卒業論文

メロディライン認識による
楽曲検索システムに関する研究

東北大学工学部通信工学科

武村 知昭

目 次

図目次

表 目 次

第1章

序論

1.1 研究の背景

従来、データベースといえば文書情報のみであったが、最近は文書以外の情報についてもマルチメディアデータベースなどとして蓄積されるものが急増しており、その規模も大きくなりつつある。また WWW 上にもホームページ制作者のオリジナル曲が多数アップロードされているなど、広い意味での大規模音楽データベースが構築されつつあるといえる。しかし、それらの検索においては楽曲名、演奏者など詳しい情報を知っておかなければならないという現状がある。しかし、楽曲を聞いたもののその楽曲の詳しい情報を知らないということも少なくなく、ユーザーの多様な検索要求に答えることができるほど十分でないのが現状である。

そこで、題名など詳しい情報がない楽曲からメロディパートを抽出し、それをもとに検索できれば、さらに検索が容易である。また、人の耳に残りやすいメロディパートをハミング入力することで検索できればさらに検索が容易であると考えられる。

1.2 楽曲からのメロディパート抽出の現状

最近のポップスに代表されるように、演奏や作曲の音素材の生成など音楽を作り出すことに計算機は利用されている（ミキシング）。しかし反対に演奏された音楽を計算機に入力して、採譜や検索などのような便利な機能を開発しようとしても、今までのところほとんど成功していないのが現状である。その最大の壁は、音の重なり合いといわれている。

人間は歌謡曲を聞けば伴奏があっても主旋律を聞き分けることができる。たとえばピアノとフルートのアンサンブル演奏を聞けばそれがピアノとフルートの音であると言い当てることができる。しかし現在のところ、計算機は伴奏と主旋律というように重なり合って

いる音があったとき、伴奏も主旋律も一緒のものとして認識してしまうのである。このような理由から、現在のところ音源分離は非常に難しいものとされている [1]。音源分離が困難な理由として、以下のことがあげられる。

1.2.1 基本周波数の抽出

音源分離が困難な理由として基本周波数の抽出の難しさがあり、大きく三つに分けられる。まず一点目は、音楽演奏では異なる音源からの音であっても高次の周波数成分は重複するのがほとんどであり、両方の音を抽出するのが難しいという点である。二点目は楽器音の特徴の変動が大きいという点である。例えば自然楽器の実演奏では楽器の個体差、奏法、収録条件等によってスペクトルパターンが大きく変動するため精度良く基本周波数を推定することは難しいのである。三点目は楽音が音声に比べて音域が広いという点である。音声では基本周波数の抽出において正しい周波数の整数倍、あるいは整数分の一の周波数を抽出してしまうという誤りであれば後処理でほぼ修正可能だが、楽音では音域が広く同様の処理はほぼ不可能であるからである [2]。

1.2.2 音源同定

また音源分離が困難な理由として音源同定の難しさがあげられる。音源同定とはある音が何の楽器の音であるのかを認識することを言う。例えば図 1.1 に示すように、ピアノ音は鍵盤をたたいた直後から振幅が増加していくが基音と倍音では立ち上がり方が一様にならない。この経過が音色を決める重要な要素となる。各種の楽器が出す音から、鳴り始めと終りを除いた定常部分だけを聞いても音源の同定は難しいとされている。

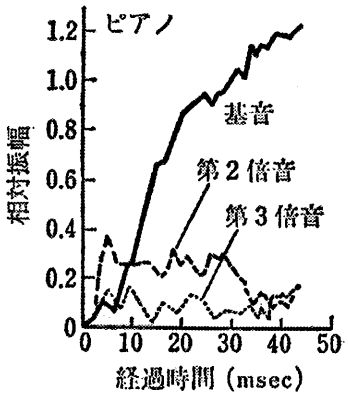

図 1.1: ピアノＣ音の立ち上がり経過

1.3 従来のハミング入力による楽曲検索システム

ハミング入力による楽曲検索システムはすでに参考文献 [3] で提案されている。その検索処理の流れを図1.2に示す。

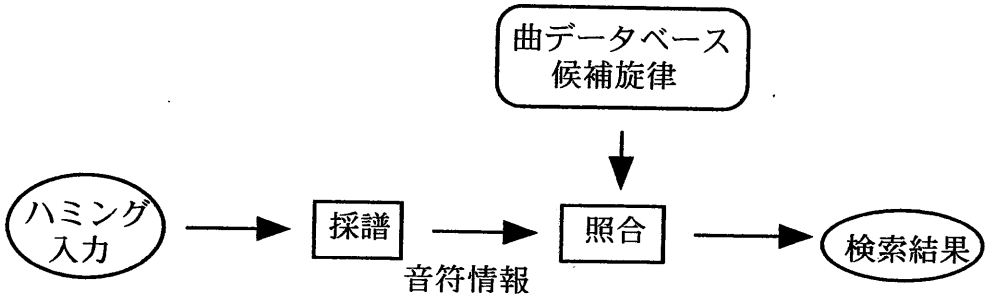

図 1.2: ハミング検索処理

検索システムの構成は大きく分けて、採譜部、データベース構築部、照合部の3点から成る。

- 採譜部

  ここでは旋律照合の前処理として、利用者が入力したハミング歌唱をハミングの音響信号から、歌唱に伴う声の揺らぎなどの検索には不要な情報を除き、音符列で表される旋律表現に変換する。

- 楽曲データベース部

  楽曲データベースには、検索対象となるすべての楽曲の候補旋律情報、および書誌情報が格納されている。候補旋律は対象楽曲の主旋律の一部であり、楽曲データベース作成時にあらかじめ定めている。個々の曲の候補旋律情報は音符列として格納されている。

  参考文献 [4] は [3] を参照にして書かれたものであるが、こちらのデータベースにはハミングでデータベースを作り、各音符を相対音高差、相対音長比の系列として格納されている。

- 照合部

  照合部では、ハミング入力の旋律と楽曲データベースにあらかじめ登録されているすべての候補旋律との間の照合を行い、ハミング入力の旋律とすべての候補旋律との間の類似度を求める。検索結果は、手がかり旋律との旋律間類似度が最大となった候補旋律をもつ楽曲とする。

従来法では、データベースの構築が非常に困難である。また検索精度も7,8割であまり良いとはいえないので改善の余地があるといえる。

1.3.1 研究の目的

本研究ではこうした現状をふまえ、まず題名など詳しい情報がわからない楽曲がある時に、その楽曲からメロディラインのみを抽出してその楽曲を検索するシステムの検討を行う。

次に、ハミング入力による楽曲検索システムはすでに提案されているが、従来法の検索システムのデータベース構築は困難であり、また検索精度は不十分である。これより、新しい楽曲データベース構築法と、ハミング入力からの特徴量抽出法の提案を行う。これを用いて新しいハミング入力による楽曲検索システムを構築し検索精度の向上を目的とする。

