論文番号 B1 － 1

類似音声検索による映像検索

須賀 啓敏　寺本 純司　片岡 良治　芳西 崇
日本電信電話株式会社　NTT サイバースペース研究所
〒 239-0847 神奈川県横須賀市光の丘 1 - 1
Tel: 0468-59-2758　Fax: 0468-59-2768
E-mail: {suga,junji,kataoka,honishi}@dq.isl.ntt.co.jp

Abstract

本稿では，音声を検索キーとし，大量に蓄積された映像の中から検索キーと類似する音声が出現するシーンを検索するシステムについて提案する．従来手法では，音声信号の非線形伸縮マッチングで類似音声検索を行っており，高速な処理が困難であった．本稿では，検索対象を線形伸縮で検索できる音声のみに絞ることでマッチングの高速化を図る手法を提案する．本手法は，線形伸縮だけをする音声である歌声の類似検索に有効である．そこで，歌声を検索キーとし映像中の類似する歌声が出現するシーンを検索するシステムについて検討し，プロトタイプシステムを作成して提案システムの有効性を確認した．提案システムを適用することにより，映像が大量に蓄積されているため，その中から見たい歌のシーンを探し出すことが困難な状況においても，歌声で検索して見ることができるようになる．

1　はじめに

1.1　背景

近年のネットワークのブロードバンド化に伴う映像配信サービスの開始，テレビ放送の多チャンネル化，今後のホームサーバーの普及に伴う各家庭における映像蓄積の大規模化等により，今後我々が見ることができる映像がますます増大していくことが予想される．このような状況下において，視聴者が大量の映像の中から見たい映像を探し出すことは大変労力のかかる困難な作業となってしまう．そこで，映像をデータベースに蓄積しておき，必要なときに必要な映像を素早く探し出してきて利用することができる技術が重要となってくる．

1.2　本稿で想定する適用先

様々な映像の中でも常に一定の地位を占めているのが，音楽番組や歌番組である．このような映像を対象とした検索は要望は大きいと推測される．例えば，最新の音楽ランキングを紹介する番組では，「この歌の順位を知りたいので，この歌のシーンを検索したい」という要望があると考えられる．また，中高年層には懐かしいメロディを紹介する番組にも根強い人気があるということから，過去の音楽番組や歌番組の映像検索にも需要があることが推測される．またコマーシャル等の BGM で，気になっていた歌を検索したいという要望も考えられる．

このような映像が，各家庭の映像蓄積用ハードディスクに大量にある状況を想定する．このとき，検索者が見たい歌のシーンが映像中のどこに入っているかわからない場合に，その歌の歌声を入力することで，その歌のシーンを頭出しすることが出来れば非常に便利である．

本稿では，蓄積された大量の映像の中から歌声を検索対象とした映像検索方式を提案する．またそのプロトタイプシステムを作成し，有効性の確認をおこなう．

2　現状と問題点

音声を検索対象とした映像検索の従来研究として，歌の映像を対象とした研究は少なく，ニュース映像を対象とした研究 [1][2] が多い．

現在の音声認識では，まず，入力音声を音声分析し，認識に有効な特徴を抽出する．そして，予め大量の音声データから学習し，音響モデルを構築しておく．また，文法規則，語彙あるいは言語統計などにより，言語モデルも構築しておく．この言語モデルで規定された探索空間の中で，入力音声と最も良く合致する音響モデルの列を探し出し，それを認識結果として表示する．しかし，雑音の影響，不明瞭な発話などの理由で音声分析がうまく出来ない場合や，文法規則等が守られないため言語モデルの構築が難しい場合は，認識率が低下するという問題がある．

