

リズムアクションゲームにおけるキー音の自動推定

福永大輝¹⁾ (非会員) 越智景子²⁾ (非会員) 大淵康成²⁾ (正会員)

1) 東京工科大学大学院バイオ・情報メディア研究科メディアサイエンス専攻

2) 東京工科大学メディア学部

Automatic Estimation of Key-sound in Rhythm Action Games

Daiki Fukunaga¹⁾ Keiko Ochi²⁾ Yasunari Obuchi²⁾

1) Graduate School of Bionics, Computer and Media Science, Tokyo University of Technology

2) School of Media Science, Tokyo University of Technology

g311801818@edu.teu.ac.jp, { ochikk , obuchiysnr }@stf.teu.ac.jp

アブストラクト

「リズムアクションゲーム」というジャンルに分類されるゲームにおいては、プレイヤーの操作に応じたサウンドが再生されることで楽曲演奏を体験できることに重点を置いたものが多く存在する。本稿では、このリズムアクションゲームを対象に、ゲームデータの自動生成を目的に「キー音」の自動推定を行った。「キー音」とはプレイヤーの操作に応じて発音されるサウンドのうち、もともと楽曲中に含まれる音を切り出したものである。ある発音時点に存在するサウンドがキー音となるか否かについて、音響的特徴とサウンドが再生される時系列情報を使用して機械学習による推定を行った。その結果、同じ楽曲のみから学習を行う場合では 90%程度、他の楽曲とデータを混合して学習を行う場合であっても 60%以上の精度を得られることがわかった。キー音の音響的特徴、またサウンドの再生時系列についての分析結果は、今後リズムアクションゲームのゲームデータの自動生成を目指すにあたり、大きな基盤になると考えられる。

Abstract

In the genre of "rhythm action games," many games emphasize the experience of performing a musical piece by producing sounds according to the player's action. In this paper, automatic estimation of "key-sound" was performed for the purpose of automatic generation of the game data of rhythm action games. A key-sound is the sound object extracted from the music and played according to the player's action. We applied machine learning to estimate whether or not a specific sound object at a certain time should be a key sound, using the acoustic features and time series information of sound events. As a result, it was found that the accuracy of about 90% was achieved when the training and estimation were conducted using the data of the same musical piece. Even when the training and estimation were conducted using different musical pieces, the accuracy was higher than 60%. The analysis results on the acoustic characteristics of the key-sounds and their time series are expected to form the basis of automatic generation of the game data of rhythm action games.

1. はじめに

音楽を主軸に置いたゲームのジャンルとして「音楽ゲーム」がある。その中でも特に、楽曲のリズムに合わせた「譜面」と呼ばれる視覚的な記号に沿ってボタンなどのデバイスの操作を行い、プレイヤーの操作の正確さにより得点が記録され高得点を目指すものを「リズムアクションゲーム」と呼ぶ[1][2]。

リズムアクションゲームにおいては、多くの場合プレイヤーが行った操作に応じてサウンドが再生される。このとき、図1のように「タップ音」または「判定音」などと呼ばれる楽曲とは無関係の効果音が楽曲に上乗せされるものと、図2のように「キー音」と呼ばれる楽曲中の音を抽出して切り出したものの2つに再生されるサウンドを分類することができる。



図1. 操作による発音の例（タップ音）



図2. 操作による発音の例（キー音）

プレイヤーが譜面に沿った演奏に失敗した場合に、タップ音を用いたゲームでは楽曲とは無関係の効果音の再生に影響が生じるだけであり、楽曲自体に変化は起こらない。一方、キー音を用いたゲームでは楽曲中のキー音が再生されない、もしくは誤ったタイミングで再生される。つまりプレイヤーの操作によって楽曲自体が変化するため、キー音を用いたゲームは実際の楽器演奏などに近いインタラクティブ性を持っているといえる。

音楽ゲームにおけるサウンドがもたらすインタラクティブ性については Pichlmair らにより詳細な考察がなされているが[3]、リズムアクションゲームの操作音に注目した場合、タップ音と比較してキー音はプレイヤーにより高い演奏感をもたらしていることが考えられる。

現在稼動中のキー音を採用しているリズムアクションゲームは「beatmania IIDX」や「pop'n music」、タップ音を採用しているリズムアクションゲームには「太鼓の達人」や「REFLEC BEAT」などのシリーズが存在する。実際に稼動している筐体を図3、および図4に示す。図3が beatmania IIDX、図4が太鼓の達人の筐体である。



図3. beatmania IIDX の筐体



図4. 太鼓の達人の筐体

本研究では、譜面の存在しない任意の音楽データから譜面を自動生成することを目指す。キー音を用いたリズムアクションゲームの譜面を作成するにあたって、必要とされる過程を図5に示す。ここで自動的に再生される音(BGM)とは、プレイヤーの操作にかかわらず再生される音である。操作が全くされない場合は楽曲中の個々の音のうち BGM のみが再生され、キー音は再生されない。

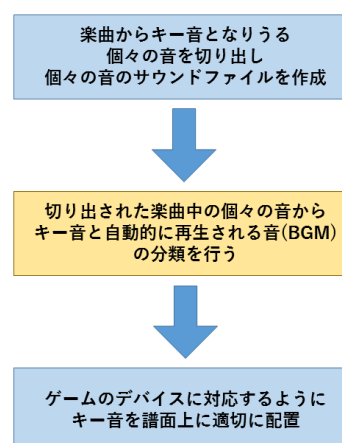


図5. キー音を用いたゲームにおける譜面作成過程

第1過程である楽曲中の個々からの音の切り出しについては、楽曲の音色ごとに独立した音源から個々の音を切り出すことが必要であることから多くの手間がかかる。このとき、切り出しただけでは元の楽曲が再現できないため、全ての音のファイル名と発音時点が保持される。

