

平成 22 年度 修士論文

Doubly Nested Circle of Fifths を用いた 和音認識

早稲田大学大学院 基幹理工学研究科 情報理工学専攻

5109B009-9

植村 あい子

指導 甲藤 二郎 教授

2011 年 1 月 31 日

指導教授印	受付印

目次

1. 研究概要	3
1.1. 研究背景	3
1.2. 研究目的	3
1.3. 本論文の概要	4
2. 計算機による音響信号の解析	5
2.1. 音響信号処理	5
2.2. スペクトル分析	5
2.3. 高速フーリエ変換	7
2.4. 短時間フーリエ変換	8
2.5. 定 Q 変換	9
2.6. 一般調和解析	10
2.7. クロマベクトル	12
3. 和音に関する音楽理論と音楽知識	15
3.1. 和音に関する音楽理論	15
3.1.1. 平均律	15
3.1.2. 音程	16
3.1.3. 和音	17
3.1.4. 調	19
3.1.5. 和声法	21
3.2. Spiral Array Model	22
3.3. Doubly Nested Circle of Fifths	23
4. 従来研究	25
4.1. 音楽情報処理の発展	25
4.2. 音高推定	25
4.3. 調推定	26
4.4. 和音認識	26
5. 提案手法	28
5.1. DNCOF ベクトル	28
5.1.1. 生成フロー	28
5.1.2. クロマベクトル	28
5.1.3. コードベクトル	29
5.1.4. DNCOF ベクトル	30

5.1.5.	DNCOF 列.....	33
5.2.	和音認識手法.....	34
5.2.1.	Hidden Markov Model.....	34
5.2.2.	Baum-Welch アルゴリズム.....	35
5.2.3.	和音認識のための HMM 設定.....	37
5.2.4.	ビタビアルゴリズム.....	39
6.	実験.....	41
6.1.	DNCOF ベクトルを用いた和音認識.....	41
6.2.	比較手法.....	42
6.3.	クロマベクトルへの情報応用.....	42
6.4.	実験結果.....	43
7.	結論.....	45
7.1.	まとめ.....	45
7.2.	今後の展望.....	45
	付録.....	46
	謝辞.....	47
	参考文献.....	48
	発表文献リスト.....	50

1. 研究概要

1.1. 研究背景

近年，ハードディスクの大容量化や情報圧縮技術の進化に伴い，我々は大量の音楽を扱えるようになった．また，ネットワークを利用した配信システムやアプリケーションが多く提供され，人々が音楽に触れる手段が増加した．それに付随して，「いつでも」「どこでも」「あらゆる」音楽を扱うことができるように，音楽の解析や検索に対する需要は高まってきている．そのため，さまざま音楽要素の自動認識の研究が多く行われている．

和音は和声をコンパクトに表現したものであり，和声は旋律・リズムと並んで音楽の3要素であることから，和音は音楽を構成する重要な要素といえる．また，和音の進行をもとに楽曲構造を判断したり，調性を推定したりできることから，他の音楽要素と関係が深い手がかりとして和音は注目されている．

和音認識は，時系列パターン認識問題として考えられることが多く，従来研究では，音響信号を入力として和音名系列の出力を得るために学習・認識の手順を踏むものが多数存在する．具体的には，学習時に音響信号と和音名系列のペアから識別器を学習して，認識時に新たな音響信号の和音名系列を出力する[2]～[5]．この識別器には，一般的にHiddenMarkov Model (HMM) が用いられている．特徴量としては，クロマベクトル[1]が主流である．これは，クロマベクトルが，オクターブで演奏されても，構成音が同一であれば同一の和音と認識されるという性質に基づいているためである．

一方，文献[6]では，Circle of Fifths の調の類似性に着目をし，写像によって得られたハーモニー情報を用いて，調性の推定を行っている．本研究では[6]の手法を踏襲し，Belloら[3]が利用した音楽知識 Doubly nested Circle of Fifths に注目して，この知識に基づき得られる和音情報を扱い，和音認識を試みる．

1.2. 研究目的

採譜，通称耳コピと呼ばれる作業は，人手で実音源を聞き取ってひとつひとつ音を楽譜に起こしていくものである．これは，音楽経験のある者にとっても労力と時間のかかる作業である．自動採譜は，この作業をコンピュータにより自動化することで，好きな音楽から誰もが楽譜を容易に得られることを目標とする．自動採譜技術の実現には，音高推定やなど多くの要素技術が必要であり，和音認識はそのひとつに含まれる．

例えば，実際に人間が採譜を行うとき，和音を手がかりにしてひとつひとつの音を求めていくように，自動採譜でも認識した和音から1音ずつ求めていく方法が考えられ，このように和音認識は自動採譜の前処理に利用できるといえる．さらに，和音認識は音楽情報

検索の手がかりにもなる。例えば、ある曲とそのカバーソングは、和音進行が類似しているため、和音がわかればカバーソング検索の手がかりになるといえる。

また、自動採譜だけではなく、音楽のデータベース化とともにいろいろなアプリケーションへの応用が期待できる。例えば、ジャズのようなソロや即興演奏の楽譜化、ユーザが口ずさんだ歌や鼻歌を楽譜化する作曲支援、さらには音楽情報を元にユーザの嗜好にあった楽曲を推薦する音楽検索システムなどが挙げられる。

そこで本研究は、自動採譜や作曲支援・音楽情報検索などへの応用を目的として、音響信号からの和音情報抽出と和音認識を試みる。

1.3. 本論文の概要

本論では、音響信号から得られる和音情報を提案する。この和音情報は音楽知識のひとつである Doubly Nested Circle of Fifths(DNCOF)から生成され、本論では「DNCOF ベクトル」と呼ぶ。DNCOF ベクトルは抽象度の高い和音情報であることから、おおまかな和音情報が得られ、この和音情報を応用で認識精度の向上が期待できる。

本論では、第 2 章で計算機を用いた音響信号処理について述べる。第 3 章では和音に関する音楽理論や音楽知識に関する説明、第 4 章では従来研究を紹介する。そして第 5 章において提案手法を述べる。第 6 章にて認識評価実験結果を述べ、提案手法の有効性を示す。最後に第 7 章で結論とまとめを述べ、本論の締めくくりとする。

2. 計算機による音響信号の解析

音楽情報処理における研究は、従来音声認識の分野で利用されてきた解析方法が広く応用されている。第 2 章では計算機による音響信号解析について、音声認識の知識をおり混ぜながら紹介する。

2.1. 音響信号処理

音響信号には、音の高低に関する情報、ジャンル情報などさまざまな情報が含まれている。人間が音楽を聞くと、これらの情報の認識は無意識的に行われている。計算機による音楽要素の自動認識を考えると、人間のように計算機も同様の認識処理を行えることが望まれる。よって、音響信号を解析し、上記の情報を抽出するための研究テーマは重要な課題となる。

2.2. スペクトル分析

人間の聴覚器官ではある種のスペクトル分析を行っている。同様に音声の分析をスペクトル分析で行うことが多い。特に人間の聴覚は、スペクトルの位相情報に鈍感であるため、パワースペクトル分析が広く用いられている。

ここで、
$$\left\{ \begin{array}{l} x(n), [n = \{0, 1, 2, \dots, N-1\}] \cdots \text{デジタル化された音響 (音声) 信号} \\ X(k), [k = \{0, 1, 2, \dots, N-1\}] \cdots \text{離散フーリエ変換 (Discrete Fourier Transform: DFT)} \\ \text{されたスペクトル} \\ r(m), [m = \{-(N-1), \dots, 0, \dots, N+1\}] \cdots x(n) \text{の自己相関関数} \end{array} \right.$$
とする。

これらの分析の関係を図 2.2.1 に示す。一般に、スペクトルは信号波形を短時間ブロックごとに窓関数を掛け算することで計算される。窓のタイプとして図 2.2.2 に示すような、ハミング窓、ハンニング窓、ブラックマン窓などがあるが、通常用いられるのはハミング窓である。ここでは、これらの窓によって切出された N 点の信号を $x(n)$ として話を続ける。

信号 $x(n)$ に DFT を施し、スペクトル $X(k)$ を得る式は次の通りである。

$$X(k) = \sum_{n=0}^{N-1} x(n) \exp(-j \frac{2\pi kn}{N}) \quad , \quad (0 \leq k \leq N-1) \quad (2.2.1)$$

当然ながら図 2.2.1 にも示されるように、このスペクトル $X(k)$ は離散逆フーリエ変換 (Inverse DFT: IDFT) で元の信号波形 $x(n)$ に戻すことができる。

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) \exp(j \frac{2\pi kn}{N}) \quad , \quad (0 \leq n \leq N-1) \quad (2.2.2)$$

また，パワースペクトル $S(k)$ は次式で計算される．

$$\begin{aligned} S(k) &= \frac{1}{N} \left| \sum_{n=0}^{N-1} x(n) \exp(-j \frac{2\pi kn}{N}) \right|^2 \quad , \quad (0 \leq k \leq N-1) \\ &= \frac{1}{N} (R(X(k))^2 + I(X(k))^2) \end{aligned} \quad (2.2.3)$$

一方で，パワースペクトルは自己相関関数 $r(m)$ から計算する方法もある．

$$r(m) = \frac{1}{N} \sum_{n=0}^{N-1-|m|} x(n)x(n+|m|) \quad , \quad (|m| = 0, 1, \dots, N-1) \quad (2.2.4)$$

$$S(k) = \sum_{m=-(N-1)}^{N-1} r(m) \exp(-j \frac{2\pi km}{N}) \quad , \quad (0 \leq k \leq N-1) \quad (2.2.5)$$

これを Wiener Khintchine の定理と呼ぶ．

また，自己相関関数は偶関数であるため，次のように \cos 変換で表せる．

$$S(k) = \sum_{m=-(N-1)}^{N-1} r(m) \cos(\frac{2\pi km}{N}) \quad , \quad (0 \leq k \leq N-1) \quad (2.2.6)$$

当然，IDFT によりパワースペクトルから自己相関関数を得ることもできる．

$$r(m) = \frac{1}{N} \sum_{k=-(N-1)}^{N-1} S(k) \exp(j \frac{2\pi km}{N}) \quad , \quad (|m| = 0, 1, \dots, N-1) \quad (2.2.7)$$

もちろん， \cos 逆変換でも計算できる．

$$r(m) = \frac{1}{N} \sum_{k=-(N-1)}^{N-1} S(k) \cos(\frac{2\pi km}{N}) \quad , \quad (|m| = 0, 1, \dots, N-1) \quad (2.2.8)$$

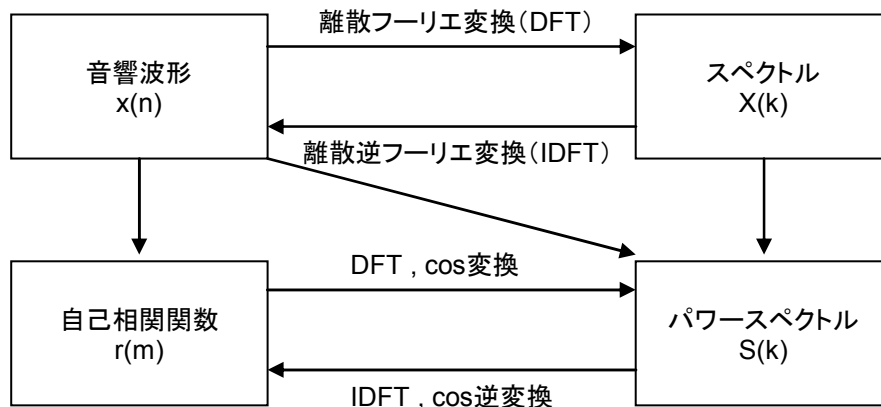


図 2.2.1 音声波形とパワースペクトルの関係

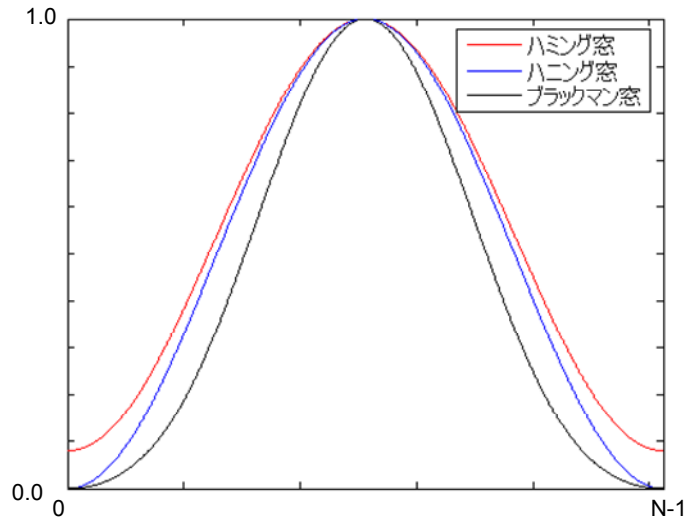


図 2.2.2 単時間スペクトル分析のための時間窓

2.3. 高速フーリエ変換

DFT や IDFT の高速な計算方法として高速フーリエ変換 (Fast Fourier Transform: FFT) がある。これは、DFT の指数関数部の掛け算を極力減らすようにして導かれる。以下に FFT の導出を示す。

まず、DFT の原理式を考える。

$$X(k) = \sum_{n=0}^{N-1} x(n)W_N^{nk} \quad W_N = e^{-\frac{2\pi}{N}} \quad (2.3.1)$$

これを直接計算すると入力信号が実数の場合、 N^2 回の複素乗算を必要とする。例えば 1024 点 DFT を計算するには 2^{20} 回の乗算が必要となり、膨大な量が問題となる。そこで、このような問題を解決するために考えられたのが FFT であり、FFT の発見によりデジタル信号処理技術が飛躍的に発展したとも言える。

ここで $N = 2^q$ のときの DFT の高速化を考える。

式 (2-3-1) を n が偶奇の場合で場合分けする。[$n = \{0,1,2,\dots,N-1\}$]

$$\begin{aligned} X(k) &= \sum_{n=0}^{\frac{N}{2}-1} x(n)W_N^{2nk} + \sum_{n=0}^{\frac{N}{2}-1} x(2n+1)W_N^{(2n+1)k} \\ &= \sum_{n=0}^{\frac{N}{2}-1} x(2n)W_{\frac{N}{2}}^{nk} + W_N^k \sum_{n=0}^{\frac{N}{2}-1} x(2n+1)W_{\frac{N}{2}}^{nk} \end{aligned} \quad (2.3.2)$$

式 (2.3.2) より N 点 DFT を $N/2$ 点 DFT の組合せに分割できたことが分かる。

同様にして、式 (2.3.2) の第 1 項 $= \sum_{n=0}^{\frac{N}{2}} x(2n)W_{\frac{N}{2}}^{nk}$ と W_N^k を除いた第 2 項 $= \sum_{n=0}^{\frac{N}{2}-1} x(2n+1)W_{\frac{N}{2}}^{nk}$

