

時系列深層学習に基づく難易度間関係モデルを用いた ダンスゲーム譜面難易度の自動調整

辻野 雄大^{1,a)} 山西 良典^{2,b)} 西原 陽子² 福本 淳一²

受付日 2018年1月12日, 採録日 2018年9月7日

概要: ダンスゲームは、幅広いユーザから親しまれている代表的な音楽ゲームの1つである。幅広いユーザがダンスゲームを楽しめる環境を用意するためには、初級者でも容易に遊ぶことができる低難易度の譜面を充実させる必要がある。しかし、低難易度の譜面を作成するためには、楽曲の特徴をとらえつつ容易なリズムに調整するという、高難易度譜面の作成にはない課題が存在する。本稿では、ダンスゲームには同じ曲に対して難易度が異なる複数の譜面が存在することに着目し、難易度が高い譜面から得られる音楽の特徴を入力、難易度が低い譜面を出力とする時系列深層学習モデルを構築した。学習させた提案モデルに高難易度のダンス譜面を入力し、低難易度の譜面において指示符を配置すべき発音タイミングを推定させることで、難易度の自動調整を実現した。性能評価の結果、時刻決定タスクにおいて提案手法は0.693の F 値が確認され、既存手法の F 値をおおよそ1.8倍上回った。向き選択タスクについて指示符の2-gram 出現頻度を集計したところ、提案手法の生成譜面とデータセット内の低難易度譜面との相関係数が0.972となり、人手で作成された低難易度のダンス譜面の特性をとらえた譜面を自動生成可能であることが確認された。

キーワード: 音楽情報処理, 譜面難易度調整, procedural content generation

Adapting Difficulty of Dance Chart on Video Game Using Relation Model Among Difficulty Levels Based on Time-series Deep Learning

YUDAI TSUJINO^{1,a)} RYOSUKE YAMANISHI^{2,b)} YOKO NISHIHARA² JUNICHI FUKUMOTO²

Received: January 12, 2018, Accepted: September 7, 2018

Abstract: Dance video game is one of the typical popular games that has a wide range of fans. In order to make music video game enjoyable to many people, it is necessary to provide many lower-difficulty charts for the beginners. To make lower-difficulty charts, the easy rhythm without losing the point of the song should be covered. In dance video game, each song has multiple charts for varied difficulty levels. In this paper, we propose time-series deep learning model that learns the relation between the lower and higher difficulty charts for the same song. By inputting the higher-difficulty chart into the trained model, it estimates rhythms for lower-difficulty charts; our proposed method adapts the difficulty of the chart. Through the experiments for step placement task, the proposed method achieves an F -score of 0.693 which is about 1.8 times higher than the existing method. For step selection task, the value for the correlation coefficient between generated charts and handmade lowest-difficulty charts is 0.972 in 2-gram frequency; it was confirmed that the proposed method generated the lower-difficulty charts reflecting the characteristic of handmade lower-difficulty charts.

Keywords: music informatics retrieval, adaptation of score level, procedural content generation

1. はじめに

ビデオゲームは、近代のエンタテインメント市場を牽引してきた。今日のビデオゲーム市場のなかで、大きなシェアを占めているジャンルに音楽ゲームが存在する。音楽ゲームの多くでは、再生される音楽と画面に表示される指示符（ゲーム上では、「譜面」と呼ばれる）に沿って、プ

¹ 立命館大学大学院情報理工学研究科
Graduate School of Information Science and Engineering,
Ritsumeikan University, Kusatsu, Shiga 525–8577, Japan
² 立命館大学情報理工学部
College of Information Science and Engineering, Ritsumei-
kan University, Kusatsu, Shiga 525–8577, Japan
^{a)} is0221rs@ed.ritsumeikan.ac.jp
^{b)} ryama@fc.ritsumeikan.ac.jp

レイヤが何らかの動作を行い、その動作のリズムに対する成否を競う。音楽ゲームのプレイヤーの中には何年も継続して遊ぶ上級者もいれば、1度だけ遊ぶような初心者もいる。幅広いプレイヤー層のニーズに応えるため、多くの音楽ゲームでは1つの楽曲に対して難易度の異なる複数の譜面を用意することで、初心者から上級者まで多くのプレイヤーが同じ曲を楽しめるよう配慮されている。

音楽ゲームのサブジャンルとして、音楽に合わせてプレイヤーが手足を動かすことで遊ぶ「ダンスゲーム」が存在する。代表例として、プレイヤーが上下左右の4方向の矢印が描かれたパネルを足で踏むことによって操作する *Dance Dance Revolution*®*1 (DDR) があげられる。DDR では、プレイヤーに対する動作の指示は画面上に現れる矢印によってなされ、1回の動作を要求する矢印を「ステップ」と呼ぶ。ダンスゲームは全身を動かしながら遊ぶため、ダイエットやフィットネスとしての効果も注目されている。一例として、アメリカのウェストバージニア州では、肥満児童問題への対応策の一環として学校授業にDDRを取り入れた事例が報告されている[1]。日本では、2012年より文部科学省の中学校学習指導要領においてダンスが必修化されたが、ダンス未経験の教員も多く存在し、指導者の確保は困難であると考えられる*2。日本国内でダンスゲームを教育に取り入れた事例は報告されていないが、任意の楽曲・広い難易度に対応した音楽ゲームをダンス教育インタフェースとして用いることで、学童に対して親しみやすく円滑なダンス授業の実施が期待される。

音楽ゲームの譜面は、ゲームの製作者が楽曲を聴取して、手作業で作成することが一般的である。このため、ある音楽ゲームのプレイヤーが好みの楽曲であっても、既存の譜面がなければ遊ぶことができない。また、ゲームの製作者にとっても、膨大な数の楽曲に対して譜面を作成していくことは、けっして容易な作業ではない。一方、情報処理においては、波形から計算機に音楽を認識させる音楽情報処理や、大量のデータを学習し未知のデータに対して予測を行う機械学習といった分野が存在し、これらに応用することで音楽ゲームの譜面を自動で生成可能になると考えられる。Donahueらは、音楽情報処理と機械学習の技術を用いて、ダンスゲーム *Stepmania**3を題材に、楽曲と譜面の関係を学習することで、入力した楽曲から譜面を生成する手法：*Dance Dance Convolution*を提案している[2]。StepmaniaはDDRを模したオープンソースの音楽ゲームである。ユーザが譜面を自作する機能が備わっており、音

楽データとそれに紐付いた譜面データを用意することで、任意の音楽でダンスゲーム用の譜面を製作することができる。Donahueらの手法は、難易度が高い譜面の学習・出力において一定の性能を示した一方で、難易度が低い譜面はうまく学習できず、良質な譜面が出力できないということが述べられている。高難易度の譜面は、楽曲中の目立つ発音すべてに沿って指示符を配置することで製作可能である。しかしながら、低難易度の譜面を製作するためには、楽曲の特徴をとらえつつ容易にプレイ可能な、少数の発音タイミングを選出する必要がある。難易度の低い譜面は、音楽ゲームの幅広いユーザのニーズに対応するために必要となる。特に、音楽ゲームをダンス能力育成支援へ応用することを考えた場合、ユーザそれぞれの習熟度に合わせて多様な難易度の譜面を作ることは必須となる。

本研究では、音楽ゲームでは1曲に対して難易度の異なる複数の譜面が存在することに着目し、難易度間の関係を学習する時系列深層学習モデル：*Dance Dance Adaptation*を提案する。提案モデルはDonahueらの学習モデルをベースとし、難易度が高い譜面を入力、低い譜面を出力とすることで、譜面難易度の自動調整を実現する。

