

# 音楽の重要な構成要素の抽出の提案 -音楽ゲーム用譜面自動生成のために-

香川 俊宗<sup>a)</sup> 手塚 宏史<sup>b)</sup> 稲葉 真理<sup>c)</sup>

概要：音楽ゲームは計算機とプレイヤーが合奏し曲を完成させるもので、プレイヤーが演奏するパートを原曲から分離・抽出し、プレイヤーのアクションを指示する譜面を作成することが難しいことが知られている。本研究ではよい譜面の自動生成のための重要音抽出を目指す。音楽の中で繰り返し現れるフレーズが重要な部分であると考え、suffix tree を利用してフレーズを抽出する頻出頻度を利用した重要音抽出法を提案・実装し、メロディ、ハーモニー、リズムそれぞれに適用し、譜面に使用する楽曲を構成し、ユーザーテストを行った。

## Frequency-Based Key Component Extraction -Automatic Generation of Instruction Scores for Music Video Games-

KAGAWA TOSHIMUNE<sup>a)</sup> TEZUKA HIROSHI<sup>b)</sup> INABA MARI<sup>c)</sup>

### Abstract:

The music video game is a game that a player plays small synthesizing device following a given instruction score to make a good ensemble with the background music which a computer plays. It is well known that generating good instruction scores is difficult. The goal of our research is automatic generation of good instruction scores. In this paper we propose "frequency-based key component extraction" which automatically extracts important key components from the music. Here, we assume that the phrases which occur repeatedly are important, and using suffix tree, we count the frequencies of phrases for melody, harmony and rhythm. Using this statistics, we extract the player's part, and performed user's testing.

### 1. はじめに

音楽ゲームとは、リズムや音楽に合わせてプレイヤーがモニターを見て鍵盤やボタンを叩いたり、パネルを踏んだりすることで進行するゲームのことである。一般に、音楽ゲームには音楽に合わせてプレイヤーへの指示が記された譜面が存在する。譜面で指示されたタイミング通りにアクションをすることで高得点を得られる仕組みである。また、ノートとは、いつ叩くか、どこを叩くか、叩くとどの音が鳴るか、の情報が含まれているプレイヤーへの指示であり、譜面は複数のノートから構成されている。音楽ゲームの最

大の特徴はモグラ叩きなどの反射神経のみを使用するゲームと違い、音楽に合わせ、予測をし、音楽を完成させることで面白さが生まれている点である。そのため、音楽ゲームのよい譜面を作成するのは難しいことが知られている。また、譜面の良し悪しの評価はプレイヤーコミュニティ内である程度一致するが、明確な判断基準は存在しない。

音楽ゲームには、トランスやハウスといったクラブミュージックから、ポップス、クラシックなど、様々なジャンルの楽曲が収録されている。収録されている楽曲のほとんどがメロディが重視されたものになっており、プレイヤーが音に合わせてやすいように BPM の速い楽曲が多い。そのためクラシックなどは編曲されることが多く、電子音やギターなどが追加され、様々なアレンジが加えられている。また、楽曲は 90 秒から 120 秒程度で終わることが多く、終盤に転調することも多い。従って、様々な楽器に渡る主旋律を

<sup>1</sup> 東京大学大学院情報理工学系研究科  
UT, Bunkyo, Tokyo 113-8654, Japan

a) kagawa.toshimune@ci.i.u-tokyo.ac.jp

b) tezuka.hiroshi@ci.i.u-tokyo.ac.jp

c) mary@ci.i.u-tokyo.ac.jp

抽出し、転調も考慮する必要がある。一般的に、同じ楽曲に複数の譜面が用意されており、易しい譜面はノート数が少なく、難しい譜面はノート数が多い。そのため、同じ楽曲でも複数の抽出を行う必要がある。また、易しい譜面は難しい譜面より重要度の高いフレーズを抽出する必要がある。

本研究ではよい譜面の自動生成のための重要音抽出を目指す。ターゲットとする音楽ゲームは概ね以下のようなものである。原曲からいくつかの音を抽出・消去し、消去した音が演奏されるタイミングで機器（鍵盤等）を操作する指示がモニターに表示され、指示通りのタイミングで機器が操作されると消去された音が復元され、原曲通りになる。譜面の作成は、原曲の音の集合  $S$  から消去する音である部分集合  $T$  を選択し、 $T$  の各要素を機器に割り付ける問題と考えられる。音楽にはジャンルを問わずメロディ、ハーモニー、リズムなど、西洋音楽の基本要素が内在し、作曲や編曲によく利用されている [1][15]。そこで、原曲をこのような要素にそれぞれ分解し、解析・抽出・再構成をすることで音楽ゲームのよい譜面の自動生成に役立つのではないかと考えた。また、このような要素は楽曲の中で繰り返し出現し、つまり、重要なフレーズや特徴的なフレーズは楽曲中に繰り返し出現すると考える。そのため、音楽を離散的に構造解析するために suffix tree を利用した。