を $N/4$ 点 DFT の組合せに分割すると、

$$X(k) = \sum_{n=0}^{\frac{N}{4}-1} b_0(2n)W_{\frac{N}{4}}^{nk} + W_{\frac{N}{2}}^{nk} \sum_{n=0}^{\frac{N}{4}-1} b_0(2n+1)W_{\frac{N}{4}}^{nk} + W_N^k \left\{ \sum_{n=0}^{\frac{N}{4}-1} b_1(2n)W_{\frac{N}{4}}^{nk} + W_{\frac{N}{2}}^k \sum_{n=0}^{\frac{N}{4}-1} b_1(2n+1)W_{\frac{N}{4}}^{nk} \right\}$$

$$\text{ただし, } x(2n) = b_0(n), x(2n+1) = b_1(n) \quad (2.3.3)$$

このような処理を W_2 まで繰り返すと、FFT では全体の演算量が $\frac{N}{2} \log_2 N$ 回となり、計算量を大幅に減らすことができる。

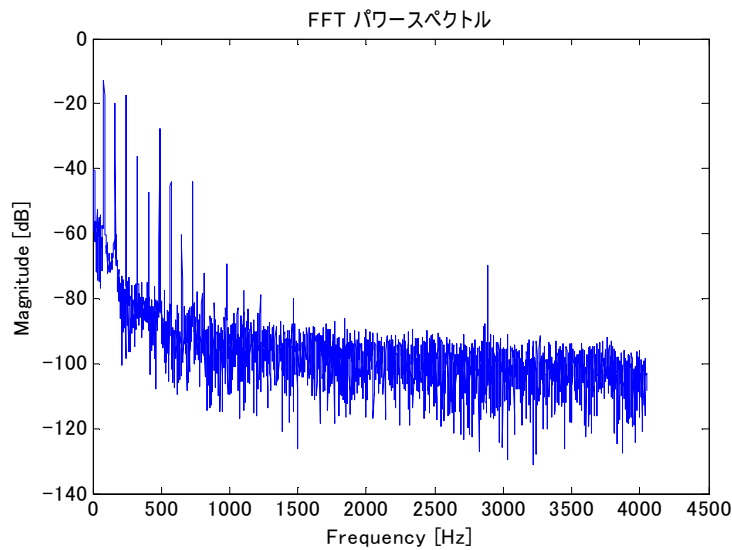


図 2.3.1 FFT パワースペクトル

2.4. 短時間フーリエ変換

短時間フーリエ変換 (short-time Fourier transform, short-term Fourier transform, STFT) とは、関数に窓関数をずらしながら掛けてセグメントを抽出し、各セグメントに対して DFT を行うことである。音声など時間変化する信号の周波数を解析するためによく使われる。

STFT では原関数に窓かけをしてセグメントを抽出する。したがって、この窓幅を小さくすれば時間分解能が上がり、より詳細な時間変化を観測できる。しかし、窓幅を小さくすると、周波数分解能が下がってしまう。つまり、STFT には時刻と周波数に関する不確定性

原理がある.

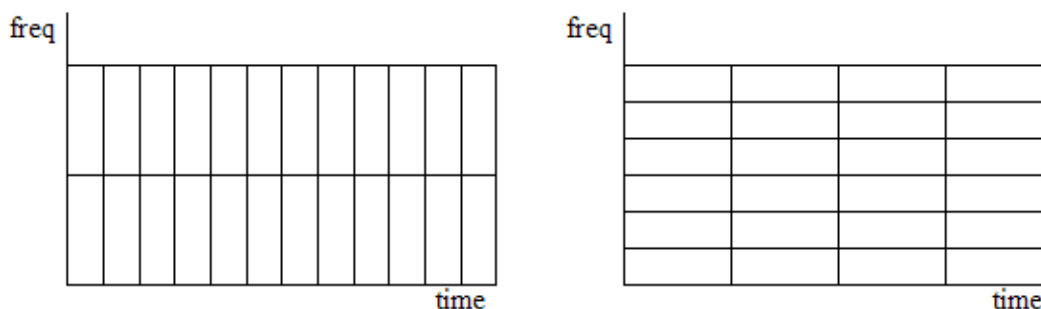


図 2.4.1 STFT の不確定性原理

2.5. 定 Q 変換

前節で紹介した FFT では, 入力として得られるサンプルの半分の bin しか存在しない. FFT の結果が, 音楽におけるオクターブのような対数スケールに変換されるとき, 分解能は低いオクターブで悪くなってしまふ. 低い周波数で半音離れている 2 つの音を分離するには, 長い時間窓が必要となる. しかし, 高い周波数において同じ分解能を用いるのは無駄になってしまう. これは, 人間の耳では対数的に連続な周波数で聞こえるのに対し, 一方フーリエ解析の線形なスケールでは両者の間に食い違いが生じる.

このような問題は, バンド幅が周波数に比例して変化する定 Q 変換 (Constant Q Transform) [7] のような手法で解決される. 図.1 のように定 Q 変換において分析窓は低い周波数では細く, 高い周波数では広くなる. 窓の長さは分析される周波数にしたがって変化する. 長い窓は低い周波数を分析し, 短い窓は高い周波数を分析する.

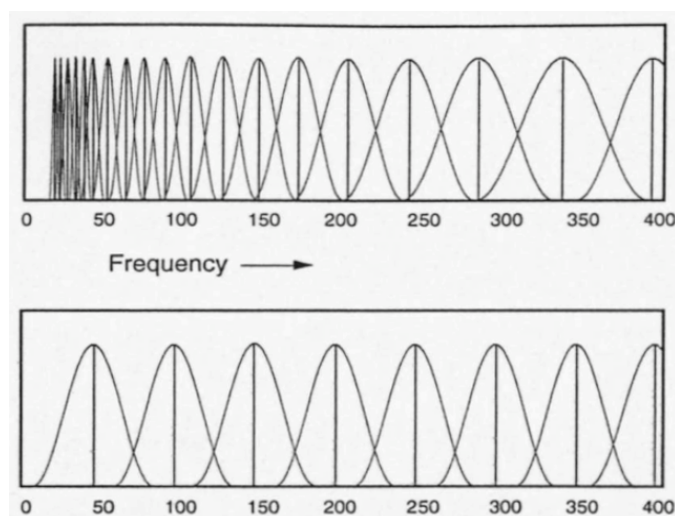


図 2.5.1 定 Q 変換とフーリエ変換のバンド幅[8]

(上: 定 Q 変換, 下: フーリエ変換)

計算方法を以下に示す. 入力信号 $x(m)$ の定 Q 変換 $X_{cq}(k)$ は以下の式で表される.

$$X_{cq}(k) = \sum_{n=0}^{N(k)-1} w(n, k)x(n)e^{-j2\pi f_k n} \quad (2.5.1)$$

$w(n, k)$ は解析窓であり, その長さが $N(k)$ で bin ポジション k の関数である. このとき k 番目の中心周波数 f_k は次式のように定義される.

$$f_k = \left(2^{\frac{1}{B}}\right)^k f_{\min} \quad (2.5.2)$$

ただし, f_{\min} は定 Q 変換における最小周波数, B は 1 オクターブの bin 数である. k 番目の bin の窓幅 $N(k)$ は以下の式で表される.

$$N(k) = \left(\frac{S}{f_k}\right)Q \quad (2.5.3)$$

S はサンプリング周波数であり, Q はその中心周波数で処理されるサイクル数である.

定 Q 変換の実装では, フィルタのバンド幅がその中心周波数に比例するフィルタバンクを用いる方法があるが, 直接的な方法では計算量において FFT の効率性を持たない. そこで, 速度向上方法として, FFT とカーネルを利用したもの[9]がある.

なお, 定 Q 変換は, 時間と周波数の不確定性の関係から逃れることはないが, 時間的な不確定性は低いオクターブに集中する.

2.6. 一般調和解析

音高解析という点に重みを置いた場合, 一般調和解析 (Generalized Harmonic Analysis: GHA) と呼ばれる手法が有効である. GHA は, 時間幅によらず高い周波数成分分解能が得られることで知られている. FFT などでは時間と周波数の間にトレードオフの関係があり, 高い周波数分解能を得るためには長い時間幅が必要となる. また, FFT が窓幅の周期信号であるのに対し, GHA は音楽など非定常状態の波形解析にも向いている.

アルゴリズムを以下に示す.

(i) 連続信号 $x_0(t)$ があるとする. $x_0(t)$ 短区間 $[0, L]$ で観測された信号とにおいてフーリエ

係数 $S(f)$ および $C(f)$ を求める.

$$S(f) = \frac{2}{nT} \int_0^{nT} x_0(t) \sin\left(\frac{2\pi t}{T}\right) dt \quad (2.6.1)$$

$$C(f) = \frac{2}{nT} \int_0^{nT} x_0(t) \cos\left(\frac{2\pi t}{T}\right) dt \quad (2.6.2)$$

式 (2.6.1) および式 (2.6.2) において, T は周期, n, nT は $nT \leq L$ を満たす整数である.

(ii) (i) で求めたフーリエ係数を用いて観測区間 $[0, L]$ における残差信号 $\varepsilon(t, T)$ とそのエネルギー $E(T)$ を求める.

$$\varepsilon(t, T) = x_0(t) - S(T) \sin\left(\frac{2\pi t}{T}\right) - C(T) \cos\left(\frac{2\pi t}{T}\right) \quad (2.6.3)$$

$$E(T) = \int_0^L \varepsilon(t, T)^2 dt \quad (2.6.4)$$

次に式 (2.6.4) における $E(T)$ を最小とする周期 T_1 と, そのフーリエ係数 $S(T_1), C(T_1)$ を求める.

(iii) 原信号 $x_0(t)$ から (ii) で求めた周波数成分 ($f_1 = 1/T_1$) を除去する.

$$x_1(t) = x_0(t) - S(T_1) \sin\left(\frac{2\pi t}{T_1}\right) - C(T_1) \cos\left(\frac{2\pi t}{T_1}\right) \quad (2.6.5)$$

そして, この $x_1(t)$ を原信号とみて (i) ~ (iii) を繰り返す.

(iv) (i) ~ (iii) の処理を所望数数の M 回繰り返すことで, はじめの観測信号 $x_0(t)$ は観測区間 $[0, L]$ において分析合成されることになる.

$$\left\{ \begin{array}{l} x_0(t) = \sum_{k=1}^M \left\{ S(T_k) \sin\left(\frac{2\pi k}{T_k}\right) + C(T_k) \cos\left(\frac{2\pi k}{T_k}\right) \right\} + R(t) \\ R(t) : \text{分析残差信号} \end{array} \right. \quad (2.6.6)$$

ここで, T_k は一般に調和関係になっていない. よって概周期関数 $x_0(t)$ により観測区間 L を超えて原信号を予測することが可能である.

また, 信号のパワースペクトルは,

$$P(T_k) = S(T_k)^2 + C(T_k)^2 \quad (2.6.7)$$

で推測される.

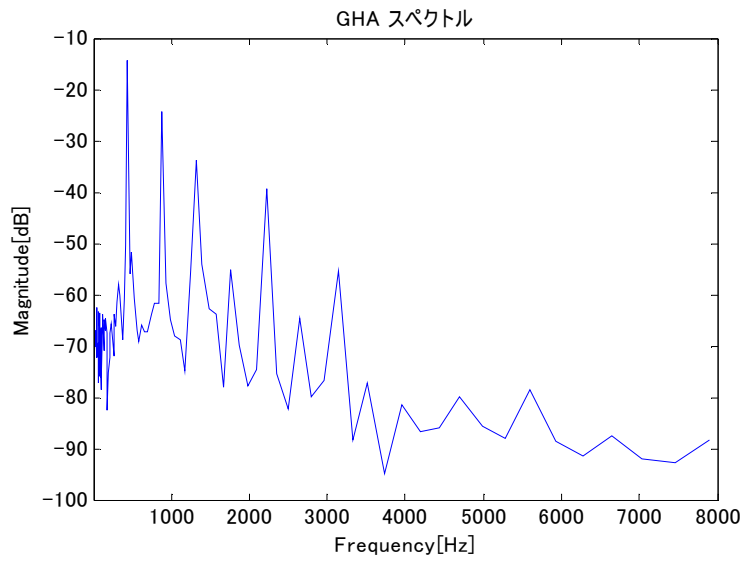


図 2.6.1 GHA パワースペクトル

2.7. クロマベクトル

クロマベクトル/Pitch Class Profile (PCP)[1]は、周波数パワースペクトルを特定のピッチクラスに振り分けたものである。多くの研究では、図のようにクロマベクトルは半音階に相当する 12 個のピッチクラスを想定し、オクターブ違いは吸収したものを使用する。

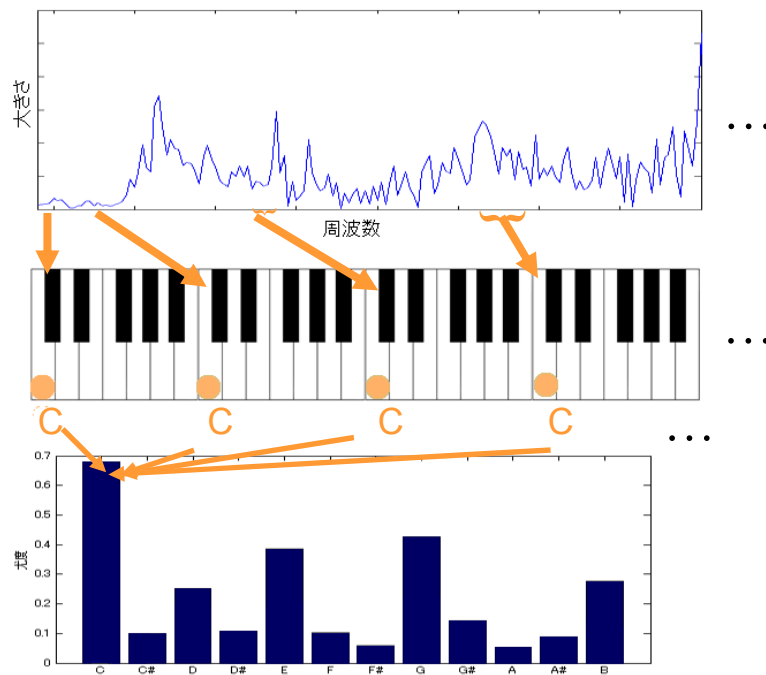


図 2.7.1 クロマベクトルのイメージ

クロマベクトルの計算方法には、FFT のスペクトルパワーを用いるもの[1,10]があるが、本稿では、文献[3,11]を参考として、定 Q 変換を用いたクロマベクトルを扱う。その理由は、クロマベクトルを計算するのに、周波数方向に対数のピッチに合わせて解析するのが容易であり、高音での時間分解能を下げずに低音の周波数分解能を上げられるからである。