ニュース映像における音声は，雑音が少なく，発話も

正確で，文法規則も守られるといった特性を持っている．これにより，音声認識の技術を利用して音声をテキストに変換することで音声の検索を実現できる．

それに対し本研究では，歌を検索するために歌声を扱うことにする．同じ音声でも，歌声はニュース音声とは異なる特性を持っているのでそれに適した検索方式が必要となる．

2.0.1 歌声の特性

以下に歌声が持つ特性を挙げる．

特性１ 文法規則が守られないことが多い．

特性２ 言葉としては同じであっても，歌い方が異なればそれは違う歌声となる．

具体例：「わたし〜」と歌った場合と「わ〜たし」と歌った場合は，言葉としては同じ「私」であるが，歌声としては歌い方が異なるため別の歌声である．

特性３ 歌にはもともとテンポが存在するので，歌声にもテンポが存在する．

これらの特性は，通常の音声とは違った特性である．

2.0.2 従来方式を歌声に適用したときの有効性

本節では 2.0.1 節で示した歌声の特性を踏まえて，従来の音声による映像検索方式を歌声に適用したときの有効性について検討する．

音声による映像検索方式は以下の２つの方式に分類することができる．

方式１：映像中の音声をテキストに変換しておき，検索キーにテキストを用いて検索する方式
具体例：Infomedia(CMU)[3]
（図１参照）

方式２：映像中の音声から特徴量を抽出しておき，検索キーに音声特徴量を用いて検索する方式
具体例：CrossMediator(RWCP)[4]
（図２参照）

上記の両方式を歌声に適用した場合の長所と短所を，2.0.1 節で挙げた特性を考慮して以下に示す．

方式１の長所と短所

長所 1-1：正確にテキストに変換することができれば，検索キーの歌声の男女差や個人差による違いをなくして検索することができる．

短所 1-1：**特性１**より，文法規則を用いることが有効でなくなる．

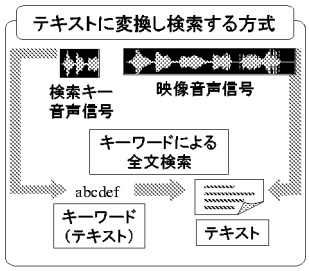

図 1: 方式１

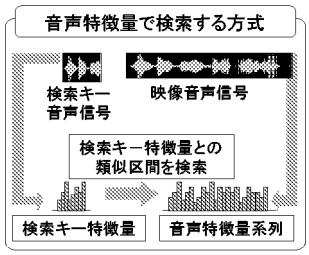

図 2: 方式２

短所 1-2：**特性２**より，異なる歌声でも同一の言葉であれば，テキストにすると同一になり，それらを区別して検索することができない．

方式２の長所と短所

長所 2-1：文法規則に依存せずに検索できるので，**特性１**に適している．また，単語の辞書や言語の種類にも依存しない．

長所 2-2：同じ言葉だが違う歌い方をしている部分を区別して検索することができるので，**特性２**に適している．

短所 2-1：男女差や個人差により同一の音素でも音声信号は異なる．音声特徴量にもそれが反映されるため，検索結果にも男女差や個人差の影響がでる．

方式１では短所 1-1 の理由から，文法規則を用いることが有効でなくなるため，音声をテキストに変換することが難しくなる．また方式１の長所 1-1 は音声がテキストに変換できない場合は有効とはならない．よって方式１を適用することは有効ではない．

一方，方式２にも短所 2-1 はある．しかし，音声認識で使われているような，音声信号から音素の性質に影響しない特徴を取り除いた音声特徴量を用いることで，音声信号の違いによる検索結果への影響は低減することができる．よって本研究では方式２を採用する．

2.0.3　音声特徴量を用いた場合の歌声による映像検索の問題点

音声特徴量による音声の検索の従来方式 [4] では，非線形伸縮した音声でも検索できるようにするため，非線形伸縮マッチングを採用して音声間の類似度を計算している．しかし，この従来手法をそのまま，本研究の目的である大量の映像からの歌声による映像検索に適用する場合には，以下の点が問題点として挙げられる．

問題点：**大規模映像データベースへの対応**
非線形伸縮マッチングを用いるため，マッチング方法が複雑となり計算コストが大きい．よって大量の映像を検索対象にすることには向いていない．

そこで，本研究ではこの問題点を解決することで，歌声による映像検索を実現する．

3　大規模映像データベースへの対応方法の提案

3.1　本提案手法の原理

本提案手法で最も重要となる原理について述べる．

まず先に，大規模映像データベースに対応するための仮定を導入しておく．この仮定は，2.0.1 節の歌声の**特性３**に基づくものである．

仮定１：
特性３より，歌声には一定のテンポがあるので，歌う時間区間が短ければ，その時間区間全体が線形に伸縮することはあっても，その時間区間内で非線形に伸縮することは少ない．

この仮定１を導入すると，短い時間区間であれば非線形伸縮はしないので，非線形伸縮マッチングする必要はなくなる．しかし，時間区間全体の線形伸縮は考慮する必要がある．そこで本研究では，**音声データを線形伸縮させて，ある一定の時間区間の長さの固定長データにそろえる．そして，固定長データ間の単純なマッチングで，音声データの類似度を表す距離を高速に計算する**．このようにして，大規模映像データベースに対応できる高速な検索を実現する方法を提案する．

