

図書館情報メディア研究科修士論文

ドラムアンサンブルにおける視聴覚情報及び
即興演奏の影響 ～情動と演奏タイミング・
リズムパターンに着目して～

2017年3月

201521617

崎野 峻光

ドラムアンサンブルにおける視聴覚情報及び即興演奏の影響
～情動と演奏タイミング・リズムパターンに着目して～
**The influence of audiovisual information and improvisation
in a percussion ensemble: with focus on player's emotion,
performance timing, and rhythm patterns**

学籍番号: 201521617

氏名: 崎野 峻光

Takaaki SAKINO

本研究は、アンサンブル演奏を行う際の視聴覚情報(共演者の姿や演奏音)と即興演奏がもたらす影響に関する調査を行った。実験で視聴覚情報の影響を調べる際は、(1)視覚+聴覚条件、(2)聴覚のみ条件、(3)視覚のみ条件の3条件を用意し、それぞれの条件で体験される情動やアンサンブルの演奏タイミングの不一致度、表出されるリズムパターンに着目し検討した。また、即興演奏の影響を調べる際は、(1)アンサンブル演奏中に即興演奏を行う条件(視覚+聴覚条件と同一)、(2)アンサンブル演奏中に即興演奏を行わない条件の2条件を用意し、それぞれの条件で体験される情動に着目し検討した。情動に関しては、各演奏に対し予め“否定的感情”、“高揚感”、“落ち着き感”の3因子でグループ分けされた感情語12項目の5段階評価を実験参加者に依頼し、因子毎の5段階評価の合計値を条件間で比較した。

分析の結果、演奏者の情動は聴覚情報の有無によって大きく左右されるが、演奏者は聴覚情報だけでなく視覚情報も参考にすることでポジティブな情動をより強く体験していることがわかった。またアンサンブルの演奏タイミングの不一致度や即興演奏時に表出されるリズムパターンの種類の豊富さは視聴覚情報の影響を受けないことがわかった。

また、アンサンブルにおける即興演奏は演奏者に高揚感をもたらすが、人によっては緊張や苦痛を感じることもある、ということが示唆された。

研究指導教員: 平賀 譲

副研究指導教員: 寺澤 洋子

ドラムアンサンブルにおける視聴覚情報及び
即興演奏の影響 ～情動と演奏タイミング・
リズムパターンに着目して～

筑波大学

図書館情報メディア研究科

2017年3月

崎野 峻光

目次

第1章	はじめに	1
1.1	アンサンブルとコミュニケーション	1
1.2	情動とは	1
1.3	視聴覚情報に関する研究	1
1.4	即興演奏に関する研究	2
1.5	本研究の目的	3
1.6	本論文の構成	3
第2章	研究の枠組み	4
2.1	リサーチクエスションと仮説	4
2.2	対象とする演奏	4
2.3	実験に使用する楽曲	5
第3章	実験	7
3.1	実験概要	7
3.2	実験デザイン	7
3.2.1	実験参加者	7
3.2.2	実験環境	7
3.2.3	実験用システム	8
3.2.4	実験条件	10
	視聴覚条件	10
	即興条件	11
3.2.5	実験手順	11
3.2.6	分析方法	13
第4章	実験結果	15
4.1	情動	15
4.1.1	視聴覚条件毎の感情尺度評価値	15
4.1.2	即興条件毎の感情尺度評価値	16
4.2	演奏タイミング	17
4.3	リズムパターン	18
4.4	結果のまとめ	21
4.5	考察	21
4.5.1	情動	21
4.5.2	演奏タイミング	23

4.5.3 リズムパターン	24
第 5 章 まとめと今後の課題	25
謝辞	27
参考文献	28
付 録 A 実験参加者の音楽経験一覧	30
付 録 B 同意書	31
付 録 C 音楽経験と実験参加者の性格調査アンケート	33
付 録 D 情動アンケート	37
付 録 E 実験参加者が即興演奏で表出した リズムパターンの各条件毎の回数	38
付 録 F 外部仕様	42
F.1 動作環境	42
F.2 実行	42
付 録 G 内部仕様	43
G.1 主要なプログラム	43
G.2 各モジュールの主要なファンクション	43
G.3 主要な外部モジュール	45
付 録 H ソースコード	46

表 目 次

2.3.1 順序パターン	6
3.2.1 実験条件	10
3.2.2 情動アンケート	13
4.1.1 分散分析 (視聴覚情報)	16
4.1.2 多重比較検定 (視聴覚情報)	16
4.1.3 t 検定 (即興演奏)	17
4.2.1 多重比較検定 (演奏タイミング)	18

目 次

2.3.1 実験で使用した楽曲	5
2.3.2 6 種類のリズムパターン	5
3.2.1 実験環境の外観	8
3.2.2 システム概要	9
3.2.3 CSV ファイルの出力例	10
3.2.4 実験全体の流れ	12
4.1.1 視聴覚条件毎の感情尺度評価値	15
4.1.2 即興条件毎の感情尺度評価値	17
4.2.1 演奏者間の演奏タイミングのずれ	18
4.3.1 30 回以上確認された頻出リズムパターン 10 個と演奏回数	19
4.3.2 頻出リズムパターン上位 5 つと演奏回数 (視覚+聴覚条件)	20
4.3.3 頻出リズムパターン上位 5 つと演奏回数 (聴覚のみ条件)	20
4.3.4 頻出リズムパターン上位 5 つと演奏回数 (視覚のみ条件)	21

第1章 はじめに

1.1 アンサンブルとコミュニケーション

我々はアンサンブル演奏を行う際、共演者と心を通わせ一つの音楽を形作る。その際に、相手と息を合わせるために視線を通わせたり呼吸の音に耳を澄ましたりすることで、綺麗に音が重なり何とも言えない高揚感を体験したことが楽器経験者ならば誰しもあるだろう。またジャズのセッションにおいて、演奏者同士がまるで楽しく会話するかのように即興を仕掛け合う様子もしばしば見かけられる。筆者も、演奏中の共演者の楽しそうな表情に喜びを感じたり、バンドの演奏において共演者が加えたアレンジに高揚し、自らも演奏を変容させたりした経験が幾度となくある。これらのことから、アンサンブル演奏を行う際の視聴覚の情報や、即興という要素は我々の情動や演奏に何らかの影響を及ぼしているのではないかと考えられる [1]。そこで本研究では、音楽でアンサンブルを演奏する時の情動や演奏はどのような要素に影響を受けるかについて「視聴覚情報」と「即興」という2つの要素に着目し、演奏実験を通して調査を行う。

1.2 情動とは

音楽を聞いた時の心の動きや体の反応を「音楽情動」と呼ぶ。Juslin と Sloboda(2010)によると、情動は「情動は内部や外部の環境において重要でありそうな事象に対する、比較的短時間の猛烈で変化が急激な反応である。また多くの場合、情動は社会的であることが多く認知の変化や主観的感情、表現的な振る舞い、行動を起こす傾向といった同時に生じる複数の構成要素を伴う」と定義される [2]。

つまりこの定義の中の“内部や外部の環境において重要でありそうな事象”が音楽であるのが音楽情動なのである。類似した言葉に「気分」や「感情」があるが、「気分」は情動よりも長期間に渡るもので、情動のように誘発した要因を特定することが難しく、表情による表現を伴わないと考えられている [3]。つまり「感情」は情動と気分を総称したものである。

1.3 視聴覚情報に関する研究

音楽を演奏する際、聴覚情報が最も重要な役割を果たしていると思われがちであるが共演者の姿のような視覚情報を演奏者がリソースの一つとして取り入れていることが報告されている。Davidson らの研究 [4] では弦楽四重奏の演奏者間での視覚的コンタクトが演奏の重要なタイミングで用いられている事が述べられており、Seddon らの研究 [5] や河瀬らの研究 [6] ではどういった視覚的情報が演奏の参考とされているのかを調査されている。また、寺澤らの研究では、「音楽による情動への訴えかけは聴覚刺激だけではなく、顔の表情やジェ

スチャーといった他の感覚 2 刺激 (視覚刺激など) を統合した形で伝搬される。そして別の演奏者は音によって伝達された 情動情報と身体動作によって伝達された情動の両方を融合した形で知覚する」と述べられていた。また、「伝搬された情動は別の演奏者の中枢神経系や末梢神経系などを含む「内部システム」に作用し、身体動作として表出される」とも述べられていた [1]。以上の研究から、視聴覚情報が演奏者に大きな影響を与える可能性があることが理解できる。

そこで、視聴覚情報が情動や演奏面に具体的にどういった影響を与えているのかを調査した。まず、音楽演奏時に視聴覚情報が与える影響について調査したところ、ドラムやベース、ピアノなど様々な楽器で対面状況と非対面状況のアンサンブルでの演奏のタイミングのずれ時間の比較がなされていた [7] [8]。また、河瀬らの研究では、身体を全て隠したり、一部分だけ見せたりなど条件を変化させてアンサンブルを演奏させる実験を行っており、身体の一部が見えるだけでも演奏のタイミングのずれ時間が短くなることが述べられていた [9]。さらに、ピアノやドラムのアンサンブルにおいて、非対面条件よりも対面条件の時に二者間の演奏のタイミングのずれ時間が減少することが報告されており [10] [11]、これらの研究から、視聴覚の情報がアンサンブルにおける演奏のタイミングのずれ時間の減少に大きく影響を与えていることがわかる。

次に、視聴覚情報が心理面にどういった影響を与えているのかを調査した。Silveira の研究では、トロンボーンカルテットの演奏者に 3 種類の演奏動作 (基本動作のみ、頭と顔を動かす、体全体を動かす) を指示し、各動作に対して聞き手が感情の豊かさを評価する実験を行ったところ、聞き手の感情の豊かさには演奏動作の大きさが影響していることが述べられていた [12]。また、ピアノの演奏やダンスを音と映像、音のみ、映像のみで聴衆に感情的な反応を尋ねる実験なども行なわれていた [13] [14]。しかし、これらの研究は聞き手と演奏者の関係を調査したものであり、演奏家同士では同一の調査が殆ど行なわれていなかった。そこで、本研究では、先行研究と同様に演奏のタイミングのずれの分析と、更にアンサンブル演奏時の視聴覚情報が人間の情動に影響を与えるかを演奏実験を通して調査する。

1.4 即興演奏に関する研究

音楽演奏でアンサンブルを行う時に情動に影響を与えうるもう一つの要素として、即興が考えられる。音楽演奏における即興とは、楽譜に頼らず即座に創作をしながら演奏を行うことを指しており、ジャズやフュージョンなど様々な音楽ジャンルで散見される。この即興演奏と情動の関係について調査を行ったところ、うつ病の患者に対して、通常の治療に加えて即興演奏での治療を取り入れたところ、取り入れていない群に比べて鬱の程度を調べるアンケートの評価が有意に高くなったという報告がなされていた [15] [16]。また、薬物乱用患者に対してドラムの即興演奏を用いた研究も存在しており、その研究において即興演奏には落ち着きをもたらす効果があることが報告されていた [17]。しかし、これらの研究はうつ病患者や薬物乱用患者など、調査の対象が特殊なケースであった。また、これらの研究は、実験前後での否定的な情動 (不安感、憂鬱など) の変化を調査したものが多く、肯定的な情動の変化も包括した調査がなされていなかった。そこで、本研究では健常者を対象に、即興演奏が肯定的な情動や否定的な情動にどういった影響を与えるかを、演奏実験を通して調査を行う。また、1.3 節の内容と合わせて、アンサンブル演奏時に視聴覚情報を変化させた際に表

1.5 本研究の目的

出される即興演奏のリズムパターンにも着目する。

1.5 本研究の目的

本研究は「アンサンブル演奏を行う際、視覚情報 (共演者の姿) や聴覚情報 (演奏音) が演奏者の情動、アンサンブルの演奏タイミングの不一致度、演奏内の即興で表出されるリズムパターンに影響を与えるか」「アンサンブル演奏を行う際、演奏内の即興演奏が情動に影響を与えるか」という2つのリサーチクエスチョンを、演奏実験によって明らかにすることを目的とする。

1.6 本論文の構成

本論文は全5章からなっており、第1章では研究背景、関連研究、本研究の目的を述べた。第2章では研究の枠組みを述べ、第3章では実験の概要、第4章では実験の結果とその考察、第5章では本研究のまとめと今後の展望を述べる。

第2章 研究の枠組み

2.1 リサーチクエスションと仮説

本研究では、「アンサンブル演奏を行う際、視覚情報(共演者の姿)や聴覚情報(演奏音)が演奏者の情動、アンサンブルの演奏タイミングの不一致度、演奏内の即興で表出されるリズムパターンに影響を与えるか」「アンサンブル演奏を行う際、演奏内の即興演奏が情動に影響を与えるか」という2つのリサーチクエスションを立てた。そして、これらの疑問に対して、以下の4つの仮説を立てた。