[11]で提案されているクロマベクトルの計算方法を以下に示す。

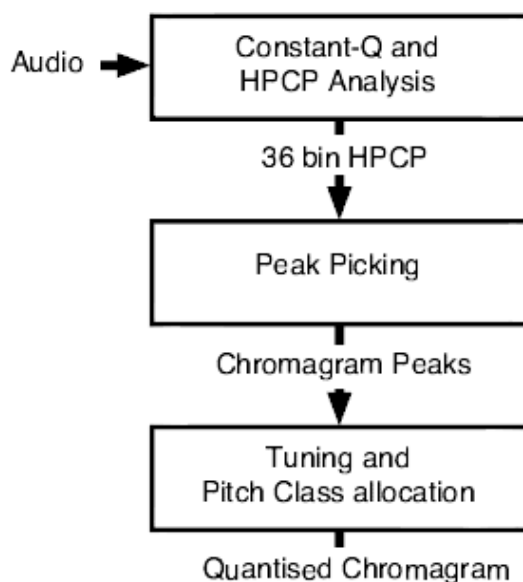


図 2.7.2 クロマベクトル計算の流れ

定 Q スペクトルの結果 X_{cq} を用いて次式から、36bin クロマベクトル $CH(b)$ を計算する。

$$CH(b) = \sum_{m=0}^M |X_{cq}(b+12m)| \quad 1 \leq b \leq 36 \quad (2.7.1)$$

ここで、 M は定 Q スペクトルの総オクターブ数、 b はクロマベクトルの bin インデックスである。

実際には各楽曲のチューニングが一定していないため、 $A4=440$ (Hz) として計算した場合にスペクトルのパワーが他の bin にも分配されてしまい、明瞭なクロマベクトルが得られないことがある。そこで、式で得られたクロマベクトルに対し、チューニングを行う。

チューニングははじめに $CH(b)$ のピークピッキングを行う。各フレームから、2 次補間を用いてピークの位置と大きさを求める。次に全てのピークのチューニングヒストグラムを計算したら、強く集中するピークが 1 つ求められる。これを中心チューニング周波数と一致するとみなす。

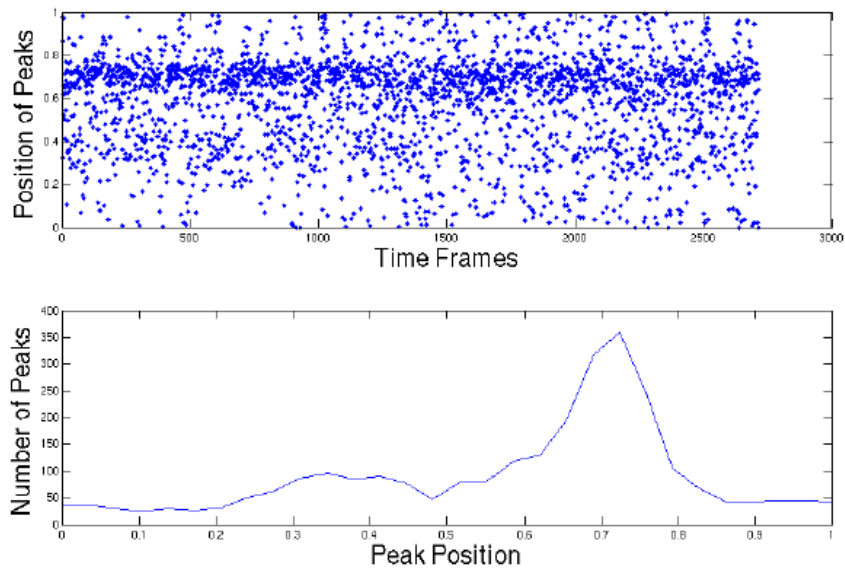


図 2.7.3 $CH(b)$ のピーク分析結果(上:ピーク位置, 下:ピークの分布)

この周波数がわかったら、ピッチクラス間の境界値が見つかるので、ピッチクラスに割り当てる。つまり、その両側に沿ったウィンドウで、各 bin へのピークを選び、分配を行う。このようにして、チューニングされた 12bin クロマベクトル $Chroma_{P \bmod 12}$ を得る。

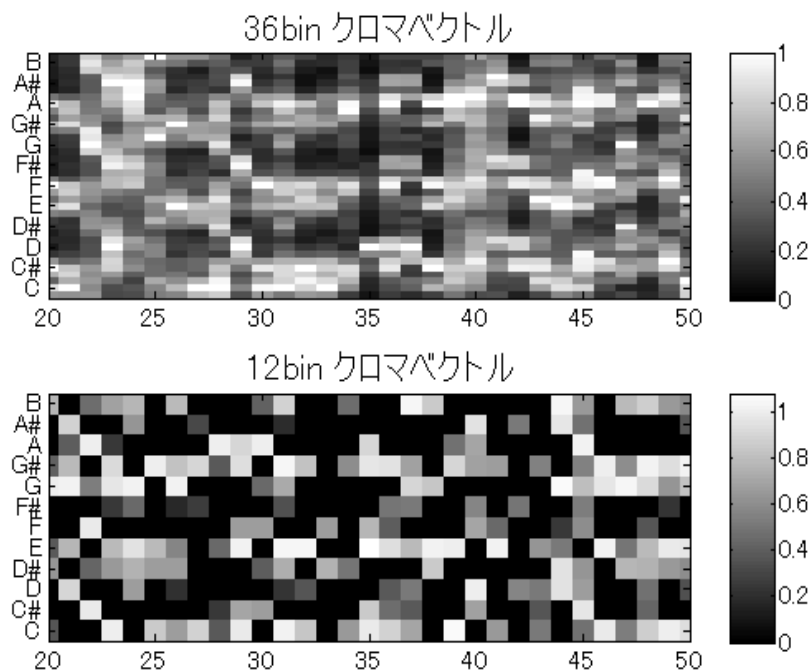


図 2.7.4 The Beatles のアルバム Beatles for Sale より”Eight Days A Week”のクロマベクトル (20 サンプル目から 50 サンプル目)

3. 和音に関する音楽理論と音楽知識

3.1. 和音に関する音楽理論

和音は他の音楽要素と関わりがあり、特に和声や調性とは密接な関係がある。和声を短期的にとらえたものが和音であり、長期的にとらえたものが調といえる。このように、それぞれの音楽要素は密接に関わりあっているため各要素を切り離さずにとらえる必要がある。

以下では、現代音楽理論において「単音」がどのように決まるかの説明から始まり、2音の関係である「音程」、3音以上の短期間特徴である「和音」、長期間特徴である「調性」へと説明を進めていく。

3.1.1. 平均律

平均律とは、一般的に1オクターブを12分して出来た音階をいう。平均律は各音の音程の幅が一定であり、どの音の間でも同じ音程なら同じ振動数比になる。我々が普段耳にする音楽のほとんどは平均律に基づいて作られている。12分した1間隔を「半音」、2間隔を「全音」と呼ぶ。

人間は音高（音の高さ）の高低を周波数の大小によって認識する。耳の構造上、この高低を対数スケールで認識する。すなわち、ある周波数 $f_1 < f_2 < f_3$ (Hz) の3音があり、これらが等間隔に聞こえるとき

$$f_3 = \frac{f_2}{f_1} \times f_2 \quad (3.1.1)$$

を満たす。また、

$$f_2 = 2f_1 \quad (3.1.1)$$

を満たすとき f_2 の音は f_1 の音より1オクターブ高いといい、2音はほぼ同じ音と認識される。これを繰り返すことによって、 f_1 の音と $2^n \times f_1$ (n : 整数) の音はほぼ同じ音と認識される。

平均律で決められる音には音名が与えられる。音名は A4 や F#0 のように、keyletter (アルファベット[+シフト記号 (#,b)]) + keynumber (数字) によって表される。keyletter は1オクターブ内の相対位置によって与えられ、keynumber は1オクターブ高くなる(低くなる)と1増える(減る)。上記よりオクターブ違いの音は同じ keyletter が与えられることになる。keyletter は C, C#, D, D#, E, F, F#, G, G#, A, A#, B の12種ある (C#=Db などの同音異名があるが、ここでは#系表記で統一する)。keynumber は理論的には全ての整数を取り

うるが、実用的には0~9の範囲である。現代音楽においては、音名の基準値として以下の式が与えられる。

$$f_{A4} = 440(\text{Hz}) \quad (3.1.2)$$

また、平均律のすべての音名は以下の式で与えることができる。

$$\begin{cases} a_{\text{half}} = 2^{1/12} \\ a_{\text{half}}^{-9} \times f_{C4} = \dots = a_{\text{half}}^{-1} \times f_{G\#4} = f_{A4} = a_{\text{half}}^1 \times f_{A\#4} = a_{\text{half}}^2 \times f_{B4} \\ f_{L_1 N_1} = 2^{N_1 - N_2} f_{L_2 N_2} \end{cases} \quad (3.1.4)$$

以下の表 3.1.1 に、実用的な範囲における音名とその周波数の対応表を示す。

表 3.1.1 平均律による音名とその周波数表[Hz]

	0	1	2	3	4	5	6	7	8	9
C	16.35	32.70	65.14	130.81	261.63	523.25	1046.50	2093.00	4186.01	8372.02
C#	17.32	34.65	69.30	138.59	277.18	554.37	1108.73	2217.46	4434.92	8869.84
D	18.35	36.71	73.42	146.83	293.66	587.33	1174.66	2349.32	4698.64	9397.27
D#	19.45	38.89	77.78	155.56	311.13	622.25	1244.51	2489.02	4978.03	9956.06
E	20.60	41.20	82.41	164.81	329.63	659.26	1318.51	2637.02	5274.04	10548.08
F	21.83	43.65	87.31	174.61	349.23	698.46	1396.91	2793.83	5587.65	11175.30
F#	23.12	46.25	92.50	185.00	369.99	739.99	1479.98	2959.96	5919.91	11839.82
G	24.50	49.00	98.00	196.00	392.00	783.99	1567.98	3136.96	6271.93	12543.85
G#	25.96	51.91	103.83	207.65	415.30	830.61	1661.22	3322.44	6644.88	13289.75
A	27.50	55.00	110.00	220.00	440.00	880.00	1760.00	3520.00	7040.00	14080.00
A#	29.14	58.27	116.54	233.08	466.16	932.33	1864.66	3729.31	7458.62	14917.24
B	30.87	61.74	123.47	246.94	493.88	987.77	1975.53	3951.07	7902.13	15804.27

表は列方向に半音ずつ等間隔に並んでおり、ある列の最下段とその右列の最上段も半音隔たっている。行方向にはオクターブ違いの音が左から右へ高くなるように並んでいる。ちなみにいわゆるピアノの「中央のド」は C4 にあたる。人間の可聴音域は 20~20000(Hz) といわれているため、音名で言う E0~E10 が網羅できればよく、上表の範囲は妥当といえる。実際に音楽で使われる音高は上限がせいぜい keynumber=7 くらいまでであるため、高調波構造を十分含んだ範囲となっている。

3.1.2. 音程

2つの音の高さの隔たりを音程という。「完全5度」「長2度」というように、「完全」「長」「短」などの接頭語と、楽譜の五線上の高さの隔たりを表す度数とを組み合わせることで表現す

る。

#・b などの変化記号を考慮せずに、五線上の高さが同じ 2 音の音程を「1 度」と呼ぶ。そして 1 ステップ離れた (2 音分にまたがる) 音程を「2 度」、2 ステップ離れた音程を 3 度と呼び (以下、同様)、7 ステップ離れたオクターブは「8 度」となる。

五線上の 1 ステップには、半音の箇所と全音の箇所の 2 種類があるため、同じ度数で表される音程であっても、半音と全音が含まれる状況によって、それぞれの音程の大きさに違いが生じる。また、変化記号を付された音を含む音程は、もとの音程とは当然異なってくる。それらの違いは「完全」「長」「短」「増」「減」などの接頭語で表される。オクターブ以内は次の通りになる。

表 3.1.2 音程の測定法[12]

音程名	五線上の ステップ数	=	全音の ステップ数	+	半音の ステップ数
完全 1 度	0	=	0	+	0
短 2 度	1	=	0	+	1
長 2 度	1	=	1	+	0
短 3 度	2	=	1	+	1
長 3 度	2	=	2	+	0
完全 4 度	3	=	2	+	1
増 4 度	3	=	3	+	0
減 5 度	4	=	2	+	2
完全 5 度	4	=	3	+	1
短 6 度	5	=	3	+	2
長 6 度	5	=	4	+	1
短 7 度	6	=	4	+	2
長 7 度	6	=	5	+	1
完全 8 度	7	=	5	+	2

同じ度数においては、長音程は短音程よりも半音広い。そして、各度数において、「完全」及び「長」よりも半音広いものを「増」、「完全」及び「長」よりも半音狭いものを「減」と称する。「増」よりもさらに半音広いものは「重増」、「減」よりもさらに半音が狭いものは「重減」と呼ぶ。いずれかの全階の中に属しうる音程を「全音階的音程」といい、単音程では表で挙げた 14 種類のみがこれに該当する。それ以外の音程は「半音階的音程」という。

3.1.3. 和音

和音とは、広義には高さの異なる複数の音が同時に鳴り響く様態一般をさすが、狭義

には、調整に基づく音楽において、その原理のひとつを担う音の垂直的な重ね合わせのことをいう。三和音、七の和音などの種類がある。

最も基礎的な和音はある音（根音）の上方に3度と5度関係にある音（第3音、第5音）を積み重ねた三和音（トライアド）である。構成音程の違いにより、長三和音（major：長3度と完全5度）、短三和音（minor：短3度と完全5度）、増三和音（augmented：長3度と増5度）、減三和音（diminished：短3度と減5度）の4種が存在する。



図 3.1.1 長三和音，短三和音，増三和音，減三和音の例[12]

三和音に根音から7度上方の音（第7音）を重ねたものを七の和音（セブンスコード）という。これには、属七の和音（長三和音と短7度）、長七の和音（長三和音と長7度）、短七の和音（短三和音と短7度）、減五短七の和音（または導七の和音。減三和音と短7度）、減七の和音（減三和音と減7度）がある。



図 3.1.2 属七の和音，長七の和音，短七の和音，減五短七の和音，減七の和音[12]

さらに根音から9度上方の音（第9音）を重ねたものを九の和音（テンションコード）という。代表的なものに属九の和音があり、第9音の音程の長短により2種類が存在する。ひとつは、長音階の属音上に成立するため「長調の属九の和音」と呼ばれ、もうひとつは、和声的短音階の属音上に成立するため「短調の属九の和音」と呼ばれる。



図 3.1.3 和音の転回形[12]

また、根音以外の音が最低音に置かれる状態を、和音の転回という。最低音に第3音が

置かれるものを第 1 転回形，第 5 音が置かれるものを第 2 転回形という．三和音の第 1 転回形は，その最低音に対する上部構成音の音程関係から六の和音とよばれる．同様に，三和音の第 2 転回形は四六の和音，七の和音の第 1 転回形は五六の和音，第 2 転回形は三四の和音と呼ばれる．七の和音・九の和音で，第 7 音が最低音となるものを第 3 転回形という．なお，九の和音において第 9 音は根音より 9 度以上上方に置かれなければならない規則があるため，九の和音の第 4 転回形は原則的に存在しない．

和音構成音はオクターブ音を任意に重複することができるが，これによって和音の機能・性質が変わることはない．ただし，厳格な声部信仰を持つテクスチャにおいては，脈絡上不可能・不適切な重複もあり，また，構成音の上下関係に制約のある場合もある[12]．

コードネーム (chord symbol) は，旋律のバックに鳴っている和音を表すために，Jelly Roll Morton (1890-1941) により考案された記号である．これは，ポピュラー音楽や，ギターで和音を演奏するとき多用され，基本的に英語が用いられている．

コードネームは，英語音名で根音を表し，それに和音の種類を表すサフィックスを添えた形で表す．サフィックスは根音とその他の構成音の相対的な音程の組み合わせで決まるため，印象は根音によらない．構成音の音程の組み合わせと，それに対応するサフィックスを以下の表に示す．

表 3.1.3 サフィックスと構成音の音程の組み合わせ

サフィックス	根音以外の構成音		
	3 度族	5 度族	その他
なし(major)	長 3 度	完全 5 度	
m	短 3 度	完全 5 度	
sus4	完全 4 度(※)	完全 5 度	
dim	短 3 度	減 5 度	長 6 度
aug	長 3 度	増 5 度	
7(b5)	長 3 度	減 5 度	短 7 度
m7(b5)	短 3 度	減 5 度	短 7 度

※3 度族に含まれるのは，長 3 度の音が半音上にシフトしていると考えられるため．

3.1.4. 調

調は，今日では，平均律に基づく 12 の音それぞれを主音とする長・短調，すなわち 24 の調が使用される．一般的な音楽では，オクターブを無視すれば C, C#, ..., B の 12 音しか存在せず，しかも同一調区間ではそのうちの 7 音を主として用いる．調は，根音とスケール (scale) の組み合わせによって決まる．スケールとは音階のことで，同一調内で用いられる音の集合の相対音程によって決まる．以下に調とスケールの例を示す．

表 3.1.4 サフィックスと構成音の音程の組み合わせ

調	スケール	構成音	備考
C major (ハ長調)	diatonic	C, D, E, F, G, A, B	一般的な長調
	pentatonic	C, D, E, G, A	民謡などでよく用いられる
	blue note	C, D, Eb, F, Gb, A, Bb	minor キーの構成音に近いが, major に分類される
A minor (イ短調)	natural minor	A, B, C, D, E, F, G	C major と構成音が等しい
	harmonic minor	A, B, C, D, E, F, G#	
	melodic minor	A, B, C, D, E, F#, G# (上行時), A, B, C, D, E, F, G (下行時)	旋律, コードが自然になる. メロデ ィの向きによって音が変わる.