なお、本稿では小節を n 等分（ただし $n \geq 4$ ）することで表現できる時刻集合を「 n 分層」と定義する。図1に、実際の楽器の楽譜と拍階層の概念との対応付けを示す。あるステップが存在する時刻が属する層のうち、最下層（すなわち n が最小）のものが l 分層であるとき、そのステップは l 分層に存在するものとし、「 l 分（のステップ）」と表現する。たとえば、図1中の位置9に存在する音符は、4分層の他に8分層や16分層にも属しているが、その中で最下層である4分層に存在する音符として扱う。この表現

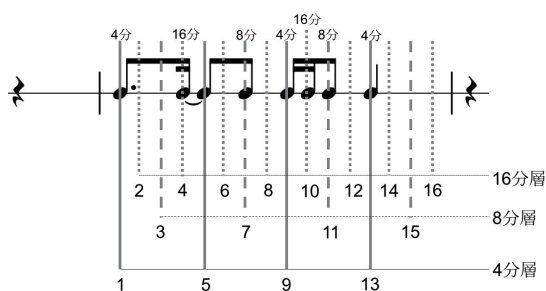


図1 拍階層・拍位置の概念。4分層に属する音符は、8分層や16分層にも属するが、その中で最下層である4分層に存在するものとする。各層の下に書かれた1~16の数字は、小節内での拍位置を示す。本稿ではこれを拍位置IDとする

Fig. 1 The concept of the beat layer and beat ID. Notes should belong to the lowest layer. For example, the last note in this figure exists in the 4th layer, though it also belongs to the 8th and 16th layer. Numbers below each layer show the position in the measure; that is the beat ID in this paper.

*1 DDR GLOBAL GATEWAY, <http://www.konami.jp/bemani/ddr/jp/> (最終閲覧日: 2018年1月11日)

*2 中学でダンス授業が必修化! ヒップホップ授業の実態…賛否両論の声…徹底解明! <https://studysapuri.jp/course/junior/parents/kyoiku/article-63.html> (最終閲覧日: 2018年1月11日)

*3 News - Stepmania, <https://www.stepmania.com> (最終閲覧日: 2017年6月13日)

方法を、本稿では拍階層表現と呼ぶ。図 1 中で、音符の上に書かれた「4 分」「8 分」が各音符の拍階層表現である。この表現は、実際の楽器の楽譜における「4 分音符」などの定義とは異なるが、音楽ゲームにおいてはこのように表現されることが多い。音楽ゲームの操作には、多くの場合「音長」の概念が存在しないことに起因すると考えられる。

2. 関連研究

楽器演奏を対象とした楽譜の生成や難易度調整手法はすでにいくつか提案されている。ニューラルネットワークモデルを利用した自動作曲システムの例として、LSTM を用いた Liang らの手法 [3] や、CNN に基づく GAN を用いた Yang らの手法 [4] などが存在する。難易度調整に関しても、様々な楽器の譜面を対象とした研究が報告されている。矢澤らはギターを対象として、演奏者の習熟度に合わせて音響信号からタブ譜を自動生成する手法を提案している [5]。ギターは 1 つの音階に対して何種類かの運指が該当する場合がある。弦を押さえる手の形の変化を重み付き有向グラフでモデル化することで最尤の運指を推定しており、演奏者の習熟度をグラフの重みに反映させることで習熟度に合わせたタブ譜の生成を実現している。福田らはピアノを対象として簡略化した楽譜を生成する演奏練習支援システムを提案している [6]。楽譜中の演奏が難しい部分を 3 パターンに分類し、パターンごとに定義されたルールに基づいて楽譜を簡略化している。簡略化ルールは人手で定義されており、事例ベースや機械学習ベースでの譜面の簡略化は行われていない。これらの楽譜簡略化は、特定の楽器においては強力であるが、楽器ごとに異なるルールを設定する必要がある。一方で、難しい譜面と簡単な譜面の関係を学習することによる機械学習ベースの簡略化は、対応付いた譜面データを大量に集めることができれば、異なる楽器やゲームであっても共通の学習モデルを用いることができる。実際の楽器では、同じ楽曲で難易度が異なる譜面を集めることは困難であるが、本稿で対象とする音楽ゲームにおいては、1 つの曲に対して難易度の異なる複数の譜面が用意されているため、難易度間で対応付いた譜面データが十分に存在している。提案手法では、この音楽ゲームの特性をとらえて、難易度間の関係モデルを構築することで従来の楽器に対する難易度調整では実現が困難であった機械学習ベースの譜面簡略化を実現する。

ゲーム内のコンテンツを自動生成する試みは PCG (Procedural content generation) と呼ばれている [7]。音楽ゲーム以外のゲームジャンルにおいては、プレイヤーの実力や行動をモデル化し、それに合わせたステージを作成する試みもなされている。例として、Pedersen らは、アクションゲーム *Super Mario Bros*® を題材に、ゲーム内の状況からプレイヤーの行動をモデル化し、それに合わせたマップを生成する手法を提案している [8]。音楽ゲームにおけるコンテンツ生成の試みとして、香川らは、音楽ゲーム中でプレイ

やに演奏させるべき音を自動抽出することを目的として、MIDI データから重要音を抽出する手法を提案している [9]。入力された MIDI データに suffix tree を適用し、楽曲中で繰り返し現れるフレーズを重要フレーズとして抽出することで、ゲームとしてプレイヤーに演奏を指示すべき音を抽出している。一方で、重要音を抽出した後の指示符配列生成や、難易度の調整については今後の課題とされている。

3. 難易度が異なる音楽ゲーム譜面間の関係性についての分析

音楽ゲームの多くでは、同一の楽曲に対して難易度の異なる複数の譜面を用意することでユーザの幅広いニーズに対応している。難易度が異なったとしても同一の楽曲に対する譜面であるため、それらの複数の譜面間には一定の関係性が存在すると考えられる。難易度が異なる譜面間の関係性について、以下の仮説を立てた；

仮説 1 簡単な譜面でステップが存在する時刻には、同一曲の難しい譜面でもステップが存在する確率が高い。

仮説 2 反対に、難しい譜面においてステップが存在しない時刻に、同一曲の簡単な譜面でステップが存在する確率は低い。

仮説 3 仮説 1 と仮説 2 により、難しい譜面におけるステップの存在情報は、簡単な譜面でステップが存在する時刻を絞り込むうえで有用である。

また、簡単な譜面に見られる音楽の特徴について、以下の仮説を立てた；

仮説 4 アクションを要求する回数が多いほど複雑で難解になるため、簡単な譜面は難しい譜面よりもステップ数が少ない。

仮説 5 簡単な譜面は、難しい譜面よりも容易なリズムで構成されることが多い：たとえば、簡単な譜面では 4 分の表打ちに比べ、裏打ちや 16 分音符の連打を含むことは少ない。

仮説 6 簡単な譜面では、小節内でも特定の拍にステップが配置されることが多い：同じ 4 分の表打ちでも、4 分の 4 拍子における 1 小節内で、1 拍目や 3 拍目は 2 拍目などに比べ知覚しやすい。

後の 3.2 節において、これらの仮説に対する検証を行う。

3.1 データセット

本研究では、Donahue らの先行研究 [2] にならい、Fraxtil*4 と ITG*5 の 2 データセットを用いる。表 1 に、それ

*4 Fraxtil's Simfile, <https://fra.xtil.net/simfiles/> (最終閲覧日：2017 年 11 月 28 日) より「Tsunami III」「Fraxtil's Arrow Arrangements」「Fraxtil's Beast Beats」の 3 パッケージ

*5 「In The Groove」<http://simfiles.stepmania-online.com/In%20The%20Groove%201.zip> および「In The Groove 2」<http://simfiles.stepmania-online.com/In%20The%20Groove%202.zip> の 2 パッケージ (最終閲覧日：2017 年 11 月 28 日)

表 1 データセットの情報 (Donahue らの論文 [2] の Table 1 を参考に作成した)

Table 1 The statistics of the database (with reference to the existing paper by Donahue et al. [2]).