3.2　本提案手法の手順

提案手法の手順を以下に示す．

手順１　基準となる長さの時間窓（基準時間窓）を少しずつずらしながら，検索対象の音声から時系列特徴量を抽出する．(図 3 参照)

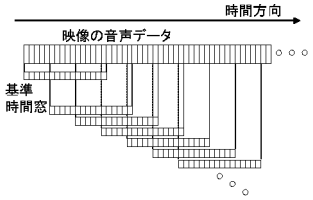

図 3: 手順１

手順２　基準時間窓の長さを中心とした複数の長さの時間窓を用意し（図 4 参照），それぞれ，手順１と同様にして音声から時系列特徴量を抽出する．

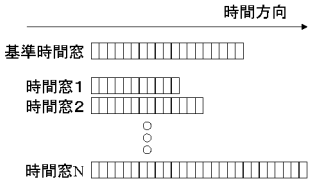

図 4: 手順２

手順３　複数の長さの時間窓で抽出されたすべての時系列特徴量を，時間軸方向に線形伸縮して，基準時間窓と同じ長さにする．(図 5 参照)

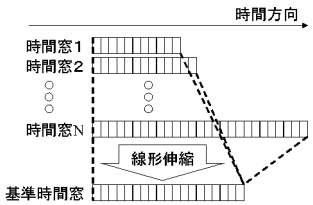

図 5: 手順３

手順４　基準時間窓の長さの時系列特徴量を線形圧縮し，時系列特徴量ベクトルを作成する．(図 6 参照)

手順５　検索キーの音声特徴量を，基準時間窓の長さで抽出して時系列特徴量を得る．これも手順４と同様に線形圧縮して，時系列特徴量ベクトルを得る．

手順６　音声の類似度を表す距離として，その時系列特徴量ベクトル間の距離を用いる．検索キーの時系列特徴量ベクトルとの距離で，検索対象の時系列特徴量ベクトルを順位付けする．よって，それに該当するシーン（映像区間）も順位付けられる．

手順７　時間窓を少しずつずらしながら音声特徴量を抽出するため，近い順位に順位付けられたシーンは，

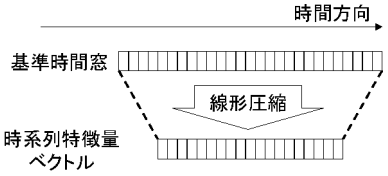

図 6: 手順 4

互いに時間区間が重複する部分が多く，同じようなシーンとなる．そこで，そのようなシーンは，最上位のシーンだけを検索結果として出力し，それ以外は検索結果として出力しないようにする．■

3.3 提案手法による効果

上記提案手法により，以下のことが実現される．

- 時系列上の線形伸縮を考慮したマッチングが，単純な固定長ベクトル間のマッチングで実現できるため，高速な検索が実現される．(図 7 参照)

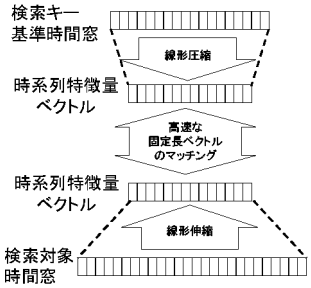

図 7: 固定長ベクトル間マッチングによる高速化

- 固定長ベクトルのマッチングでは，多次元空間インデックス（図 8 参照）を導入して，Tree 構造を利用した高速な検索を実現できる．

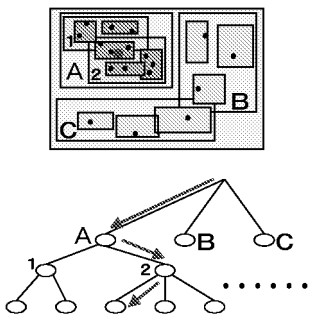

図 8: 多次元空間インデックス

これらより，高速な検索が実現されるため映像データベースが大規模化しても対応することが可能となる．また多次元空間インデックスは，データ量の増加に対して検索の計算量は対数のオーダーでしか増えないので，データベースが大規模になるほど有利な方式である．

4 提案方式によるシステムの実装

4.1 メル周波数ケプストラム係数 [6]

本研究で提案するシステムでは，音声信号の特徴量に MFCC （Mel-Frequency Cepstrum Coefficent : メル周波数ケプストラム係数）を用いる．

人間の音声は，声帯の振動などの「音源」により生成された音が，「調音フィルタ」となる声道を通過することで様々に変化したあと，口または鼻から放射され発生している．音声言語の最小単位である音素は，主に調音フィルタの性質によって規定される．一方，音の高さや大きさといった音声の韻律的な性質は，主に音源の性質から規定される．