1.4 本論文の構成

本論文の構成は次の通りである。

第1章 序論

研究の背景、目的を述べた。

第2章 一次元パターンの照合法

ハミング入力の音高差系列と楽曲の音高差系列の照合の際に用いるDPマッチング（動的計画法）について述べる。

第3章 音源分離によるメロディパート抽出法の検討

ある手持ち楽曲からメロディライン（vocal部）のみを抽出する方法を提案する。

第4章 新しいハミング入力による楽曲検索システムの提案

前章の音源分離が困難ということから、入力をハミング入力とし、その特徴量からデータベースと照合するシステムの提案を行う。

第5章 結論

本研究の成果、今後の課題について述べる。

第2章

一次元パターンの照合法

2.1 はじめに

本章では、ハミング入力による楽曲検索システムで、ハミング入力の音高差系列と楽曲データベースの音高差系列の類似度の計算の際に用いるDP マッチングについて述べる。DPマッチングとはハミング入力の音高差系列と楽曲データベースのどの部分とマッチしているかを見いだすこと、つまり整列化を行いそれに基づいてパターン間の類似度を計算するものであり、音声や文字のパターン認識だけでなく非常に広い分野で用いられている。

2.2 DPマッチング

2.2.1 一次元パターンの類似度

ハミング入力の音高差系列も、楽曲データベースの音高差系列も数字が一列に並んで構成されているパターンである。そして、数字はこのパターンの基本的な構成要素となっている。このように構成要素の集合があり、その要素が一列に並んでできた有限長のパターンを、1次元パターンと呼ぶ。

一次元パターンの構成要素の集合をVで表すと、その写像

$$d : V \times V \to [0, \infty)$$

が与えられており、これがVの要素の類似度を表す。すなわちdは

$$x \text{が} y \text{に似ている} \Leftrightarrow d(x, y) \text{の値が小さい}$$

という性質を持っており、特に

$$x = y \Rightarrow d(x, y) = 0$$

が成り立っている。この写像 d をもとにして、V の要素から構成される1次元パターン間の類似度を計る写像

$$D_0 : V^* \times V^* \rightarrow [0, \infty)$$

を定める。ここで、V^* は V の要素から構成される有限長の1次元パターンの全体である。D_0 は類似度なので

$$\begin{aligned} A &= a_0 a_1 \cdots a_I \in V^* \\ B &= b_0 b_1 \cdots b_J \in V^* \end{aligned}$$

に対して

・$D_0(A, B) \geq 0$

・A が B に似ている $\Leftrightarrow$ $D_0(A, B)$ の値が小さい

という性質を持っていて欲しいのであるが、さらに

・パターン A はパターン B を長さ方向に伸縮したパターンである $\Rightarrow$ $D_0(A, B) = 0$

が成り立つものを探すことにする。このような類似度がうまく定義できれば、複雑な伸縮によって見かけが変わってしまったパターンが、元はどのようなパターンに似ていたかを判定することができる。このような類似度を定義するには、1つのパターンのある部分が別のパターンのどの部分に対応するかという、対応付けを考える必要がある。パターン A からパターン B への対応付けは、A の構成要素の位置の集合 $\{0, 1, 2, \cdots, I\}$ から B の構成要素の位置の集合 $\{0, 1, 2, \cdots, J\}$ への写像

$$w : \{0, 1, 2, \cdots, I\} \rightarrow \{0, 1, 2, \cdots, J\}$$

を与えることによって定まる。ただし w は次のような性質を持つものとする。

・w は A と B の両端点を一致させる写像である：$w(0) = 0, w(I) = J$

・w はパターンの構成要素が現れる順序を逆転させない

このような w を伸縮写像と呼ぶことにする。そこで w のもとでの A, B 間の類似度 $D_0(A, B; w)$ を、w によって対応付けられた要素の間の類似度の和をとって

$$D_0(A, B; w) \equiv \sum_{i=0}^{I} d(a_i, b_{w(i)})$$

と定義する。次に

伸縮写像の全体を $W(I, J)$ と表すと

$$W(I,J) \equiv \{w | w : \{0,1,2,\cdots,I\} \to \{0,1,2,\cdots,J\}$$
$$w(0) = 0, w(I) = J$$
$$\forall i,j \in \{0,1,2,\cdots,I\} : (i < j \Rightarrow w(i) \leq w(j))\}$$

そして、伸縮写像 w が $W(I,J)$ の要素のうち $D(A,B;w)$ の最小値

$$\min\{D_0(A,B;w) | w \in W(I,J)\}$$

について考えてみる。$w \in W(I,J)$ に対してパターン

$$B'(w) = b'_0 b'_1 \cdots b'_I = b_{w(0)} b_{w(1)} \cdots b_{w(I)}$$

はパターン B を伸縮したものである。そして

$$\begin{aligned} D_0(B'(w),B;w) &= \sum_{i=0}^{I} d(b'_i, b_{w(i)}) \\ &= \sum_{i=0}^{I} d(b_{w(i)}, b_{w(i)}) \\ &= 0 \end{aligned}$$

が成り立つ。したがって、ある $w_0 \in W(I,J)$ があって

$$A = B'(w_0)$$

つまり、パターン A がパターン B を w_0 によって伸縮したパターンならば

$$\min\{D_0(A,B;w) | w \in W(I,J)\} = \min\{D_0(B'(w_0),B;w) | w \in W(I,J)\} = 0$$

が成り立つ。このことから、$D_0(A,B)$ を

$$D_0(A,B) \equiv \min\{D_0(A,B;w) | w \in W(I,J)\} \tag{2.1}$$

と定義すれば、少なくとも $W(I,J)$ に属する伸縮写像で表されるような伸縮の範囲内では、要請を満たす類似度であることがわかる。

2.2.2 DPマッチング

式(2.1)を定義通り計算しようとすると、$D_0(A,B)$を$|W(I,J)|$回計算しなくてはならない。これでは、時間がかかりすぎてしまう。そこで、この種の最小化問題をもっと少ない計算量で解くアルゴリズムが知られている。写像

$$w : \{0,1,2,\cdots,i\} \rightarrow \{0,1,2,\cdots,j\} \tag{2.2}$$

は整数の組$(w(0),w(1),\cdots,w(i))$で

$$\forall m \in \{0,1,2,\cdots,i\} : 0 \leq w(m) \leq j \tag{2.3}$$

を満たすものと考えられる。したがって、式(2.2)の写像と式(2.3)を満たす整数の組を同一視すれば

$$W(I,J) = \{w(0),w(1),\cdots,w(I))|0 = w(0) \leq w(1) \leq \cdots \leq w(I) = J\}$$

と表すことができる。そこで、整数$0 \leq i \leq I, 0 \leq j \leq J$に対して

$$W(i,j) \equiv \{w(0),w(1),\cdots,w(I)|w(0),w(1),\cdots w(i)\text{は整数},\\ 0 = w(0) \leq w(1) \leq \cdots w(i) = j\}$$

とし

$$g_0(i,j;w) \equiv \sum_{k=0}^{i} d(a_k, b_{w(k)}), (w \in W(i,j)), \tag{2.4}$$

$$g_0(i,j) \equiv \min\{g_0(i,j;w)|w \in W(i,j)\} \tag{2.5}$$

とおく。そうすると、$W(0,0) = \{w(0)\}$より

$$g_0(0,0) = d(a_0,b_0)$$

が成り立つ。また、

$W(1,j) = \{(0,j)\}$なので、式(2.4),(2.5)から$0 \leq j \leq J$に対して

$$g_0(1,j) = d(a_0,b_0) + d(a_1,b_1) = g_0(0,0) + d(a_1,b_j)$$

が得られる。さらに、$2 \leq i$のときは、$W(i,j)$が

$$W(i,j) = \cup_{k=0}^{j} W_k(i,j)$$

$$W_k(i,j) \equiv \{u(0),\cdots,w(i-j),j)|w(0),\cdots w(i-1)) \in W(i-1,k)\}$$

と表されることと、一般に有限集合Sが、その部分集合の和集合として

$$S = S_1 \cup \cdots \cup S_M$$

と表されるとき、S上の実数値関数fに対して

$$\min\{f(x)|x \in S\} = \min\{\min\{f(x)|x \in S_1)\}, \cdots, \min\{f(x)|x \in S_M\}\}$$