データセット	Fraxtil	ITG
作者の人数	1	8
バックの数	3	2
楽曲数	90 (3.1 時間)	133 (3.9 時間)
譜面数	450 (15.3 時間)	652 (19.0 時間)
ステップ数/秒	3.135	2.584
公開開始年	2013~2014	2004~2005

それぞれのデータセットの情報を示す。

両データセットにおいて、1 曲につき難易度別に 5 種類の譜面が収録されている；ただし、ITG 内では、4 種類の譜面のみ収録されている楽曲が 13 曲存在する。5 種類の譜面には、難易度が低い順に Beginner, Easy, Medium, Hard, Challenge という名称が付与されている。本稿でも譜面の難易度には、これらの名称を用いる。

3.2 仮説の検証

ある時刻に Challenge 譜面でステップが存在する事象を E_c 、Beginner 譜面で存在する事象を E_b として、

- Beginner 譜面で矢印が存在する時刻に、Challenge 譜面でも矢印が存在する確率 $P(E_c|E_b)$
- Challenge 譜面で矢印が存在しない時刻に、Beginner 譜面でも矢印が存在しない確率 $P(\neg E_b|\neg E_c)$

の 2 種類の値を、各データセットから算出した。時刻の最小単位は、既存研究 [2] に基づいて 10 ms に設定し、この 1 単位を 1 フレームとした。このとき、各楽曲の Challenge 譜面で最初のステップが存在するフレームから最終ステップが存在するフレームまでを、計算の対象とした。

表 2 と表 3 に、データセット Fraxtil と ITG それぞれにおいてステップが存在するフレーム数を示す。このデータに基づく、Fraxtil では $P(E_c|E_b) \approx 0.912$, $P(\neg E_b|\neg E_c) \approx 0.999$ 、ITG では $P(E_c|E_b) \approx 0.895$, $P(\neg E_b|\neg E_c) \approx 0.999$ と算出された。

また、難しい譜面のステップ存在情報が、簡単な譜面でのステップ存在時刻を絞り込むために有用であるかどうかを確認するため、ランダム選出によるチャンスレベルを算出した。Fraxtil において、全 1,069,769 フレームのなかから、 $n(E_c)$ と同じ 62,971 フレームを無作為に選んだとき、そのなかに含まれる E_b を満たすフレーム数の期待値は、 $\sum_{k=0}^{7158} ({}^{7158}C_k \times 1062611 {}^{62971}C_{62971-k}) / 1069769 {}^{62971}C_{62971} \approx 421.35$ となる。一方、表 2 より、 E_c を満たす 62,971 フレーム中には、 $E_c \cap E_b$ を満たすフレームが 6,531 フレーム存在する。これは、 E_b を満たすフレームを選ぶにあたって、 E_c を満たすフレームをすべて選ぶという単純作業であってもチャンスレベルを 15 倍近く上回ることを意味する。

表 2 Challenge 譜面と Beginner 譜面の関連 (Fraxtil データセット)

Table 2 The relation between Challenge and Beginner charts in the Fraxtil dataset.

		Challenge		合計
		E_c	$\neg E_c$	
Beginner	E_b	6,531	627	7,158
	$\neg E_b$	56,440	1,006,171	1,062,611
合計		62,971	1,006,798	1,069,769

表 3 Challenge 譜面と Beginner 譜面の関連 (ITG データセット)

Table 3 The relation between Challenge and Beginner charts in the ITG dataset.

		Challenge		合計
		E_c	$\neg E_c$	
Beginner	E_b	7,001	824	7,825
	$\neg E_b$	53,890	1,174,716	1,228,606
合計		60,891	1,175,540	1,236,431

表 4 各データセットの難易度別情報

Table 4 The statistics of the dataset for each difficulty level.

難易度	Beginner	Easy	Medium	Hard	Challenge
Fraxtil データセット	楽曲数	90	90	90	90
	ステップ総数	7,158	19,813	30,345	45,350
	ステップ数/秒	0.669	1.838	2.802	4.175
ITG データセット	楽曲数	133	133	133	120
	ステップ総数	8,648	20,796	32,536	48,254
	ステップ数/秒	0.627	1.504	2.349	3.485

したがって、

- 仮説 1 の確率はおおよそ 90%である
- 仮説 2 の確率はおおよそ 100%である
- 仮説 3 は、少なくとも再現率の向上において有用であるといえる。

の 3 点が確認され、仮説 1~3 は証明された。

次に、表 4 に、各データセットの難易度ごとの 1 秒あたりのステップ数をそれぞれ示す。同表から、両データセットにおいて、Beginner 譜面や Easy 譜面における 1 秒あたりのステップ数は、Hard 譜面や Challenge 譜面と比較して少ないことが分かる。このことから、仮説 4 の妥当性が示唆された。

また、難易度ごとにステップが存在する拍階層の構成比を算出した。図 2 に、難易度ごとの各拍階層のステップ存在率を示す。同図から、Challenge 譜面では 4 分・8 分・16 分それぞれの層に 20%以上のステップが存在する一方で、Beginner 譜面においては 99%以上、Easy 譜面においては 93%以上のステップが 4 分層に存在することが分かる。このことから、簡単な譜面では 4 分音符の表打ちのような、人間が容易に理解できるリズムで構成されていることが分かり、仮説 5 での要件はほぼ満たされるものと判断される。

最後に、仮説 6 の検証として、難易度ごとの小節内における各拍位置でのステップ生起確率を調査した。Stepmania の譜面データはすべて 4 分の 4 拍子を基準として作成されている。また、前述のとおり、ステップの 90%以上は 4 分、

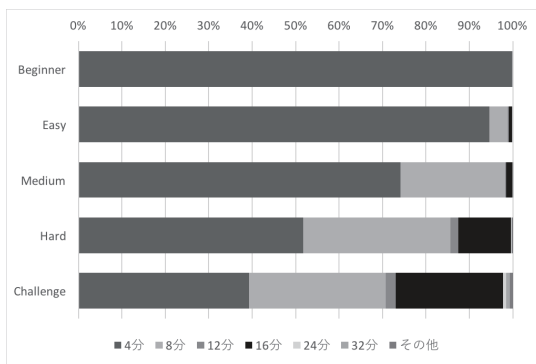


図 2 拍階層ごとのステップ存在比

Fig. 2 The step frequency in each beat layer.

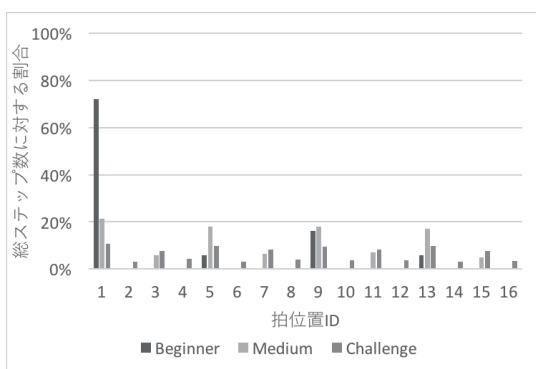


図 3 小節内のステップ存在分布

Fig. 3 The step distribution in the measure.