MFCC は，音素の性質に直接影響を与えない音源の性質を，音声信号から分離することが出来るものである．MFCC を用いることで，音源の違いによる影響を受けずに，音素の性質に影響をもつ調音フィルタの性質だけを扱うことが出来る．

MFCC の算出方法は，付録 A に示す．

この MFCC の低次元の項には調音フィルタの性質が反映され，高次元の項には音源の性質が反映される．このうち音素の性質に関係があるのは調音フィルタであるので，MFCC の低次元の項のみを使うことで，音源の男女差や個人差などの違いを取り除くことができる．

さらに，マイクロフォンの特性等の音響系の影響を軽減するために，CMS （Cepstrum Mean Subtraction : ケプストラム平均除去）処理 [6] をおこなっておく．

本提案システムでは，この MFCC を時系列上に並べたものを音声特徴量データとして扱うことにする．

4.2 音声特徴量データの生成

4.2.1 MFCC の生成

本提案システムでは，映像データとして MPEG1[サンプリング周波数 44100Hz，量子化ビット数 16bit，2ch] を使う．その音声データ部分を wave ファイル [サンプリング周波数 44100Hz，量子化ビット数 16bit，1ch（元の MPEG1 の左 ch）] に変換して MFCC の生成を行う．

この wave ファイルの音声を短時間区間分析窓（フレーム）に分割し，FFT（高速フーリエ変換）を行い，短時間スペクトル分析を行う．この分析条件は以下の通りである．

フレームサイズ	1152 サンプル（約 26msec）
FFT サイズ	2048 サンプル
分析窓関数	ハミング窓
フレームシフト幅	1152 サンプル（約 26msec）

得られた短時間スペクトルを，メル周波数軸上でフィルタバンク分析する．フィルタバンクの数は，33 個とする．従来の MFCC を利用する研究 [6][7] では，サンプリング周波数を 16000Hz とし，0Hz から 8000Hz の間にフィルタバンク数を 24 個配置して分析を行っている．そこで本システムでは，22050Hz までに 33 個のフィルタバンクを配置すると，8000Hz 程度までで 24 個が配置されるようになるので，フィルタバンク数を上記のように決定した．

また本提案システムでは，人間の音声を対象としているので，人間の音声の周波数帯域である約 200Hz〜約 3500Hz にある信号だけを利用すればよい．そこで，使用する周波数帯域を以下のようにして制限する．

制限 1 ：上限の制限 33 個のフィルタバンクの内，下位の周波数帯域から 16 個だけを使う．

下位から 16 個のフィルタバンクで，0Hz〜約 3506Hz の周波数帯域を覆っている．

制限 2 ：下限の制限 $k = 0, ...11$ までのスペクトルチャネルの振幅スペクトル $|S(k)|$ の値に 0 を代入する．

$k = 0, ...11$ までのスペクトルチャネルで，0Hz〜約 237Hz の周波数帯域を覆っている．

以上の制限を加えた合計 16 個のフィルタバンクにおけるパワー $m(l)$ $(l = 1, ..., 16)$ の系列に対して，対数をとって離散コサイン変換を行い，MFCC を得る．

音源の性質を取り除くため，この MFCC の低次項だけを使用する．本提案システムでは，MFCC の低次項の次元数を，前もって行った予備実験の知見から 5 次元とした．

4.2.2 音声の時系列特徴量ベクトルの生成

3 章で提案した本研究の提案手法に従い，こうして得られた 5 次元 MFCC を時系列上に並べて，時間窓を少しずつずらしながら音声の時系列特徴量抽出して，時系列特徴量ベクトルを生成していく．

本提案システムでは，前もって行った予備実験の知見より，以下の条件で時系列特徴量ベクトルをとる．

時間窓長	118,126,134,142,150, 158,166,174,182 フレーム
時間窓シフト幅	2 フレーム
時系列特徴量ベクトルの線形圧縮比	1/6

このときの基準時間窓長は，150 フレームである．

9 種類の長さの時間窓を用いて，時系列上に並んだ 5 次元 MFCC のデータを抽出する．時系列特徴量ベクトルの次元数は，

$$5\,(\text{MFCC 次元数}) \times 150\,(\text{基準時間窓長}) \times \frac{1}{6}\ (\text{線形圧縮比}) \quad = 125\,(\text{次元数})$$