ここで、本論文で用いる用語の定義をする。音楽ゲームとは以下のようなゲームである。不完全な曲とともにプレイヤーが取るべきアクションが提示され、プレイヤーがアクションに成功すると音が鳴り楽曲が完成し、高得点が得られるゲームである。「(1) どのタイミングで (2) どこを叩くか」という指示と、その結果 (3) どの音が鳴るか」という3つ組の情報をノートと呼び、ノートはプレイ中にプレイヤーに与えられ続ける。この、楽曲1曲分のノート列のことを譜面と呼ぶ。ノートに含まれている情報のうち、(3) どの音が鳴るか、の情報をキー音と呼び、キー音になりうる全ての音の候補をキー音候補と呼ぶ。原曲から全てのキー音が消去された音楽を BGM と呼ぶ。任意の連続した音符の集合をフレーズと呼ぶ。

本稿では、MIDI の形式で与えられた音楽から重要なフレーズの抽出をすることでキー音候補を自動生成することを目標とする。以下、2章では関連研究について紹介する。3章では音楽を離散的に構造解析し、重要なフレーズを抽出する頻出頻度を利用した重要音抽出法について述べる。4章ではシステムの実装と評価実験について報告し、最後に5章で本研究の到達点と今後の課題について述べる。

## 2. 関連研究

### 2.1 譜面自動生成

音楽を入力とし、譜面を自動生成し出力する音楽ゲームは国内外問わず開発されており、beat gather[10]、Any

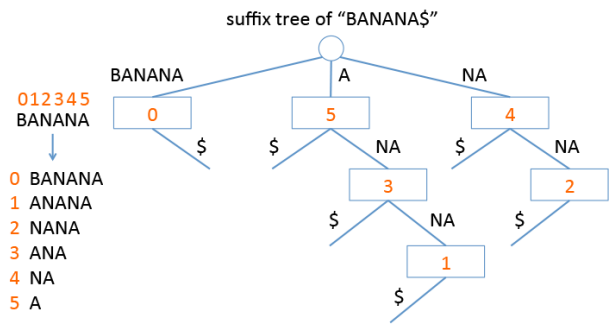


図 1 文字列”BANANA”の suffix tree[9]  
Fig. 1 Suffix tree of ”BANANA” [9].

Beats[11]、昇天ビート [12]、The Real DJ[13] など、プラットフォーム問わず多岐に渡っている。しかし、いずれも譜面自動生成の精度はよいとは言えず、あまり音楽に合った譜面は生成されない。また、beat gather や Any Beats などは、他のプレイヤーが譜面を手打ちで生成することが可能であり、それを共有して遊ぶ方がメインになっている。

### 2.2 suffix tree

suffix tree は Weiner らが提唱した構造体であり、与えられた集合  $S$  の全ての接尾辞を表した木構造のデータ構造で、通常文字列検索などに利用される [5]。全部分文字列の出現回数が検出でき、また、あるパターンに対して完全に一致したパターンがあればその場所を検索できる [8] (図 1)。

### 2.3 音楽解析

Jekovec らは、音楽に suffix tree を適用することにより、メロディ解析を行った [2]。Jekovec らは、前後の音符の音高の相対値と音の長さを利用して音楽を解析しているが、我々は音高の絶対値・相対値、楽器、音の長さをそれぞれ抽出したいフレーズに合わせて利用して音楽を解析した。また、その抽出したフレーズに対しさらに抽出を行い、譜面自動生成のための重要音抽出を行った。高田らは、MIDI の繰り返し構造を検出することにより、サビ区間の特定を行った [4]。Ioannis らは、楽曲中の aggressive accession を利用し、全ての楽曲中の最大長の繰り返しパターンの検出を行った [3]。

## 3. 重要音抽出法の提案

### 3.1 譜面自動生成における課題

音楽ゲームの譜面を手作業で作る上で困難とされるのは、どの音を抽出するか、ということと、抽出した音を譜面のどこに割り当てるか、という2点である。特に音の抽出は困難であり、プレイヤーが曲に合わせて叩けるように抽出する必要がある。しかし、だからといってメロディばかりを抽出すればよいわけではなく、例えば、ある部分では

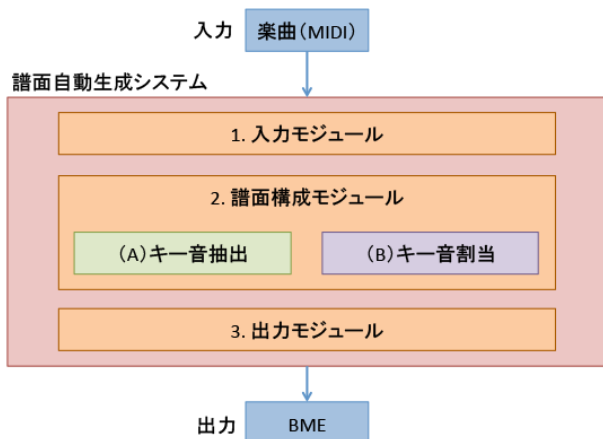


図 2 譜面自動生成システム

Fig. 2 Automatic generation system of instruction scores.