が成り立つという事実によって

$$\begin{aligned} g_0(i,j) &= \min\{g_0(i,j;w)|w \in W(i,j)\} \\ &= \min\{\min\{g_0(i,j;w)|w \in W_0(i,j)\}, \\ &\qquad \min\{g_0(i,j;w)|w \in W_1(i,j)\}, \\ &\qquad \cdots \\ &\qquad \min\{g_0(i,j;w)|w \in W_j(i,j)\}\} \end{aligned}$$

となる。ところが

$$\begin{aligned} &\min\{g_0(i,j;w)|w \in W_k(i,j)\} \\ &= \min\{\textstyle\sum_{m=0}^{i} d(a_m, b_{w(m)})|w \in W_k(i,j)\} \\ &= \min\{\textstyle\sum_{m=0}^{i-1} d(a_m, b_{w(m)}) + d(a_i, b_j))|w \in W_k(i-1,j)\} \\ &= \min\{\textstyle\sum_{m=0}^{i-1} d(a_m, b_{w(m)})|w \in W(i-1,k)\} + d(a_i, b_J)\} \\ &= g_0(i-1,k) + d(a_i, b_j) \end{aligned}$$

が成り立つので、$2 \le i \le I, 0 \le j \le J$に対して

$$g_0(i,j) = d(a_i, b_j) + \min\{g_0(i-1,0), g_0(i-1,1), \cdots, g_0(i-1,j)\}$$

が導かれる。また$1 \le i \le I, 0 \le j \le J$に対して

$$g_0(i,j) = d(a_i, b_j) + \min\{g_0(i-1,k)|0 \le k \le j\}$$

と表すことができる。

iを0から始め1ずつ増加させながら、順次

$g_0(0,0,)$

$g_0(1,0), g_0(1,1), \cdots, g_0(1,J),$

$\cdots$

$g_0(I-1,0), g_0(I-1,1), \cdots, g_0(I-1,J),$

$g_0(I,J)$

の値を計算すれば、$D_0(A,B) = g_0(I,J)$を求めることができる。また、i,jごとに$\min\{g_0(i-1,k)|0 \le k \le j)\}$を与える$k$、すなわち$\arg min\{g_0(i-1,k)|0 \le k \le j\}$を記憶しておけば$g_(I,J)$の計算が終わったあとで最適伸縮関数を定めることができる。このように、ある問題を解きたいとき、それと同じタイプでそれよりサイズが小さい一群の問題の解を利用すると、同じ手続きの繰り返しで計算が進む原理を動的計画法という。動的計画法を用いて二つのパターンの要素間の対応付け（整列化）を行い、それによって類似度を計算することをDPマッチングという。

2.2.3 脱落と挿入

これまで述べた考え方によれば、パターン A の一つの構成要素に対してパターン B の2つの構成要素が対応していたが、問題によっては1つの構成要素に複数の構成要素が対応するのは不自然なことも有りうる。この時、$b_{i(k)}$ に対応する A の要素がないと考えなければならない。つまり、要素 $b_{i(k)}$ はパターン B にもともとは存在せず、後で挿入されたものである、あるいはパターン A にもともと存在した $b_{i(k)}$ に対応する要素が脱落したと考えることにする。このような状況は、脱落を表す記号 $*$ を導入することにより、次のように表される。

各パターンの要素に今度は1から始まる番号を付けて、

$A = a_1 a_2 \cdots a_I$

$B = b_1 B_2 \cdots b_J$

とする。そして格子点の集合 $\{p_1, \cdots, p_k\}$ から $V \cup \{*\}$ への二つの写像 α, β を

$$\alpha(p_k) \equiv \begin{cases} a_i(k) & (i(k) = i(k-1) + 1 \text{のとき}) \\ * & (i(k) = i(k-1) \text{のとき}) \end{cases}$$

$$\beta(p_k) \equiv \begin{cases} b_i(k) & (i(k) = i(k-1) + 1 \text{のとき}) \\ * & (i(k) = i(k-1) \text{のとき}) \end{cases}$$

と定義する。そうすると

$A' = \alpha(p_1)\alpha(p_2)\cdots\alpha(p_k)$

$B' = \beta(p_1)\beta(p_2\cdots\beta(p_k))$

はそれぞれパターン A、B の脱落個所に $*$ が挿入されたパターンになる。このように脱落、挿入を考慮したパターン間の類似度を定義する必要がある。格子点 (0,0) から格子点 (I,J) への経路

$$\tilde{p} = (p_0, p_1, \cdots, p_k) \in P((0,0),(I,J))$$

のもとでの A、B 間の類似度 $d_3(A, B; \tilde{p})$ を

$$D_3(A, B; \tilde{p}) \equiv \sum_{k=1}^{K} d(\alpha(p_k), \beta(p_k))$$

と定義しておき、A、B 間の類似度 $D_3(A, B)$ を

$$D_3(A, B) \equiv \min\{D_3(A, B; \tilde{p}) | \tilde{p} \in P((0,0),(I,J))\}$$

と定める。この類似度も動的計画法により効率よく計算することができる。まず (0,0) から (i,j) への経路

$$\tilde{p} = (p_0, p_1 \cdots, p_k) \in P((0,0),(i,j))$$

に対して

$$g_3(i,j;\tilde{p}) \equiv \sum_{k=1}^{K} d(\alpha(p_k),\beta(p_k))$$

とおく。そして、$g_3(0,0) \equiv 0,(i,j) \neq (0,0)$ に対して

$$g_3(i,j) \equiv \min\{g_3(i,j;\tilde{p}|tildep \in P((0,0),(i,j)))\}$$

と定義すると

$$\begin{aligned} g_3(i,0) &= d(a_1,*) + d(a_2,*) + \cdots + d(a_i,*) \quad (1 \leq i \leq I) \\ g_3(0,j) &= d(*,b_1) + d(*,b_2) + \cdots + d(*,b_j) \quad (1 \leq j \leq J) \\ g_3(I,j) &= D_3(A,B) \end{aligned}$$

の成り立つことがわかる。また、$(1 \leq i \leq I), I(1 \leq j \leq J)$ に対しては

$$\begin{aligned} g_3(i,j) = \min\{&g_3(i-1,j) + d(a_i,*), \\ &g_3(i-1,j-1) + d(a_i,b_j), \\ &g_3(i,j-1) + d(*,b_j)\} \end{aligned}$$

