

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 山西 良典² 西原 陽子² 福本 淳一²

概要：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、初級者用の低難易度の譜面を充実させることが必要となる。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、難易度が高い譜面とそこから得られる音楽的特徴を入力、難易度が低い譜面を出力とする時系列深層学習モデルを構築した。提案手法によって、低難易度のダンス譜面において指示符を配置すべき発音タイミングの推定モデルを構築することで、難易度の自動調整を実現した。時刻決定タスクと向き決定タスクのそれぞれに対して行った性能評価の結果、提案手法は、人手で作成された低難易度のダンス譜面の特性を捉えた譜面を自動生成可能であることを確認した。

1. はじめに

今日のビデオゲーム市場で、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、プレイヤー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*[®] (DDR) が挙げられる。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。アメリカの 웨스트バージニア

州においては、肥満児童問題への対応策の一環として、学校授業にDDRを取り入れた事例も報告されている [1]。

音楽ゲームの譜面は、ゲームの制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。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、これらを活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*^{*1} 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全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、楽曲の特徴を捉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

¹ 立命館大学大学院情報理工学研究科
College of Information Science and Engineering,
Ritsumeikan University, 1-1-1 Nojihigashi, Kusatsu
525-8577, Japan

² 立命館大学情報理工学部
College of Information Science and Engineering,
Ritsumeikan University, 1-1-1 Nojihigashi, Kusatsu
525-8577, Japan

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

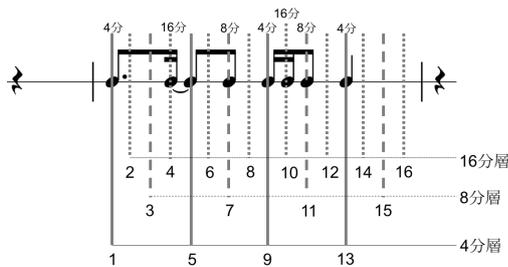


図 1 拍階層・拍位置の概念．4 分層に属する音符は，8 分層や 16 分層にも属するが，その中で最下層である 4 分層に存在するものとする．各層の下に書かれた 1~16 の数字は，小節内での拍位置を示す．本稿ではこれを拍位置 ID とする．

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 $n \geq 4$ ）することで表現出来る時刻集合を「 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を対応付けを示す．そして，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．例えば，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

香川らは，音楽ゲーム中でプレイヤー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 [3]．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．

音楽ゲームではなく，実際の楽器を対象とした楽譜簡略

化の事例として，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 [4]．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，事例ベースでの譜面の簡略化は行われていない．これは，実際の楽器の楽譜は基本的に 1 曲につき 1 つであり，任意の楽曲が低難易度に編曲された事例を大量に集めることが難しいことが理由の一つとして挙げられる．一方で，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譜面が用意されていることが多い．これは，簡単な譜面と難しい譜面がペアとなったデータが多数存在していることを意味し，上記の問題点を解決していると考えられる．譜面を自動で生成するという観点では，音楽情報処理における自動作曲とも関連する．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は多数提案されており，例えば RNN を用いた Liu らの手法 [5] や，CNN に基づく GAN を用いた Yang らの手法 [6] などが存在する．

また，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]．

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 仮説 1 の対偶より，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的特徴について，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

表 1 データセットの情報

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

仮説 5 簡単な譜面は、難しい譜面よりも容易なりズムで構成されることが多い：例えば、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

今研究では、Donahue らの先行研究 [2] にならい、Fraxtil *2 と ITG *3 の 2 データセットを用いる。表 1 に、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 10ms 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と、Fraxtil では $P(E_c|E_b) \approx 0.912$, $P(\neg E_b|\neg E_c) \approx$

*2 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

*3 「In The Groove」<http://stepmaniaonline.net/downloads/packs/In%20The%20Groove%201.zip> および「In The Groove 2」<http://stepmaniaonline.net/downloads/packs/In%20The%20Groove%202.zip> の 2 パッケージ [最終閲覧日: 2017 年 11 月 28 日]