リズムを抽出し、ある部分ではハーモニーを抽出し、また、ある部分ではメロディとリズムを抽出し、と、プレイヤーを飽きさせない工夫も必要である。また、メロディと一口に言っても音楽ゲームに収録されている楽曲は多楽器が基本であり、適度に様々な楽器の音を抽出する必要がある。

### 3.2 譜面生成フレームワーク作成と予備実験

研究対象に属する音楽ゲームの中でも、beatmaniaIIDX<sup>\*1</sup>を模した BME フォーマットで記載された Be Music Script (以下、BMS) [16] を使用し、音楽ゲーム用譜面生成フレームワークを設計・実装した(図 2)。このフレームワークは、MIDI ファイルを入力すると、組み込んだ(A)キー音抽出モジュールと(B)キー音割当モジュールに従って、自動で譜面(BME ファイル)を出力するものである。ここで、(A)キー音抽出モジュールと(B)キー音割当モジュールは組み替え可能である。このフレームワークを利用し、まずは下記の2つの方法で譜面を自動生成し、予備実験を行った。

- (1) (A) 全ての音からランダムに音を抽出し、(B) 全てランダムに譜面に割り当てたもの。
- (2) (A) パーカッショントラック以外の音を全て抽出し、(B) 全てランダムに譜面に割り当てたもの。

BMS プレイヤーには Lunatic Rave2 (beat2 100201) [14] を使用した。このようにして自動生成された譜面に対して、評価者に実際にプレイしてもらった。予備実験には、情報理工学系の修士学生7名・博士学生2名の計9名(うち、音楽ゲーム経験者4名)に参加してもらった。ここで、音楽ゲーム経験者としての定義は特に設けず、本人の自己申告とした。実験(1)に関しては、9名が楽しくなかったと解答した。実験(2)に関しては、3名が楽しかった、2名が普通、4名が楽しくなかったと回答した。また、実験(2)ではパーカッショントラック以外の全ての音を抽出してい

\*1 <http://www.konami.jp/bemani/bm2dx/>

るため、必然的に難易度が上がってしまい、そもそもプレイすることすらできない評価者もいた。この結果より、ランダムに抽出するだけではよい譜面は得られないということを検証した。また、パーカッション以外の音のような、音楽の要素に基づいた音を抽出したほうがよいということを検証した。キー音抽出とキー音割当の中でもまずはキー音抽出から始める必要があると考え、本論文ではキー音抽出についての研究を述べる。

### 3.3 キー音のための重要音抽出法の提案

音楽ゲームの音楽はキー音とBGMで構成されており、BGMに合わせてキー音を鳴らすことで原曲が完成する。また、一般にキー音には場所によってメロディ、ハーモニー、リズムそれぞれを叩かせるパートが存在し、プレイヤーはそのパートの流れを認識・予測可能であるため、音楽ゲームを楽しんでいる面がある。

本論文では、頻出頻度を利用した重要音抽出法(frequency-based key component extraction, 以下、FBKCE法)の提案を行う。FBKCE法とは、音符が持つ絶対時間、楽器(チャンネル番号)、音高(ノートナンバー)、音の長さの4つの情報の中の一部を利用し、任意の部分音符列の音数と出現回数を数え二次元グラフ化し、特定のしきい値を与え、しきい値以上の音数及び出現回数を重要フレーズとし、抽出する。なお、しきい値は曲によって変化する。音楽において、重要なフレーズは繰り返し現れると考えた。そこで、音楽を離散的に構造解析し、全部分音符列の出現回数を調べる。その際にメロディ、ハーモニー、リズムそれぞれの特性に合わせて suffix tree を適用し、それぞれの重要なフレーズを抽出し、キー音候補を自動生成する。

## 4. FBKCE法の実装と評価

### 4.1 システムの実装

本システムの目標は、MIDIを入力とし、全ての部分音符列の出現回数と、任意のフレーズのMIDIを出力するものである。入力において、MIDIのフォーマットのままでは扱づらいため、相対時間を全て絶対時間に変換し、ノートオンとノートオフのイベントをまとめ、音符形式に変換した。MIDIには複数のトラックが内在するが、そのうち任意のトラックのみを選択できるようにした。また、任意のトラックを別のトラックに繋げたり、1つのトラックにまとめることも可能にした。文字列を扱う suffix tree ではノードに文字を格納するだけであるが、FBKCE法では音符を扱うため、複数の情報を格納できるように設計した。出力において、任意の部分音符列のMIDIの他、部分音符列の音数と出現回数の範囲を指定して、その範囲内の音をMIDIとして出力できるようにした。プログラムはC++で作成し、2000行程度である。

表 1 音符情報比較表 1

Table 1 Comparison table of note information1.

	トラック	楽器	音高	音の長さ
1	全トラックまとめ		絶対音	
2	全トラックまとめ		絶対音	—
3	全トラックまとめ		相対音	
4	全トラックまとめ		相対音	—
5	全トラックまとめ		—	
6	全トラックまとめ		—	—
7	全トラックまとめ	—	絶対音	
8	全トラックまとめ	—	絶対音	—
9	全トラックまとめ	—	相対音	
10	全トラックまとめ	—	相対音	—
11	全トラックまとめ	—	—	
12	全トラック別		絶対音	
13	全トラック別		絶対音	—
14	全トラック別		相対音	
15	全トラック別		相対音	—
16	全トラック別		—	
17	全トラック別		—	—
18	全トラック別	—	絶対音	
19	全トラック別	—	絶対音	—
20	全トラック別	—	相対音	
21	全トラック別	—	相対音	—
22	全トラック別	—	—	