第2、第3過程についても、楽曲からどのような音をキー音として選択し、そのゲームのデバイスに合わせた譜面上に配置することが望ましいかについては研究が不十分であり、キー音の選択および譜面上への配置の良し悪しについての判断は、譜面データ制作者の感覚や経験に委ねられているのが現状である。

そこで、キー音を用いたリズムアクションゲームの開発に貢献することを目的に、譜面を作成するために必要な3つの過程を支援・自動化するシステムを構築することを最終目標として研究を行う。本研究ではサウンドファイル名と発音時点、また処理中で以前の発音時点のデータから再帰的に得られるキー音と BGM のラベルを使用し、第2過程であるキー音と BGM の分類を自動化することを目指す。

我々はこれまで、実際のリズムアクションゲームの楽曲中で用いられているサウンドファイルが持つ音響的特徴を用いて、予め切り出された個々のサウンドファイルがキー音として選択される確率について予測を行ってきた[4]。本研究では、実際のゲームデータに含まれるサウンドファイルについて音響的特徴に加え発音される時系列の抽出を行い、あらかじめ楽曲から切り出されている個々の音がキー音となるか否かについて、機械学習による推定を行った。

2. 関連研究

2.1 キー音のないゲームにおける事例

プレイヤーの操作による発音が存在しないリズムアクションゲームである「Dance Dance Revolution」について、Donahueらにより楽曲が持つスペクトル情報とニューラルネットワークを利用して自動的に譜面を生成する手法が提案されている[5]。

PCでプレイ可能なDance Dance Revolutionのクローンゲームであり、インターネット上に多数公開されている「StepMania」のデータ223曲分を分析の対象とすることで、ニューラルネットワークを用いたモデルに十分な大量の学習データを得ている。

譜面データの自動生成手順としては、まず楽曲データのスペクトル情報をSTFT(短時間フーリエ変換)により得たうえで、スペクトルが持つピークなどの特徴をCNN(畳み込みニューラルネットワーク)に入力することでオブジェクトを配置するのにふさわしい時点を特定し、さらに長期的な時系列の学習に適した2層のLSTM(Long short-term memory)[6]モデルを用いて、設定した難易度に応じた最終的なオブジェクトの配置を行っている。

高い難易度の譜面と比較して低い難易度の譜面の品質が低いことが問題点として挙げられるため、今後の精度改善が期待される。また、キー音を用いたリズムアクションゲームにおける譜面自動生成については想定されていない。

2.2 キー音のあるゲームにおける事例

キー音を用いたリズムアクションゲームにおける譜面自動生成については、MIDI形式の楽曲に対して自動的に譜面データを生成する手法が香川らにより提案されている[7]。

香川らは音楽には重要なフレーズが繰り返し出現するという前提をもとに、楽曲をメロディ、ハーモニー、リズムの3層に分解し離散的に構造解析することで、MIDI形式の楽曲データから重要と思われる音をキー音として抽出するFBCKE法の提案・実装を行った。FBCKEとはfrequency-based key component extractionの略である。この手法は木構造のデータ構造体であり、文字列検索などの用途で用いられるsuffix tree[8]に基づいている。

キー音を用いたリズムアクションゲームの譜面データを作成する際には、どの楽曲中の音をキー音として抽出するか、抽出したキー音を譜面上にどう配置するか2点が困難であることについても述べられている。FBCKE法はこのうちキー音となる音の候補を抽出することについてのみ扱い、譜面の難易度に応じて実際にキー音を選択することや、ノーツを譜面上へどう配置するかについては今後の課題としている。

キー音を用いたリズムアクションゲームである「beatmania」シリーズのPC用クローンゲームである「BMS」のデータを使用し[9]、MIDI形式の楽曲データを入力する入力モジュール、キー音の抽出と譜面上への配置を行う譜面構成モジュール、BMSの譜面ファイルを出力する出力モジュールから成る自動譜面生成フレームワークが設計・実装されている。このうち、譜面構成モジュールはシステムの改善に従って組み替えが可能である。

ランダムに抽出したキー音候補と比較して、FBCKE法により生成したキー音候補の方が楽曲から重要と思われる音を抽出できているということが主観評価実験により示されている。しかし、実際のリズムアクションゲームに用いられる楽曲は多くの場合WAV形式などの波形データであるため、MIDIデータとは異なり楽譜に相当する情報を得ることが難しい。そのため、FBCKE法を実際のリズムアクションゲームの譜面データ自動生成に活用するにあたって、何らかの手法により波形データを楽譜データの形に変換しない限り、構造解析を行うことが困難であると考えられる。また、suffix treeを用いてフレーズの長さや出現頻度から楽曲の重要音を推定しているが、これらの特徴を用いて抽出した音が実際のゲーム上でキー音としてふさわしいかについては検証されていない。

3. 使用データ

学習に用いるデータには、香川らと同様にキー音を用いたリズムアクションゲームである「BMS」のデータを使用した[9]。BMS データは楽曲から切り出されたキー音候補である多数のサウンドファイルとテキストデータである譜面ファイルから成り、個々のサウンドファイルを再生するタイミングを譜面ファイル内で定義・再生を行うことで、サウンドファイルの集合体として楽曲を構成している。BMS データのディレクトリ構造を図6に、BMS のプレイ画面を図7に示す。

