

転調を考慮したコード進行解析システム

清水 祐輔^{†1} 沼尾 雅之^{†1}

概要: 和声とは、和音の連鎖であり、音楽はこの和声に基づいて作編曲される。計算機が和声解析を行う事が出来れば、自動作曲システムや、表現技法やその楽曲、作曲家の特徴の解析に用いる事が出来る。今までの研究では、ベースラインから和音を絶対コードとして認識をするシステムが一般的であった。しかし、楽曲解析には、転調も考慮した主調上での和音認識が必要であり、そのためには転調部分の認識などが課題であった。本稿では、和音間の距離を定量化する事が出来る計算論的音楽理論 Tonal Pitch Space を用いた自動和声解析システムを提案する。デジタル化された楽曲 MIDI を対象にして、主調上のコード進行、転調位置および転調された調上でのコード進行を認識する。認識には、コード進行を状態遷移として隠れマルコフモデルをベースとして、拍節構造を考慮し、適切な和音区間を設定することでより人間の感性に近いシステムの実装を行う。

1. はじめに

近年、自動伴奏システムや、自動採譜システム、自動作曲システム、作曲支援システムなど、音楽を計算機上で扱う研究が幅広く行われている。しかしながら、作曲や編曲を計算機が行うという事は非常に難しいことであり、計算機がただ適当に音を割り当てただけでは音楽として成り立たない聞き苦しいメロディになる。これは、音楽が和声学と呼ばれるある種の規則を持っており、人間はその和声に従ったメロディを美しいと感じ、逆に反したメロディには違和感を覚える。

和声とは和音の連鎖であり、代表的な例として修礼の和声がある。このメロディを聞くととても聴きやすく感じられる。反面、禁則の和声と呼ばれるメロディは聴き辛く感じる。この様な和声の良し悪しを判断するにはカデンツと呼ばれる規則により定められている。カデンツとはクラシック音楽において楽章の終結を意味する進行であり、修礼の和声ではカデンツの規則の一つである $I \rightarrow V \rightarrow I$ の終止形に従っていることから聴きやすいメロディだと判断することができる。この様に、ある和音の次にどんな和音が来れば美しい進行となるかはこのカデンツにより決められているが、特定の和音の場合にはカデンツの進行に従っていても認められないといったように様々な制約が存在する。

ここで、一般的に楽曲は和声に従って作られているものの、それは絶対ではなく作曲者の意図により無視されている場合がある。和声学とは良い進行、悪い進行をまとめた

ものであるが、結局は過去の楽曲から得られた進行についての情報に過ぎず、経験則によって成り立っている。さらにこれは作曲のために作られたものであり、音楽の解析するためのものではない。

和声解析についての研究は、研究対象として wav ファイル、MP3 ファイルといった、音声データを扱った和声解析と、MusicXML ファイルや Standard MIDI ファイルのような記号化された楽譜データを扱った和声解析の二つに大別される。音声データを扱う場合は確率論的モデルを用いた解析手法がメジャーであり、楽曲中の各ピッチクラスの発音時間を頻度としてベクトル化し、学習データと比較を行っており、Hidden Markov Model(HMM) が採用されている場合が多い [6]。楽曲データを用いる場合はさらに二分化でき、一つは先述した確率論的モデルを用いた解析手法を用いることで、もう一つはルールベースを用いた手法である。本研究では楽譜データを扱い、ルールベースを用いた解析手法を提案する。

用いるルールとして、Lerdahl の提唱する計算論的音楽理論 Tonal Pitch Space(TPS)[1] を採用し、クラシック音楽の楽曲データを用いたコード進行解析システムを提案する。

先行研究との相違点として、MusicXML を入力とする。坂本らは実際の楽譜を、人手を使ってコードネームに直した記号的楽譜を入力としていたが、本研究では実際の楽譜データを用いることで和声解析システムを完全に自動化する。また、楽譜データを入力とするにあたって拍節構造を考慮し、適切な和音区間を設定することでより人間の感性に近いシステムを目指す。

^{†1} 現在、電気通信大学
Presently with University of Electro-Communication

2. Tonal Pitch Space

本章では, Tonal Pitch Space で用いられる理論について述べる.

