筑波大学大学院博士課程

システム情報工学研究科修士論文

演奏練習における音声ログの楽譜への対応付けと可視化および演奏に関する議論への活用

坂本 雄彦

修士（工学）

(コンピュータサイエンス専攻)

指導教員　高橋 伸

2015 年 3 月

概要

複数人数で行う演奏活動は、演奏練習においてしばしば議論が行われる。しかし限られた練習回数・練習時間の中で議論に時間を割きすぎてしまうと、演奏に充てる時間が圧迫されてしまうという問題がある。先行研究では、本来練習時間中に行われていた議論を、web を介して遠隔非同期的に行うシステムを実現することによって演奏練習の効率化を図った。本研究では先行研究のシステムに、演奏練習の音声ログの「楽譜への対応付け」、「対応付けの可視化」、「議論への活用」という 3 つのアプローチすることで、議論の質の向上や更なる練習の効率化を狙う。今回作成したシステムは web アプリケーションとして実装され、演奏練習中・演奏練習以外の 2 つの場面においてユーザからアクセスされる。演奏練習中には音声ログの録音と、楽譜への対応付けのための簡易な操作をユーザに行ってもらう。演奏練習以外にはユーザは音声ログを聴取し、時には音声ログの一部を参照しながら演奏に関する議論を楽譜上で行うことができる。本研究で提案した音声ログの活用手法と作成したシステムの有用性を評価するため、3 つの評価実験を行った。音声ログの楽譜との対応付け推定に関する実験では、概ね期待通りの成果をあげるとともに新たな課題を発見した。可視化に関する実験では、可視化された音声ログは非常に高い評価を得られ、またユーザがどのような色付けを好んでいるかを明らかにした。演奏練習と議論におけるシステムの試用実験では、システムの有用性と需要について特に高い評価が得られ、今後のシステムの発展に対して多くの示唆を得ることができた。

目 次

第 1 章　序論 **1**

1.1　背景 1

1.2　目的とアプローチ 2

1.2.1　目的 2

1.2.2　アプローチ 2

音声ログの楽譜への対応付け 3

対応付けの可視化 3

音声ログの議論への活用 3

web アプリケーションとしての実装 4

1.3　構成 4

第 2 章　関連研究 **5**

2.1　コンピュータを用いた演奏練習の支援 5

2.2　音声ログの楽譜との対応付け 6

2.3　遠隔非同期な協調作業支援 7

第 3 章　演奏練習の音声ログの活用手法 **8**

3.1　楽譜への対応付け 8

3.2　可視化 11

3.3　議論での活用 13

第 4 章　Co-Musicator システム **15**

4.1　システム概要 15

4.2　画面構成 16

4.2.1　ナビゲーションメニュー 17

4.2.2　楽譜ビュー 18

4.2.3　音声ログビュー 18

4.3　システム利用方法 19

4.3.1　演奏練習時 20

4.3.2　演奏練習外 20

第 5 章　実装 **22**

5.1　システム構成 22

5.2 音声の取得・録音 24
5.3 演奏位置推定・楽譜への対応付け 26
5.4 可視化 27
5.5 楽譜上での議論 28

第 6 章 評価 **31**
6.1 評価に用いた演奏練習データ 31
6.2 音声ログの楽譜への対応付けの推定 31
6.2.1 実験内容 31
6.2.2 結果と考察 32
6.3 音声ログの楽譜への対応付けの可視化 35
6.3.1 実験内容 35
6.3.2 結果と考察 36
6.4 演奏練習と議論におけるシステムの有用性 37
6.4.1 実験内容 37
(1) 対面環境におけるシステムを利用した練習 38
(2) 遠隔非同期環境におけるシステムを利用した議論 39
(3) 対面環境における 2 度目の練習 39
6.4.2 結果と考察 41

第 7 章 結論 **48**

謝辞 **49**

参考文献 **50**

図目次

1.1 楽曲内の演奏位置に基づく色付け 3

3.1 音声ログの楽譜への対応付けの概念図 9
3.2 音声ログの楽譜への対応付けのアルゴリズム 10
3.3 音声ログの可視化のアルゴリズム 11
3.4 音声ログの可視化例 12
3.5 色相・温度・明暗による色付け方法 13
3.6 音声ログの議論での活用 14