視聴覚情報の有無に関する仮説

アンサンブル演奏時に視覚情報(共演者の姿)や聴覚情報(演奏音)のどちらかが欠けている時に比べ、両方の情報が揃っているときの方が、

情動

肯定的な情動を体験する。

演奏タイミング

演奏者間の演奏タイミングの不一致度が低くなる。

リズムパターン

即興演奏時に表出されるリズムパターンの数が増える。

即興演奏の有無に関する仮説

情動

アンサンブル演奏内で即興演奏を行わない場合よりも、即興演奏を行う方が肯定的な情動を体験する。

これらの仮説の検証のため、本研究では演奏実験を実施し、その時の演奏者の情動と演奏を測定する。

2.2 対象とする演奏

本研究では、演奏実験の対象としてドラム演奏を採用した。理由は、他の楽器よりも演奏が簡単であり、演奏する曲の印象が調整に左右されないため、演奏そのものが引き起こす情動の変化の測定が行いやすいからである。

2.3 実験に使用する楽曲

実験に使用した楽曲は、図 2.3.1 に示した自作の楽曲である。本楽曲は、実験参加者が演奏に積極的に参加出来るよう、ユニゾンや即興での掛け合いなどアンサンブルにおける音楽的要素を含むような工夫を施した。なお、BPM(Beats Per Minute) は 120 である。

$\text{♩} = 120$

※ 実験ではこの16小節を3回繰り返し、4回目に1～4小節目を弾いて演奏終了となる

図 2.3.1: 実験で使用する楽曲

今回の実験は、実験実施者と実験参加者の2名で行った。今回の実験では、実験実施者がパート A、実験参加者がパート B を担当した。1～4、9～12 小節目は2人がユニゾンで演奏する。また5～8、13～16 小節目では一人が2拍即興演奏を行い、次の2拍でその即興演奏をもう一人が真似をする。この16小節を3回繰り返し演奏し、4回目に最初の1～4小節目を演奏することを1試行とする。

また、実験実施者の即興演奏の難易度が実験参加者間で同じになるよう、実験実施者は実験参加者には伝えず、6種類のリズムパターンを即興パートで演奏した。それぞれのリズムを図 2.3.2 に示す。なお、これらのリズムパターンは本実験を行う前に行った予備実験において類似した実験を行った際に、実験参加者の即興で観察された頻出のリズムパターン6個を使用している。

図 2.3.2: 6種類のリズムパターン

2.3 実験に使用する楽曲

これらのリズムパターンの演奏回数に偏りが出ないように、予め4つのリズムの順序パターンを作成した。それぞれの順序パターンを表 2.3.1 に示す。なお、表中の数字は、図 2.3.2 の丸数字にそれぞれ対応している。これらの順序パターンは8試行の内それぞれランダムに2回ずつ演奏された。リズムパターンと同様に、順序パターンがあることも被験者には伝えなかった。

表 2.3.1: 順序パターン

パターン	繰り返し 1 回目の即興	繰り返し 2 回目の即興	繰り返し 3 回目の即興
1	① → ②	③ → ④	⑤ → ⑥
2	③ → ④	⑤ → ⑥	① → ②
3	⑤ → ⑥	① → ②	③ → ④
4	⑥ → ⑤	④ → ③	② → ①

第3章 実験

3.1 実験概要

本実験は「アンサンブル演奏を行う際、視覚情報(共演者の姿)や聴覚情報(演奏音)が演奏者の情動、アンサンブルの演奏タイミングの不一致度、演奏内の即興で表出されるリズムパターンに影響を与えるか」「アンサンブル演奏を行う際、演奏内の即興演奏が情動に影響を与えるか」という2つのリサーチクエスチョンを検証する為に実施された。前者の検証では、視覚+聴覚条件、視覚のみ条件、聴覚のみ条件の3条件下でそれぞれドラムアンサンブルを行った。また後者の検証では、演奏内で即興を行う条件(視覚+聴覚条件と同一)と行わない条件の2条件下でドラムアンサンブルを行った。実験条件は全部で8試行(4条件を各2回演奏)であり、1試行が終了する毎に演奏中の情動に関するアンケートを実施した。また、実験の最後にそれぞれの条件で演奏した感想等をインタビューにて調査した。実験の様子はビデオカメラで撮影された。

3.2 実験デザイン

3.2.1 実験参加者

実験参加者は筑波大学の学生を中心に、楽器の演奏経験が1年以上ある人に参加を呼びかけた。本実験に参加したのは27名(男性14名、女性13名)であり、そのうちの6名(男性4名、女性2名)は打楽器演奏の経験があった(詳細は付録Aを参照)。

3.2.2 実験環境

本実験にはパーソナルコンピュータ2台(Mac Book Pro, Surface Pro 3)と、ドラムパッド(ALESIS社製, PERFORMANCE PAD PRO)と、ヘッドホン2台(SONY社製, MDR-CD900ST), ミキサー(MACKIE社製, 802VLZ4)を用いた。実験参加者にはドラムパッドの叩く位置を指定せず、起立した状態で叩いてもらった。ドラムパッドの高さは自由に変更することが可能であり、無理のない姿勢で実験を行うことが可能である。また、ドラムパッドの全8パッドのうち右4パネルにはハイウッドブロック(ノートナンバ: 76), 左4パネルにはローウッドブロック(ノートナンバ: 77)の音を割り当てた。実験は筑波大学春日キャンパス 7B203にて行われた。実験環境の外観を図3.2.1に示す。

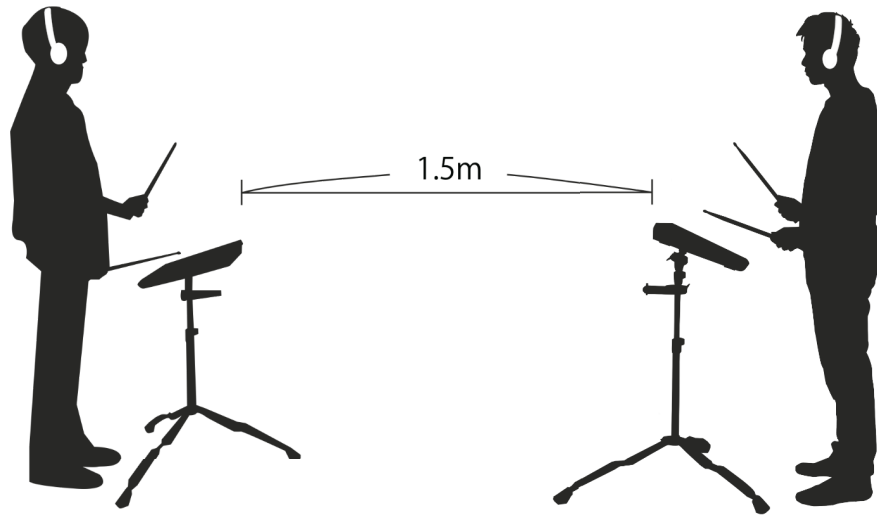


図 3.2.1: 実験環境の外観

3.2.3 実験用システム

本研究は、ドラムアンサンブルを行う際に視聴覚の情報や即興の要素が変化するすることで、演奏者の演奏や情動に変化が表れるかどうかを検証することが目的である。その中でも、1.3 節と 1.4 節で述べた演奏タイミングと即興演奏で表出されるリズムパターンを検証するにあたり、様々な条件下で演奏している時の演奏者の演奏データ (演奏時間) を記録する必要があると考えた。そこで、本研究では演奏の様子分析に必要な演奏データ (演奏時間) を取得するシステムを作成した。なお演奏データの分析のしやすさを考慮し、本実験では音楽の演奏情報を伝達する規格である MIDI を使用した。システムの概要を図 3.2.2 に示す。図中の「MIDI メッセージ」はデータを送受信する際の経路区分を表す MIDI チャンネル番号、入力デバイスであるドラムパッドを叩いた際の位置を表すノートナンバ、ドラムパッドを叩いたときの強さを表すベロシティの 3 つの値を指す。また、ステータス情報は音を鳴らす (止める) ことを表すシグナルを指す。

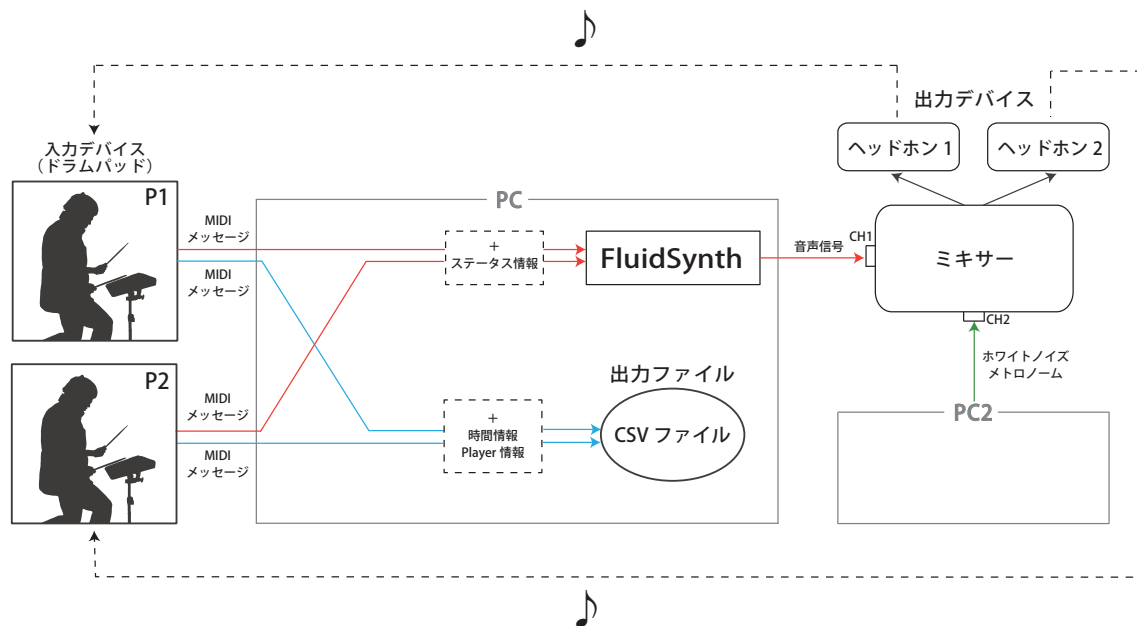


図 3.2.2: システム概要

音生成部 (処理 A)

機能

入力デバイス (ドラムパッド) から受け取った MIDI メッセージを、フリーでオープンソースのシンセサイザである FluidSynth [18] に送信し音を鳴らす。なお MIDI 音源はフリーのサウンドフォントの Titanic を使用した。

データ

ドラムパッドからチャンネル番号、ノートナンバ、ベロシティを受け取り、音を鳴らすことを表すステータス情報を付与して FluidSynth に送信する。

分析用ファイル生成部 (処理 B)

機能

入力デバイス (ドラムパッド) から受け取った MIDI メッセージを、CSV ファイル形式 (幾つかの項目をカンマで区切ったテキストファイル) で出力する。

データ

ドラムパッドからチャンネル番号、ノートナンバ、ベロシティを受け取り、誰が音を鳴らしたかを表す player 情報 (“player0” または “player1”) と記録を開始してからの時間情報を付与して CSV ファイルで出力する。

CSV ファイルの出力例を図 3.2.3 に示す。それぞれの値はデータ記録開始時からの時間、誰が音を鳴らしたかの情報、チャンネル番号、ノートナンバ、ベロシティを表す。

3.2 実験デザイン

5.483787	player	0	76	87
5.484535	player	0	76	0
5.565095	player	0	77	73
5.565984	player	0	77	0
5.573765	player	1	77	85
5.574556	player	1	77	0
5.811727	player	1	76	60
5.812514	player	1	76	0
5.841528	player	0	76	83
5.842255	player	0	76	0

図 3.2.3: CSV ファイルの出力例

3.2.4 実験条件

本実験では4つの条件を用意し、各条件の演奏を2回ずつ行った(全体で8試行)。演奏順はランダム化された。実験条件の詳細を表 3.2.1 に示す。

表 3.2.1: 実験条件

	即興+真似	真似のみ
視覚+聴覚	対面，音あり	対面，音あり
聴覚のみ	背中合わせ，音あり	
視覚のみ	対面，音あり (耳栓+ホワイトノイズ)	

視聴覚条件

視聴覚の条件の違いが情動や演奏に与える影響を調べる際は、2.3 節で紹介した楽曲を3つの条件で演奏する。

1. 視覚+聴覚条件

相手と向き合って演奏を行う。互いの演奏音を聞くことが出来る。

2. 聴覚のみ条件

相手と背中合わせで演奏を行う。互いの演奏音を聞くことが出来る。

3. 視覚のみ条件

相手と向き合って演奏を行う。ミキサーでドラムパッドからの音をミュートした上で耳栓を着け、ホワイトノイズを流すことでドラムパッドを叩いた時の物理的な音も完全に遮断する。そのため、自分の演奏音も相手の演奏音も聞くことは出来ない。

3.2 実験デザイン

即興条件

即興演奏が情動に与える影響を調べる際は、2.3節で紹介した楽曲を2つの条件で演奏する。

1. 即興+真似条件 (上記の視覚+聴覚条件と同一)
2.3節で紹介した通り、実験参加者と実験実施者がターン交代制で即興と真似を繰り返す。
2. 真似のみ条件
この条件では、2.3節で紹介した楽曲の1~4小節目のみを3回繰り返し、最後に1, 2小節目を演奏して終了となる。つまり、実験実施者のみが即興演奏を行うので、実験参加者は実験実施者の真似のみを行う。