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

		Challenge		合計
		E_c	$\neg E_c$	
Beginner	E_b	6,531	627	7158
	$\neg E_b$	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)

		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

難易度	Beginner	Easy	Medium	Hard	Challenge
Fraxtil データセット					
楽曲数	90	90	90	90	90
ステップ総数	7,158	19,813	30,345	45,350	65,476
ステップ数/秒	0.669	1.838	2.802	4.175	6.015
ITG データセット					
楽曲数	133	133	133	133	120
ステップ総数	8,648	20,796	32,536	48,254	61,457
ステップ数/秒	0.627	1.504	2.349	3.485	4.923

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 フレームの中から、 $n(E_c)$ と同じ 62,971 フレームを無作為に選んだ時、その中に含まれる E_b を満たすフレーム数の期待値は、 $\sum_{k=0}^{7158} ({}_{7158}C_k \times 1062611 C_{62971-k}) / 1069769 C_{62971} \approx 421.35$ 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全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。

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言える。

の 3 点が確認され、仮説 1~3 は証明された。

つぎに、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、

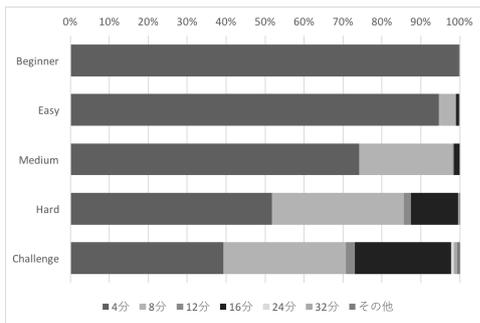


図 2 拍階層ごとのステップ存在比

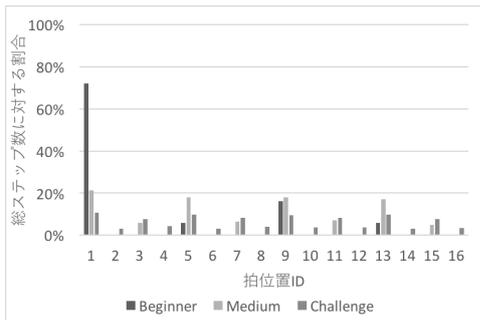


図 3 小節内のステップ存在分布

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, 人間が容易に理解出来るリズムで構成されていることが分かり, 仮説 5 は証明されたと言える.

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, 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全て 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言える. 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言える.

4. 提案手法

3 章での分析結果を基に, 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同士」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 ^{*4} 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 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 [9] 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 [10] 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 [11] 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スペクトログラムを入力, 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 10ms に設定した.

^{*4} Chollet, François et al., <https://github.com/fchollet/keras>

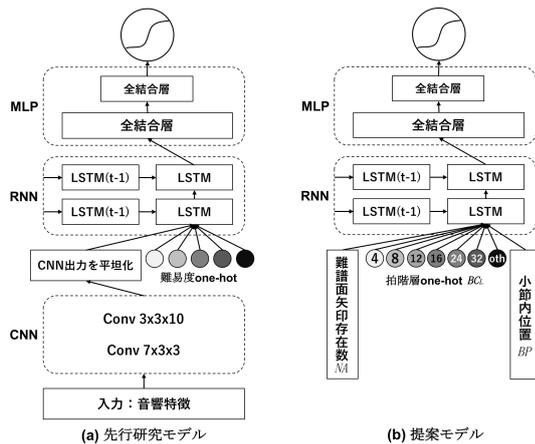


図 4 時刻決定タスクに対する既存モデルと提案モデルの比較
図 (a) は Donahue らの論文 [2] の Figure.5 を参考に作成した

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、ゲームのプレイヤー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, 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 v_t は、($NA, BC_4, BC_8, BC_{12}, BC_{16}, BC_{24}, BC_{32}, BC_{other}, BP$) 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 $NA = 1$ 、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 $BC_{16} = 1$ 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 $BP = 0.188$ がそれぞれ得られ、入力ベクトル $v_{7.58}$ 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。楽曲中の全ての時刻について、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