## 4.2 評価実験

### 4.2.1 予備実験

まず、予備実験として、単純な楽曲について suffix tree を作成する。実験は、シャルル＝ルイ・アノンの60の練習曲によるヴィルトゥオーゾ・ピアニスト第一部（以下、ハノン）\*2 と、ヨハン・ゼバスティアン・バッハのフーガ BWV578（以下、小フーガ）\*3 の2曲について行った。音の含まれるトラック数はそれぞれ、ハノンは1トラック、小フーガは4トラックである。

それぞれの2曲について表 1 の22パターンの suffix tree を作成した。表 1 は suffix tree を作成する種類であり、比較するものを変えてそれぞれに番号を割り振っている。全トラックまとめとは、音符を絶対時間でソートし、1つのトラックに再構成している。また、全トラック別とは、全トラックを横に並べて繋げ、音符を絶対時間でソートせず、1つのトラックに再構成している。音高の絶対音とは、音符のノートナンバーをそのまま比較している。また、相対音とは、前後の音符のノートナンバーを比較し、その差を比較している。横線が引かれている箇所は比較を行っていない。

横軸が部分音符列の音数を示しており、縦軸が部分音符列の楽曲中の出現回数を示している。また、音楽の特性として部分音符列の音数が少ないものほど出現回数が増え、

\*2 <http://tukinoyakata.otogirisou.com/top.html>

\*3 <http://www1.ocn.ne.jp/~bach/index0.html>

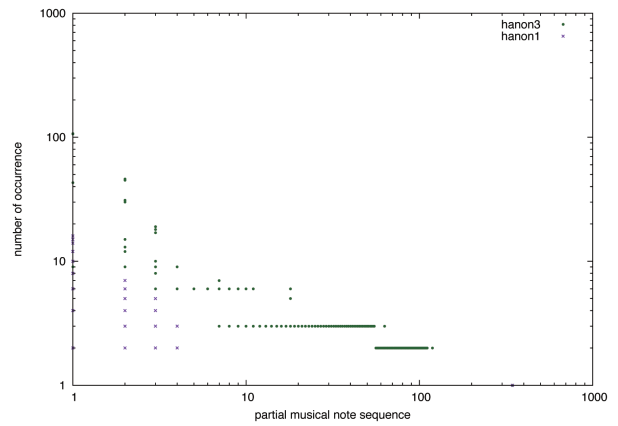


図 3 部分音符列出現頻度-ハノン 1, 3-  
Fig. 3 Frequency of partial musical note sequence -hanon1, 3-.

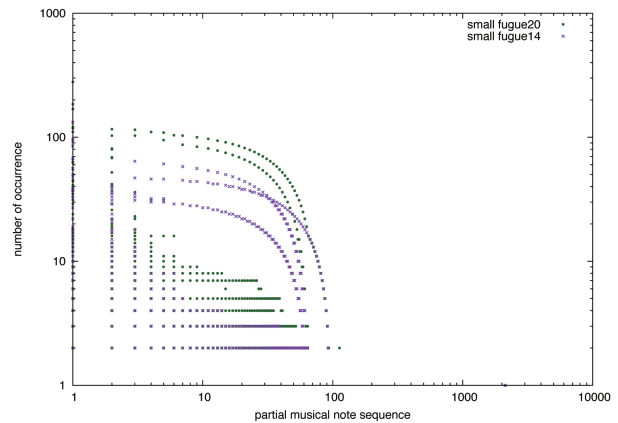


図 4 部分音符列出現頻度-小フーガ 14, 20-  
Fig. 4 Frequency of partial musical note sequence -small fugue14, 20-.

音数が多いものほど出現回数が減るため、どの部分音符列の出現回数が多いかを確認しやすくするために対数軸にしている。グラフにする際、出現回数が同じ部分音符列に対して、ある部分音符列の一部と完全に一致する部分音符列に関しては、ある部分音符列から1つずつ音符を減らしていった部分音符列であるため、プロットしないようにしている。また、グラフの特性として同一の座標に2つ以上の点がプロットされる場合があるので、ヒートマップのように点の色を変えることにより表現している。ただし、2つのグラフを同時にプロットする場合、視認性の問題のためヒートマップにはしていない。

ハノンについて、図 3 で1番と3番の実験結果を比べると、絶対音に比べ相対音で比較すると音数の多い部分音符列を抽出可能となる。この結果より、絶対音よりも相対音の方がメロディやハーモニーを抽出する際に、転調なども抽出できるのではないかと考えた。転調は音楽ゲームの楽曲でよく使われる手法であり、また、音楽を構成する上で重要な要素であるため、抽出できる必要がある [6]。