3.2.5 実験手順

実験は次のような順序で行った。

1. 被験者から実験の同意を貰い、実験の概要を説明する。
2. 被験者の性格と楽器演奏経験を尋ねるアンケートを行う。
3. 実験で使用する楽曲を弾けるようになってもらうための練習を行う。
4. 演奏を8試行(4条件×2回)行う。演奏1試行が終わる度に、被験者の演奏中の情動を尋ねるアンケートを行う。
5. 演奏課題や実験条件に関するインタビューを行う。

実験全体の流れを図3.2.4に示す。

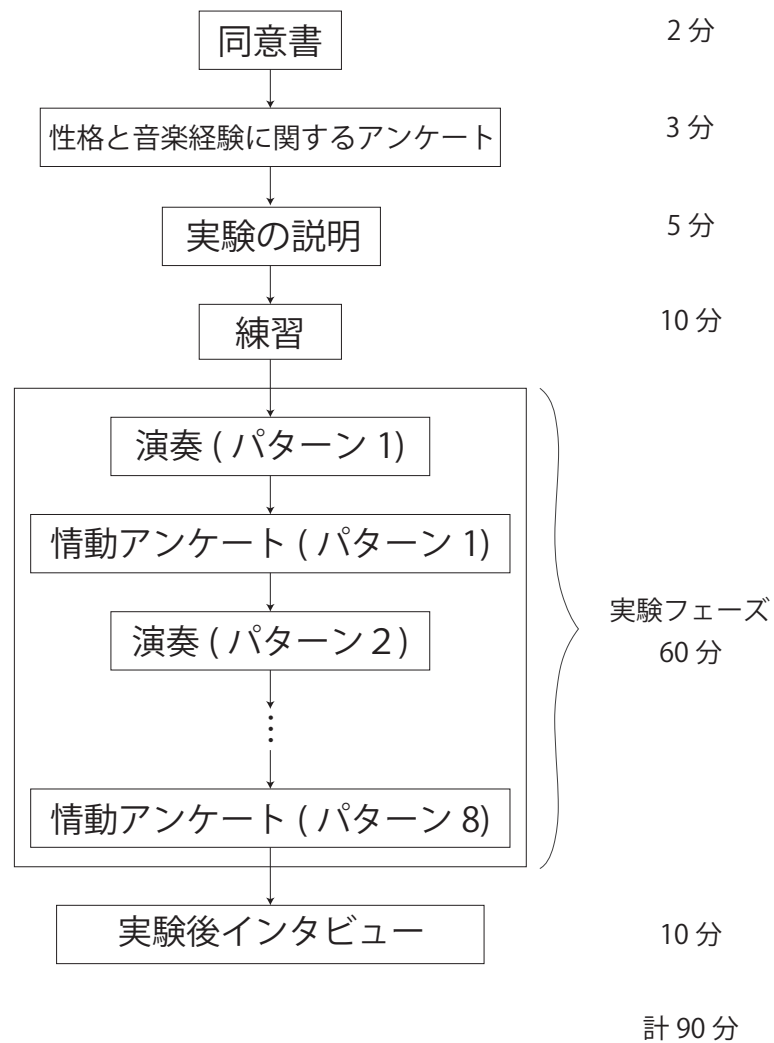


図 3.2.4: 実験全体の流れ

実験前

初めに実験参加者に実験の概要を説明し、同意書(付録B)に署名を貰った。その後、実験参加者の楽器演奏経験と性格を問うアンケート(付録C) [19]に回答してもらった。アンケート終了後は、ドラムパッドの叩き方や譜面の確認を行った。その後、実験参加者には実験で使用する楽曲を弾けるようになってもらうために、練習を行った。練習時間は最長10分とした。

実験中

演奏1パターンが終わるたびに、実験参加者に演奏中の情動に関するアンケート(付録D) [20]に記入してもらった。このアンケートで用いられた感情尺度は本来、一過性の運動における情動を測定するものだが、1) 否定的な情動だけでなく肯定的な情動も含めて測定

が可能, 2) ドラムの演奏は身体を大きく活用するものであり, 運動とよべる行為を行う, という理由から本実験にて使用した.

実験後

実験後, 各実験条件に対する印象の調査のため, 実験後インタビューを実施した. インタビューでは主に以下の項目について質問した.

1. 視覚のみ条件において流したホワイトノイズに対してどういった印象を抱いたか?
2. 共演者が実施した即興演奏はどうだったか?
3. 即興演奏をする条件としない条件で印象は変化したか?
4. 視聴覚の情報を変化させた時で印象は変化したか?
5. 実験の難しさを 0~100 の間で表すと, どの位だったか?

3.2.6 分析方法

情動の分析では, 3.2.5 節の実験中の項目にて述べたアンケートを使用する. このアンケートでは, 直前の演奏に対し感情語 12 項目に対して 5 段階評価を行う. 5 段階評価は数値が大きいほどその感情語の印象が大きくなることを表す. 感情語 12 項目とその語に共通する因子 3 つを示したものを表 3.2.2 に示す.

表 3.2.2: 情動アンケート

否定的感情	高揚感	落ち着き感
(N1) - 沈んだ	(E1) - 燃え上がった	(R1) - のんびりした
(N2) - うろたえた	(E2) - 夢中な	(R2) - 落ち着いた
(N3) - 心苦しい	(E3) - 胸おどる	(R3) - 安らいだ
(N4) - いやがった	(E4) - わくわくした	(R4) - 安心した

分析では, 各因子ごとに評価の合計値を算出する (4~20 点). 以下に計算式を掲載する.

$$P_{Negative} = N1 + N2 + N3 + N4$$

$$P_{Elation} = E1 + E2 + E3 + E4$$

$$P_{Relax} = R1 + R2 + R3 + R4$$

演奏タイミングの分析では、アンサンブルの2者間の演奏タイミングの不一致度を調査する。本研究で使用する楽曲は、ユニゾンパートと即興パートが交互に訪れる楽曲であるが、被験者全体で計測箇所の統一を計るために、今回はユニゾンパートのみを分析の対象とした。不一致度 $D_{disagreement}$ は以下のように定義される。

初めに計時の開始時刻を、実験実施者または実験参加者の打拍の中で最も時間が早いものとした。そして、実験参加者がある音を叩いた時の時刻を x_i 、実験実施者が同じある音を叩いた時の時刻を X_i 、1曲あたりの音数を n としたとき不一致度 $D_{disagreement}$ は次のように表される。

$$D_{disagreement} = \frac{1}{n} \sum_{i=1}^n \sqrt{(X_i - x_i)^2}$$

演奏内の即興で表出されるリズムパターンの分析では、全員分の各条件の即興演奏を採譜し、各条件で何種類のリズムパターンがあったかを調べることで多様性を調査する。

第4章 実験結果

4.1 情動

4.1.1 視聴覚条件毎の感情尺度評価値

視聴覚情報を変化させた際の、情動に関するアンケートの感情尺度評価値を比較した結果を図 4.1.1 に示す。グラフの縦軸方向はアンケートの感情尺度評価値、グラフの横軸方向は条件を表しており、エラーバーは信頼区間 95%を表す。グラフ中の記号は統計的な信頼性を表すものであり、本論文では P 値が 0.10 以下の場合は“†”，0.05 以下の場合は“*”，0.01 以下の場合は“**”，0.001 以下の場合は“***”の記号にて表す。なお、この記号は本論文での全ての統計分析で共通である。

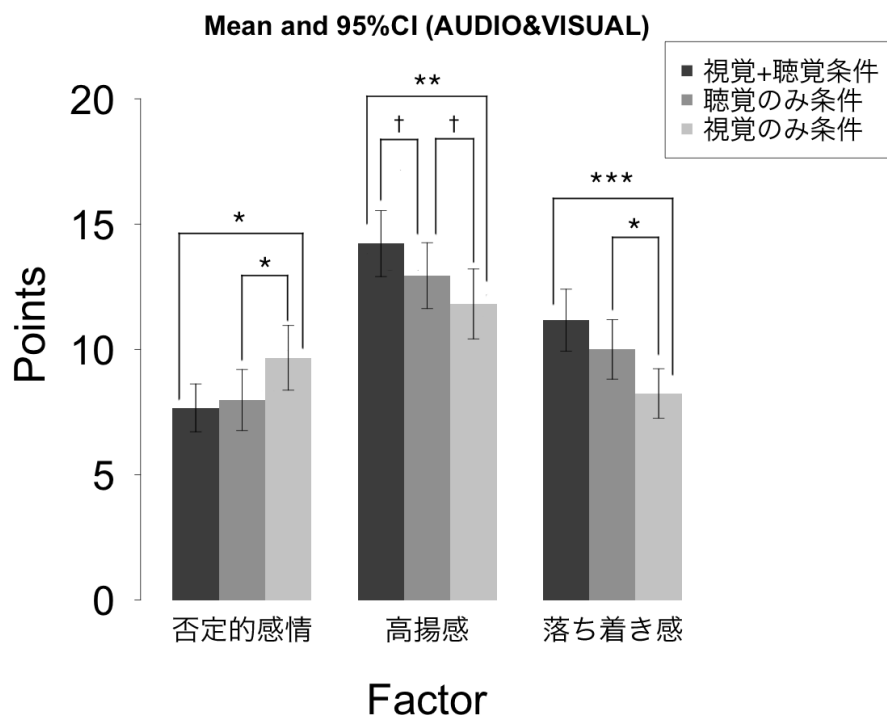


図 4.1.1: 視聴覚条件毎の感情尺度評価値

グラフを見ると、“否定的感情”においては視覚のみ条件，聴覚のみ条件，視覚+聴覚条件の順に評価値が低くなり，“高揚感”と“落ち着き感”においては視覚のみ条件，聴覚のみ条件，視覚+聴覚条件の順に評価値が高くなっていった。このことから 2.1 節で述べた「視覚情報や聴覚情報のどちらかが欠けている時に比べ、両方の情報が揃っているときの方が、肯定

4.1 情動

的な情動を体験する」という仮説通りの結果が得られていることが確認できた。

そこで、実際に条件の違いが感情評価尺度値に影響しているかどうかを確かめる為一元配置分散分析を実施した。分散分析の結果を表 4.1.1 に示す。表の上段は、左から因子、平方和、平均平方、統計量、P 値を表している。

表 4.1.1: 分散分析 (視聴覚情報)

因子	Df	Sum Sq	Mean Sq	F value	Pr(>F)	
否定的感情	2	62.5	31.225	3.599	0.032	*
高揚感	2	78.3	39.17	3.391	0.0387	*
落ち着き感	2	117.2	58.58	6.99	0.00161	**

検定の結果，“否定的感情”，“高揚感”，“落ち着き感”の全てにおいて視聴覚の条件が評価値に影響を与えることが示された。次に、どの条件間で差があったか調べるために、多重比較検定を実施した。なお、多重比較検定の際の P 値の補正にはホルム法を用いた。検定の結果、有意差または有意傾向が見られた条件の組み合わせを表 4.1.2 に示す。表の上段は左から因子、条件の組み合わせ、P 値を表している。

表 4.1.2: 多重比較検定 (視聴覚情報)

因子	条件の組み合わせ	Pr(>F)	
否定的感情	視覚+聴覚条件 - 視覚のみ条件	0.015	*
	視覚のみ条件 - 聴覚のみ条件	0.021	*
高揚感	視覚+聴覚条件 - 視覚のみ条件	0.0047	**
	視覚+聴覚条件 - 聴覚のみ条件	0.064	†
	視覚のみ条件 - 聴覚のみ条件	0.064	†
落ち着き感	視覚+聴覚条件 - 視覚のみ条件	0.00079	***
	視覚のみ条件 - 聴覚のみ条件	0.03315	*

4.1.2 即興条件毎の感情尺度評価値

即興条件を変化させた際の、情動に関するアンケートの感情尺度評価値を比較した結果を図 4.1.2 に示す。グラフの縦軸方向はアンケートの感情尺度評価値、グラフの横軸方向は条件を表しており、エラーバーは信頼区間 95%を表す。

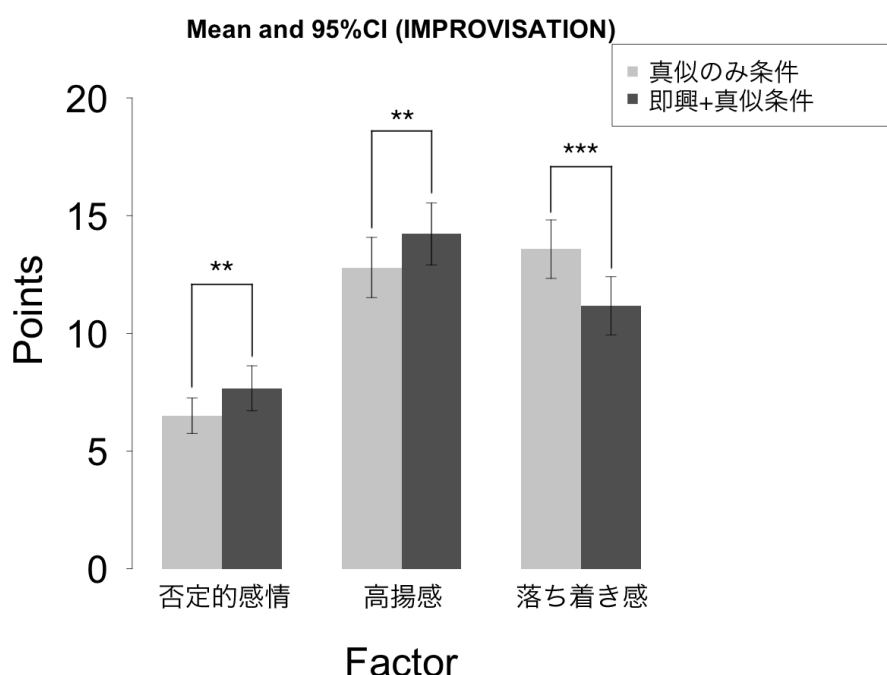


図 4.1.2: 即興条件毎の感情尺度評価値

グラフを見ると，“否定的感情”と“高揚感”においては即興+真似のみ条件の方が評価値が高くなり，逆に“落ち着き感”においては真似のみ条件の方が評価値が高くなっていた．このことから 2.1 節で述べた「アンサンブル演奏内で即興演奏を行わない場合よりも，即興演奏を行う場合の方が肯定的な情動を体験する」という仮説に対して，“高揚感”に関しては一致した傾向が得られたが，“否定的感情”と“落ち着き感”に関しては真逆の傾向が得られたことが確認できた．

