

# 博士論文

音楽ゲーム譜面の分析と生成に基づく  
音楽と身体の相互認知メカニズムの究明

**A Study of Reciprocal Perception  
Mechanisms of Music and  
Body Action Based on Analyses and  
Generations of Rhythm-based Video  
Game Charts**

2022年3月

立命館大学大学院 情報理工学研究科  
情報理工学専攻 博士課程後期課程

辻野 雄大

立命館大学審査博士論文

音楽ゲーム譜面の分析と生成に基づく  
音楽と身体の相互認知メカニズムの究明

**A Study of Reciprocal Perception  
Mechanisms of Music and  
Body Action Based on Analyses and  
Generations of Rhythm-based Video  
Game Charts**

2022年3月

March 2022

立命館大学大学院 情報理工学研究科  
情報理工学専攻 博士課程後期課程

Doctoral Program in

Advanced Information Science and Engineering

Graduate School of Information Science and Engineering

Ritsumeikan University

辻野 雄大

TSUJINO Yudai

研究指導教員 : 山下 洋一

Supervisor : Professor YAMASHITA Yoichi

# あらまし

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。音楽ゲームのプレイヤーは、再生される音楽と、ゲーム譜面と呼ばれる指示符系列にしたがって、何らかの身体動作を行うことでゲームを楽しむ。ゲーム譜面は、対象となる音楽を、ゲーム制作者が指示符というシンボルで表現した制作物である。プレイヤーが行う身体動作も、ダンスや楽器の演奏といった音楽的表現活動を模したものである。プレイヤーは、音楽ゲームの上達にしたがって、より複雑なゲーム譜面を理解し、より高度な身体動作を実行できるようになる。これは、プレイヤーが音楽ゲームを通して、音楽をより詳細に認知できるようになり、音楽的表現活動に関わる動作を習得していると言える。

本論文では、音楽の一側面を切り取り人間に提供しているゲーム譜面を仲介することによって、「プレイヤーがゲームを通してどのように音楽を認識しているのか」という音楽的認知と、「ゲームが要求する身体動作をプレイヤーはどのように習得するのか」という身体的認知の2つの認知メカニズムを、数理的に究明する。音楽ゲームの難易度デザインを分析することで、簡単な譜面が4分の表打ちを主体とする少数の指示符で構成される一方で、難しい譜面では8分以上のリズムを含む多数の指示符で構成されることを解明した。また、音楽ゲーム譜面の面白さの特性に影響する特徴量を定め、k-means法によるクラスタリングを実施することで、指示符の配置数、特殊な動作の頻度、楽曲のテンポなどに共通点が見られる譜面がクラスタを形成することを確認した。身体動作に関する分析として、隠れマルコフモデルを用いて、ある譜面に対するプレイヤーの動作の自動推定を実現した。

究明結果の応用として、プレイヤー個人に合わせた音楽ゲーム譜面を、任意の楽曲に対して自動生成するシステムを構築する。音楽データからゲーム譜面を自動で生成する試みは存在するが、難易度を正しく調整できておらず、特徴のない譜面が生成されるという課題が存在する。究明結果を応用し、人間の認知を反映させた特徴あるゲーム譜面を生成する方法を提案する。同一楽曲に対する難易度が異なる譜面間の関係性に注目し、難しい譜面から簡単な譜面に難易度を調整する学習モデルを構築した。このモデルは、簡単な譜面に対する指示符配置タイミングの推定において、既存手法のF-scoreをおおよそ1.8倍上回った。また、特性が類似した譜面のクラスタごとに、音楽と譜面の関係を学習させることによって、多様な面白さに対応した特徴的な譜面を生成できることを確認した。

# Abstract

The rhythm-based video game is one of the typical popular games. Players perform some actions corresponding to the music and the displayed chart. The chart would be considered as a result of the symbolization of the music. The required actions are the imitation of musical expressions, such as dancing or playing instruments. Therefore, the rhythm-based video game serves one of the aspects of music. Playing rhythm-based video game make players' musical recognition and expression skills improve.

In this paper, I figure out the perception mechanisms of music and body action bypassing the game chart. By analyzing the difficulty level of the rhythm-based video game, I find out that easy charts require few actions with easy rhythm while hard charts require many actions with a complex rhythm. I design features concerning factors that may characterize the charts. As conducting the k-means clustering with the features, the characteristically similar charts structure the clusters. To study the body action of players, I propose a method to estimate the player's movements for the given chart based on Hidden Markov Model.

I propose an automatic chart generation system that serves personalized charts for a given audio track as an application. It has been reported that the charts generated by the existing method do not have enough quality and have no characteristics. The proposed method automatically generates a chart that reflects the perception mechanisms. I propose a deep learning model that learns the relation between the lower and higher difficulty charts for the same song. For the task of estimating the timing to action for easy charts, this method outperforms the existing method with about 1.8 times higher F-score. By learning the relationships between songs and charts in each cluster, it was confirmed that the model can automatically generate the charts with the characteristics corresponding to the learning cluster.

# 目次

<b>第 1 章</b>	<b>序論</b>	<b>1</b>
1.1	はじめに . . . . .	1
1.2	研究目標 . . . . .	1
1.2.1	音楽と身体動作の認知メカニズムの究明 . . . . .	1
1.2.2	音楽ゲーム譜面の自動生成 . . . . .	2
1.3	研究対象および用語の定義 . . . . .	2
<b>第 2 章</b>	<b>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</b>	<b>6</b>
2.1	はじめに . . . . .	6
2.2	関連研究 . . . . .	6
2.3	難易度が異なるゲーム譜面間の関係性についての分析 . . . . .	7
2.3.1	仮説の検証 . . . . .	8
2.4	提案手法 . . . . .	10
2.4.1	参考手法: <i>Dance Dance Convolution</i> . . . . .	11
2.4.2	提案手法: <i>Dance Dance Adaptation</i> . . . . .	12
2.5	評価実験 . . . . .	13
2.5.1	時刻決定モデルの評価 . . . . .	13
2.5.2	向き選択モデルの評価 . . . . .	18
2.6	提案手法の拡張性に関する考察 . . . . .	20
2.7	おわりに . . . . .	22
<b>第 3 章</b>	<b>ダンスゲーム譜面の特性分析とクラスタリングに基づく特徴的な譜面の自動生成</b>	<b>23</b>
3.1	はじめに . . . . .	23
3.2	ダンスゲーム譜面の多様性 . . . . .	24
3.3	ダンスゲーム譜面から得る特徴量の提案と分析 . . . . .	26
3.3.1	譜面特徴量 . . . . .	27
3.3.2	楽曲特徴量 . . . . .	29
3.3.3	特徴量に基づく譜面の分析と考察 . . . . .	30
3.4	クラスタリングに基づく譜面の自動生成 . . . . .	32

3.4.1	前処理：外れ値を持つ譜面の検出	32
3.4.2	クラスタリング	33
3.4.3	譜面の学習と自動生成	34
3.4.4	生成譜面の評価	35
3.5	おわりに	36
<b>第4章</b>	<b>プレイヤーのゲーム習熟度と嗜好を反映したダンスゲーム譜面に対する動作の推定</b>	<b>39</b>
4.1	はじめに	39
4.2	用語定義	41
4.3	有向グラフによるプレイヤーの姿勢および動作の表現	42
4.3.1	ノード：姿勢	42
4.3.2	エッジ：動作	45
4.4	隠れマルコフモデルを用いたプレイヤー動作の推定	47
4.5	推定結果例	48
4.5.1	条件設定	48
4.5.2	考察	49
4.6	おわりに	51
<b>第5章</b>	<b>結論</b>	<b>53</b>
5.1	本論文のまとめ	53
5.1.1	音楽と身体動作の認知メカニズムの究明	53
5.1.2	音楽ゲーム譜面の自動生成	54
5.2	今後の展望	54
	<b>謝辞</b>	<b>56</b>
	<b>研究業績</b>	<b>57</b>
	<b>参考文献</b>	<b>59</b>

# 目 次

1.1	音楽・ゲーム譜面・人間の認知の相互関係 . . . . .	2
1.2	プレイヤーが操作するパネルを天井方向から見た模式図 . . . . .	3
1.3	ダンスゲームにおけるゲーム画面の例 . . . . .	3
1.4	拍階層・拍位置の概念 . . . . .	5
2.1	拍階層ごとのステップ存在比 . . . . .	10
2.2	小節内のステップ存在分布 . . . . .	11
2.3	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. . . . .	13
2.4	譜面データから得る特徴量 . . . . .	14
2.5	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 . . . . .	18
2.6	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) . . . . .	19
2.7	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) . . . . .	19
3.1	“ <i>Disconnected</i> ” (Hard) の一部 . . . . .	25
3.2	“ <i>Sweet World</i> ” (Hard) の一部 . . . . .	25
3.3	“ <i>Remember December</i> ” (Challenge) の一部 . . . . .	26
3.4	“ <i>Bend Your Mind</i> ” (Hard) の一部 . . . . .	26
3.5	1秒あたりステップ数の集計例 . . . . .	27
3.6	$n_{\mu}$ の分布を示す箱ひげ図 . . . . .	30
3.7	$t_{\mu}, t_l, t_s, t_f$ の分布を示す箱ひげ図 . . . . .	31
3.8	楽曲 “ <i>Walking on Fire</i> ” を各モデルに入力し, 出力された譜面の一部 . . . . .	34
3.9	楽曲 “ <i>Hillbilly Hardcore</i> ” を各モデルに入力し, 出力された譜面の一部 . . . . .	35
3.10	楽曲 “ <i>Bubble Dancer</i> ” を各モデルに入力し, 出力された譜面の一部 . . . . .	36
3.11	楽曲 “ <i>Twilight</i> ” を各モデルに入力し, 出力された譜面の一部 . . . . .	37
4.1	本章で用いる座標系の模式図 . . . . .	41
4.2	プレイヤーの姿勢と動作を表現する有向グラフ (抜粋) . . . . .	43
4.3	推定対象譜面と姿勢ノードの対応関係 (抜粋) . . . . .	48
4.4	推定対象とした <i>Bumble Bee</i> (Medium) の一部 . . . . .	50
4.5	<i>Bumble Bee</i> (Medium) に対する各エージェント推定結果 (抜粋) . . . . .	51

# 表 目 次

1.1	データセットの情報	4
2.1	Challenge 譜面と Beginner 譜面の関連 (Fraxtil データセット)	9
2.2	Challenge 譜面と Beginner 譜面の関連 (ITG データセット)	9
2.3	各データセットの難易度別情報	9
2.4	時刻決定モデル評価実験結果	15
2.5	1 秒あたりのステップ数	17
2.6	1-gram 出現頻度上位 7 件	20
2.7	1-gram 難易度間相関係数	20
2.8	2-gram 出現頻度上位 7 件	21
2.9	2-gram 難易度間相関係数	21
3.1	出現頻度上位 3 種類の 3-gram	29
3.2	クラスタリング結果	32
3.3	各モデルの $F$ -score	35
3.4	各モデルが生成した譜面の $n_{-\mu}, j_{-r}, j_{-d}$ の平均値	36
4.1	ノード評価値に関わるプレイヤー固有パラメーター一覧	43
4.2	エッジ遷移確率に関わるプレイヤー固有パラメーター一覧	45
4.3	4 種類のエージェントおよびそれぞれのプレイヤー固有パラメータの値	49
4.4	<i>Bumble Bee</i> (Medium) に対する各エージェント推定結果の代表値	49



# 第 1 章

## 序論

### 1.1 はじめに

今日、ビデオゲームは単なるエンタテインメントとしてのみではなく、運動不足解消や教育などの社会的問題に対する解決策としても注目を集めている。健康増進や教育への応用が期待されるゲームジャンルの1つとして、音楽ゲームが挙げられる。音楽ゲームの多くでは、楽曲と画面に表示される指示符に沿って、ゲームを遊ぶ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。プレイヤーが行う身体動作は、ダンスや楽器の演奏などの音楽的表現活動を模したものであることが多い。楽曲と紐づいた指示符の配置パターンをゲーム譜面または譜面と呼ぶ。音楽ゲームを活用し、音楽的表現活動を疑似的に楽しませつつユーザの運動を促進しようとする試みが存在する。例として、足でパネルを踏む動作によってダンスを疑似体験する音楽ゲーム *Dance Dance Revolution*<sup>®</sup> (DDR) を、体育授業に取り入れた事例 [1] が存在する。DDR に代表される、ダンスを模した身体動作を行うゲームは、ダンスゲームという音楽ゲームのサブジャンルに分類される。

身体運動を伴うゲームを認知症治療に利用した試み [2] では、患者個人に合わせてゲームの難易度を調整することで認知機能改善効果を得たと述べられている。したがって、音楽ゲームを運動促進に活用するためにも、プレイヤー個人に合わせた難易度のゲーム譜面を提供する仕組みが必要といえる。しかし、音楽ゲームの難易度に影響する要素を解明する研究はなされていない。ゲーム譜面の認知には、リズムに対する音楽的認知や、動作に対する身体的認知など、ゲーム外の認知能力が要求される。個人に合わせたゲーム譜面を提供するためには、プレイヤーの音楽・身体的認知メカニズムと、それらの認知と対応関係にある音楽ゲーム内の要素を分析し、モデル化することが重要と考えた。

### 1.2 研究目標

本研究の目標は、音楽ゲームの分析と譜面自動生成を通して、人間の音楽に対する認知と身体運動の相互関係を究明することである。これは、1.2.1 節と 1.2.2 節に示す 2 つの段階に分けられ、それぞれの段階は相互に影響する。

#### 1.2.1 音楽と身体動作の認知メカニズムの究明

音楽ゲームを分析し、人間が音楽的知識や音楽的表現活動に関わる動作を認知し習得する過程を究明する。図 1.1 のように、「音楽とゲーム譜面」「ゲーム譜面と人間の認知」の関係をそれぞれ分析す

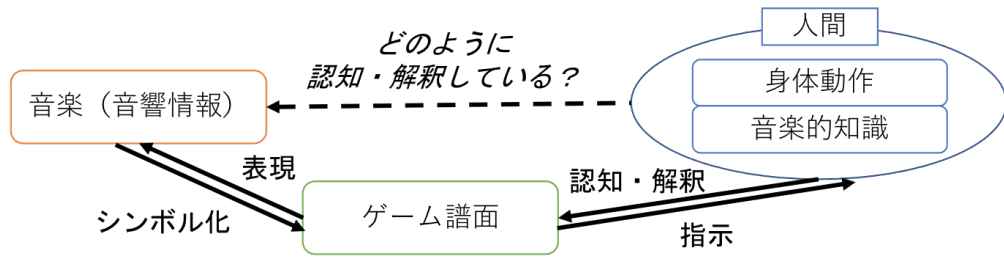


図 1.1: 音楽・ゲーム譜面・人間の認知の相互関係

ることが、「音楽と人間の認知」の関係を究明することに繋がると考えられる。ゲーム譜面は、音楽ゲームの制作者が音楽を聴取し、指示符というシンボルで表現した制作物である。ゲームのプレイヤーは、ゲーム譜面を認知し解釈することで、指示された動作を行う。それぞれの関係を計算機上で分析し再現することで、「人間は音楽をどのように認知しているか」という問いを、ゲーム譜面を介して数理的に究明することが可能と考える。

### 1.2.2 音楽ゲーム譜面の自動生成

1.2.1 節で得た知見の応用として、プレイヤー個人に合わせた音楽ゲーム譜面を、任意の楽曲に対して自動生成するシステムを構築する。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、ゲームの制作者にとっても、膨大な数の楽曲に対して譜面を作成していくことは、決して容易な作業ではない。この問題を解決する手段として、ゲーム譜面を自動で生成するシステムを構築することが考えられる。ゲーム譜面の自動生成を試みた研究として、Donahue らによる *Dance Dance Convolution (DDC)* [3] が挙げられる。DDC は、音楽情報処理と深層学習の技術を用いて、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ものである。しかしこの手法は、音楽とゲーム譜面の組を愚直に学習させたものであり、ゲーム譜面の特性やプレイヤーの認知特性をとらえたものではなかった。1.2.1 節で究明した、人間の音楽的・身体的認知を反映させたゲーム譜面を自動で生成する方法を模索する。生成された譜面に対する評価や検討を通して、1.2.1 節で得た知見の妥当性を確認する。

## 1.3 研究対象および用語の定義

本論文では、音楽ゲームの中でも特にダンスゲームを対象とする。[1] などで取り上げられていることに加え、著者がダンスゲーム制作メーカ主催の大会で入賞経験があるなど十分な知見を有しているためである。ただし、本論文の基本理念は、ダンスゲーム以外の音楽ゲームにも適用できるものと考えている。

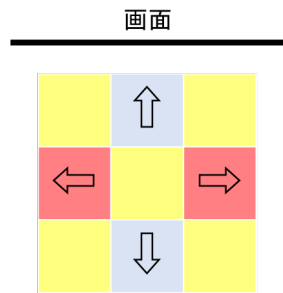


図 1.2: プレイヤが操作するパネルを天井方向から見た模式図

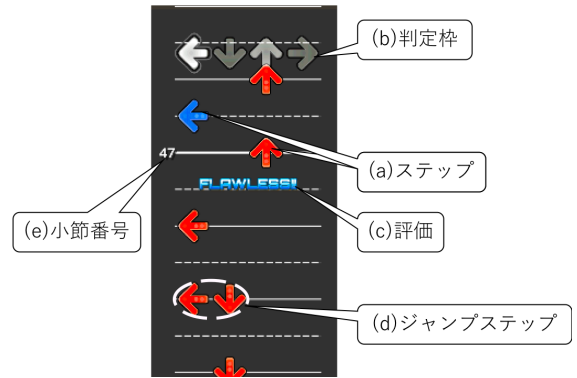


図 1.3: ダンスゲームにおけるゲーム画面の例。楽曲に合わせて (a) ステップが画面下部から上部へ移動し、(b) 判定枠とステップが重なったタイミングでプレイヤーが対応するパネルを踏む。(c) 評価は動作を正しいリズムで行えたかどうかを表す。ステップの中には、2 箇所以上のパネルを同時に踏むことを指示する (d) ジャンプステップも存在する。(e) 小節番号は、その時刻が楽曲先頭から数えて何小節目にあたるかを示す。

プレイヤーが操作に用いるパネルが存在する平面領域を、本論文ではステージと呼称する。図 1.2 にステージの模式図を示す。プレイヤーはステージ上に立ち、前方の画面に表示されるゲーム譜面に従って、パネルを足で踏むことによってダンスゲームを遊ぶ。ステージは、正方形の平面領域を  $3 \times 3$  に等分割したマス目状になっている。画面が存在する方向を「上」とした時、上端の 3 マスのうち中央にあたる 1 マスが  $\uparrow$  パネルとなっている。同様に、左端 3 マスのうち中央のマスが  $\leftarrow$  パネル、右端 3 マスのうち中央のマスが  $\rightarrow$  パネル、下端 3 マスのうち中央のマスが  $\downarrow$  パネルとなっている。プレイヤーは、左足または右足のどちらかを用いてパネルを踏む。また、プレイヤーが 1 つの足で同時に踏むことができるパネルの数は 1 つまでとする。

図 1.3 に、本論文で対象とするダンスゲームの画面例を示す。ダンスゲームにおける指示符は、図 1.3 内 (a) のように矢印の形で示され、ステップと呼称される。ステップは画面下部から上部へと移動し、(b) 判定枠とステップが重なったタイミングでプレイヤーが対応するパネルを踏む。本論文では、パネルを踏み始めた瞬間のことをパネルオンセットの発生と表現する。(c) 評価は、パネルオンセットの発生タイミングと、ステップと判定枠が重なるタイミングが、どの程度一致しているかを示す。(d) のように、同一の行に複数の矢印が存在するステップは、両足を用いて同一タイミングで複数のパネルを踏むことを指示している。このようなステップをジャンプステップと呼ぶ。(e) 小節番号は、その時刻が楽曲先頭から数えて何小節目にあたるかを示す。ダンスゲームにおけるゲーム譜面は、表示

表 1.1: データセットの情報 ([3] を参考に作成)

データセット	Fraxtil	ITG
作者の人数	1	8
パックの数	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

