuracy of the overall system. As a direct result of this overlap effect, the usefulness of this automatic system is significantly higher if there are fewer classes of data in the data set. This can be seen in Figure 4.

When genres are removed from the trial dataset, any overlap that genre contributed is also removed. This causes behavior that shows nearly exponential improvement in accuracy as genres are removed from the trial.

Taking this into account, it is also interesting to look beyond the accuracy rates of the various trials and examine where the errors were. In looking

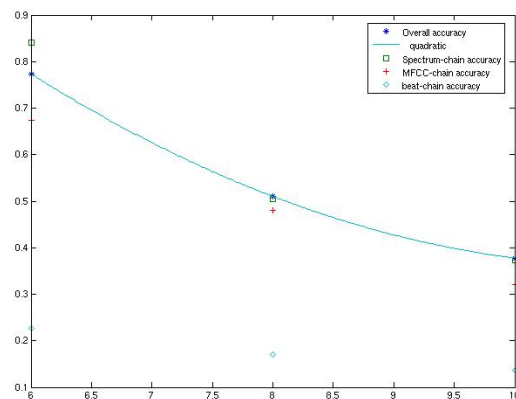


Fig. 4: Accuracy of classification against an increasing number of genre classes for each chain and the system as a whole.

at these errors, especially those in the large initial trial (seen in Table 1), it seems there are some telling patterns. There seems to be large variation in the topology of genre classes used to describe the set of 400 songs used. Looking at the hip-hop/rap genres distribution, 48% were correct, but a full 36% were thought to be of the genre R & B/soul. Conversely, R & B/soul, though a bit less accurate overall confirms this clear overlap of definition. Its accuracy rate is 32% yet 16% were incorrectly categorized as hip-hop/rap. This relationship is further exposed by the observable leap in accuracy in the hip-hop/rap genre when the R & B/soul genre is eliminated from the data set in between the eight class and six class trials. Similar patterns can be seen to a varying degree amongst many of the other genre classes. There seems to exist a triangular overlap of definition between alternative, blues and folk that continued throughout all three trials. All of these overlapping genres are of course dependant on the feature vectors extracted. The overlap is observable from the perspective of the features used in this system but there may exist features that would eliminate one or more of these boundary definition problems (i.e. melodic structure feature vectors). Another question that emerges from these trials is that of effect of training song selection. One of the more notable differences between the smaller first

data set used in the earlier trials and the larger one of ten genre classes is the means of selecting training songs. In the first data set the training songs were manually chosen to best represent their genre in the training model. In contrast, when using the larger data set the training songs were selected at random out of the available 40 songs genre group. This was done in an attempt to improve objectivity in the testing set-up however it may have had a substantial negative impact on the accuracy of the results.

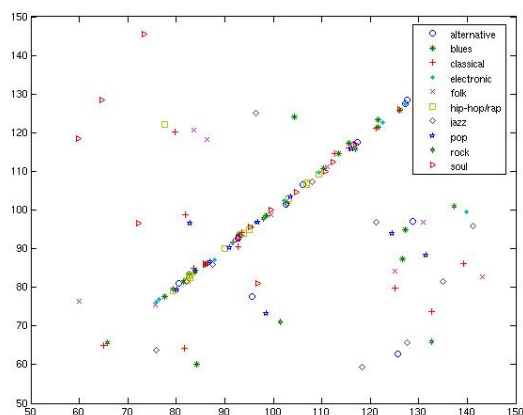


Fig. 5: A scatter plot of the BPM taken at two points in each song in the ten song dataset, training subset. The Y-axis is the 1st BPM reading and the X-axis is the 2nd. A data point that sits on the line $y = x$ means that the BPM tracker measured the same BPM at both points for that song.

5.2. What this says about commercial taxonomies

From the trials run on the data set from the iTunes Music Store, it seems clear that, at least through the eye of this sorting system, the genre topology used is far from ideal (as defined in [15]). It seems safe to say that this contributed at least somewhat to the high degree of error seen in the trials (especially the full test of all ten genres). That said, it is difficult to tell exactly how well this automatic sorting system could perform on a large number of genre classes if those genre classes were closer to ideal in their topology. Given the comparative last trial run on the two

four-genre datasets, it can be inferred that an improvement on the order of 10% - 15% would be reasonable to expect in a ten-genre test using data that employed a more intelligent taxonomy, with larger gains possible.