おおざっぱに言えば、根音の種類は 12 種で、長 / 短の 2 タイプしかないので、ほとんどの場合、「長調」「単調」のみである。blue note スケールは diatonic と組み合わせて用いられるため楽典上長調として扱われる特殊なスケールの 1 つである。pentatonic は diatonic に完全に包含されており、A minor の natural minor スケールの構成音は C major の diatonic と完全に等しい。このように、調号の同じ長調・短調は平行調とよばれる。そして、上記表の音名をシフトしていけば全ての調を得ることができる。また、主音を同じくする長調・短調は同主調（同名調）と呼ばれる。

12 の各調・単調は音階構成音の共通性や和音・和声機能の関連性を持って、属音（完全 5 度上）を主音とする調（属調）、及び下属音（完全 5 度下）を主音とする調（下属調）と近親関係にある。この関係が連鎖したものを Circle of Fifths（5 度圏）という。図ではさらに、円の内側に各々の平行調を書き加えて、使用の便宜を図っている。

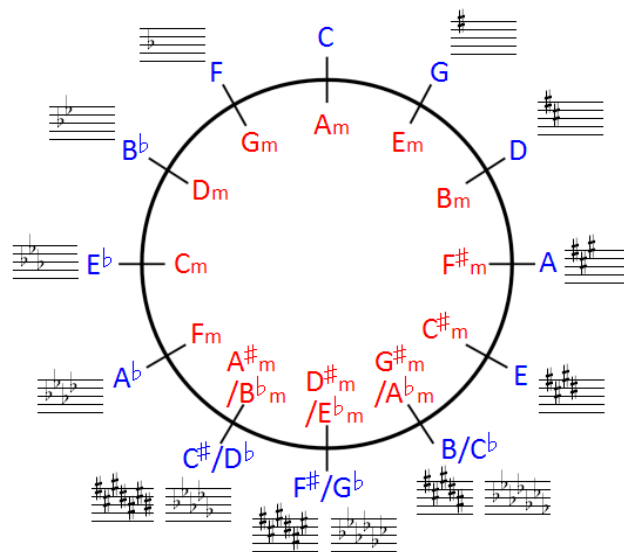


図 3.1.4 Circle of Fifths (5 度圏)

ある調に対して、属調，下屬調，平行調，同主調の4つをその調の近親調という。属調の平行調・同主調，下屬調の平行調・同主調を含めた8つを近親調とする場合もある。これ以外の調を遠隔調とする場合もある。

3.1.5. 和声法

和声法とは、和音を継続的に連結していく方法、および、それを基礎として旋律その他の音楽表現を形作っていく方法のことである。したがって、広義の和音をもつ音楽であれば、必ずそこに何らかの和声法を見出せる。しかし通常は長・短調組織にもとづく調的和声法のことをさす。これは17世紀から19世紀の音楽、すなわち調性音楽の根本的な構造原理をなすものであり、今日でも音楽理論や作曲法の学習において、和声法実習が広く行われている。ここでは部分的に説明をする。

長調・短調の音階構成音を根音として構成された和音を調固有和音という。根音の温度にしたがい、IからVIIの和音名称が与えられる。Iの和音を主和音、IVの和音を下屬和音、Vの和音を属和音とも呼び、この3つをその調の主要三和音という。

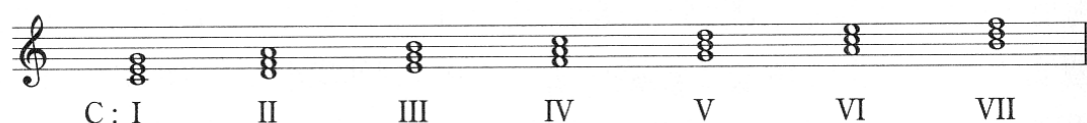


図 3.1.5 調固有和音（ハ長調の場合） [12]

各和音は調内の脈絡においてそれぞれに機能を持つ。これは言葉の文法における品詞や格に相当するもので、この重要性を持って調的和声法は機能と声とも呼ばれる。

トニック(T)機能	:	I, VI, III (およびそれぞれ七の和音)
ドミナント(D)機能	:	V, VII, III (およびそれぞれ七・九の和音)
サブドミナント(S)機能	:	IV, II (およびそれぞれの七の和音)

それぞれ冒頭に示した文字が、その機能を最も強く表す和音(主要三和音)である。その他の和音は代理機能と声と呼ばれることもある。なお、D機能においてはしばしばVよりもV₇の方がより強い機能を示す。

カデンツとは元来、フレーズの終止部分における和音や旋律の進行を意味するものだが、和声法の用語としては、機能に則った和音の連結法全般をさす。次の3パターンが基本で、これらが連鎖していくことにより、楽曲が構成される。

T→D→T 型	:	I → V → I, I → V ₇ → VI など
T→S→D→T 型	:	I → IV → V → I, IV → II ₇ → V ₇ → I など
T→S→T 型	:	I → IV → I など

3.2. Spiral Array Model

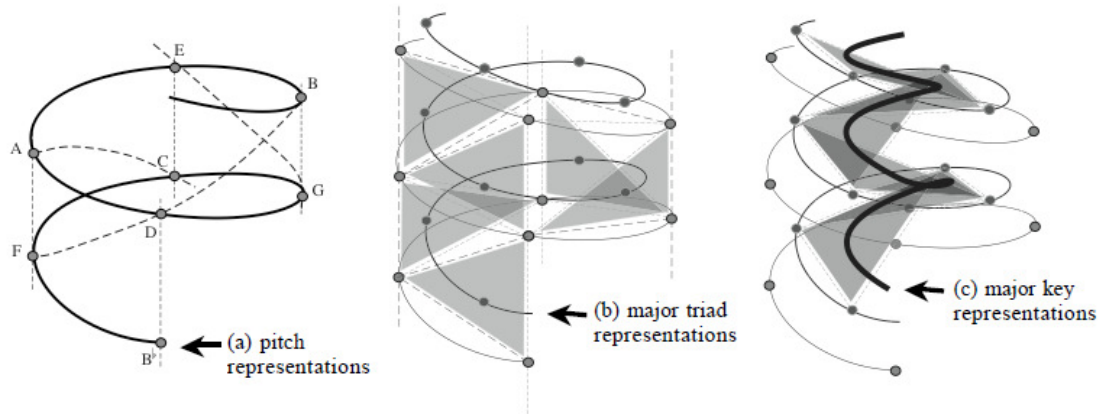


図 3.2.1 Spiral Array Model

Spiral Array Model[13,14]は上図のように、音高、和音、調を螺旋状に配置したモデルのことである。

図の (a) では音高が完全 5 度どうしで（螺旋上で）隣り合うように配置されている。垂直方向に隣り合うものは長 3 度の音程の関係にある。これがまず、音高螺旋である。

この螺旋内の点から、長三和音をつくる点を取り出すと図の (b) のような 3 角形の列を作ることが出来る。1 つの 3 角形は 1 つの長三和音を示している。これらの 3 角形が重なる（無限の点を共有する）ことはない。各 3 角形内の代表点（3 角形上にあり、3 点からの適当な重み付けで一意に決まる）が、各長三和音を表す点で、これらは音高螺旋の内側に螺旋を形成する。これが、和音螺旋である。

調にはそれぞれ、主要 3 和音（tonic, dominant, sub-dominant と呼ばれる和音で、ハ長調の場合、それぞれ C, G, F の和音がそれにあたる。）という重要な和音を 3 つ持つ。それらが表す点からさらに 3 角形をつくり、その代表点をその調を表す点として定義することが出来る。それらは和音螺旋のさらに内側にキー螺旋を形成する。

この 3 重の螺旋が Spiral Array Model であり、音高、和音、調の関係性を表現した重要なモデルである。以下にモデルを表す式を示す。

音高螺旋上の点、つまり音高の点は次式で表される。

$$P(k) = \begin{bmatrix} x_k \\ y_k \\ z_k \end{bmatrix} = \begin{bmatrix} r \sin(k \cdot \pi / 2) \\ r \cos(k \cdot \pi / 2) \\ kh \end{bmatrix} \quad (3.2.1)$$

k は音高を表すインデックス, h は螺旋の上下の開き具合を表す定数である. 音高の点は完全五度の間隔で並んでいるので, ある音高 $P(k)$ を根音としたとき $P(k+1)$, $P(k+4)$ によって 3 和音の一つを作ることができる. 空間上でこの 3 点は 3 角形を成し, その重み付き重心を和音の点と定める.

$$C(k) = w_1 \cdot P(k) + w_2 \cdot P(k+1) + w_3 \cdot P(k+4) \quad (3.2.2)$$

$$w_1 \geq w_2 \geq w_3 > 0, \quad \sum_{i=1}^3 w_i = 1$$

w_i は重みであり, 和音を構成する 3 つの音が和音に対してどれほどの重要度を持っているかを表している. 楽典上, 根音 \geq 5 度音 \geq 3 度音という重要性を持っているため (3.2.3) 式のような関係性を盛り込んである. ある和音 $C(k)$ を tonic と考えたとき, $C(k+1)$ を dominant, $C(k-1)$ を sub-dominant とするような調性が一つ存在する. $C(k), C(k+1), C(k-1)$ を頂点とする 3 角形を考え, その重み付き重心をその調の点と定める.

$$T(k) = w_1 \cdot C(k) + w_2 \cdot C(k+1) + w_3 \cdot C(k-1) \quad (3.2.2)$$

w_i は重みであり, 調の主要三和音のうちどれがその調にとって重要かを反映している. 楽典上, tonic \geq dominant \geq sub-dominant という重要性を持っているため式のような関係性を盛り込んである.

3.3. Doubly Nested Circle of Fifths

Doubly Nested Circle of Fifths [3]とは, 音楽知識の 1 つで, 下の図で示されるような二重入れ子状の円のことをいう. 赤い円である大文字が major (長三和音), 青い円である小文字が minor(短三和音)を表している.

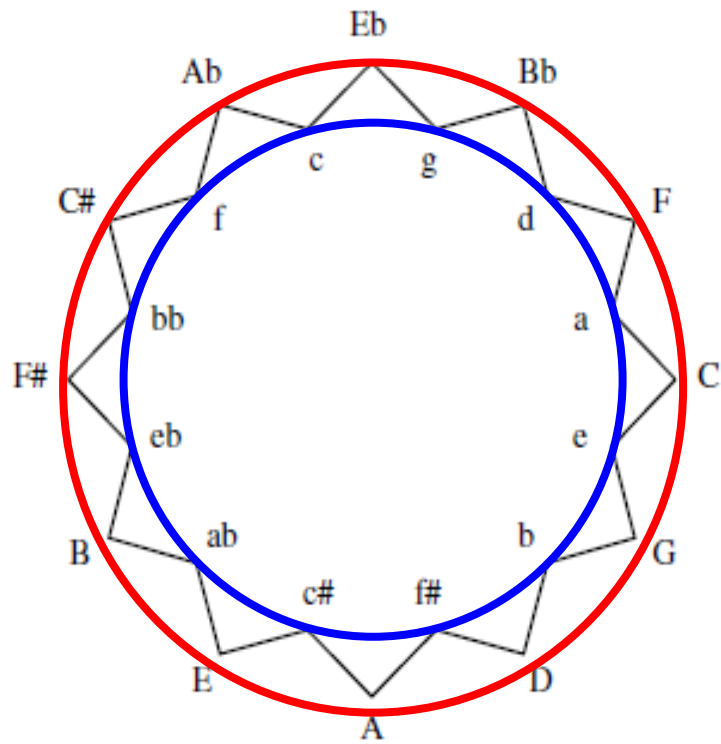


図 3.3.1 Doubly Nested Circle of Fifths[3]

Doubly Nested Circle of Fifth では隣り合った和音同士の構成音を比べると，使用される 3 音中 2 音が等しいという類似性を持っていることがわかる．つまり，隣り合う和音は似ているが，対角上にある和音は似ていないといえる．

4. 従来研究

4.1. 音楽情報処理の発展

ハードウェアが進歩し、コンピュータの処理速度向上とメモリ・ハードディスクの大容量化によって、音楽情報処理の研究はここ 10 数年の間に大きく展開した。デジタル信号処理技術の発展で、従来は思い付かなかった作業の計算が可能となり、それが新しい手法を生み出すきっかけとなっている。特に統計的手法が広く導入され、隠れマルコフモデル (Hidden Markov Model: HMM) 等の確率モデルに基づく手法や、最尤推定、ベイズ推定などを活用した様々な手法が利用されている。また、研究対象も大きな広がりを見せている。以前は音響信号の個々の構成音を分離・抽出する音源分離、音源の種類とその音高を推定する音源同定、そしてそこから楽譜を作成する自動採譜システムの研究が主流であったが、最近では人間の音楽理解の仕方（構成音の分離信号や楽譜に基づいて音楽を理解しているわけではない）を考慮した上で、ビートトラッキングやメロディ抽出、類似楽曲の検索といったテーマが発表されている。研究事例が多いためここではその一部を取り上げて紹介する。