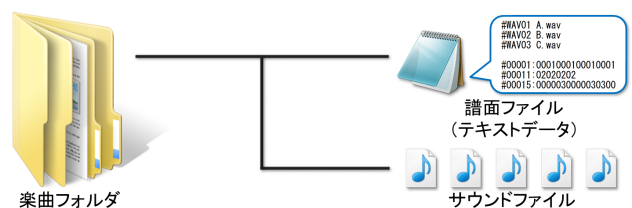


図6. BMSデータのディレクトリ構造

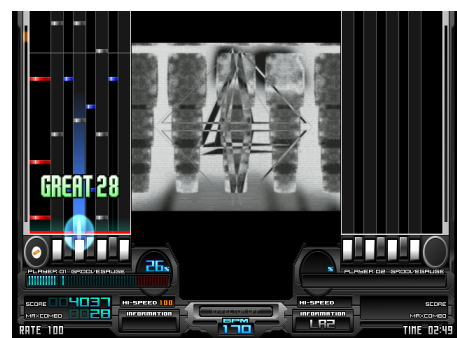


図7. BMSのプレイ画面

プレイ画面中央のムービー表示部分の両脇には7つの鍵盤と丸いスクラッチのレーンが存在し、左側が1プレイヤー側、右側が2プレイヤー側である。両プレイヤー側において、最も左にある白い鍵盤を1キー、その隣の黒い鍵盤を2キー...として扱い、最も右にある白い鍵盤が7キーである。配置されたオブジェクトは奇数キーであれば白色に、偶数キーであれば青色に、スクラッチであれば赤色に表示される。演奏に成功すれば鍵盤下部のゲージが増加、失敗すれば低下し、楽曲終了時にゲージが80%を超えていればゲームクリアとするのが最も基本的なゲームルールである。

BMSデータを学習データとして使用する利点として、商業用のゲームとは異なり楽曲内で使用されているサウンドデータが抽出可能であることや、ゲームデータに相当する譜面ファイルがテキストデータであるため、解析を容易に行うことができることが挙げられる。

譜面ファイルは**bms**、**bme**、**bml**などの拡張子を持っており、一つの楽曲フォルダには多くの場合難易度が異なる複数の譜面ファイルが含まれている。一つの楽曲フォルダに含まれる譜面ファイルは基本的に同じ楽曲を構成するため、異なる譜面ファイルであっても同じサウンドファイルの再生を行っている。

譜面ファイルには特定のサウンドファイルが再生されるタイミングだけでなく、プレイヤーの操作がなければ再生されないキー音、もしくは操作がなくても再生されるBGMのいずれかを選択する「チャンネル」と呼ばれる記述がなされている[10]。そのため、譜面ファイルの解析を行うことで「X分X秒の時点で再生されるXXXX.wavはキー音として選択されている」といった、個々のオブジェクトについての情報が時系列に並んだデータを得ることができる。

各時点に配置されたサウンドファイルの定義を「オブジェクト」、キー音であるオブジェクト、つまりプレイヤーが演奏するオブジェクトを「ノート」、個々の譜面におけるノーツの総数を「トータルノーツ」と呼ぶ。

全てのノートは7つの鍵盤とスクラッチのいずれかのレーンに割り当てられており、プレイヤーが譜面として認識するのはこれらのレーンに配置されたノートの集合である。そのため、譜面の自動生成を実現するためにはキー音であるか否かの推定だけではなく、ノートをどのレーンに配置すべきかについても推定を行う必要がある。

特定のサウンドファイルは一つの譜面ファイル内で何度でもオブジェクトとして再生することができ、オブジェクトごとにキー音とBGMの選択が行われている。つまり、同じサウンドファイルを再生しているオブジェクトであっても、キー音として選択されるものもBGMとして選択されるものも存在する。

また、複数の難易度の譜面を作成するにあたって難易度別にキー音の抽出を行う必要があるが、低難易度の譜面でキー音として選択されているオブジェクトが高難易度の譜面でもキー音として選択されるとは限らない。さらに、楽曲の作者と譜面の作者が異なることも多く、譜面作成の自由度を高めるためにほとんどの楽曲では目立たない音であっても楽曲から切り出され、BGMオブジェクトとして譜面ファイル上に配置されている。

本研究では、高品質であるBMSデータが85曲分選定された「GENOSIDE -BMS StarterPackage-」を使用した[11]。効率的に譜面の難易度によるキー音抽出傾向を学習するため、各楽曲に含まれる最もトータルノーツが少なく難易度が低いと判断できる「最低難易度の譜面ファイル」と、最もトータルノーツが多く難易度が高いと判断できる「最高難易度の譜面ファイル」を分析対象として採用した。楽曲によって難易度のばらつきが大きいと、将来的には楽曲の長さやトータルノーツの値を特徴量のひとつとし、全ての譜面ファイルから学習を行う予定である。

機械学習に用いる特徴量は、既存の譜面ファイルを解析することで得られる各オブジェクトについての時系列データと、各オブジェクトに対応するサウンドファイルから得られる音響特徴量の掛け合わせから成る。時系列データは図5における第1過程の出力に相当し、音響情報自体は含まない。サウンドファイルから抽出した音響特徴量については4章、時系列データと音響特徴量の組み合わせについては5章に詳細を記述する。