が導かれる。これを用いれば $g_3(I,J) = D_3(A,B)$ を効率よく計算できる。

2.3　まとめ

本章ではハミング入力による楽曲検索システムで、ハミング入力の音高差系列と楽曲データベースの音高差系列の類似度の計算に用いるDPマッチングについて述べた。

第3章

音源分離によるメロディパート抽出法の検討

3.1 はじめに

一般楽曲からメロディパートのみを抽出することは、楽器音の高次の周波数成分が重複してまうため、非常に困難であるということは序論で述べた。本章では、音声認識の分野で音素情報を表すために用いるサウンドスペクトログラムと、基本周波数抽出にケプストラム法を用いて、一般楽曲からメロディパートの抽出を試みる。ここでいう一般楽曲とはメロディパートが歌で入力されているもので、メロディパートの抽出はvocalパートの抽出を指している。

3.2 サウンドスペクトログラムによるメロディパート認識実験

どのくらい音源分離が困難か分析するため、実際に一般楽曲を入力として、サウンドスペクトログラムを用い、楽音のスペクトルを分析した。楽器音によりスペクトルが違うので、vocalパートをサウンドスペクトログラムで特定できるのではないかと考え、実際に実験を行い分析する。

3.2.1 実験条件

入力には、vocal,guitar2本の音から成る楽曲Aを用いる。楽曲Aには「どれくらい」とvocalが入っている。「どれ」はvocalパートのみの音であるが、「くらい」にはvocalの音にguitarが混ざって入っている。

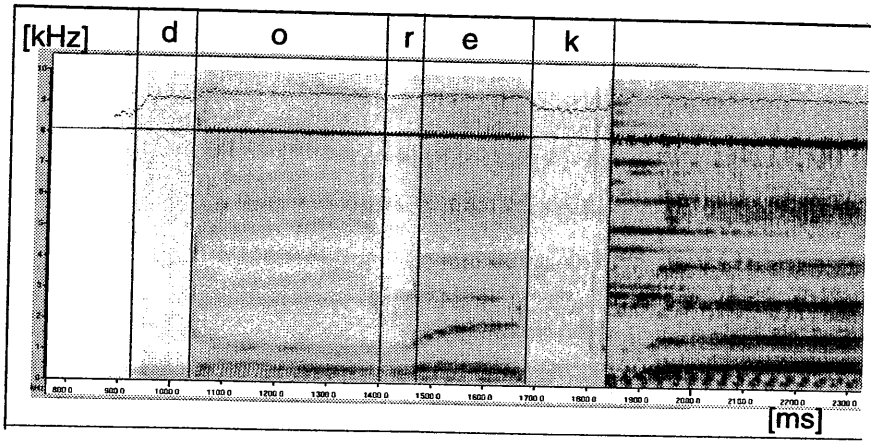

図 3.1: vocal,guitar2 本から成る楽曲 A のサウンド スペクトログラム

3.2.2 実験結果及び考察

図 3.1 に結果を示す。サウンド スペクトログラムの横軸は時間軸であり単位は msec である。横軸は周波数であり単位は kHz である。図中の濃淡はスペクトルの強さであり、濃い方がスペクトルが強いことを表す。図中に見られる縞模様が調波構造であり、基本周波数とその整数倍の周波数成分のスペクトルが濃く現れている。

メロディラインは vocal パートが認識できれば良いわけであるが、認識できたのは vocal のみが入っている/d/o/r/e/k/までであった。guitar が混入してくると、vocal と guitar のスペクトルが混ざり、vocal パートのみを認識することは不可能であった。人の声も楽音に分類される音であり調波構造を持つため、このように音楽が重畳した音声は認識が困難であるということが確認できた。

3.3 基本周波数抽出によるメロディパート認識実験

前節の実験結果からサウンドスペクトログラムによるメロディパートの認識が困難であるということがわかった。これはサウンドスペクトログラムがすべての音を対象としていたためである。ここで音の基本周波数の範囲を指定すればメロディパートが認識できるのではないかと考え、音声認識の分野で用いられているケプストラム法による基本周波数の抽出実験を行う。ここで用いるケプストラム（cepstrum）とは音声認識の分野で用いられている特徴量で、音声波形から認識に有効な特徴量パラメータが抽出され、入力音声は特徴量ベクトルの時系列として表されるものである。このケプストラムから、ゆう度の高い順に10個のピークを抽出し、発話全体の連続性を考慮したDP法を用いてゆう度の高いピークを滑らかにつなぐアルゴリズムを用いている。これによりvocalパートが認識できるのではないかと考え、実際に先ほどの楽曲を用い実験を行う。

3.3.1 実験条件

入力を3.2節の実験で用いた楽曲A（vocal,guitar2本）と楽曲Aのサビの部分（vocal,guitar,bass,drum等）の2つの部分とした。基本周波数抽出条件を表3.1に示す。

表 3.1: 基本周波数抽出条件

サンプリング周波数	20kHz
窓	ハミング窓
フレーム周期	8msec
フレーム幅	32msec
1フレーム抽出ピーク数	10
抽出範囲	50-1000Hz
	50-300Hz
	300-1000Hz

3.3.2 実験結果

まず入力が楽曲A（vocal,guitar2本）についての実験結果を図3.2に示す。横軸は時間で単位はsec、縦軸は周波数で単位はHzである。ここで周波数の抽出範囲を3つにすることで次のように抽出することができた。

- 50-1000Hz

  この抽出範囲で楽曲全体の基本周波数を抽出できた。

- 50-300Hz

  男性の音声の第一フォルマント周波数が50-300Hzなので、この範囲でvocalパートを抽出することができた。

- 300-1000Hz

  この抽出範囲でguitarパートの基本周波数が抽出できた。

図3.2ではパートが3つと少ないため、うまくパートを分離することができた。

次に楽曲Aのサビの部分（vocal,guitar,bass,drum等）ついての実験結果を図3.3に示す。楽曲A（vocal,guitar2本）では、抽出範囲を3つに絞ることで、パートを分離することができたが、今回はパートが複数のため抽出範囲を50-300Hzにしても、うまくvocalパートを特定することができなかった。これは、他のパートで基本周波数が50-300Hzにあるものがあったためであると考えられる。

これより、入力楽曲のパートが少なければ音源の分離は行えるが、複数の場合は困難であることが確認された。しかし、一般楽曲は複数パートから構成されているものが多いため、一般楽曲からのメロディパートの抽出は困難であるものと考えられる。