となる．この 125 次元の時系列特徴量ベクトル間の距離が，音声の類似度を表す．

4.2.3 多次元空間インデックスの実装

本提案システムでは，多次元空間インデックスを実装しているデータベースエンジンである Lite Object ver.7 を使用する．このエンジンの多次元空間インデックスは，VAMsplit R-Tree を用いている．

今回用いるデータの次元数は 125 次元であり，このままでは次元数が大きすぎて Tree を使う効果が得られにくい．そこで，本提案システムでは 5 次元の Tree を 25 個構築して個々の Tree の検索結果を統合し，最終的な検索結果を得る事とする．なお，事前の予備実験より，次元を分割して Tree を構築しても，ほとんど検索結果は変わらないという知見が得られている．

5 有効性を確認するための実験

本提案システムの有効性を確認するため，2 種類の実験を行う．

まず実験 1 で，本提案システムが実装している線形伸縮に対応したマッチングが，線形伸縮に対応しないマッチングに比べて，歌声の検索において有効であるかどうかを確認する．

つぎに実験 2 で，非線形伸縮マッチングを実装してる従来システムと比較しても，本提案システムが歌声の検索において十分に有効であることを確認する．

5.1 実験条件

両実験に共通するの実験条件を以下に示す．

本実験は，映像から抽出した wave ファイルではなく，映像なしで録音した wave ファイル [サンプリング周波数 44100Hz，量子化ビット数 16bit，1ch] を用いることとする．

被験者（合計 15 人）を 3 つのグループ（女性，男性，混合）にわける．それぞれのグループに 62 曲の歌名リストを渡す．歌名リストの中で，その歌を知っている人にフレーズの一部（約 10 秒程度）を歌ってもらい，62 × 3 個の歌声（合計約 30 分）をデータベースに格納す

る．１つの被験者グループの歌声の中から任意に 12 曲選び，そこから１フレーズ程度（基準時間窓長：150 フレーム分，約４秒）を取り出して検索キーとする．そして，他の２グループの被験者の歌声の同一フレーズ部分を適合結果として検索する．

なお，本実験では，検索結果の適合性を判断する順位を 20 位までとし，それ以下の順位に適合結果があっても検索できなかったものとする．

歌声の検索が可能であるかを確認するための評価基準として，平均探索長の平均 [8]（付録 B 参照）を用いる．本実験では，３グループのうちの１つの歌声を検索キーとし，他の２つのグループの歌声を検索するので，適合結果の総数は 2 となる．

5.2 実験 1

線形伸縮に対応しないシステムとして，1 種類の長さの時間窓だけを使って時系列特徴量を抽出するシステムを用意する．一方，線形伸縮に対応するシステムとして，9 種類の時間窓を使って時系列特徴量を抽出する本提案システムを用意する．この両システムの平均探索長の平均の値を比較する．

5.3 実験 1 の結果

実験を行った結果を以下に示す．平均探索長の平均を表 1 で比較する．なお，表 1 中の×は，システムが 2 つの適合結果を検索できなかったことを示すものとする．

	平均探索長の平均	
歌名	線形伸縮なし	提案手法
歌 A(LOVE LOVE LOVE)	×	1
歌 B(青いイナズマ)	1	1
歌 C(DEPARTURES)	×	1.5
歌 D(これが私の生きる道)	1.75	1
歌 E(チェリー)	1	1
歌 F(島唄)	×	4.25
歌 G(津軽海峡冬景色)	4.25	2.5
歌 H(硝子の少年時代)	1	1
歌 I(flower)	4	2
歌 J(大切なあなた)	×	1
歌 K(今すぐ KissMe)	×	4
歌 E(ペッパー警部)	1.25	1

表 1: 線形伸縮に対応する方式としない方式の比較

平均探索長の平均は，歌 B，D，H で同じ値となるものの，それ以外の歌ではすべて本提案システムのほうが上回っている．よって，本提案システムの線形伸縮に対応する方式は，歌声の検索において有効であると言える．

5.4 実験 2

非線形伸縮に対応できる従来システムとして，メディアドライブ株式会社の CrossMediator for Video V.2.0(R1)[4] のボイス検索機能を用いる．この従来システムと本提案システムの平均探索長の平均を比較する．

また，単純なマッチングにより検索時間を削減できているかも確認する．検索時間は，表示部上の検索を開始するためのボタンを押した後から検索結果が表示されるまでの時間を手動で 10 回計測し，その平均値を検索時間とする．なお今回実験に使用したのは，CPU が Pentium 4(1.7GHz)，主記憶容量が 654,812KB の PC である．

