

楽曲を対象とした難易度選択可能なリズムパートを再現する ピアノ伴奏生成方式

小宮 和真[†] 岡田 龍太郎[†] 峰松 彩子[†] 中西 崇文[†]

[†] 武蔵野大学データサイエンス学部 〒135-8181 東京都江東区有明 3-3-3

E-mail: [†] s2122027@stu.musashino-u.ac.jp, ryotaro.okada@ds.musashino-u.ac.jp,
ayako.minematsu@ds.musashino-u.ac.jp, takafumi.nakanishi@ds.musashino-u.ac.jp

あらまし 本稿では、楽曲の伴奏パターンリズムの特徴をベクトルとして抽出する伴奏パターンリズム特徴抽出手法を提案し、それを用いた、楽曲を対象とする難易度選択可能なリズムパートを再現するピアノ伴奏生成手法について示す。一般的に、楽曲をピアノ演奏のために独自でアレンジする際、特に伴奏となる左手パートに関して多くの知識が必要である。本方式は、元楽曲音源とそのコード進行データを入力として与えることで、難易度の異なるいくつかの伴奏楽譜を出力する。本方式では、入力された楽曲音源からドラムパートのみを分離した上でその音源からリズム特徴量を抽出し、伴奏パターンリズム特徴抽出手法を用いて事前に用意した伴奏パターンから抽出したリズム特徴量との類似度を計量することで、楽曲に合わせた伴奏パターンを採択する。その伴奏パターンに入力されたコード進行を当てはめることで、伴奏楽譜を自動生成する。本方式を実現することにより、任意の楽曲でのユーザ自身の演奏スキルに合致したピアノ伴奏楽譜の提案を実現し、ユーザにとって楽曲演奏の機会を増大させることが可能となる。

キーワード ピアノ伴奏自動生成, 楽曲メディアコンテンツ, リズム特徴抽出, 伴奏パターンリズム特徴抽出手法, 難易度選択

1. はじめに

近年、インターネットや DTM の普及により、膨大で多様な楽曲メディアコンテンツが生成、蓄積、配信されており、多くのユーザが気軽に膨大で多様な楽曲メディアコンテンツを楽しむ機会が増大している。これらの多様な楽曲メディアコンテンツを対象として、ユーザ自らアレンジし、演奏したいという要求も存在する。特に、ピアノなどの鍵盤楽器を楽しむユーザは多く、yahoo の楽器検索ランキングでは、ピアノが 3 年連続で上位 2 位以内に入っている[1]。

一般的に、楽曲を演奏する際に、楽譜を必要とするピアノ演奏者は多い。楽譜は主に、インターネットや楽器店などで販売、配信されている中から、自身の技量に合わせたものを選ぶ。しかしながら、膨大で多様なインターネット上の楽曲メディアコンテンツの中から演奏する楽曲を選択する場合、自身の技量に合致した楽譜が見つからない場合が多い。その際、音楽的素養の高い演奏者は、楽曲メディアコンテンツを聴きながら自力で演奏可能なアレンジをし、演奏を行うことができる。しかしながら、音楽的素養の低い初心者の方の演奏者においては、ユーザ自身が演奏可能なアレンジを自力で行うことは難しく、その楽曲の演奏を諦めることが多い。特に伴奏部分のアレンジに関しては、音楽的素養、知識を要することがあり、難解に感じる演奏者が多い。多くのピアノ演奏者がピアノ演奏をより楽し

むためには、各楽曲で、各演奏者が自身の技量に合致した楽譜を入手できるようにする必要がある。

ユーザが選択した楽曲メディアコンテンツを対象として、そのユーザの演奏技量に合致したピアノアレンジ楽譜を自動生成することが可能になれば、インターネット上の膨大で多様な楽曲からユーザが選択したとしても、ユーザがピアノ演奏を楽しむ機会が増大すると考えられる。

本研究では、任意の楽曲、難易度におけるピアノアレンジ楽譜の自動生成の実現に向け、まず自力でのアレンジが特に難しいとされる伴奏部分に着目し、任意の楽曲での伴奏楽譜の自動生成方式について考える。