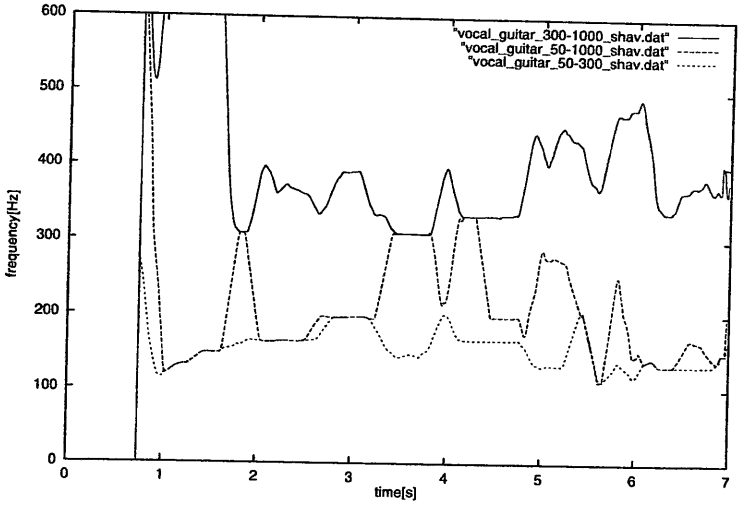

図 3.2: 楽曲 A（vocal,guitar2 本）の基本周波数

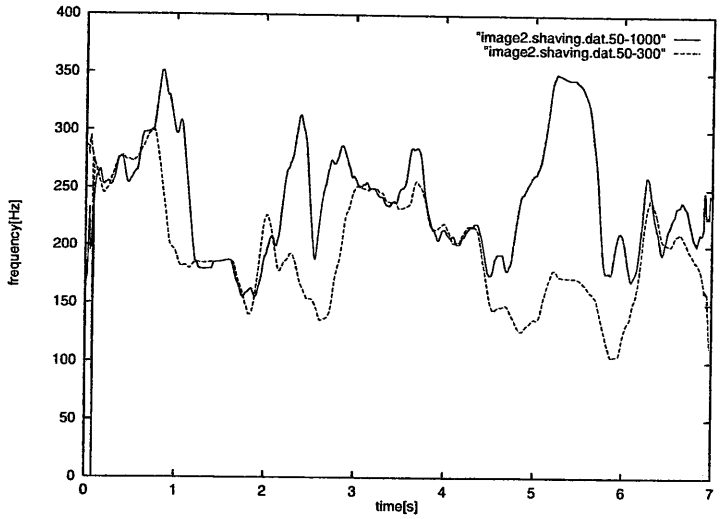

図 3.3: 楽曲 A のサビ（vocal,guitar,bass,drum 等）の基本周波数

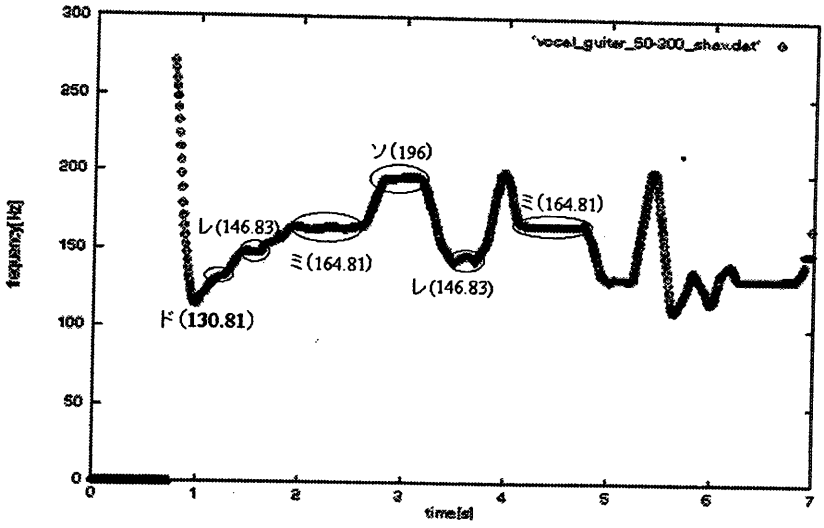

図 3.4: ある楽曲の vocal 部のみの基本周波数

3.3.3 考察

ここで vocal,guitar2 本の楽曲について、抽出範囲 50-300Hz のグラフに着目する。このグラフは vocal パートの基本周波数である。図 3.4 はこのグラフの平行な部分に注目し、その周波数で音符を与えたグラフである。始めの平行な部分の周波数は約 130.81Hz なのでドの音を、次に平行な部分の周波数は約 146.83Hz なのでレの音を、というふうに一番近い周波数を持つ音符を与えていった。このようにして採譜した音符列と実際の楽曲を比較してみると完全に一致しているということがわかった。

この結果より、実際の楽曲を入力としたときは音源の分離が困難であるためにメロディラインは抽出できなかったが、入力を実際に歌った音だけにすればメロディラインを正確に抽出できることがわかった。よって入力をハミングにすることで、その情報から楽曲を検索するシステムを構築することができると考えられる。

3.4 まとめ

本章では実際の楽曲を入力として、音声認識の分野で用いられているサウンドスペルトログラムと、ケプストラム法による基本周波数抽出を実際に行い、音源分離の困難さを実証した。少数パートでメロディラインを歌って入力している楽曲ならば、基本周波数の抽出範囲を絞ることでメロディパートを特定することができるが実際は多数のパートから構成されている楽曲がほとんどなので、入力が一般楽曲ならばメロディパート抽出は困難であると思われる。

しかし少数パートの一般楽曲入力で、基本周波数抽出範囲を男性の第一フォルマント周波数の現れる50-300Hzに絞ったとき、メロディラインを歌っているvocalパートを正しく抽出することができた。さらにその基本周波数抽出のグラフの平行な部分の周波数から音符を与えると、実際の楽曲の音符と合致する結果を得ることができた。これより、入力を一般楽曲ではなくハミングすることとして、メロディラインを特徴量としたシステムを構築することで正確に楽曲を検索できると考えられる。

第4章

ハミング入力による楽曲検索システム

4.1 はじめに

前章では一般楽曲の音源分離がとても困難なことということが実験から確認することができた。しかし入力をvocalパートのみとすれば、基本周波数抽出のプログラムからうまく音符を与えることができることがわかった。そこで本章では入力をハミング入力とし、採譜した音符から楽曲を検索するシステムの構築を目的とする。

序論では従来法のハミング入力による楽曲検索システムについて述べたが、従来法ではデータベースの構築が非常に困難であり、また検索精度も7,8割であった。これではまだ十分とはいえないので、新しくハミング入力による検索システムを構築し検索精度向上を検討する。

4.2 提案する検索システム

今回提案したハミング入力による楽曲検索システムの流れを図4.1に示す。検索システムは次の3つの処理から成る。

- ハミング入力の音高差系列抽出

  ケプストラム法による基本周波数の抽出を行い、ハミング入力の音高差系列を抽出する

- データベース構築

  Standard MIDI fileから音高差系列の抽出を行う

- DPマッチングによる照合

  それぞれの音高差系列のDPマッチングによる照合を行う

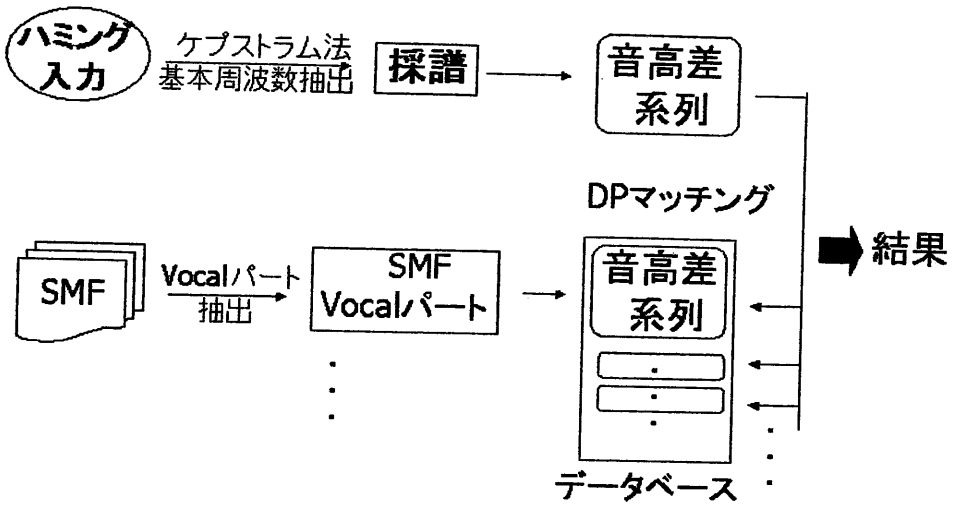

図 4.1: 今回提案したハミング入力による楽曲検索システム

4.2.1 ハミング入力の音高差系列抽出過程

ここではハミング入力からの音高差系列抽出過程について詳しく述べる。抽出過程の流れは次のようになっている。

1. ハミング入力した情報からケプストラム法による基本周波数抽出を行う
2. 基本周波数のグラフから平行な部分を抽出し採譜する
3. 採譜した音高に数値を与えその差をハミング入力の音高差系列とする