先行研究にならひ、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 カテゴリにランダムで振り分けた。実験では、時刻決定モデルと向き選択モデルそれぞれについて、評価した。

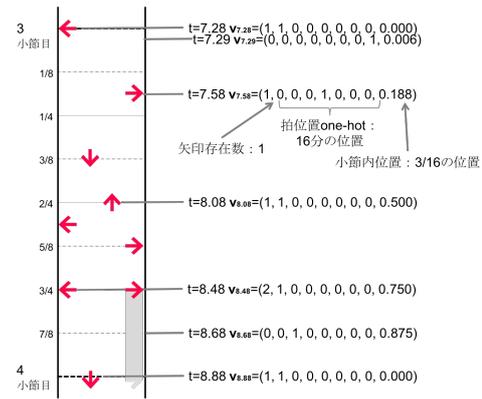


図 5 譜面データから得る特徴量

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。

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 -score^c とする。ステップ単位で算出したマイクロ F 値は、 F -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。ただし、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 種類の難易度全てについての 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、楽曲毎の適合率平均が 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。これは、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が要因と考えられる。この問題については、今後、出力された時系列からの極大値検出アルゴリズムを改良し、検出する極大値の個数を調整する。

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た場合の生成譜

表 5 学習モデル評価実験結果

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	(全て)	DDC	0.681	0.756
ITG	(全て)	DDC	0.697	0.721

面について統計をとった．データセット内の全ての譜面の統計結果と比較することで，生成された譜面がデータセットの性質を反映しているかを確認した．3章で述べた仮説4~6に基づき，「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」の3つの観点を用意した．

5.1.2.1 1秒あたりのステップ数

楽曲 m の1秒あたりステップ数 D_m は下式で算出する．

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

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，下式で算出した．

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

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，元となるデータセットについては学習データ・検証データ・テストデータ全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，元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^{*5}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 及び D の値を得た．また，参考手法は，元データセット及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 及び 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]の通り難易度が高いほど参考手法の精度が上がる事が確認された．提案手法で生成したMedium譜面は， E_m 及び D の値が元データセットよりも高く，5.1.1節で述べた通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} <http://deepx.ucsd.edu/ddc> [最終閲覧日：2017年11月29日]

表 6 1秒あたりのステップ数

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

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，提案手法と元となった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．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節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，全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ステップ生起確

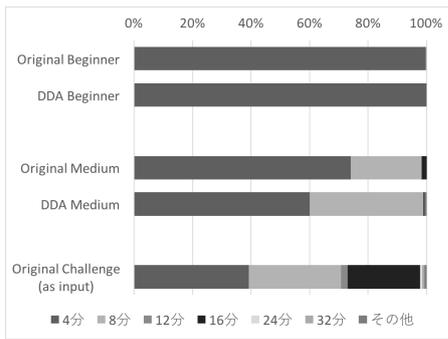


図 6 生成譜面および元データセットの拍階層ごとのステップ存在比

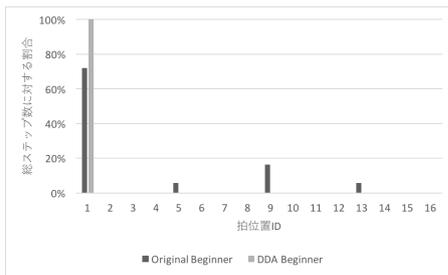


図 7 生成譜面および元データセットの小節内位置分布 (Beginner 難易度)

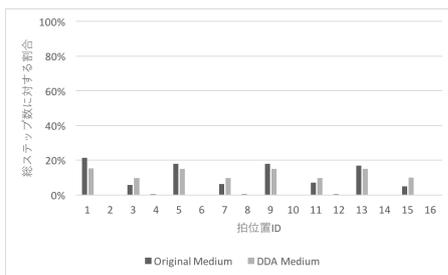


図 8 生成譜面および元データセットの小節内位置分布 (Medium 難易度)