本稿では、楽曲の伴奏パターンリズムの特徴をベクトルとして抽出する伴奏パターンリズム特徴抽出手法を示し、それを用いた、楽曲を対象とする難易度選択可能なリズムパートを再現するピアノ伴奏生成手法について示す。本方式では、楽曲音源とその楽曲のコード進行を入力とし、右手で演奏するメロディに合わせて左手で演奏するための伴奏楽譜を出力する。本方式では、伴奏楽譜を、コード進行とコードを構成する音をどのように配置するかを定めた伴奏パターンの組み合わせでなるものとする。また、コード進行は入力として与えることを想定し、楽曲に対して適切な伴奏パターンを提案する方式を考える。具体的には、入力された楽曲音源からドラムパートのみを分離した上

本方式を実現することにより、ピアノ演奏者がより膨大で多様な楽曲の中から自身の嗜好で楽曲を選択し、演奏する機会が増大すると考えられる。

本稿は以下のように構成される．2 節では，関連研究について示す．3 節では，本方式である難易度選択可能なリズムパートを再現するピアノ伴奏生成方式について示す．4 節では，実際に構成したシステムから伴奏楽譜を生成し，5 節で本稿をまとめる．

本節では、本方式に関連した研究について述べる。

越井ら[2]は、楽曲音源のみを入力とし、その楽曲のピアノ弾き語り用伴奏譜を自動で生成する手法を提案している。この研究[2]では、まず、88音分のピアノ単音のスペクトログラムのデータと、リズムパターンのデータを事前に用意する。そして、楽曲音源から推定したコードと用意した二種類のデータを組み合わせることで各伴奏パターンのスペクトログラムを生成し、楽曲音源のスペクトログラムと類似度を測ることで、適切な伴奏パターンを選択する。この研究は、楽曲音源のみを入力として、元楽曲の再現度を重視したピアノ弾き語り伴奏を生成することを目的にしている。本研究では、入力楽曲に対して様々な難易度の伴奏を提示する手法を実現する。これにより、入力音源に対して、よりユーザの演奏スキルに合致した伴奏パターンを提案できると考えられる。

本節では、楽曲を対象とした難易度選択可能なリズムパートを再現するピアノ伴奏生成方式について述べる。

図 1 に本方式の全体像を示す。本方式は、ドラムパート抽出機能、拍推定機能、音源分割機能、リズムのベクトル化機能、伴奏パターンのベクトル化機能、最適パターン抽出機能、コード・パターン合成機能からなる。本方式は、入力として楽曲全体にドラムパートを含むポップス楽曲を想定している。本方式では、楽曲のコードが変わるタイミングを区切りとし、それぞれの区画ごとに適切な伴奏パターンを推定することで、伴奏楽譜を生成する。

ユーザは、楽譜化したい楽曲の音源とその楽曲のコード進行を入力する．そこで、入力された楽曲音源から、ドラムパートの抽出と拍の推定を行う．さらに、推定された拍の位置と、入力されたコード進行を照ら



合わせることで、抽出したドラム音源をコードが変わるタイミングで分割する．この分割された各ドラム音源のリズムを事前に用意した各伴奏パターンとのリズムと比較し、類似性を調べることで、いくつかの難易度の適切な伴奏パターンを推定する．具体的には、ドラム音源のリズムと各伴奏パターンとのリズムをベクトル化し、各伴奏パターンとのベクトルの中で、ドラム音源のベクトルとのコサイン類似度が最も大きな値をとった伴奏パターンを採択する．ここで採択された伴奏パターンに、入力されたコード進行が持つ音高情報を合成することで、いくつかの難易度の適切な伴奏楽譜が生成される．

本節では、楽曲メディアコンテンツからドラム音源のみを抽出する機能について述べる。一般的に、伴奏のパターンは主にリズムに起因することが多く、その楽曲メディアコンテンツからリズムを一番よく表現す

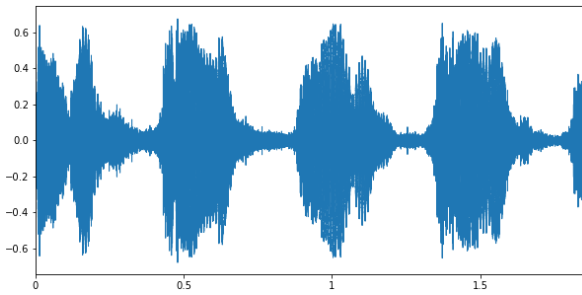


図 2 ドラム音源の音声波形の例

るドラムパートのみを抜き出し、そこからリズムの特徴を抽出することが重要であると考え。本機能は、楽曲メディアコンテンツからリズムの特徴を抽出するためのドラムパートを抜き出す機能である。