2.1 TPS 概要

TPS の背景には, Lerdahl により提案された音楽理論 A Generative Theory of Tonal Music(GTTM)[2] がある. GTTM は楽曲を一種の言語として捉え, 楽曲を二分木化する. しかし, GTTM には和声を考慮した規則が記述されておらず, それを補うために TPS が発表された. TPS は調や和音の五度圏を用いて和声学を数学的構造から再構成する理論であり, GTTM では行えなかった和音, 調からの比較が可能である. 計算結果が整数値で返されるため, 単純に距離として扱う事が出来る.

しかしながら, 細部において曖昧な記述や矛盾した部分が存在するため, 実際に計算機に実装するのは困難である. 計算を行う和音同士が遠隔調の関係にある場合, 計算結果は和声学的な認識とは異なる. 坂本ら [3] はそれらの問題点を解決するために TPS を整備した. 本研究では原著に従わず坂本らが修正した式を用いる.

2.2 TPS の和音表記

TPS では独自の表記方法が定義されている. I/C といった表記であり, ローマ数字による音度とアルファベット表記の調の間にスラッシュを挿んだものである. これは C(ハ長調) の 1 番目の和音ということになる. 和音は I から VII までの 7 通りあり (図 1), 調は 24 種種類存在する. ここで, 各調名に対応する英語音名になっている. TPS における和音とは, その和音の主要三和音のことを指す. 例えば I/C であればド, ミ, ソである.

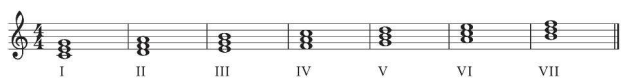


図 1 ハ長調の和音の構成

2.3 ベーシックスペースによる階層分け

TPS では和音の構成音の影響力を階層構造で表現する概念としてベーシックスペースという理論が存在する. ある音がその和音の中でどれほどの影響力があるか示しており, level 毎に階層分けされている.

- level a にあるピッチクラスはその和音の根音である.
- level b にあるピッチクラスはその和音の根音と五度音である.
- level c にあるピッチクラスはその和音の構成音である.
- level d にあるピッチクラスはその和音における調の構成音である.

図 2 は I/C のベーシックスペースである. 横軸がピッチクラス, 縦軸が影響度の level を示している. ここでピッチクラスとは, 音名を数値に変換したものであり, 0 から 11 の整数値を音名に割り振っている. 表 2.1 にそれぞれの音名とピッチクラスの対応を示す. 図から分かる通りピッチクラスの値が 0, すなわちドが一番影響力の高い音であることが分かり, その次に 7(ソ), 4(ミ) が続いている. またベーシックスペースを導入することの利点として, その和音の中でも実際に発音されていない音も暗黙的に響いている音であると考えられる. 例えば, レの音は I/C では発音されていないが, ベーシックスペースでは少なからず I/C の和音に関わっていることが分かる.

level a	0											
level b	0											
level c	0											
level d	0											
level e	0	1	2	3	4	5	6	7	8	9	10	11

図 2 I/C のベーシックスペース

表 1 音名とピッチクラスの対応値

音名	C	C# / D \flat	D	D# / E \flat	E	F
値	0	1	2	3	4	5
音名	F# / G \flat	G	G# / A \flat	A	A# / B \flat	B
値	6	7	8	9	10	11

2.4 和音間距離ルール

和音間距離関数 $\delta(x, y)$ を定義する. 以下の式 1 は, TPS で定義した式ではなく坂本らが修正した遠隔調にも対応した計算式である. $\delta(x, y)$ は, 和音 x と y が近親調にある時は上部の式を用い, 遠隔調にある時は下部の式を使う. 上部の式は, 後述する 3 つの関数の線形和であり, 下部の式は, x と近親調にあるトニックの和音 I/R_1 と, y と近親調にあるトニックの和音 I/R_2 を置くことにより, x から I/R_1 の和音巻距離, R_1 から R_2 の調間距離 $\Delta(r_1, r_2)$, I/R_2 から y の和音間距離の和である. この時, R_1 と R_2 は最小となるように選択する. 以上により完全版の和音間距離計算式を得ることができる. ここで $P(x)$ はその和音の近親調内にあるトニックの集合を求める関数である.

$$\delta(x, y) = \begin{cases} region(x, y) + chord(x, y) + basicspace(x, y) \\ \min\{\delta(x, I/R_1) + \Delta(R_1, R_2) + \delta(I/R_2, y) \\ |R_1 \in P(x), R_2 \in P(y)\} \end{cases} \quad (1)$$

