社団法人 情報処理学会 研究報告
IPSJ SIG Technical Report

2008－MUS－74 (18)
2008－SLP－70 (18)
2008／2／9

Content-based 音楽検索におけるビットレート違いを考慮した楽曲特徴量正規化手法

浜脇 修平† 石先 広海‡ 帆足 啓一郎‡ 滝嶋 康弘‡ 甲藤 二郎†

† 早稲田大学大学院基幹理工学研究科
‡ KDDI 研究所

あらまし

一般に大規模な音楽情報検索データベース内には異なったエンコード形式の楽曲ファイルが混在した形で保存されている．このような形式の違いは Content-based 音楽検索に於いて類似度を計算する際に意図しない誤差を生じさせ，最終的な検索結果に大きく影響を及ぼす場合がある．

本稿では音楽情報検索データベース内に混在するビットレートの異なる MP3 ファイルから抽出した音響的特徴に対して好適な補正方法を検討する．

Normalization of acoustic features to compensate diverse bit rates for content-based music retrieval

Shuhei Hamawaki† Hiromi Ishizaki‡ Keiichiro Hoashi‡ Yasuhiro Takishima‡ Jiro Katto†

† Graduate School of Science and Engineering, Waseda University
‡ KDDI R&D Laboratories Inc.

Abstract

In order to realize highly accurate content-based music information retrieval (MIR), it is necessary to compensate with the various bit rates of the songs which are included in the music collection, because the bit rate differences are expected to apply a negative effect to MIR results. In this paper, we propose methods to normalize MFCC features extracted from MP3 files with various bit rates, and analyze their effects to content-based MIR.

1. はじめに

記憶媒体の大容量化やネットワークのブロードバンド化によって，大規模な音楽配信サービスや個人での自作曲の公開などが行われるようになり大量の楽曲ファイルを参照，所持することが可能になってきた．

また，高音質のまま楽曲データの圧縮が行える MP3(MPEG-1 Audio Layer-3)や WMA(Windows Media Audio)，Ogg(Ogg Vorbis)といった技術を利用した小型携帯音楽再生機が普及し数千から数万曲の楽曲データを持ち歩くことが可能になった．

こうして膨大な楽曲ファイルを扱えるようになった一方，その中からユーザが自分の好みや状況に合った楽曲を探すことが困難になってきた．

このような問題に対しユーザにあまり負担を掛けず音楽データを検索する content-based 音楽検索が研究されるようになった．一般の content-based 音楽検索では楽曲ファイルの音響波形を扱い，音響的特徴を何らかの形で量子化し，それらの類似度をコサイン距離などで測ることによって検索クエリ楽曲との類似度が高い楽曲を検索結果として提示する．

音響的特徴には主にテンポやスペクトル，MFCC(Mel-Frequency Cepstrum Coefficient)などが用いられる．しかしこうした特徴量を抽出する際に楽曲波形ファイルのエンコード時のビットレートが異なると同じ楽曲であっても異なる特徴量が抽出されてしまう場合がある．参考論文[1]に於いては MP3 形式でエンコードを行ってもビットレート品質が高い場合，エンコード前の音楽波形ファイル(wav)と MFCC 値の誤差は小さいと述べられているが実際の音楽検索システムにビットレート品質の異なる楽曲ファイルが混在する場合には検索結果にビットレート品質による偏りが生じてしまう．

そこで本研究ではこのビットレート品質が音楽検索システムに与える影響を軽減するためデータベース内の楽曲ファイルのビットレートの統一や音響的特徴に対して正規化を行いその有効性を示す．

2. ビットレート品質の検索結果への影響

まず初めにビットレート品質による MFCC 値への影

響について検証してみる．

図1のようにCDからリッピングした各楽曲wavファイルに対して一度それぞれのビットレート品質でMP3へエンコードを行う．それらのファイルに対してMFCC値計算のため再度wavへデコードを行い，分析窓長25ms窓間隔10 msにて13次元(12次元+power)でMFCC値を算出したMP3_MFCCを作成する．エンコード，デコードにはLame (ver 3．97)を使用し，MFCC値計算にはHTK(ver 3．3)を使用した．

また指標としてエンコードを行っていないwav(Raw)から直接算出したMFCC値をRaw_MFCCとする．この作業により各wavファイルは数万フレームのMFCC13次元ベクトルの羅列として表現される．