8 分, 16 分のいずれかの層に存在している。そこで, 16 分割された小節の各位置にステップをクオンタイズした。これにより, 3 連符 (12 分層) などの特殊な拍位置に存在するステップもすべて 16 分割されたいずれかの拍位置に存在するものとして扱う。図 3 に, Beginner, Medium, Challenge についての, 小節内での各位置のステップ生起確率を示す。ここで, 図中の横軸 1~16 は図 1 中の 1~16 の各位置に対応する。Medium および Challenge では, どの位置でのステップ生起確率も 25% を超えておらず, 小節全体に広く分布しているといえる。一方で, Beginner においては 70% 近くが 1 番の位置, すなわち小節の先頭に存在している。よって, 仮説 6 について, 簡単な譜面では小節の第 1 拍目にステップが存在する確率が高いといえる。

4. 提案手法

3 章での分析結果をもとに, ダンス譜面の難易度を自動的に調整する手法を提案する。Donahue らの先行研究 [2] では, ダンス譜面の生成を「時刻決定」「向き選択」というサブタスクへ切り分け, 各タスクについて音響特徴を入力としたニューラルネットワークモデルによって実現している。本稿では, 先行研究における「音響-ダンス譜面」の関

係学習に対して「難易度が異なるダンス譜面どうし」の関係学習を行うことで, 先行研究の手法の問題点であったダンス譜面の難易度調整を実現する。なお, モデル構築のためのライブラリとして Keras^{*6}を利用した。

4.1 参考手法: *Dance Dance Convolution*

先行研究 *Dance Dance Convolution* では, 譜面の生成を 2 つのサブタスクに分割している。

1 つ目は, 音楽データをもとに, 曲のどのタイミングに合わせてステップを配置するかを推定する, 時刻決定タスクである。時刻決定タスクでは, ステップが存在する時刻のみを推定し, ステップが示す矢印の向きは扱わない。このタスクは, 音楽情報処理における, 楽器の発音検出技術である onset detection [10] と密接に関わる。Donahue らはスペクトログラムに対して CNN を用いる onset detection 技術 [11] を応用している。

2 つ目のタスクは, 時刻決定タスクで得た時刻集合をもとに, どの時刻にどの向きの矢印を配置するかを推定する向き選択タスクである。こちらは, 生成した系列データから, 次のデータを予測・生成する課題であり, 自然言語処理における文章生成タスク [12] と類似している。

2 つのサブタスクでは, どちらも時系列情報を扱う。Donahue らは, 両タスクに対して, RNN の一種である LSTM [13] を用いた学習モデルを提案している。時刻決定タスクに対しては, 音楽データから得られる複数のスペクトログラム [14] を入力, 各時刻のステップ存在確率を出力とする, CNN と LSTM を組み合わせたモデルを構築している。向き選択タスクに対しては, 64 ステップの内容を入力し, 続く 65 ステップ目の内容を推定し出力する LSTM モデルを構築している。各モデルの詳細については Donahue らの論文 [2] を参照されたい。

4.2 提案手法: *Dance Dance Adaptation*

本稿で提案するモデル: *Dance Dance Adaptation* は, 時刻決定タスクに対して, ステップを配置する位置の難易度を調整する。図 4 に, 参考手法と提案手法の時刻決定モデルの比較を示す。参考手法のモデルでは, 入力された音楽データを時刻単位で切り分け, CNN 層に通すことで得られた音響特徴を, LSTM 層の入力としている。一方で提案手法では, 音楽データの代わりに難易度が高い譜面のデータを入力とし, 時刻単位で切り分けた譜面データから得られる特徴量を LSTM 層の入力とする。時刻の最小単位は, Donahue らにならって 10 ms に設定した。

難易度が高い譜面のデータから獲得する特徴量は, 3 章で述べた仮説に基づき, 以下の 3 要素とする。

NA: 矢印存在数 対象の時刻に存在し, ゲームのプレイ

^{*6} Chollet, François et al., <https://github.com/fchollet/keras> (最終閲覧日: 2018 年 1 月 11 日)

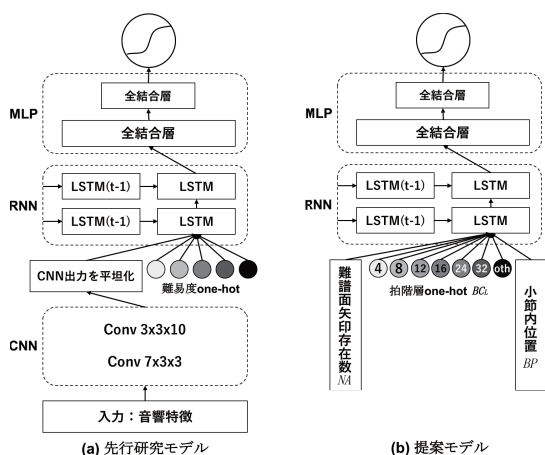


図 4 時刻決定タスクに対する既存モデルと提案モデルの比較. 図 (a) は Donahue らの論文 [2] の Fig. 5 を参考に作成した

Fig. 4 Comparison of the existing model (a) and the proposed model (b). (a) is made with reference to the paper by Donahue et al. [2].

やに足を動かすことを要求する矢印の数. 非負整数値の 1 次元で表される.

BC_L : 拍階層 対象時刻が存在する拍階層 ($L: 4, 8, 12, 16, 24, 32, \text{other}$) の 7 次元で構成され, 対応する要素のみが 1 となる one-hot ベクトルで表す.

BP : 小節内位置 対象の時刻が属する小節の先頭を 0, 終端を 1 として, 対象時刻が対応する数値. $[0, 1]$ の連続値をとる 1 次元で表される.

モデルに入力する際にはこれらを結合し, t 時刻における入力ベクトル \mathbf{v}_t は,

$$(\mathbf{NA}, \mathbf{BC}_4, \mathbf{BC}_8, \mathbf{BC}_{12}, \mathbf{BC}_{16}, \mathbf{BC}_{24}, \mathbf{BC}_{32}, \mathbf{BC}_{\text{other}}, \mathbf{BP}),$$

とした 9 次元のベクトルとして扱う.

図 5 に, 譜面データから得られる特徴量の例を示す. $t = 7.58$ 時の譜面を例にとると, 右の矢印のみが存在するため $\mathbf{NA} = 1$, 対象の時刻を表すために小節を 16 等分する必要があるため $\mathbf{BC}_{16} = 1$ となり $(0, 0, 0, 1, 0, 0, 0)$, 対象の時刻は 3 小節目内の $\frac{3}{16} = 0.1875$ の位置に存在するため $\mathbf{BP} = 0.188$ がそれぞれ得られ, 入力ベクトル $\mathbf{v}_{7.58}$ は $(1, 0, 0, 0, 1, 0, 0, 0, 0.188)$ と表現される.

図 4 の出力層は sigmoid 関数であり, $(0, 1)$ の連続値をとる. この出力値は時刻 t にステップが存在する確率 $SP(t)$ を表す. 楽曲中のすべての時刻について, ステップ存在確率 $SP(t)$ を推定することで, 時系列データ SP を生成する. SP 中の極大値を示す t にステップを配置する.

5. 評価実験

先行研究にならい, 各データセット内の楽曲を 8 : 1 : 1 の割合で, 学習データ・検証データ・テストデータの 3 カ

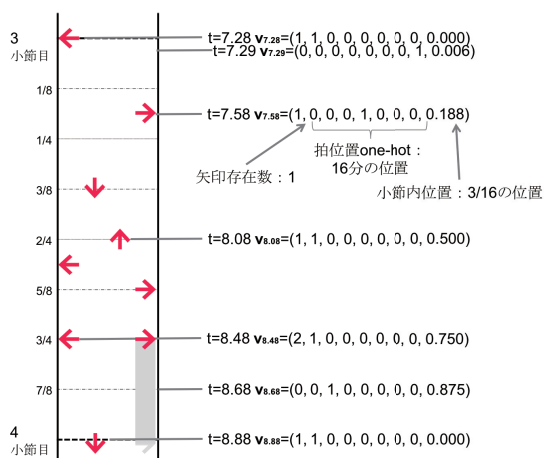


図 5 譜面データから得られる特徴量

Fig. 5 The features extracted from chart data.