するステップのタイミングおよび種類を，対象の楽曲に基づいて定義したものである．同じ譜面を繰り返し遊ぶとき，ステップの内容は変化しない．

ステップは，プレイヤーがどちらの足を用いてパネルを踏むかについては指示しておらず，どのような姿勢でパネルを踏むかはプレイヤーの解釈に委ねられている．例えば，図 1.3 中で判定棒に最も近い↑のステップは，プレイヤーにとって前方に存在する↑のパネルを踏むことを指示しているが，これを右足で踏むか左足で踏むかは定められていない．また，ステップによって指示されていないパネルについては，足の有無は問われない．例えば，前述の↑パネルを踏むことを指示するステップにおいて，左足で↑パネルを踏んだ場合，右足はどのパネルを踏んでいる状態であっても良く<sup>1</sup>，どのパネルも踏んでいない状態でも良い．どのパネルも踏んでいない状態とは，足がステージに接地していない状態のほか，ステージ内のパネルが存在しない部分に足が接地している状態も含める．

先行研究 [3] にならって，Fraxtil データセット<sup>2</sup>と ITG データセット<sup>3</sup>を使用する．表 1.1 にデータセットの情報を示す．これらのデータセットには，オープンソースのダンスゲーム *Stepmania*<sup>4</sup> で遊べるように作られた譜面データが含まれている．1 曲につき 4~5 種類の譜面が収録されており，難易度が低い順に Beginner, Easy, Medium, Hard, Challenge というラベルが付与されている．本論文では，“曲名”（難易度ラベル名）という書式で，データセット内に含まれる特定の譜面を指すものとする．

本論文では，動作を行うタイミングが楽曲聴取上で知覚しやすいタイミングであるかどうかを，拍階層表現を用いて表現する．小節を  $n$  等分（ただし  $n \in \mathbb{N}$  かつ  $n \geq 4$ ）することで表現出来る時刻集合を  $n$  分層と定義する．あるステップが存在する時刻において，ある  $n (\geq 4)$  が以下の 2 つの条件をともに満たすとき，そのステップは  $n$  分層に存在するものとし， $n$  分（のステップ）と表現する．

<sup>1</sup>ただし，左足で↑パネルを踏む以前から右足で↑パネルを踏み続けていた場合は，パネルオンセットが発生しないためステップの指示を満たさない．

<sup>2</sup>Fraxtil's Simfile, <https://fra.xtil.net/simfiles/>（最終閲覧日：2021 年 12 月 22 日）より「Tsunami III」「Fraxtil's Arrow Arrangements」「Fraxtil's Beast Beats」の 3 パッケージ

<sup>3</sup>“In The Groove” <http://east.stepmania-online.com/In%20The%20Groove%201.zip> および “In The Groove 2” <http://east.stepmania-online.com/In%20The%20Groove%202.zip> [最終閲覧日：2021 年 12 月 22 日]

<sup>4</sup>News - Stepmania, <https://www.stepmania.com>, [最終閲覧日：2021 年 12 月 22 日]

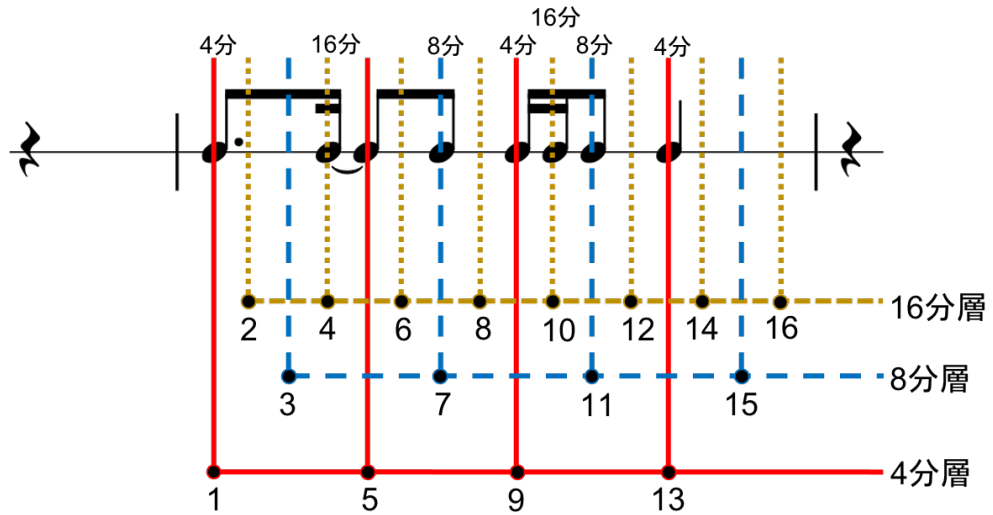


図 1.4: 拍階層・拍位置の概念. 4分層に属する音符は, 8分層や16分層にも属するが, その中で最下層である4分層に存在するものとする. 各層の下に書かれた1~16の数字は, 小節内での拍位置を示す. 本論文ではこれを拍位置 ID とする.

条件 A ステップが存在する時刻が  $n$  分層の時刻集合に含まれている.

条件 B  $n$  が条件 A を満たす最小の整数である.

*Stepmania* においては, ステップが存在する拍階層によって, 図 1.3 のように異なる色でステップが表示される. 図 1.4 に, 実際の楽器の楽譜と拍階層表現の対応付けを示す. ダンスゲームにおける「ステップに合わせてパネルを踏む」という動作は, 楽器の演奏における「音符に合わせて発音を開始する」という動作に対応する. ステップが存在している階層の  $n$  が大きいほど, そのステップは複雑なリズムで構成されており, 知覚しにくいと言える. 反対に,  $n$  が小さい層のステップが連続しているとき, それらのステップは知覚しやすいリズムであると言える.

## 第 2 章

# 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

### 2.1 はじめに

Donahue らによる DDC [3] 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制作可能である。しかしながら、低難易度の譜面を制作するためには、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、音楽ゲームを運動促進に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。

本章では、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、高難易度譜面と低難易度譜面間との関係、および高難易度譜面と比較した低難易度譜面の音楽的特徴について分析を行う。分析結果に基づいた、難易度間と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。

### 2.2 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既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 [4] や、CNN に基づく GAN を用いた Yang らの手法 [5] 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 [6]。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 [7]。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、本論文で対象とする音楽ゲームにおいては、1つの曲に対して難度の異なる複数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捉えて、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 [8]。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*<sup>®</sup> 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 [9]。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 [10]。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。

## 2.3 難易度が異なる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。難易度が異なる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

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：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 2.3.1 節において、これらの仮説に対する検証を行う。

### 2.3.1 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、

- Beginner 譜面でステップが存在する時刻に、Challenge 譜面でもステップが存在する確率  $P(E_c|E_b)$
- Challenge 譜面でステップが存在しない時刻に、Beginner 譜面でもステップが存在しない確率  $P(\neg E_b|\neg E_c)$

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 [3] 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 2.1 と表 2.2 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  $P(E_c|E_b) \approx 0.912$ 、 $P(\neg E_b|\neg E_c) \approx 0.999$ 、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、 $\sum_{k=0}^{7158} ({}_{7158}C_k \times 1062611 {}_{62971}C_{62971-k}) / 1069769 {}_{62971}C_{62971} \approx 421.35$  となる。一方、表 2.1 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。



表 2.1: 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	7,158
	$\neg E_b$	56,440	1,006,171	1,062,611
合計		62,971	1,006,798	1,069,769

表 2.2: 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

の3点が確認され、仮説 1~3 は証明された。

つぎに、表 2.3 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 2.1 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、人間が容易に理解出来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

表 2.3: 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

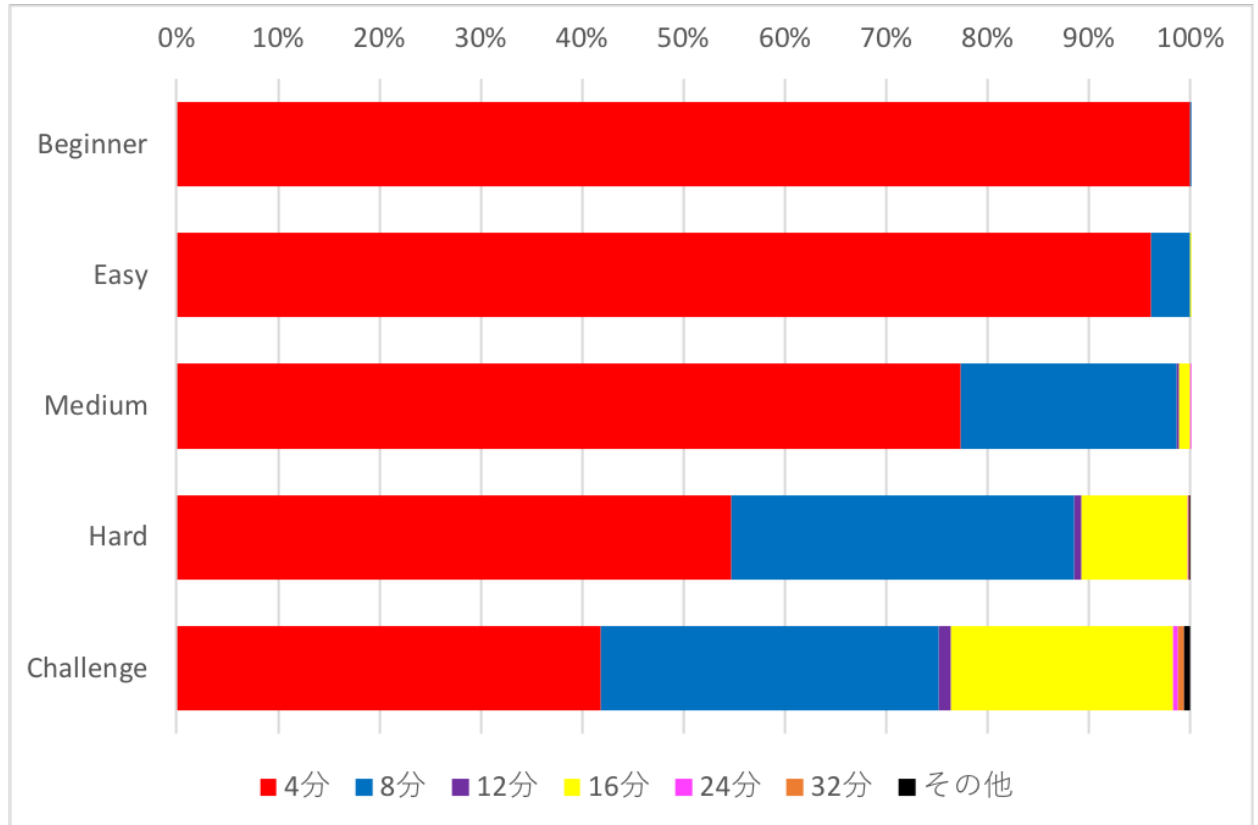


図 2.1: 拍階層ごとのステップ存在比

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 2.2 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 1.4 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。

## 2.4 提案手法

2.3 節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 DDC [3] 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。本章では、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

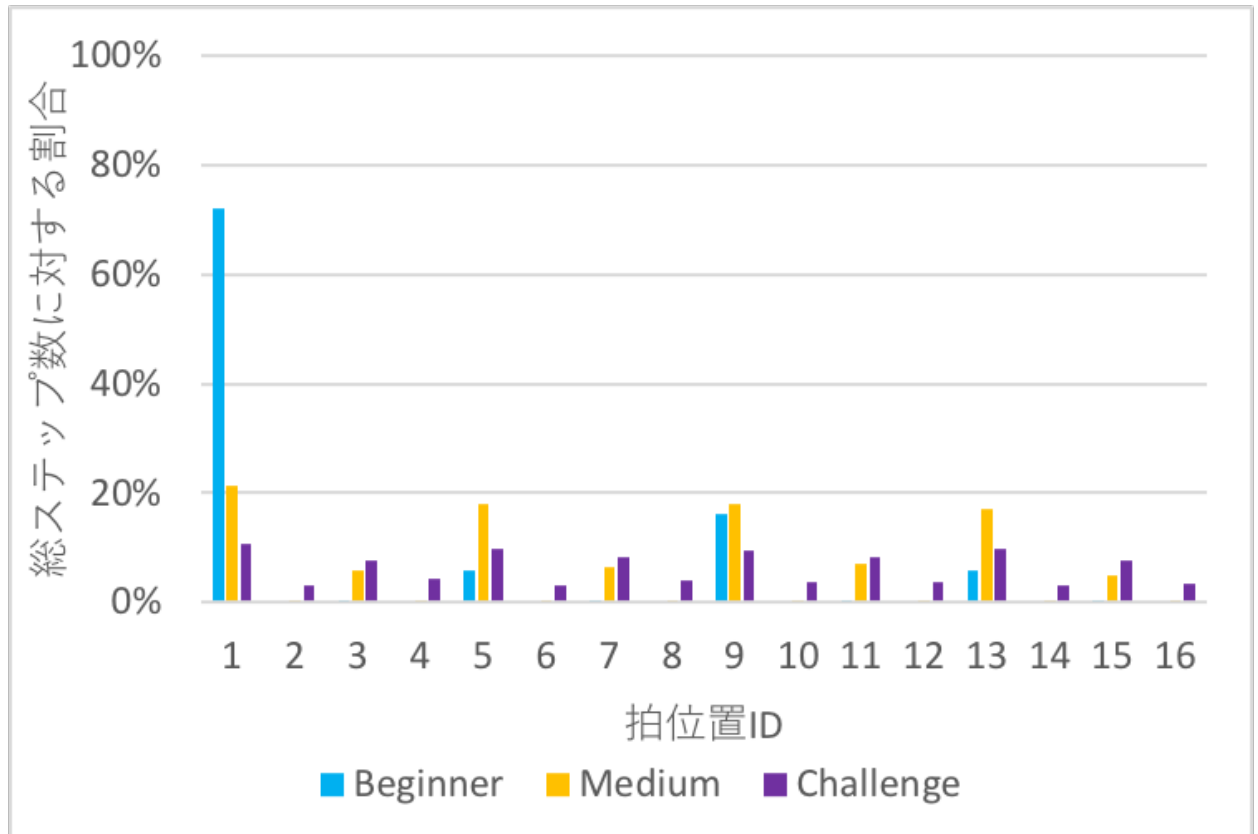


図 2.2: 小節内のステップ存在分布

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<sup>1</sup>を利用した。

#### 2.4.1 参考手法：Dance Dance Convolution

先行研究 DDC 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 [11] 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 [12] 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ステップ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 [13] 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

<sup>1</sup>Chollet, François et al., <https://github.com/fchollet/keras> [最終閲覧日：2021 年 12 月 22 日]

一種である LSTM [14] 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 [15] 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 [3] を参照されたい．

#### 2.4.2 提案手法：*Dance Dance Adaptation*

本章で提案するモデル：*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 2.3 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．

難易度が高い譜面のデータから獲得する特徴量は，2.3 節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ステップの数．非負整数値の 1 次元で表される．

*BC<sub>L</sub>*: 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 2.4 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，右のステップ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 2.3 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．

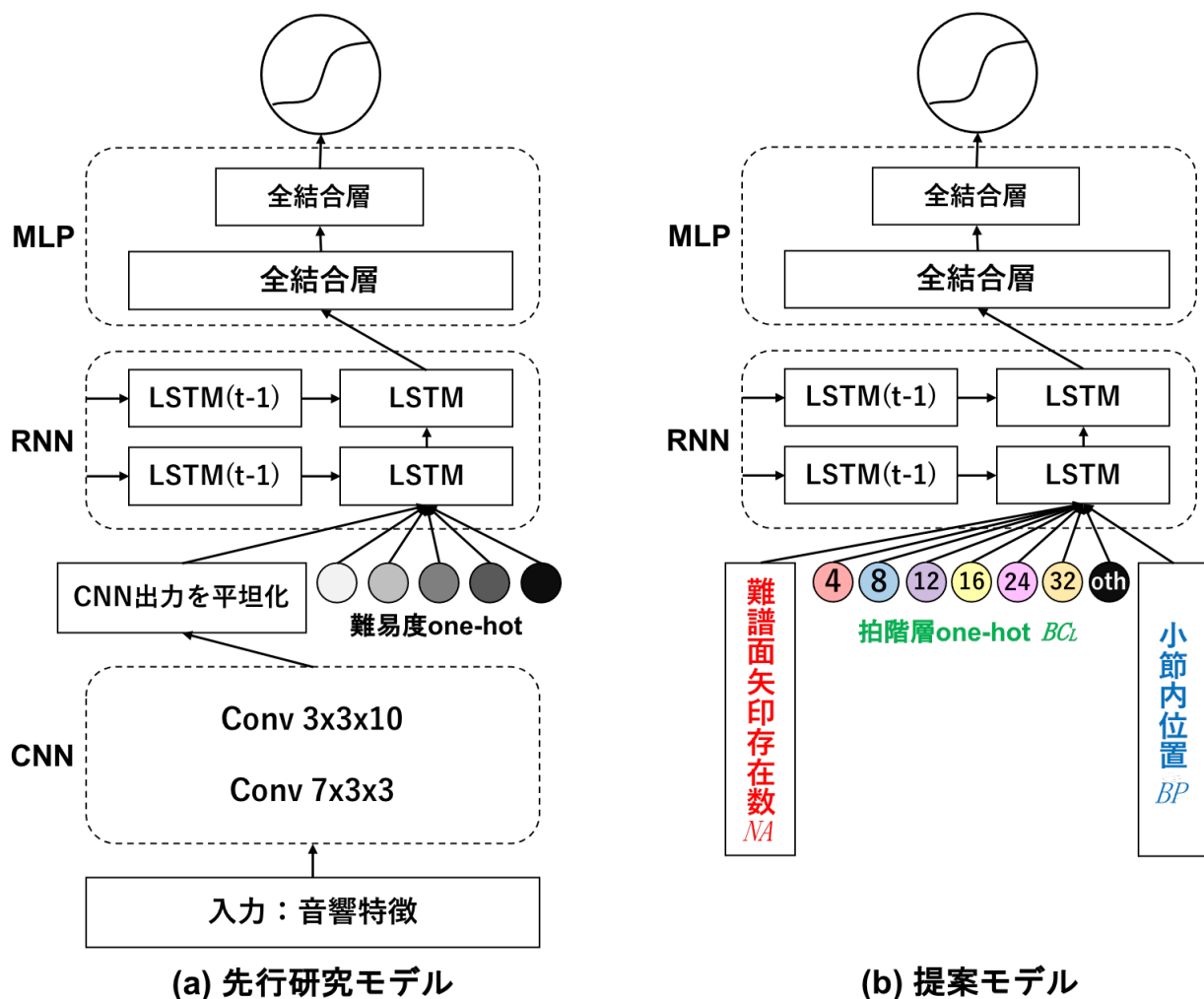


図 2.3: 時刻決定タスクに対する既存モデルと提案モデルの比較. 図 (a) は Donahue らの論文 [3] の Figure 5 を参考に作成した

## 2.5 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.

### 2.5.1 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

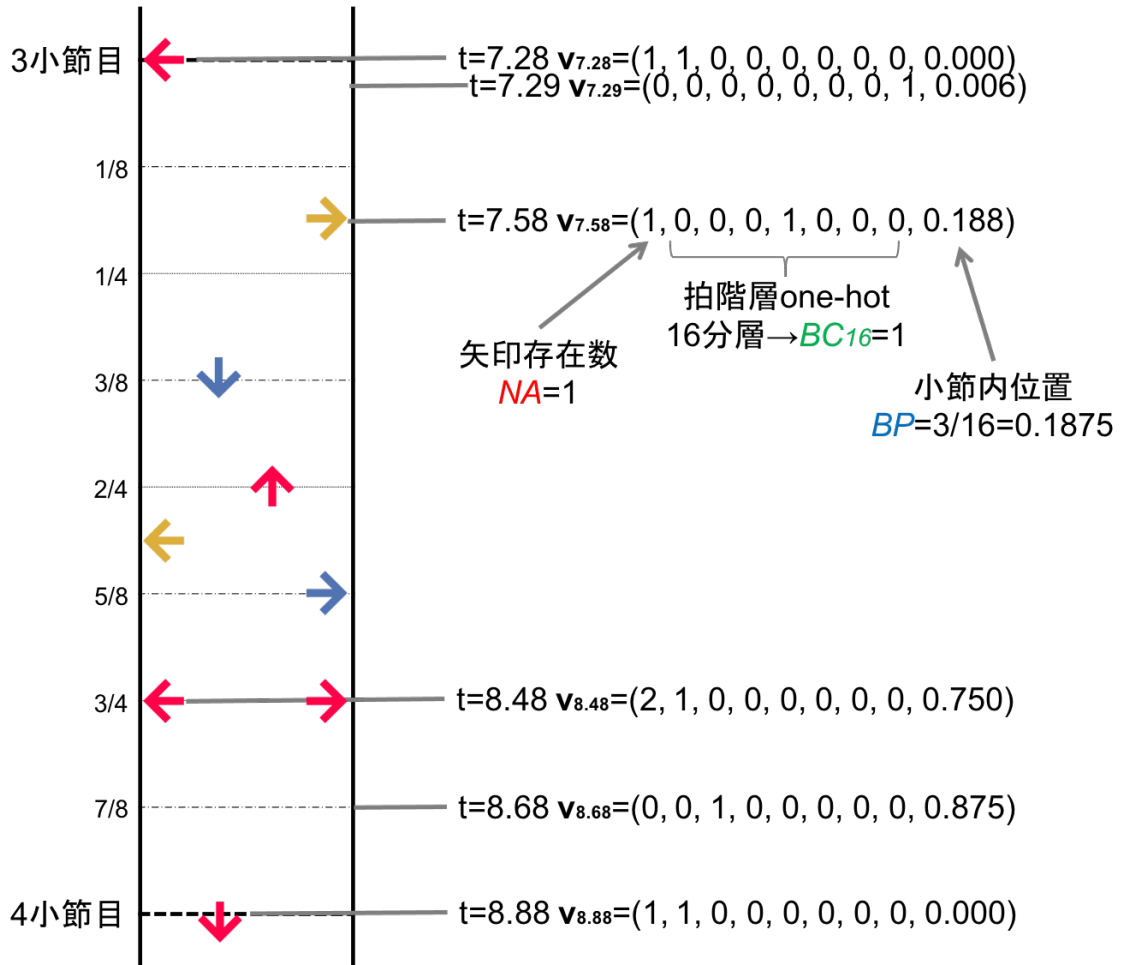


図 2.4: 譜面データから得る特徴量

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 × 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 [3] で用いられた  $F$ -score 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。これらの組み合わせにより、提案手法の有効性を総合的に評価した。