この従来システムでは，IPM[5] というネットワークを構築することで検索対象データの圧縮を図っている．これにより，検索対象のデータに，非線形伸縮マッチングを総なめにおこなうよりも速い検索が可能になっている．また，このシステムでは，あいまい～厳密の範囲で検索の精度を指定できるが，この実験ではその中間を指定しておく．

5.5 実験 2 の結果

実験を行った結果を以下に示す．まず平均探索長の平均を表 2 で比較する．なお，表 2 中の×は，実験 1 と同様に，システムが 2 つの適合結果を検索できなかったことを示すものとする．

	平均探索長の平均	
歌名	従来手法	提案手法
歌 A(LOVE LOVE LOVE)	1	1
歌 B(青いイナズマ)	1	1
歌 C(DEPARTURES)	1	1.5
歌 D(これが私の生きる道)	1	1
歌 E(チェリー)	×	1
歌 F(島唄)	×	4.25
歌 G(津軽海峡冬景色)	1	2.5
歌 H(ガラスの少年時代)	1.75	1
歌 I(flower)	×	2
歌 J(大切なあなた)	2.5	1
歌 K(今すぐ KissMe)	×	4
歌 E(ペッパー警部)	×	1

表 2: 非線形伸縮と線形伸縮の比較

表 2 より，歌C，G では，本提案手法の結果が若干下回っているものの，その他すべての結果では同等以上の

結果が得られている．非線形マッチングを使わない本提案システムで，従来システムと同等以上に検索できており，3.1 節の仮定が妥当であったと考えられる．

次に，検索時間の比較を表 3 に示す．検索キーとして表 2 中の歌 B（青いイナズマ）を用い，10 回検索を繰り返し，その平均を検索時間とした．なお，本提案システムでは，表示部分を除いた検索時間をシステム内部の時計を用いて計測することが可能であるので，検索時間も記しておく．

従来システム 平均検索時間	提案システム 平均検索時間
4.49 sec	2.42 sec (結果表示部を含まない : 1.74 sec)

表 3: 検索時間の比較

表 3 より，検索時間において本提案システムの方が 2.07 秒ほど速い．よって，単純なマッチングにより，検索時間を短縮できることが確認できる．

6　実際の歌番組を対象とした実験

つぎに，実際に雑音や BGM を含むような歌番組を対象に，本提案システムを適用して実験を行い，その有効性を確かめた．

検索対象の映像データとして，約 1 時間の歌番組を利用した．この番組中で歌われている歌を，被験者（男女 1 名ずつ）に合計 9 曲を歌ってもらい，その歌声の一部分を検索キーとした．

本実験では，歌番組の映像から歌のシーンを検索しようとしたときに，同じ歌のシーンが複数個はない．よって，1 シーンが検索できれば十分であるので，適合結果の総数を 1 として，評価基準として平均探索長を用いることにする．

しかし，この検索対象データを本提案システムに適用しようとすると，今回実験で用いた PC（主記憶容量，654,812KB）では，インデックスとデータが主記憶に乗り切らなかった．そこで，9 種類の時間窓を使わず，150 フレームの基準時間窓だけで検索対象データから特徴量を抽出することとし，検索キーの時間窓を線形伸縮させて実験をおこなうこととする．実験 1 の条件の場合と比べて，1 種類の時間窓しか使わないため，検索対象データ数が少なくなる．その結果，検索時間が速くなることが予想される．しかし，平均探索長については，検索対象と検索キーのどちらを伸縮させるかが変わっただけなので，9 種類の時間窓を使った場合と変わらないと予想できる．そこで本実験では，平均探索長だけを従来手法と比較することとする．なお，従来手法は，5.4 節と同じ手法（CrossMediator）を用いる．

使用する PC は，5 章の実験と同じであり，検索結果の適合性の判断も同様に 20 位までとする．

6.1　歌番組映像を対象とした実験の結果

実験を行った結果を以下に示す．平均探索長を表 4 で比較する．なお，表 4 中の×は，システムが適合結果を検索できなかったことを示すものとする．

	平均探索長	
歌名	従来手法	提案手法
歌 A(春よ来い)	6.25	1
歌 B(You Go Your Way)	×	1
歌 C(Innocent World)	×	2
歌 D(DAYONE)	×	×
歌 E(Secret Base)	×	×
歌 F(瞳そらさないで)	×	2
歌 G(夢追い虫)	×	×
歌 H(シングルベッド)	×	2
歌 I(愛しさと切なさと心強さと)	2	4