ドラムパートの抽出には、`spleeter[3][4]`を用いる。`spleeter`とは、音源分離を行うためのオープンソースのソフトウェアである。`spleeter`を用いて、入力された音源を、ドラムパートを含むいくつかのパートに分離させることで、ドラム音源のみを抽出する。

3.3 拍推定機能

本節では、楽曲内の拍の位置を推定する機能について述べる。本方式では、楽曲のコードが変わるタイミングを区切りとし、区画ごとに適切な伴奏パターンを推定する。入力されたコード進行を元に、楽曲を正確に区切るために、拍の推定が必要となる。各楽曲メディアコンテンツについて、拍を正しく推定し、保持する機能となる。

拍の推定には、`librosa[5][6]`を用いる。`librosa`は、音楽を解析するためのpythonパッケージであり、`librosa`の`beat`モジュールを用いることで、元楽曲音源から拍の位置を推定することができる。

一方、よりリズムを強調して拍抽出を行うために、元楽曲音源からではなく、3.2節で示したドラムパート抽出機能において抽出されたドラム音源を対象として、拍推定するという手法も考えられる。しかしながら、実際に前実験を行った結果、拍の細かさが2倍ないしは1/2倍になることがあり、精度の低下が見られたため、元楽曲音源からの推定を行う。これは、ドラムパートのみならず、実際の伴奏パートに含まれる要素が拍推定に貢献していると考えられる。

3.4 音源分割機能

本節では、抽出されたドラム音源を楽曲のコードが変わるタイミングで分割する機能について述べる。

本方式では、伴奏パターンの変化をコードが変わるタイミングで指定することを想定している。本機能は、コードの切り替わるタイミングを抽出するもので、抽出されたタイミングを伴奏パターンの切り替わるタイミングとみなすものである。

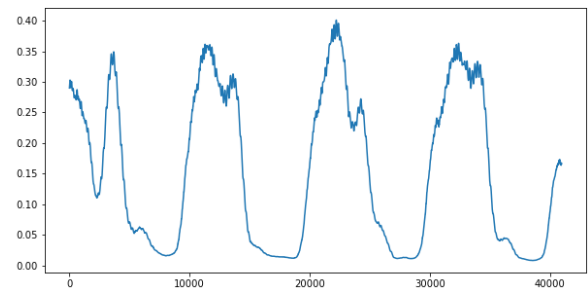


図 3 図 2 に示すドラム音源の波形データの振幅の絶対値を取り、移動平均を取った値

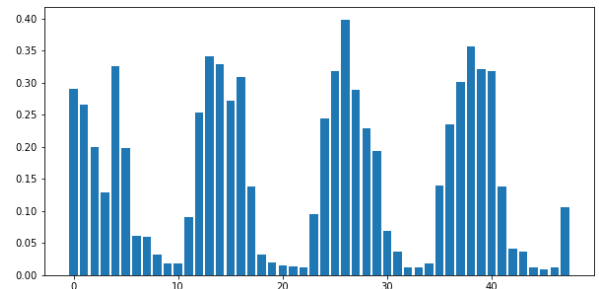


図 4 図 3 をリサンプリングしたベクトル値

抽出されたドラム音源は、推定された拍の位置と、ユーザによって入力されたコード進行を参照することで、楽曲のコードが変わるタイミングで分割される。

また、入力するコード進行は、D が 2 拍→A が 2 拍→Bm が 4 拍→…のように、コードとその拍の長さを示した配列を入力する。この配列内の拍の数を表した数値にのみ着目し、曲の頭から 2 拍分、2 拍分、4 拍分のように切り取ることで、ドラム音源を楽曲のコードが変わるタイミングで分割する。

3.5 リズムのベクトル化

本節では、分割された各ドラム音源のリズムをベクトルとして表す手法について述べる。

3.2 節のドラムパート抽出機能から抽出されたドラム音源からリズムの特徴量を抽出する機能である。

ある区間におけるドラム音源の音声波形の例を図 2 に示す。一般的に、楽曲におけるドラムパートは、その楽曲のリズムを表す上で最も重要なパートとなっている。つまり、図 2 に示したようなドラム音源の波形は、その区間における楽曲のリズムを明瞭に表していると考えられる。そこで、値をグラフとして表した際に、図 2 のような波形と概形が似るベクトルを生成する。厳密には、値をグラフとして表した際に、ドラム音源の音声波形が縦に伸びている位置と同じ位置で山になるような一次元の配列を生成する。