2.4.1 region(x, y)

region(x, y) では, 調の五度圏による距離を求める. 調の五度圏とは, 12 の長調, 12 の短調からなる円であり, それぞれ隣の調が完全五度の関係になっている. また, 短調と長調は並行調の関係で接続されている. 調の五度圏を図 3 に

示す。この関数は調の五度圏において視覚的にどれだけ離れた孤に存在するかで考えることができる。任意の調からその隣の調までの距離を1とし、例えばハ長調からニ長調の距離は2となる。また、短調の場合は、その並行調のピッチクラスに変換する必要がある。平行調への変換は主音を三つ高い音へのシフトすることにより実現できる。以下の式2を用いて和音を長調のピッチクラスに変換する。x.keyは和音xの調のピッチクラス値を表している。

$$x.key = \begin{cases} x.key & (x.key = Major) \\ (x.key + 3) \bmod 12 & (x.key = Minor) \end{cases} \quad (2)$$



図3 調の五度圏

2.4.2 chord(x, y)

chord(x, y)では、和音の五度圏における距離を求める(図4)。region(x, y)と同様に、和音の五度圏において二つの音度がどれほど離れているかを定量化する。

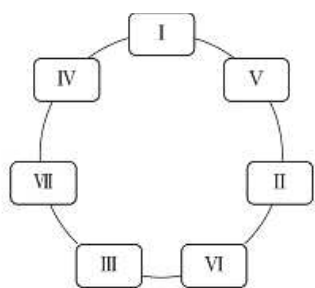


図4 和音の五度圏

また、I/CとI/C#のような音度が変わらない、変換が不可能な和音では十分に大きな数となるようにする。

2.4.3 basicspace(x, y)

basicspace(x, y)では、xのベーシックスペースとyのベーシックスペースを比較した時に異なっているピッチクラスの数を求める。式3のようにピッチクラス毎に差を合計することにより求めることができる。iはピッチクラス

値、+1はピッチクラス値の相和が奇数の場合に小数点を切り上げていることを示す。これは、三和音とセブンス・コードのように構成音の数が違う和音間で比較を行った時のための処理である。本研究のシステムとは直接関係ないが、原著と一致させるために行っている。例としてI/CとV/Cの比較を行う(図5, 図6)。式2.6より計算を行うと値は4となる。

$$basicspace(x, y) = \frac{\sum_{i=0}^{11} |x.basicspace[i] - y.basicspace[i]| + 1}{2} \quad (3)$$

level a	0											
level b	0									7		
level c	0			4						7		
level d	0		2	4	5					7	9	
level e	0	1	2	3	4	5	6	7	8	9	10	11

図5 I/Cのベーシックスペース

level a												7	
level b												7	
level c												7	
level d	0			2	4	5					7	9	11
level e	0	1	2	3	4	5	6	7	8	9	10	11	

図6 V/Cのベーシックスペース

2.4.4 遠隔調への対応

上記の和音間距離では、和音間距離を定量化することに成功した。しかしながら、計算を行う和音同士が遠隔調の関係にある場合、計算結果は和声学的な認識とは近似しない。例えば、 $\delta(I/C, III/c\sharp)$ という進行は和声学ではありえない禁則の和声である。しかしながら、TPSの和音間距離計算式ではそれ程離れておらず進行としては現実的な計算結果が出力される。理由としてはregion(x, y)の値が最大6、chord(x, y)の値が最大3、basicspace(x, y)は最大値22が出力されているため、TPSの和音間距離計算式はbasicspace(x, y)の依存度が大きく、和声学において重要な要素である調を求めるregion(x, y)の依存度があまり高くないことが挙げられる。そのため求める2つの和音の調が離れているほど違った計算結果が出力される。そこで坂本らは、和音間の距離に加え、調間についても距離の定量化を行い、和音間距離式を修正した。

まず最初に調性空間を構築する。調性空間とはどの調とどの調が近い関係にあるのかを示すもので、各調のIの和音と和音間距離ルールにより求められる。次のような手順により求められる。

- (1) ある調を選択し、調性空間におく。
- (2) 選択された調のトニックから他のトニックへの和音間距離を計算した時、属調・下屬調・平行調・同主調の関係にある和音同士は計算結果が最小となる。これらに関係に応じて上下左右に配置する(属調は上、下屬上は下、長調であれば同主調は右、平行調は左、短調であれ

ば平行調は右, 同主調は左; 図 7)

(3) 2 で新しく置かれた調を選択し, それらに対して 2 を行う. 同様にすべての調が選択されるまで繰り返す.