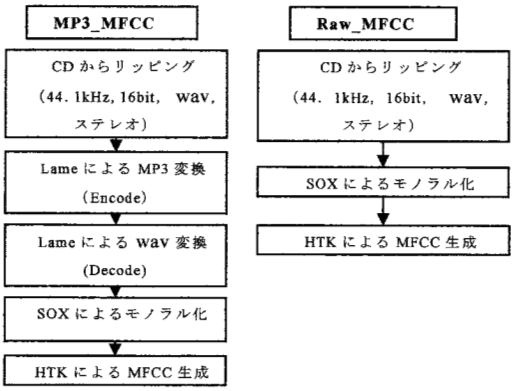

図1 MFCC値作成の様子

このように作成されたMFCCデータに対してまず同じ楽曲のRaw_MFCC(Raw)とビットレート品質128 kbps，64kbpsで作成したMP3_MFCC(128kbps，64kbps)の各次元での値を見比べてみた．図2ではある楽曲のMFCC値の内，1次元目のある短区間に於けるRaw_MFCC(Raw)とMP3_MFCC(128kbps，64kbps)の値を比較したものである．

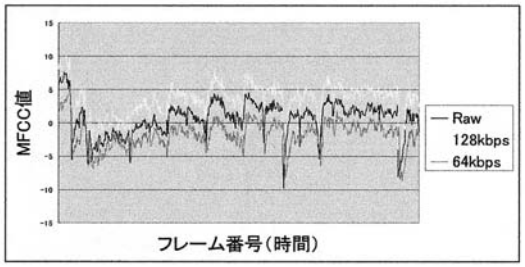

図2ビットレート品質によるMFCC値の変化

図2の様に各次元で平行移動をしたようなずれ方をしている箇所が多く見られた．

このようなMFCC値の誤差が検索にどの様に影響するのかを実際に音楽検索システムを構築し，検索結果を見て調べてみる．

今回は参考論文[2]と同様にベクトル量子化手法である TreeQ[3][4]を用いて先ほど求めた音響的特徴であるMFCC値(MP3_MFCC，Raw_MFCC)をヒストグラム(多次元ベクトル)化する．

TreeQでは初めにグループ分けされた学習データの楽曲から抽出したMFCC値の各フレームが形成する特徴空間をグループ毎に最適に区切るVQtree (VectorQuantizationTree)を生成する．今回は初めにジャンル分けされたRWC楽曲100曲を用いてジャンルに基づいた初期VQtreeを作成する．

ジャンル毎に分かれるようにtreeを学習させた後，各検索対象楽曲は以下の図3のようにtreeにより数次元ベクトル（ヒストグラム）化され，これら検索対象楽曲の多次元ベクトルからなる楽曲特徴空間(hist)を作成する．

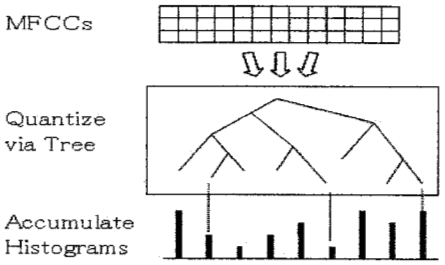

図3 TreeQによる楽曲特徴空間作成

しかしジャンルに基づいて構築された楽曲特徴空間ではpopなどの特定ジャンルの領域へ検索対象楽曲が密集し，偏りが生じてしまうためこの特徴空間内で検索対象楽曲に対してクラスタリングを行い学習データの選別を行う．そして最終的にその学習データを用いてVQtreeの再構築を行うことにより，偏りを解消する．その様子を図4にて示す．

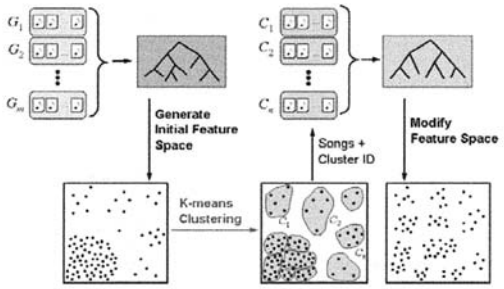

図4 特徴空間(VQtree)再構築の様子

最終的に構築されたVQtreeによる各検索対象楽曲

ヒストグラムの互いの距離を測ることで好みの楽曲に近い楽曲を探し出すことが可能になる．

上記手順にてビットレート品質が混在する楽曲データベースを用いて楽曲特徴空間を構築し，その様子を主成分分析にて示したものが図 5 である．

使用した楽曲データベースは様々なビットレート品質(96kbps～192kbps)にてエンコーディングされ作成された複数ジャンルの日本人アーティスト楽曲(過去の HMV Japan のランキング上位曲，708 曲)と韓国のアーティスト楽曲(1875 曲)，の MP3 ファイル 2513 曲から構成される．