4.2. 音高推定

音源の基本周波数を推定する処理を音高推定と呼ぶ。1999 年、後藤らによって音源数を仮定しない音高推定手法 PreFEst (Predominant-F0 Estimation Method) が提案された[15]。この手法は帯域制限された混合音の中で最も優勢な音高を推定する手法の 1 つである。音高ごとに高調波構造の形状を表す確率分布を用意し、入力音の周波数構造をそれらの混合分布としてモデル化する。そして EM アルゴリズムを用いて最大事後確率推定を行うことで混合音の各構成音の音量（重み）および高調波構造の形状が求められる。その他、2003 年に Klapuri が混合音の基本周波数推定を実現するために、優勢な音高を求めその高調波成分の除去を繰り返し行うことで混合音の構成要素を順次求めていく手法を提案している[16]。また、音の波形の重み付き和として時間領域でモデル化を行い MCMC (Markov chain Monte Carlo) アルゴリズムを用いて同時発音数、各音を構成する周波数成分数、音高、音量などのモデルパラメータを推定する手法[17]や、周波数成分を高調波構造の拘束の下でクラスタリングする問題と定式化し、赤池情報量基準 (AIC) を最小にするクラスタ数を求めて、各クラスタの重心 (= 音高) と重み (= 音量) を推定する手法[18]などがある。

4.3. 調推定

調の推定では、和音認識手法と同様のアプローチが多い。ここでは HMM を利用し、楽曲のクロマベクトルなどと事前に用意してあるテンプレートとの類似性を測るといった手法がみられる。これも何かしらの事前情報を多く必要とし、[19]などは音楽心理学に基づいた膨大なプロファイルを必要としている。

4.4. 和音認識

和音認識においては和音名に加え、和音境界を求めることが課題となる。

特徴量には一般的にクロマベクトルが用いられる。Fujishima らは[1]においてクロマベクトルを提案し、以後クロマベクトルを改善するアプローチも多い。例えば、[10]ではクロマベクトルを線形で重み付けすることにより改善を図った。また、音響信号の調波音を強調したり、チューニングを行ったりしてクロマベクトルの改善を図るものもある[20]。他にも、周波数スペクトルのフレームは理想的な音パターンの線形結合で表せると仮定をし、NNLS (Non-Negative Least Square) 問題を解いて得られる NNLS chroma を提案するものもある[21]。また、従来手法における認識の手法は、次の 4 つのパターンに大別される。

(1) テンプレートに基づく手法

これは検出したい和音をテンプレートで定義をし、フレームごとに最もデータに合うテンプレートの和音を選ぶ手法である。[22]では、6 倍音まで考慮した和音のテンプレートを用いて認識を行っている。

(2) 音楽情報に基づく手法

これは、システムの構造で音楽理論の情報を利用する手法である。具体例としては、和音やリズム構造間の遷移に音楽知識を反映したパラメータでモデル化しているものが挙げられる。Bello らの研究[3]では HMM の初期値に音楽的知識をモデルとして取り込み、認識率を改善した。

(3) データに基づく手法

これは、正解データを利用し、音楽データにあうシステムを構築している。すべてのパラメータは学習で評価を行っている。和声モデルに HMM を用いた研究では Sheh[2]らのものがあり、彼らは音楽信号と和声進行から EM アルゴリズムを用いて推定を行った。また、[4]ではラベルデータはシンボル音楽ファイルの音響解析で求め、特徴ベクトルは同じシンボルデータである合成された音から計算し、それらを使って学習を行っている。

(4) ハイブリッド手法

これは、(1)～(3)の手法を組み合わせた手法である。[5]では、学習データを使用し、それに音楽理論を組み合わせている。

他にも、和音区間推定・和音同定に加えて、調性の推定を行うものもみられる[23]。調性も、和音とは密接な関係があり、これを加えることでさらにロバストな推定を目指している。

和音を推定するソフトウェアに河合楽器のバンドプロデューサー[24]がある。これは、はじめに BPM を計算するためにテンポをキーで入力するため、テンポが途中で変わったり、BPM が少しずれたりすると、曲の途中から和音のご認識が増えてしまう。また、和音変化の激しいものは誤検出が目立つ。これらの解決方法として、バンドプロデューサーは人手で和音名を直すことが可能になっている。

5. 提案手法

5.1. DNCOF ベクトル

5.1.1. 生成フロー

はじめに，DNCOF ベクトルの生成フローを以下に示す．

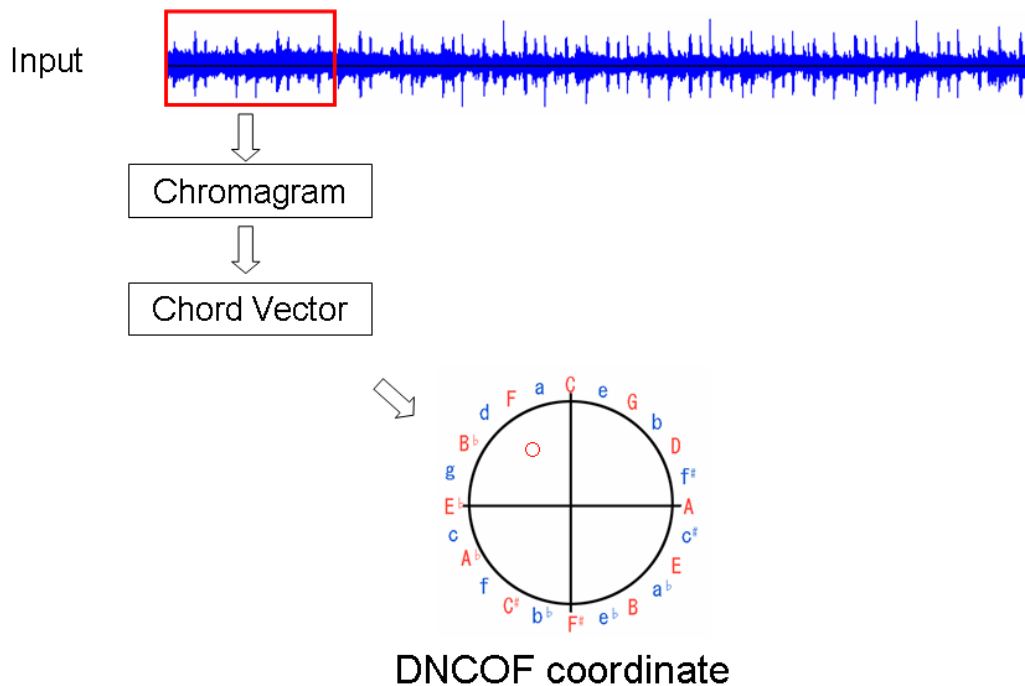


図 5.1.1 DNCOF 生成フロー

処理は 11250Hz にダウンサンプリングした wav 信号を，複数のフレームに切り出して行う．各フレームは 8192 個のサンプルから構成され，1つのフレームに対して1つの DNCOF ベクトルが求められる．

5.1.2. クロマベクトル

はじめに 2.5 を参考に定 Q スペクトルを求める．なお，ここでは定 Q 変換は 96Hz から 5250Hz の範囲で行う．次に，2.7 を参照しチューニングを行った 12bin クロマベクトル *Chroma* を計算する．

5.1.3. コードベクトル

コードベクトルは major と minor の 24 種の和音がどのような尤度を持つかを表す 24 次元のベクトルである。この尤度の高いものが、そのフレームにおいて尤もらしい和音となる。コードベクトル $C(t)$ は次式で定める。

$$C(t) = \begin{bmatrix} C_C(t) \\ \vdots \\ C_{B_{\min}}(t) \end{bmatrix} \quad (5.1.1)$$

$$C_{P_n}(t) = \sum_{i=0}^{11} w_i Chroma_{P_{(i+n) \bmod 12}}(t) \quad (5.1.2)$$

$$P_0 = C, P_1 = C\#, \dots, P_{11} = B \text{ または } P_0 = C_{\min}, P_1 = C\#_{\min}, \dots, P_{11} = B_{\min}$$

コードベクトルの各要素は、12bin クロマベクトル $Chroma$ を入力とし、その重み付け和とする。和音の構成音のうち、各音の重要度は異なり、この重要度を反映させたものが w_i となる。例えば、Cmajor の基本構成音は C, E, G であるが、これらの倍音を考慮すると D や B など他の音も含むことになる。そこで本研究は[22]を参考に 6 倍音まで考慮した重み付けをする。ここでは、 i 番目の倍音には 0.6^{i-1} だけ振幅を加え、最終的に振幅の合計が 1 になるよう正規化を行う。結果として、図のように重みを定めた。

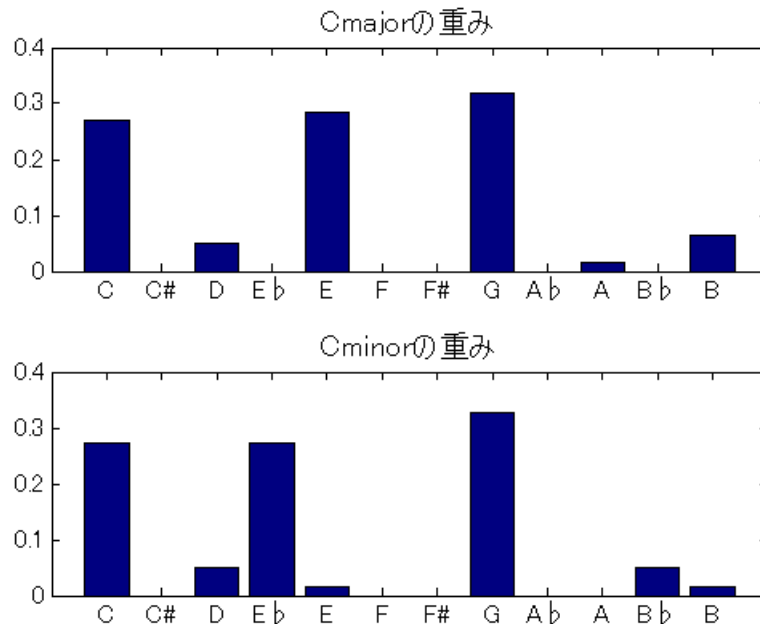


図 5.1.2 コードベクトルの重みの例 (上: Cmajor, 下: Cminor)