まずはじめにハミング入力からケプストラム法による基本周波数抽出を行う。表4.1は基本周波数抽出条件であり、今回は男性を対象としたので抽出範囲を50-300Hzとしている。

表 4.1: 基本周波数抽出条件

サンプリング周波数	20kHz
窓	ハミング窓
フレーム周期	8msec
フレーム幅	32msec
1フレーム抽出ピーク数	10
抽出範囲	50-300Hz

この条件下で実際にかえるの歌をハミングして、基本周波数を抽出し採譜したグラフを図4.2に示す。

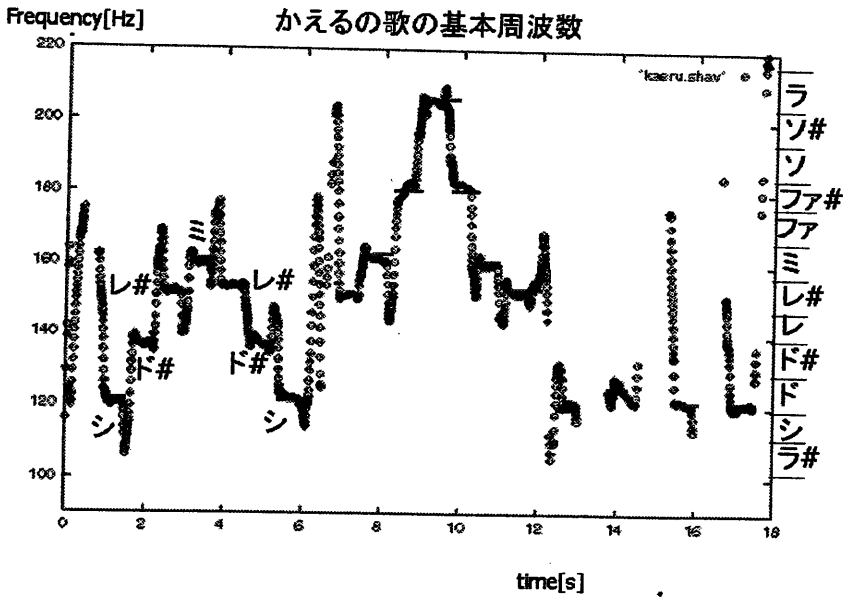

図 4.2: かえるの歌をハミングした基本周波数

横軸は時間で単位は sec、縦軸は周波数で単位は Hz である。このグラフから平行な部分に注目しその周波数から音符を与えた。周波数による採譜例を表 4.2 に示す。

表 4.2: 周波数による採譜条件

音高 (Hz)	コード
127 <= ド (=130) < 134	60
134 <= ド♯ (=138) < 142	61
142 <= レ (=146) < 151	62
151 <= レ♯ (=155) < 160	63
160 <= ミ (=165) < 170	64
170 <= ファ(=165) < 180	65

まず様々な曲をハミングして、基本周波数のグラフを分析した。その結果、各点の周波数の差が0.5Hz以下で12点以上続く部分を平行な部分として抽出すればうまく採譜できることがわかった。以降、この値をしきい値として基本周波数のグラフから採譜を行っている。次に抽出した平行な部分の周波数で採譜し、表4.2に示すコードに直して、その差をハミング入力の音高差系列としている。

今回音高差系列を用いたのは、歌声の調が人により異なるためである。例えば、先ほどのかえるの歌をシ、ド♯、レ♯、ミ、レ♯、ド♯、シと歌っているが、これをハ長調で歌った、ド、レ、ミ、ファ、ミ、レ、ドでも同じ曲として扱うことができるからである。

4.2.2 データベース構築過程

次のような手順を用いてデータベースを構築した。今回データベース構築にはSMF (Standard MIDI file) を用いた。これはMIDI (Musical Instrument Digital Interface) の一般的なファイルである。まずMIDIについて述べる。

- 電子楽器の演奏情報をデジタルで通信するための統一規格
- WWW上で簡単にファイル (SMF) が入手できる
- 音の高さ、長さ、音色、効果がすべて数値で入力されている

SMFはMIDIによる演奏情報の記録や配布において、一般的に使用されているデータ形式であり、演奏データの部分と演奏データには含まれない、テンポ、拍子、曲のタイトルなどのメタ情報の部分から構成されている。さらに、演奏データの部分は、複数のMIDIチャンネルで構成される。通常、それぞれの演奏パートごとにMIDIチャンネルを割り当てていることが多い。

しかしどのMIDIチャンネルをメロディパートにするかは作成者により異なるため、メロディチャンネルの抽出は手作業で行っている。この取り出したメロディチャンネルから音高データを取り出し、その差をとって音高差系列のデータベースを作っている。こうすることで簡単にSMFから音高差系列を抽出することができる。また楽曲全体をデータベースとしているので楽曲のどこから歌っても検索可能になっている。

図4.3はSMFから音高差を抽出しデータベースを構築した例である。

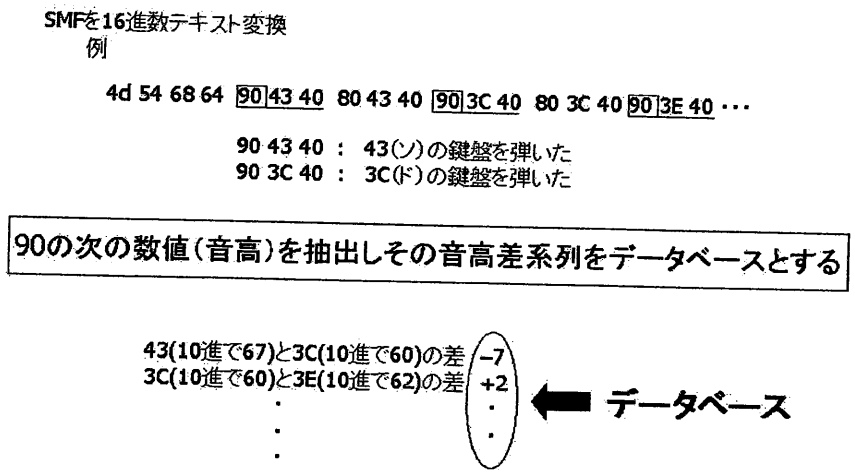

図 4.3: SMFからの音高差系列抽出例

4.2.3 DPマッチングによる照合

次にハミング入力から得られた音高差系列と、データベースの楽曲の音高差系列をDPマッチングする。DPマッチングの手法は2章で述べたとおりであり、最もコストの小さかったものを検索結果としている。