#### $F$ -score 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、 $F$ -score を算出した。楽曲ごとの適合率と再現率から得られる  $F$ -score を合計し、楽曲数で除算し平均したものを、 $F$ -score<sup>c</sup> とする。ステップ単位で算出したマイクロ  $F$ -score は、 $F$ -score<sup>m</sup> とする。表 2.4 に、提案手法と比較手法それぞれの  $F$ -score を示す。表中、DDA は提案手法の値を示し、DDC は先行研究 [3] 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、本章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

表 2.4: 時刻決定モデル評価実験結果

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

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 譜面、Challenge 譜面についての  $F$ -score<sup>c</sup> 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  $F$ -score 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<sup>c</sup> 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  $F$ -score 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<sup>c</sup> 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、 $F$ -score 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、2.5.1 節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、 $F$ -score 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ると考える。

#### 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、データセット内の全ての譜面について、2.3 節で述べた仮説 4~6 に基づき以下の統計指標を算出した。

- (a) 1 秒あたりのステップ数
- (b) ステップが存在する拍階層の構成比

(c) 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。用意した指標のうち、(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易度調整モデルへの入力として妥当であると言える。

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。

$$D_m = \frac{\text{譜面中のステップ数 (個)}}{\text{楽曲の長さ (秒)}}. \quad (2.1)$$

$D_m$  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。

$$D = \frac{\text{対象譜面集合内のステップ数の総和 (個)}}{\text{対象楽曲集合内の楽曲の長さの総和 (秒)}}, \quad (2.2)$$

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 2.5 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<sup>2</sup>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 [3] の通り難易度が高いほど参考手法の精度が上がることを確認された。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、2.5.1 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。

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 2.3.1 節で行った仮説 5 の検証と同一の方法をとった。図 2.5 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

<sup>2</sup><http://deepx.ucsd.edu/ddc> [最終閲覧日：2021年12月22日]



表 2.5: 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	<b>DDA</b>	<b>0.654</b>	<b>0.036</b>	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	<b>DDA</b>	<b>0.625</b>	<b>0.007</b>	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	<b>DDA</b>	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	<b>DDA</b>	4.006	0.758	4.021
ITG	Medium	DDC	3.283	1.938	3.248

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 2.5 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したがって 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, 2.5.1 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.

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, 2.3.1 節で行った仮説 6 の検証と同一の方法をとった.

図 2.6 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

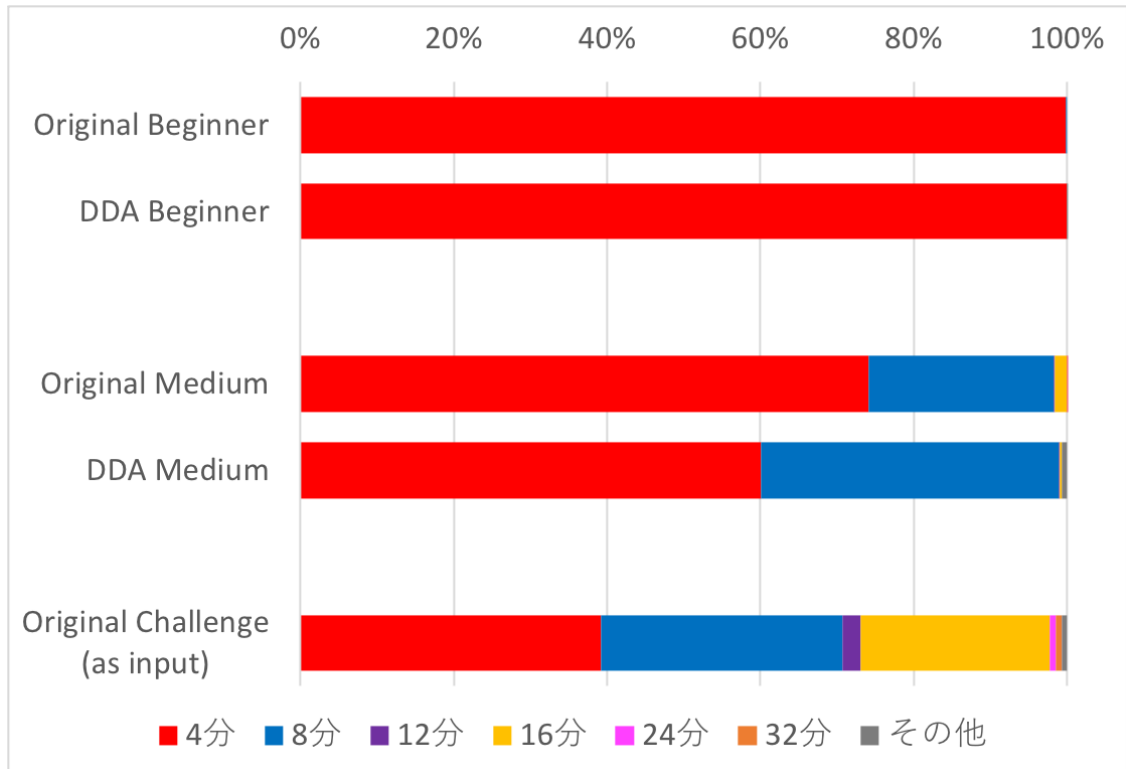


図 2.5: 生成譜面および元データセットの拍階層ごとのステップ存在比

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 2.7 に，Medium 難易度についての小節内各拍位置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。

### 2.5.2 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 [3] 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

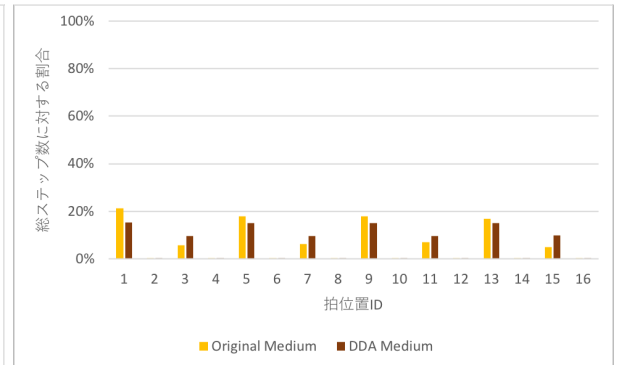
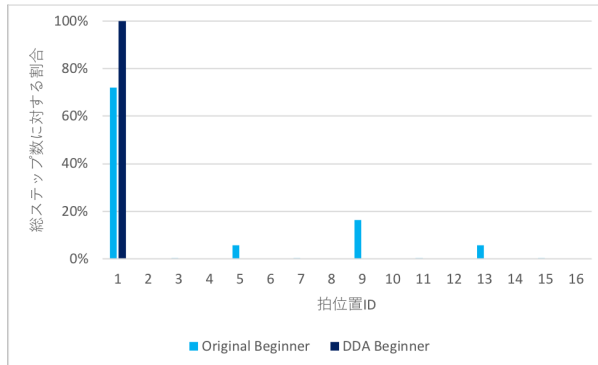


図 2.6: 生成譜面および元データセットの小節内位置分布 (Beginner 難易度) 図 2.7: 生成譜面および元データセットの小節内位置分布 (Medium 難易度)

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<sup>3</sup>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 2.6 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ジャンプステップ、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<sup>4</sup>である。表 2.7 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 2.7 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 2.8 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 2.9 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 2.7 と表 2.9 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

<sup>3</sup> (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。

<sup>4</sup> 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、該当パネルから足を離してもよいし、踏み続けてもよい。

表 2.6: 1-gram 出現頻度上位 7 件

Original								Generated			
All		Beginner		Easy		Medium		Beginner		Medium	
Step	Count	Step	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 2.7: 1-gram 難易度間相関係数

		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	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

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言え、これは今後の検討課題である。

## 2.6 提案手法の拡張性に関する考察

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

表 2.8: 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	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 2.9: 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	<b>0.972</b>	0.965	0.927	0.904	0.896
	Medium	0.966	0.976	<u>0.987</u>	0.968	0.954	0.951

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既に提案されており [16, 17], 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, 更なる応用として, 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出来る点が挙げられる。本章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 (例えば, 太鼓の達人<sup>®</sup><sup>5</sup>など) に応用可能である。

<sup>5</sup>太鼓の達人シリーズ公式ポータルサイト — 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: 2021 年 12 月 22 日]

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良い．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基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．本章の研究における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に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出来る可能性がある．

## 2.7 おわりに

本章では，同じ楽曲に対する難易度が異なるゲーム譜面間の関係，および難易度の違いによる譜面の音楽的特徴の違いを究明した．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．

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．2.5.2 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．

## 第 3 章

# ダンスゲーム譜面の特性分析と クラスタリングに基づく 特徴的な譜面の自動生成

### 3.1 はじめに

幅広い実力のプレイヤーが面白いと感じるゲームとするためには、簡単な内容から始まり、少しずつ難しくなっていくように難易度バランスを構成することが重要である [18]. 多くのダンスゲームでは、難易度の異なるたくさんの譜面を用意することで、多くのプレイヤーが面白いと感じるよう配慮されている。譜面の難易度に影響する要素は複数あるとされている。仮に、リズムの複雑さが譜面の難易度に影響する唯一の要素であるならば、同程度の難易度とされている譜面は楽曲によらず同程度に複雑なリズムを含んでいることになる。しかし、実際には同程度の難易度の譜面であってもリズムの複雑さは様々であり、動作の要求頻度やジャンプステップの頻度なども難易度に影響すると考えられる。このように、同程度の難易度の譜面であっても、プレイヤーに要求する動作の組み合わせによって、それぞれ異なる「難しさの個性」を持つと言える。この難しさの個性が、ダンスゲーム譜面の「面白さ」にも大きく影響する。同程度の難易度の譜面であっても、どの譜面をより面白いと感じるかはプレイヤーによって異なる。その事例として、ゲーム制作者によって同程度の難易度であるとラベル付けされた譜面間でプレイヤーのゲームログを比較したとき、プレイヤーによって好んでプレイしている譜面は異なることが多い。以上から、プレイヤーは譜面の特性を「難易度」という単一の基準のみではなく、「面白さ」という異なる観点でも評価していると言える。これら 2 つの観点から譜面を評価した例として、「難しいがやりがいがあり、面白い譜面である」「簡単に遊びやすく、面白い譜面である」「簡単だがつまらない譜面である」などが挙げられる。

ゲームの難易度や面白さに影響する要素を調査した研究はいくつか存在している。Wehbe らは、シンプルなアクションゲームに対して、画面スクロールの速さやジャンプ動作の複雑さなどのパラメータを変化させることで、難易度およびプレイヤー体験が変化するかを調査している [19]. Spiel らは、パズルゲーム *TETRIS* を題材にして、自明に難易度に影響するゲームの速度ではなく、ピースの選択アルゴリズムを変化させることで、難易度および面白さが変化するかを調査している [20].

一方で、音楽ゲームに関しては、人類学的な研究 [21] であるか、譜面の自動生成に関わる研究が主流であった。Donahue らの *Dance Dance Convolution (DDC)* [3] は、譜面の自動生成を試みた代表的な研究である。DDC では、データセット内の譜面全てを用いてモデルを学習させている。とこ

ろが、前述の通り譜面データセットの中には様々な特性を持った譜面が混在している。多様性のあるデータを、各データが持つ特性を明示することなく同一のモデルに学習させると、特性が混在した譜面が生成されてしまう。このような生成結果の一例として、30秒近くわたって一切の指示符が存在しない区間と、指示符が極端に多い区間が混在する譜面が出力されることを確認した。DDCは指示符が少ない譜面と多い譜面を両方学習しているため、指示符の頻度に関するこれらの特性を共に反映させようとした結果、生成譜面のある区間では指示符が少なく、またある区間では指示符が多いという、複数の特性が混在した中途半端な生成結果になったと考えられる。このような生成結果は、音響信号からダンスゲーム譜面を生成するという Donahue らの目標自体は達成していても、多様な面白さの感性に対して充分に対応できているとは言えない。また、大多数のデータが共通して持つ特性を学習する過程で、中程度の事例数がある特性が学習されず、本来あるべき特性が獲得されないことがある。例えば、ジャンプステップを特性として持つ譜面は一定数の存在が認められるが、全体の傾向としてはジャンプステップの事例数は多くはない。そのため、全ての譜面を学習したモデルでは、ジャンプステップを特性とする譜面は生成できない。極端にデータ数が少ない特性は、統計的機械学習のアプローチでは学習が困難である。しかしながら、一定数の事例が存在する特性は、大多数の特性と分離することで学習可能になるため、難しさの個性として扱い、譜面生成モデルを構築できると考えた。これらの問題を解決するためには、音響信号のみではなくゲーム譜面そのものの特性に着目し、学習データとなる譜面を特性ごとに分類した上で学習することが有効であろう。

データを分類した上でモデルを学習させることの効用は、ゲーム以外の生成モデルにおいても指摘されている。Saroffらは、自動作曲を行う Autoencoder モデルを 10 ジャンルの楽曲を用いて学習させた [22]。これに対して Colonelらは、文献 [23] p.5において

Including several genres of music in a corpus trains an autoencoder to be a jack of all trades, but master of none. By focusing the corpus of our exercise on tonal sounds, we encourage the net to master representations of those sounds.

と、学習データに複数のジャンルの楽曲を含めず特定のジャンルに集中させるべきであると指摘している。

本章では、譜面の難易度と面白さに影響する要素として考えられる特徴を挙げ、各特徴に基づいてゲーム譜面をクラスタリングする。これによって、データセット内に面白さの特性が異なるデータが含まれていることを示し、特性に基づくデータセットの整理を実施する。そして、クラスタごとに譜面生成モデルを学習させることによって、一貫した面白さの特性を備えた特徴的な譜面の自動生成を実現する。

### 3.2 ダンスゲーム譜面の多様性

同程度の難易度とされているダンスゲーム譜面であっても、それぞれの譜面が持つ特性は大きく異なる。図 3.1, 図 3.2 は、それぞれ “Disconnected” (Hard), “Sweet World” (Hard) の一部であ



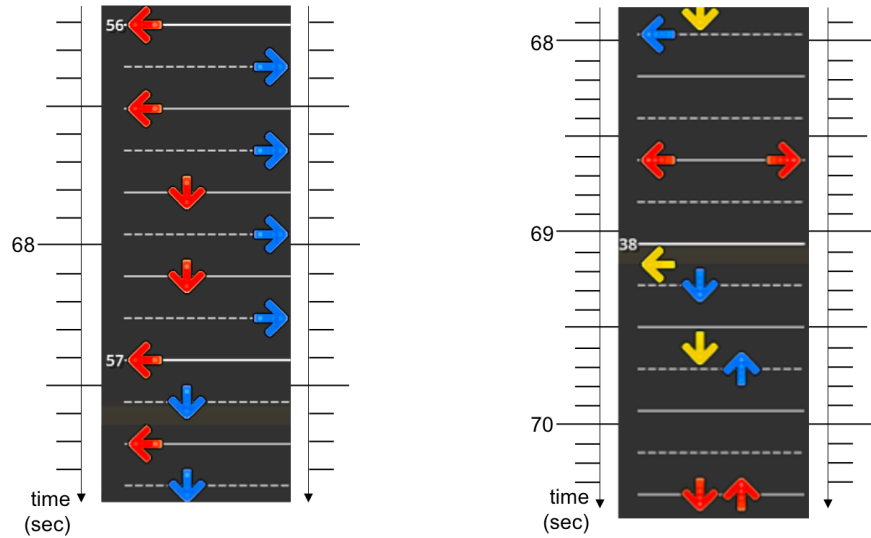


図 3.1: “*Disconnected*” (Hard) の一部. 図 3.2: “*Sweet World*” (Hard) の一部.  
テンポは 200BPM. テンポは 132BPM.

る. 図中の時間軸は 0.1 秒ごとに目盛を付している. 図 3.1 の “*Disconnected*” (Hard) は, テンポが 200BPM (Beats Per Minute) と速く, 図の範囲で 1 秒あたりに約 6.1 ステップが存在しているため, 動作を素早く行う必要がある. 一方で, ステップ同士の時間間隔が一定であり, 4 分と 8 分のステップのみで構成されているため, リズムが単純である. 図 3.2 の “*Sweet World*” (Hard) は, テンポが 132BPM と遅めであり, 図の範囲で 1 秒あたり約 2.8 ステップとステップ頻度が低いため, 素早い動作はあまり求められない. しかし, 4 分のステップが少なく 8 分や 16 分のステップが多いという, いわゆる裏拍を主としたリズムであるため, 動作を行うタイミングが難解である. この 2 つの譜面は, ともに Hard 難易度として用意された譜面であるが, ステップ頻度とリズムの複雑さという特徴に注目すると, 譜面の特性が異なることがわかる. そのため, プレイヤによって, この 2 つの譜面のうちどちらをより難しいと感じるか, どちらをより面白いと捉えるかは, 異なると考えられる. 例えば, 体を動かすことは好きだが, リズム感に自信がないプレイヤにとっては, 前者の方が遊びやすく楽しい譜面であると感じるであろう.