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された。このような少数データについての対応は今後の課題としたい。

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

表 7 1-gram 出現頻度上位 7 件

Step	Original						Generated	
	All	Beginner		Easy		Medium	Beginner	Medium
Count	Step	Count	Step	Count	Step	Count	Step	Count
76,275		4,205		9,352		14,339	372	2,280
73,910		4,180		9,250		13,841	345	2,273
69,192		3,607		9,217		12,967	343	2,161
68,482		3,537		8,955		12,853	327	2,077
6,653		746		1,162		1,515	13	hold 25
3,762		128		474		847	1	hold 15
hold	3,029		14	hold	277	564	1	release 13

表 8 1-gram 難易度間相関係数

		Original				
		All	Beginner	Easy	Medium	Hard
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
	Medium	0.994	0.994	0.997	0.993	0.990

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*6 それぞれの出現頻度を集計し 1-gram とし, 連続する 2 ステップの組み合わせについては 2-gram として集計した。

表 7 に, 元となる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 種類の難易度全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*7 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, データセット内の全ての難易度と 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

*6 (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。

*7 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 9 2-gram 出現頻度上位 7 件

Original						Generated					
All		Beginner		Easy		Medium		Beginner		Medium	
1	2	Count	1	2	Count	1	2	Count	1	2	Count
29,481		1,613		3,277		5,524		121		738	
29,180		1,587		3,265		5,511		115		697	
22,445		1,318		3,224		4,227		110		695	
22,411		1,316		3,169		4,209		94		663	
18,443		812		2,179		3,364		90		600	
17,753		810		2,147		3,230		87		561	
17,643		805		2,115		3,197		87		554	

表 10 2-gram 難易度間相関係数

		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	1.000
Generated	Beginner	0.922	0.972	0.965	0.927	0.904	0.896
	Medium	0.966	0.976	<u>0.987</u>	0.968	0.954	0.951

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み合わせの相関係数は、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–Generated 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言えます、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 等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. 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概ね良い性能を示した。今後は入出力の難易度組み合わせを変更してモデルを学習させ、より多様な入出力に対応させる。また、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、楽曲から任意の難易度の譜面を生成することも可能になると考える。この 2 手法をつなぐ技術についても今後検討する。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元に、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。

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, pp. 24–29 (2007).
- [2] Donahue, C., Lipton, Z. C. and McAuley, J.: Dance Dance Convolution, *Proceedings of the 34th International Conference on Machine Learning* (2017).
- [3] 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).
- [4] 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), No. 21, pp. 1–4 (2017).
- [5] Liu, I. and Ramakrishnan, B.: Bach in 2014: Music Composition with Recurrent Neural Network, *CoRR*, Vol. abs/1412.3191 (online), available from <http://arxiv.org/abs/1412.3191> (2014).
- [6] Yang, L., Chou, S. and Yang, Y.: MidiNet: A Convolutional Generative Adversarial Network for Symbolic-Domain Music Generation, *Proc. of ISMIR 2017*, pp. 324–331 (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 Transactions on Computational Intelligence and AI in Games*, Vol. 2, No. 1, pp. 54–67 (2010).
- [9] Schluter, J. and Bock, S.: Improved musical onset detection with Convolutional Neural Networks, *Proc. of ICASSP 2014*, pp. 6979–6983 (2014).
- [10] Kim, Y., Jernite, Y., Sontag, D. and Rush, A. M.: Character-Aware Neural Language Models, *Proc. of AAAI 2016*, pp. 2741–2749 (2016).
- [11] Hochreiter, S. and Schmidhuber, J.: Long short-term memory, *Neural computation*, Vol. 9, No. 8, pp. 1735–1780 (1997).