4.1 システム概念図 15
4.2 画面構成：楽譜ビュー 16
4.3 画面構成：音声ログビュー 17
4.4 ナビゲーションメニュー 18
4.5 音声ログ上に表示されたタグ 19

5.1 システム構成図 22
5.2 ソフトウェア構成図 23
5.3 録音の流れ 25
5.4 楽曲構造データ, 音声ログの音量データ, クリップデータの JavaScript 表現 . . 26
5.5 3 種類の色付け方法 27
5.6 楽譜ビューの HTML 構造 29

6.1 システムが推定した対応付け（青）と実際の対応（赤）：音声ログ A 32
6.2 システムが推定した対応付け（青）と実際の対応（赤）：音声ログ B 32
6.3 システムが推定した対応付け（青）と実際の対応（赤）：音声ログ C 33
6.4 システムが推定した演奏位置と実際の演奏位置との差異 34
6.5 説明に使用した音声ログのシステム上の可視化表現 35
6.6 アンケート結果：音声ログの楽譜への対応付けの可視化 36
6.7 (1),(3) における演奏練習の環境 38
6.8 (1),(3) におけるシステムの利用環境 39
6.9 (3) において過去の議論内容を確認する被験者 40
6.10 アンケート結果：演奏練習と議論におけるシステムの有用性 42
6.11 実験で得られたコメント：短い議論 44

6.12 実験で得られたコメント：長い議論 45
6.13 実験で得られたコメント：音声ログを含むコメント 46
6.14 実験で得られたコメント：休符に対するコメントエリア 47

第1章　序論

1.1　背景

オーケストラ・吹奏楽・ロックバンドなどの複数人数で行う演奏活動は、今日ではプロ・アマチュア問わず盛んに行われている。多くの場合これらの演奏活動には演奏会や収録などの本番があり、それらにむけて繰り返し演奏練習を行っていく。演奏練習にはいくつかの種類があり、あらゆる楽曲に通ずる演奏の基礎的な技能を高めるために行われる「基礎練習」と、上述の本番へ向けて楽曲の演奏の完成度を高めるために行われる「演奏練習」があり、またそれぞれについて個人で行われるものと複数人数で行われるものがある。これらのうち、本研究では特に複数人数での演奏練習に焦点を当てる。

一般的な演奏練習では演奏団体のメンバーがある程度の音量を出しても問題ない場所に集まり、演奏したり会話したりする。演奏者たちは音を出し演奏する以外に、楽曲の解釈について議論を交わすことがある。これは楽譜には演奏者の解釈によって左右される曖昧性（音の長さやテンポ、音の質など）があるためである。しかし、本番までの限られた練習回数・練習時間の中で演奏練習時に議論が白熱しすぎると、演奏に充てる時間が圧迫されてしまうという問題がある。この問題を解決するために、演奏に関する議論を web 上で遠隔非同期的に行うことで演奏練習の効率化を狙う取り組みを先行研究で行ってきた [1]。先行研究のシステムでは、web ページに表示された楽譜上へのコメントの投稿や、音の形の図示・音楽記号の書込みをすることができた。

本研究では先行研究を発展させる形で、更なる演奏者の支援を行う。先行研究において、システムのユーザは演奏練習以外の空いた時間を利用して演奏に関する議論を行うことができたが、演奏練習の時間では特にシステムを利用することはなかった。そこで今回、演奏練習の時間と演奏練習以外の時間をより密に連携するため、演奏練習の録音に着目した。自身の演奏を客観的に聴くことで演奏中には気がつかなかったテンポや音程のズレなどの課題を発見することができ、多くの演奏団体が演奏練習の録音を行なっている。録音された音声（以降、音声ログ）は WAV ファイルや MP3 ファイルとなって web 上のファイルアップローダやファイル共有ソフトなどを介してメンバーに共有され、演奏者各自で聴取して反省したり今後の練習の方針を考えたりする。