5.1.4. DNCOF ベクトル

ここで、2.2 節で得られたコードベクトルを図 3 の平面に射影する。この手順は次の通りである。

(1) DNCOF を円に見立て、major と minor それぞれ各和音の方向に向かうベクトル u を用意

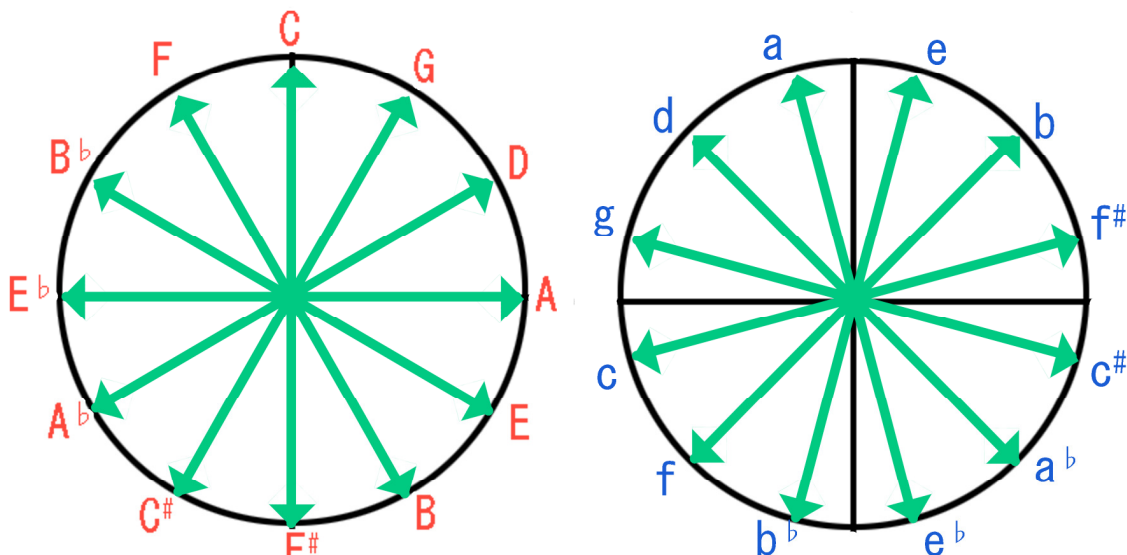


図 5.1.3 各和音の方向に向かうベクトル u

(2) 1 にある調における和音の重み w_{key} をかける

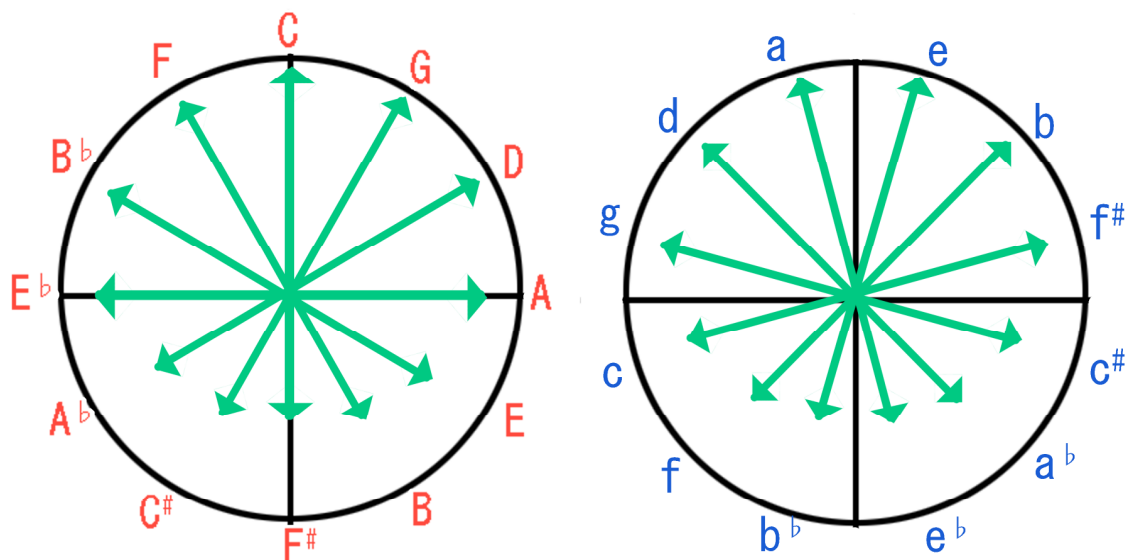


図 5.1.4 和音の重みをかけた状態

ここでは、単位ベクトルに対し、調性を反映させるため、ある調で用いられる和音(調固有和音)の成分が強くなるように、正規分布で重み付けした。

(3) それぞれにおいて，コードベクトル $C(t)$ の要素倍する

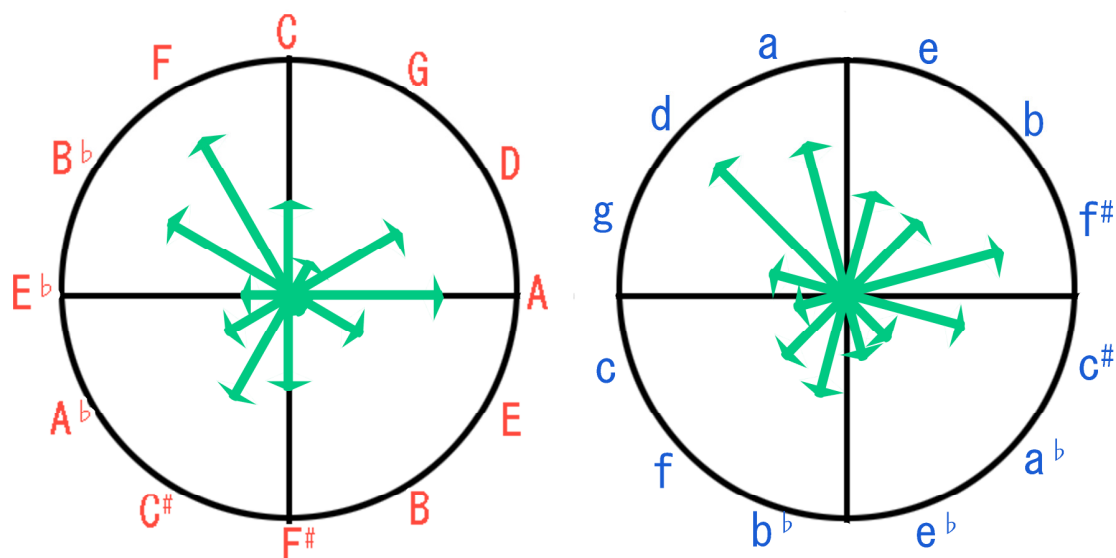


図 5.1.5 それぞれコードベクトルをかけた様子

(4) 成分が閾値より大きいものだけ残す

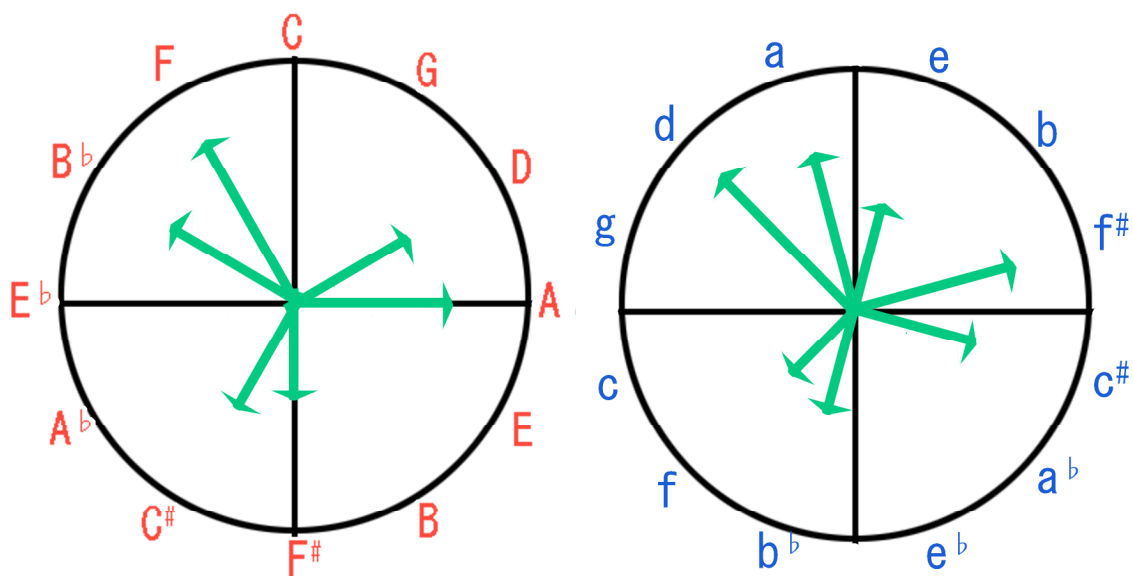


図 5.1.6 閾値より大きい成分

この成分の閾値は，実験的に定めた。

(5) major と minor それぞれで求まったベクトルの大きさを比較し，大きい方を DNCOF ベクトルとする

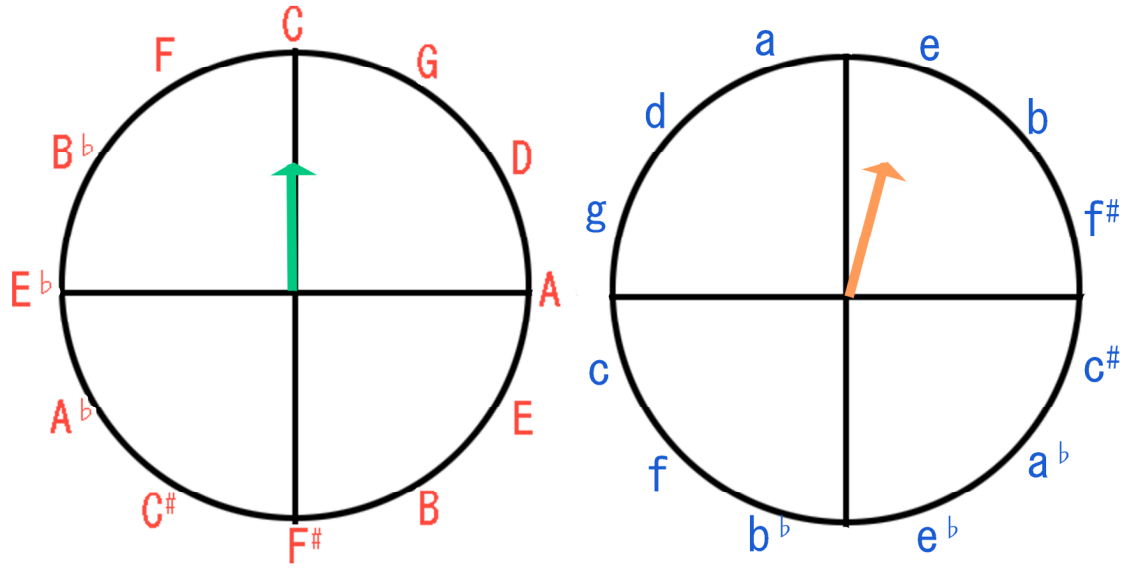


図 5.1.7 DNCOF ベクトルの決定

これらの手順を式で表すと以下の通りになる. はじめに各和音の方向に向かうベクトル u は次式で表される.

$$u_{maj} = \begin{bmatrix} w_{key\ Cmaj} \cos\left(\frac{\pi}{2} - 0 \times \frac{\pi}{24}\right), \dots, w_{key\ Fmaj} \cos\left(\frac{\pi}{2} - 22 \times \frac{\pi}{24}\right) \\ w_{key\ Cmaj} \sin\left(\frac{\pi}{2} - 0 \times \frac{\pi}{24}\right), \dots, w_{key\ Fmaj} \sin\left(\frac{\pi}{2} - 22 \times \frac{\pi}{24}\right) \end{bmatrix} \quad (5.1.3)$$

$$u_{min} = \begin{bmatrix} w_{key\ Emin} \cos\left(\frac{\pi}{2} - 1 \times \frac{\pi}{24}\right), \dots, w_{key\ Amin} \cos\left(\frac{\pi}{2} - 23 \times \frac{\pi}{24}\right) \\ w_{key\ Emin} \sin\left(\frac{\pi}{2} - 1 \times \frac{\pi}{24}\right), \dots, w_{key\ Amin} \sin\left(\frac{\pi}{2} - 23 \times \frac{\pi}{24}\right) \end{bmatrix} \quad (5.1.4)$$

w_{keyi} : ある調 key における和音 i のマスク (正規分布)

このベクトル u と, コードベクトル $C(t)$ の major 成分と minor 成分をそれぞれ掛け合わせ, 大きさの大きいほうを DNCOF ベクトルと定める.

$$DNCOF(t) = \begin{bmatrix} x(t) \\ y(t) \end{bmatrix} = \begin{cases} u_{maj} C_{maj}(t) & \text{if } |u C_{maj}(t)| \geq |u C_{min}(t)| \\ u_{min} C_{min}(t) & \text{otherwise} \end{cases} \quad (5.1.5)$$

$$C_{maj}(t) = [C_{Cmaj}(t) \quad C_{Gmaj}(t) \quad \dots \quad C_{Fmaj}(t)] \quad (5.1.6)$$

$$C_{min}(t) = [C_{Emin}(t) \quad C_{Bmin}(t) \quad \cdots \quad C_{Dmin}(t)] \quad (5.1.7)$$

ここで、 $DNCOF(t)$ を極座標表示すると、

$$\begin{bmatrix} r(t) \\ \theta(t) \end{bmatrix} = \begin{bmatrix} [x(t), y(t)] \\ angle([x(t), y(t)]) \end{bmatrix} \quad (5.1.8)$$

となる。 $angle(v)$ は y 軸の正方向から時計回りに測った角度とし、 $-\pi < angle(v) \leq \pi$ である。 $r(t)$ はベクトルの大きさであり、ある方向にコードベクトルの値が偏るほど大きくなる。つまり、似た和音の尤度が高いということであり、そのフレームにおいて和音の純度が高いとみなすことができる。また、 $\theta(t)$ はベクトルの偏角を表し、離散的な 24 和音の方向を示すのではなく、中間の方向を連続的に表し、和音のタイプを示す。ここではその中間の方向というものを、「両側にある 2 つの和音をそのない割合で混ぜたような和音のタイプ」というようにとらえる。例えば、C major と A minor のちょうど中間を指すような DNCOF ベクトルがあった場合には、「C,E 音をメインに用い、G, A が半々で現れるような和音情報」といえる。

5.1.5. DNCOF 列

フレームごとに DNCOF ベクトルを求めていくと、楽曲全体にわたり図のような DNCOF ベクトルの時系列が得られる。図では縦軸が和音の種類、横軸はフレーム番号を示している。プロットの濃さは和音の純度を示しており、ある和音の方向にコードベクトルの値が偏るほど純度が高くなる。これを本研究では DNCOF 列と呼ぶ。この DNCOF 列を楽曲の和音情報とみなし、解析を試みる。

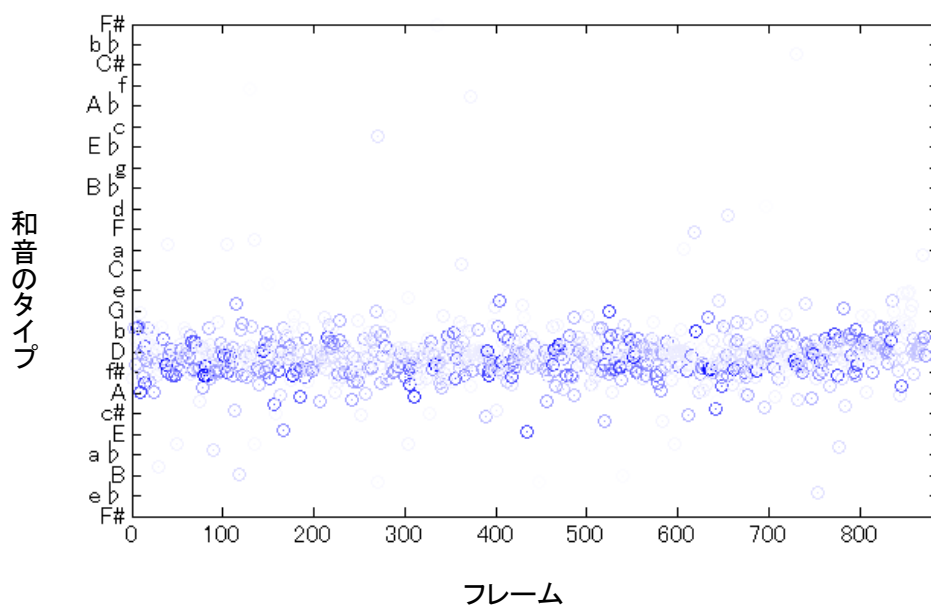


図 5.1.8 The Beatles のアルバム Beatles for Sale より”Eight Days A Week”の DNCOF 列
(プロットの濃さ：和音の尤度)

5.2. 和音認識手法

5.2.1. Hidden Markov Model

Hidden Markov Model (HMM) は確率モデルの一つである IBM により考案された。「システムがパラメータ未知のマルコフ過程である」と仮定し、観測可能な情報からその未知のパラメータを推定する。音声認識、ゲノミクス、形態素解析（自然言語処理）などに応用され、連続的かつ伸縮しうる信号列のパターン抽出には適しているが、反面、長い距離をはさんで呼応しているような信号列からのパターン認識には、間の距離の長さに応じて状態数を増やす必要があり、計算量の観点から実用的ではない。また、局所最適に陥りやすいため、対象に応じて適切なパラメータの初期値を設定する必要がある。

HMM の構成要素は以下の通りである。

(a) 状態集合: $\Sigma = \{S_i \mid 0 \leq i \leq M\}$

想定するモデルがとりうる状態を定義する。

(b) 状態遷移確率: $A = \{a_{ij} = P(S_i S_j \mid 0 \leq i, j \leq M)\}$

状態間の遷移しやすさを規定する。全ての i, j について a_{ij} を定め、遷移しえない状態の組み合わせの状態遷移確率は 0 とする。

(c) 出力確率: $B = \{f(o; S_i) \mid 0 \leq i \leq M\}$

状態 S_i において、信号 o を出力する確率。信号が有限個の確率事象の生起である場合には、確率値

$$f_i(o) = P(o \mid S_i) \quad \text{ただし} \quad \sum_o P(o \mid S_i) \quad (5.2.1)$$

として定義される。

(d) 出力信号系列: $o(t) (t = 0, \dots, T)$

HMM の遷移過程において出力された信号の系列。各時刻単位に出力される信号は 1 つずつである。

(e) 状態系列: $s(t) (t = 0, \dots, T)$

時刻 n における状態の系列を表す。

5.2.2. Baum-Welch アルゴリズム

HMM を学習するためには、HMM を構成する要素のうち状態遷移確率、出力確率の2つを、与えられた学習をもとに定める必要がある。両者の学習規範としては、最尤基準が適用されることが一般的である。HMM の学習では、学習用に与えられた出力信号系列が、どのような状態系列をたどって、出力されたものかが観測できないことがあるので、与えられた出力信号系列の下で状態遷移の期待値を求めて、この期待値に基づきパラメータの推定を行う。

Baum-Welch アルゴリズムは、観測系列の生成確率を最大にするモデル λ のパラメータの局所的最適値を求める方法である。

モデル λ が観測系列 $O = o_1, o_2, \dots, o_T$ を生成する場合において、時刻 t で状態 i から状態 j に遷移する確率 $\xi_t(i, j)$ を次のように定める。

$$\begin{aligned}\xi_t(i, j) &= P(s_{t-1} = i, s_t = j \mid O, \lambda) \\ &= \frac{\alpha_{t-1}(i) a_{ij} b_{ij}(o_t) \beta_t(j)}{P(O \mid \lambda)} \quad (1 \leq t \leq T)\end{aligned}\tag{5.2.2}$$

ここで、シンボルの生成過程で、時刻 t で状態 j にいる確率 $\gamma_t(j)$ を定義する。

$$\begin{aligned}\gamma_t(j) &= P(s_t = j \mid O, \lambda) \\ &= \sum_{i=1}^N \xi_t(i, j) \quad (1 \leq t \leq T)\end{aligned}\tag{5.2.3}$$