生成の手順として、まず、ドラム音源の振幅を絶対値で表した配列を生成する。次に、移動平均法を用いて、配列を均す。図 2 に示すドラム音源の波形データにここまでの手順を適応させた配列のグラフを図 3 に

示す。この時点で、ドラム音源のリズムをベクトルで表すことは成功しているが、伴奏パターンのリズムのベクトルとの比較を行うため、二つのベクトルの次元数を揃えることについて考える必要がある。そこで、1 拍あたりの次元数を定め、その次元数に合わせて二つのベクトルの次元数を揃えるためにリサンプリングを行う。具体的には、図 3 で示したような配列から、定めた次元数に合わせて等間隔に値を取得する。

1 拍あたりの次元数を 12 とし、図 3 で示した配列から新たな配列を生成する。それをグラフ化したものを図 4 に示す。本方式では、このような配列をドラム音源のリズムのベクトルとする。

3.6 伴奏パターンリズム特徴抽出手法

本節では、伴奏パターンリズム特徴抽出手法について述べる。

伴奏パターンリズム特徴抽出手法とは、伴奏パターンからその音が鳴っているタイミングをベクトルとして表現するとともに、音が低い場合にリズムに貢献する度合いが高いという仮説から、そのベクトルの要素を音高に沿った重み付けを行う手法である。一般的に低音楽器、および低音部の音はその楽曲においてリズムに貢献する可能性が高いと考えられる。そのことから、音高が低い音に対応するベクトルの要素について、音高に対応した重み付けを行うことで、よりリズムの特徴を表現できる。

本機能は、事前に用意した各伴奏パターンからリズムの特徴量をベクトルとして抽出する機能である。これにより、3.5 節で抽出されたベクトルと類似度を計量することで、楽曲に対して適切な伴奏パターンを決定することができる。

伴奏パターンのリズムのベクトルとして、伴奏パターン内で音が鳴る瞬間を 1、他を 0 とした要素によるベクトルを考える。その際、ベクトルの次元数は、ドラム音源のリズムのベクトル化の際に定めた 1 拍あたりの次元数に準拠する。

このベクトル化手法では、音が鳴るタイミングのみに着目しているため、伴奏パターンが持つ音高の情報が含まれていない。リズムのベクトル化であることを考えると、音高の情報を含む必要はないとも考えられるが、音が鳴るタイミングが同じであっても、音高によって印象が大きく異なる場合が存在する。そこで、伴奏パターンのリズムをベクトル化する際に、音高によって重み付けを行う手法を考える。具体的には、音高が低い音の方がリズムへの影響が大きいと仮定し、伴奏パターンのベクトルの要素を音高が低くなるほど値が大きくなるように重み付けを行う。

各要素の重みは $a^{\frac{n}{12}}$ とする。 a は 0 より大きく 1 より



図 5.2 拍分の伴奏パターンの例

小さい実数とし、 n は半音の高低を 1 とする音の高さを表す自然数とする。こうすることで、音が半音 12 個分、すなわち 1 オクターブ上がるごとに重みが a 倍されるようになり、 a の値が小さくなるほど、低い音に対応するリズムがより強調されるベクトルが生成される。また、同時に複数の音が鳴っている場合、一番低い音を基準に重み付けを行う。例として、ある 2 拍分の伴奏パターンを図 5 に示す。1 拍あたりの次元数を 2、 a の値を $1/2$ 、この伴奏パターン内の最も低い音にあたる n の値を 0 とした場合、この伴奏パターンのリズムのベクトルは、 $[1, 0, 0.33, 0.5]$ となる。

本研究では、この重み付け手法を用いる場合と用いない場合の二種類の手法で実験を行う。

3.7 最適パターン抽出機能

本節では、事前に用意した伴奏パターンの中から、ドラム音源から生成した各ベクトルに対して最も適切な伴奏パターンを抽出する機能について述べる。

本機能では、3.5 節で抽出されたドラム音源のリズムのベクトルと事前に用意した各伴奏パターンのリズムのベクトルを類似度計量することにより、最適な伴奏パターンを決定する機能である。

本方式では、楽曲のリズムと最も近いリズムを持った伴奏パターンを最適パターンとしている。そこで、難易度ごとに、ドラム音源から生成した各ベクトルと、事前に用意した伴奏パターンの中で拍の数が一致する各パターンのベクトルとのコサイン類似度を測り、値が最も大きくなった伴奏パターンを最適パターンとして抽出する。分割された各ドラム音源のリズムのベクトルに対して最適パターンを求めることで、楽曲全体の伴奏パターンが定まる。