テゴリにランダムで振り分けた. 時刻決定モデルと向き選択モデルそれぞれについて, 客観評価実験を行った.

5.1 時刻決定モデルの評価

提案する時刻決定モデルを以下の設定で学習させた.

- Challenge 譜面を入力, Beginner 譜面を正解データ
- Challenge 譜面を入力, Medium 譜面を正解データ

これによって得られた, 4 種類の学習結果 (2 データセット \times 2 設定) それぞれについて考察した.

評価指標として, 先行研究 [2] で用いられた F 値に加え, リズムの複雑性をどの程度表現できているかを評価する指標を用意した. これらの組合せにより, 提案手法の有効性を総合的に評価した.

5.1.1 F 値による参考手法との比較

テストデータに振り分けられた楽曲に対して, 学習済みモデルが予測した時刻と, 正解データにおいてステップが存在する時刻を比較し, F 値を算出した. 楽曲ごとの適合率と再現率から得られる F 値を合計し, 楽曲数で除算し平均したものを, $F\text{-score}^c$ とする. ステップ単位で算出したマイクロ F 値は, $F\text{-score}^m$ とする. 表 5 に, 提案手法と比較手法それぞれの F 値を示す. 表中, **DDA** は提案手法の値を示し, **DDC** は先行研究 [2] で報告された値を示す. 先行研究は, 音楽音響信号から譜面を生成するという, 本研究よりも広範囲かつ困難な問題設定でダンスゲーム譜面を生成している. したがって, 本実験で性能が上回ったからといって, 必ずしも本手法が比較手法を上回る能力を持った深層学習モデルであるということにはならない. しかし, 難易度調整という限られた目的における提案手法の性能の目安とするため, 既存手法で難易度調整を行った場合の性能比較を行った. 先行研究では譜面難易度ごとの詳細結果は示されておらず, Fraxtil データセット内の Beginner 譜

表 5 時刻決定モデル評価実験結果

Table 5 The results for step placement experiments.

データセット	生成難易度	手法	F-score ^c	F-score ^m
Fraxtil	Beginner	DDA	0.693	0.693
Fraxtil	Beginner	DDC	0.389	(報告なし)
ITG	Beginner	DDA	0.755	0.747
Fraxtil	Medium	DDA	0.756	0.748
ITG	Medium	DDA	0.625	0.626
Fraxtil	(すべて)	DDC	0.681	0.756
ITG	(すべて)	DDC	0.697	0.721

面, Challenge 譜面についての F -score^c のみの報告であったため, 該当の値のみを記載する. また, 参考として先行研究で報告された 5 種類の難易度すべてについての F 値についても記載する.

Fraxtil データセット・Beginner 譜面について, 提案手法は参考手法のおおよそ 1.8 倍の F -score^c が確認された. ITG データセット・Beginner 譜面についても, 0.7 を超える F 値が確認され, 参考手法の Fraxtil データセット・Beginner 譜面についての F -score^c を上回る値が確認された. 提案手法の結果のうち, ITG データセット・Medium 譜面については, 楽曲ごとの適合率平均が 0.514, 再現率平均が 0.884 と適合率が再現率に比べ大幅に低い値を示し, F 値が他と比較して低い値を示した. 適合率が低い要因は, 正解データ中に存在するステップ数に比べて多いステップ時刻を推定していることと考えられる. 高い適合率は確認されなかったものの, 5.1.2 項に詳細を後述するステップの拍階層ごとの分布や小節内の位置についての分析から, 各難易度のリズムの複雑性が十分に反映された譜面が生成されたと考察される. 一方で, F 値の向上も今後の課題として取り組むべきであり, この問題については, 出力された時系列から極大値を検出する際にしきい値を設定し, 検出する極大値数を調整することで解決されと考える.

5.1.2 データセットと提案手法生成譜面の統計比較

学習済みモデルにテストデータを入力し得られた生成譜面, および, データセット内のすべての譜面について, 3 章で述べた仮説 4~6 に基づき以下の統計指標を算出した.

- 1 秒あたりのステップ数
- ステップが存在する拍階層の構成比
- 小節内位置の分布

生成譜面とデータセットの各統計指標を比較することで, 生成された譜面がデータセットの性質を正しく反映しているかを確認した. 留意した指標のうち, (a) はモデルへの入力特徴に用いていない観点であり, この指標が生成譜面とデータセット間で近い値を示せば, 提案モデルはデータセットの性質を学習するモデルとして妥当であると考えられる. (b) と (c) は, 時刻決定モデルへの入力特徴 BC_L および BP と関連し, これらの指標が生成譜面とデータセット間で近ければ, それぞれに対応する入力特徴が譜面の難

表 6 1 秒あたりのステップ数

Table 6 The average and variance of steps per second.

データセット	難易度	手法	D_m		
			E_m	V_m	D
Fraxtil	Beginner	Original	<u>0.673</u>	<u>0.012</u>	0.669
Fraxtil	Beginner	DDA	0.654	0.036	0.645
Fraxtil	Beginner	DDC	0.824	0.356	0.875
ITG	Beginner	Original	<u>0.629</u>	<u>0.009</u>	0.627
ITG	Beginner	DDA	0.625	0.007	0.623
ITG	Beginner	DDC	1.215	0.550	1.203
Fraxtil	Medium	Original	<u>2.793</u>	<u>0.501</u>	2.802
Fraxtil	Medium	DDA	3.904	0.773	4.020
Fraxtil	Medium	DDC	2.726	0.162	2.702
ITG	Medium	Original	<u>2.351</u>	<u>0.319</u>	2.349
ITG	Medium	DDA	4.006	0.758	4.021
ITG	Medium	DDC	3.283	1.938	3.248

易度調整モデルへの入力として妥当であるといえる.

5.1.2.1 1 秒あたりのステップ数

楽曲 m の 1 秒あたりのステップ数 D_m は式 (1) で算出する.

$$D_m = \frac{\text{譜面中のステップ数 (個)}}{\text{楽曲の長さ (秒)}}. \quad (1)$$

D_m の平均を E_m , 分散を V_m とした. また, 全体の 1 秒あたりのステップ数 D は, 式 (2) で算出した.

$$D = \frac{\text{対象譜面集合内のステップ数の総和 (個)}}{\text{対象楽曲集合内の楽曲の長さの総和 (秒)}}, \quad (2)$$

ここで, 対象譜面集合と対象楽曲集合は, モデル生成結果については全テストデータに対する生成譜面および楽曲, もととなるデータセットについては学習データ・検証データ・テストデータすべての譜面および楽曲がそれぞれ相当する.

表 6 に, E_m , V_m , D をそれぞれ示す. 手法の列が「Original」となっているものは, もととなったデータセット全体について算出した結果である. 「DDA」となっているものは, 提案モデルが生成した譜面について算出した結果であり, 「DDC」となっているものは, テストデータに振り分けられた楽曲を参考手法のオンラインデモ^{*7}に入力し得られた譜面について算出した結果である.