以上の手順により得られた調性空間が図 8 である. 全ての長調について, 上が属調, 下が下屬調, 左が平行調, 左上が属調並行調, 左下が下屬調並行調, 右が同主調というように関係調が隣接して並んでいる. 短調も長調と左右逆ではあるが近親調が隣接している.

調性空間を用いて調間距離 $\Delta(r_1, r_2)$ を求める. r_1, r_2 は調を表す変数である. それぞれ以下のルールに従い求める.

- ある調から移動できる範囲はその関係調のみである.
- それらへの距離はそのトニック同士の距離とする (表 2).
- 目標の調への経路の最小値を計算結果の値とする.

このルールにより, 遠隔調の関係について調間距離を用いて再定義でき, $\Delta(r_1, r_2) \leq 10$ の時 r_1 と r_2 は近親調となり, その他は遠隔調となる.

表 2 調の関係と距離

関係	平行調	属調・下屬調	同主調	属調平行調	下屬調平行調
距離	7	7	7	9	10

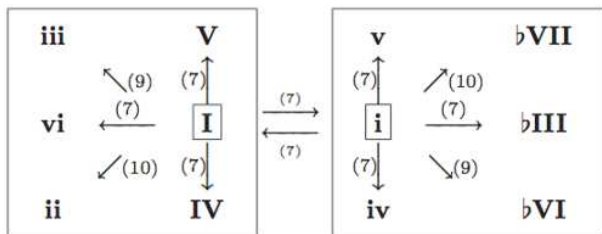


図 7 近親調の距離と配置

d#	F#	f#	A	a	C	c
g#	B	b	D	d	F	f
c#	E	e	G	g	Bb	bb
f#	A	a	C	c	Eb	eb
b	D	d	F	f	Ab	ab
e	G	g	Bb	bb	Db	db
a	C	c	Eb	eb	Gb	gb

図 8 調性空間

3. コード進行解析システムの実装

本章では, 提案システムの概要を述べ, 計算機への実装方法について示す.

3.1 コード進行解析システム概要

第 3 章で述べた和音間距離計算式を用いて和音解析を行うシステムを提案する. 図 3.1 に本システムの概略図を示す. 構成音に対して, 適合する和音を列挙し, それらの調の中でどれが最もふさわしいかを TPS による和音間距離を基準として選択する. よって, 幾つかの和声の中からどれが一番妥当な進行なのかを推定している.

提案システムは和音候補生成部, 和音間距離計算部に分けられ, 和音間距離総和が最も小さくなる進行を最適解として出力する. 本研究の特徴として, 坂本ら [3] はコードシーケンスを入力としていたが, 本システムでは実際の楽曲データである MusicXML データを入力としている. また, 本システムは拍節構造を考慮している. 和音区間を拍節単位で解説するため, より人間の感性に近い解析が可能となる. 以下に本システムの各部位について記述する.

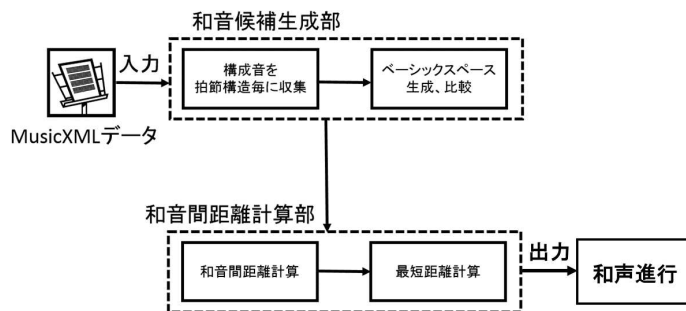


図 9 提案システムの概要

3.2 和音候補生成部

和音候補生成部では, 入力された楽曲データから構成音を収集し, ベーシックスペースと比較, 和音候補を列挙する. また, 構成音を収集する際に拍節構造を考慮する. 拍節構造とは, 楽曲を拍子, 小節に分割して考えることであり, 人間は実際拍子, 小節毎に和音を振っている. 本システムでは拍節構造をコンピュータに認識させるため, 一度ホモリズム的な変形を施し, 拍子で分割する (図 10). また, 集めた構成音をベーシックスペースを用いて比較する.

この時, level c 以上の構成音が 2 つ以上含まれるベーシックスペースの和音を候補とする (図 11). また, 構成音が 1 つのみの場合, その構成音が根音となりうる和音を候補とする. ここで, 調号や臨時記号がついていない場合は転調が起きていないものとして現行調を最優先する.