音声ログの共有から利用までの過程には、いくつかの問題点が存在する。まず、録音から音声ログのメンバー間での共有までの手間が大きいことが挙げられる。多くの演奏団体は、まず演奏練習を IC レコーダなどの機器を用いて録音し、PC へ取り込みファイル形式や圧縮率の変換を行い、web 上のファイルアップローダやファイル共有ソフトなどを利用する。この一連のプロセスはやや冗長であると言える。次の問題点として、音声ログの中から聴きたい

箇所を探しづらいことが挙げられる。例えば「1 時間ほどの長さのある音声の中から、150 小節目を演奏している部分を聴きたい」という場合には、音声再生アプリケーションで細かくシークバーを動かして該当する箇所を探さなければならない。これは演奏練習の音声ログに限らず多くの音声ファイルに共通することではあるが、演奏練習の音声ログは演奏と議論や会話とが混在しているため、上述のような特定の箇所を抜き出して聴きたいという思いは強くなると思われる。また別の問題点として、音声ログの聴取から議論へ繋げにくいとことが挙げられる。音声ログは演奏者一人ひとりが聴取して、議論内容のおさらい・演奏の問題点の洗い出し・次回以降の練習の方針案の練り出しなどを行う。そのため音声ログをベースにした議論や複数人数での分析は行われづらく、その後の演奏練習の際に更に時間を欠けて議論を行うことになる。

本研究ではこれらの問題を解決するため、演奏練習の録音・共有・議論が同一のプラットフォームで行うことができるシステムを開発した。ユーザは演奏練習時に web ブラウザからシステムにアクセスし、幾つかの簡単な操作を交えて演奏練習を録音する。音声ログはシステムに予め与えられた楽曲構造データと用いることで直ちに楽譜と対応付けられ、音声ログのどの地点が楽譜上のどの箇所を演奏しているかが可視化される。またユーザは、演奏練習時以外にも同様にシステムにアクセスし、音声ログを聴取したり音声ログを引用したりして楽譜上で議論を行うことができる。これらの機能によってユーザは録音にかかる手間の削減や音声ログの有効活用を実現する。

1.2 目的とアプローチ

1.2.1 目的

本研究の目的は、音楽活動団体を対象として前節で上げた 3 つの問題点を解決することにある。すなわち、録音から音声ログの共有までの手間を削減すること、音声ログの聴きたい箇所を聴けるようにすること、音声ログを用いた議論や分析を複数人数で円滑に行えるようにすることである。またこれらの問題の解決によって、演奏練習の効率化ひいては演奏そのものの質の向上を最終的なゴールとして掲げる。

なお本研究の対象となる音楽活動団体とは、基本的にはオーケストラ・吹奏楽・ロックバンドなどの複数人数で行われる演奏団体であるが、これには声楽などの楽器を使わない音楽団体も含まれるものとする。楽器を用いる団体であっても演奏団体の中でも楽譜をまったく必要としない演奏の場合は対象外となる。また、複数人数であると述べたが音楽教室のレッスンのような教師と生徒が 1 対 1 の場合であっても、実際の演奏者が複数人数でなくても演奏に関する議論を行う者が複数人数であれば対象となる。

1.2.2 アプローチ

前節の目的を達成するために、本研究では以下のアプローチを取る。それぞれについて詳細な説明は第 3 章を参照されたい。

音声ログの楽譜への対応付け

システムには予め演奏練習を行う楽曲の構造データ（テンポ・拍子・小節数など）を与えておく。演奏練習の録音中に、ユーザは演奏を開始する練習番号ないし小節番号をシステム上で選択してもらい、演奏停止時に再度システム上の演奏停止ボタンを選択してもらう。これらのユーザ操作による時刻と小節番号のデータを楽曲構造データと照らし合わせることで、音声ログの楽譜上における地点推定を行う。

対応付けの可視化

音声ログと楽譜との対応付けをユーザが利用しやすい形にするため、音声ログをその対応関係と音量を用いて可視化する。音声ログの経過時間を横軸に置き、音量を縦軸に取る。すなわち音量が大きい箇所（＝演奏している箇所）の幅は大きく、音量が小さい箇所（＝会話などをしている箇所）の幅を小さく表示する。楽曲の先頭から末尾までを一律な色相の変化と見立て、上述の対応付けを基に色付けを行う（図 1.1）。また、演奏していない箇所を灰色で塗る。これにより、長い音声ログが複数ある場合でもユーザは容易に聴きたい箇所を探し出すことができる。