両データセットの Beginner 難易度について, 提案手法が参考手法よりも, 元データセットに近い E_m , V_m および D の値を得た. また, 参考手法は, 元データセットおよび提案手法と比較して高い分散 V_m を示した. これは, 参考手法で Beginner 譜面を出力する際, 入力音源によってステップ数にばらつきがあるという問題点があり, 提案手法はこれを解決できたことを意味する. 一方で, Medium 難易度については, 参考手法の E_m および D の値が, 提案手法よりも元データセットに近い値を示し, Donahue らの報告 [2] のとおり難易度が高いほど参考手法の精度が上がる

^{*7} <http://deepx.ucsd.edu/ddc> (最終閲覧日: 2017 年 11 月 29 日)

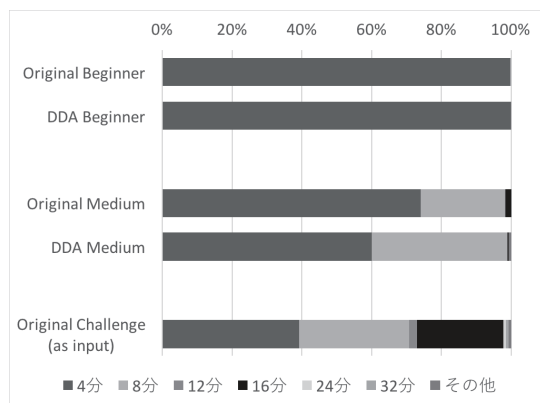


図 6 生成譜面および元データセットの拍階層ごとのステップ存在比
Fig. 6 The step frequency for each beat layer: comparison of the generated and handmade charts.

ことが確認された。提案手法で生成した Medium 譜面は、 E_m および D の値が元データセットよりも高く、5.1.1 項で述べたとおり正解データより多数のステップを出力していることが確認された。

5.1.2.2 ステップが存在する拍階層の構成比

参考手法のオンラインデモで出力される譜面データには拍階層および小節の情報が含まれていないため、提案手法と元データセットで比較した。集計方法は 3.2 節で行った仮説 5 の検証と同一の方法をとった。図 6 に、元データセットおよび提案手法による生成譜面それぞれの各拍階層ステップ存在率を示す。ここで、Original は元データセットの拍階層構成比、DDA は提案手法生成譜面の拍階層構成比をそれぞれ示す。Beginner, Medium, Challenge はそれぞれの難易度名を示す。図 6 中、最下段の Original Challenge は、提案モデルに入力した Challenge 譜面の拍階層構成比である。

Original Beginner–DDA Beginner 間の相関係数は 0.999、Original Medium–DDA Medium 間の相関係数は 0.957 となり、両難易度において強い正の相関を示した。また、入力した Challenge 譜面には、8 分のステップが約 31%、16 分のステップが約 25%含まれているが、生成された Beginner 譜面には 8 分、16 分ともに 1 つも存在せず、Medium 譜面に含まれる 16 分のステップは約 0.22%であった。したがって、提案手法は、元データセットの拍階層構成比に従って 8 分や 16 分のステップを削減することで、難易度が低い譜面を生成可能であることが示唆された。これにより、モデルへの入力特徴として拍階層 BC_L を用いるの妥当性が示唆された。しかし、Medium 譜面については、元データセットの Medium 譜面と比較して 8 分ステップの存在率が高い結果を示した。現時点で出力される譜面から一部の 8 分ステップを適切に削減可能となれば、5.1.1 項で述べた適合率の問題も解決され、より元データセットに近い性質の譜

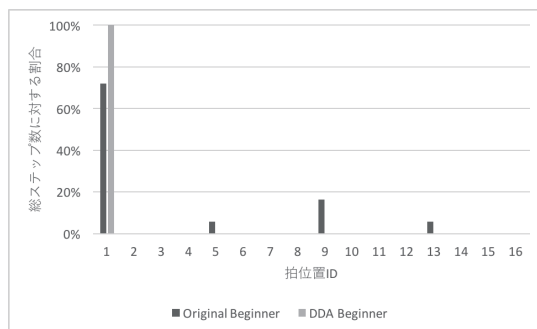


図 7 生成譜面および元データセットの小節内位置分布 (Beginner 難易度)

Fig. 7 The steps distribution in measure: the comparison of the generated and handmade Beginner charts.

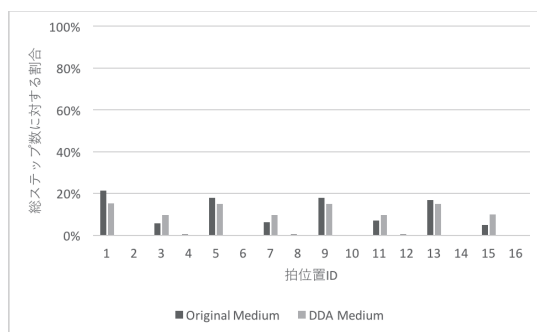


図 8 生成譜面および元データセットの小節内位置分布 (Medium 難易度)

Fig. 8 The steps distribution in measure: the comparison of the generated and handmade Medium charts.

面が生成可能になると考えられる。

5.1.2.3 小節内の各拍位置でのステップ生起確率

本観点についても、拍階層構成比と同様の理由で、提案手法と元データセットの比較のみを行った。集計方法は、3.2 節で行った仮説 6 の検証と同一の方法をとった。

図 7 に、Beginner 難易度について、元データセットと提案手法生成譜面それぞれの小節内各拍位置でのステップ生起確率を示す。提案手法で生成された Beginner 譜面は、すべてのステップが小節の 1 拍目に存在していた。図 8 に、Medium 難易度についての小節内各拍位置でのステップ生起確率を示す。Original Beginner–DDA Beginner 間の相関係数は 0.971、Original Medium–DDA Medium 間の相関係数は 0.939 となり、両難易度において正の相関を示した。相関係数の値から、提案手法が生成する譜面の小節内位置分布は、元データセットの小節内位置分布に従うことが確認された。しかしながら、Beginner 譜面について、元データセット内に 16%程度存在するはずの位置 9 (4 分層の 3 拍目) のステップなどが学習できず、最頻値である位置 1 (4 分層の 1 拍目) にステップが集中した譜面が生成

表 7 1-gram 出現頻度上位 7 件

Table 7 The top seven steps of 1-gram frequency.

Original				Generated			
All	Beginner	Easy	Medium	Beginner	Easy	Medium	Challenge
Step	Count	Step	Count	Step	Count	Step	Count
↓	76,275	←	4,205	←	9,352	↓	14,339
↑	73,910	→	4,180	→	9,250	↑	13,841
←	69,192	↑	3,607	↑	9,217	←	12,967
→	68,482	↓	3,537	↓	8,955	→	12,853
↔	6,653	↔	746	↔	1,162	↔	1,515
↓↑	3,762	↓↑	128	↓↑	474	↓↑	847
↓ hold	3,029	←↓	14	← hold	277	←↓	564

された。これより、モデルへの入力特徴として小節内位置 BP を用いることについて、ある程度の妥当性はあるものの、学習がデータセット内の多数データに大きく依存し、少数データが反映されない出力を示す要因となっている可能性が示唆された。学習が不十分であった少数データについては、小節内位置を入力しない学習や、他の特徴量を補助的に追加した学習などの工夫により対応可能と考える。

5.2 向き選択モデルの評価

向き選択モデルは Donahue らと同一のモデルを利用しているため、モデルの性能評価については先行研究 [2] を参照されたい。テストデータを学習済みモデルに入力し生成された譜面から得られる統計と、元データセットから得られる統計を比較し、学習結果がデータセットの性質をどの程度反映しているかを確認した。先行研究でステップとされている 256 通りのステップ*8それぞれの出現頻度を集計し 1-gram とし、連続する 2 ステップの組合せについては 2-gram として集計した。