同程度の難易度とされている譜面同士で特性が異なる事例がある一方で, 難易度が異なる譜面同士で特性が類似している事例も存在する. 図 3.3 は “*Remember December*” (Challenge) の一部であり, 図 3.1 および図 3.2 よりも難しい譜面として制作されている. テンポは 192BPM であり, 図の範囲のステップ頻度は 1 秒あたり約 8.3 ステップである. リズムは 4 分と 8 分のステップを中心に, 時折 16 分のステップが混ざる構成となっており, 図 3.1 の “*Disconnected*” (Hard) よりも複雑だが図 3.2 の “*Sweet World*” (Hard) ほど難解なリズムではないと考えられる. これらの特徴から形成される図 3.3 の “*Remember December*” (Challenge) の特性は, 図 3.2 の “*Sweet World*” (Hard) よりも図 3.1 の “*Disconnected*” (Hard) に近いと考えられる. このように譜面の特徴を考慮することで, データセットを類似した特性を持つ譜面の集合に分割し, 相反する特性を持つ譜面同士を分離するこ

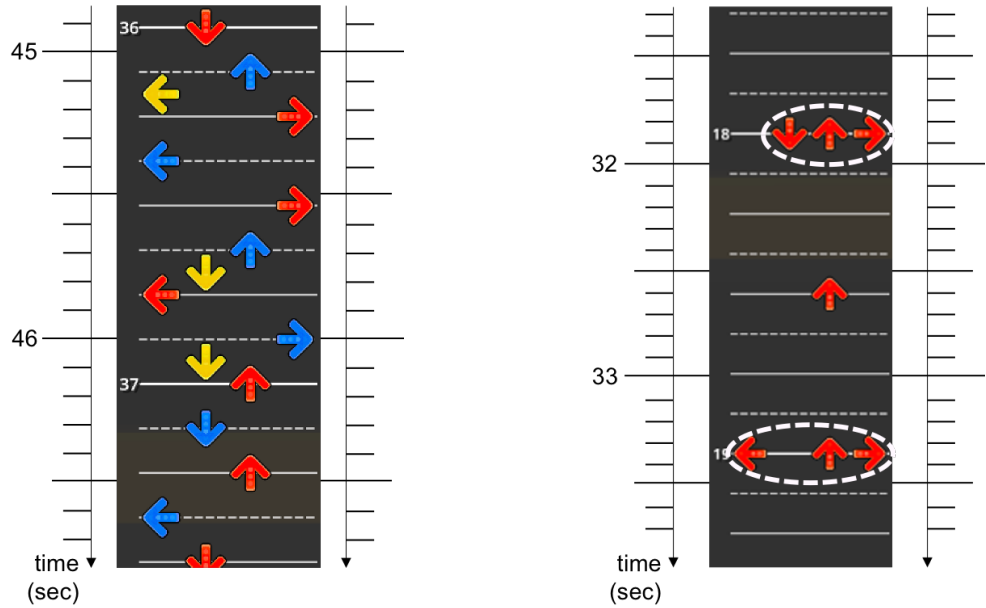


図 3.3: “Remember December” (Challenge) 図 3.4: “Bend Your Mind” (Hard) の一部。  
 の一部。テンポは 192BPM. テンポは 140BPM.

とができる。

ITG データセットの中には、他の譜面とは大きく異なる特性を持つ譜面が存在する。極端な特性を持つ譜面の一例として、図 3.4 に “Bend Your Mind” (Hard) の一部を示す。図中の白点線で囲んだステップは、プレイヤーに足元のパネルを 3 箇所同時に踏むことを指示している。1.3 節で定めた「プレイヤーは足を用いて、片足につき 1 つまでパネルを踏むことができる」という前提において、この指示は達成不可能であり、手と足を併用するなど前提に背く方法で無理やりパネルを反応させる必要がある。このようなステップを含む譜面は、ダンスゲーム本来の「音楽に合わせて、単純な動作を連続して行う」という面白さとは異なる、「困難な動作の指示に対して、攻略法を考え出す」という新たな面白さをプレイヤーに提供するものと位置づけることができる。しかしながら、このような特性を持つ譜面はデータセット内に少数しか存在しない。その他の極端な特性を持つ譜面の例として、64 分のステップが譜面中の 14.8% を占め、非常に繊細なリズム把握能力が求められる “Go 60 Go” (Challenge) や、譜面中の約 23.7 秒間に渡ってステップの頻度が 1 秒あたり約 10.7 ステップと一定であり、素早い動作を長時間持続することを求められる “Delirium” (Challenge) などが挙げられる。これらのような譜面はクラスタリングや譜面生成モデルの学習においてノイズとなることが懸念される。これらのような極端な特性を持つ譜面をあらかじめ検出し、データセットから取り除くことによって、多くの譜面に共通する特性をモデルに学習させる。

### 3.3 ダンスゲーム譜面から得る特徴量の提案と分析

著者の知見に基づき、譜面データから得る特徴量を設定した。特徴量は、大きく分けて「譜面特徴

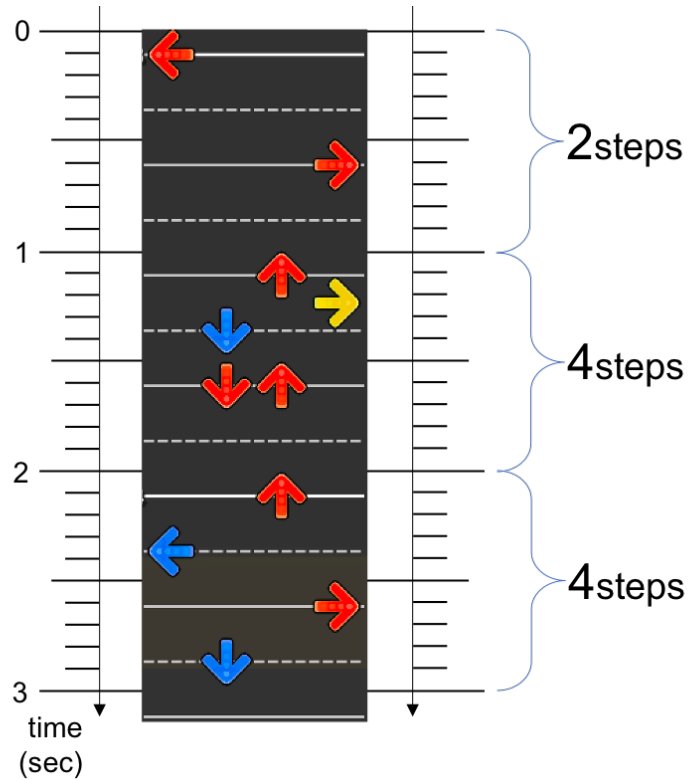


図 3.5: 1 秒あたりステップ数の集計例

量」と「楽曲特徴量」に分類される。譜面特徴量は、同じ曲に対して用意された譜面であったとしても、得られる値は譜面ごとに異なる。一方で、楽曲特徴量は、同じ曲に対して用意された譜面であれば、異なる譜面であっても同一の値が得られる。いくつかの特徴量は Donahue らの論文 [3] 中でも触れられている。特徴量の計算には、譜面中の最初のステップから最後のステップまでの部分を用いた。

### 3.3.1 譜面特徴量

#### 1 秒あたりステップ数

譜面全体を 1 秒ごとに切り分ける。切り分けられた 1 秒ごとの各区間に含まれるステップ数を数えることで、以下の特徴量を得る。

$n_l$  最もステップ数が少ない区間のステップ数。

$n_m$  最もステップ数が多い区間のステップ数。

$n_\mu$  譜面中の全ての区間のステップ数の平均。

$n_{\sigma^2}$  譜面中の全ての区間のステップ数の分散。

例として、図 3.5 のような 3 秒間のみの譜面を考える。0 秒から 1 秒の区間に 2 ステップ、1 秒から 2 秒の区間に 4 ステップ、2 秒から 3 秒の区間に 4 ステップが存在する。この譜面に対する特徴量の値は、

$$\begin{aligned} n_l &= 2, \\ n_m &= 4, \\ n_\mu &= \frac{2 + 4 + 4}{3} \approx 3.33, \\ n_\sigma^2 &= \frac{(2 - n_\mu)^2 + (4 - n_\mu)^2 + (4 - n_\mu)^2}{3} \approx 0.89, \end{aligned}$$

となる。

### ジャンプステップ

1.3 節に述べたとおり、同一のタイミングに 2 つ以上の矢印が存在するジャンプステップは、プレイヤに跳躍という特殊な動作を指示するステップである。ある譜面中のジャンプステップの回数を集計し、以下の特徴量を得る。

$j_r$  = ジャンプステップの回数/総ステップ数.

$j_d$  = ジャンプステップの回数/譜面の長さ [秒].

### 拍階層割合

1.3 節の定義に基づいて、譜面中のそれぞれのステップがどの拍階層に存在するかを集計し、以下の特徴量を得る。

$l_4$  = 4 分のステップ数/総ステップ数.

$l_8$  = 8 分のステップ数/総ステップ数.

$l_{12}$  = 12 分のステップ数/総ステップ数.

$l_{16}$  = 16 分のステップ数/総ステップ数.

$l_{24}$  = 24 分のステップ数/総ステップ数.

$l_{32}$  = 32 分のステップ数/総ステップ数.

$l_{oth} = 1 - \sum_{q \in \{4, 8, 12, 16, 24, 32\}} l_q$ .

表 3.1: 出現頻度上位 3 種類の 3-gram

Step 1	Step 2	Step 3	出現頻度	出現譜面数
↑	↓	→	4,457	600
↓	↑	↓	4,238	553
←	↑	↓	4,216	589

### ステップ向き 3-gram の $tf - idf$

連続する 3 ステップの向きの組み合わせについて出現頻度を集計し、3-gram とする。表 3.1 に、出現頻度上位 3 種類の 3-gram を示す。ITG データセット内には、6,910 種類の 3-gram が存在し、出現頻度の総和は 176,279 であった。このうち、総出現頻度の 90% を占める出現頻度上位 979 種類の 3-gram を、分析の対象とした。以下、979 種類の 3-gram の集合を  $G$  とし、 $g \in G$  とする。

それぞれの  $g \in G$  に対して、ある 1 譜面中の出現頻度  $tf_g$  と、データセット内で出現している譜面の数  $df_g$  から、数式 (3.1) を用いて  $tf - idf$  値を計算し、 $s_g$  とする。

$$s_g = \frac{tf_g}{\sum_{k \in G} tf_k} \times \log_2 \frac{m}{df_g}. \quad (3.1)$$

数式 (3.1) で、 $m$  はデータセット内の譜面の数 652 である。 $s_g$  は、 $g$  がその譜面にどの程度固有であるか、すなわちどの程度特徴付けているかを示す。

### 3.3.2 楽曲特徴量

#### テンポ

ITG データセットに含まれる譜面データには、楽曲のテンポ値の情報が記録されている。テンポ値の単位には BPM (Beats Per Minute) が用いられている。楽曲の途中でテンポが変化する場合は、全ての変化点のタイミングと変化後のテンポ値が記録されている。

*Stepmania* においては、テンポが速いほど画面上のステップの移動速度が早くなり、プレイヤーに素早い判断を要求することになるため、一般的にテンポが速い曲の方が難しいとされている。また、楽曲途中でテンポの変化は、通常は等速運動をするステップの移動速度の変化を引き起こし、プレイヤーにとって予測しづらいギミックとして作用する。

以下に、テンポに関して取得する特徴量を挙げる。単位は BPM である。 $t_{\mu}$  を除き、小数第一位を四捨五入し、整数値として扱う。

$t_l$  譜面中で最も長い時間を占めるテンポ値。

$t_s$  譜面中で最も遅い瞬間のテンポ値。

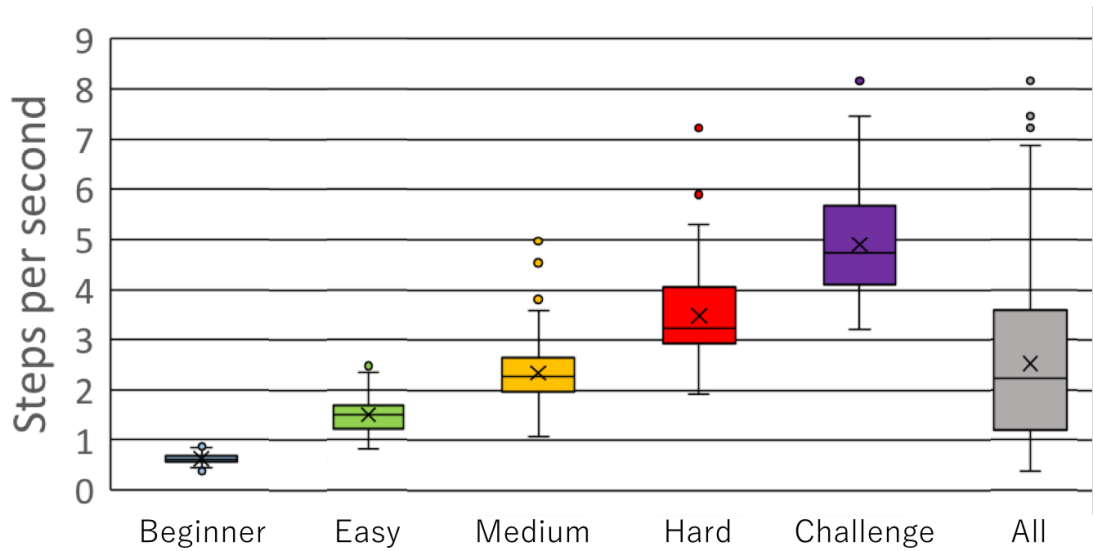


図 3.6:  $n_{\mu}$  の分布を示す箱ひげ図

$t_f$  譜面中で最も速い瞬間のテンポ値.

$t_{\mu}$  = 総拍数/楽曲の長さ [秒]  $\times$  60.

さらに、テンポの変化に関する以下の特徴量を得る.

$t_r = t_f/t_s$ .

$t_n$  テンポ変化が記録されている回数.

$t_t$  変化前と変化後のテンポの比が  $2^m$  ( $m \in \mathbb{Z}$  and  $m \neq 0$ ) であるテンポ変化が記録されている回数.

ステップの移動速度変化のみを目的としたテンポ変化の回数とみなすことができる.

### 譜面停止ギミック

ITG データセット内のいくつかの楽曲には、指定された時間の間ステップの移動を一時停止する「譜面停止ギミック」が設定されている。これは、楽曲途中のテンポ変化と同様、ステップの移動速度に変化を与えることでプレイヤーを騙すギミックとして設定されている。譜面停止ギミックに関する特徴量は、以下の1つである。

$p_n$  譜面停止ギミックが設定されている回数.

### 3.3.3 特徴量に基づく譜面の分析と考察

ITG データセット内の全ての譜面について特徴量を取得し、特徴量が譜面の難易度に影響していること、および他の譜面とかけ離れた値を持つ譜面が存在することを確認した。図 3.6 に、 $n_{\mu}$  の集

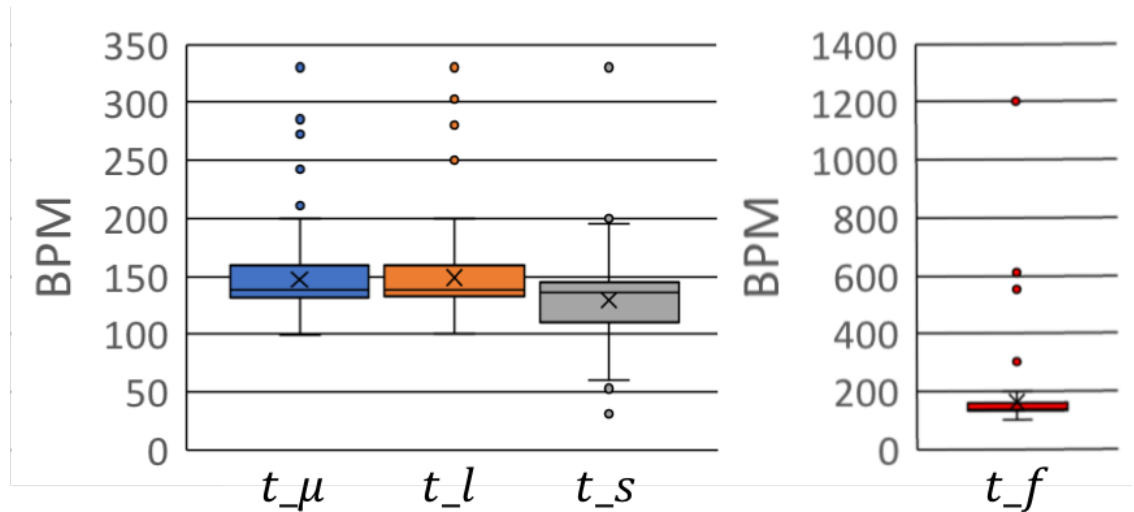


図 3.7:  $t_{\mu}, t_l, t_s, t_f$  の分布を示す箱ひげ図

計結果を表す箱ひげ図を示す。Beginner, Easy, Medium, Hard, Challenge はそれぞれの難易度レベルがつけられた 133 譜面（Challenge のみ 120 譜面）の集計結果，All はデータセット内の全 652 譜面についての集計結果を示す。また，図 3.7 に，データセット内の 133 楽曲に対する  $t_{\mu}, t_l, t_s, t_f$  の集計結果を表す箱ひげ図を示す。両方の図において， $Q_1, Q_3$  をそれぞれ第 1 四分位数，第 3 四分位数として， $Q_1 - 1.5(Q_3 - Q_1)$  を下回る値，および  $Q_3 + 1.5(Q_3 - Q_1)$  を上回る値を外れ値として示している。

### 1 秒あたりのステップ数平均

図 3.6 より，難易度が上がるにしたがって，平均ステップ数が多い範囲に分布しており，ステップ数は難易度が高いほど多くなる傾向がわかる。また，難易度が上がるにしたがって，四分位範囲  $Q_3 - Q_1$  が広がっている。このことから，簡単な譜面はどの楽曲でも似たような頻度でステップが現れる一方，難しい譜面では楽曲によってステップ頻度に差があると言える。さらに，Medium 難易度と Hard 難易度には，それぞれ 1 つ上の難易度の第 3 四分位数よりも高い値を示す譜面が存在が外れ値として存在する。これらの譜面は，少なくとも  $n_{\mu}$  の観点においては，該当の難易度としては不適切に難しい譜面であるとみなすことができる。

### テンポ

図 3.7 中の  $t_{\mu}, t_l$  より，ITG データセットの楽曲のうち半分が 130~160BPM の範囲に該当することがわかった。一方で，例外的に速いテンポの楽曲も存在する。“Pandemonium” は楽曲を通して 330BPM であるため， $t_s = t_f = 330$  という値を示した。この値は，特に  $t_s$  において極端な外れ

表 3.2: クラスタリング結果

ID	譜面数	特徴量のクラスタ内平均値							クラスタの特性
		$n_{-}\mu$	$j_{-}r$	$j_{-}d$	$l_4$	$l_8$	$l_{16}$	$t_{-}\mu$	
1	133	3.512	7.8%	0.268	53.9%	34.0%	11.9%	140.265	ステップ数が多い.
2	122	1.229	5.8%	0.070	95.4%	4.5%	0.1%	134.592	ステップ数が少ない.
3	82	1.868	13.8%	0.256	86.6%	12.9%	0.5%	143.252	ジャンプステップが多い.
4	58	2.493	8.1%	0.200	74.0%	21.1%	4.7%	140.788	ステップ数が中程度.
5	26	2.397	6.2%	0.150	84.5%	14.9%	0.6%	185.993	テンポが速い.
6	24	0.758	5.8%	0.070	99.7%	0.3%	0.1%	135.177	4分のステップの割合が高い.
7	1	4.374	11.2%	0.492	36.4%	30.7%	26.2%	107.004	“Zodiac” (Challenge) のみ.
8	1	2.293	1.3%	0.030	47.4%	40.8%	8.3%	127.994	“Hustle Beach” (Hard) のみ.