そこで，実際にそれぞれの因子で t 検定を行ったところ全ての因子において条件間で有意差が見られた．結果を表 4.1.3 に示す．

表 4.1.3: t 検定 (即興演奏)

因子	条件の組み合わせ	Pr(>F)	
否定的感情	即興+真似条件 - 真似のみ条件	0.006795	**
高揚感	即興+真似条件 - 真似のみ条件	0.005327	**
落ち着き感	即興+真似条件 - 視覚のみ条件	0.0003442	***

4.2 演奏タイミング

視聴覚条件を変化させた際の，演奏者間の演奏の不一致度を比較した結果を図 4.2.1 に示す．グラフの縦軸方向は演奏時の条件，グラフの横軸方向はアンサンブルの演奏者間の演奏タイミングのずれ (ms)，エラーバーは信頼区間 95%を表す．

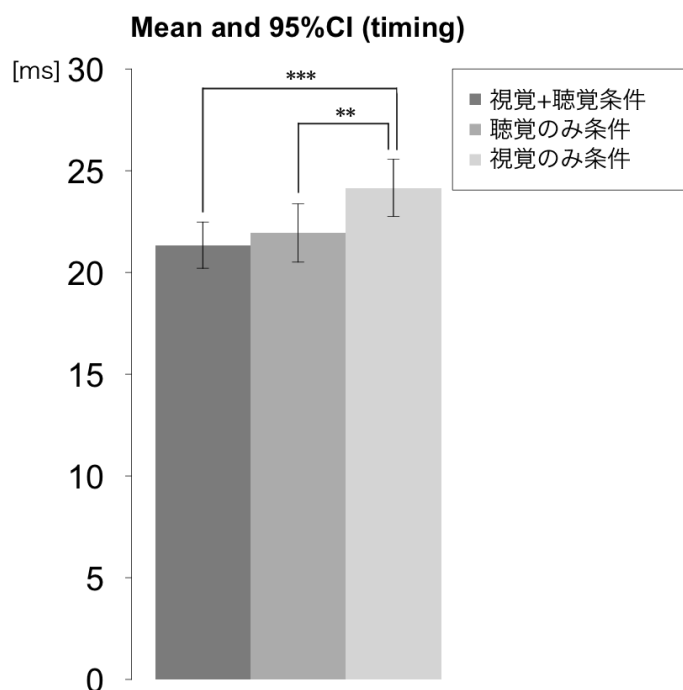


図 4.2.1: 演奏者間の演奏タイミングのずれ

条件によって演奏の不一致度に差が生じるかどうかを検証するために、一元配置分散分析を実施したところ、 $p\text{-value} = 0.00684$ となった。このことから、条件によって演奏の不一致度に差が生じることが示された。次にどの条件間で差があったかを確認するため、多重比較検定を実施した。なお、多重比較検定の際の P 値の補正にはホルム法を用いた。検定の結果、視覚+聴覚条件と視覚のみ条件の間、聴覚のみ条件と視覚のみ条件の間で有意差が確認された。結果を表 4.2.1 に示す。

表 4.2.1: 多重比較検定 (演奏タイミング)

条件の組み合わせ	Pr(>F)	
視覚+聴覚条件 - 聴覚のみ条件	0.2904	
視覚+聴覚条件 - 視覚のみ条件	3.3e-05	***
視覚のみ条件 - 聴覚のみ条件	0.0011	**

4.3 リズムパターン

条件毎に表れたリズムパターンの数を集計した結果、視覚+聴覚条件が 56 個、聴覚のみ条件が 58 個、視覚のみ条件が 56 個であり、条件間でリズムパターンの数に差がないことが判明した。全体では 80 通りのリズムパターンが確認された。全 949 回の即興演奏の内、30 回以上演奏されたリズムパターン 10 個を抽出したものを、以下の図 4.3.1 に示す。

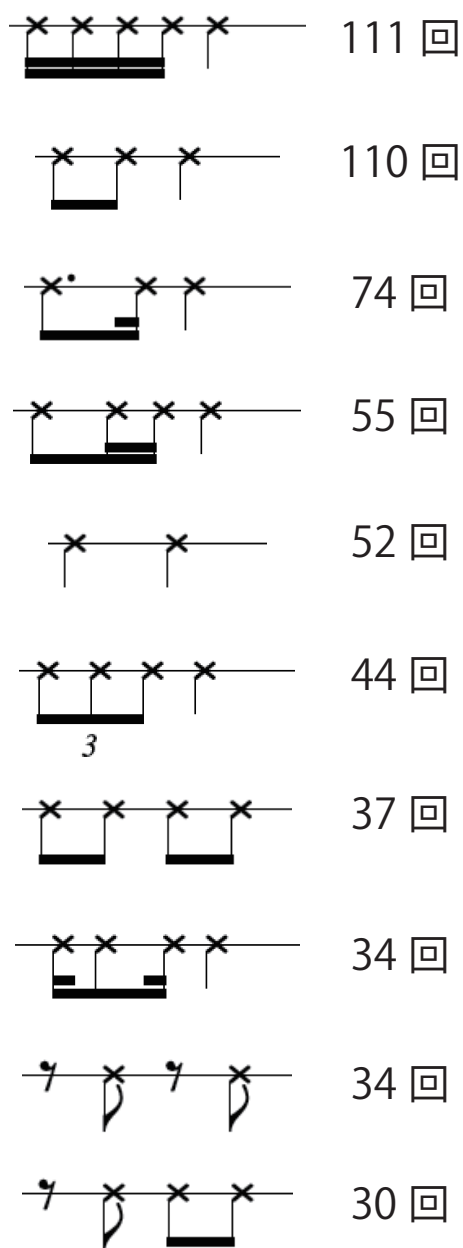


図 4.3.1: 30 回以上確認された頻出リズムパターン 10 個と演奏回数

次に、各条件毎に頻出だったリズムパターンの上位 5 つを抽出した。視覚+聴覚条件が図 4.3.2、聴覚のみ条件が図 4.3.3、視覚のみ条件が図 4.3.4 である。

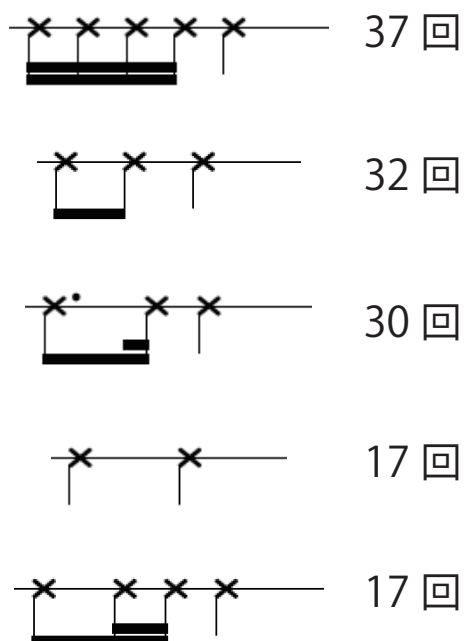


図 4.3.2: 頻出リズムパターン上位 5 つと演奏回数 (視覚+聴覚条件)

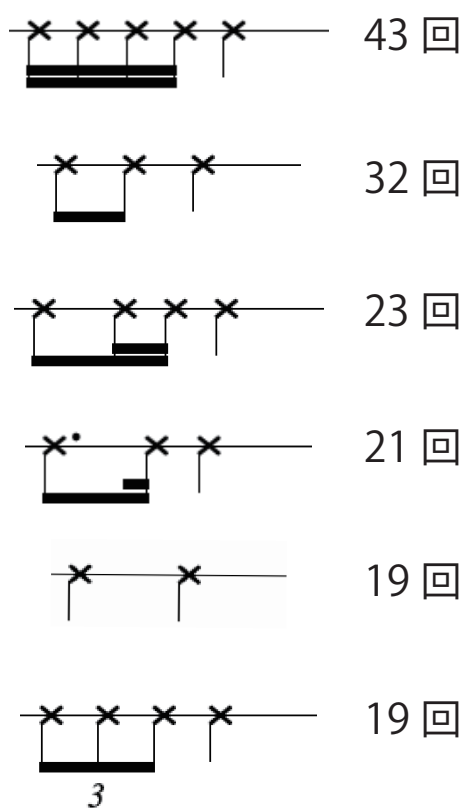


図 4.3.3: 頻出リズムパターン上位 5 つと演奏回数 (聴覚のみ条件)

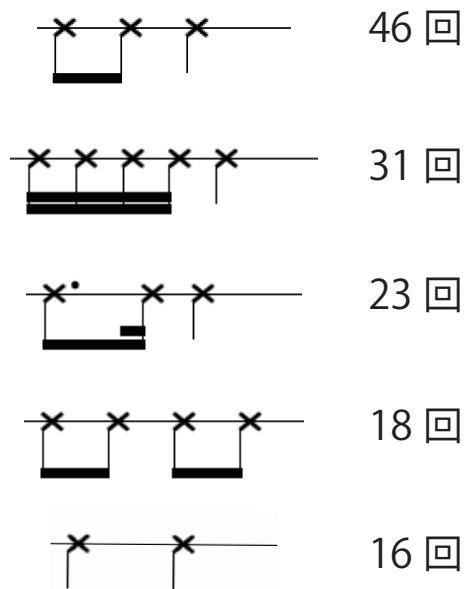


図 4.3.4: 頻出リズムパターン上位5つと演奏回数 (視覚のみ条件)

4.4 結果のまとめ

- アンサンブル演奏時に視覚情報 (共演者の姿) や聴覚情報 (演奏音) のどちらかが欠けている時に比べ、両方の情報が揃っているとき、
 - － 情動アンケートの“否定的感情”の因子の評価値が低くなり、“高揚感”と“落ち着き感”の因子の評価値が高くなった。
 - － 演奏者間の演奏の不一致度が微量ではあるが低くなる傾向が見られた
 - － リズムパターンの種類の豊富さには差が見られなかった
- アンサンブル演奏内で即興演奏を行わない場合よりも、即興演奏を行うとき、
 - － 情動アンケートの“高揚感”の因子の評価値が高くなった

4.5 考察

4.5.1 情動

視聴覚情報

2.1 節で述べた本研究の視聴覚情報と情動の関係に関する仮説は以下のとおりであった。

アンサンブル演奏時に視覚情報 (共演者の姿) や聴覚情報 (演奏音) のどちらかが欠けている時に比べ、両方の情報が揃っているときの方が、肯定的な情動を体験する

分析の結果、全ての因子において視覚+聴覚条件と視覚のみ条件の間で有意差が確認された。また、“否定的感情”においては視覚のみ条件、聴覚のみ条件、視覚+聴覚条件の順に評価値が低くなり、“高揚感”と“落ち着き感”においては視覚のみ条件、聴覚のみ条件、視覚+聴覚条件の順に評価値が高くなっていた。よってこれらの結果から以下の二つのことが推測される。

- 演奏者の情動は聴覚情報の有無によって大きく左右される
- しかし、演奏者は聴覚情報だけでなく視覚情報も参考にすることでポジティブな情動をより強く体験する

しかし実験後のインタビューにおいて、視覚のみ条件で聴覚情報を遮断するために使用されたホワイトノイズに対し、不快感を感じた実験参加者が27名中7名おり、この点が視覚のみ条件の評価を大きく下げてしまったと考えられる。そのため今後同様の実験を行う場合は、音をノイズのように別の音で遮断するのではなく、別室環境や映像などを使用してなるべく実験参加者がストレスフリーに演奏を行える環境づくりをする必要がある。

また、“高揚感”と“落ち着き感”のような相反する関係の情動評価がどちらも高くなる現象が見られた。この理由を探るために実験後のインタビューを調査した所「相手の姿が見えている時は、相手の反応が気になった」「相手の演奏の音の高低を真似しようと頑張った」というコメントがあった。これらの意見から、以下の二つの現象が同時に生じた為に今回のような結果になったと考えられる。

- 視覚と聴覚の両方の情報が合わさることで相手の演奏のニュアンスや表情がより詳細に伝わりそれらに対応する楽しさが生まれた
- 演奏の中で相手とコミュニケーションを取れているという実感が安心感を生み出した

一方、視覚+聴覚条件と聴覚のみ条件では“高揚感”の有意傾向を除いて統計的な差は確認出来なかった。このことから、共演者の視覚情報は“否定的感情”や“落ち着き感”にあまり影響を与えないということが考えられる。しかし、楽器演奏の経験年数のみを考慮して実験参加者を募集したため、そもそも連弾やセッションなど相手を見ながら演奏する習慣が全くない演奏者がいたことが実験後のインタビューから判明した。また、視覚の遮断のような制限のある条件が楽しかった、と条件のチャレンジングな側面に面白さを感じていた実験参加者もいた。そのため、今後同一の実験を行う場合はこれらの要素を考慮する必要がある。

次に、実験参加者一人ひとりの評価の傾向を観察した。すると27名中2名が本研究で立てた仮説と真逆の傾向を示していた(“否定的感情”の因子では、視覚+聴覚のみ条件の評価値が3条件の中で最も高く、“高揚感”と“落ち着き感”の因子では視覚+聴覚のみ条件の評価値が3条件の中で最も低くなっていた)。そこで、これら2名の実験参加者の実験参加者を外して再度多重比較検定を実施したところ、表4.1.2で有意傾向が確認されていた“高揚感”の視覚+聴覚条件と聴覚のみ条件、視覚のみ条件と聴覚のみ条件が有意差(どちらも $p\text{-value} = 0.0479$)になり、“落ち着き感”では視覚+聴覚条件と聴覚のみ条件の間に有意傾向を確認

4.5 考察

することができた ($p\text{-value} = 0.066$).