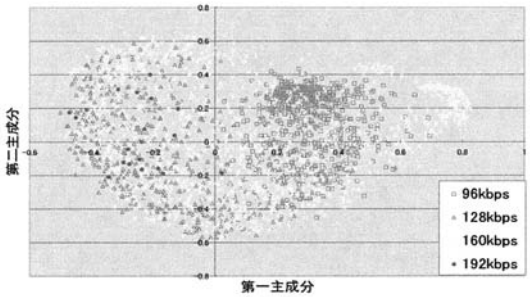

図 5　特徴空間内の様子

ビットレート毎の特徴空間内での領域の偏りが図 5 に見られる．この様に MP3 エンコード時のビットレート品質が高い場合には，その楽曲自体のエンコード前と後での MFCC 値にはあまり大きな誤差はないが，データベースの中に元の wav に対して MP3 エンコード時に高いビットレートによって音響的特徴があまり変化していないものと，低いビットレートによって大きく変化したものが混在している場合には，そのまま MFCC からそれらの類似度を算出した場合，エンコード時のビットレート品質に左右された結果が返ってきてしまうことが分かった．特に TreeQ のような MFCC の値に対して閾値を設定して特徴空間を区切るような方法ではその影響が顕著であることが分かった．

この様に検索システムにビットレート毎の偏りがある場合には，検索時に検索クエリと似ている楽曲であってもビットレートが異なり類似度が小さいと判断され提示されないといった事や，異なるビットレートの似ていない楽曲でも類似度が大きいとされ，提示されるといったことが起こり得る．

そこで我々は様々なビットレート品質が混在するデータベースであってもエンコードの影響を受けていない wav(Raw)から作成した楽曲特徴空間に近い結果(精度)が出せるような音響的特徴の補正方法について検討を行った．

3. ビットレート統一による補正

初めに楽曲間のビットレート品質の違いを軽減する方法として考えられるのが一番低いビットレート品質に全ての楽曲を統一してしまう方法である．

3．1 実験データ

市販されている９６曲の JPOP 楽曲をＣＤからリッピングした wav ファイルを実験に使用した．

3．2 実験方法

ビットレート品質が検索結果に与える影響の傾向を見るためにビットレート品質をそれぞれ 192，128，64kbps で統一した実験データを用いて前章の方法で楽曲特徴空間を構築し，それぞれ 192，128，64_hist とする．また，混在したデータベースを想定した 96 曲それぞれのビットレート品質を 192，128，64kbps のどれかにランダムで変換した実験データから作成した特徴空間を mix_hist とする．

最後にエンコードによってビットレート品質に影響の受けていない，そのままの wav(Raw)からなる実験データから作成した楽曲特徴空間を Raw_hist とする．
192，128，64，mix_hist について楽曲ヒストグラム間のコサイン距離に基づいた検索結果を提示して，それらがどれだけ Raw_hist での結果に近いのかを調査する．

3．3 評価方法

192，128，64，mix_hist について実験データの楽曲の内どれかを検索クエリとした場合の他の楽曲全てとのコサイン距離(行列)について Raw_hist での結果との相関を測りこれを全ての実験データをクエリとした場合について平均をとる．
N 個のデータを持つ行列 x, y の互いの相関 r は以下の式により求めることが出来る．

$$r = \frac{\sum_{k=1}^{N}(x_k - \bar{x})(y_k - \bar{y})}{\sqrt{\sum_{k=1}^{N}(x_k - \bar{x})^2}\sqrt{\sum_{k=1}^{N}(y_k - \bar{y})^2}} \quad \cdots (1)$$

また別の指標として 192，128，64，mix_hist それぞれについて実験データの内どれかを検索クエリとした場合の検索結果上位 10 曲の中に Raw_hist で同じ実験データを検索クエリとした場合の検索結果上位 10 曲と一致するものが何曲含まれているかの一致度を測り全ての実験データをクエリとした場合に対して平均をとる．

3．4 実験結果

図 6 は Raw_hist にてある楽曲をクエリとして他の楽

曲とのコサイン距離を計算し，コサイン距離の大きな順(類似度の高い順)に楽曲並べたものと他の楽曲特徴空間(hist)に於いて同じ楽曲をクエリとした時のコサイン距離を重ねて表示したものである．ビットレート品質の楽曲間距離に与える影響を示している．