値となっている。“Robotix”は $t_l = 150$ だが、楽曲中盤に徐々に加速し $t_f = 1,200$ を記録したのちに減速するという構成になっており、 $t_f$ において他の楽曲と大きくかけ離れた値が観測されている。以上から、ITG データセットには140BPM前後のテンポの楽曲が多く収録されている一方で、200BPMを超えるような高速テンポの楽曲もわずかに存在していることが確認された。

### 3.4 クラスタリングに基づく譜面の自動生成

3.3節で設定した特徴量を用いることで、譜面の特性を数値で表現することを実現した。特徴量の傾向が近い譜面同士をクラスタリングすることによって、データセット内の譜面を特性に基づいて分類し、共通の特性を持つ譜面の集合を得ることができる。クラスタごとに譜面生成モデルを学習させることによって、そのクラスタの特性を反映した特徴的な譜面の自動生成を実現する。

#### 3.4.1 前処理：外れ値を持つ譜面の検出

3.3.3節より、ある特徴量において他の譜面とかけ離れた値を持つ譜面がデータセット内に存在することが確認された。このような譜面は面白いと評価される一方で、類似した性質を持つ他の譜面がほぼ存在しない特殊な事例とみなされる。統計的な機械学習のアプローチにおいて、データ数が少ない特殊事例は学習の対象として適切ではないうえ、大多数の譜面が共有している特性の学習を阻害するノイズデータとして作用してしまう。ノイズデータが引き起こす問題の一例として、クラスタリングにおいて少数のデータのみで1つのクラスタを占めてしまうという事例を確認した。また、譜面生成モデルがノイズデータを学習すると、生成される譜面にノイズデータの性質が中途半端に反映されてしまう。

これらの問題に対処するため、外れ値を持つ譜面をあらかじめデータセットから除外する。データセット内の全652譜面を母集団として、各特徴量に対して増山の棄却検定( $p = 0.01$ ) [24]を実施



し、外れ値を検出した。ただし、特徴量の次元数が 979 と非常に大きい  $s_g$  については、主成分分析 (PCA) を実施し、寄与率が高い上位 7 つの主成分にのみ棄却検定を実施した。652 譜面のうち、少なくとも 1 つの特徴量が外れ値と見なされた 205 譜面を、データセットから除外した。

### 3.4.2 クラスタリング

クラスタリングを行うにあたって、3.3 節にあげた特徴量を用いて各譜面を表す特徴ベクトルを生成する。譜面ごとに、以下の 4 種類の特徴ベクトルを生成する。

$$\mathbf{n} = (n_l, n_m, n_\mu, n_{\sigma^2}, j_r, j_d),$$

$$\mathbf{l} = (l_4, l_8, l_{12}, l_{16}, l_{24}, l_{32}, l_{oth}),$$

$$\mathbf{s} = (s_g | g \in G),$$

$$\mathbf{t} = (t_\mu, t_l, t_s, t_f, t_r, t_n, t_t, p_n),$$

ただし、 $\mathbf{s}$  は 979 次元のベクトルとなる。

それぞれの特徴ベクトルに対して PCA を実施した。 $\mathbf{n}, \mathbf{l}, \mathbf{t}$  については、全ての主成分を利用し、それぞれ同一の次元数をもつベクトル  $\mathbf{pc}\text{-}\mathbf{n}, \mathbf{pc}\text{-}\mathbf{l}, \mathbf{pc}\text{-}\mathbf{t}$  に変換した。 $\mathbf{s}$  は次元数が非常に大きいため、累積寄与率が 90% を超えるように寄与率が高い主成分から順に選択し、208 次元のベクトル  $\mathbf{pc}\text{-}\mathbf{s}$  に変換した。

$\mathbf{pc}\text{-}\mathbf{n}, \mathbf{pc}\text{-}\mathbf{l}, \mathbf{pc}\text{-}\mathbf{s}, \mathbf{pc}\text{-}\mathbf{t}$  のうち任意数を結合することで、譜面ごとに 1 つのベクトルを生成する。本論文では、4 つ全てを結合した 229 次元のベクトルを用いてクラスタリングした結果について報告する。

3.4.1 節で外れ値が検出されなかった 447 譜面を対象に、 $k$ -means 法によるクラスタリングを行った。クラスタ数  $k$  を、4 から 12 の範囲で変更しながらクラスタリングを試行した。本論文では、試行した中で各クラスタの特性が最も明確に現れていると著者が判断した、 $k = 8$  の結果を代表として報告する。表 3.2 に、クラスタリング結果を示す。表内の「クラスタ内の特性」は、特徴量の平均値に基づいて著者が考察し与えたものである。クラスタ 1, 2, 4, 6 は、ステップの頻度  $n_\mu$  とリズムの複雑さ  $l_k (k \in \{4, 8, 16\})$  に差が見られた。クラスタ 3 は  $j_r$  と  $j_d$  が他に比べて高く、ジャンプステップが多い譜面が属した。クラスタ 5 は  $t_\mu$  が高く、テンポが速い楽曲の譜面が属した。以上のように、クラスタによってそれぞれ異なる特性を示すことが確認された。

外れ値特徴量を持つ譜面を除外したにも関わらず、クラスタ 7 とクラスタ 8 は 1 譜面のみが属するクラスタとなった。これらは、単一の特徴量で外れ値となることはないが、高い値を示す特徴量の組み合わせが他の譜面とは異なっているために起きた現象と考えられる。今後の課題として、Minimum Covariance Determinant [25] のような多変量データに対する異常検出手法を適用し、この問題を解決することがあげられる。また、本論文ではクラスタリング数の妥当性を著者の主観に基づいて評価した。客観的な指標を用いた妥当性の評価についても今後の課題としたい。

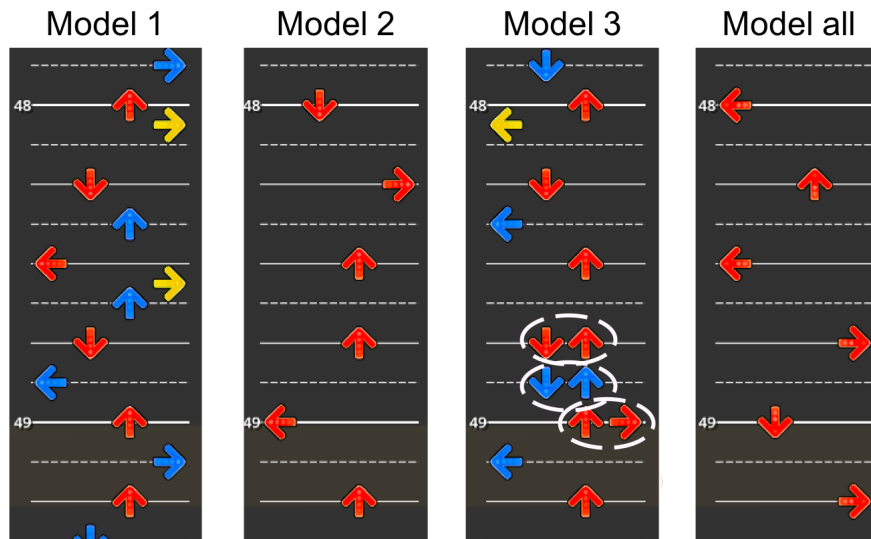


図 3.8: 楽曲 “Walking on Fire” を各モデルに入力し，出力された譜面の一部

### 3.4.3 譜面の学習と自動生成

クラスタごとに，音楽データを入力，譜面データを出力とする LSTM モデルを学習させた．モデルの構造は，Donahue らの先行研究 DDC [3] と同一のものを用いた．

譜面数が比較的多い，クラスタ 1, 2, 3 を学習の対象とした．クラスタごとにランダムに，64 譜面を学習データ，9 譜面を検証データ，9 譜面をテストデータに振り分けた．クラスタ  $n$  を学習したモデルを，Model  $n$  と名付けた．また，比較対象として，上記 3 クラスタ全ての学習データ 192 譜面と検証データ 27 譜面を用いて学習したモデルを用意し，Model all とした．

図 3.8, 図 3.9, 図 3.10, 図 3.11 に，それぞれ楽曲 “Walking on Fire” (クラスタ 1 テストデータ), “Hillbilly Hardcore” (クラスタ 2 テストデータ), “Bubble Dancer” (クラスタ 2 テストデータ), “Twilight” (クラスタ 3 テストデータ) を入力音楽データとした際に，各モデルが生成した譜面の一部を示す．同一図中の 4 つの譜面は，それぞれ入力楽曲の同一部分に対して生成された譜面である．破線丸で囲ったステップはジャンプステップである．Model 1, 2, 3 に注目すると，配置されたステップ数について，Model 2 が最も少なく，Model 3, Model 1 の順に多くなっていくことが確認される．また，Model 3 が生成した譜面にはジャンプステップが多く存在するが，Model 1 と Model 2 が生成した譜面にはジャンプステップが見られなかった．以上から，Model 1, 2, 3 が生成した譜面は，3.4.2 節に記述したクラスタ 1, 2, 3 の特性をそれぞれ強く反映しており，それぞれが異なる面白さの特性に指向した譜面であると考えられる．その一方で，3 つのクラスタ全てを学習した Model all の生成譜面におけるステップ数は，図 3.9 と図 3.10 では Model 2 と同程度である一方で，図 3.8 と図 3.11 では Model 2 と 3 の中間程度となっている．図 3.8, 図 3.9, 図 3.11 の Model all の生成譜面にはジャンプステップが見られなかったが，図 3.10 中にはいくつかのジャンプステップが確認された．Model all の生成譜面にはそれぞれのクラスタの特性が混在しており，一貫した特性を持つ譜面

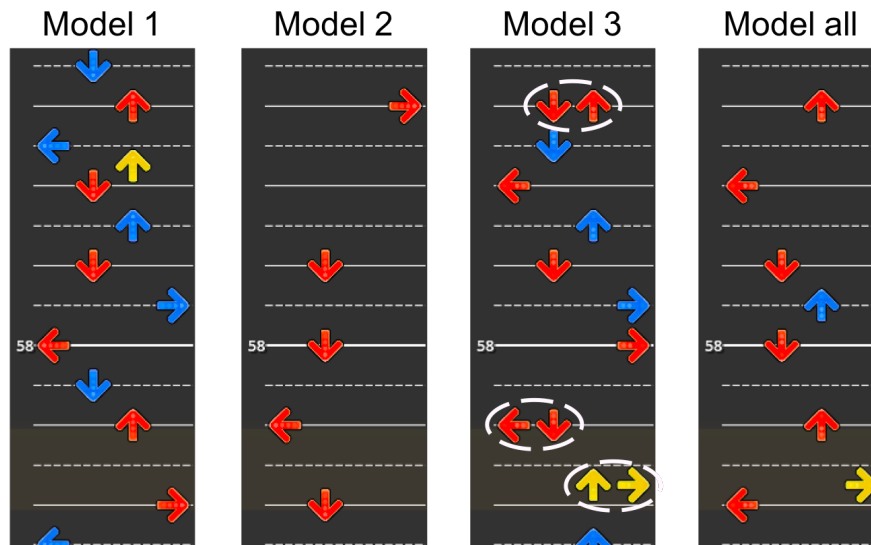


図 3.9: 楽曲“Hillbilly Hardcore”を各モデルに入力し，出力された譜面の一部

表 3.3: 各モデルの  $F$ -score

	Test 1	Test 2	Test 3
Model 1	<b>0.678</b>	0.364	0.474
Model 2	0.546	<b>0.455</b>	0.508
Model 3	0.676	0.401	0.506
Model all	0.662	0.444	<b>0.517</b>

を生成していないと考えられる。

### 3.4.4 生成譜面の評価

#### $F$ -score

それぞれのクラスターでテストデータに分類された楽曲を，各モデルに入力することによって，譜面を生成した．生成された譜面でステップが配置された時刻と，正解データにおいてステップが存在する時刻を比較し， $F$ -score を計算した．表 3.3 に結果を示す．ここで，Test  $n$  はクラスター  $n$  から選ばれたテストデータを表す．

Test 1 および Test 2 については，それぞれ対応する Model 1 と Model 2 が，他のモデルよりも高い  $F$ -score を示した．一方で，Test 3 に対して最も高い数値を示したのは，Model 3 ではなく Model all であった．このことから，Model 1 および Model 2 は，ステップを配置する時刻についての特徴をうまく捉えられていると言える．しかし，Model 3 については，ステップ配置時刻の特徴を捉えていない．

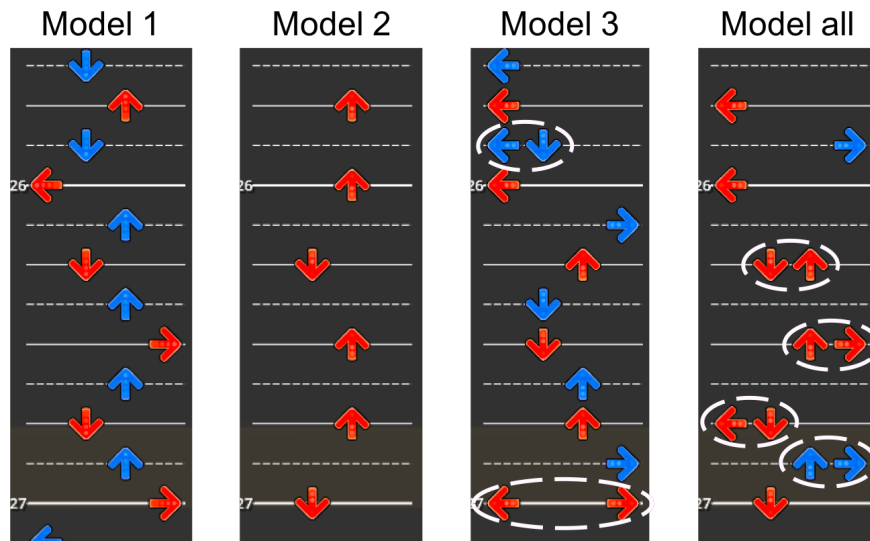


図 3.10: 楽曲 “Bubble Dancer” を各モデルに入力し，出力された譜面の一部

表 3.4: 各モデルが生成した譜面の  $n_{\mu}, j_r, j_d$  の平均値

	$n_{\mu}$	$j_r$	$j_d$
Model 1	4.481	1.7%	0.067
Model 2	1.736	1.2%	0.020
Model 3	3.816	11.2%	0.423
Model all	2.998	6.7%	0.181

### データセットと自動生成譜面の統計比較

各モデルに Test 1, Test 2, Test 3 の楽曲データを入力し，計 24 譜面を生成した．表 3.4 に，生成された各譜面に対して  $n_{\mu}, j_r, j_d$  を計算し，24 譜面の平均を求めた結果を示す．表 3.4 と表 3.2 を比較すると， $n_{\mu}$  について，Model 1 および Model 2 はそれぞれある程度クラスタ 1 およびクラスタ 2 に近い値を示したが，Model 3 はクラスタ 3 よりも明らかに高い値を示した． $j_r$  と  $j_d$  について，Model 3 は他のモデルよりも高い値を示した．以上より，Model 1 および Model 2 については，対応するクラスタのステップ頻度についての特性を，ある程度学習できたと言える．Model 3 は，ジャンプステップが多いというクラスタ 3 の特性を反映できている一方で，ステップ頻度についてはうまく学習できなかったのではないかと考えらえる．

### 3.5 おわりに

本章では，ダンスゲーム譜面の難易度の個性，すなわち面白さの特性に影響する特徴量を設定した．設定した特徴量は，難易度によって確かに差があることを確認した．また，いくつかの特徴量で例外的に高い値を示す譜面が存在することを確認した．このような極端な譜面をデータセットから除

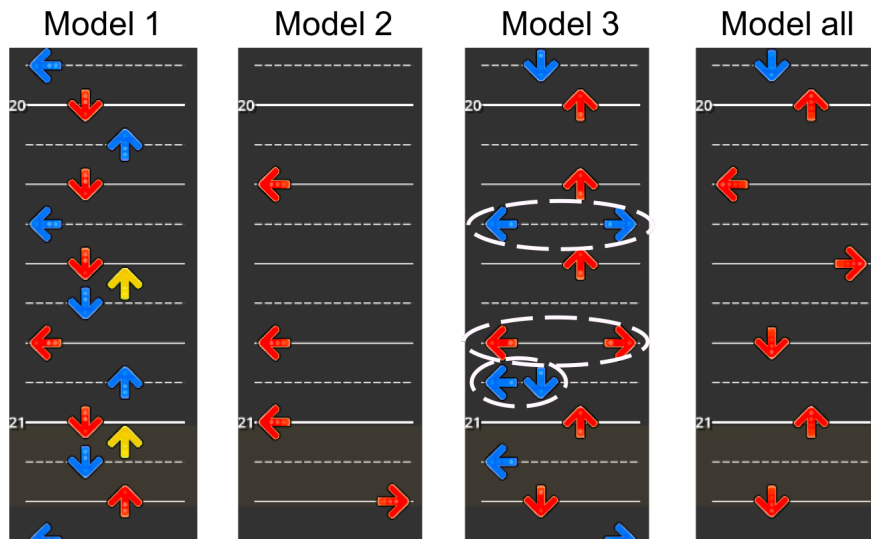


図 3.11: 楽曲 “*Twilight*” を各モデルに入力し，出力された譜面の一部

外した上で，クラスタリングを実施した．クラスタリングを通して，ステップの頻度，リズムの複雑さ，ジャンプステップの頻度などの特徴に基づいて，特性ごとに譜面を分類することができた．クラスタごとに譜面生成モデルを学習させることで，各クラスタの特性を反映した，異なる面白さの特性を持つ譜面が生成できた．客観評価実験を通して，各譜面生成モデルが生成した譜面の特性が，学習したクラスタに含まれる譜面の特性と一致していることを確認した．

本章の提案手法において，それぞれの譜面生成モデルが学習に用いたデータ数は決して多くはないが，生成されたダンス譜面は面白さの各特性に指向していることが確認された．従来研究においては，譜面の特性を考慮せずに学習データセットを構築し，譜面生成モデルを学習させていた．その結果，学習データ量は十分にあるものの，「生成譜面に特性が混在している」「本来あるべき特性が反映されていない」といった問題が生じており，プレイヤーにとっては面白いと感じられない譜面が生成されていたと考察される．これは，エンタテインメント分野におけるコンテンツ生成タスクにおいて，データセットからノイズを取り除き，コンテンツの特性に指向した学習データセットを用意することで，エンタテインメント性の高いコンテンツの生成が可能になることを示唆している．提案手法のアプローチは，本来存在するデータ量を減らすことになるが，面白さの観点でより高品質なコンテンツ生成を目指す上で有効であると考えられる．一方で，本論文で外れ値として除外した 205 譜面は，それぞれが他の譜面とは異なる面白さの特性を備えている譜面である．このような独自の特性を備えた譜面を生成することは，最終的な目標の 1 つである．譜面のデータ数が増加し，ITG データセットのみでは特殊事例であった特性が譜面の傾向として形成されれば，クラスタリングなどを通してこれらの特性を獲得し生成譜面に反映させることが可能になると考えられるが，これは今後の課題としたい．

今後の展望として，ダンスゲームのプレイヤーが生成された譜面の特性を判別できるかどうかを，被験者実験によって評価することを検討している．この被験者実験を通して，クラスタリングによって獲得した譜面の特性が，難しさや面白さの個性としてプレイヤーが認識している譜面の特性であること

を確認できると考える。また、クラスタリングに用いる特徴量の精査、各クラスタに含まれた譜面に対する考察などを通して、より譜面の特性を詳細に分析することも今後の課題としたい。本章では、学習データセットの変化による効用を検証するため、譜面生成には Donahue ら [3] と同じ構造のモデルを用いた。モデル構造や入力特徴量を変更することによって、入力楽曲やプレイヤーの嗜好に応じて、難易度および面白さの特性を調整したダンス譜面を生成する方法についても検討する。

## 第 4 章

# プレイヤーのゲーム習熟度と嗜好を反映した ダンスゲーム譜面に対する動作の推定

### 4.1 はじめに

ダンスゲームにおける指示符であるステップは、プレイヤーがどちらの足を用いてパネルを踏むかについては指示しておらず、どのような姿勢でパネルを踏むかはプレイヤーの解釈に委ねられている。また、ステップによって指示されていないパネルについては、足の有無は問われない。したがって、あるゲーム譜面の指示を満たすプレイヤーの動作は一意には定まらない。本章の研究課題は、入力されたゲーム譜面に対して、その譜面の指示を満たし、かつプレイヤーの身体姿勢として破綻しない一連の動作系列を、計算機を用いて推定することである。