そして今回、演奏実験前に実験参加者の性格を調査したが、その結果と感情評価尺度値との関係性を調べるにまで至らなかったため、今後分析する必要があると考える。

即興演奏

2.1 節で述べた本研究の即興演奏と情動に関する仮説は以下の通りであった。

アンサンブル演奏内で即興演奏を行わない場合よりも、即興演奏を行う方が肯定的な情動を体験する

分析の結果、“高揚感”の項目では、即興演奏を行わない場合よりも即興演奏を行うほうが評価値が高くなっていた。しかしその一方で、“否定的感情”の項目では即興演奏を行わない場合よりも即興演奏を行うほうが評価値が高く、“落ち着き感”の項目では即興演奏を行わない場合よりも即興演奏を行うほうが評価値が低くなるという、仮説と反対の傾向が得られた。

また実験後のインタビューを見てみると、7名の実験参加者が「即興演奏のように考えて演奏を行うことが楽しい」という回答していたが、その一方で「即興演奏をしないほうが気楽だった」「即興演奏が難しかった」と回答した実験参加者も確認された。これらの結果をまとめると、以下の二つのことが推測される。

- 即興演奏のクリエイティブな側面は、演奏者の情動を高ぶらせる
- 即興演奏は人によっては負担が大きい作業であり、緊張や苦痛などの否定的な情動を引き起こす可能性がある

1.4 節で述べたように、今回の実験は健常者を対象に即興演奏が肯定的な情動や否定的な情動にどういった影響を与えるかを調査したものであるが、この結果からは即興演奏は健常者の肯定的かつ活性の高い情動に強く働きかける作用をもつことが読み取れる。しかし、そもそも高揚感が上昇した原因が即興演奏という難しいタスクをこなしたという達成感と緊張感が“高揚感”の上昇を引き起こしたとも考えられる。また、否定的感情と落ち着き感において高揚感と逆の結果になった原因としては、今回の実験における即興演奏が義務的かつ短期のスパンで何度も行う必要のある負担の大きい作業であり、即興演奏に馴染みが薄い実験参加者の否定的な情動の評価を高くしたためだと考えられる。そのため、今後この実験と同様に即興演奏がアンサンブルの演奏者にもたらす効果を調査する際には、実験参加者の演奏経験の設定や、演奏の中にどの程度即興演奏を盛り込むかなど、細かく考慮する必要があると考えられる。

4.5.2 演奏タイミング

2.1 節で述べた本研究の演奏タイミングに関する仮説は以下の通りであった。

アンサンブル演奏時に視覚情報 (共演者の姿) や聴覚情報 (演奏音) のどちらかが欠けている時に比べ、両方の情報が揃っているときの方が演奏者間の演奏タイミングの不一致度が低くなる

実際に確認された演奏タイミングの不一致度を見てみると、視覚のみ条件、聴覚のみ条件、視覚+聴覚条件の順に値が小さくなっており、視覚+聴覚のみ条件が最もずれの少ない演奏をしていた。そこで、次にこのずれが演奏にどの程度影響を及ぼすのかどうかを調査した。調査の結果、最も不一致度の高かった視覚のみ条件で、二者間が平均しておよそ 24 ミリ秒ほどずれていることがわかった。過去の先行研究では、演奏者 2 人の演奏が困難になるずれ時間は 50 ミリ秒以上という結果が得られており [21]、殆ど演奏に支障がない程度のずれであったことがわかる。また、最も不一致度の高かった視覚のみ条件と最も不一致度の低かった視覚+聴覚のみ条件の間の時間差はおよそ 3 ミリ秒であり、ほとんど無視できるほどの差でしかなかったことわかる。つまり視聴覚情報の欠落は演奏タイミングの不一致度に影響を及ぼさないということがわかった。しかし、今回の実験でこのような結果になったのは実験の設定や楽曲といった外部の要因による影響も考えられるため、以下のポイントを今後の研究では再考すべきだと考える。

- 視覚のみ条件で演奏のテンポが変動しないように、全条件でメトロノームを鳴らした
- 調査の対象となったユニゾンのパートが、付点音符や三連符などを含む難しいリズムではなく簡単なリズムだった
- リタルダンドやアツェレランドなど速度記号を含む楽曲ではなかった

4.5.3 リズムパターン

2.1 節で述べた本研究のリズムパターンに関する仮説は以下の通りであった。

アンサンブル演奏時に視覚情報 (共演者の姿) や聴覚情報 (演奏音) のどちらかが欠けている時に比べ、両方の情報が揃っているときの方が、即興演奏時に表出されるリズムパターンの数が多くなる

実際に確認されたリズムパターンを条件間で比較してみると、視覚+聴覚条件が 56 個、聴覚のみ条件が 58 個、視覚のみ条件が 56 個であり、視聴覚情報の欠落は即興演奏で表出されるリズムの多様性には影響を及ぼさないことが判明した。また、各条件で頻出リズムパターン上位 5 個のうち、4 つが共通していることから、頻出のリズムパターンの複雑さにも影響を及ぼさないことが判明した。しかし、1, 2 回しか演奏されていない数の少ない即興演奏の内容の複雑さに関しては検証を行えておらず、そちらの方では差が生じている可能性もあるため、今後分析を行いたい。また、難しいリズムを表出した実験参加者の演奏の経験年数や楽器、即興演奏経験などを調査してみるのも興味があるので、同様に分析を行いたい。

次に実際に表出されたリズムパターンを調査してみると、全体で 30 回以上表出された頻出リズムパターン上位 10 個のうち、6 個は実験実施者 (パートナー) が演奏したリズムパターンであった。このことから、即興演奏を行う際に、共演者のリズムパターンを参考にしていることが示唆された。

第5章 まとめと今後の課題

本研究は「アンサンブル演奏を行う際、視覚情報(共演者の姿)や聴覚情報(演奏音)が演奏者の情動、アンサンブルの演奏タイミングの不一致度、演奏内の即興で表出されるリズムパターンに影響を与えるか」と「アンサンブル演奏を行う際、演奏内で即興演奏が情動に影響を与えるか」という2つのリサーチクエスチョンを調査した。実験は27名の実験参加者が実験実施者とペアになってドラムのアンサンブルを行うという演奏実験を行った。考察を踏まえた上での結果のまとめを以下に記す。

- アンサンブル演奏時に視覚情報(共演者の姿)や聴覚情報(演奏音)のどちらかが欠けている時に比べ、両方の情報が揃っているとき、
 - － 情動アンケートの“否定的感情”の因子の評価値が低くなり、“高揚感”と“落ち着き感”の因子の評価値が高くなった
 - － 演奏者間の演奏の不一致度には差が見られなかった
 - － リズムパターンの種類の豊富さには差が見られなかった
- アンサンブル演奏内で即興演奏を行わない場合よりも、即興演奏を行うとき、
 - － 情動アンケートの“高揚感”の因子の評価値が高くなった

情動(視聴覚情報)では、“否定的感情”に関しては視覚のみ条件、聴覚のみ条件、視覚+聴覚条件の順に評価値が低くなり、“高揚感”と“落ち着き感”においては視覚のみ条件、聴覚のみ条件、視覚+聴覚条件の順に評価値が高くなっていった。また、視覚+聴覚条件と視覚のみ条件の間では全ての因子で有意差があり、視覚+聴覚条件と聴覚のみ条件の間では“高揚感”の因子のみで有意傾向があったことから、アンサンブル演奏時の演奏者の情動は聴覚情報に大きく左右されるが、演奏者は聴覚情報だけでなく視覚情報も参考にすることでポジティブな情動をより強く体験するということがわかった。

情動(即興演奏)では、“高揚感”に関しては即興演奏を含む演奏の方が評価が高くなったが、しかしその一方で“否定的感情”と“落ち着き感”の2つに関しては、即興演奏を含まない演奏のほうが評価が高かった。また、実験後インタビューでは即興演奏のクリエイティブな側面に楽しさを感じる実験参加者がいた一方で、即興演奏は負担が大きい作業だったと語る実験参加者もいた。このことから、即興演奏は演奏者(健常者)の情動を高ぶらせるが、その一方で負担が大きく否定的な情動を引き起こす可能性があることが示唆された。

アンサンブルの演奏タイミングの不一致度の調査では、視覚+聴覚条件が視覚のみ条件や聴覚のみ条件に比べて不一致度が低くなっていたが、その差が3ミリ秒程度のかかなり小さな差でしかなかった。このことから演奏タイミングの不一致度に関しては、視聴覚情報は影響

を及ぼさないことが確認できた。

演奏内の即興で表出されるリズムパターンの調査では、視覚+聴覚のみ条件が56通り、聴覚のみ条件が58通り、視覚のみ条件が56通りであった。このことから、表出されるリズムパターンの多様性に関しては、視聴覚情報は影響を及ぼさないことがわかった。

今回の実験では、実験参加者の条件を細かく設定せずに実験を実施したが、打楽器経験者や即興演奏の経験者に絞って実験を行ったり行った場合に今回の結果と同一の傾向が得られるかどうかを検証してみるのも興味があるので是非実施してみたいと考えている。また、演奏経験年数毎での結果の違いが生じるのかや、アンサンブルの相手を映像にした場合で印象が変化するのかなどの疑問も検証したい。

謝辞

本修士論文を書くにあたり、手厚くご指導して下さいった本学図書館情報メディア系の平賀譲先生、寺澤洋子先生、松原正樹先生に深く感謝致します。また実験の組み立て方や分析方法など実験全体に関してご指導して下さいったアムステルダム大学/ラドバウト大学の貞方マキ子先生、実験システムの構築を協力して下さいった東京藝術大学博士課程の濱野峻行さんにも深く御礼申し上げます。加えて、研究に必要な物品購入や各種手続きなど様々な面で多大なサポートをして下さった山田靖子さん、知的コミュニティ基盤研究センターの横井大輔さん、同期として研究に関わらず様々な手助けをしてくれた中山くん、狩野さん、曹さん、金澤さん、ゼミでの発表内容に関する指摘や、実験参加者の募集に協力してくれたLSPCのM1、B4、B3の皆様、実験に参加して下さいった皆様、そして大学から大学院まで通わせてくれた両親にお礼を申し上げます。

参考文献

- [1] 寺澤洋子, 星-柴玲子, 柴山拓郎, 大村英史, 古河聖, 牧野昭二, 岡ノ谷一夫. 身体機能の統合による音楽情動コミュニケーションモデル. *Cognitive Studies*, Vol.20(1), pp.112-129, 2013.
- [2] Patrik N. Juslin and John A. Sloboda. *Music and Emotion*. Chapter 4.1, p.74, 2010.
- [3] Ekman P. and Davidoson R. J. *Affective Science: A Research Agenda*. The nature of emotion: Fundamental questions, New York: Oxford University Press, pp.411-430, 1994.
- [4] Jane W. Davidson and James M.M. Good. Social and Musical Co-Ordination between Members of a String Quartet: An Exploratory Study. *Psychology of Music*, Vol.30(2), pp.186-201, 2002.
- [5] Frederick Seddon and Michele Biasutti. A comparison of modes of communication between members of a string quartet and a jazz sextet. *Psychology of Music*, Vol.37(4), pp.395-415, 2009
- [6] Satoshi Kawase et al. Communication channels performers and listeners use: A survey study. *Proceedings of ICoMCS December 2007*, pp.76-79, 2007.
- [7] Tomohito Yamamoto. Synchronisation and Fluctuation of Rhythm in Musical Cooperative Performance. *Human-Computer Interaction, Part I, HCII2011, LNCS 6761*, pp.517-526, 2011.
- [8] 三宅明日香, 山内千尋, 村松慶一, 小島一晃, 松井辰則. 行動情報の時系列解析による演奏者間のコミュニケーションモデルの構築. *日本感性工学会論文誌*, Vol.12, No.3, pp.415-423, 2013.
- [9] 河瀬諭, 望月愛. ピアノデュオ演奏における視覚的手がかりとタイミング調整. *日本認知心理学会発表論文集 2010*, p.20, 2010.
- [10] 谷口智子, 中村敏江, 河瀬諭, 片平建史, 安田晶子, 小幡哲史. ピアノアンサンブルにおける演奏者間の非言語コミュニケーションー演奏音の音響的特性の研究ー. *日本心理学会第 71 回大会発表論文集*, p.715, 2007.
- [11] Kenji Katahira et al. The role of body movement in co-performers' temporal coordination. *Proceeding of ICoMCS December 2007*, pp.72-75, 2007.