ハミング入力は必ずしも正確ではなく、楽曲データベース中の利用者が意図した音高差系列とハミング入力による音高差系列が一致するとは限らない。旋律間類似度計算では、ハミング入力の不正確さを考慮する必要がある。つまり次の3点を考慮しなければならない。

- 脱落

  同じ高さの隣接する音を区切らないで歌った時、一つの音符として採譜される

- 挿入

  同じ高さの音を余計に歌ったとき、2つの音符として採譜される

- 相違

  利用者の記憶間違いや、歌唱の不正確さによりハミング入力の音高差系列とデータベースの楽曲の音高差系列と異なった時

コストの付け方は、ハミング入力の音高差系列とデータベースの楽曲の音高差系列が一致の時0, 脱落、挿入の時1, 相違の時2を与える。これは、人により歌い方が異なるため脱落、挿入では最小コストを与えるようにしている。また音高差の相違は、脱落、挿入よりも起こりにくいため、より重いペナルティを与えるようにしている。

4.3 実験

このシステムの有効性の検討を行うため実際にハミングを入力し、曲の検索実験を行った。データベースは4.2.2節の手法を用いて20曲作成し、ハミングは7人の歌い手の自由なハミングによる25サンプルを用いた。一回のハミングは5秒から20秒程度としている。なおデータ取得時には、曲名のみを伝え楽曲の好きな部分からハミングしてもらった。

4.3.1 実験結果

実験結果を表4.3に示す。評価指標は、ハミング入力と楽曲データベースとのDPマッチングのコストが最も小さかったものを正答率1位としている。正答率5位以内は同じくDPマッチングのコストが5位以内であった時である。

表 4.3: 検索精度

正答率1位	56 %
正答率5位以内	68 %

次に実験結果についてくわしく分析する。

4.3.2 実験結果例

まず実験結果成功例を図4.4に示す。

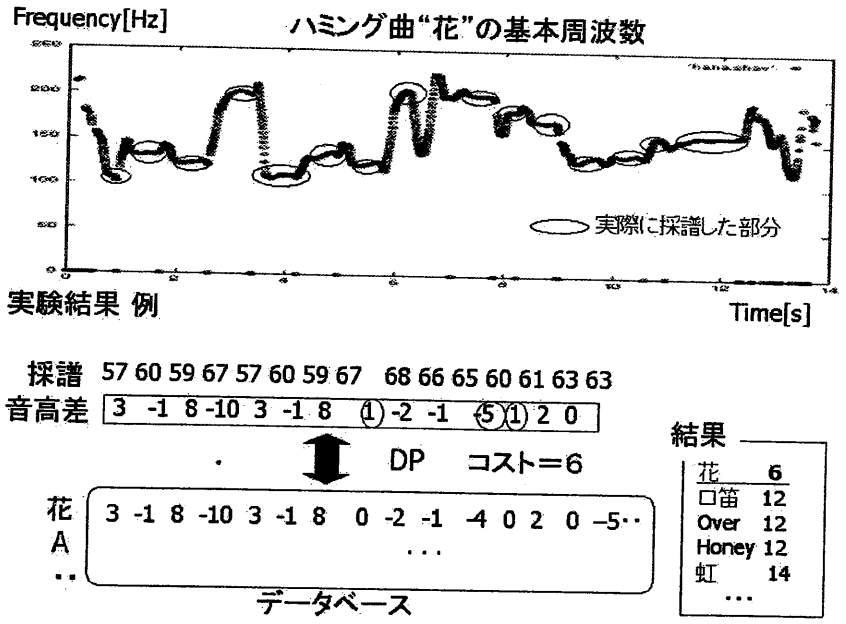

図 4.4: 実験結果成功例

この例は、楽曲「花」のハミング入力したときの例である。まずハミングした曲の基本周波数の平行な部分の周波数で採譜しコードを与える。そのコードの音高差系列をハミング入力の音高差系列とし、データベース中のすべての曲とDPマッチングする。この例はハミング入力の音高差と、データベース中の楽曲「花」のある部分とほとんど同じであったため最小コスト6を出力し、正確に検索できた。

図中で丸で囲んだ個所は、データベース中の「花」の音高差系列とハミング入力の音高差系列で異なった部分である。ここで採譜に失敗したのは、歌い手が音をはずしたため丸で囲んだ3個所のみ異なったものとなった。しかしこの程度の相違はDPマッチングで考慮できたので、他の曲に比べて最も似ている曲として扱われたものと考えられる。この例ではハミングをゆっくりと歯切れ良く歌ったため、基本周波数の抽出が正確になり採譜が正確であった。

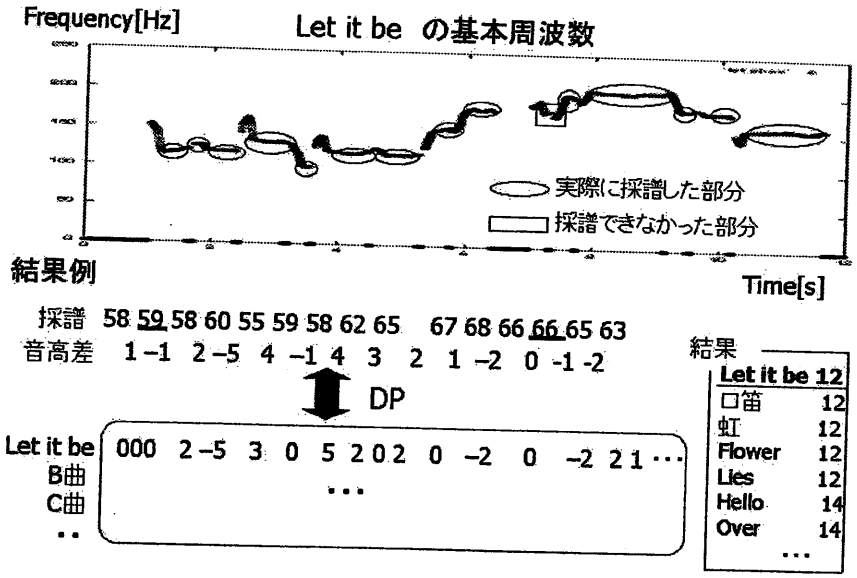

図 4.5: 実験結果例

次に楽曲「let it be」の冒頭の部分をハミングしたときの実験結果を図4.5に示す。これは検索結果1位を出力したが、他の曲とも同程度マッチングしてしまいコストの差が出なかった例である。

まず採譜の時の基本周波数のグラフに余計に平行な部分が表れてしまい、その余計な平行の部分も抽出してしまったため、採譜が正確にできなかったものと思われる。採譜の数字列に下線がひいてあるものが余計に採譜してしまった個所である。また、入力が短い音符であったため採譜できなかった部分もあった。これらにより、ハミング入力音高差系列は正確な系列とならなかったが、部分的にデータベースと合うところがあったため1位に検索されたものと思われる。

また入力したハミングが、部分的にしか正解の楽曲である「let it be」とマッチしなかったため、他の曲にもハミング入力の音高差系列と似た曲が存在し、結果的にコストの差が生じなかったと考えられる。しかし、この程度の系列の違いならば、DPマッチングすることで正しく検索することができた。