表 4: 平均探索長の比較

表 4 より，本提案システムで検索できている場合があることが確認できる．しかし，雑音がない場合に比べて，従来システムでも，本提案システムでも，検索ができなくなることも多くなっている．なお，従来システムで同一ファイル内を検索する場合に，類似度は 5 段階だけの表示となる．よって平均探索長が大きくなってしまう傾向があり，結果の数値が大きくなっていると考えられる．

7　考察と今後の検討課題

今回の実験では本提案システムで歌声による検索が実際に可能であるかという確認をおこなった．しかし，検索結果の精度という点ではまだまだ十分でないと考えられ，精度向上のための検討が必要であると考えられる．また，単純なマッチングで高速な検索ができるという利点を活かすため，今回の実験よりも大規模な映像データベースに適用して実験をおこなわなくてはならない．しかし，現状では，特徴量データの量とインデックスが大きいという問題がある．そこで特徴量データ量の削減と，インデックス方式の工夫が必要であると考えられる．

8　まとめ

本稿では，歌の任意の一部分を歌った歌声を検索キーとして，蓄積された大量の映像の中から検索キーと同じ

部分が歌われているシーンを検索するシステムについて提案し，そのプロトタイプシステムを作成した．本稿で作成したシステムを適用することにより，見たいシーンが映像中のどこにあるかをすぐに知ることができ，そしてそのシーンを見ることができる．

参考文献

[1] 西崎博光, 中川聖一: “音声入力によるニュース音声検索システム”: 信学技報, SP99-108, 1999

[2] 鷹尾誠一, 緒方淳, 有木康雄: “ニュース音声に対する検索方法の比較”: 情処研究報告「音声言語情報処理」, No.029-017,1999

[3] Witbrock, M., Hauptmann, A.: “Speech Recognition for a Digital Video Library”: Journal of the American Society for Information Science (JASIS) 49(7), , 1998: http://www.informedia.cs.cmu.edu/

[4] 技術研究組合新情報処理開発機構（RWCP），情報ベース機能つくば研究室: “CrossMediator”: http://www.rwcp.or.jp/lab/mmtl/crossmedia.html

[5] 遠藤隆, 高橋裕信, 豊浦潤, 向井理朗, 岡隆一: “動画像の自己組織化ネットワークによるモデル化とその動的特徴の可視化-Video Intra-structure Visualization-” 信学技報, PRMU97-78, Jul, 1997

[6] 鹿野清宏他、編著: “IT Text 音声認識システム”: オーム社,2001

[7] 山口義和他: “音声認識エンジン VoiceRex によるニュース放送音声認識” 日本音響学会公演論文集, 2-1-2, pp.93-94, 1999

[8] 徳永健伸: “言語と計算 5　情報検索と言語処理”: 東京大学出版, 1999

付録

A　MFCC の算出方法

この MFCC を算出するには，まず音声信号を短時間区間の分析窓（フレーム）に分割して，FFT を用いた短時間スペクトル分析を行い，各スペクトルチャネルの振幅スペクトル $|S(k)|$（$k = 0, ..., N-1$，N は FFT サイズの半分）を算出する．そしてメル周波数[1] 軸上でのフィルタバンク分析を行う．このフィルタバンク分析は，メル周波数軸上で等間隔に L 個の三角窓（フィルタバンク）を配置し（図 9 参照），窓の幅に対応する周波数帯域の信号のパワー $m(l)(l = 1, ..., L)$ を計算するものである．このパワー $m(l)$ は，式 (1)，(2) により，単一スペクトルチャネルの振幅スペクトル $|S(k)|$ の，三角窓での重み $W(k;l)$ による重み付け和で計算する．

$$m(l) = \sum_{k=lo}^{hi} W(k;l)|S'(k)| \tag{1}$$

$$W(k;l) = \begin{cases} \frac{k-k_{lo}(l)}{k_c(l)-k_{lo}(l)} & \{k_{lo}(l) \le k \le k_c(l)\} \\ \frac{k_{hi}(l)-k}{k_{hi}(l)-k_c(l)} & \{k_c(l) \le k \le k_{hi}(l)\} \end{cases} \tag{2}$$