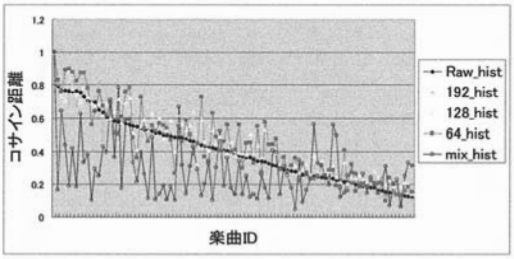

図 6 特定楽曲と他の楽曲とのコサイン距離例(降順)

上図のように様々なビットレート品質が混在した特徴空間(mix_hist)ではもとの特徴空間(Raw_hist)から大きく誤差のある値が出てしまっているのが分かる．

他の楽曲をクエリとした場合でも同様の傾向が見られる．

続いてこれらのコサイン距離(行列)に対して 3．3 の評価方法にて述べた評価方法で測った相関と上位 10 曲の一致度の平均を表 1 にて示す．

表 1．各特徴空間の Raw_hist との相関と一致度

特徴空間	相関	一致度(曲数)
192_hist	0．899753	6．28
128_hist	0．898979	6．25
64_hist	0．850452	5．25
mix_hist	0．219189	2．23

基本的にビットレートの品質が低くなる毎にもとの Raw_hist との結果と比べて誤差が大きくなり，mix_hist では更に誤差が大きくなってしまうことが分かる．

MFCC の値はその楽曲ファイルのビットレートが一番低い時の状態で決定されるので例えば 128kbps から 64kbps へ品質を下げてエンコードを行い算出した MFCC 値と 192k bps から 64kbps へ品質を下げてエンコードを行い算出した MFCC 値にほとんど差は無い．このことと上の二つの結果から混在するビットレート品質のまま楽曲データを使用し特徴空間(mix_hist)を作成するよりは mix_hist の楽曲データを一度 64kbps のように一番小さなビットレートにすべて変換し特徴空間を作成する方が元の楽曲特徴空間 Raw_hist に検索結果が近づくことが分かった．

4. 正規化による補正

続いて楽曲データのビットレート品質が混在する場合の他の改善策として，図 2 のようなずれ方をしていることから MFCC の値を正規化することにより Raw_MFCC と各 MP3_MFCC の値を近づけることが出来ると考えられる．これによりビットレート品質が異なる(mix)楽曲データベースとエンコードを行っていない元(Raw)の楽曲データベースでの MFCC 値での誤差が小さくなり，互いに近い楽曲特徴空間が構築されると考えられる．また正規化での補正の場合，音響モデル（楽曲特徴空間）が使用するデータベース内の最低品質のビットレートに依存するビットレート統一手法に対して，最低品質のビットレートが異なるデータベース間に対しても同じ音響モデルを使用することが出来る．

MFCC の正規化手法には以下の三つのようなものがある[5].

4．1 正規化手法

・CMN:Cepstral Mean Normalization

CMN は，ケプストラムから，ある区間でのケプストラム平均を引くことで正規化を行う

$$\hat{C}(i) = C(i) - \mu(i) \quad \cdots (2)$$

$\hat{C}(i)$ は正規化後の第 i 次元ケプストラム $C(i)$ は正規化前の第 i 次元ケプストラム $\mu(i)$ はある区間での第 i 次元ケプストラム平均で，正規化した区間の平均を 0 にして一般に乗算歪みを除去するために用いられる．

・CVN:Cepstral Variance Normalization

CVN は，ケプストラムをある区間でのケプストラム，標準偏差で割ることで正規化を行う

$$\hat{C}(i) = \frac{C(i)}{\sigma(i)} \quad \cdots (3)$$

$\sigma(i)$ はある区間での第 i 次元ケプストラム標準偏差で，正規化した区間の分散は 1 にして波形のレンジ，スケールを統一することで耐雑音性を高めると考えられている．

・MVN:Mean and Variance Normalization

MVN は，CMN と CVN を組み合わせた正規化法で，ある区間で計算したケプストラム平均と標準偏差で正規化を行う

$$\hat{C}(i) = \frac{C(i) - \mu(i)}{\sigma(i)} \quad \cdots (4)$$

上に述べた3つの手法について音声認識の分野では正規化を行う際の区間について，精度とリアルタイム性のトレードオフを考慮しなければならないが，今回は音響モデルの学習のため，一般にリアルタイム性は必要ないので処理の区間を楽曲全体に対して正規化を行った．