4. 音響特徴量の抽出

各楽曲フォルダに含まれるサウンドファイルからの音響特徴量抽出には、音声・音楽情報処理向けの特徴量抽出ツールであるopenSMILE[12]を使用した。

BMSデータに含まれるサウンドファイルは、WAVファイルの場合ファイルサイズ削減のため量子化ビット数やサンプリングレート、チャンネル数を意図的に低くしている場合や、Ogg Vorbisファイル、MP3ファイルなどの圧縮音声フォーマットが用いられている場合がある。openSMILEにより特徴量を行う際にはサウンドファイルの形式を統一する必要があるため、特徴量抽出の前処理として全てのサウンドファイルについて16bit/44.1kHz/2chのWAVファイルに変換を行った。

このとき、変換前のWAVファイルのサンプリングレートが44.1kHzよりも低い場合はサンプリングレートの1/2の値より大きい高周波数の成分が欠損しており、量子化ビット数が8bitである場合はダイナミックレンジが狭く顕著な量子化ノイズが見られるなど、音質の劣化が生じている。

前処理を実行する前のサウンドファイル49,688個について、サンプリングレートの内訳を表1に示す。また、WAVファイルにおける量子化ビット数の内訳を表2に、Ogg VorbisファイルとMP3ファイルにおけるビットレートの内訳を表3に示す。

表1. サウンドファイルのサンプリングレート

ファイル形式	サンプリングレート	ファイル数(個)
WAV	11025 Hz	35
	22050 Hz	1,044
	44100 Hz	4,384
Ogg Vorbis	22050 Hz	457
	24000 Hz	74
	32000 Hz	117
	44100 Hz	43,560
MP3	48000 Hz	2
	44100 Hz	15

表2. WAVファイルの量子化ビット数

ファイル形式	量子化ビット数	ファイル数 (個)
WAV	8 bit	2,581
	16 bit	2,882

表3. Ogg Vorbis、MP3ファイルのビットレート

ファイル形式	ビットレート	ファイル数 (個)
Ogg Vorbis	32 kbps	648
	56~64 kbps	201
	77~90 kbps	38
	105 kbps	495
	110~112 kbps	431
	128 kbps	14,606
	160~180 kbps	1,421
	192 kbps	20,752
	200~240 kbps	1,010
	320 kbps	2,580
MP3	500 kbps	2,028
	192 kbps	15

チャンネル数については、サウンドファイル 49,688 個中 275 個がモノラル、49,413 個がステレオのデータであった。なお、音響特徴量抽出はopenSMILEの内部で2つのチャンネルを合成し、モノラル化した状態で行った。

openSMILE による特徴量抽出はまず 25ms のフレーム単位でフレーム特徴量を抽出し、それぞれのフレーム特徴量に対して全フレーム分の特徴を表す全体特徴量を抽出することで行った。使用したフレーム特徴量を表 4 に、全体特徴量を表 5 に示す。音響特徴量の総数は、フレーム特徴量数と全体特徴量数を乗算した数である。本研究では、キー音として適する音は楽曲中で聴感上目立つ音であるという仮定から、音量やスペクトルの変化を学習することが効果的であると考え、文献[4]で抽出した 559 種類の特徴量から、予備実験における学習結果の attribute weights を参考に、60 種類の選択を行った。

表4. 使用したフレーム特徴量

名称	説明
LogEnergy	音量 (対数)
HNR	調波成分と非調波成分の比率
F0	基本周波数
SpectralFlux	スペクトルの変化量
SpectralVariance	スペクトルの分散
SpectralSkewness	スペクトルの歪度

表5. 使用した全体特徴量

名称	説明
Max	最大値
Min	最小値
Range	最大値と最小値の差
LinRegC1	線形回帰係数 (傾き)
LinRegC2	線形回帰係数 (オフセット)
LinRegErrQ	線形回帰二乗誤差
Reg-centroid	回帰直線の重心
Skewness	歪度
Kurtosis	尖度
MeanPeakDist	ピーク間の距離の平均

5. 時系列データの機械学習

譜面ファイルのテキスト解析を行い、個々のオブジェクトが持つ情報を再生時点順に並べた時系列データを得た。抽出した時系列データの例を図 8 に示す。

[u' crashr. wav', '16', 0, 0.625, 158.0, 0.949]
[u' closed606. wav', '01', 1, 0.0, 158.0, 1.51]
 wavファイル名 ch 位置 BPM 秒

図8. 時系列データの例

時系列データはテキスト 1 行が 1 つのオブジェクトを表すカンマ区切りのデータとなっており、角括弧の中に再生するサウンドファイルの名称、チャンネル記述、再生小節、4/4 拍子を基準として 1.0 とした際の小節内での位置、再生時の BPM、秒単位の再生時点の順に記述されている。このうち、秒単位の再生時点は再生小節、小節内での位置、再生時の BPM から算出したものである。

チャンネルの記述は値が'01'であればそのオブジェクトは BGM であることを意味し、値が 10 台または 50 台であれば図 6 における 1 プレイヤー側のノーツである。また、同様に 20 台または 60 代であれば 2 プレイヤー側のノーツである。そのため、チャンネル記述が'01'の場合は BGM(false)とし、それ以外の値であればキー音(true)として分類を行う。時系列データのうち、チャンネル記述以外は図 5 の第 1 過程で得られる情報である。