表 7 に、元データセット内の譜面と学習済みモデルによって生成された譜面それぞれにおける、1-gram 出現頻度が高い上位 7 件のステップを示す。表中の Original は元データセットの集計結果を表し、Generated は学習済みモデルが生成した譜面の集計結果を表す。また、All は 5 種類の難易度すべてを集計した結果を表し、Beginner, Easy, Medium はそれぞれ該当する難易度の譜面のみを集計した結果を示す。表中の Step 欄において、2 つの矢印が並んでいるものは 2 方向を同時に踏むことを指示するステップ、hold はその矢印を踏み続けることを指示するステップ、release は矢印を踏み続けた状態から離すことを表すステップ*9である。表 8 に、データセット内の各難易度間、およびデータセットの各難易度と生成譜面間についての、1-gram 出現頻度の相関係数を示す。表 8 上部より、データセット内の各難易度間の相関係数がいずれも 0.98 を超えており、強い正の相関を示した。これは、学習データ中の 1-gram 生起確率は難易度によって大きく変わらないことを

*8 (左, 下, 上, 右) 4 種類の矢印それぞれについて (踏む, 踏まない, 踏み続けている, 踏み続けた状態から離す) の 4 種類の状態が存在する, とされている。

*9 踏み続けをやめるタイミングの正確さはゲームの評価に含まれていない。このステップが来たあとは、該当矢印パネルから足を離してもよいし、踏み続けてもよい。

表 8 1-gram 難易度間相関係数

Table 8 The correlation coefficient matrix among difficulty levels for 1-gram frequency.

		Original					
		All	Beginner	Easy	Medium	Hard	Challenge
Original	Beginner	0.989	1.000	—	—	—	—
	Easy	0.998	0.996	1.000	—	—	—
	Medium	1.000	0.990	0.998	1.000	—	—
	Hard	0.999	0.984	0.995	0.999	1.000	—
		Challenge	1.000	0.986	0.997	0.999	1.000
Generated	Beginner	0.994	0.995	0.997	0.993	0.990	0.992
	Medium	0.994	0.994	0.997	0.993	0.990	0.993

表 9 2-gram 出現頻度上位 7 件

Table 9 The top seven step combinations of 2-gram frequency.

Original				Generated			
All	Beginner	Easy	Medium	Beginner	Easy	Medium	Challenge
1	2	Count	1	2	Count	1	2
↓	↑	29,481	→	←	1,613	↑	↓
↑	↓	29,180	←	→	1,587	↓	↑
→	←	22,445	↑	↓	1,318	→	←
←	→	22,411	↓	↑	1,316	←	→
↔	↔	18,443	↔	↔	812	↔	↔
↓↑	↓↑	17,753	↓↑	↓↑	810	↓↑	↓↑
↓→	↓→	17,643	↓→	↓→	805	↓→	↓→

表 10 2-gram 難易度間相関係数

Table 10 The correlation coefficient matrix among difficulty levels for 2-gram frequency.

		Original					
		All	Beginner	Easy	Medium	Hard	Challenge
Original	Beginner	0.954	1.000	—	—	—	—
	Easy	0.987	0.986	1.000	—	—	—
	Medium	1.000	0.959	0.989	1.000	—	—
	Hard	0.998	0.936	0.976	0.997	1.000	—
		Challenge	0.997	0.932	0.975	0.995	0.997
Generated	Beginner	0.922	0.972	0.965	0.927	0.904	0.896
	Medium	0.966	0.976	0.987	0.968	0.954	0.951

意味する。そのため、生成譜面–データセット内の特定難易度間の相関についても、Generated Beginner, Generated Medium とともに、データセット内のすべての難易度と 0.99 以上の強い正の相関を示した。

表 9 に、元データセット内の譜面と学習済みモデルによって生成された譜面における 2-gram 出現頻度が高い上位 7 件のステップをそれぞれ示す。表中の 1, 2 の欄は 2-gram の組を示し、1 のステップの次に 2 のステップが来ることを意味する。表 10 に、データセット内の各難易度間、およびデータセットの各難易度と生成譜面間についての、2-gram 出現頻度の相関係数を示す。表 8 と表 10 を比較すると、Original Beginner–Original Challenge 間など、難易度が離れている組合せの相関係数は、1-gram と比べて 2-gram の方が低いことが分かる。これより、難易度の高低によって、1-gram の出現頻度に大きな差はないが、2-gram の出現頻度には差があることが確認できる。生成譜面とデータセット内の譜面間の相関係数について、Generated Beginner は Original Beginner と最も強い相関を示す一方で、Original Challenge と最も弱い相関を示している。このことから、学習モデルは、Beginner 譜面など難易度の低い譜面でよく見られるステップ順序を学習し、出力できていると考えら

れる。しかしながら、Original Medium–Original Beginner 間の相関は、Original Challenge–Original Medium 間の相関より弱いにもかかわらず、Generated Medium–Original Beginner 間の相関は Generated Medium–Original Challenge 間よりも強い結果を示した。すなわち、生成された Medium 譜面は、データセット内の Medium 譜面よりも、より簡単な譜面の性質が反映されたものであるといえ、これは今後の検討課題である。

本稿では、2-gram までの統計を出したが、より長い 3-gram, 4-gram などについて統計することでさらに難易度ごとの特色が出ることも考えられる。より柔軟で適切な難易度調整を実現するため、より複数のステップの n -gram についても調査・考察を進めていく。

6. 提案手法の拡張性に関する考察

5 章の評価実験では、入力と出力の難易度に大きく差がある組合せで学習を行い、難易度を大幅に低下させる課題に対しおおむね良好な結果が得られることを確認した。この入出力の組合せを変えることで、入力譜面の難易度をより柔軟に調整する課題を解決できると考える。この際、複数難易度の譜面を混合して学習したり、異なる組合せで学習した複数の学習結果を併用したりすることで、現状の 5 段階よりもさらに細かい段階での難易度調整が可能となると考えられる。たとえば、Beginner 譜面を正解データとしたデータと、Easy 譜面を正解データとしたデータを 1:1 の割合で用意し、同じ学習器に学習させると、出力は Beginner と Easy の中間程度の難易度になることが期待される。このような入出力データへの工夫の考察と、それによる性能変化や効用についての考察は、今後の課題とする。

音響特徴から高難易度譜面を生成できる Donahue らの手法と、高難易度譜面から低難易度譜面を生成できる提案手法を併用することで、楽曲の音響信号から任意の難易度の譜面生成が可能になると考える。しかし、Donahue らの手法で出力されるダンス譜面には、テンポや拍など、提案手法への入力として必要な情報が含まれていない。一方で、拍などの音楽的な情報を音響信号から取得する手法はすでに提案されており [15], [16], これらの技術を Donahue らの手法に適用すれば、提案手法への入力として用いる音楽特徴を取得可能となる。

また、さらなる応用として、ダンスゲーム以外の音楽ゲーム、および楽器演奏用の譜面への適用が考えられる。機械学習ベースである提案手法のルールベースに対する優位点として、学習データが用意できれば同一構造もしくは類似構造のモデルが、他の音楽ゲームなどにも利用できる点があげられる。本稿で提案した時刻決定モデルであれば、

入力データ 発音 (指示符) の存在・拍・小節

正解データ 発音 (指示符) の存在

というデータの組が十分に用意できれば、Stepmania に限らず様々な音楽ゲーム (たとえば、太鼓の達人[®]*10 など) に応用可能である。

上記のデータが抽出できる形式であれば、入力データと正解データのフォーマットは一致していなくてもよい。これは、異なる音楽ゲーム間で譜面の関係を学習し、任意の音楽ゲームの譜面をもととして他の音楽ゲームの譜面を生成できる可能性を示している。本研究の 2 種類のサブタスクのうち、向き選択タスクは各音楽ゲームごとの操作デバイスに強く依存する課題であり、解決のためには各ゲームごとで学習を行う必要がある。一方、時刻決定タスク、すなわち難しいリズムを簡略化するタスクは、多くの音楽ゲーム、あるいは実際の楽器にも共通する課題である。そのため、ある音楽ゲームでの時刻決定モデルの学習結果が、他の音楽ゲームや実際の楽器の譜面における難易度調整にも利用できる可能性がある。