指示符系列に対する人間の動作が一意に定まらないという問題は、音楽ゲームに限らず、実際の楽器演奏動作においても発生している問題である。楽譜に対する楽器演奏動作を計算機を用いて推定する課題は「運指決定」や「運指推定」と呼ばれ、ピアノ [26] やギター [27, 28] などを対象とした先行研究が存在している。これらの研究においては、楽譜を入力して、その楽譜通りの発音系列を得られる運指系列を推定するモデルを構築している。ピアノを対象としたモデル [26] であれば、楽譜内の音符 1 つ 1 つに対して、両手 10 本の指のうちどの指を用いて対応する鍵盤を押さえれば良いかを、運指番号という表現形式で出力している。ギターを対象としたモデル [27, 28] では、楽譜内の各和音に対して、どの弦のどの位置を押さえれば良いかを表すタブ譜を出力している。どちらの出力内容も、各発音時刻においてそれぞれの指が「発音のために、楽器の特定の場所を押さえている」または「発音に関与せず、楽器のどの場所も押さえていない」のどちらの状態にあるか、という情報を与えているものと解釈できる。楽器演奏においては、楽譜によって指示された音を過不足なく発音することが求められるため、上記の出力情報が必要十分である。一方で、本論文で研究対象とするダンスゲームにおいては、前述の通り、ステップによって指示されていないパネルを踏むという操作が許容されている。したがって、ゲーム譜面中でステップが存在する各時刻において、それぞれの足は「ステップの指示を満たすために、あるパネルを踏んでいる」「どのパネルも踏んでいない」という状態の他に、「ステップの指示とは関係ないが、あるパネルを踏んでいる」という第三の状態を取りうる。この第三の状態が許容されることによって、プレイヤーが両足を接地させて身体姿勢を安定させることが可能となっている。ダンスゲームを遊ぶ上で、身体姿勢が安定しなければプレイヤーは転んでしまい、ゲームの続行が不可能になってしまうであろう。そのため、ダンスゲームにおいて動作推定を実現するためには、第三の「ステップの指示と関係ないが、あるパネルを踏んでいる」状態を考慮し、身体姿勢

の安定を重要視する必要がある。この点において、ダンスゲームにおける動作推定課題は、楽器における運指推定課題と類似した目的を持ちつつも、異なるアプローチで解決しなくてはならない問題であると位置付けられる。

ダンスゲームを遊ぶプレイヤー層は様々であり、何年も継続して遊ぶ上級者もいれば、数回遊んでみて興味を持ち始めたばかりの初心者もいる。多くのプレイヤーがゲームを楽しめるよう、ダンスゲームには難易度の異なるたくさんの譜面が用意されている。また、3章の通り、各譜面は異なる「面白さの特性」を有しており、プレイヤーによって得意とする譜面、および好んで遊ぶ譜面が異なる。したがって、同じダンスゲームに収録されている譜面であっても、譜面ごとに想定されているターゲットプレイヤーは細かく分かれていると言える。逆を言えば、ダンスゲームを遊ぶプレイヤーもまた、ゲームの習熟度および嗜好の観点において、個人個人が異なるプロファイルを持つと言える。ダンスゲームを対象とした身体動作推定モデルは、このようなプレイヤーごとのプロファイルを反映し、それに合わせた推定を行う機能を実装することが望ましい。これを実現するため、プレイヤーのプロファイルを表現するためのヒューリスティックなプレイヤー固有パラメータを提案する。プレイヤー固有パラメータとして与えられた値を制約条件として、最尤の身体動作を推定することによって、プレイヤーのプロファイルに合わせた動作推定を実現する。

本章では、あるダンスゲーム譜面に対してプレイヤーがどのように体を動かすかを、隠れマルコフモデル (HMM) によって推定するシステムを提案する。システムの構築にあたって、ダンスゲームにおける全身動作を捉えるために必要なパラメータを提案する。提案するパラメータは、重心移動距離などの定量的なもののみではなく、プレイヤーごとの異なるプロファイルを表現するためのプレイヤー固有パラメータも含めている。また、身体姿勢の安定を考慮し、ゲーム譜面の指示を遂行する上で必須ではない足の位置も含めたモデル化手法を提案する。

本章で提案する、ゲーム譜面に対するプレイヤーの動作を自動推定するシステムは、ダンスゲームを遊ぶプレイヤーと、ダンスゲーム制作者の双方において利用価値が存在すると考える。プレイヤーにとっての利点として、お手本となる動作を提示することにより、プレイヤーの身体動作習得支援を行えることが挙げられる。これは特に、譜面から適切な動作をイメージし実行する練習を行う初心者にとって、有効な支援と考えられる。制作者にとっては、制作した譜面をテストプレイする過程を、提案システムによって補助することが可能である。譜面制作時に想定したターゲットプレイヤーに対応したプレイヤー固有パラメータを設定し、制作者が想定した身体動作系列とシステムの出力動作系列が一致するかを確認することで、実プレイヤーによるテストプレイを介さずに譜面の妥当性を検証できる。このテストプレイの補助という利用価値は、手作業によるゲーム譜面制作のみではなく、ゲーム譜面の自動生成技術の発展にも貢献できる。ニューラルネットワークを用いてゲーム譜面を自動で生成する試みは以前より報告されている [3] が、生成された譜面がプレイヤーにとって破綻なく遊べる譜面であるかどうかを自動で評価する研究はこれまで存在していなかった。本章の提案システムを用いて、自動生成された譜面を対象に身体動作を推定した結果に基づいて、生成譜面の評価値を自動で与えることが可能になると考えられる。これによって将来的に、譜面の制作とテストプレイを繰り返してより良い譜



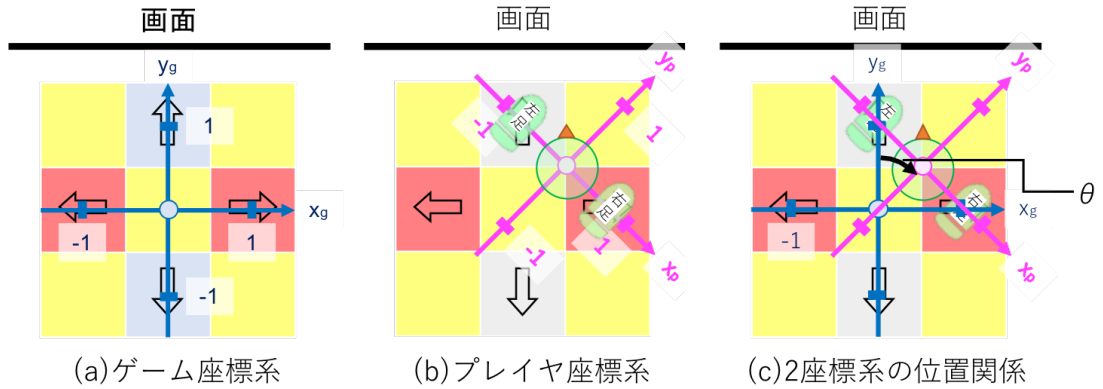


図 4.1: 本章で用いる座標系の模式図. (a): ゲーム座標系. (b): プレイヤ座標系. ↑パネルと→パネルの間にある大きな丸は頭の位置を示し, その上に接している三角形は目線の方角を示す. プレイヤの位置は一例である. (c): (a) と (b) の位置関係.  $\theta$  は  $y_g$  と  $y_p$  の間の角である. 図中では  $|\theta|$  が小さくなる方の角のみを示している.

面にするという, ゲーム制作におけるサイクルを, 計算機上で再現することが可能になると考えられる. 提案システムはプレイヤー固有パラメータの条件下で最尤の身体動作を推定するため, 提案システムによる評価を自動生成技術に組み込むことで, ターゲットプレイヤーが望む譜面を自動で生成することが実現可能となる.

## 4.2 用語定義

プレイヤーが立つステージ上に, 平面直交座標のゲーム座標系を定義する. 図 4.1 の (a) に, 操作パネルとゲーム座標系の模式図を示す. ステージの中央を原点とし, 原点から→パネルの中央に向かうベクトルを  $x$  軸正方向, 原点から↑パネルの中央に向かうベクトルを  $y$  軸正方向と定め, それぞれ  $x_g$  軸,  $y_g$  軸と表記する. 各パネルの 1 辺の長さを 1 とし, 座標上の距離を表現する. 原点から各パネル中央までの距離は, 全て 1 となる. プレイヤの足の位置は, ゲーム座標系上に存在する点として表現し, 左足の座標を  $(x_{g-l}, y_{g-l})$ , 右足の座標を  $(x_{g-r}, y_{g-r})$  とする. それぞれの足の位置は,  $\{(0, 0), (-1, 0), (0, -1), (0, 1), (1, 0)\}$  のいずれかの座標を取るものとする. すなわち, 足が存在する点は, ステージ全体の中央, または 4 方向いずれかのパネルの中央の, 計 5 点のいずれかに限定する.

プレイヤーの両足の位置を結ぶ線分の中点を, プレイヤの重心と定める. すなわち, ゲーム座標系におけるプレイヤーの重心の座標を  $(x_{g-o}, y_{g-o})$  とすると,

$$x_{g-o} = \frac{x_{g-l} + x_{g-r}}{2}, \quad y_{g-o} = \frac{y_{g-l} + y_{g-r}}{2},$$

が成り立つ. プレイヤの重心を原点として, プレイヤから見た前後左右を表現するプレイヤー座標系を定義する. 図 4.1 の (b) に, プレイヤ座標系の模式図を示す. プレイヤ座標系は, ゲーム座標系と同

一平面上に存在する直交座標系である。原点であるプレイヤーの重心から右足位置に向かうベクトルを  $x$  軸正方向とし、 $x$  軸を原点中心に  $90^\circ$  回転させた軸を  $y$  軸と定め、それぞれ  $x_p$  軸、 $y_p$  軸と表記する。距離の単位はゲーム座標系と同じく、パネルの 1 辺の長さを 1 とする。 $y_p$  軸正方向は、プレイヤーのへそが向いている方向と一致する。

ゲーム譜面が表示される画面は、ゲーム座標の  $y_g$  軸正方向の先に存在する。ゲームをプレイしている間、プレイヤーは原則として画面上の譜面を視認し続ける。したがって、プレイヤーの目線の方向は、常に  $y_g$  軸正方向を向いていると仮定する。一方で、プレイヤーのへその方向、すなわちプレイヤー座標系の  $y_p$  軸正方向は、必ずしもゲーム座標系の  $y_g$  軸正方向と一致していない。ゲーム座標系をプレイヤー座標系に変換する際の回転量を、体の向き  $\theta$  とする。 $\theta$  の定義域は  $-360^\circ < \theta < 360^\circ$  とする。すなわち、ゲーム座標系を正方向に回転させてプレイヤー座標系に変換する場合と、負方向に回転させて変換する場合の両方を考慮する。 $\theta$  は図 4.1 の (c) における、 $y_g$  軸と  $y_p$  軸の間の角である。 $\theta$  は、目線の方向とへその方向の間の角度、すなわちプレイヤーの首の旋回角に相当する。

### 4.3 有向グラフによるプレイヤーの姿勢および動作の表現

プレイヤーの身体動作を有限有向グラフとして表現した。図 4.2 に、構築したグラフの抜粋を示す。グラフ内の各ノードは、プレイヤーの姿勢とパネルの反応状況を、(左足位置, 右足位置, 体の向き, パネルオンセット) の 4 要素で表現している。パネルオンセットは、厳密にはエッジ遷移の結果として発生する情報であるが、本章では以降の計算の簡単化のためノードの情報として保持している。ステージ中央ほどのパネルでもないが、便宜上パネルオンセットが発生するものとする。また、ノード間の有向エッジは、プレイヤーの姿勢の遷移、すなわちプレイヤーの動作を表現する。構築したグラフは、ノード数が 126、エッジ数が 7,428 となった<sup>1</sup>。グラフ上のパスを探索させ、得られた遷移経路から、プレイヤーの身体動作系列を復号することが可能である。

#### 4.3.1 ノード：姿勢

ノード  $n$  は、左足のゲーム座標  $(x_{g-l_n}, y_{g-l_n})$ 、右足のゲーム座標  $(x_{g-r_n}, y_{g-r_n})$ 、体の向き  $\theta_n$ 、およびパネルオンセットの情報を持つ。

ノード  $n$  に対して、「該当の姿勢で静止した際に、プレイヤーの身体がどの程度安定しているか」を表す評価値  $S_n$  を付与する。ここで、ある静止姿勢において安定するかどうかは、プレイヤーのゲームの習熟度などによって異なる。例えば、足元を時々確認しながらでない満足にゲームを遊べない初心者にとっては、画面と反対方向に位置する ↓ パネルが特に踏みづらいということが指摘されている [29] ため、↓ パネルに足が存在するノードの評価値は、↓ パネルに足が存在しないノードの評価値より低

<sup>1</sup>直接の遷移が不可能なノードの組の間にはエッジが存在しないため、エッジ数は  $126^2 = 15,876$  より少なくなる。エッジが存在しない例として、「左足を ← パネルから ↑ パネルに移動させているのに、↑ パネルのオンセットが発生していない」などが挙げられる。

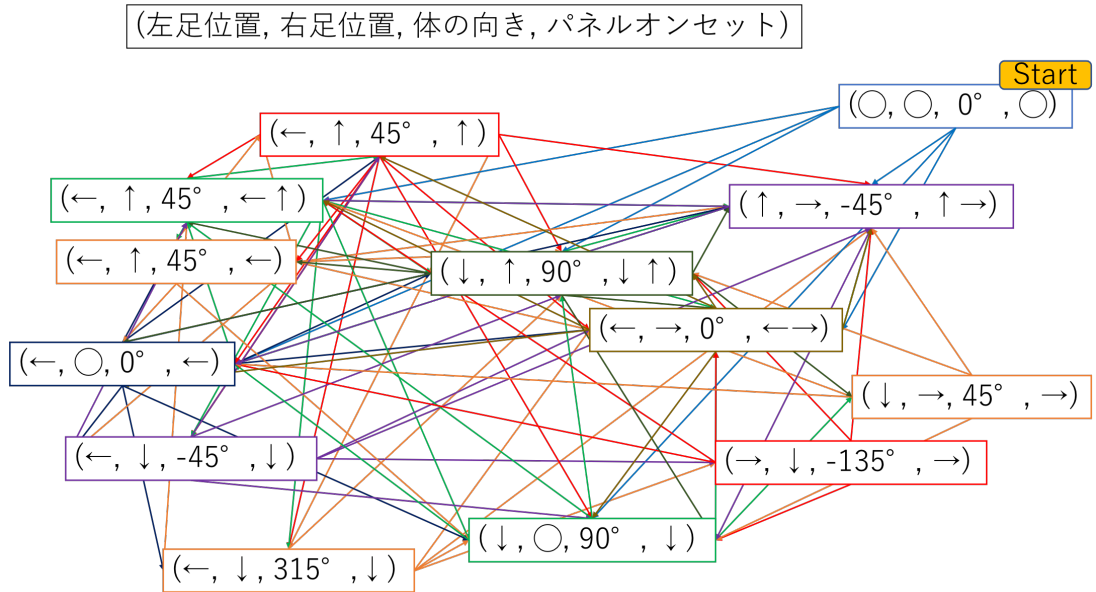


図 4.2: プレイヤの姿勢と動作を表現する有向グラフ (抜粋). 図中の長方形がノード, ノード間を結ぶ矢印がエッジである. 図中では, 「左足位置」「右足位置」「パネルオンセット」を, それぞれ $\leftarrow$ ,  $\downarrow$ ,  $\uparrow$ ,  $\rightarrow$ ,  $\circ$  (中央) の 5 種の記号で表現している.

表 4.1: ノード評価値に関わるプレイヤー固有パラメータ一覧. 表中の「足位置把握度」は「左足位置把握度」と「右足位置把握度」の両方を指す.

パラメータ	範囲	説明	対応する評価項目	備考
$PG_{\sigma^2}$	$(0, \infty]$	正規分布の分散の値	足位置把握度	値が大きいほど, どの位置に対しても把握度が高いことを意味する.
$PG_{.s}$	$[0, 1]$	足位置把握度の最小値	足位置把握度	
$PG_{.w}$	$(0, 1]$	足位置把握度の最大値	足位置把握度	
$PT$	$(0, 1]$	両足の存在軸が一致しない姿勢における安定度	両足存在軸の不一致	
$P\theta_{\sigma^2}$	$(0, \infty]$	正規分布の分散の値	体の向きの影響	値が大きいほど, どの向きにおいても姿勢が安定していることを意味する.
$P\theta_{.w}$	$(0, 1]$	体の向きの影響の最大値	体の向きの影響	
$P\theta_L$	$(0^\circ, 360^\circ]$	首を回旋できる角度の限界値	体の向きの影響	文献 [30, 31] より, 現実的にとりうる値の範囲は $[150^\circ, 210^\circ]$ 程度

くなるべきである. ゲームに慣れた上級者にとっては, 意識しなくとも $\downarrow$ パネルを踏むことができるため,  $\downarrow$ パネルに足が存在するノードの評価値は, 他のノードの評価値と大きく変わらないはずである. このように, 各ノードに与えるべき評価値はプレイヤー個人ごとに異なる. このようなプレイヤー個人の違いを表現するため, いくつかのプレイヤー固有パラメータを設定した. 表 4.1 に, ノード評価値の算出に関わるプレイヤー固有パラメータを列挙する. プレイヤー固有パラメータには, 対象とするプレイヤーのプロファイルに沿うように, 適切な値を手手で代入する. プレイヤー固有パラメータおよびノード評価値算出式は, ダンスゲームに関する書籍 [29] 等を参考に, 著者の知見に基づき設定した.

静止姿勢の安定度に影響する要素として以下の 4 項目を定め, ノード  $n$  におけるそれぞれの項目別評価値を得る式を定義した. 各項目は, 「各パネルの位置は,  $\uparrow$ パネルが最も把握しやすく,  $\downarrow$ パネルが最も把握しにくい [29]. 位置をうまく把握できていないパネルの上に足がある姿勢は不安定である.」 「片足が $\downarrow$  $\uparrow$ どちらかのパネル, もう片足が $\leftarrow$  $\rightarrow$ どちらかのパネルにあるような姿勢<sup>2</sup>は不安定

<sup>2</sup>文献 [29] において「90 度踏み」という用語で言及されている姿勢.

である。」「首旋回角が大きいほど姿勢は不安定であり、その角度には限界がある [30, 31].」という仮定に基づいて設定した。

**左足位置把握度  $S_{l_n}$**  左足が置かれているパネルの位置を、プレイヤーがどの程度把握できているか。

ゲーム座標系における  $y_g$  軸正方向のベクトル  $(0, 1)$  と左足存在点の位置ベクトル  $(x_{g-l_n}, y_{g-l_n})$  がなす小さい方の角を  $\alpha_{l_n}$ 、平均  $\mu$  分散  $\sigma^2$  の正規分布の確率密度関数を  $f_{N(\mu, \sigma^2)}(x)$  とし、式 (4.1) で算出する。

$$S_{l_n} = \left( \frac{f_{N(0, PG_{\sigma^2})}(\alpha_{l_n})(1 - PG_{-s})}{f_{N(0, PG_{\sigma^2})}(0)} + PG_{-s} \right) PG_{-w}. \quad (4.1)$$

値域は  $PG_{-s} < S_{l_n} \leq PG_{-w}$  で、値が大きいほど正しく把握できていることを示す。

**右足位置把握度  $S_{r_n}$**  右足が置かれているパネルの位置を、プレイヤーがどの程度把握できているか。

ゲーム座標系における  $y_g$  軸正方向のベクトル  $(0, 1)$  と、右足存在点の位置ベクトル  $(x_{g-r_n}, y_{g-r_n})$  がなす小さい方の角を  $\alpha_{r_n}$  とし、式 (4.2) で算出する。

$$S_{r_n} = \left( \frac{f_{N(0, PG_{\sigma^2})}(\alpha_{r_n})(1 - PG_{-s})}{f_{N(0, PG_{\sigma^2})}(0)} + PG_{-s} \right) PG_{-w}. \quad (4.2)$$

値域は  $PG_{-s} < S_{r_n} \leq PG_{-w}$  で、値が大きいほど正しく把握できていることを示す。

**両足存在軸の不一致  $S_{t_n}$**  片足が  $x_g$  軸上、もう片足が  $y_g$  軸上に存在する姿勢において、どの程度安定するか。式 (4.3) に基づき、 $\{PT, 1\}$  のどちらかの値を与える。

$$S_{t_n} = \begin{cases} PT & (x_{g-l_n} \neq x_{g-r_n} \text{ and } y_{g-l_n} \neq y_{g-r_n}) \\ 1 & (\text{otherwise}). \end{cases} \quad (4.3)$$

値が 1 に近いほど、該当する姿勢において安定していることを示す。

**体の向きの影響  $S_{\theta_n}$**  体の向き  $\theta_n$  において、どの程度姿勢が安定するか。式 (4.4) で算出する。

$$S_{\theta_n} = \begin{cases} \frac{f_{N(0, P\theta_{\sigma^2})}(\theta_n)P\theta_{-w}}{f_{N(0, P\theta_{\sigma^2})}(0)} & (|\theta_n| < P\theta_{-l}) \\ 0 & (|\theta_n| \geq P\theta_{-l}). \end{cases} \quad (4.4)$$

値域は  $0 \leq S_{\theta_n} \leq P\theta_{-w}$  で、値が大きいほどその  $\theta$  において姿勢が安定していることを示す。

ノード  $n$  の最終的な評価値  $S_n$  は、上記 4 種の項目別評価値の総積、すなわち

$$S_n = S_{l_n} \cdot S_{r_n} \cdot S_{t_n} \cdot S_{\theta_n},$$

で算出する。

表 4.2: エッジ遷移確率に関わるプレイヤー固有パラメータ一覧. 表中の「足移動方向把握度」は「左足移動方向把握度」と「右足移動方向把握度」の両方を指す.