この時系列データに基づき、本研究では個々のオブジェクトに対して同時または直前の時点に存在する直近 10 個のオブジェクトの探索を行い、オブジェクト自身と直近のオブジェクトについてサウンドファイルの名称、チャンネル記述、秒単位の再生時点を参照した。なお、楽曲開始から 10 番目までのオブジェクトについては、直近 10 個分のオブジェクトの時系列データを得ることができないため、学習には使用しない。

サウンドファイルの名称からはopenSMILEにより抽出した音響特徴量を、チャンネル記述からはキー音であるか否か(true, false)を、秒単位の再生時点からは現在のオブジェクトの時点との時間差の情報を、機械学習に用いる特徴量として得た。

openSMILE を用いて抽出した音響特徴量と、譜面ファイルを解析して得た時系列データから機械学習を行うための学習データを作成するにあたり、ブロック図を図 9 に示す。

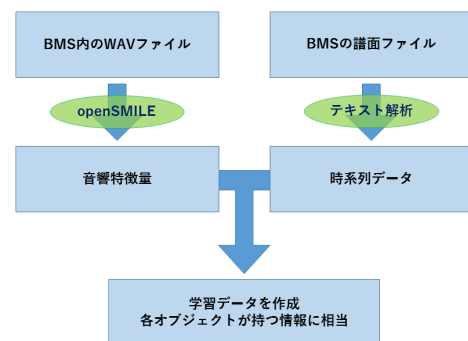


図9. 学習データ作成のフロー

フレーム特徴量の総数を L 、全体特徴量の総数を F 、探索する直近のオブジェクトの数を P とすると、実際に学習に用いる特徴量の次元 D は数式(1)で表すことができる。式中の定数'2'は現在のオブジェクトの時点との時間差と、キー音であるか否か($true, false$)の2つの特徴量を表している。また、直近のオブジェクトについて以後「先行音」と呼び、 P は先行音の数である。個々のオブジェクトがこれらの特徴量を持っており、それらのデータが再生時系列の順に並んだものが学習データである。

$$D = P * (L * F + 2) + L * F \quad (1)$$

本研究では $L=6$ 、 $F=10$ 、 $P=10$ であるため、計 $D=680$ 次元の特徴量を使用した。全楽曲についてこれらの特徴量を機械学習ソフトウェアである Weka[13]に入力し、現在のオブジェクトがキー音として選択されるか否かについて推定を行った。学習アルゴリズムには、SVM(サポートベクターマシン)によりクラス分類を行う手法の中でも特に高速な処理を行うことが可能な SMO(逐次最小問題最適化法)[14]を用いた。

最低難易度の譜面ファイルから得たオブジェクトの学習データと最高難易度の譜面ファイルから得たオブジェクトの学習データを入力として、それぞれの難易度について、楽曲ごとに学習を行い10分割のクロスバリデーションで評価を行う「楽曲別の推定」と、全ての楽曲に含まれるオブジェクトの時系列データを混合して50等分し、そのうちの1/50のデータから学習を行い、残り49/50のデータを評価する「楽曲によらない推定」の2通りの方法で、現在のオブジェクトがキー音となるか否かについて機械学習を行った。「楽曲別の推定」は、途中まで手動でキー音とBGMの分類を行うことで、残りのオブジェクトは自動で分類できる譜面作成を支援するシステムを想定しており、本研究においては予備実験の位置づけにあたる。なお、「楽曲によらない推定」においても学習データと評価データのそれぞれには一部同じ楽曲のオブジェクトが含まれているため、異なる楽曲のみから学習を行うことについては、今後の課題とする。

6. 推定結果

「楽曲別の推定」の結果を6.1節に、全ての楽曲のデータを用いた「楽曲によらない推定」の結果を6.2節に示す。なお、キー音でないBGMのオブジェクトは $false$ 、キー音であるオブジェクトは $true$ と推定した場合に、正しく分類がなされているとして評価する。

6.1 楽曲別の推定

楽曲ごとに学習と評価を行った場合の結果を表6および表7に示す。表6は最低難易度の譜面ファイルを学習・評価に用いた場合、表7は最高難易度の譜面ファイルを用いた場合の結果である。最低難易度では譜面ファイルから抽出したオブジェクト276,156個、最高難易度ではオブジェクト286,450個を評価した。表中の正解率は $true, false$ それぞれのみから算出したものである。

表6. 楽曲別の推定結果 (最低難易度)

正解/予測	false	true	再現率
BGM (false)	223,603	7,599	96.71%
キー音 (true)	9,850	35,104	78.09%
適合率	95.78%	82.20%	

表7. 楽曲別の推定結果 (最高難易度)

正解/予測	false	true	再現率
BGM (false)	131,847	11,705	91.85%
キー音 (true)	11,234	131,664	92.14%
適合率	92.15%	91.84%	

最低難易度の譜面ファイルから得た時系列データを用いた場合では、現在のオブジェクトがキー音であるか否かについて、BGMであるオブジェクトを $false$ と正しく推定した確率と、キー音であるオブジェクトを $true$ と正しく推定した確率の平均(再現率の平均)で、87.40%の精度で推定することができた。また同様に、 $false$ と推定したオブジェクトのうち実際にBGMであった確率と、 $true$ と推定したオブジェクトのうち実際にキー音であった確率の平均(適合率の平均)で、88.99%の精度で推定することができた。

また、最高難易度の譜面ファイルから得た時系列データを用いた場合では、再現率の平均で91.99%の精度で推定することができ、適合率の平均でも91.99%の精度で推定することができた。