7. おわりに

本稿ではダンスゲームを題材とし、難易度が高い譜面と音楽的特徴を入力、難易度が低い譜面を出力とする時系列深層学習モデルによって、ダンス譜面の難易度を自動調整する手法 *Dance Dance Adaptation* を提案した。客観評価実験により、難易度が低い譜面を出力するタスクにおいて、提案手法は既存手法よりもおおむね良い性能を示した。

評価実験を通して、データセット内の少数データへの対処という新たな課題が得られた。この課題に対し、入力特徴とした拍階層・小節内位置の妥当性を検証するとともに、音響情報など新たな特徴量を併用する改善案が考えられる。今後はこれらの検討を進めたいと考えている。また、向き選択モデルについては、Donahue らのモデルをそのまま利用している。5.2 節の実験を通して得た知見をもとに、今後はこちらの改良についても検討したいと考えている。

今後の展望として、入力および正解データの難易度組合せを変更してモデルを学習させ、より細かな難易度調整を実現させる。さらに、提案モデルと学習結果が、ダンスゲーム以外の音楽ゲームや実際の楽器にも応用できる可能性についても検討する。

謝辞 本研究は、一部、科研費若手 B#16K21482 の助成のもと行われた。記して謝意を示す。

参考文献

- [1] KONAMI: ゲームが開く新しい可能性~KONAMI の「ダンスダンスレボリューション」を活用したアメリカ・ウェストバージニア州の「子どもたちの健康維持・増進」のための取り組み~, 2007 CESA ゲーム白書, コンピュータエンターテインメント協会, pp.24–29 (2007).
- [2] Donahue, C., Lipton, Z.C. and McAuley, J.: *Dance*

*10 太鼓の達人シリーズ公式サイト ドンダページ — バンダイナムコエンターテインメント公式サイト, <http://taiko-ch.net> (最終閲覧日: 2018 年 1 月 11 日)

Dance Convolution, *Proc. ICML 2017*, pp.1039–1048 (2017).

- [3] Liang, F.T., Gotham, M., Johnson, M. and Shotton, J.: Automatic Stylistic Composition of Bach Chorales with Deep LSTM, *Proc. ISMIR 2017*, pp.449–456 (2017).
- [4] Yang, L., Chou, S. and Yang, Y.: MidiNet: A Convolutional Generative Adversarial Network for Symbolic-Domain Music Generation, *Proc. ISMIR 2017*, pp.324–331 (2017).
- [5] 矢澤一樹, 糸山克寿, 奥乃 博: ギター演奏者の習熟度に合わせた音響信号からのタブ譜自動生成, 情報処理学会研究報告 (音楽情報科学), Vol.2013-MUS-100, No.17, pp.1–6 (2013).
- [6] 福田 翼, 中村栄太, 糸山克寿, 吉井和佳: 楽譜簡略化と自動補完伴奏によるピアノ演奏練習支援システム, 情報処理学会研究報告 (音楽情報科学), Vol.2017-MUS-114, No.21, pp.1–4 (2017).
- [7] Shaker, N., Togelius, J. and Nelson, M.J.: *Procedural Content Generation in Games: A Textbook and an Overview of Current Research*, Springer (2016).
- [8] Pedersen, C., Togelius, J. and Yannakakis, G.N.: Modeling player experience for content creation, *IEEE Trans. Computational Intelligence and AI in Games*, Vol.2, No.1, pp.54–67 (2010).
- [9] 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, EC 2015 予稿集, Vol.2015, pp.326–333 (2015).
- [10] Bello, J.P., Daudet, L., Abdallah, S., Duxbury, C., Davies, M. and Sandler, M.B.: A tutorial on onset detection in music signals, *IEEE Trans. Speech and Audio Processing*, Vol.13, No.5, pp.1035–1047 (2005).
- [11] Schluter, J. and Bock, S.: Improved musical onset detection with Convolutional Neural Networks, *Proc. ICASSP 2014*, pp.6979–6983 (2014).
- [12] Kim, Y., Jernite, Y., Sontag, D. and Rush, A.M.: Character-Aware Neural Language Models, *Proc. AAAI 2016*, pp.2741–2749 (2016).
- [13] Hochreiter, S. and Schmidhuber, J.: Long short-term memory, *Neural Computation*, Vol.9, No.8, pp.1735–1780 (1997).
- [14] Hamel, P., Bengio, Y. and Eck, D.: Building Musically-relevant Audio Features through Multiple Timescale Representations, *Proc. ISMIR 2012*, pp.553–558 (2012).
- [15] 後藤真孝, 村岡洋一: 音楽音響信号を対象としたビートトラッキングシステム—小節線の検出と打楽器音の有無に応じた音楽的知識の選択, 情報処理学会研究報告音楽情報科学 (MUS), Vol.1997, No.67, pp.45–52 (1997).
- [16] Maezawa, A., Okuno, H.G., Ogata, T. and Goto, M.: Polyphonic audio-to-score alignment based on Bayesian latent harmonic allocation hidden Markov model, *Proc. ICASSP 2011*, pp.185–188, IEEE (2011).



辻野 雄大 (学生会員)

2017年立命館大学情報理工学部メディア情報学科卒業。2017年同大学大学院情報理工学研究科情報理工学専攻博士前期課程入学。現在に至る。エンタテインメントコンピューティング、音楽情報処理に興味を持つ。ダンスゲームの公式世界大会に出場した実力と経験をもとに、ダンスゲーム譜面の自動生成手法に関する研究に従事。KONAMI Arcade Championship (KAC) 2013 Dance Dance Revolution 第5位, The 7th KAC Dance Dance Revolution 第5位, World Pump Festival 2016 Speed Male 部門日本代表。



山西 良典 (正会員)

2007年名古屋工業大学工学部知能情報システム学科卒業。2009年同大学大学院工学研究科情報理工学専攻博士前期課程修了。2012年同大学院博士後期課程修了。博士(工学)。2012年立命館大学情報理工学部助手, 2013年同特任助教。2014年同助教, この間, UBC(カナダ)客員助教。2018年同講師, 現在に至る。感性情報処理, Webインテリジェンス, マルチメディア情報処理, 音楽情報処理, 言語情報処理に関する研究に従事。電子情報通信学会, 人工知能学会, 日本感性工学会, ACM, ACL, 日本知能情報ファジイ学会各会員。



西原 陽子 (正会員)

2003年大阪大学基礎工学部卒業。2005年同大学大学院基礎工学研究科博士前期課程修了。2007年同大学院研究科博士後期課程修了。博士(工学)。日本学術振興会特別研究員を経て, 2008年東京大学大学院工学系研究科助教, 2009年同講師, 2012年立命館大学情報理工学部准教授, 現在に至る。インタラクション研究, コミック工学研究に興味を持つ。人工知能学会, 日本感性工学会各会員。



福本 淳一 (正会員)

1984 年広島大学工学部第 2 類卒業。
1986 年同大学大学院工学研究科システム工学専攻博士前期課程修了。同年
沖電気工業(株)入社。1992～1994 年
英国マンチェスター科学技術大学 Ph.D.
コース在学。2000 年立命館大学理工

学部情報学科助教授，2004 年同大学情報理工学部メディア
情報学科助教授，2006 年同教授，現在に至る。Ph.D. 談話
構造解析，質問応答技術，情報抽出，比喩表現理解等に興
味を持つ。電子情報通信学会，人工知能学会，言語処理学
会，観光情報学会各会員。