パラメータ	範囲	説明	対応する評価項目	備考
$PM_{\sigma^2}$	$(0, \infty]$	正規分布の分散の値	足移動方向把握度	値が大きいほど、どの方向に対しても把握度が高いことを意味する.
$PM_{-s}$	$[0, 1]$	足移動方向把握度の最小値	足移動方向把握度	
$PM_{w}$	$(0, 1]$	足移動方向把握度の最大値	足移動方向把握度	
$PL$	$(0, \infty]$	左足移動距離の重み	左足移動距離	
$PR$	$(0, \infty]$	右足移動距離の重み	右足移動距離	
$PO$	$(0, \infty]$	重心移動距離の重み	重心移動距離	
$PA$	$(0, \infty]$	体の向き変化量の重み	体の向き変化量	
$PJ$	$[1, \infty]$	ジャンプ動作に対する抵抗	両足同時動作	値が大きいほど該当の動作を苦手とすることを意味する.
$PS$	$[1, \infty]$	同一パネル上の足の入れ替え動作に対する抵抗	同一パネル上の足の入れ替え	値が大きいほど該当の動作を苦手とすることを意味する.

#### 4.3.2 エッジ：動作

エッジ  $e$  は、始点となるノードと終点となるノードの情報を保持する有向エッジである. 各エッジは、始点ノードの姿勢から終点ノードの姿勢への変化、すなわちプレイヤーの動作を表現している.

エッジ  $e$  に対して、「プレイヤーが該当の動作を問題なく実行できる確率」を意味する遷移確率  $P_e$  を付与する. ここにおける「動作を実行できる確率」も、ノード評価値と同様に、プレイヤーのゲーム習熟度や動作の好みなどによって異なる. 例えば、利き足が右足のプレイヤーは、右足の動作を好み、右足を大きく動かす動作に対応するエッジの遷移確率の方が、左足を大きく動かす動作の遷移確率よりも高くなるはずである. したがって、エッジ遷移確率の計算においても、いくつかのプレイヤー固有パラメータを設定した. 表 4.2 に、エッジ遷移確率の算出に関わるプレイヤー固有パラメータの一覧を示す. ノード評価値の計算と同様、プレイヤー固有パラメータの値は人手で適切に設定する. エッジ遷移確率の算出式およびプレイヤー固有パラメータも、著者の知見に基づいて設定した.

動作の成功確率に影響する要素として以下の 8 項目を定め、エッジ  $e$  におけるそれぞれの項目別評価値を得る式を定義した. 各項目は、「プレイヤーにとって、前方に足を動かすことは自然な動作だが、後方に足を動かすことは難しい動作である.」「足および重心が大きく動くほど難しい動作である.」「体の向きが大きく変化する動作は難しい動作である.」「片足ずつ動かすよりも、両足でジャンプする方が難しい動作である.」「両足で同一のパネルを連打する動作は、先に踏んでいた足をしっかりと上げないと、後から踏んだ足でパネルオンセットが発生しないため、もも上げのような特殊な動作が必要になり難しい.」という仮定に基づいて設定した.

左足移動方向把握度  $M_{dl_e}$  左足を移動させる方向に対するプレイヤーの自信. プレイヤー座標系において、

$y_p$  軸正方向のベクトルを  $\mathbf{y}_p = (0, 1)$ , 始点ノードにおける左足の位置ベクトルを  $\mathbf{l}_{(e,s)} = (x_p-l_s, y_p-l_s)$ , 終点ノードにおける左足の位置ベクトルを  $\mathbf{l}_{(e,g)} = (x_p-l_g, y_p-l_g)$  とした時、ベクトル  $\mathbf{y}_p$  とベクトル  $(\mathbf{l}_{(e,g)} - \mathbf{l}_{(e,s)})$  がなす小さい方の角を  $\beta l$  として、式 (4.5) で算出する.

$$M_{dl_e} = \left( \frac{f_{N(0, PM_{\sigma^2})}(\beta l)(1 - PM_{-s})}{f_{N(0, PM_{\sigma^2})}(0)} + PM_{-s} \right) PM_w. \quad (4.5)$$

値域は  $PM_{-s} < M_{dl_e} \leq PM_w$  で、値が大きいほど自信を持って左足を移動できる方向であることを示す.

右足移動方向把握度  $M\_dr_e$  右足を移動させる方向に対するプレイヤーの自信。プレイヤー座標系において、 $y_p$  軸正方向のベクトルを  $\mathbf{y}_p = (0, 1)$ 、始点ノードにおける右足の位置ベクトルを  $\mathbf{r}_{(e,s)} = (x_p-r_s, y_p-r_s)$ 、終点ノードにおける右足の位置ベクトルを  $\mathbf{r}_{(e,g)} = (x_p-r_g, y_p-r_g)$  とした時、ベクトル  $\mathbf{y}_p$  とベクトル  $(\mathbf{r}_{(e,g)} - \mathbf{r}_{(e,s)})$  がなす小さい方の角を  $\beta_r$  として、式 (4.6) で算出する。

$$M\_dr_e = \left( \frac{f_{N(0, PM\_s^2)}(\beta_r)(1 - PM\_s)}{f_{N(0, PM\_s^2)}(0)} + PM\_s \right) PM\_w. \quad (4.6)$$

値域は  $PM\_s < M\_dr_e \leq PM\_w$  で、値が大きいほど自信を持って右足を移動できる方向であることを示す。

左足移動距離  $M\_ml_e$  ゲーム座標系上で計測した左足の移動距離。始点ノードにおける左足の座標を  $(x_g-l_s, y_g-l_s)$ 、終点ノードにおける左足の座標を  $(x_g-l_g, y_g-l_g)$  として、式 (4.7) で算出する。

$$M\_ml_e = PL \sqrt{(x_g-l_g - x_g-l_s)^2 + (y_g-l_g - y_g-l_s)^2}. \quad (4.7)$$

右足移動距離  $M\_mr_e$  ゲーム座標系上で計測した右足の移動距離。始点ノードにおける右足の座標を  $(x_g-r_s, y_g-r_s)$ 、終点ノードにおける右足の座標を  $(x_g-r_g, y_g-r_g)$  として、式 (4.8) で算出する。

$$M\_mr_e = PR \sqrt{(x_g-r_g - x_g-r_s)^2 + (y_g-r_g - y_g-r_s)^2}. \quad (4.8)$$

重心移動距離  $M\_mo_e$  ゲーム座標系上で計測したプレイヤー重心の移動距離。始点ノードにおけるプレイヤー重心の座標を  $(x_g-o_s, y_g-o_s)$ 、終点ノードにおけるプレイヤー重心の座標を  $(x_g-o_g, y_g-o_g)$  として、式 (4.9) で算出する。

$$M\_mo_e = PO \sqrt{(x_g-o_g - x_g-o_s)^2 + (y_g-o_g - y_g-o_s)^2}. \quad (4.9)$$

体の向き変化量  $M\_m\theta_e$  体の向きの変化量。始点ノードにおける体の向きを  $\theta_s$ 、終点ノードにおける体の向きを  $\theta_g$  として、式 (4.10) で算出する。

$$M\_m\theta_e = PA |\theta_g - \theta_s|. \quad (4.10)$$

両足同時動作  $M\_j_e$  両足で同時にパネルオンセットを発生させる動作、すなわちジャンプ動作に対するプレイヤーの自信。終点ノードで発生しているパネルオンセットの数を  $p$  として、式 (4.11) に基づき  $M\_j_e = \{1, PJ\}$  のどちらかの値を与える。

$$M\_j_e = \begin{cases} PJ & (p = 2) \\ 1 & (p < 2). \end{cases} \quad (4.11)$$

値が 1 に近いほど、該当の動作に対して自信があることを示す。

同一パネル上の足の入れ替え  $M_{s_e}$  始点ノードにおける一方の足が存在するパネルの上に、終点ノードでは他方の足が存在するような動作におけるプレイヤーの自信. ゲーム座標系において、始点ノードの左足座標を  $(x_{g-l_s}, y_{g-l_s})$ , 右足座標を  $(x_{g-r_s}, y_{g-r_s})$ , 終点ノードの左足座標を  $(x_{g-l_g}, y_{g-l_g})$ , 右足座標を  $(x_{g-r_g}, y_{g-r_g})$  とし、式 (4.12) に基づき  $M_{s_e} = \{1, PS\}$  のどちらかの値を与える.

$$M_{s_e} = \begin{cases} PS & ((x_{g-l_s}, y_{g-l_s}) = (x_{g-r_g}, y_{g-r_g})) \\ PS & ((x_{g-r_s}, y_{g-r_s}) = (x_{g-l_g}, y_{g-l_g})) \\ 1 & (\text{otherwise}). \end{cases} \quad (4.12)$$

値が 1 に近いほど、該当の動作に対して自信があることを示す.

上記 8 項目の項目別評価値を用いて、式 (4.13) でエッジ評価値  $M_e$  を算出する.

$$M_e = \frac{1}{\log(T_e)}, \quad (4.13)$$

ただし、

$$T_e = -M_{dl_e} - M_{dr_e} + M_{ml_e} + M_{mr_e} + M_{mo_e} + M_{m\theta_e} + M_{j_e} + M_{s_e}.$$

$e$  と始点ノードが同一であるエッジの集合 ( $e$  自身を含む) を  $E$  とし、式 (4.14) で最終的なエッジ遷移確率  $P_e$  を求める.

$$P_e = \frac{M_e}{\sum_{k \in E} M_k}. \quad (4.14)$$

#### 4.4 隠れマルコフモデルを用いたプレイヤー動作の推定

4.3 節で構築した有限有向グラフに基づいて、指定されたゲーム譜面に対してプレイヤーの身体動作を推定するモデルを構築する. 吉永らによるギター譜面に対する運指推定手法 [27] にならい、ゲーム譜面内のステップ系列を観測系列、グラフ上のノードで表されたプレイヤーの姿勢を隠れ状態とすることによって、隠れマルコフモデル (HMM) を構築した. 図 4.3 に、観測系列となるゲーム譜面例の一部と、それに対応する姿勢ノードを一部抜粋した図を示す. プレイヤ固有パラメータに代入した値によって、異なる遷移確率および出力確率を持つ HMM が形成される. したがって、1 つの HMM は、プレイヤー固有パラメータの値から想定される、あるプレイヤー像に対応する.

ノード  $n$  に対応する姿勢から  $k$  番目のステップ  $s_k$  が出力される確率  $o_{(n,k)}$  は、4.3.1 節に基づき、 $n$  が保持するパネルオンセット情報がステップの指示を満たす場合  $o_{(n,k)} = S_n$ , そうではない場合  $o_{(n,k)} = 0$  と定めた. また、ノード間の状態遷移確率については、4.3.2 節で求めた  $P_e$  の値を採用した.

以上の確率設定のもとで、指定されたゲーム譜面に対するプレイヤー身体動作系列の推定は、ゲーム譜面内のステップ系列を出力する確率が最尤となる、ノード遷移の系列を求める復号化問題に帰着できる. この問題は Viterbi アルゴリズムによって解くことが可能である. 各確率は表 4.1 および表 4.2

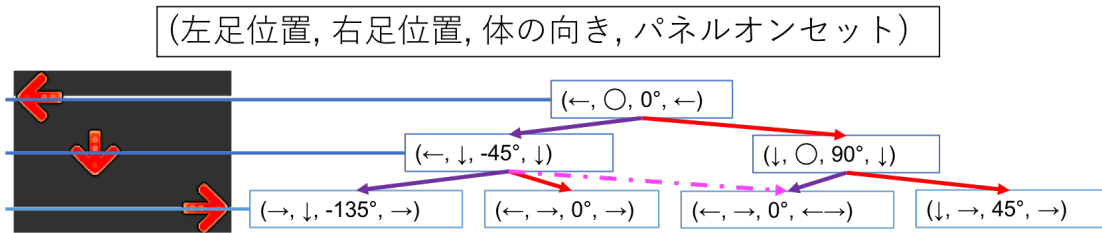


図 4.3: 推定対象譜面 (=観測系列) と姿勢ノード (=隠れ状態) の対応関係 (抜粋). 長方形が姿勢ノードである. 図中では, 「左足位置」「右足位置」「パネルオンセット」を, それぞれ←, ↓, ↑, →, ○ (中央) の 5 種の記号で表現している.

のプレイヤー固有パラメータに代入した値によって変化するため, 指定されたゲーム譜面を出力する最尤ノード遷移系列もプレイヤー固有パラメータの値によって変化する. したがって, 構築した HMM を用いたプレイヤーの身体動作推定結果は, プレイヤー固有パラメータの値から想定されるプレイヤーのゲーム習熟度と嗜好を反映したものであると言える. 1 つの譜面に対して複数の HMM で身体動作推定を行うことで, プレイヤー像ごとに異なる推定結果を得ることができる.

## 4.5 推定結果例

### 4.5.1 条件設定

4.4 節で構築した HMM を用いて, ゲーム譜面に対するプレイヤー身体動作を推定した. プレイヤー固有パラメータの値は, 表 4.3 に示す 4 組の値を用いた. 各パラメータ組は, 基準となる standard, 右足利きのプレイヤーを想定した right, 初心者プレイヤーを想定した beginner, あらゆる身体動作を厭わない perfect の 4 種類のプレイヤー像を想定して設定した. 以下, それぞれのパラメータ組を与えられたモデルで動作推定を行うエージェントを, それぞれ standard エージェント, right エージェント, beginner エージェント, perfect エージェントとする.

ITG データセット内の譜面 *Bumble Bee* (Medium) を対象とし, 上記 4 エージェントでプレイヤー動作を推定させた. 図 4.4 に対象譜面の抜粋を示し, 図 4.5 に同区間の standard, right, beginner, perfect それぞれのエージェントの推定結果の模式図を示す. また, 表 4.4 に, 推定結果の代表値として, 得られたノード遷移系列に対して下記 5 項目 8 種の値を計算した結果を示す.

**パネルオンセット関与回数** 左足・右足それぞれが発生させたパネルオンセットの回数. ステップによって指示されていないパネルオンセットも含む. 両足で同時にパネルオンセットを発生させた場合は, 左右両方でカウントしている.

**ジャンプ回数** 両足で同時にパネルオンセットを発生させた回数. 上記のパネルオンセット関与回数において, 左右両方でカウントされた回数と言い換えることが可能である.



表 4.3: 4 種類のエージェントおよびそれぞれのプレイヤー固有パラメータの値

	standard	right	beginner	perfect
$PG_{\sigma^2}$	90 <sup>2</sup>	90 <sup>2</sup>	90 <sup>2</sup>	<b>10,000<sup>2</sup></b>
$PG_s$	0.9	0.9	<b>0.5</b>	1
$PG_w$	1	1	<b>0.5</b>	1
$PT$	1	1	<b>0.7</b>	1
$P\theta_{\sigma^2}$	10,000 <sup>2</sup>	10,000 <sup>2</sup>	<b>100<sup>2</sup></b>	10,000 <sup>2</sup>
$P\theta_w$	1	1	1	1
$P\theta_l$	210°	210°	<b>180°</b>	<b>360°</b>
$PM_{\sigma^2}$	90 <sup>2</sup>	90 <sup>2</sup>	90 <sup>2</sup>	<b>10,000<sup>2</sup></b>
$PM_s$	1	1	1	1
$PM_w$	1	1	1	1
$PL$	1	<b>100</b>	1	1
$PR$	1.1	1	1.1	1
$PO$	2	2	2	<b>1</b>
$PA$	1/180	1/180	1/180	<b>1/720</b>
$PJ$	10,000	10,000	10,000	<b>1</b>
$PS$	100	100	<b>10,000</b>	<b>1</b>

表 4.4: *Bumble Bee* (Medium) に対する各エージェント推定結果の代表値

		standard	right	beginner	perfect
パネルオンセット関与回数	左足	90	74	122	96
	右足	84	<b>102</b>	102	96
ジャンプ回数		27	29	<b>77</b>	<b>45</b>
移動距離合計	左足	74.40	39.77	<b>121.85</b>	64.40
	右足	54.84	<b>88.30</b>	<b>111.95</b>	67.81
	重心	55.55	53.55	<b>97.25</b>	<b>46.47</b>
最大の絶対値を与える体の向き		-90°	-135°	-90°	-90°
体の向き変化量合計		4,320°	4,140°	<b>2,520°</b>	3,510°

**移動距離** 左足移動距離合計は、通過したエッジの  $M_{ml_e}$  の総和を表す。同様に、右足移動距離合計は  $M_{mr_e}$  の総和を、重心移動距離合計は  $M_{mo_e}$  の総和を表す。

**最大の絶対値を与える体の向き** 通過したノードのうち、 $|\theta_n|$  が最大となるノードの  $\theta_n$ 。

**体の向き変化量合計** 通過したエッジの  $M_{m\theta_e}$  の総和。

上記の値は、モデルの性能を評価しているわけではなく、プレイヤー固有パラメータの設定と出力結果に齟齬がないかを確認するために計算した値である。

#### 4.5.2 考察

図 4.5 中において、standard エージェントの推定結果は右足が→パネルから移動していない一方で、right エージェントの推定結果は左足が↑パネルから移動していないことがわかる。特に right エー

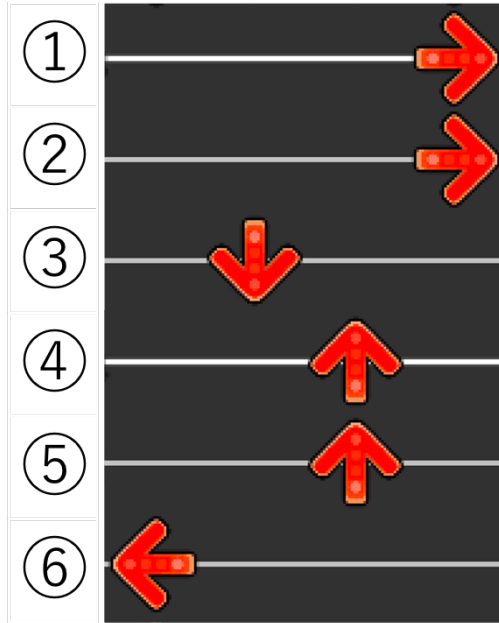


図 4.4: 推定対象とした *Bumble Bee* (Medium) の一部. 推定時にステップ間の時間間隔情報は用いていないため, ステップ間隔は等間隔に修正している. 左側の番号は図 4.5 内の番号と対応する.