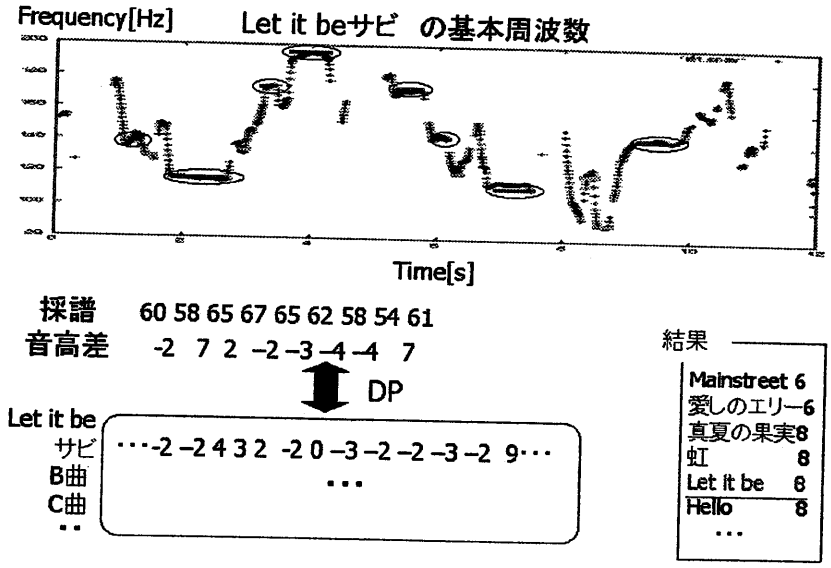

図 4.6: 実験結果失敗例

最後に検索失敗例を図4.6に示す。ここでハミング入力は「let it be」のサビの部分としている。

この例ではハミング入力の音程は正確なのだが、少し歌い方が早かったため発音時間の短い音符を正確に抽出できなかったためであると考えられる。つまり抽出できた音符列が少なかったため、ハミング入力の音高差系列も少なくなってしまい、他の曲のある部分とマッチングしてしまったものと考えられる。このことから、ハミング入力の音高差の系列が少ないと、その曲の特徴をつかみにくいため、他の曲とのコストの差が生じにくいことがわかる。

4.4 考察

実験結果から検索に失敗したものは採譜の時点で失敗したものが多かった。ゆっくり歯切れ良く歌えば、基本周波数の抽出が正確になり正確に採譜できるが、早く歌うと発音時間の短い音符が抽出できないことがわかった。これは採譜プログラムの際に設定したしきい値をもっと広げれば短い音符も抽出できると考えられるが、余計に採譜してしまう部分がさらに多くなってしまうという問題点がある。

またハミングの仕方は人それぞれであり、音程がうまく取れるとも限らない。しかも、うまく歌ったとしてもデータベースの音符列と完全にマッチするとも限らないわけである。

このことから今回提案したハミングによる楽曲検索システムの検索精度の向上を図るには、採譜の精度を上げるよりもさらに脱落、挿入、相違を考慮できるようなDPマッチングの再考察、あるいは新たな照合法が必要であると考えられる。

4.5 まとめ

本章では実際に音高差系列による新しいハミング検索システムを構築した。またその検索システムの評価を行うため、実際に実験を行った。楽曲データベースは20曲、ハミングは25サンプルを用いて実験を行ったが検索精度は正解楽曲が1位となる確率が56％，正解楽曲が5位以内に入る確率が68％であった。この検索精度は、採譜の精度を上げるよりもさらにあいまいさを考慮したDP法、あるいはそれ以外のマッチング手法を用いることで向上できるものと考えられる。

第5章

結論

5.1 本研究の成果

本研究の成果について述べる。

第2章では、第4章で用いたハミング入力の音高差系列と、データベース中の楽曲の音高差系列の照合に用いたDPマッチングの手法について述べた。DPマッチングを用いることにより系列同士の類似度を求め、コストの最も小さいものを検索結果とした。

第3章では、楽曲を入力しその楽曲からメロディパート（vocalパート）を抽出して、メロディラインから楽曲の検索を行おうと試みた。しかし実際は、人の声も楽音であり他のパートの周波数と混ざってしまうため、メロディパートの分離が困難であることがわかった。しかしこの実験から少数パートならば、基本周波数の抽出範囲を男性の第一フォルマント周波数の出力範囲である50-300Hzに絞ることでメロディパートを抽出することができた。この結果から、入力を楽曲全体ではなくハミング入力とすれば、同様の方法でメロディパートが抽出できることがわかり、ハミング入力による楽曲検索システムの提案を行った。

第4章では、新しくハミング入力による楽曲検索システムを構築し検討を行った。検索精度は5割から6割程度であったが、SMFを用いることで楽曲データベースの構築が容易となり、また楽曲データベースにインデックスをつけていないのでどこから歌っても検索できるというメリットがある。さらにDPマッチングによる照合法を考察すればさらなる検索精度の向上が予想できるものと思われる。

5.2 今後の課題

今後の課題としてまず検索精度の向上があげられる。そのためには次の2点を考察することが必要である。

- 正確な採譜のためのしきい値の設定
- DPマッチングのコストの付け方の再考察

1点目は、今回は採譜の時に4分音符は正確に採譜できたが、発音時間の短い8分音符はあまり正確に採譜できなかったので、採譜の時のしきい値の再考察が必要と考えられる。しかし、ハミングは人が入力するため必ずしも正確ではないので採譜にはある程度までの正確性を求めるまででよいと思われる。

2点目は、DPマッチングのコストの付け方を変えることにより他の曲とさらにコストの差を生じさせれば、検索精度は向上するのではないかと考えられる。具体的には相違の時は常に2を与えるのではなく、ハミング入力の音高差系列とデータベース中の楽曲の音高差系列同士の差をとるようにできれば、さらにコストの差が生じるので、検索精度の向上が期待できるものと思われる。

謝辞

本研究を進めるにあたり、多大な御指導とともにこの研究の機会を与えて下さった東北大学大学院工学研究科教授 阿曽弘具氏に心から感謝致します。

音声認識の分野におきましては、東北大学大型計算機センター教授 牧野正三氏、宇都宮大学工学部助手 森大毅氏、東北大学大型計算機センター助手 鈴木基之氏に御指導をして頂いたことを深く感謝致します。

日々の研究におきましては、昼夜を問わず御指導、及び計算機環境の整備をして頂きました東北大学大学院工学研究科助教授 大町真一郎氏、東北大学大学院工学研究科 林貴文氏、馬場雅美氏、門谷信愛希氏ならびに阿曽研究室、牧野研究室の皆様に感謝致します。

参考文献

[1] bit 別冊 コンピュータと音楽の世界 共立出版 1998

[2] 柏野邦夫、村瀬洋" アンサンブル実演奏の自動アンミキサ" 信学技報、SP 97-104,pp.33-40,1998

[3] 蔭山哲也、高島洋典 'ハミング歌唱を手がかりとするメロディ検索" 電子情報通信学会論文誌 Vol.J77-D-2 No.8 pp.1543-1551 ,1994

[4] 園田智也、後藤真孝、村岡洋一"WWW 上での歌声による曲検索システム" 電子情報通信学会論文誌 Vol.J82-D-2 No.4 pp721-731, 1999

[5] 三井田淳郎 音響工学 7 章 音声と聴覚