3.8 コード・パターン合成機能

本節では、コード進行と伴奏パターンを合成することで、楽譜情報を生成する機能について述べる。

本機能は、3.7 節で採択された各伴奏パターンに、入力されたコード進行が持つ音高情報を合成することで、伴奏楽譜を生成する機能である。入力として与えたデータや各機能からの出力を統合し、伴奏楽譜を生成するための機能となる。

伴奏パターンのデータベースには、伴奏パターンが、

表 1 事前に用意した伴奏パターンの数

	Level 0	Level 1	Level 2
1 拍	4	4	4
2 拍	10	10	13
3 拍	6	6	6
4 拍	22	20	14

コードを構成する 3 つの音がどのように配置されているかを示した情報として格納されている。コードを構成する 3 つの音は、低い音から順に、0 番目の音、1 番目の音、2 番目の音と数値を割り当てることで区別する。図 5 に示した伴奏パターンを例に考える。この伴奏パターンは、コードを C とし、C2 の音を基準とした場合、「0 番目の音と 1 オクターブ上の 0 番目の音が 1 拍、1 オクターブ上の 2 番目の音と 2 オクターブ上の 0 番目の音が 0.5 拍、1 オクターブ上の 0 番目の音が 0.5 拍」と表すことができる。そして、3 つのそれぞれの音に任意のコードを構成する音に対応させることで、任意のコードで同じリズムを持った伴奏楽譜が生成される。また、現在用意した伴奏パターンでは構成音が 3 種類のものにのみ対応しているため、セブンスコードなどの 4 つ以上の音を持ったコードの場合は、低い音から順に 3 つの音を選択し、その 3 つの音のみを使用する。構成音が 4 つ以上のコードについての対応は今後の課題とする。

本機能では、3.7 節で採択された各伴奏パターンに、以上のような処理を行うことで伴奏楽譜を生成する。

4. 評価実験

本節では、本方式の評価実験について述べる。3 節で示した方式を実装し、実際に生成した伴奏についての評価を行う。4.1 節では、実験の概要について述べる。4.2 節では、実験 1 について述べる。4.3 節では実験 2 について述べ、4.4 節で総合的な考察を行う。

4.1 実験の概要

まず実験 1 では、3.6 節で述べた伴奏パターンのリズムのベクトル化の際に、重み付け手法を用いる場合と用いない場合での出力の違いについて検証する。

また、本方式の実用化には、本方式を用いて生成した伴奏が原曲に対して適切なものであり、かつ、多様なピアノ演奏者がピアノ演奏を楽しめるものである必要がある。以上を踏まえ、実験 2 では、「生成した伴奏のリズムの再現度」、「難易度の違いを含めた総合的な完成度」の二つについて被験者実験を行い、検証を行う。

4.2 実験 1

本節では、伴奏パターンのリズムをベクトル化する際の、音が鳴った瞬間の値の音高に応じて重み付けを行う手法と行わない手法について比較実験を行う。



図 6 音が鳴るタイミングは同じだが鳴らす音が異なる伴奏パターンの例

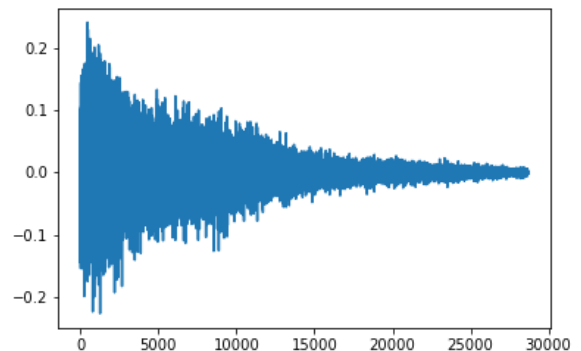


図 7 アルペジオが採択された場面のドラム音源の音声波形の例

4.2.1 実験方法

二種類の手法に関する実験では、いくつかの楽曲を元に生成した複数難易度の伴奏楽譜に、楽曲のメロディを乗せた MIDI ファイルをそれぞれの手法で生成する。また、音高によって重み付けを行う手法では、重みにおける a の値を $1/2$, $2/3$, $3/4$ に設定し、それぞれの値で実験を行うことで a の値についても評価を行う。

この実験には、歌ネット 2021 歌詞検索ランキング [7] の上位 5 曲である、うっせえわ/ado、夜に駆ける/YOASOBI、ドライフラワー/優里、廻廻奇譚/Eve、炎/LiSA を使用し、生成する伴奏の難易度は 3 段階とする。この実験を行うにあたり事前に用意した伴奏パターンの数を、難易度、拍数ごとに表 1 に示す。