候補とされた和音をノードとして扱い、TPS の和音間計算式を遷移コストとした状態遷移グラフを構築する (図 12).

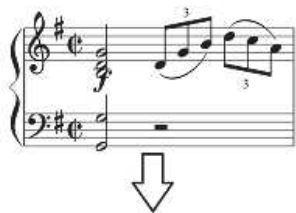


図 10 拍節構造のイメージ

level a	0											
level b	0						7					
level c	0			4				7				
level d	0	2		4	5		7		9		11	
level e	0	1	2	3	4	5	6	7	8	9	10	11

図 11 I/C のベーシックスペース

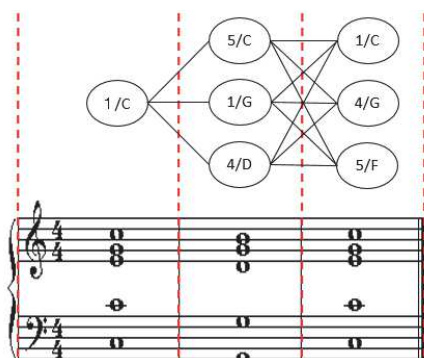


図 12 拍節構造のイメージ

3.3 和音間距離計算部

和音間距離計算部では、上記により得られた和音候補から和音間距離を計算を行う。計算式は第 2 章で述べた式 1 を用いる。ここで、viterbi アルゴリズムを採用し最短経路を求めることで、近傍解ではなく楽曲全体としての最適な進行を出力する。また、和音候補に最短経路が複数出現することがあるが、 $region(x, y) > chord(x, y) > basicspace(x, y)$ の優先順位を設け、調が一番優先されるように調整を行う。その次に和音が優先されるが、 $basicspace(x, y)$ が和音 x と y がどれだけ似ているかを定量的に表す関数であることに、 $chord(x, y)$ は和音の五度圏の距離から、進行のスムーズさを表す関数である。和声学は和音の進行をまと

めたものであり、良い和声というのは似ている和音同士の進行ではなくより聴きやすい和音の進行である。その点から、和音の五度圏による距離が近い和音同士がより聴きやすい和音の遷移であるため、 $chord(x, y)$ を優先させる。

4. コード進行システム出力結果・考察

本章では、第 3 章で述べたシステムに対し、実際に楽曲データを入力した結果を示し、その考察を行う。

4.1 実際の楽譜からの解析結果

入力として Beethoven の Piano Sonate, Op49, No2 を用いて解析を行った。システムの出力結果を図 13 に示す。本研究では拍子単位を 1/4 小節, 1/2 小節, 1 小節の 3 パターンで解析を行い、time は拍子パターン毎の拍子を表し、chord は TPS による和音表記である。

```
time = 0 chord = 1/G
time = 1 chord = 1/G
time = 2 chord = 1/G
time = 3 chord = 5/G
time = 4 chord = 1/G
time = 5 chord = 1/G
time = 6 chord = 5/G
time = 7 chord = 1/G
time = 8 chord = 5/G
```

図 13 出力結果 (1/4 小節)

また、和声進行の正解データとの比較を行い、精度を求める。精度は拍節単位での一致数を総拍節数で割った一致率で求める。

この式によって得た値を精度とする。また、小節毎に 3 パターンの中から精度の良かった拍子単位で計算する best-effort 値も求めた。計算結果を表 3 に示す。今回は精度を求めるにあたり、和音名での比較ではなくコードネームでの比較を行った。これに関しては得られた正解データはコードネームのみで記されており、転調の判断を和声に習熟していない著者が勝手にしてしまうのは危険だと判断したからである。

表 3 音名とピッチクラスの対応値

拍子パターン	精度
1/4 小節	77.0%
1/2 小節	70.4%
1 小節	50.8%
best-effort	94.5%

5. まとめ

まず、本システムで得られた成果について述べ、考察と今後の課題について示す。

5.1 研究成果

本研究では、TPS を用いたコード進行解析システムを計算機上に実装した。坂本らは TPS の和音巻距離計算式を原著にできるだけ沿った形で、実装が用意になるようルールを修正を行ったが、実際の楽曲を入力していない問題点があった。そのため、本研究では実際の楽曲データを入力したシステムを実装した。楽曲データを取り扱う事で和音区間の認識が必要であったが、拍節構造を考慮することにより一定ではあるが区間の認識をすることが出来た。