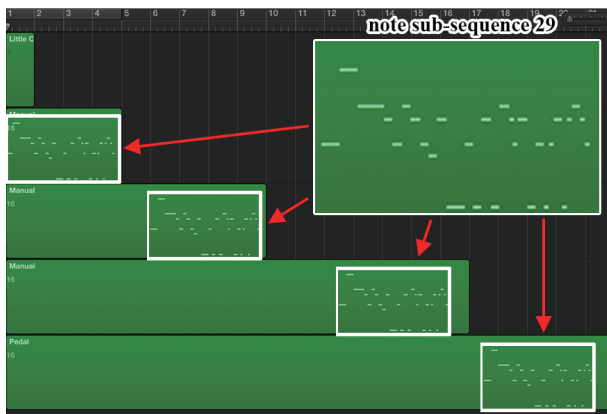


図 5 GarageBand 譜-小フーガ 20-  
Fig. 5 GarageBand-small fuge20-

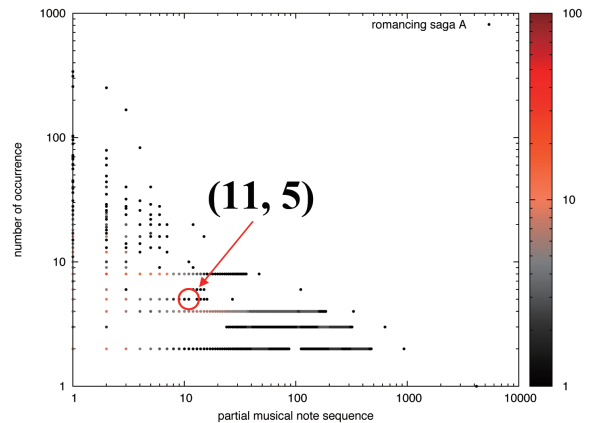


図 8 部分音符列出現頻度-ロマンシングサ・ガ A-  
Fig. 8 Frequency of partial musical note sequence -Romancing SaGa A-

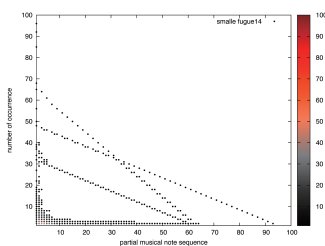


図 6 部分音符列出現頻度-20-  
Fig. 6 Frequency of partial musical note sequence -small fuge20-

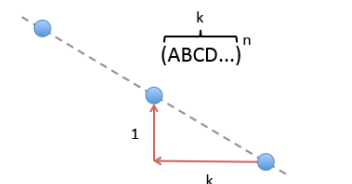


図 7 部分的に繰り返される音符列  
Fig. 7 Partially repeated note sequence

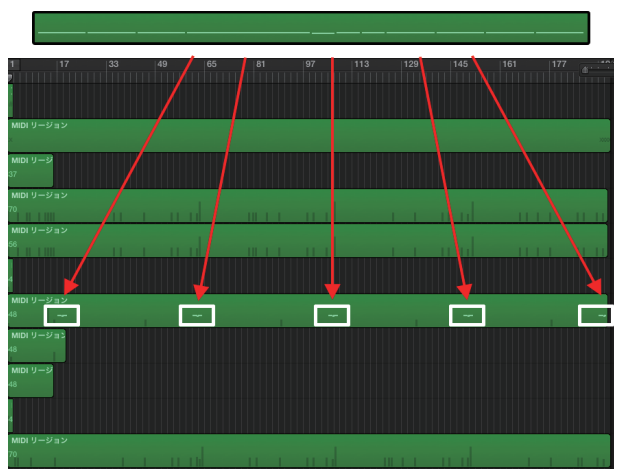


図 9 GarageBand 譜-ロマンシングサ・ガ A-  
Fig. 9 GarageBand-Romancing SaGa A-

表 2 音楽情報比較表 2

Table 2 Comparison table of note information.

	トラック	楽器	音高	音の長さ
A	パーカッショントラック以外	—	相対音	
B	パーカッショントラック以外まとめ		相対音	
C	パーカッショントラックまとめ		絶対音	

小フーガについて、図 4 で 14 番と 20 番の実験結果を比べると、楽器を比較すると 1 楽器ごとのメロディは抽出可能だが、主旋律が複数楽器に渡るようなメロディは抽出できない。例えば、20 番でプロットされており 14 番でプロットされていない音数 29、出現回数 4 の部分音符列を表している (29,4) について音を調べると、図 5 のように複数楽器に渡るような主旋律が抽出できている。

また、小フーガの 20 番の実験結果について、x の範囲を 1 から 100 まで、y の範囲を 1 から 100 まで対数軸ではなく線形軸で表すと図 6 のような直線がいくつか表れる。これは、図 7 のように、(ABCD...) の音符の k 個の組み合わせが n 回繰り返されている部分音符列について、(ABCD...) の音符の k 個の組み合わせが 1 つ減ると、その部分音符列は出現回数が 1 増えることを表している。

#### 4.2.2 本実験

4.2.1 の結果より、メロディ、ハーモニー、リズムは、表

2 の音楽情報を比較し、suffix tree を作成すれば抽出可能と考えた。それぞれ、メロディが A、ハーモニーが B、リズムが C に対応する。

本実験では、実際に音楽ゲームで使われているような楽曲である、ロマンシングサ・ガ 2 より伊藤賢治の七英雄バトル \*4 について行った。音の含まれるトラック数は 13 トラックである。MIDI の波形と音の確認には GarageBand \*5 を使用した。

図 9、図 11、図 13 の GarageBand 譜について、それぞれ下図が楽曲の全体を示しており、横軸が時間を表しており、縦軸がトラック（または楽器）ごとに分けられており、その中で音高を表している。また、上図は下図の四角で囲った一部分を拡大した図である。図 8 の (11,5) にプロットされた音の 1 つについて GarageBand で見ると、図 9 のように 1 つのトラックで音数 11 のメロディが 5 回繰り返されているのが確認できた。また、図 10 の (5,5) にプロットされた音の 1 つについて GarageBand