また、本方式は楽曲全体にドラムパートが存在することを前提に構成されているが、先で述べた 5 つの楽曲にはドラムパートが存在しない範囲が含まれている楽曲が存在するため、ドラムパートが存在する、主にサビ部分のみを対象として MIDI ファイルを生成する。

4.2.2 実験結果

まず初めに、伴奏パターンのリズムのベクトル化における二種類の手法に関する実験結果について述べる。

伴奏パターンのリズムをベクトル化する際に重み付けを行わずに伴奏を生成する際の大きな問題として、音が鳴るタイミングが全く同じ伴奏パターンを使い分けることができないという点が挙げられる。この問題は、ベクトルの値に重み付けを行うことで解決される。

ことが期待できる。

図 6 に例として二つの伴奏パターンを楽譜として示す。この二つの伴奏パターンは、音が鳴るタイミングは全く同じであるが、鳴らす音が大きく異なるため、実際に演奏した際の印象は大きく異なる。音高によってベクトルの値に重み付けを行わない場合、この二つの伴奏パターンは同じパターンとして扱われてしまうが、重み付けを行う場合、この二つの伴奏パターンは違うパターンとして識別されるため、場面によって使い分けることができる。実際にベクトルの値に重み付けを行い生成した MIDI ファイルの中には、図 6 に示した二つの伴奏パターンが使い分けられているのが見られた。

図 6 の右側に示されている一つの伴奏パターンについて考える。以下、この伴奏パターンをアルペジオと呼ぶ。重み付けを行う手法における、アルペジオが採択された場面のドラム音源の音声波形の例を図 7 に示す。この場面は鳴る音が冒頭のシンバルのみという場面で、アルペジオが事前に用意されていた難易度では、実験を行なった全ての a の値でアルペジオが採択された。

次に、重みにおける a の値についての実験結果を述べる。 a の値は $1/2$, $2/3$, $3/4$ の三つを用意し、それぞれの値で実験を行なった。

a の値に変化が起きることで採択される伴奏パターンが変化する割合について調べる。実験では、5 つの楽曲で難易度 3 つ分の計 15 個の MIDI ファイルを生成する。それぞれの a の値でこの実験を行い、採択される伴奏パターンが a の値によって変化する割合について調べる。15 個の MIDI ファイルを生成する際の伴奏パターンを推定する機会は合計 357 回であった。その内、3 つの a の値で同じ伴奏パターンが採択された回数は 258 回であり、割合は約 0.72 となった。

4.2.3 考察

実験結果から、重み付けを行う手法の場合、図 7 で示したような音声波形から感じ取ることができるフェードアウトするような印象は、アルペジオで再現され则认为。また、事前に用意した 4 拍分の伴奏パターンの中には、全音符で 4 拍分音を伸ばすという伴奏パターンが含まれており、リズムの再現性に着目した際に一般的に選ばれやすいのは、こちらの伴奏パターンだと考える。このことから、図 7 で示した音声波形をアルペジオで表現することが適切だと仮定すると、重み付けを行うことで、一般的には思いつきづらい伴奏パターンを提案することが可能になると考えられる。また、本システムを自力での伴奏アレンジを行う際に同時に使用し、出力結果を一つの選択肢として活用することも可能になると考える。

表 2 アンケート結果

	リズムの再現度	総合的な完成度
うっせえわ level 0	4.57	3.50
うっせえわ level 1	4.29	3.17
うっせえわ level 2	4.00	4.00
夜に駆ける level 0	4.14	3.67
夜に駆ける level 1	4.00	3.50
夜に駆ける level 2	4.71	4.67
ドライフラワー level 0	4.29	3.83
ドライフラワー level 1	4.43	3.83
ドライフラワー level 2	4.57	4.67

重みにおける a の値については、実験結果に示した通り、 a の値の変化による伴奏パターンの大きな変化は見られておらず、ある値を指し、この値が適切と言い切ることとはできないと考える。今後の発展として、楽曲のテンポの参照や A メロ、B メロ、サビなどの推定をし、遅い曲のサビ部分では音域を広く活用することで壮大に見せるために高い音の影響を大きくするなど、場面に応じた使い分けが可能となれば、より適切な伴奏楽譜を提案できると考える。