エージェントの 6 歩目では, 右足で←パネルのオンセットを発生させている. この姿勢における体の向きは  $-135^\circ$  と, 4 エージェント全ての推定結果中で最大の  $|\theta_n|$  を示した. standard エージェントと right エージェントの推定結果は, プレイヤ固有パラメータの  $PL$  と  $PR$  が意図通り作用したと考えられる. 表 4.4 から, standard エージェントと比較した right エージェントの推定結果は, 右足のパネルオンセット関与回数と移動距離合計の値が大きいことがわかる.

beginner エージェントの推定は, 図 4.5 中の 3 歩目から 5 歩目のように, 1 つのパネルを両足で同時に踏む動作を多く含む結果となった. これは, プレイヤ固有パラメータの  $PT$  および  $P\theta_{\sigma^2}$  が作用した結果, ↓と↑のパネルを体の向き  $0^\circ$  で踏む動作を推定したためと考えられる. 表 4.4 を見ると, ジャンプ回数および各種移動距離合計の値が他エージェントより大きく, 体の向き変化量合計が他エージェントより小さい値を示した. 各種移動距離合計が大きい点については, 体の向きの変化をなるべく避けた結果余計な動作が増えている, すなわち身体動作が最適化されていないと解釈でき, 想定した初心者プレイヤーの特徴と合致すると言える. 一方で, 特殊動作であり難易度が高いはずのジャンプ回数が多い点は, 初心者プレイヤーという想定において, 直感に反する結果である. プレイヤ固有パラメータ  $PJ$  の設定値も含めて, 実際の初心者プレイヤーの動作と比較した妥当性の検証が必要と考える.

perfect モデルの推定結果は, 表 4.4 より, ジャンプ回数が beginner モデルに次いで多い一方で, 重心移動距離合計は他のどのモデルよりも小さくなった. 図 4.5 中では 4 歩目でジャンプ動作を行なっている. この動作は, 後の 6 歩目で指示される←のステップに先んじて, あらかじめ左足を←パネル上に移動させておくことを意図した推定と考えられる. プレイヤ固有パラメータを  $PJ = 1$  に設定し

	standard	right	beginner	perfect
①				
②				
③				
④				
⑤				
⑥				

図 4.5: *Bumble Bee* (Medium) に対する各エージェント推定結果 (抜粋). 左側の番号は図 4.4 内の番号と対応する. 着色されたパネルは, そのパネルでパネルオンセットが発生していることを示す. ステージ中央はどのパネルでもないが, 便宜上パネルオンセットが発生するものとしている.

た結果, このような先回りのためのジャンプ動作が増加し, 結果として重心移動距離を抑える推定結果になったと考察できる.

## 4.6 おわりに

本章では, ダンスゲーム譜面に対するプレイヤーの動作を捉えるために必要なパラメータを提案し, HMM によってゲーム譜面に対するプレイヤーの動作を推定するモデルを構築した. プレイヤー固有パラメータの値をヒューリスティックに変更することで, 推定される動作系列が変化することを確認し, 異なるプレイヤー像に対応した推定結果を得られることが示唆された.

今後の課題として, ダンスゲームのプレイヤーを対象としたアンケート調査を実施し, 推定結果の妥当性を検証することが挙げられる. これと並行して, 今回設定した各種パラメータ, 特にプレイヤー

固有パラメータの精査も進めたい。表 4.1 および表 4.2 に示したプレイヤー固有パラメータは、どちらかといえばダンスゲームを熟知した譜面制作者の視点から設定したものであり、4.1 節で対象ユーザ例とした初心者プレイヤーにとって分かりやすいものとは言い難い。そのため、例えば身長など、どのプレイヤーにとっても直感的な指標によってプレイヤー固有パラメータを表現することが、提案システムの発展につながると考えられる。また、現時点ではプレイヤー固有パラメータは人手で設定するものとしているが、これをゲームのプレイログなどをもとに自動で設定する手法についても、検討を進めたい。

# 第 5 章

## 結論

### 5.1 本論文のまとめ

#### 5.1.1 音楽と身体動作の認知メカニズムの究明

本論文では、音楽ゲーム譜面の分析を通し、プレイヤーの音楽的・身体的認知メカニズムの究明を推進した。2章では、ゲーム譜面データセットの難易度デザインの分析を行った。同一楽曲に対する難易度が異なる譜面間の関係について、低難易度の譜面におけるステップの配置時刻は、高難易度譜面の配置時刻の部分集合となる傾向があることを解明した。加えて、低難易度譜面に見られるステップ配置時刻の音楽的特徴として、4分層のステップや小節頭のステップが多く見られることを、統計的に示した。3章では、ゲーム譜面の面白さの特性に影響する特徴量を設定し、ゲーム譜面を特性に基づいてk-means法によるクラスタリングを実施した。譜面データセット内には、ある特徴量において外れ値を示し、特性が類似する譜面が少ない例外的な譜面が存在することを確認した。このような譜面を除外した上でクラスタリングを実施し、ステップの頻度、リズムの複雑さ、ジャンプステップの頻度などの特徴に基づいて、特性ごとに譜面を分類できることを確認した。4章では、ゲーム譜面に対するプレイヤーの動作を捉えるために必要なパラメータを提案し、HMMによって譜面に対するプレイヤーの動作を推定するモデルを構築した。ヒューリスティックなプレイヤー固有パラメータを用いることで、プレイヤーごとに異なるプロファイルを表現し、異なるプレイヤー像に対応した推定結果を得られることが示唆された。同一の譜面に対する推定結果を比較することで、初心者想定したパラメータ設定では、体の重心移動が大きく体の向きの変化が少ない動作を推定し、中級者以上を想定したパラメータでは、体の向きの変化が大きく重心移動が小さい動作を推定する傾向があることを確認した。また、同程度の実力を想定しつつ、プレイヤーごとの好みの動作を反映した動作推定が可能であることも示唆された。

以上の知見を統括することにより、低難易度の譜面は、移動量大きい身体動作でも間に合う程度の頻度で、体の向きの変化を必要としない単純な動作を、4分の表拍など楽曲聴取上知覚しやすいタイミングにおいて要求していると考えられる。低難易度譜面は、ゲームに興味を持ち始めたばかりの初心者を対象としており、多くのプレイヤーが一度は遊ぶことになる。したがって、多くの人間は共通して、楽曲に合わせて知覚しやすいリズムで、身体を大きく動かすことが楽しいと認知する傾向があり、低難易度の譜面はこの楽しさを提供できるように作られていると考えられる。

一方で、ゲーム譜面の難易度が上昇するにしたがって、体の向きの変化なども取り入れて身体動作を最適化し、より複雑なリズムを知覚することを要求するようになる。このとき、プレイヤーごとの動

作の好みによって、自身にとって最適となる身体動作は異なることが示唆された。また、低難易度の譜面にはあまり見られなかった、64分のステップなどの特殊なリズムや、ジャンプステップなどの特殊な動作は、譜面に固有の特性を与える作用があると考えられる。以上から、人間が音楽的知識および動作を習得するにしたがって、複雑なリズムに合わせた指示を満たす、自身にとって最適な身体動作がどのようなものであるかを認知できる能力を身につけると考えられる。加えて、動作の頻度、リズムの複雑さ、特殊な動作の存在などに基づいて、譜面に対する得意苦手や好き嫌いなどを評価するようになると考えられる。

### 5.1.2 音楽ゲーム譜面の自動生成

分析結果を応用し、音楽ゲーム譜面を自動生成し、その難易度を調整するモデルを構築した。2章では、同一楽曲に対する高難易度譜面と低難易度譜面の関係性を学習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。モデルへの入力特徴として、高難易度譜面の情報に加えて、拍階層および小節内位置を用いた。提案モデルは、難易度が低い譜面を出力するタスクにおいて、楽曲の音響情報のみを用いた既存モデルよりも概ね良い性能を示した。これにより、特に低難易度譜面の生成における、拍階層および小節内位置の有用性が示唆された。3章では、譜面の面白さの特性に基づいて分類されたクラスタ毎に譜面生成モデルを学習させることで、各クラスタの特性を反映した、異なる特性を持つ譜面が生成できた。各モデルが生成した譜面の特性は、学習したクラスタに含まれる譜面の特性と一致しており、クラスタを問わずに学習したモデルによる生成譜面と比較して、それぞれの特性に指向していることを確認した。学習モデルの構造は既存モデルと同一であり、本来存在するデータ量を減らして学習させるアプローチを採ったが、よりエンタテインメント性の高いコンテンツの生成が可能となることが示唆された。生成結果の分析から、設定した特徴量によるクラスタリングの有用性が示唆された。

## 5.2 今後の展望

本研究の成果によって、よりプレイヤー個人に合わせたコンテンツを提供し、音楽ゲームを教育や健康増進などに利用する取り組みが効果的に推進可能になると考える。音楽ゲームの難易度デザイン分析を通して得た、人間の音楽的知識および身体動作の習得段階の知見は、音楽や体育教育におけるカリキュラムの設定に有用であると考えられる。また、本論文で述べた研究成果を組み合わせることによって、ユーザが好みの楽曲に合わせて、自身の実力に合った難易度で音楽的表現活動を楽しみながら、ユーザに運動を促進できる可能性が示唆された。本論文では、特定の入出力インターフェースを用いたダンスゲームを対象として究明を進めたが、このインターフェース自体も変更することで、ダンスゲームでは表現できなかった動作を含む、特定の動作に指向した運動促進システムを構築可能であると考えられる。エンタテインメント外の分野におけるニーズに合わせたインターフェースの構築、およ

びコンテンツの生成は，研究成果の最終的なアプリケーションの1つである．

一方で，プレイヤーが好みの楽曲に対して，音楽的認知および身体的認知の双方の観点においてプレイヤーが望む特性のゲーム譜面を自動生成することは，エンタテインメントコンピューティングの分野においても非常に有意義であると考えられる．この目標の達成に向けては，本論文での研究成果をより発展させ，ゲームのプレイログを用いた実力および嗜好の推定，動作推定結果を用いた自動生成譜面の評価システムの構築などが必要と考えられる．本論文の成果が，ダンスゲーム，ひいてはエンタテインメントコンピューティングにおける，ユーザに寄り添ったコンテンツ生成技術開発の一助となることを望んでいる．

## 謝辞

博士課程後期課程に在籍する3年間、指導教員として親身にご指導していただいた、立命館大学音声言語研究室の山下 洋一教授に、深く感謝いたします。立命館大学大学院博士課程後期課程進学時に自然言語処理研究室より移籍する、というイレギュラーな形で研究室所属ではありましたが、移籍以前より長きにわたって、研究方針について相談に乗っていただきました。また、一研究者、ひいては一人の人間として未熟であった私を、時に優しく時に厳しくご指導していただきました。山下教授のもとで研究室での生活を通して、高い専門性をもつ「博士」という学位を授与されることの重み、心構えを学ぶことができました。重ねて御礼申し上げます。

自然言語処理研究室に所属した学部生の時より師事し、関西大学に移籍されて以降も大学を超えてご指導していただいた、関西大学総合情報学部の山西 良典准教授に深く感謝いたします。ただの趣味にすぎなかった音楽ゲームが研究になるという可能性を提示し、私がエンタテインメントコンピューティングの研究者を志すきっかけとなったのは、ひとえに山西准教授のおかげでございます。加えて、知識面での支援はもちろんのこと、国内外での学会発表にあたっては経済面での支援も頂きました。時には先生と学生として、そして時にはまるで友達か相棒のように接していただき、貴重な経験をたくさんいただきました。重ねて御礼申し上げます。

博士課程後期課程を含む学生生活の中で、所属研究室の教員として指導して下さった、立命館大学自然言語処理研究室の福本 淳一教授、同志社大学理工学部の井本 桂右准教授、そして立命館大学音声言語研究室の福森 隆寛講師に、深く感謝いたします。また、合同ゼミなどを通してアドバイスをくださった立命館大学対話デザイン研究室の西原 陽子教授に深く感謝いたします。ダンスゲームという、専門性が高く、ともすれば遊んでいるだけで捉えられても仕方がないような研究テーマでありましたが、非常に親身に相談に乗っていただきました加えて、実験実施場所および設備の貸出しなどにも、快くご協力をいただきました。

学生生活の中では、非常に多くの研究室の仲間に支えられました。立命館大学音声言語研究室の皆様はもちろん、自然言語処理研究室、対話デザイン研究室の皆様、そして関西大学文化芸術計算機科学研究室の皆様に感謝いたします。

ダンスゲームを含む音楽ゲームについての情報を共有して下さり、趣味としての音楽ゲームでの私の活動を厚くサポートして下さった、立命館大学登録団体の Ritsumeikan EVOLVED の皆様に感謝いたします。最後になりましたが、9年間にわたる長期の学生生活を、不自由がないよう協力して下さった、家族に感謝いたします。



# 研究業績

## 学術論文

1. 辻野雄大, 山西良典, 西原陽子, 福本淳一, “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,” 情報処理学会論文誌, Vol. 59, No. 11, pp.1953–1964, 2018年11月
2. 辻野雄大, 山西良典, 山下洋一, 井本桂右, “ダンスゲーム譜面の特性分析とクラスタリングに基づく特徴的な譜面の自動生成,” 情報処理学会論文誌, Vol. 61, No. 11, pp.1718–1728, 2020年11月

## 国際会議論文（査読あり）

1. **Yudai Tsujino** and Ryosuke Yamanishi, “Dance Dance Gradation: a generation of fine-tuned dance chart,” International Conference on Entertainment Computing 2018, pp.175–187, Sep. 2018
2. **Yudai Tsujino**, Ryosuke Yamanishi, and Yoichi Yamashita, “Characteristics Study of Dance-charts on Rhythm-based Video Games,” IEEE Conference on Games 2019, pp.157–160, Aug. 2019

## 国内研究発表（査読なし）

1. 辻野雄大, 山西良典, 西原陽子, 福本淳一, “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,” 第46回情報処理学会エンタテインメントコンピューティング研究会, 2017年12月
2. 辻野雄大, 山西良典, 山下洋一, 井本桂右, “ダンスゲーム譜面の特性分析とクラスタリングに基づく特徴的な譜面の自動生成,” Entertainment Computing 2019, 2019年9月
3. 辻野雄大, 山西良典, 山下洋一, “プレイヤーのゲーム習熟度と嗜好を反映したダンスゲーム譜面に対する動作の推定,” Entertainment Computing 2021, 2021年8月

## 特許

1. 山西良典, 辻野雄大, “譜面データ生成モデルの学習装置、譜面データ生成器、システム、譜面データの学習方法、譜面データの生成方法、及びコンピュータプログラム,” 特願 2018-099294 号, 特開 2019-201939 号, 2019 年 11 月 28 日公開
2. 山西良典, 辻野雄大, “演算装置、演算方法、及び、コンピュータプログラム,” 特願 2021-143487 号, 2021 年 9 月 2 日出願

## 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, コンピュータエンターテインメント協会, 2007.
- [2] N. Swinnen, M. Vandenbulcke, E.D. de Bruin, R. Akkerman, B. Stubbs, J. Firth, and D. Vancampfort, “The efficacy of exergaming in people with major neurocognitive disorder residing in long-term care facilities: a pilot randomized controlled trial,” *Alzheimer’s Research & Therapy*, vol.13, no.1, p.70, 2021.
- [3] C. Donahue, Z.C. Lipton, and J. McAuley, “Dance Dance Convolution,” *Proceedings of ICML 2017*, pp.1039–1048, 2017.
- [4] F.T. Liang, M. Gotham, M. Johnson, and J. Shotton, “Automatic stylistic composition of bach chorales with deep LSTM,” *Proceedings of the 18th International Society for Music Information Retrieval Conference, ISMIR 2017*, pp.449–456, 2017.
- [5] L. Yang, S. Chou, and Y. Yang, “Midinet: A convolutional generative adversarial network for symbolic-domain music generation,” *Proceedings of the 18th International Society for Music Information Retrieval Conference, ISMIR 2017*, pp.324–331, 2017.
- [6] 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.
- [7] 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.
- [8] N. Shaker, J. Togelius, and M.J. Nelson, *Procedural Content Generation in Games: A Textbook and an Overview of Current Research*, Springer, 2016.
- [9] C. Pedersen, J. Togelius, and G.N. Yannakakis, “Modeling player experience for content creation,” *IEEE Transactions on Computational Intelligence and AI in Games*, vol.2, no.1, pp.54–67, 2010.
- [10] 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.

- [11] J.P. Bello, L. Daudet, S. Abdallah, C. Duxbury, M. Davies, and M.B. Sandler, “A tutorial on onset detection in music signals,” *IEEE Transactions on speech and audio processing*, vol.13, no.5, pp.1035–1047, 2005.
- [12] J. Schluter, and S. Böck, “Improved musical onset detection with Convolutional Neural Networks,” *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp.6979-6983, 2014.
- [13] Y. Kim, Y. Jernite, D. Sontag, and A.M. Rush, “Character-aware neural language models,” *Proceedings of AAAI 2016*, pp.2741–2749, 2016.
- [14] S. Hochreiter, and J. Schmidhuber, “Long short-term memory,” *Neural computation*, vol.9, no.8, pp.1735–1780, 1997.
- [15] P. Hamel, Y. Bengio, and D. Eck, “Building musically-relevant audio features through multiple timescale representations.,” *Proceedings of the 13th International Society for Music Information Retrieval Conference, ISMIR 2012*, pp.553–558, 2012.
- [16] 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.
- [17] A. Maezawa, H.G. Okuno, T. Ogata, and M. Goto, “Polyphonic audio-to-score alignment based on bayesian latent harmonic allocation hidden markov model,” *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2011*, pp.185–188, 2011.
- [18] R. Hunicke, “The case for dynamic difficulty adjustment in games,” *Advances in Computer Entertainment Technology*, pp.429–433, 2005.
- [19] R.R. Wehbe, E.D. Mekler, M. Schaeckermann, E. Lank, and L.E. Nacke, “Testing incremental difficulty design in platformer games,” *Proceedings of the 2017 CHI*, pp.5109–5113, 2017.
- [20] K. Spiel, S. Bertel, and F. Kayali, ““Not another Z piece!” adaptive difficulty in TETRIS,” *Proceedings of the 2017 CHI*, pp.5126–5131, 2017.
- [21] B.G. Behrenshausen, “Toward a (kin)aesthetic of video gaming: The case of Dance Dance Revolution,” *Games and Culture*, vol.2, no.4, pp.335–354, 2007.
- [22] A.M. Sarroff, and M. Casey, “Musical audio synthesis using autoencoding neural nets,” *Joint 40th International Computer Music Conference (ICMC) and 11th Sound & Music Computing conference (SMC)*, pp.1411–1417, 2014.

- [23] J. Colonel, C. Curro, and S. Keene, “Improving neural net auto encoders for music synthesis,” Audio Engineering Society Convention 143, 2017.
- [24] K. Kobayashi, and K. Pillai, A Handbook of Applied Statistics Pharmacology, CRC Press, 2012.
- [25] P.J. Rousseeuw, and K.V. Driessen, “A fast algorithm for the minimum covariance determinant estimator,” *Technometrics*, vol.41, no.3, pp.212–223, 1999.
- [26] 中村栄太, 齋藤康之, 吉井和佳, “ピアノ運指データを用いた統計学習手法による運指推定と演奏難易度の定式化,” Technical Report 12, 2019.
- [27] 吉永悠真, 深山覚, 嵯峨山茂樹, “隠れマルコフモデルによるギターのための運指決定および自動編曲,” 日本音響学会春季研究発表会講演集, vol.2012, 春季号, pp.1011–1014, 2012.
- [28] G. Hori, H. Kameoka, and S. Sagayama, “Input-Output HMM applied to automatic arrangement for guitars,” *Journal of Information Processing*, vol.21, no.2, pp.264–271, 2013.
- [29] コナミ株式会社, ダンス ダンス レボリューション 2ndMIX パーフェクトガイド, 新紀元社, 1999.
- [30] 畑田豊彦, “人工現実感に要求される視空間知覚特性,” *人間工学*, vol.29, no.3, pp.129–134, 1993.
- [31] 米本恭三, 石神重信, 近藤徹, “関節可動域表示ならびに測定法:(平成7年4月改訂),” *リハビリテーション医学*, vol.32, no.4, pp.207–217, 1995.