図1で示した手順と同様の方法で同じ楽曲自身のRaw_MFCC(Raw)の各次元での値とビットレート品質128kbps，64kbps で作成した MP3_MFCC(128kbps，64kbps)に対し各正規化手法により補正を行った後の値を比べてみる．

図7は正規化によるMFCC値補正の一例として図2と同様の楽曲の次元，区間での各 MFCC 値についてMVN正規化後を行いその結果を示したものである．

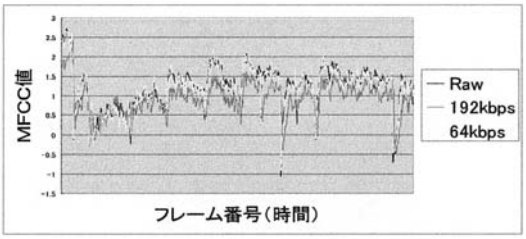

図7 MFCC の補正された様子

正規化を行わない図 2 の場合と比べて一度エンコードを行った楽曲から抽出した MFCC 値(MP3_MFCC)であっても，正規化により元の MFCC 値(Raw_MFCC)にその値が近づいていることが分かる．他の楽曲についても同様の効果が見られた．

そこで第3章と同様の実験データを用いて，混在した楽曲データベースを想定した 96 曲それぞれのビットレート品質を 192，128，64kbps のどれかにランダムで変換した実験データから MP3_MFCC を作成し，楽曲全体に対して CMN，CVN，MVN を行い MFCC 値を補正し，それを元に構築した楽曲特徴空間(mix_hist)と，Raw_MFCC に対してそれぞれの正規化手法で MFCC 値を補正し，それによって作成した楽曲特徴空間(Raw_hist)を構築し第 3 章と同様の評価方法を用いて実験を行った．

4．2 実験結果

図8は正規化のうち MVN を用いて MFCC 値の改善を行った実験データを用いて図5と同じ様に，ある楽曲をクエリとして他の楽曲とのコサイン距離を計算し値の大きな値の順に Raw_hist について並べたものと mix_hist にて同じ楽曲をクエリとした時の各楽曲のコサイン距離を重ねて表示したものである．

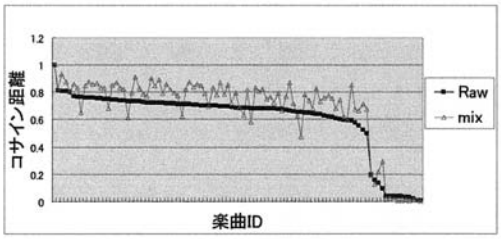

図8　正規化後のコサイン距離の様子

続いてこれらのコサイン距離(行列)に対して3．3の評価方法を用いて測った相関と上位 10 曲の一致度の平均を表3にて示す．

表 3．各正規化手法を用いて構築された特徴空間 Raw_hist と mix_hist の相関と一致度

正規化手法	相関	一致度(曲数)
CMN	0．714248	4．04
CVN	0．438516	2．77
MVN	0．859172	4．52

図6の Raw_hist と mix_hist の関係と図8の Raw_hist と mix_hist の関係を比べてみると，後者の方が Raw_hist の値が mix_hist の値に近づいていることが分かった．

また表3でも表1に於ける mix の項の値からそれぞれ改善されていることが分かった．

最後に図5で使用したビットレート品質が混在した楽曲データに対して 64kbps でビットレートを統一した場合(図9)と MFCC 値に対して MVN(or CMN)にて補正を行った場合(図 10)でそれぞれ楽曲特徴空間を作成しその様子を主成分分析にて示す．

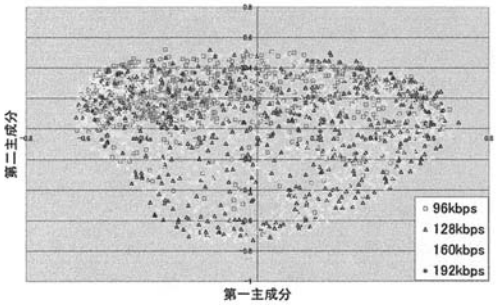

図9　補正された特徴空間（ビットレート統一）

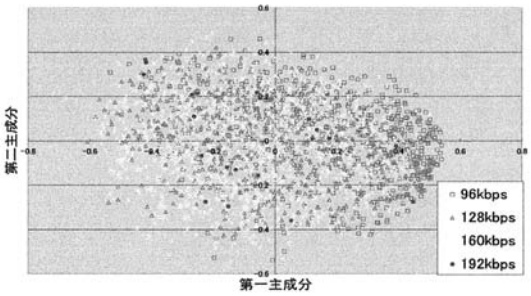

図 10 補正された特徴空間(正規化)