5.3. Future Work

There are a number of possible avenues of further study that can continue where this research is ending. Without changing the system, a worthwhile investigation could be seen in the use of a variety of song collections arranged in many genre topologies. One of particular interest would be a dataset with genres assigned to songs by a surveyed group of listeners. Then the automated genre assignment process can be evaluated against a group of people who have no commercial interest in the genre assignment (unlike the genre assignment of the iTunes Music Store, in which genre decisions have clear commercial effects).

There is also potential in improving the structure of the system itself. The most immediate of these possible changes is to supplement the existing three-chain system with other feature-decision chains based on more (preferable highly dissimilar to those currently in the system) feature vectors. Of particular interest are musically aware features, as features that has a deeper description of musical structure have a greater potential to separate what might otherwise be an overlapping topology. This could improve the accuracy of the system a great deal, though with a clear cost of computational time. Along these lines, clearly there is room for some improvement in the performance of the tempo-based feature decision chain. Perhaps some form of clustering could be used to increase independence of each genre prior to taking the distance. Although based on the distribution seen in the data set used for the trial in this document, there may not be much to gain through this course of action.

Lastly, there is great potential in the use of the concept used in the genre sorting system described above as a means to index a songs similarity to other songs along the dimensionality of the feature decision chains used in such a system. A quick way to achieve this would be to have every song be a genre by itself. Then each song would be tested against each of these one-song-genres. The genre that a song was placed in would in fact be the most similar song

(along the dimensions of the features in the system) to the test song. A modified system like this could offer a far more flexible and adaptable classification solution to a landscape of constantly changing music and culture.

6. REFERENCES

- [1] M. Casey, "Mpeg-7 sound-recognition tools," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 11, pp. 737 – 747, June 2001.
- [2] K. Kosina, "Music genre recognition," Master's thesis, University of Hagenberg, June 2002.
- [3] G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," *IEEE Transactions on Speech and Audio Processing*, vol. 10, July 2002.
- [4] A. Lampropoulos, P. Lampropoulou, and G. Tsihrintzis, "Musical genre classification enhanced by improved source separation technique," in *Int. Symposium on Music Information Retrieval*, 2005.
- [5] B. Logan, "Mel frequency cepstral coefficients for music modeling," in *Int. Symposium on Music Information Retrieval*, 2000.
- [6] M. Casey and et al., "All-xm.zip, audio reference software(iso/iec 15938-4:2001)." <http://mpeg7.doc.gold.ac.uk/mirror/index.html>, april 2001, 2003.
- [7] Z. Xiong, R. Radhakrishnan, A. Divakaran, and T. S. Huang, "Comparing mfcc and mpeg-7 audio features for feature extraction, maximum likelihood hmm and entropic prior hmm for sports audio classification," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2003.
- [8] J. Aucouturier and F. Pachet, "Tools and architecture for the evaluation of similarity measures : Case study of timbre similarity," in *Int. Symposium on Music Information Retrieval*, 2004.
- [9] J. Martinez, "Mpeg-7 overview (version 10)." <http://www.chiariglione.org/mpeg/standards/mpeg-7/mpeg-7.htm>, Oct. 2005.
- [10] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proc. of the IEEE*, vol. 77, pp. 257 – 286, Feb 1989.
- [11] G. D. Forney, "The viterbi algorithm," *Proc. of the IEEE*, vol. 61, pp. 268 – 278, March 1973.
- [12] E. Scheirer, "Tempo and beat analysis of acoustic musical signals," *J. Acoust. Soc. Am.*, vol. 103, pp. 588 – 601, Jan. 1998.
- [13] M. Cooper and J. Foote, "Automatic music summarization via similarity analysis," in *Proc. of Int. Symposium on Music Information Retrieval*, pp. 81 – 85, 2002.
- [14] B.Fields, "On the viability of using mixed feature extraction with multiple statistical models to achieve song categorization by genre," Master's thesis, University of Miami, Coral Gables, FL, May 2006.
- [15] C. Beghtol, "The concept of genre and its characteristics," *Bulletin of the American Society for Information Science and Technology*, vol. 27, Dec./Jan. 2001.