-
- [12] Jason M. Silver. The effect of body movement on listeners' perceptions of musicality in trombone quartet performance. *International Journal of Music Education*, Vol.32(3), pp.311-323, 2014.
- [13] R.J. David Frego. Effects of Aural and Visual Conditions on Response to Perceived Artistic Tension in Music and Dance. *Journal of Research in Music Education*, Vol.47(1), pp.31-43, 1999.
- [14] Eduard Coutinho and Klaus R. Scherer. The effect of context and audio-visual modality on emotion elicited by a musical performance. *Psychology of music*, Online First Published 26 Oct, 2016.
- [15] Jaakko Erikkiä et al. Individual music therapy for depression : randomized controlled trial. *The British Journal of Psychiatry*, Vol.199(2), pp.132-139, 2011.
- [16] Yadira Alborno. The effects of group improvisational music therapy on depression in adolescents and adults with substance abuse: randomized controlled trial. *Nordic Journal of Music Therapy*, Vol. 20(3), pp.208-224, 2011.
- [17] Michael Winkelman. Complementary Therapy for Addiction: "Drumming Out Drugs". *American Journal of Public Health*, Vol. 93(4), 2003.
- [18] FluidSynth, <http://fluidsynth.elementsofsound.org/>, 2016 年 12 月 28 日アクセス
- [19] 和田さゆり. 性格特性用語を用いた Big Five 尺度の作成. *The Japanese Journal of Psychology*, Vol.67(1), pp.61-67, 1996.
- [20] 荒井 弘和, 竹中 晃二, 岡 浩一郎. 一過性運動に用いる感情尺度 — 尺度の開発と運動時における感情の検討 —. *The Japanese Journal of Health Psychology*, Vol.16(1), pp.1-10, 2003.
- [21] 西堀佑, 多田幸生, 曾根卓朗. 遅延のある演奏系での遅延の認知に関する実験とその考察. *情報処理学会研究報告*, 第 127 号, pp.37-42, 2003.
- [22] JACK Audio Connection Kit, <http://jackaudio.org/>, 2017 年 2 月 1 日アクセス
- [23] Express, <http://express.com/>, 2017 年 2 月 1 日アクセス

付 録 A 実験参加者の音楽経験一覧

今回の実験では実験参加者番号が2, 13, 14, 16, 19, 25の6名が打楽器経験者である。
なお、半年以下の年数は切り捨てとする。

実験参加者番号	演奏経験の有る楽器とその年数
1	ギター, 3年
2	ピアノ, 10年 / ドラム, 10年 / ギター, 10年
3	ピアノ, 6年
4	ピアノ, 7年
5	ピアノ, 3年 / ギター, 5年 / クラリネット, 7年
6	ピアノ, 15年 / エレキベース, 4年
7	ギター, 6年
8	トランペット, 14年
9	ピアノ, 18年 / ユーフォニアム, 3年 / クラリネット, 6年
10	ピアノ, 10年
11	ピアノ, 3年 / エレクトーン, 10年
12	ピアノ, 10年 / サックス, 5年
13	ピアノ, 19年 / トロンボーン, 2年 / 和太鼓, 3年 / 篠笛, 3年
14	ピアノ, 12年 / フルート, 3年 / サックス, 3年 / ドラム, 2年 / ギター, 1年
15	ピアノ, 11年 / エレクトーン, 13年
16	エレクトーン, 2年 / ドラム, 3年
17	ピアノ, 16年 / ヴァイオリン, 17年 / ホルン, 5年
18	ピアノ, 10年 / ギター, 3年
19	チェロ, 10年 / ヴァイオリン, 3年 / 打楽器, 2年
20	ギター, 2年 / ベース, 1年
21	ピアノ, 1年 / エレクトーン, 6年 / エレキギター, 6年
22	トロンボーン, 2年 / トランペット, 3年 / ヴァイオリン, 1年 / エレキベース, 2年
23	ピアノ, 10年 / トランペット, 1年
24	ギター, 2年 / ヴァイオリン, 2年
25	ピアノ, 7年 / ドラム, 2年
26	ピアノ, 6年 / トロンボーン, 3年
27	ピアノ, 2年 / ヴァイオリン, 12年 / ヴィオラ, 7年

付 録 B 同意書

「即興演奏を伴うドラムアンサンブルに関する研究」 実験のご参加について

この実験は、様々な状況下でドラムパッドを用いてアンサンブルを行った時の**演奏時の情動と、音楽演奏について検討**することを目的としています。実験参加者にはまず音楽経験や性格に関する事前調査を行い、その後、提示された短く簡単な曲を指示に従って演奏し、アンケートによって情動状態を報告していただきます。ドラムパッドで打拍を行った際の時間や打拍の強さ、叩いた場所に対応する MIDI ノート番号といったデータが自動的に記録・保存されるほか、演奏動作の把握と分析のために演奏時のビデオ映像をご提供いただきます。実験に要する時間はおよそ 1 時間半を予定しています。

以上をご了承いただいたうえで、実験のご参加をお願いします。なお、実験参加はあくまでも任意であり、いつでも実験参加をとりやめることが可能です。また、本実験によって得られた個人情報は厳重に管理され、本研究の成果は、個人が特定されない形で発表・出版されます。

不明点、疑問やご意見がありましたら、いつでもお尋ねください。

[実験者]

筑波大学大学院図書館情報メディア研究科 博士前期課程 2 年 崎野 峻光

筑波大学図書館情報メディア系 助教 寺澤 洋子

アムステルダム大学 / ラドバウト大学 助教 貞方 マキ子

筑波大学図書館情報メディア系 教授 平賀 譲

筑波大学図書館情報メディア系 特任助教 松原 正樹

連絡先 〒305-8550

茨城県つくば市春日 1-2

筑波大学 図書館情報メディア系

Tel. 029-859-1302, Fax. 029-859-1093

実験同意書

筑波大学図書館情報メディア系
寺澤 洋子 殿

私は、研究課題「即興演奏を伴うドラムアンサンブルに関する研究」に関して、研究の目的や研究内容・方法、プライバシーの保護、身体面、精神面への配慮、不利益及び危険性に対する配慮、同意しないことの自由の保証等について説明文書に基づいて十分説明を受け、内容を理解し納得しましたので、私の自由意志により本研究の被験者になることに同意します。

ただし、説明にもあったように、この同意は一切の不利益を受けることなくいつでも撤回できるものであることを確認します。

平成 年 月 日

所属・職又は学年

連絡先電話番号又は E-mail address

実験参加者番号

ご署名

[実験者]

筑波大学大学院図書館情報メディア研究科 博士前期課程 2 年	崎野 峻光
筑波大学図書館情報メディア系 助教	寺澤 洋子
アムステルダム大学 / ラドバウト大学 助教	貞方 マキ子
筑波大学図書館情報メディア系 教授	平賀 譲
筑波大学図書館情報メディア系 特任助教	松原 正樹

付 録 C 音楽経験と実験参加者の性格調査アンケート

Big Five Personality Traits

*必須

実験参加者ID *

回答を入力

あなたの楽器演奏経験を記入して下さい。 *

回答を入力

以下のそれぞれの項目は、あなた自身にどれくらいあてはまりますか。非常にあてはまる～まったく当てはまらないの中で、自分にもっともあてはまると思うものを選択して下さい。 *

	非常にあ てはまる	かなりあ てはまる	ややあて はまる	どちらと も言えな い	あまり当 てはまら ない	ほとんど あてはま らない	まったく あてはま らない
話し好き	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
悩みがち	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
独創的な	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
いい加減な	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
温和な	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
無口な	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
不安になり やすい	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
多才の	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
ルーズな	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
短気	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
陽気な	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
心配性	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
進歩的	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
怠惰な	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
怒りっぽい	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Big Five Personality Traits

*必須

以下のそれぞれの項目は、あなた自身にどれくらいあてはまりますか。非常にあてはまる～まったく当てはまらないの中で、自分にもっともあてはまると思うものを選択して下さい。*

	非常にあ てはまる	かなりあ てはまる	ややあて はまる	どちらと も言えな い	あまりあ てはまら ない	ほとん どあては まらない	まった くあては まらない
外交的	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
気苦労の多い	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
洞察力のある	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
成り行きまかせ	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
寛大な	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
暗い	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
弱気になる	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
想像力に富んだ	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
不精な	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
親切な	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
無愛想な	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
傷つきやすい	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
美的感覚の鋭い	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
計画性のある	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
良心的な	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Big Five Personality Traits

*必須

以下のそれぞれの項目は、あなた自身にどれくらいあてはまりますか。非常にあてはまる～まったく当てはまらないの中で、自分にもっともあてはまると思うものを選択して下さい。*

	非常にあ てはまる	かなりあ てはまる	ややあて はまる	どちらと も言えな い	あまりあ てはまら ない	ほとん どあては まらない	まった くあては まらない
社交的	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
動揺しやすい	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
頭の回転の速い	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
無頓着な	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
協力的な	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
人嫌い	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
神経質な	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
臨機応変な	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
軽率な	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
とげがある	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
活動的な	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
くよくよしない	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
興味の広い	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
勤勉な	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
かんしゃくもち	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Big Five Personality Traits

*必須

以下のそれぞれの項目は、あなた自身にどれくらいあてはまりますか。非常にあてはまる～まったく当てはまらないの中で、自分にもっともあてはまると思うものを選択して下さい。*

	非常にあ てはまる	かなりあ てはまる	ややあて はまる	どちらと も言えな い	あまりあ てはまら ない	ほとん どあては まらない	まった くあては まらない
意思表示し ない	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
悲観的な	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
好奇心が強 い	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
無節操	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
自己中心的	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
積極的な	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
緊張しやす い	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
独立した	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
几帳面な	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
素直な	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
地味な	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
憂鬱な	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
呑み込みの 速い	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
飽きっぽい	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
反抗的な	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

付 録D 情動アンケート

アンケート

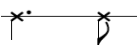

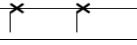
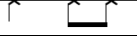
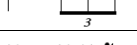


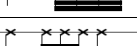
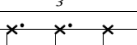
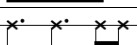
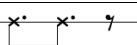


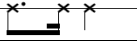
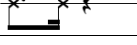


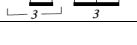

下の各項目について、現在あなたはどの程度感じていますか？
当てはまる数字に○をつけてください。

1. 全く感じない
2. あまり感じない
3. どちらでもない
4. すこし感じる
5. かなり感じる

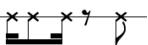
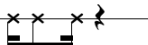
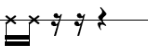
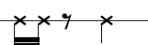

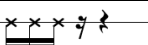
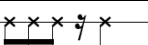
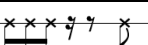
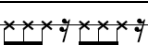
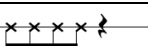
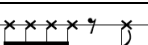
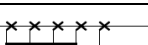
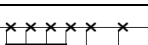
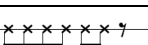
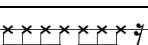
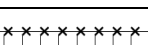
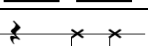
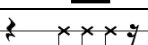
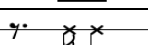
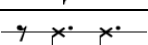
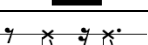
・ 沈んだ	1	2	3	4	5
・ のんびりした	1	2	3	4	5
・ うろたえた	1	2	3	4	5
・ 落ち着いた	1	2	3	4	5
・ 燃え上がった	1	2	3	4	5
・ 夢中な	1	2	3	4	5
・ 心苦しい	1	2	3	4	5
・ いやがった	1	2	3	4	5
・ 胸おどる	1	2	3	4	5
・ 安らいだ	1	2	3	4	5
・ 安心した	1	2	3	4	5
・ わくわくした	1	2	3	4	5

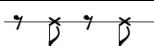
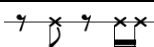

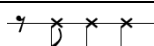
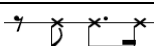
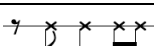

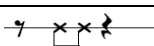
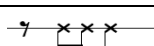
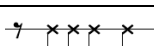
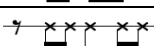
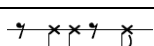
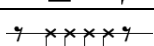

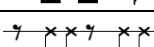
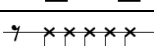

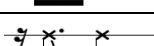
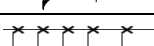
付 録 E 実験参加者が即興演奏で表出した リズムパターンの各条件毎の回数

実験参加者が、演奏実験の即興演奏のパートで表出した 2 拍のリズムパターンとその演奏回数を記録した。最も左の列の数字がパターンの通し番号であり、各条件の下に記されている数字が演奏回数を表している。なお、正確なリズムの判別が出来なかった等の理由から、視覚+聴覚条件は 8 個、聴覚のみ条件は 2 個、視覚のみ条件は 13 個の即興演奏のデータが欠落している。

	リズムパターン	視覚+聴覚条件	聴覚のみ条件	視覚のみ条件
1		5	4	2
2		1		
3		17	19	16
4		8	4	10
5		1		
6		1	2	
7		1	1	
8				2
9		1	1	1
10		7	4	5
11				1
12		2	1	2
13			1	
14		5	6	2
15		30	21	23
16				2
17		2	4	2
18			2	2
19		2	1	

20		16	19	9
21		1	4	
22		32	32	46
23		4	1	4
24		14	5	18
25		1		2
26		2	1	
27			2	
28		2		1
29				1
30		17	23	15
31		1	1	1
32		9	6	5
33			1	3
34			1	
35		2		1
36				1
37		1	1	
38				1
39		3	7	5
40		11	15	8

41		1	2	1
42		3	1	1
43		1		1
44		2	5	2
45		1		
46		3	3	1
47		4	3	4
48			1	
49		2	3	2
50		4	5	4
51			2	4
52		37	43	31
53		7	3	5
54		2	2	3
55		3	2	
56		2	1	2
57		4	2	
58				1
59		1		
60				1
61			1	

62		9	11	14
63		1	2	5
64		3	9	8
65		10	10	10
66		1		
67			2	3
68		1		4
69			1	
70		2	3	5
71		7	8	3
72			1	
73			2	1
74			1	
75				1
76		1		
77		4	2	2
78		1		
79			1	
80				1

付 録 F 外部仕様

F.1 動作環境

本研究における実験時の演奏データの記録を行うシステムの作動には Mac Book Pro (OS X バージョン 10.10.2, CPU 2.8 GHz Intel Core i7, メモリ 16GB) を使用した.