楽曲の傾向によってキー音の選択基準はある程度差があると考えられるため、楽曲の音楽的な分類や楽曲別の推定における正解率をもとに楽曲のクラスタリングを行うことが、より高い精度を得るために効果的である可能性がある。さらに、先行音だけでなく時間的に後方のオブジェクトからも音響特徴量を学習することなどにより、より深く楽曲構造を学習することについても今後の課題としている。

6.2 楽曲によらない推定

次に、全楽曲に含まれるオブジェクトの時系列データを混合し楽曲によらない学習と評価を行った結果を、表8～表10に示す。表8は最低難易度の譜面ファイルを学習・評価に用いた場合であり、表9は最高難易度の譜面ファイルを学習・評価に用いた場合の結果である。最低難易度では譜面ファイルから抽出したオブジェクト13,531,644個、最高難易度では14,036,050個を評価した。

表8. 楽曲によらない推定結果 (最低難易度：重み付けなし)

正解/予測	false	true	再現率
BGM (false)	11,006,116	322,782	97.15%
キー音 (true)	1,872,407	330,339	15.00%
適合率	85.46%	50.58%	

表9. 楽曲によらない推定結果 (最高難易度)

正解/予測	false	true	再現率
BGM (false)	5,303,878	1,730,170	75.40%
キー音 (true)	1,844,564	5,157,438	73.66%
適合率	74.20%	74.88%	

表8に示すように、最低難易度の譜面ファイルから得た時系列データを用いた場合では、現在のオブジェクトがキー音となるか否かについて、BGMであるオブジェクトをfalseと正しく推定した確率と、キー音であるオブジェクトをtrueと正しく推定した確率の平均(再現率の平均)で、56.07%の精度で推定することができた。

また同様に、falseと推定したオブジェクトのうち実際にBGMであった確率と、trueと推定したオブジェクトのうち実際にキー音であった確率の平均(適合率の平均)で、68.02%の精度で推定することができた。また表9に示すように、最高難易度の譜面ファイルから得た時系列データを用いた場合では、再現率の平均で74.53%の精度で推定することができ、適合率の平均では74.54%の精度で推定することができた。

上記の2種類の実験において、最高難易度ではキー音でないオブジェクトの数とキー音であるオブジェクトの数がほぼ等しく、trueと判定されたオブジェクトとfalseと判定されたオブジェクトの総数もほぼ等しい結果であった。しかし、最低難易度ではキー音でないBGMのオブジェクト数と比べてキー音であるオブジェクト数が大幅に少ないことから、キー音であるオブジェクトについてもfalseと判定されやすく、特にキー音における再現率が極端に低下するという問題が見られた。これは、キー音として配置するオブジェクトを減らし、演奏を要求する密度を下げることで、譜面の難易度を低減させているからである。

そこで、キー音であるオブジェクトをfalseと判定した場合に実際のデータにおけるtrue : falseの比である5.14倍の重みを付けて評価するよう学習アルゴリズムを修正し、最低難易度の評価実験を行ったところ、表10のような結果が得られた。推定精度は再現率の平均で68.78%、適合率の平均でも61.51%となった。適合率は重み付けを行わない場合よりも低下しているが、これは重み付けを行うことによりBGMであるオブジェクトに対してもtrueと誤推定されるケースが増加し、キー音の適合率が18.97%低下したためである。なお、再現率に関しては大幅に改善されている。

表10. 楽曲によらない推定結果 (最低難易度 : 重み付けあり)

正解/予測	false	true	再現率
BGM (false)	8,239,618	3,089,280	72.73%
キー音 (true)	774,652	1,428,094	64.83%
適合率	91.41%	31.61%	

本実験では楽曲ごとの最高難易度の譜面、最低難易度の譜面をそれぞれ混合したデータを用いているが、3章でも述べた通り楽曲によって難易度のばらつきが大きいので、ある楽曲の最高難易度の譜面よりもある楽曲の最低難易度の譜面の方が難しい、といったケースも存在する。そのため、楽曲の長さやトータルノーツの値から個々の譜面ファイルの客観的な難易度を得て、キー音とBGMのオブジェクト数の比に大きな差がある譜面を混合することなく学習を行うことが今後の課題である。

6.3 時系列データの有効性の検証

時系列データを用いて学習することの有効性を検証するため、先行音の特徴量からキー音であるか否か(true, false)を除いた状態での実験を行った。また、学習に用いる先行音の個数を段階的に減らした状態での実験を行った。これら2種類の実験は、「楽曲によらない推定」のみ実施した。

最低難易度の譜面ファイルから得た時系列データを用いたうえで5.14倍の重み付けを行った場合に、先行音についてキー音であるか否か(true, false)の特徴量を除外して学習を行ったところ、表11の結果が得られた。キー音とBGMの再現率の平均は66.56%、適合率の平均は59.90%であった。

同様に、最高難易度の譜面ファイルから得た時系列データを用いた場合に、先行音についてキー音であるか否か(true, false)の特徴量を除外して学習を行ったところ、表12の結果が得られた。キー音とBGMの再現率の平均は67.43%、適合率の平均も67.43%であった。両難易度で再現率と適合率は2%~7%低下しており、先行音がキー音であるか否かの情報は現在のオブジェクトがキー音であるか推定するにあたって有用であるといえる。

表11. 先行音のキー音ラベル無し (最低難易度)

正解/予測	false	true	再現率
BGM (false)	7,961,768	3,367,130	70.28%
キー音 (true)	818,583	1,384,163	62.84%
適合率	90.68%	29.13%	