4.3 実験 2

本節では、生成した伴奏のリズムの再現度と、難易度の違いを含めた総合的な完成度について実験を行う。

4.3.1 実験方法

この実験では、実験 1 の内容を踏まえて伴奏生成における手法を定め、実際に生成した伴奏についてアンケートをとる。この実験では、伴奏パターンのリズムのベクトル化の際に重み付けを行う手法を用いて伴奏を生成する。また、重みにおける a の値は $2/3$ とする。この実験では、ピアノ演奏の経験者 6 人を含む計 7 人の大学生を被験者とした。

具体的な条件については、まず実験 1 同様、いくつかの楽曲を元に 3 難易度のメロディ付きの MIDI ファイルを生成し、次にそれぞれの MIDI ファイルをピアノロールの映像と共に再生させた動画を被験者に見せ、質問項目に対して主観で評価させる。また、事前に用意した伴奏パターンは実験 1 と同様である。

使用する楽曲は実験 1 同様、歌ネット 2021 歌詞検索ランキングより選択するが、被験者の負担を考え上位 5 曲ではなく上位 3 曲のうっせえわ、夜に駆ける、ドライフラワーを使用し、難易度は 3 段階の合計 9 個の MIDI ファイルについてアンケートを行う。MIDI ファイルの再生に使用するソフトウェア音源には SYNTHOLOGY/Ivory II American Concert D[8]を使用し、ピアノロールの映像は domino[9]を用いて撮影する。評価する項目は「生成した伴奏のリズムの再現度」、「難易度の違いを含めた総合的な完成度」の二つとし、そ



図 8 夜に駆けるの level 2 の楽譜

それぞれ 5 段階で評価させる．難易度の違いを含めた総合的な完成度については，被験者をピアノ経験者のみとする．また，楽曲ごとに被験者の自由なコメントを収集することで，難易度の違いなどを含めた，システムに対する改善点を考察する．

4.3.2 実験結果

アンケート結果を，生成した楽譜ごとに平均値をとり，表 2 に示す．level は難易度を表しており，数値が大きいものほど難易度の高い楽譜となっている．本方式を用いて生成した伴奏の例として，アンケートでの評価が二つの項目で最も高い夜に駆けるの level 2 の楽譜を図 8 に示す．また，難易度の比較として，夜に駆けるの level 0~2 について，各楽譜の冒頭 1 段目を抜粋し図 9 に示す．

また，楽曲ごとに収集した被験者の意見を一部抜粋する．夜に駆けるに対しては，「level 1 の楽譜で 8 分音符がより強調されると良い」や，「易しい楽譜ではオクターブを調整して近い音が選択されるようにすると良い」という意見が集まり．ドライフラワーに関しては，「level 0 の楽譜は易しく仕上がっていて良いが，level 1, level 2 の楽譜はもう少し難易度を上げてほしい」という意見が集まった．ドライフラワーの level 2 の楽譜を図 10 に示す．

4.3.3 考察

まず初めに，全体的なアンケートの結果について考



図 9 夜に駆けるの level 0~2 の楽譜の冒頭 1 段目
(上から level 0, 1, 2)



図 10 ドライフラワーの level 2 の楽譜

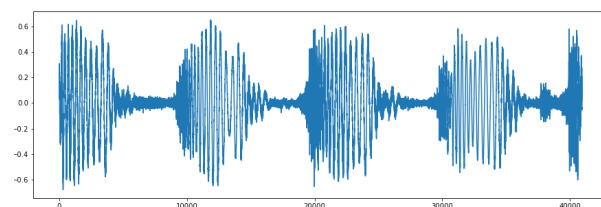


図 11 夜に駆けるのサビの初めから 1 小節分のドラム音源

察する．表 2 に示した平均値はいずれの楽譜，項目で，中間である 3 よりも大きな値となった．また，相対的な評価として，リズムの再現度に対して総合的な完成度の方が低い値を取っている楽譜が多い．値としては，リズムの再現度，総合的な完成度共に高い評価を得ら

れており、特にリズムを再現するという部分については、ある程度実現できていると考えられる。しかし、被験者から収集した具体的な意見をみると、実用化には多くの課題があり、特に総合的な完成度については、本方式で考慮できていない要素が多く存在すると思われる。

次に、被験者から収集した具体的な意見についての考察を行う。夜に駆けるの level 1 に対する、「8 分音符がより強調されると良い」という意見について、まず夜に駆けるのサビの初めから 1 小節分のドラム音源の音声波形を図 11 に示す。このドラム音源のリズムは 8 ビートであるが、音声波形に着目した場合、裏拍にあたる位置の振幅が小さくなっているため、結果的に 4 ビートのような音声波形になっている。その結果として、4 ビートの伴奏パターンが採択されるようになったのだと考える。