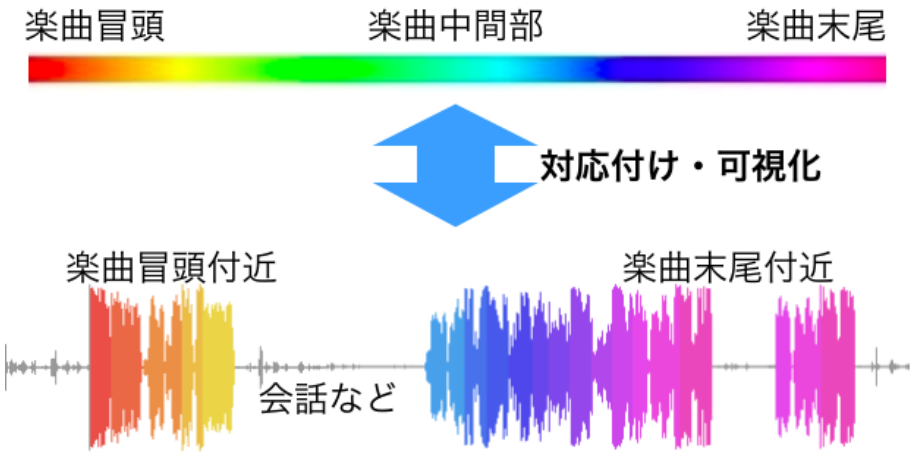

図 1.1: 楽曲内の演奏位置に基づく色付け

音声ログの議論への活用

先行研究のアプローチである楽譜上の特定箇所にコメントを付ける手法を踏襲しつつ、そこに音声ログを添付することで音声ログについての議論・分析を可能にする。音声ログの中から一部範囲を指定して切り出し、コメントを投稿する際に切り出した音声ログを添える選択をすることで、音声ログの特定箇所について言及することができる。

web アプリケーションとしての実装

また、これらの機能を単一の web アプリケーションとして実装することで、マルチデバイス・マルチプラットフォームでの利用を可能にする。いわゆる HTML5 の技術を用いることによって、システムのユーザは各々が空いた時間に、各々の PC・スマートフォン・タブレット端末からシステムにアクセスして議論をすすめることができる。

1.3 構成

本論文の構成は以下の通りである。本章では、演奏練習における録音を中心として現状の問題点を挙げ、それを踏まえて本研究の目的とアプローチを述べた。第 2 章では、録音を用いた演奏練習の支援や遠隔非同期な議論の支援など、本研究に関連する研究について述べ本研究の立ち位置を明確にする。第 3 章では、本研究における音声ログの活用手法について述べる。第 4 章では作成したシステムの設計について述べ、続く第 5 章ではシステムの実装について述べる。第 6 章では実験を通じて行なったシステムの評価について述べ、最後に第 7 章で結論を述べる。

第2章　関連研究

2.1　コンピュータを用いた演奏練習の支援

コンピュータを用いた演奏練習の支援は数多く行われてきている。Bellini らの研究 [2] ではアンサンブル練習をコンピュータが支援するシステムとして MOODS を開発している。MOODS はオーケストラ等のアンサンブル練習の場面で奏者それぞれの譜面台をディスプレイとし、指揮者や各パートの首席奏者などが楽譜に対して書き込みを行うと、他の奏者のディスプレイに表示されている楽譜が同期されて書き込みが行われるというシステムである。Sawchuk らはオーケストラの演奏において奏者それぞれが遠隔な場所から一つのオーケストラの練習に参加するためのシステム DIP を開発している [3]。DIP では演奏者それぞれが全く別の場所に居ながら、巨大なスクリーンとスピーカーを通してリアルタイムで相互に演奏を聞くことができる。また応用として演奏者がその場に居ないバーチャルなコンサートや遠隔での楽器のレッスン、非同期式に演奏を鑑賞することも可能となっている。Akoumianakis らの研究 [4] では遠隔で同期式のアンサンブル練習を行うためのプロトタイプツールキット DIAMOUSES を開発している。DIAMOUSES はパートごとの演奏を録音し、オンラインで録音データを通信することで遠隔に居ながらアンサンブルセッションを楽しむことができる。また、DIAMOUSES には楽譜を表示する機能もあり、簡単な書き込みを行って楽譜を共有することができる。Goto らの開発したシステムである Remote-GIG[6] では、遅延の大きいインターネットを介して、遠隔地にいるユーザ同士が 12 小節のブルース等をリアルタイムに合奏することを可能にした。お互いにコード進行の 1 周期分（数十秒）だけ遅れた相手の演奏を聴くことで、ネットワーク遅延を吸収した合奏を実現した。Gurevich は音楽初心者向けの、リアルタイムな音楽コラボレーションシステムである JamSpace を開発している [8]。