\*4 <http://aika.joo.lt/>

\*5 <https://www.apple.com/jp/mac/garageband/>

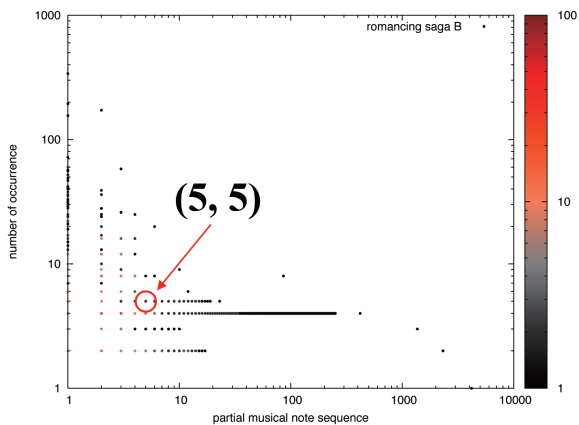


図 10 部分音符列出現頻度-ロマンシングサ・ガ B-  
Fig. 10 Frequency of partial musical note sequence -Romancing SaGa B-

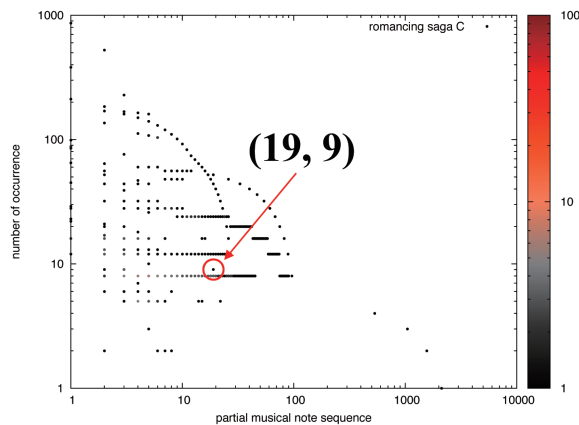


図 12 部分音符列出現頻度-ロマンシングサ・ガ C-  
Fig. 12 Frequency of partial musical note sequence -Romancing SaGa C-

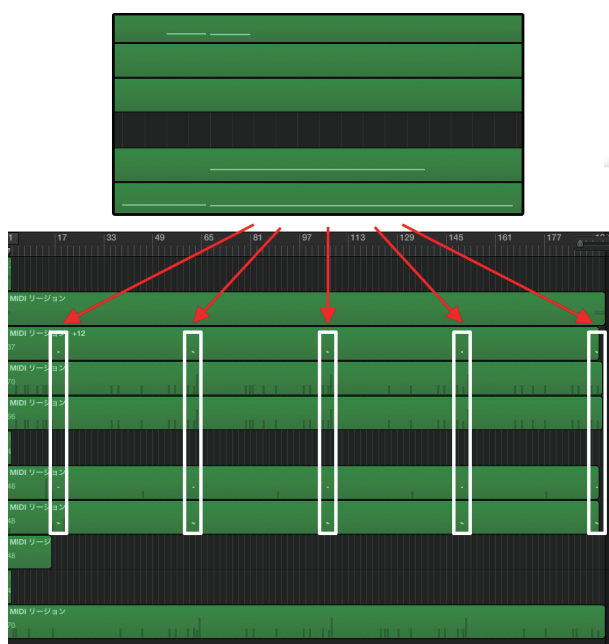


図 11 GarageBand 譜-ロマンシングサ・ガ B-  
Fig. 11 GarageBand-Romancing SaGa B-

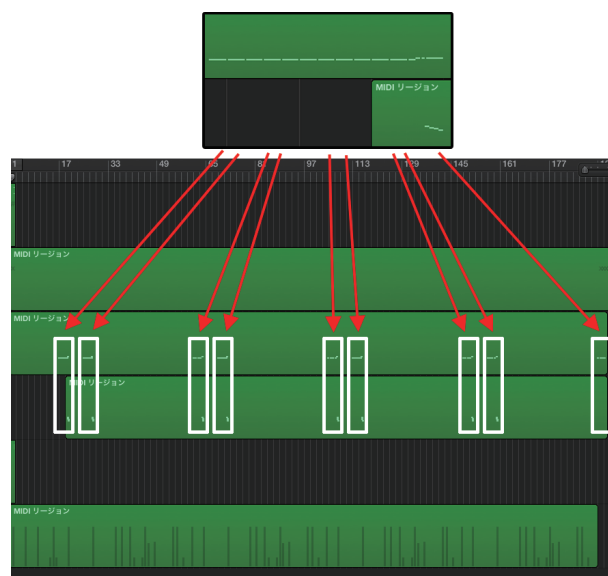


図 13 GarageBand 譜-ロマンシングサ・ガ C-  
Fig. 13 GarageBand-Romancing SaGa C-

で見ると、図 11 のように複数トラックにまたがって音数 5 の短いハーモニーが楽曲中に 5 回繰り返されているのが確認できた。同様に、図 12 の (19,9) にプロットされた音の 1 つについて GarageBand で見ると、図 13 のように複数トラックにまたがって音数 19 のリズムが 9 回繰り返されているのが確認できた。