また、ドライフラワーの楽譜の難易度についての意見についても考察を行う。難易度が全体的に易くなっている原因として、楽曲のテンポが比較的遅いことが挙げられる。ドライフラワーの bpm は 74 と実験に使用した 3 曲の中では最も遅いが、伴奏パターンのデータベースは他の楽曲と同じものを使用している。そのため、テンポが遅くなったことによる難易度の易化のみが楽譜として反映されていると考えられる。

4.4 実験全体の考察

本節では、実験全体の結果を踏まえての考察を行う。実験 1 では、伴奏パターンのリズムをベクトル化する際の重み付け手法について調査し、主に重みにおける a の値の調整の必要性などを考察した。実験 2 では、実際に生成した伴奏についてアンケートをとることでリズムの再現度と総合的な完成度について考察し、本方式の課題を明らかにした。以上の実験から、本方式を用いた自動伴奏生成の実用化には、楽曲に応じたより柔軟な対応が必要だと考える。伴奏パターンを採択する際に、ドラム音源から得られたリズムのベクトルだけでなく、それぞれの考察で述べたように、A メロ、B メロ、サビのような楽曲の構成情報や、テンポの情報などを加味した上での採択が可能になれば、入力した楽曲に対してより適切な伴奏楽譜をユーザに提案できると考える。

5. おわりに

本稿では、伴奏パターンリズム特徴抽出手法を用いた、楽曲を対象とする難易度選択可能なリズムパートを再現するピアノ伴奏生成手法について述べた。本研究では、楽曲音源とコード進行を入力とし、事前に用意した伴奏パターンとのリズムが最も近い伴奏パターンを抽出することで、楽曲に対して適切な伴奏楽譜を出力した。また、出力する伴奏楽譜を難易度ごとに生成す

ることで、多様なピアノ演奏者に対して各演奏者の技量に合致した伴奏楽譜の提案を目指した。生成した楽譜に対して被験者にアンケートをとった結果、本方式におけるリズムの再現性が確認された。

本方式の実用化については、楽曲に対してより柔軟な対応を可能にする必要があると考える。今後の発展として、ドラム音源以外からのリズム特徴量抽出を可能とすることで、ドラムパートのない場面が存在する楽曲への対応が挙げられる。また、コード進行の自動推定を行い、ユーザ自身が入力すべき情報を削減することで、より実用的な方式になると考える。本方式をさらに発展させ、多様なピアノ演奏者が、任意の楽曲で自身の技量に合致した伴奏楽譜を容易に入手することが可能になれば、初心者をはじめとした多くのピアノ演奏者がピアノ演奏をより楽しむことができるようになると思われる。

参考文献

- [1] 2021 年「楽器」検索ランキング 1 位は「ピアノ」 - トピックス - ヤフー株式会社, <https://about.yahoo.co.jp/topics/20210603.html>
- [2] 越井琢巳, 斎藤博昭, "スペクトログラム間の類似度に基づくピアノ弾き語り用伴奏譜の自動生成", 研究報告音楽情報科学 (MUS), 2017-MUS-115, 49, pp. 1-6, 2017.
- [3] deezer/spleeter: Deezer source separation library including pretrained models., <https://github.com/deezer/spleeter>
- [4] R. Hennequin, A. Khelif, F. Voituret, M. Moussallam. "Spleeter: a fast and efficient music source separation tool with pre-trained models". Journal of Open Source Software, 5(50), 2154, 2020.
- [5] librosa — librosa 0.8.1 documentation, <https://librosa.org/doc/latest/index.html>
- [6] B. McFee, C. Raffel, D. Liang, D. P. Ellis, M. McVicar, E. Battenberg, O. Nieto, "librosa: Audio and music signal analysis in python", In Proceedings of the 14th python in science conference, vol. 8, pp. 18-25, 2015.
- [7] 2021 年 年間アクセスランキング TOP30- 歌ネット, https://www.uta-net.com/close_up/2021_ranking
- [8] Ivory II American Concert D | Media Integration, Inc., <https://www.minet.jp/brand/synthogy/ivory-american/>
- [9] MIDI 音楽編集ソフト「Domino (ドミノ)」 | TAKABO SOFT, <https://takabosoft.com/domino>