表12. 先行音のキー音ラベル無し (最高難易度)

正解/予測	false	true	再現率
BGM (false)	4,763,339	2,270,709	67.72%
キー音 (true)	2,301,043	4,700,959	67.14%
適合率	67.43%	67.43%	

次に、学習に用いる先行音の数を段階的に減らした状態での実験を行った。最低難易度の譜面ファイルから得た時系列データを用いた場合の再現率の変化を図10に、適合率の変化を図11に示す。グラフの縦軸は1.0を最大とした場合の正解率であり、グラフの横軸は用いた先行音の数を表す。横軸が0であれば現在のオブジェクトの音響特徴量のみから学習を行った結果であり、10であれば6.2節と同様に10個の先行音の情報を学習を行った結果である。青いデータはBGM(false)の精度、赤いデータはキー音(true)の精度、灰色のデータはBGM(false)の精度とキー音(true)の精度の単純平均である。

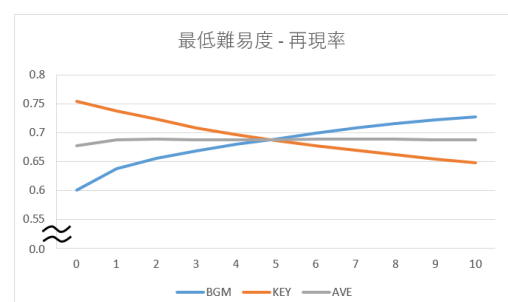


図10. 時系列を削減した場合の再現率 (最低難易度)

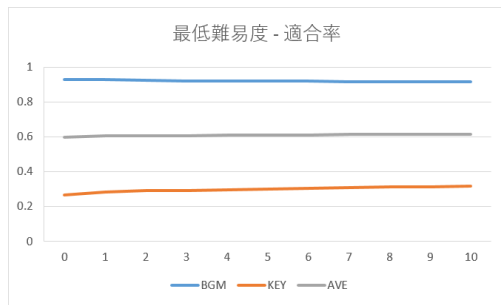


図11. 時系列を削減した場合の適合率（最低難易度）

再現率の平均は飽和しており、適合率の平均は僅かに上昇している。用いる先行音の数が多いほどキー音の再現率に関しては低下しているが、これは時系列が増えることによって推定が難しいデータをBGMと推定しやすくなったことが原因である。キー音の適合率は26%から31%に増加している、つまりキー音と予測したオブジェクトのうち実際にキー音であるオブジェクトの数は増えているため、より長い時系列から学習を行うことによる性能向上は有効であると考えられる。

同様に、最低難易度の譜面ファイルから得た時系列データを用いた場合の再現率の変化を図12に、適合率の変化を図13に示す。最高難易度の場合は長期的な時系列を用いるほどまんべんなく精度が向上している。また、現在のオブジェクトの音響特徴量のみを用いて学習を行う場合と、現在のオブジェクトの音響特徴量に加えて最も現在のオブジェクトと再生時点に近い先行音1個の情報も学習に用いる場合に特に顕著な差が現れていることがわかる。いずれの難易度であっても、時系列の長期化が性能向上に有効であることが確かめられた。

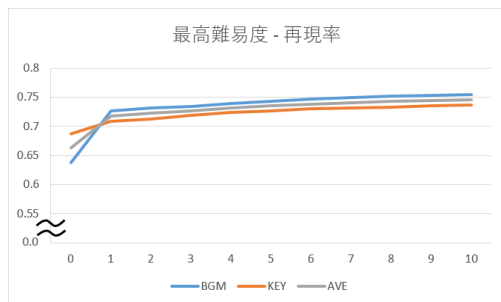


図12. 時系列を削減した場合の再現率（最高難易度）

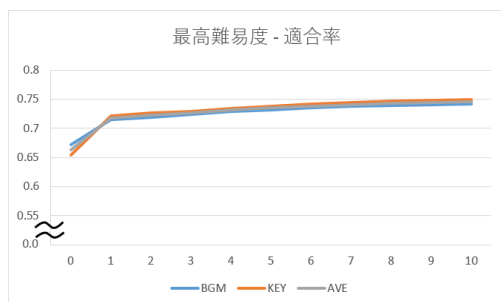


図13. 時系列を削減した場合の適合率（最高難易度）

7. まとめ

キー音を用いたリズムアクションゲームにおいて、楽曲に含まれる音が切り出されたものであるサウンドファイルが持つ音響特徴量と、譜面ファイルから得られる時系列データから機械学習を行うことで、個々のオブジェクトがキー音であるか否かについて一定の精度で推定することができた。

楽曲別の推定については十分に高い精度が得られたため、未知の楽曲の譜面データ作成を一部だけ人手で行うことで、残りのオブジェクトについては人手により作成した部分から特徴を学習し、キー音であるか否かの判別を自動化することが可能であるといえる。

現時点では、対象のオブジェクトから10個分過去のオブジェクトまで学習に用いているが、同時発音数が多い場合や短時間にオブジェクトが集中しているような場合には時間差のあるデータを得ることができないという問題がある。そこで、今後はより多い直近オブジェクトの情報を取得して、さらに長期的な時系列情報から学習を行うことでも、精度の向上が見込めると考えられる。

これに加えて、1つの譜面ファイルからだけでも数1000個以上のオブジェクトについての情報が得られることから、深層学習によるモデルの適用が有効であると考えられる。Donahueらの手法を応用しリカレントニューラルネットワーク(RNN)などによる時系列の学習が可能なモデルを導入することで、より高い推定精度を実現することができると考えられる。