#### 4.2.3 出力結果に対する評価

頻出頻度を利用した重要音抽出法を用いて抽出した音に対して評価実験を行った。評価実験には、情報理工学系の修士学生 7 名・博士学生 5 名・研究生 3 名・助教 1 名の計 16 名(うち、音楽ゲーム経験者 8 名)に参加してもらった。評価者は GarageBand 譜などを見ずに音のみを聴いてもらい、それぞれの抽出音に対し、原曲の印象をどれが一番抽出できているかを回答してもらった。楽曲は 4.2.2 で使用

した楽曲と同様、七英雄バトルを使用した。まず、評価者に原曲を聴いてもらい、その後、原曲から全てランダムに抽出した音の集合(1.ランダム抽出)、原曲から FBKCE 法により抽出した音に対し、楽器ごとかつ各小節ごとに均等間隔で抽出した音の集合(2.FBKCE 法+均等抽出)、原曲から FBKCE 法により抽出した音に対し、楽器ごとにランダムに抽出した音の集合(3.FBKCE 法+ランダム抽出)の 3 つを、抽出した方法など説明せず 1, 2, 3 の順番で聴いてもらった。

2, 3 において、それぞれのフレーズは図 14 のように抽出した。メロディフレーズは、音数 10 以上、出現回数 5 回以上の部分音符列を抽出した。ハーモニーは最低 3 音が必要であり、音数が増えれば不協和音になる可能性も増す [1][6]。また、音数が多すぎると本来分けて抽出したいはずのハーモニーが連続したハーモニーとして抽出される可能性があるため、ハーモニーフレーズは図 14 より、音

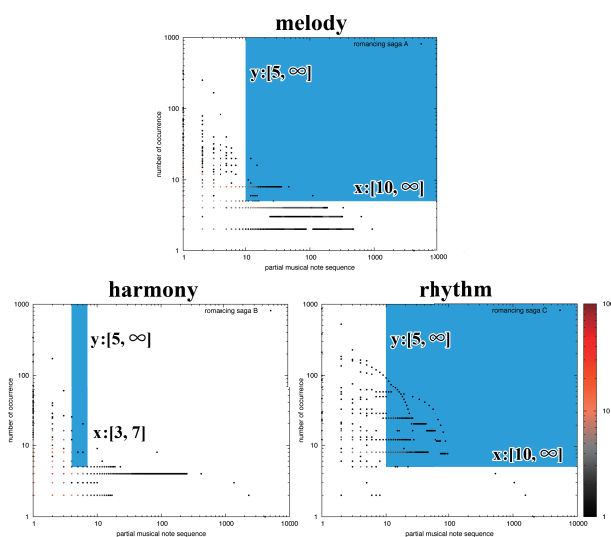


図 14 FBKCE 法における抽出範囲

Fig. 14 Range of extraction for frequency-based key component extraction.

表 3 ユーザーテスト結果

Table 3 User's testing.

	回答者数
1. ランダム抽出	2(0)
2. 重要音抽出+均等抽出	7(3)
3. 重要音抽出+ランダム抽出	7(5)

数 3 以上 7 以下，出現回数 5 回以上の部分音符列を抽出した．リズムフレーズは図 14 より，音数 10 以上，出現回数 5 回以上の部分音符列を抽出した．

また，2 の小節ごとの均等間隔での抽出について，音楽ゲームでは難易度を簡単に調整する時，間の音を抜く傾向にある（例えば，「ドレミファソ」という連続した音であれば「ド\_\_ミ\_\_ソ」と抽出する）．そのため，この抽出法を採用した．また，小節の切れ目によっては次の小節が裏拍で抽出されてしまうこともあるため，小節の最初の音は必ず抽出するようにしている．なお，原曲の音の数は 6,296 音なのに対し，実験の公平性を保つため，1 は 2,640 音，2 は 2,631 音，3 は 2,637 音と，ほぼ同じ音の数を抽出した．原曲の曲の長さは 4 分 50 秒程度であるが，前半と後半で曲の構成（音の構成）がほぼ全く一緒であるため，評価実験では前半部分の 2 分 30 秒程度を聴いてもらった．

その結果を表 3 に示す．表の回答者数は，1，2，3 のうち，どれが一番原曲の印象を抽出できているかについて回答した回答者の人数である．括弧内は，そのうちの音楽ゲーム経験者の人数である．結果より，少なくとも 1. ランダム抽出よりは 2，3 の FBKCE 法による抽出の方が原曲の印象を抽出できているであろうことが検証された．また，FBKCE 法による抽出の後の抽出について，2. 均等抽出と 3. ランダム抽出では意見が二分した．2 を回答した評価者の意見として，「一番原曲に近い印象を受けた」，「リズム

ム of 安定感がある」，「比較的旋律のようなものがあつた」などの意見があつた．また，3 を回答した評価者の意見として，「複数の楽器が満遍なく聴こえ，それぞれが原曲の印象を代表するようなメロディを構成していた」，「音がない部分があつたり，切れているところはややあるが，主旋律が一番多く残っている」などの意見があつた．中には，2 と 3 を両方混ぜるとよいのではないか，という音楽ゲーム経験者からの意見もあつた．