F.2 実行

ターミナル上で該当ファイルのあるディレクトリに移動し, **sudo npm start** と入力することで実行できる. また, 同時にサーバが立ち上がるのでウェブブラウザの URL に **localhost** と入力することで実験用のシステムをブラウザ上で操作することが出来る.

付 録 G 内部仕様

本システムは Node.js を用いて東京藝術大学博士課程の濱野峻行氏の協力のもと実装した。なお学士課程と修士課程の研究内容の違いからシステム内で使用されていない部分があるが、それらの説明は今回省略する。

G.1 主要なプログラム

app.js	システム全体を統括 (起動時に実行される)
rocktrap.js	インタフェース上で変更された情報を取得
index.js	rocktrap.js で取得された情報を各モジュールに送信
audio.js	音の入出力の設定
fluid.js	シンセサイザーである FluidSynth に関する処理
setting.js	音のマッピングなどの設定
stdin.js	標準入力に関する設定
midi_event.js	MIDI イベントの設定
midi_in.js	ドラムパッドから送信された MIDI データの処理
recorder.js	演奏データの記録

G.2 各モジュールの主要なファンクション

audio.js

本文中のジャックはオーディオのルーティングを行うプログラムを指す [22].

init	使用するモジュールの設定などを行う
setDevice	使用するデバイスの設定を行う
getOutputSettings	音の出力の設定を行う
initJack	ジャックのバッファサイズなどの初期設定を行う
connect	ポートの接続を行う
onExit	ジャックを終了させる

fluid.js

Fluid	Fluid.init を呼び出す
init	使用するモジュールの設定などを行う

Fluid.prototype 内部

init	ポートを開き, FluidSynth を立ち上げる
write	midi(外部モジュール) にメッセージを送る
restart	FluidSynth の再起動の設定を行う

外部から参照可能

writeAll	fluid.write を呼び出す
noteOn	writeAll に MidiEvent.noteOn の値を添えた MIDI データを送る
noteOff	writeAll に MidiEvent.noteOff の値を添えた MIDI データを送る
allnoteOff	全チャンネルで writeAll に MidiEvent.noteOff の値を添えた MIDI データを送る
restart	fluid.restart を呼び出す

midi_in.js

init	使用するモジュールの設定などを行う
createcallback	データの記録の指示や FluidSynth に音を鳴らさせる指示を出すファンクションを返す
getPortCount	ポートの数を取得する
setup	ポートの設定を行う

recorder.js

start	データの記録を開始する
stop	データの記録を停止する
write	ある音が鳴らされた時までの経過時間の情報を送信されてきた MIDI データに付加する
save	csv ファイルへの書き込みを行う

G.3 主要な外部モジュール

実験用システムにて使用した外部モジュールの一覧を記す.

express	Web アプリケーションフレームワークとして使用する [23]
async	非同期処理を行う際に使用する
jack-connector	Jack Audio を使用する際に使用する
jade	JavaScript のテンプレートエンジンとして使用する
microtime	現在時間をマイクロ秒で取得する
midi	MIDI 入出力処理を行う
midi-file-parser	MIDI データを Node.js 上で使えるようにするための処理を行う
nanotimer	処理の経過時間をナノ秒で取得する
underscore	配列やオブジェクトの処理を簡単に行うことが出来る

付 録H ソースコード

app.js

```
1  var express = require('express');
2  var path = require('path');
3  var favicon = require('static-favicon');
4  var logger = require('morgan');
5  var cookieParser = require('cookie-parser');
6  var bodyParser = require('body-parser');
7
8  var app = express();
9
10 var _ = require('underscore');
11 var modules = [
12   'audio',
13   'fluid',
14   'midi_player',
15   'midi_in',
16   'stdin',
17   'metronome',
18   'recorder'
19 ];
20 modules = _.map(modules, function (m) {
21   return require('./routes/' + m);
22 });
23 _.each(modules, function (m) {
24   if (_.has(m, 'init')) {
25     m.init();
26   }
27 });
28
29 var routes = require('./routes/index');
30
31 // view engine setup
32 app.set('views', path.join(__dirname, 'views'));
33 app.set('view engine', 'jade');
34
35 app.use(favicon());
36 app.use(logger('dev'));
37 app.use(bodyParser.json());
38 app.use(bodyParser.urlencoded());
39 app.use(cookieParser());
40 app.use(express.static(path.join(__dirname, 'public')));
41
42 app.use('/', routes);
43
44 /// catch 404 and forward to error handler
45 app.use(function(req, res, next) {
46   var err = new Error('Not Found');
```

```
47     err.status = 404;
48     next(err);
49 });
50
51 /// error handlers
52
53 // development error handler
54 // will print stacktrace
55 if (app.get('env') === 'development') {
56     app.use(function(err, req, res, next) {
57         res.status(err.status || 500);
58         res.render('error', {
59             message: err.message,
60             error: err
61         });
62     });
63 }
64
65 // production error handler
66 // no stacktraces leaked to user
67 app.use(function(err, req, res, next) {
68     res.status(err.status || 500);
69     res.render('error', {
70         message: err.message,
71         error: {}
72     });
73 });
74
75 // shutdown hook
76 process.on('SIGINT', function () {
77     process.exit();
78 });
79
80 process.on('exit', function () {
81     _.each(modules, function (m) {
82         if (_.has(m, 'onExit')) {
83             m.onExit();
84         }
85     });
86     _.delay(function () {
87         console.log('byebye. ');
88     }, 1000);
89 })
90
91 module.exports = app;
```

rocktrap.js

```
1  /*jslint browser: false,
2      newcap: false,
3      node: true,
4      nomen: true,
5      plusplus: false,
6      undef: false,
7      unparam: true,
8      vars: false,
9      evil: true,
10     white: false */
11  /*global _: true,
12     $: true,
13     GLOBAL: true
14  */
15
16  $(function () {
17
18      'use strict';
19
20      // audio out
21      (function ($el) {
22          _.each(GLOBAL.AUDIO_OUTPUTS, function (dev) {
23              $el.append(
24                  $('<option>')
25                      .val(dev)
26                      .text(dev)
27              );
28          });
29          $el.val(GLOBAL.CURRENT_AUDIO_OUTPUT).selectmenu('refresh');
30          $el.on('change', function () {
31              $.get('/audio-out', {dev: $(this).val()});
32          });
33      })($('#audio-out'));
34
35      $('#flip-solo').on('change', function () {
36          $.get('/solo/' + ($(this).prop('checked') ? 'on' : 'off'));
37      });
38
39      $('#flip-metro').on('change', function () {
40          var flag = $(this).prop('checked');
41          if (flag) {
42              $.get('/bpm/' + ($('#bpm-slider').val());
43          }
44          $.get('/metro/' + ($(this).prop('checked') ? 'on' : 'off'));
45      });
46
47      $('#bpm-slider').on('change', function () {
48          $.get('/bpm/' + $(this).val());
49      });
50
51      $('#flip-smf').on('change', function () {
52          var flag = $(this).prop('checked');
53          $('#flip-metro')
54              .prop('checked', 0)
55              .flipswitch('refresh');
56          $('#flip-metro-smf').flipswitch(flag ? 'disable' : 'enable');
```

```

57     $('#flip-metro').flipswitch(flag ? 'disable' : 'enable');
58     $('#bpm-slider').slider(flag ? 'disable' : 'enable');
59     $.get('/smf/' + (flag ? 'play' : 'stop'));
60 });
61
62 // MIDI_PLAY_SPEED
63 $('input[name="r-speed"]:radio').on('change', function () {
64     $.get('/midispeed/' + $('[name=r-speed]:checked').val());
65 });
66
67 /* MUSICAL_EXPRESSION
68 $('#musical-expression').on('change', function () {
69     $.get('/expression/' + ($(this).prop('checked') ? 'on' : 'off'));
70 });
71 */
72
73 // MUSICAL_EXPRESSION_PLUS_FAST
74 $('input[name="m-expression"]:radio').on('change', function () {
75     $.get('/expression/' + $('[name=m-expression]:checked').val());
76 });
77
78 $('#flip-metro-smf').on('change', function () {
79     $.get('/smf-metro/' + ($(this).prop('checked') ? 'on' : 'off'));
80 });
81
82 $('#flip-rec').on('change', function () {
83     $.get('/rec/' + ($(this).prop('checked') ? 'start' : 'stop'));
84 });
85
86 $('#button-save').on('click', function () {
87     if ($('#text-file').val() !== '') {
88         $.get('/save', {file: $('#text-file').val()});
89     }
90 });
91
92 $('#flip-fixvel').on('change', function () {
93     $.get('/fixvel/' + ($(this).prop('checked') ? 'on' : 'off'));
94 });
95
96 });
97
98 //EOF

```

index.js

```
1  /*jshint browser: false,
2      newcap: false,
3      node: true,
4      nomen: true,
5      plusplus: false,
6      undef: false,
7      unparam: true,
8      vars: false,
9      white: false */
10 /*global _: true
11 */
12
13 'use strict';
14
15 var express = require('express');
16 var router = express.Router();
17 var metro = require('./metronome');
18 var player = require('./midi_player');
19 var recorder = require('./recorder');
20 var audio = require('./audio');
21 var fluid = require('./fluid');
22
23 /* GET home page. */
24 router.get('/', function (req, res) {
25
26     var data;
27     try {
28         data = {
29             title: 'Rock Trap',
30             audioOutputs: audio.outputs.join(', '),
31             currentAudioOutput: audio.currentDev
32         };
33     } catch (e) {
34         console.log(e);
35     }
36     res.render('index', data);
37
38 });
39
40 router.get('/metro/:status', function (req, res) {
41
42     metro[req.params.status]();
43     res.send({});
44
45 });
46
47 router.get('/smf/:status', function (req, res) {
48
49     player[req.params.status]();
50     res.send({});
51
52 });
53
54 router.get('/smf-metro/:status', function (req, res) {
55
56     player.metro[req.params.status]();
```

```

57     res.send({});
58
59 });
60
61 router.get('/bpm/:val', function (req, res) {
62
63     metro.set(req.params.val);
64     res.send({});
65
66 });
67
68 router.get('/rec/:val', function (req, res) {
69
70     recorder[req.params.val]();
71     res.send({});
72
73 });
74
75 router.get('/save', function (req, res) {
76
77     recorder.save(req.query.file);
78     res.send({});
79
80 });
81
82 router.get('/audio-out', function (req, res) {
83
84     audio.setDevice(req.query.dev);
85     res.send({});
86
87 });
88
89 router.get('/solo/:status', function (req, res) {
90
91     fluid.setParam('solo', req.params.status === 'on');
92     res.send({});
93
94 });
95
96 router.get('/fixvel/:status', function (req, res) {
97
98     fluid.setParam('fixVelocity', req.params.status === 'on');
99     res.send({});
100
101 });
102
103 router.get('/midispeed/:val', function (req, res) {
104
105     player.midispeed(req.params.val);           // slow or middle or fast
106     res.send({});
107
108 });
109
110 router.get('/expression/:val', function (req, res) {
111
112     player.expression(req.params.val);           // off or on or onslow or
113     metro
114     res.send({});