この $\gamma_t(j)$ と $\xi_t(i, j)$ とからモデル λ の再推定を次のように行う。

(1) 初期状態確率

$$\bar{\pi}_i = \gamma_0(i) = \frac{\alpha_0(i) \beta_0(i)}{P(O \mid \lambda)} \quad (1 \leq i \leq N)\tag{5.2.4}$$

(2) 状態遷移確率

$$\bar{a}_{ij} = \frac{\sum_{t=1}^T \xi_t(i, j)}{\sum_{t=1}^T \gamma_{t-1}(i)} = \frac{\sum_{t=1}^T \alpha_{t-1}(i) a_{ij} b_{ij}(o_t) \beta_t(j)}{\sum_{t=1}^T \alpha_{t-1}(i) \beta_{t-1}(i)}\tag{5.2.5}$$

(3) シンボル出力確率

$$\bar{b}_{ij}(o_t) = \frac{\sum_{t=1}^T \xi_t(i, j)}{\sum_{t=1}^T \xi_t(i, j)} = \frac{\sum_{t=1}^T \alpha_{t-1}(i) a_{ij} b_{ij}(o_t) \beta_t(j)}{\sum_{t=1}^T \alpha_{t-1}(i) a_{ij} b_{ij}(o_t) \beta_t(j)} \quad (5.2.6)$$

再推定された $\bar{\lambda}$ の評価は次のようになる。

- ① $\bar{\lambda} = \lambda$: (局所的な) 収束状態
- ② $P(O|\bar{\lambda}) > P(O|\lambda)$: シンボル系列 O を出力するより最適なモデル λ を推定

Baum-Welch アルゴリズムは、学習データの尤度を最大にするようにパラメータを学習する。ただし、基本的には **gradient** 学習によるパラメータ収束の学習方法であるため、**local maximum** の方向にしか学習は進まない。そのため初期値が重要になる。音響モデルでは通常 **left-light** モデルが使用されるため あまり問題にならないが、全ての状態が全ての状態に接続される **Ergodic HMM** では、この初期値が問題になる。

5.2.3. 和音認識のための HMM 設定

以下の図のように，和音認識を行うための HMM 設定として前項の (a) ~ (e) を定める．

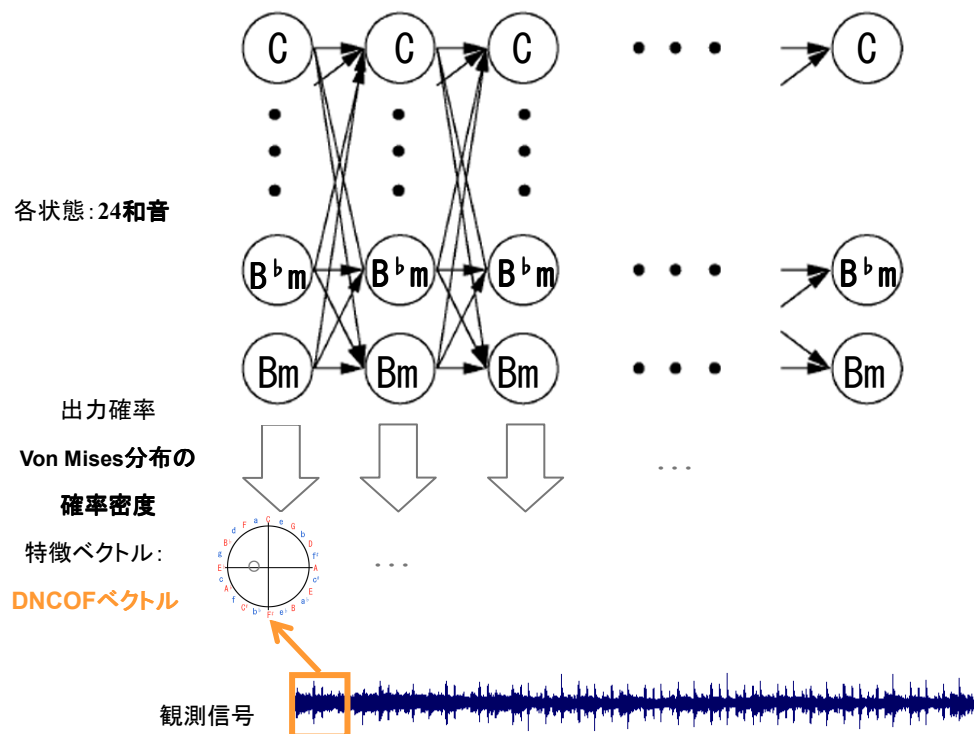


図 5.2.1 DNCOF ベクトルを用いた HMM のモデル

(a) 状態集合 : $\Sigma = \{S_i \mid 0 \leq i \leq M\}$

各状態を各和音 Cmajor, C#major, ..., Bminor に対応させる．つまり全 24 状態が定義される．また，ここでは初期状態，終了状態は特に規定しない．

(b) 状態遷移確率 : $A = \{a_{ij} = P(S_i S_j \mid 0 \leq i, j \leq M)\}$

各状態間の遷移確率，すなわち各和音間の転調のしやすさを定義する．ここでは，初期値を[3]に基づき DNCOF 順で設定し，学習を行う．

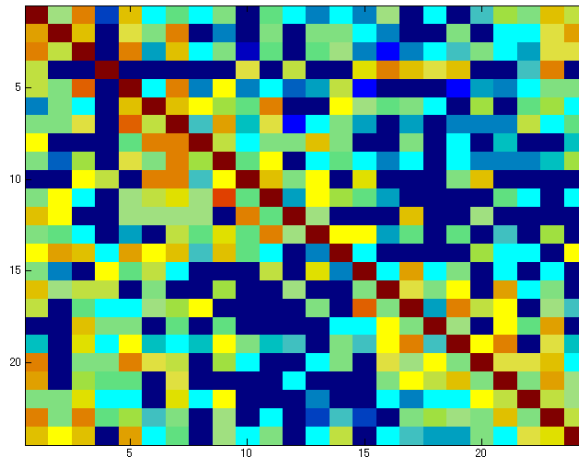


図 5.2.2 状態遷移確率の例 (180 曲学習後)

(c) 出力確率 : $B = \{f(o; S_i) \mid 0 \leq i \leq M\}$

出力確率 $f(o; S_i)$ は状態 S_i において信号 o を出力する確率である. 信号 o は DNCOF 列の 1 プロット, ある座標 $[x, y]$ とする. ここではある和音に対して, DNCOF プロット $[x, y]$ がどのくらいの確率で出現するかに基づいて定義する. 本研究では von Mises 分布[25]の確率密度関数を用いて出力確率を定めた.

(d) 出力信号系列 : $o(t) (t = 0, \dots, T)$

ここでは DNCOF 列そのものに相当する.

(e) 状態系列 : $s(t) (t = 0, \dots, T)$

状態系列は状態, つまり和音名の系列を表している. (d) の出力信号系列から, この状態系列を推定することが目的となる.

5.2.4. ビタビアルゴリズム

最尤な和音遷移を決定するにはビタビアルゴリズムを用いる。モデルパラメータが既知の時に、与えられた配列を出力した可能性（尤度）が最も高い状態列（最尤状態列）を計算するアルゴリズムである。動的計画法の一種である。

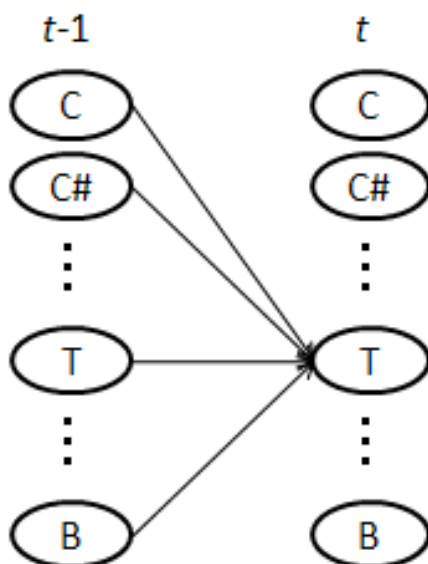


図 5.2.3 最尤パス推定

時刻 t においてある和音 T を表した状態 S_T であったとする。このときの確率は、前の状態から S_T へ遷移して、状態 S_T で $o(t)$ を出力するのであるから、

$$P(S_T | t) = P(S_T | t-1) P(S_i | S_T) f(o(t); S_T) \quad (5.2.7)$$

と表せる。ここで、 S_i は複数通りあるが、 $P(S_T | t)$ を最大にするようなパスのみ記憶しておけばよい。

$$P(S_i | t) = \max \{ P(S_T | t-1) P(S_j | S_T) f(o(t); S_T) \} \quad (5.2.8)$$

$$J = \operatorname{argmax} \{ P(S_j | t-1) P(S_j | S_T) f(o(t); S_T) \} \quad (5.2.9)$$

つまり、 $P(S_i | t)$ を最も高い確率にするには $S_{j(t,i)} \rightarrow S_i$ という遷移を行えばいい、ということがわかる。このようにして、最終フレームまで $P(S_i | t)$ を計算し、さらにその中で最大の確率を持つ状態を決める。

$$I = \operatorname{argmax} \{ P(S_i | t_{last}) \} \quad (5.2.10)$$

最終状態で最尤な状態は S_I である。このとき $P(S_I|t_{last})$ を出力するような状態は $S(t_{last}-1) = S_{j(t_{last},I)}$ であることがわかる。このときさらに時刻 $t_{last}-1$ での最尤な状態を決めることができ、以上を繰り返すことで全体を通しての状態遷移の最尤パスを決めることができる。これがビタビアルゴリズムの概要である。

6. 実験

6.1. DNCOF ベクトルを用いた和音認識

ここで, DNCOF ベクトルを特徴ベクトルとして, The Beatles のアルバム(Please Please Me, Beatles For Sale)の 28 曲に対し, 和音認識を試みた. 実験パラメータは以下のように定めた.

表 6.1.1 実験環境

楽曲形式	wav データ, リニア PCM
実験楽曲	Beatles のアルバムより全 28 曲 (Please Please Me, Beatles For Sale)
学習曲数	The Beatles のアルバム 179 曲の正解データ[26]を利用
扱う和音	major と minor の 24 種
サンプリング周波数	11025 Hz
チャンネル数	1 (モノラル)
A4 の周波数	440 Hz
ウインドウサイズ	8192 サンプル (0.74 秒)
オーバーラップサイズ	ウインドウサイズの 4 分の 1
定 Q 変換における 周波数の範囲	96Hz~5250Hz

なお, 本研究で扱う和音は major と minor の 24 種のみとし, 正解データは[]に公開されているラベルデータを使用した. このラベルには major, minor 以外の和音も含まれるので, 根音と第 3 音により major と minor に分けた. 例えば, Csus4 や Caug は Cmaj に, Cmin7 や Cdim は Cmin に分類した.

また, 認識の正解率は次式のように定めた.

$$\text{正解率} = \frac{\text{正しく出力されたフレーム数}}{\text{全フレーム数}} \quad (6.1.1)$$

6.2. 比較手法

DNCOF ベクトルはクロマベクトルの抽象度を挙げたものであるため、情報量が削減される前の精度を調べるために、比較手法として特徴ベクトルにクロマベクトルを用いて実験を行った。HMM のモデルは次のように定めた。

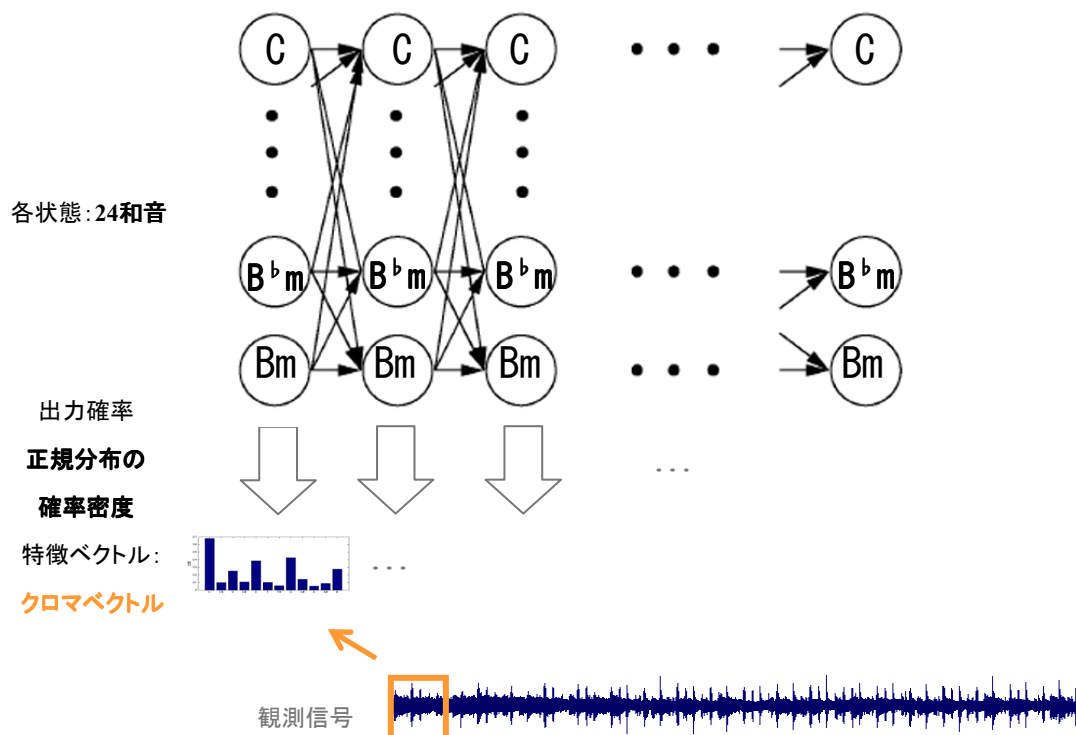


図 6.2.1 クロマベクトルを用いた HMM のモデル

DNCOF ベクトルとの違いは特徴ベクトルだけであるため、認識率によりクロマベクトルと DNCOF ベクトルそのものの精度比較ができる。

6.3. クロマベクトルへの情報応用

クロマベクトルでは、根音が同じである major と minor は 3 音の構成音のうち 1 音が違うだけなので、誤まって認識される可能性が高い。例えば、正解は C major であるのに、認識結果が C minor となることが考えられる。しかし、DNCOF ベクトルでは、DNCOF 順で両者が離れているため、この major と minor の間違いを防げると期待できる。

ここで、DNCOF ベクトルによる認識で求めた和音情報をクロマベクトルの認識への応用を試みる。応用方法は次の通りである。

(1) DNCOF ベクトルから 3 つの和音候補を求める

(2) 1 の候補とクロマベクトルで求めた和音正解候補を比較し、1 の候補に近くなるよう順位を入れ替える

例えば、あるフレームにおいて DNCOF ベクトルによる認識で求めた 1～3 位の候補が D, F#m, B だとする。そして、そのフレームでのクロマベクトルによる候補が E, D, A だとする。ここで、E, D, A のうち DNCOF ベクトルでの候補に DNCOF 順で一番近い和音は D になるので、クロマベクトル候補における順位を入れ替え、出力されるラベルは D となる。

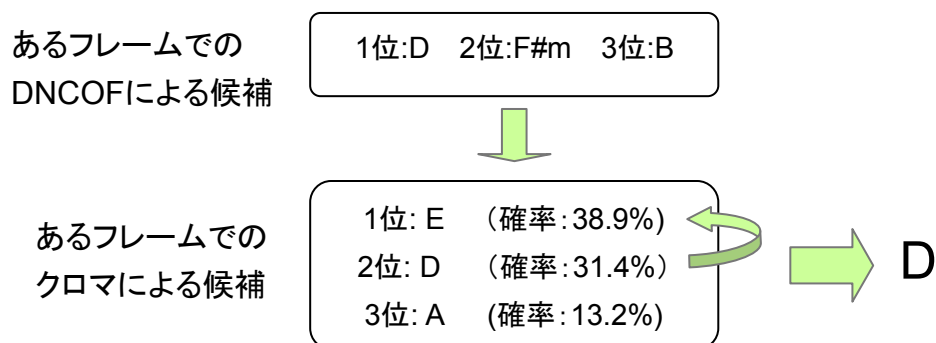


図 6.3.1 DNCOF 情報応用方法

この処理を各フレームに対して行う。ただし、クロマベクトルの候補の 1 位と 3 位の確率の差が閾値より大きかった場合には、クロマベクトルの 1 位の候補のラベルをそのまま出力する。

6.4. 実験結果

それぞれの認識結果を表 6.4.1 に示す。

表 6.4.1 特徴ベクトルの違いによる認識結果

特徴ベクトル	Please Please Me (14 曲の平均)	Beatles For Sale (14 曲の平均)	28 曲全体
DNCOF ベクトル	35.3%	38.5%	36.9%
クロマベクトル[27]	41.3%	47.8%	44.5%
クロマベクトル + DNCOF 情報	42.3%	48.1%	45.2%

DNCOF ベクトル自体の精度はクロマベクトルよりも低くなったが、DNCOF 情報を応用することによって、クロマベクトルの認識率が上がった。次に、それぞれの結果のプロ

ットを示す.