図 5 の様にビットレート品質の混在したデータベースをそのまま使用すると，楽曲のビットレート品質毎に特徴空間内での分布の様子が異なり，検索結果の中にクエリ楽曲と異なるビットレート品質の楽曲がある場合，その音響的特徴がクエリ楽曲と大きく異なることがあった．

しかし図 9，10 に示したような音響的特徴に対して補正を行った特徴空間では，ビットレート品質による偏りが改善されて検索クエリ楽曲とビットレート品質が異なる楽曲であっても音響的特徴の似た楽曲が検索される頻度が増すことが確認できた．

5. 考察

エンコード時のビットレート品質が音楽検索システムに影響を与える原因を考えると，第一にエンコード時の品質によって検索対象楽曲の音響的特徴が変化し，学習により得られていたカテゴリー分けされた音響モデルとの誤差が生じ，本来分類されるべきカテゴリーと異なったカテゴリーに分類される可能性がある．

もう一つは今回の様にデータベースに合わせて音響モデルを作成する場合，学習時にデータベース内の楽曲を使用するため，音響モデル自体が不適正なもとなり，検索対象楽曲が楽曲の雰囲気などからではなく，偶然に各カテゴリーの音響モデル作成の際に分かれた楽曲のビットレート品質毎に分類されてしまう可能性が考えられる．

第 3 章に示したようにエンコード時のビットレート品質が低ければ低いほどエンコードしない場合との差が大きくなるが，データベース内の楽曲のビットレート品質が全て高い場合などは一度ビットレート品質を一番低いものに統一することによって本来のエンコードを行わなかった場合の楽曲特徴空間に近づけることが出来る．

ビットレートの統一を行わずそのまま楽曲データベースを使用し音響的特徴を抽出する際にはエンコード時のビットレート品質による音響的特徴への影響について考慮する必要がある．

今回は音響的特徴の一つである MFCC 値について正規化にて改善を行ったが，各手法について比べてみると CVN は精度が他の手法に対しあまり改善されていないため楽曲ファイルに対しては分散により正規化するだけでは効果があまり得られないことが分かる．

但し CMN と MVN を比べると若干 MVN の方が精度が高いため複合的に用いることによって成果が出ると考えられる．

但し今回の実験で得られた結果はビットレートを統一した場合などに比べてまだ精度が低いため，今後更なる改良の余地があると考えられる．今回の精度 が低い理由の一つとして正規化の処理により，同じ楽曲に対してエンコードの前後での楽曲間の MFCC 値が近づいた一方で，本来離れているべき楽曲とも MFCC 値が近づいてしまった可能性が考えられる．今後はそれらに対しても検討を行っていく．

6. 終わりに

本稿では楽曲の音響的特徴である MFCC の値にエンコード時のビットレート品質が与える影響と，そのようなファイルが混在するデータベースを利用して content-based 音楽検索システムを構築した場合の問題点について検証を行うと伴に，音響的特徴に対して補正を行うことによりビットレート品質による検索結果への影響を軽減できることを確認した．

7. 謝辞

本研究に対して様々な助言を下さった早稲田大学甲藤研究室の皆様や実験用楽曲の収集，選定に関してご協力して下さったＫＤＤＩ研究所の研究員の皆様に深く感謝致します．

文　献

[1] Sigurdur Sigurdsson, Kaare Brandt Petersen and Tue Lehn-Schiøler , Mel Frequency Cepstral Coefficients: An Evaluation of Robustness of MP3 Encoded Music, Proceedings of the Seventh International Conference on Music Information Retrieval (ISMIR 2006).

[2] 石先 広海，帆足 啓一郎，松本 一則，甲藤 二郎 ，ユーザ嗜好に基づく音楽情報検索のための学習データ抽出手法，情報処理学会，春季全国大会 3G-6 ，2005

[3]J. Foote, Content-based retrieval of music andaudio, Proceedings of SPIE, Vol 3229, pp 138-147, 1997

[4] J. Foote, TreeQsoftware, http://treeq. sourceforge. net/

[5] 小川 厚徳，毛呂 良寛，高橋 敏， ケプストラム正規化の実行単位に関する実験的検証， 電子情報通信学会論文誌 D, Vol. J90-D, No. 9 pp. 2648-2651