また、コードネームからではなく区間内の構成音から和音候補を列挙しなければいけないため、ベーシックスペースを導入した和音候補列挙法を実装した。これにより、計算に必要な和音を取り除き、計算速度、精度が向上した。

これらを踏まえた評価実験の結果、本システムは best-effort で 94.5%の精度を出力する事が出来た。このことから、TPS は和声学に近似しており、有用性のある理論であると示すことが出来たと考える。

5.2 考察・今後の課題

以上の成果が得られたが、本システムにおける改良点として、非和声音を考慮出来ていない事が考えられる。和音候補列挙において、非和声音は非常に邪魔な存在であり、コード認識の妨げになってしまっている。非和声音を取り除く処理が必要だが、非和声音とは楽曲を色付けするための音であり、様々なパターンが存在する。そのため非和声音を取り除くためのルールを実装することは困難であるため、違った手法を用いて非和声音を処理することが考えられる。

図 14 は、楽曲の中で問題となった部分、表 4 は 1/4 小節単位で解析した結果だが、赤線で囲まれた部分は正解データによると非和声音という扱いだ、本システムはイ短調のトニックという結果になっている。構成音が 1 音のみという拍節に対して、本システムではトニック以外認めていないためこのような結果になった。前後での転調を考慮したり、非和声音だという認識をさせる必要性がある。

TPS を用いた楽曲解析の場合、和音候補生成部分に対して、TPS の理論は用いる事が出来ないため、非和声音などはアドホックな処理を行わなければならない。このことから、TPS は構成音からのコード認識に向かない理論である。川上ら [9] は確率モデルを用いたコード認識システムを提案し一定調の旋律であれば精度が 100%という高い精度を得ている。しかし、確率モデルの問題点として、音楽理論を極力用いない数理モデルという事があり、転調の検出が上手く出来ていない。転調においては、和声学の知識は必須であり、確率モデルは転調の検出に向かない事が分かる。

今後の課題として、コード分析部分において、確率モデルでコード推定をし、TPS などの計算論的音楽理論を用いた新しいコード進行解析システムの実装を目指す。これにより、考察で得られたコード認識の問題を解決できると考え

られる。

また、TPS の和音間距離計算式に関しても改善の余地があり、 $\text{basic-space}(x, y)$ の値が大きくなってしまい、調間距離がさほど重要視されていない事が挙げられる。 $\text{basic-space}(x, y)$ とは、和音 x と y がどれだけ似た和音であるかという事を定量化したものであり、調やコードよりも優先度は低いと考えられる。転調におけるコストを重くするため、TPS の和音間距離計算式にパラメータの付与が考えられる。

表 4 問題の出力結果

time(1/4 小節)	78	79	80	81	82	83
コード	1/a	1/a	1/D	5/D	1/D	5/D



図 14 出力結果 (1/4 小節)

参考文献

- [1] Fred Lerdahl. *Tonal Pitch Space*. Oxford University Press, 2001
- [2] Fred Lerdahl and Ray Jackendoff. *A Generative Theory of Tonal Music*. MIT Press, 1983.
- [3] 坂本 鐘期. Tonal Pitch Space を用いた楽曲の和声解析. 修士論文, 北陸先端科学技術大学院大学, 2010
- [4] 調 - Wikipeda(日本語版). <http://ja.wikipedia.org/wiki/調>.
- [5] 小方 厚. 音律と音階の科学-ドレミ... はどのようにして生まれたか. ブルーバックス, 2007
- [6] Thomas Rocher, Matthias Robine, Pierre Hanna, Robert Strandh. *Dynamic chord analysis for symbolic music*. In Proceedings of the International Computer Music Conference(ICMC 2009), 2009
- [7] Satoshi Tojo, Keiji Hirata, Masatoshi Hamanaka. *Computational Reconstruction of Cognitive Music Theory* Japan Advanced Institute of Science and Technology, Future University Hakodate, University of Tsukuba, 2012
- [8] 島岡 譲. 和声と楽式のアナリーゼ. 音楽之友社, 1964
- [9] 川上 隆, 中井 満, 下平 博, 嵯峨山 茂樹. 隠れマルコフモデルを用いた旋律への自動和声付け. 北陸先端科学技術大学院大学
- [10] 横井 史也, 平田 圭二, 竹川 佳成. “ルール処理と HMM を統合した和音認識器の実現”. 公立はこだて未来大学, JSAI, 2015