## 5. 本研究の到達点と今後の課題

音楽ゲームの譜面自動生成のために，音楽を離散的に構造解析した．音楽の中の重要なフレーズは繰り返し出現すると考え，FBKCE 法を提案・実装し，部分音符列の出現頻度を調べ，メロディフレーズ，ハーモニーフレーズ，リズムフレーズの抽出を可能にした．これにより，キー音候補の抽出が可能になった．また，抽出したそれぞれのフレーズをまとめ，そこからさらに均等抽出とランダム抽出を行うことで，様々な難易度への調整を可能にした．最後に，FBKCE 法により抽出した楽曲に対して評価実験を行った．その結果，ランダム抽出に比べ，FBKCE 法を用いた方がキー音候補の抽出はできていることを検証した．

本論文では，図 2 のキー音抽出とキー音割当の中のキー音抽出について述べたが，キー音割当については今後の課題である．まずは，FBKCE 法と 3.2 で実装したフレームワークの統合をしたい．また，プレイヤーコミュニティの中では一般化されている様々な音楽ゲーム特有の譜面特徴を考慮したキー音の譜面への割り付けを考えていきたい [17]．また，キー音抽出についても，FBKCE 法のフレーズ抽出の際の部分音符列の音数と出現回数のしきい値の最適化や，様々な難易度を実現するための FBKCE 法の後の抽出について考える必要がある．

音楽ゲームの譜面の自動生成を実現することにより，多種多様なジャンルの音楽を音楽ゲームとして楽しむことが可能になると期待される．また，人に合わせた様々な難易度やバリエーションの譜面を量産することが可能になり，その人にあった練習用の譜面の作成や，子供の教育，高齢者のリハビリなどに利用できると期待される [7]．さらには，音楽ゲームの面白さがどこから生まれてくるかの手がかりになると期待される．

## 参考文献

- [1] 芥川也寸志：音楽の基礎，岩波新書，pp 87-127 156-178 (1971).
- [2] Jekovec, M., J. Demsar, and A. Brodnik: *computer-aided-melodic-analysis-using-suffix-tree*, ICMA, University of Ljubljana (2012).
- [3] Ioannis Karydis, Alexandros Nanopoulos, Yannis Manolopoulos: *Finding maximum-length repeating patterns in music databases*, Multimedia Tools and Applications, Volume 32, Issue 1, pp 49-71(2007)

- [4] 高田友則, 橋口博樹 : *MIDI* におけるメロディ情報を利用した繰り返し構造の検出, 埼玉大学紀要. 工学部 第 1 編 第 1 部 論文集 39, pp 107-109 (2006).
- [5] P. Weiner: *Linear pattern matching algorithms*, SWAT '73 Proceedings of the 14th Annual Symposium on Switching and Automata Theory, pp 1-11 (1973)
- [6] 小方厚 : 音律と音階の科学, 講談社 , pp 61-64, 115 (2007)
- [7] Diana L. Graf, BS, Lauren V. Pratt, MS, Casey N. Hester, MD, Kevin R. Short, PhD : *Playing Active Video Games Increases Energy Expenditure in Children*, PE-DIATRICS Vol. 124 No. 2 August 1, pp 534 -540(2009)
- [8] 渋谷哲朗 : 接尾辞木について ,  
入手先 (<http://www.hgc.jp/tshibuya/classes/index.html>) (2015.04.30).
- [9] Jekovec, M., J. Demsar, and A. Brodnik : COMPUTER AIDED MELODIC ANALYSIS USING SUFFIX TREE, 入手先 (<http://lusy.fri.uni-lj.si/sites/lusy.fri.uni-lj.si/files/publications/jekovec2012-icmc12-article.pdf>) (2012).
- [10] コナミ株式会社 : beat gather,  
入手先 ([http://www.konami.jp/products/touch\\_beatgather/](http://www.konami.jp/products/touch_beatgather/)) (2011).
- [11] 株式会社テクノン : Any Beats (エニービーツ) ,  
入手先 (<http://www.anybeats.jp/>) (2011).
- [12] 株式会社ソニー・コンピュータエンタテインメント : 昇天ビート,  
入手先 ([http://www.jp.playstation.com/software/title/jp9000npjg00001\\_00syotenbeat000001.html](http://www.jp.playstation.com/software/title/jp9000npjg00001_00syotenbeat000001.html)) (2008).
- [13] Float 32 Inc. : The Real DJ,  
入手先 (<https://itunes.apple.com/jp/app/the-real-dj-rhythm-game/id467707681?mt=8>) (2011).
- [14] lavalse, cycليا, SHiNKA, Kamiyu, mur : LunaticRave2,  
入手先 (<http://www.lr2.sakura.ne.jp/index2.html>) (2008).
- [15] wikipedia : 音楽,  
入手先 (<https://ja.wikipedia.org/wiki/%E9%9F%B3%E6%A5%BD>)
- [16] wikipedia : BMS (音楽ゲーム),  
入手先 ([https://ja.wikipedia.org/wiki/BMS\\_\(%E9%9F%B3%E6%A5%BD%E3%82%B2%E3%83%BC%E3%83%A0\)](https://ja.wikipedia.org/wiki/BMS_(%E9%9F%B3%E6%A5%BD%E3%82%B2%E3%83%BC%E3%83%A0)))
- [17] 泡井七月 : TexTage[テクステーヂ],  
入手先 (<http://textage.cc/>) (2004)