ただし，$k_{lo}(l)$，$k_c(l)$，$k_{hi}(l)$ は，それぞれ l 番目のフィルタバンクの下限，中心，上限のスペクトルチャネル番号であり，隣り合うフィルタバンク間で，

$$h_c(l) = k_{hi}(l-1) = k_{lo}(l+1) \tag{3}$$

の関係となる．また，$h_c(l)$ はメル周波数軸上で等間隔になるように置かれる．

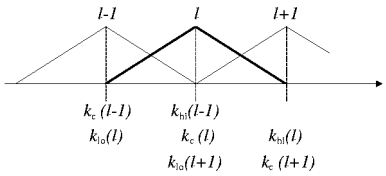

図 9: フィルタバンク

最後に，このフィルタバンク分析により得られた L 個のフィルタバンクにおけるパワー $m(l)$（$l = 1, ..., L$）の系列を，式 (4) により，対数をとってから離散コサイン変換することで MFCC : $C_{mfcc}(i)$ （$i = 1, ..., L$）が得られる．

$$C_{mfcc}(i) = \sqrt{\frac{2}{N}} \sum_{l=1}^{L} \log m(l) \cos\left\{\left(l - \frac{1}{2}\right)\frac{i\pi}{L}\right\} \tag{4}$$

B　平均探索長の平均

まず先に，平均探索長について説明する．

平均探索長は，検索結果として順序付けられた集合を評価する尺度である．検索結果として順位付けられた結果が返ってきた場合，実際には検索者は，検索結果の適合性を上位の結果から逐一判断していかなければいけない．平均探索長は，このような検索者の適合性判断の過程を考慮し，検索者が必要な数の適合結果を得るためには，どれだけ結果の適合性を判断しなければならないかというユーザーの手間を計測する尺度である．

[1] メル周波数 $Mel(f)$ は，音の高低に対する人間の聴覚の感覚尺度で，

$$Mel(f) = 2595 \log_{10}\left(1 + \frac{f}{700}\right)$$

により計算される．ただし，f は周波数 $[Hz]$ を表す．

例えば，検索結果が図 10 のように順序付き集合[2]S_1，S_2，S_3 にわけることができたとする．ただし，集合間の順序は S_1，S_2，S_3 の順であり，○，×はそれぞれ適合結果，不適合結果を表す．

S_1： {○，×，×，×}
S_2： {○，○，○，○，×，×}
S_3： {○，○，×，×}

図 10: 平均探索長の例

今検索者が，適合結果を 1 つ得たいとする．まず，集合 S_1 を検査することになる．この集合の中では順序が付いていないので，適合結果を見つけるまでに検査しなければいけない結果の個数の期待値は，

$$1 \times \frac{1}{4} + 2 \times \frac{1}{4} + 3 \times \frac{1}{4} + 4 \times \frac{1}{4} = 2.5$$

となる．これは検索結果から適合結果をひとつ見つけるためには，検索者は平均的に 2.5 個の検索結果の適合性を判断しなければならないことを示している。つまり，この検索結果から 1 つの適合結果を見つけ出すのに必要な平均探索長は，2.5 個である．

また，適合結果を 2 つ見つけるためには，集合 S_1 を全部検査した後，集合 S_2 から 1 つ見つければよいから，検査すべき結果の個数の期待値は，

$$(4+1) \times \frac{4}{6} + (4+2) \times \frac{4}{15} + (4+3) \times \frac{1}{15} = 5.4$$

となる．つまり，2 つの適合結果を見つけ出すのに必要な平均探索長は，5.4 個である．

上記の例では，検索結果が順位付けられた集合で与えられている場合であったが，検索結果の個々に全順序が付けられている場合でも，各集合の要素を 1 つと考えれば平均探索長を計算できる．

上述内容からもわかるように，平均探索長はひとつの尺度とはならず，必要な適合結果の個数に依存した値となる．そこで，平均探索長の平均として，必要な適合結果 1 つあたり平均探索長の値を計算する．

必要な適合結果数を i，総適合結果数を M，i 個の必要な適合結果を見つけるの必要な平均探索長を $l(i)$ とすると，平均探索長の平均 $\bar{l}$ は，

$$\bar{l} = \frac{1}{M} \sum_{i=1}^{M} \frac{l(i)}{i} \tag{5}$$

で表される．

例えば，検索の結果，適合した結果が 2 位と 6 位に検索された場合を考える．必要な適合結果の個数を 1 とした場合，平均探索長は 2 となり，必要な検索結果の個数を 2 とした場合，平均探索長は 6 となる．これらの平均探索長の平均は，$(2/1 + 6/2)/2 = 4$ となる．

[2]例えば，CrossMediator のように類似度を 5 段階評価で表す場合では，1 位の集合が 5 の評価が付けられた検索結果の集合であり，2 位の集合が 4 の評価が付けられた検索結果の集合であるというように考えられる．