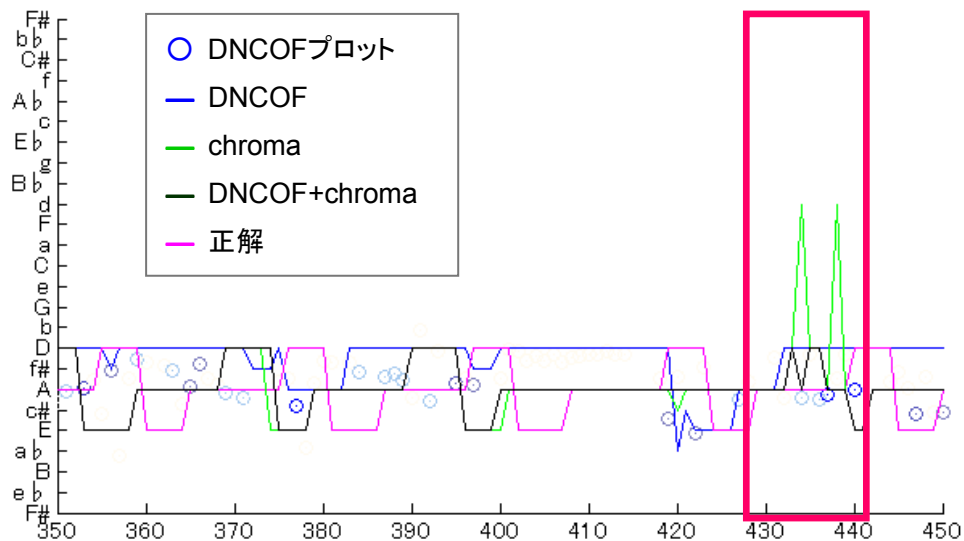


図 6.4.1 The Beatles のアルバム Beatles for Sale より”Words of Love”の
350～450 フレーム目の結果

赤色で囲まれた区間では、クロマベクトルによる認識では Dm であるが、DNCOF 情報を用いたクロマベクトルでの認識では、D になっており、major と minor の一音違いの誤認識が改善されている。また、DNCOF プロットは正解データに近い位置にあることが確認できる。

しかし、DNCOF プロットは局所的になっているため、DNCOF 順で離れた和音への遷移の場合ではプロットは正解から離れてしまった。これは、DNCOF ベクトル生成時において調固有和音を強調するような合成を行っており、ばらつきが抑えられてしまって調固有和音以外への対応ができていないことが原因である。実際、DNCOF ベクトルはクロマベクトルよりも情報量が削減されてしまっているため、和音と関連の深い音楽要素を反映させる際には情報が偏らないように注意が必要である。

さらに、DNCOF 情報を用いたクロマベクトルの認識では各フレームにおいて順位を入れ替えてしまったので、前後のフレームとの関係が考慮されていない。これは、和音区間やビートトラックで区間を求めてから、その区間ごとに候補を入れ替えることで対処できると考えられる。また、和音の前後関係を反映させるという点で和声の考え方を取り入れることも有効だといえる。

7. 結論

7.1. まとめ

本研究は、音楽知識である Doubly Nested Circle of Fifths に基づいた和音情報である DNCOF ベクトルを提案した。DNCOF ベクトルは和音の尤度を表す「大きさ」と和音のタイプを表す「角度」の情報を持っており、楽曲全体にわたって計算すると DNCOF ベクトルの時系列である DNCOF 列を得られる。

DNCOF 列を用いた和音認識実験では、DNCOF ベクトル単体ではクロマベクトルに比べて認識率は下がってしまったが、DNCOF ベクトルで得られた和音情報をクロマベクトルへ応用することによって、少しではあるがクロマベクトルの major と minor の誤認識を改善できた。今後、DNCOF ベクトルのさらなる精度改善により、DNCOF ベクトルは新たな和音情報となることが期待される。

7.2. 今後の展望

DNCOF プロットはおおまかに和音が求まるものの、DNCOF 順で離れた和音間の遷移の場合、認識率が落ちてしまうことがわかった。これは、プロットが調固有和音を強調するような合成になってばらつきが抑えられ、和音タイプでも狭い範囲に収まってしまっているのが原因である。今後は、調固有和音を強調するだけでなく、DNCOF ベクトル自身の精度を上げる手法を検討する必要がある。

また、和音は他の音楽要素とも深く関わっているため、他の音楽要素と組み合わせることも検討している。例えば、和声の知識を取り入れたり、和音区間を求めたりして改善を試みたい。

そして、今回[11]のクロマベクトルで実験を行ったが、DNCOF ベクトルの精度はクロマベクトルにも依存するので MIREX2010[28]で一番高い認識率であった NNLS chroma[21]への応用も検討している。

付録

von Mises 分布

von Mises 分布[25]とは、円周上に定義された連続型の確率分布である。方向統計学における代表的な分布モデルであり、確率変数を角度の関数として表す分布モデルなどに使われる。

von Mises 分布の分布関数 $F(\theta)$ と確率密度関数 $f(\theta)$ は以下の式で表される。

$$F(\theta) = \{2\pi I_0(\kappa)\}^{-1} \left[\theta I_0(\kappa) + 2 \left\{ \sum_{j=0}^{\infty} \frac{I_j(\kappa) \sin(j(\theta - \mu))}{j} \right\} \right]$$
$$f(\theta) = \frac{1}{2\pi I_0(\kappa)} \exp(\kappa \cos(\theta - \mu)) \quad (\text{ap.1})$$

$$0 \leq \theta \leq 2\pi; \kappa \geq 0, 0 \leq \mu < 2\pi$$

θ は周期関数であり、 I_0 は第一種変形 Bessel 関数で式(2)のように表される。

$$I_\nu(\kappa) = \left(\frac{\kappa}{2}\right)^{2r+\nu} \sum_{r=0}^{\infty} \frac{1}{\Gamma(\nu+r+1)r!} \quad (\text{ap.2})$$

パラメータ κ は集中度、 μ は平均方向を表す。

以下に von Mises 分布の確率密度関数を示す。

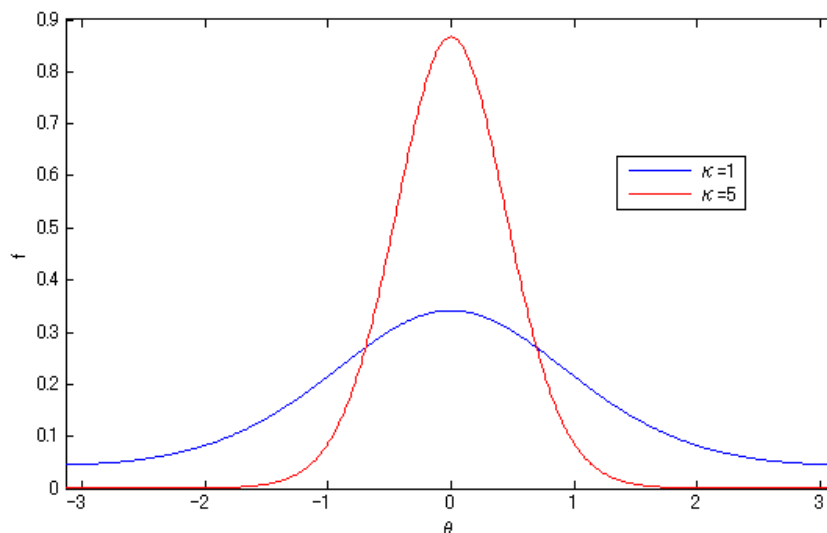


図 ap.1 von Mises 分布の確率密度関数($\mu=0, -\pi \leq \theta \leq \pi$)

謝辞

本研究を行うにあたり，懇切丁寧にご指導いただいた早稲田大学大学院基幹理工学研究科・甲藤二郎教授に心より感謝申し上げます。そして，共に研究に励み数々の貴重なアドバイスを下さった同期のみなさま，研究を通じて活発な議論にお付き合いいただいた小野君，甲藤研究室のみなさまに感謝いたします。

2011年 1月 31日

植村 あい子

参考文献

- [1] T. Fujishima, "Real-time chord recognition of musical sound: A system using common lisp music," Proc. ICMC, pp. 464-467, Oct.1999.
- [2] A. Sheh and D. P. Ellis, "Chord segmentation and recognition using EM-trained hidden markov models," Proc. ISMIR, pp. 183-189, Oct.2003.
- [3] J. P. Bello and J. Pickens, "A robust mid-level representation for harmonic content in music signal," Proc. ISMIR, pp. 304-311, Sep.2005.
- [4] K.Lee and M.Slaney,"Acoustic chord transcription and key extraction from audio using key-dependent HMMs trained on synthesized audio," IEEE Trans.on Audio,Speech and Language Procsing,16(2), pp.291-301, 2008.
- [5] M.P.RyynHänen and A.P. Klapri, "Automatic transcription of melody, bassline, and chords in polyphonic music," Computer Music Journal, 32(3), pp.72-86, 2008.
- [6] T.Inoshita and J. Katto, "Key Estimation using Circle of Fifths", 15th International Multimedia Modeling Conference, Jan.2009.
- [7] Judith C. Brown, "Calculation of a constant Q spectral transform," J. Acoust. Soc. Am., 89(1), pp.425-434, 1991.
- [8] Judith C. Brown and MillerS. Puckette, "An efficient algorithm for the calculation of a constant Q transform," J. Acoust. Soc. Am., 92(5), pp.2698-2701, 1992.
- [9] Curtis Roads, 青柳龍也, 後藤 真孝 他, "コンピュータ音楽—歴史・テクノロジー・アート," 東京電機大学出版局, Jan.2001.
- [10] Cabral, G., J.P. Briot, and F. Pachet, "Impact of distance in pitch class profile computation," Proceedings of the Brazilian Symposium on Computer Music 2005a.
- [11] C. Harte and M. Sandler, "Automatic chord identification using a quantised chromagram," in Proc. Audio Eng. Soc., Spain, May.2005.
- [12] 久保田慶一 他, "キーワード 150 音楽通論," アルテスパブリッシング, p.82-100, 2009.
- [13] E. Chew, "Towards a mathematical model of tonality", Thesis (Ph.D.) - Massachusetts Institute of Technology, Sloan School of Management, 2000.Includes bibliographical references, pp. 163-166.
- [14] E. Chew, "Measuring Musical Dissimilarity: First and Second Order Center of Effect (CE) Differences Inside the Spiral Array." Special Session on Mathematical Techniques in Musical Analysis Evanston, IL (Oct 24).
- [15] M. Goto, "A real-time music scene description system: Detecting melody and bass lines in audio signals," Working Notes of the IJCAI-99 Workshop on Computational Auditory Scene Analysis, pp. 31-40 ,1999.

- [16] A. P. Klapuri, "Multiple fundamental frequency estimation based on harmonicity and spectral smoothness," *IEEE Trans. Speech Audio Process*, 11, 804-816, 2003.
- [17] M. Davy and S. Godsill, "Bayesian Harmonic Models for Musical Signal Analysis," to appear in *Seventh Valencia International meeting (Bayesian Statistics 7)*, Oxford University Press, 2002.
- [18] H.Kameoka, T.Nishimoto, and S.Sagayama, "Extraction of multiple fundamental frequencies from polyphonic music using harmonic clustering," *Proc ICA 2004*, pp.I-59-62, 2004.
- [19] Katy Noland, Mark Sandler, "Key Estimation Using a Hidden Markov Model," *ISMIR proc.*, 2006.
- [20] 上田 雄, 小野 順貴, 嵯峨山 茂樹, "調波音／打楽器音分離手法とチューニング補正手法を用いた音楽音響信号からの自動和音認識," *情報処理学会研究報告*, Jul.2009.
- [21] M.Mauch et al., "Approximate Note Transcription for the Improved Identification of Difficult Chords," *Proc.ISMIR*, Aug.2010.
- [22] Oudre. et.al, "Template-Based Chord Recognition : Influence of the Chord Types ," *Proc. ISMIR*, pp. 153-158, Oct.2009.
- [23] T. Yoshioka. et.al, "Auto-matic Chord Transcription with Concurrent Recognition of Chord Symbols and Boundaries", *Proc. ISMIR*, Oct.2004.
- [24] バンドプロデューサー: <http://www.kawai.co.jp/cmusic/products/bp/>
- [25] 清水邦夫,"方向統計学の最近の発展," *計算機統計学*, 19(2), pp.127-150,2006.
- [26] isophonics : <http://isophonics.net/>
- [27] Supervised Chord Recognition for Music Audio in Matlab : <http://labrosa.ee.columbia.edu/projects/chords/>
- [28] MIREX : http://www.music-ir.org/mirex/wiki/MIREX_HOME

発表文献リスト

- [1] 植村あい子, 甲藤二郎, “Doubly Nested Circle of Fifths を用いた和音認識”, 情報処理学会音楽情報研究会, 2010 年 2 月.
- [2] 植村あい子, 甲藤二郎, “Doubly Nested Circle of Fifths を用いた和音認識の特性改善”, 情報科学技術フォーラム(FIT2010), 5G-3, 2010 年 9 月.
- [3] 植村あい子, 甲藤二郎, “Doubly Nested Circle of Fifths に基づく和音情報と HMM を用いた和音認識”, 電子情報通信学会総合大会, 2011 年 3 月. (投稿済み)