さらに、BGMがキー音かにかかわらず直前の10オブジェクトを学習に用いるのではなく、直前のキー音であるオブジェクトのみから得た特徴のみを学習に用いることで、次元の数を低減しつつキー音の前後関係を適切に学習することができる可能性がある。

本研究で音響特徴量として使用している音量やスペクトルの変化だけでなく、発音タイミングに相当するオンセットなどの特徴量の使用を検討することや、主成分分析や変数減少法(Backward Stepwise Selection)を行い、使用する音響特徴量の次元を削減することについては、今後の課題である。

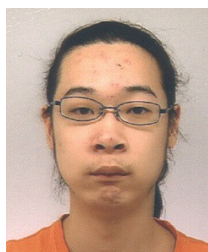
譜面ファイルの自動生成を行うためには、キー音として選択されたオブジェクトをどのレーンに相当するチャンネルに割り当てるかについても自動化する必要があるため、将来的にはこの過程においても機械学習による推定を行い、実際に楽曲データから指定する難易度に応じた譜面ファイルを自動生成するシステムの制作を目指す。

また、楽曲から個々の音を切り出す手間が大きいことがキー音を用いたリズムアクションゲームの開発が難しい理由の一つであると考えられるため、楽曲のトラックごとに独立した音源がなくても音の切り出しが自動化できるように、チャンネル数よりも音源数の方が多い楽曲データから音源分離が可能なブラインド音源分離技術を応用し、図5における第1過程に適用することが期待される。

参考文献

- [1] Karen Collins, Game Sound: An Introduction to the History, Theory, and Practice of Video Game Music and Sound Design, The MIT Press, United States of America, pp. 74-75, 2008.
- [2] 橋本祐輔, 橋田光代, 片寄晴弘, 音楽音響信号を対象とした指揮演奏システムの開発, 情報処理学会研究報告, SIG-EC-12, pp. 43-50, 2009.
- [3] Martin Pichlmair, Fares Kayali, Levels of Sound: On the Principles of Interactivity in Music Video Games., Proceedings of the Digital Games Research Association 2007 Conference Situated play, Tokyo, Japan, Vol. 4, pp. 424-430, 2007.
- [4] 福永大輝, 大淵康成, リズムアクションゲームにおけるキー音の音響的特徴分析, NICOGRAPH 2017, pp. 134-135, 2017.
- [5] Chris Donahue, Zachary C. Lipton, Julian McAuley, Dance Dance Convolution, ICLR 2017 Workshop, 2017.
- [6] Sepp Hochreiter and Jürgen Schmidhuber. Long shortterm memory. Neural computation, Vol. 9, No. 8, pp. 1735-1780, 1997.
- [7] 香川俊宗, 手塚宏史, 稲葉真理, 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-, エンタテインメントコンピューティングシンポジウム, EC2015, pp. 326-333, 2015.
- [8] Peter Weiner, Linear pattern matching algorithms, SWAT '73 Proceedings of the 14th Annual Symposium on Switching and Automata Theory, pp. 1-11, 1973.
- [9] Urao Yane, BM98Data_format_specification, <http://bm98.yaneu.com/bm98/bmsformat.html>, 1998. 参照:2018-5-16
- [10] hitkey, BMS command memo (JP), <http://hitkey.nekokan.dyndns.info/cmdsJP.htm>, 2014. 参照:2018-5-16
- [11] lobsak, GENOSIDE -BMS StarterPackage-, <http://nekokan.dyndns.info/~lobsak/genoside/>, 2012. 参照:2018-5-16
- [12] Florian Eyben, Felix Weninger, Florian Groß, Björn Schuller, Recent Developments in openSMILE, the Munich Open-Source Multimedia Feature Extractor, Proceedings of ACM Multimedia 2013, Barcelona, Spain, ISBN 978-1-4503-2404-5, pp. 835-838, 2013.
- [13] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, Ian H. Witten, The WEKA Data Mining Software: An Update, SIGKDD Explorations, Vol. 11, No. 1, pp. 10-18, 2009.
- [14] J.C. Platt, Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines, Microsoft Research Technical Report MSR-TR-98-14, 1998.

福永 大輝



1995年生。2018年東京工科大学大学メディア学部メディア学科卒業。2018年東京工科大学大学院バイオ・情報メディア研究科メディアサイエンス専攻博士前期課程入学。音声、音響、音楽分野のメディアコンテンツへの応用に興味を持つ。音楽ゲームの研究に従事。

越智 景子



2011年東京大学情報理工学系研究科電子情報学専攻博士課程修了。国立障害者リハビリテーションセンター研究所流動研究員(2011-2014)。同客員研究員(2014-2016)。国立情報学研究所特任研究員(2015-2016)。同特任助教(2016-2017)。2017年より東京工科大学メディア学部助教。博士(情報理工学)。音声合成、韻律、音声分析、音声インターフェース、言語訓練の研究に従事。

大淵 康成



1966年生。1990年東京大学大学院理学系研究科物理学専攻修士課程修了。1992年同博士課程中退。1992年より2015年まで(株)日立製作所中央研究所および基礎研究所勤務。その間、Carnegie Mellon University客員研究員(2002-2003)、早稲田大学客員研究員(2005-2010)、クラリオン(株)(2013-2015)。2015年より東京工科大学メディア学部教授。博士(情報理工学)。