```

```
114  
115  });  
116  
117  module.exports = router;  
118  
119  //EOF
```

audio.js

```
1  /*jshint browser: false,
2      newcap: false,
3      node: true,
4      nomen: true,
5      plusplus: false,
6      stupid: true,
7      undef: false,
8      unparam: true,
9      vars: false,
10     white: false */
11 /*global _: true
12 */
13
14 'use strict';
15
16 var Audio, _, settings, fluid, cp, jack, async;
17
18 _ = require('underscore');
19
20 Audio = (function () {
21
22     var cmd, initialized;
23
24     cmd = {
25         audiodevice: './resources/audiodevice'
26     };
27
28     return {
29
30         setDevice: function (dev) {
31
32             this.currentDev = dev;
33             cp.exec(
34                 [cmd.audiodevice, 'output', '\"' + this.currentDev + '\"'].
35                 join(' '),
36                 function (error, stdout, stderr) {
37                     if (dev === 'JackRouter') {
38                         try {
39                             jack.closeClientSync();
40                         } catch (e) {}
41                         jack.openClientSync('JackRouter');
42                         jack.activateSync();
43                         console.log('JACK OPENED!');
44                     }
45                     if (fluid.initialized()) {
46                         fluid.restart();
47                     }
48                 }
49             );
50         },
51
52         getOutputSettings: function () {
53
54             async.series([
55                 function (callback) {
```

```

56         cp.exec(
57             [cmd.audiodevice, 'output', 'list'].join(' '),
58             function (error, stdout, stderr) {
59                 this.outputs = _.compact(stdout.split('\n'));
60                 callback();
61             }.bind(this)
62         );
63     }.bind(this),
64     function (callback) {
65         cp.exec(
66             [cmd.audiodevice, 'output'].join(' '),
67             function (error, stdout, stderr) {
68                 this.currentDev = stdout.replace(/\n/, '');
69                 callback();
70             }.bind(this)
71         );
72     }.bind(this)
73 ], function (err) {
74     if (err) {
75         console.log(err);
76     }
77     initialized = true;
78 });
79
80 },
81
82 initJack: function () {
83
84     var jackd;
85
86     jackd = cp.spawn('jackd', ['-d', 'coreaudio', '-p', 32]);
87     jackd.stdout.on('data', function (chunk) {
88         chunk = chunk.toString();
89         if (chunk.indexOf('running...') >= 0) {
90             this.setDevice('JackRouter');
91             this.getOutputSettings();
92         }
93         process.stdout.write(chunk);
94     }.bind(this));
95
96 },
97
98 connect: function (index) {
99
100     var port = {};
101
102     if (this.currentDev !== 'JackRouter') {
103         return;
104     }
105
106     port.output = 'fluidsynth' + ((index > 0) ? '-0' + index : '')
107         + ':out';
108     port.input = 'system:playback_';
109
110     _.times(2, function (i) {
111         var tryConnect = function () {
112             console.log(

```

```

113         port.output + (i + 1).toString() +
114         ' => ' +
115         port.input + (index * 2 + i + 1 + settings.
            audioChannelOffset).toString()
116     );
117     try {
118         jack.connectPortSync(
119             port.output + (i + 1).toString(),
120             port.input + (index * 2 + i + 1 + settings.
                audioChannelOffset).toString()
121         );
122     } catch (err) {
123         console.log(err);
124         _.delay(tryConnect, 1000);
125     }
126 };
127 tryConnect();
128 });
129
130 },
131
132 onExit: function () {
133
134     try {
135         jack.deactivateSync();
136         jack.closeClientSync();
137     } catch (e) {}
138     cp.exec('killall jackd');
139
140 }
141
142 };
143
144 }());
145
146 function init() {
147
148     settings = require('./settings');
149     fluid = require('./fluid');
150     cp = require('child_process');
151     jack = require('jack-connector');
152     async = require('async');
153
154     if (settings.useJack) {
155         Audio.initJack();
156     } else {
157         Audio.getOutputSettings();
158     }
159
160 }
161
162 module.exports = (function () {
163
164     return _.extend(Audio, {
165         init: init
166     });
167
168 }());

```

169	
170	//EOF

fluid.js

```
1  /*jshint browser: false,
2      newcap: false,
3      node: true,
4      nomen: true,
5      plusplus: false,
6      undef: false,
7      unparam: true,
8      vars: false,
9      white: false */
10 /*global _: true
11 */
12
13 'use strict';
14
15 var Fluid, _, settings, N, MidiEvent, spawn, exec, midi, audio, instance,
    initialized;
16
17 initialized = false;
18 _ = require('underscore');
19
20 Fluid = function (index) {
21     this.init(index);
22 };
23
24 Fluid.prototype = {
25
26     init: function (index) {
27
28         var openPort;
29
30         if (_.isUndefined(index)) {
31             index = this.index;
32         }
33         this.index = index;
34
35         openPort = function () {
36
37             var p;
38
39             _.times(this.output.getPortCount(), function (i) {
40                 if (this.output.getPortName(i) === 'fluid-' + index) {
41                     p = i;
42                 }
43             }).bind(this));
44
45             if (_.isUndefined(p)) {
46                 setTimeout(openPort, 200);
47                 return;
48             }
49
50             this.portOpened = true;
51             try {
52                 this.output.openPort(p);
53             } catch (e) {
54                 this.portOpened = false;
55             }
56         }
57     }
58 }
```

```

56         if (settings.useJack) {
57             audio.connect(index);
58         }
59
60     }.bind(this);
61
62     this.portOpened = false;
63     this.output = new midi.output();
64     openPort();
65
66     spawn(
67         'fluidsynth',
68         [
69             '-a',
70             'coreaudio',
71             '-z',
72             '32',
73             '-g',
74             '2',
75             '-p',
76             'fluid-' + index,
77             '-j',
78             '-f',
79             'resources/fluid.conf',
80             'resources/titanic.sf2'
81         ],
82         {stdio: ["pipe", process.stdout, process.stderr]}
83     );
84
85 },
86
87 write: function (msg) {
88
89     if (this.portOpened) {
90         this.output.sendMessage(msg);
91     }
92
93 },
94
95 restart: function () {
96
97     if (this.portOpened) {
98         try {
99             this.output.closePort();
100         } catch (e) {}
101     }
102
103     exec('killall fluidsynth', function () {
104         Fluid.prototype.init.apply(this);
105     }).bind(this));
106
107 },
108
109 };
110
111 function init() {
112
113     settings = require('./settings');

```

```

114     N = settings.nAudioOutputGroups;
115     MidiEvent = require('./midi_event');
116     spawn = require('child_process').spawn;
117     exec = require('child_process').exec;
118     midi = require('midi');
119     audio = require('./audio');
120
121     instance = [];
122     _.times(N, function (i) {
123         _.delay(function () {
124             instance.push(new Fluid(i));
125             if (i === N - 1) {
126                 initialized = true;
127             }
128         }, 1000 * i + 1000);
129     });
130
131 }
132
133 module.exports = (function () {
134
135     return {
136         init: init,
137         initialized: function () {
138             return initialized;
139         },
140         writeAll: function (msg) {
141             _.each(instance, function (fluid) {
142                 fluid.write(msg);
143             });
144         },
145         noteOn: function (ch, nn, vel, all) {
146             if (this.fixVelocity) {
147                 vel = settings.fixedVelocity;
148             }
149             if (!all && this.solo) {
150                 instance[ch].write([MidiEvent.noteOn, nn, vel]);
151             } else {
152                 this.writeAll([MidiEvent.noteOn + ch, nn, vel]);
153             }
154         },
155         noteOff: function (ch, nn, all) {
156             if (!all && this.solo) {
157                 instance[ch].write([MidiEvent.noteOff, nn, 0]);
158             } else {
159                 this.writeAll([MidiEvent.noteOff + ch, nn, 0]);
160             }
161         },
162         allNotesOff: function (ch) {
163             this.writeAll([MidiEvent.allNotesOff.MSB + ch, MidiEvent.
164                 allNotesOff.LSB, 0]);
165         },
166         restart: function () {
167             _.each(instance, function (fluid) {
168                 fluid.restart();
169             });
170         },
171         setParam: function (param, val) {

```

```
171         this[param] = val;
172     }
173 };
174
175 }();
176
177 //EOF
```

setting.js

```
1  /*jslint browser: false,
2      newcap: false,
3      node: true,
4      nomen: true,
5      plusplus: false,
6      undef: false,
7      unparam: true,
8      vars: false,
9      white: false */
10 /*global _: true
11 */
12
13 module.exports = {
14     useJack: true,
15     nAudioOutputGroups: 1, // 4
16     audioChannelOffset: 0, // 2
17     fixedVelocity: 100, // 100
18     mapping: {
19         midiIn: {
20             pitch: {
21                 36: 77,
22                 38: 77,
23                 42: 76,
24                 46: 76,
25                 48: 77,
26                 45: 77,
27                 49: 76,
28                 51: 76
29             }
30         },
31         filePlay: {
32             pitch: {
33                 54: 43,
34                 57: 45,
35                 60: 47,
36                 63: 48,
37
38                 39: 61,
39                 40: 60
40             }
41         }
42     }
43 };
44
45 //EOF
```

stdin.js

```
1  /*jshint browser: false,
2      newcap: false,
3      node: true,
4      nomen: true,
5      plusplus: false,
6      undef: false,
7      unparam: true,
8      vars: false,
9      evil: true,
10     white: false */
11  /*global _: true
12  */
13
14  'use strict';
15
16  var metro, player, rec, fluid, _;
17
18  metro = require('./metronome');
19  player = require('./midi_player');
20  rec = require('./recorder');
21  fluid = require('./fluid');
22  _ = require('underscore');
23
24  process.stdin.on('data', function (chunk) {
25      try {
26          eval(chunk.toString());          // convert to String
27      } catch (e) {
28          console.log('- ' + e.message);    // error message
29      }
30  });
31
32  exports.module = {};
33
34  //EOF
```

midi_event.js

```
1  /*jslint browser: false,
2      newcap: false,
3      node: true,
4      nomen: true,
5      plusplus: false,
6      undef: false,
7      unparam: true,
8      vars: false,
9      white: false */
10 /*global _: true
11 */
12
13 module.exports = {
14     noteOn: 144,
15     noteOff: 128,
16     allNotesOff: {
17         MSB: 176,
18         LSB: 123
19     }
20 };
21
22 //EOF
```

midi_in.js

```
1  /*jslint browser: false,
2      newcap: false,
3      node: true,
4      nomen: true,
5      plusplus: false,
6      undef: false,
7      unparam: true,
8      vars: false,
9      white: false */
10 /*global _: true
11 */
12
13 'use strict';
14
15 var MidiIn, _, settings, MidiEvent, midi, fluid, recorder;
16
17 _ = require('underscore');
18 settings = require('./settings');
19
20 MidiIn = (function () {
21
22     var mapping, input;
23
24     mapping = settings.mapping.midiIn.pitch;
25
26     function createCallback(i, dt, msg) {
27
28         return function (dt, msg) {
29             console.log('In ' + i.toString() + ':\t' + msg.join('\t'));
30             if (msg[0] >= MidiEvent.noteOn && msg[0] < MidiEvent.noteOn +
31                 16) {
32                 fluid.noteOn(i, mapping[msg[1]] || msg[1], msg[2]);
33                 _.delay(function () {
34                     fluid.noteOff(i, mapping[msg[1]] || msg[1]);
35                 }, 100);
36                 recorder.write(['player', i, mapping[msg[1]] || msg[1], msg
37                     [2]]);
38             }
39         };
40     }
41
42     return {
43
44         getPortCount: function () {
45
46             var tmp, nPorts;
47
48             tmp = new midi.input();
49             nPorts = tmp.getPortCount();
50             tmp.closePort();
51
52             return nPorts;
53         },
54     }
```

```

55     setup: function () {
56
57         var nPorts;
58
59         nPorts = this.getPortCount();
60
61         input = [];
62         _.times(nPorts, function (i) {
63             var port;
64             port = new midi.input();
65             if (port.getPortName(i) === 'USB2MIDI') {
66                 port.openPort(i);
67                 port.on('message', createCallback(input.length));
68                 input.push(port);
69             } else {
70                 port.closePort();
71             }
72         });
73     }
74 }
75
76 };
77
78 }());
79
80 function init() {
81
82     MidiEvent = require('./midi_event');
83     midi       = require('midi');
84     fluid      = require('./fluid');
85     recorder   = require('./recorder');
86
87     MidiIn.setup();
88
89 }
90
91 module.exports = (function () {
92
93     return _.extend(MidiIn, {
94         init: init
95     });
96
97 }());
98
99 //EOF

```

recorder.js

```
1  /*jshint browser: false,
2      newcap: false,
3      node: true,
4      nomen: true,
5      plusplus: false,
6      undef: false,
7      unparam: true,
8      vars: false,
9      white: false */
10 /*global _: true
11 */
12
13 /*jshint browser: false,
14      newcap: false,
15      node: true,
16      nomen: true,
17      plusplus: false,
18      undef: false,
19      unparam: true,
20      vars: false,
21      white: false */
22 /*global _: true
23 */
24
25 'use strict';
26
27 var Recorder, _, time;
28
29 _ = require('underscore');
30 time = require('microtime');
31
32 Recorder = (function () { // module.exports = Recorder;
33
34     var recording, data, timeStamp;
35
36     return {
37
38         start: function () {
39
40             recording = true;
41             data = [];
42             timeStamp = time.now();
43
44         },
45
46         stop: function () {
47
48             recording = false;
49
50         },
51
52         write: function (msg) {
53
54             if (!recording) {
55                 return;
56             }
```

```
57         msg.unshift((time.now() - timeStamp) / 1000000);
58         data.push(msg);
59     },
60     save: function (file) {
61         var str = '';
62         _$.each(data, function (v) {
63             str += v.join(',') + '\n';
64         });
65         require('fs').writeFile('data/' + file + '.csv', str);
66     }
67 };
68 }());
69 module.exports = Recorder;
70 //EOF
```