また、演奏練習そのものではなく音楽理解や音楽学に関する支援の研究も行われている。Hirata らの研究 [5] では音楽デザインを支援するシステム Music Resonator を開発している。Music Resonator は複数のユーザがアノテーションの付加された楽曲の断片を投稿することで、インタラクティブな音楽共創とコミュニケーションの行うことができる。Lim らは演奏を録音し音程の情報を多角的に表示することでピッチの揺れやずれを指摘するシステムを開発している [7]。また、松原らの研究 [9] ではオーケストラスコアの理解を支援するシステム Score Illuminator を開発している。オーケストラにおいて練習する楽曲のスコアを読み楽曲構造を理解することはスコアリーディングと呼ばれ、音楽表現を考える上で非常に重要であるとされている。Score Illuminator はオーケストラスコアのパート間の類似度を自動解析した上で色付けを行い、楽曲内における役割（主旋律・対旋律・伴奏など）の分類をすることができる。またユーザがその自動で色分けされたスコアに手を加えていくことで、音楽的な解釈や理解

を深めることを目的としている。iPad 用アプリケーションである piaScore[10] では、デジタル化された楽譜上にユーザが自由な書き込みを行うことができる他、メトロノーム機能やジェスチャによる譜めくり機能など実用性の高い機能が多数組み込まれている。

これらの研究に対し本研究は、演奏練習という行為そのものではなく演奏に関する議論を支援することを目的としている。音声ログを流しながら重ねて演奏したり、複数の音声ログを合わせてひとつの音楽を作ったりなどといったものではなく、あくまで対面環境での演奏練習を主軸に据えつつ付加的な議論を遠隔非同期で行ったり演奏練習の効率を高めたりすることができる。

2.2 音声ログの楽譜との対応付け

音楽情報処理の分野において、演奏の録音を楽譜と対応付ける研究は長い間研究され続けてきた。音高を利用した自動採譜は、後藤によって音源数を仮定しない音高推定手法 PreFEst[11] が提案された。この手法では、高調波構造の形状を表す確率分布をあらゆる音高に対して用意し、それらの混合分布（重み付き和）として入力の周波数成分をモデル化する。そのモデルパラメータを、EM(Expectation-Maximization) アルゴリズムを用いて最大事後確率推定することで、混合音の各構成音の音量（重み）と高調波構造の形状が同時に求められる点に新規性があった。原理的に非高調波構造にも拡張でき、混合音理解のフレームワークと位置付けられる。Klapuri は最も優勢な音高の推定とその高調波成分の除去を繰り返すことで、複数の楽器からなる音声の構成要素を順次求めていく手法 [12] を提案した。Davy らは音の波形の重み付き和として時間領域でモデル化し、MCMC(MarkovchainMonteCarlo) アルゴリズムによって同時発音数・各音を構成する周波数成分数・音高・音量等のモデルパラメータを推定する手法 [13] を提案した。Cemgil らは楽譜が（局所的な）あるテンポで演奏されて波形が生成される過程を、グラフィカルモデルで音符とテンポ・波形を関連付けてモデル化し、そのすべてを推定する手法 [14] を提案した。Kameoka らは、周波数成分を高調波構造の拘束下でクラスタリングする問題と定式化し、赤池情報量規準 (AIC) を最小にするクラスタ数（音源数）を求めながら、各クラスタの重心（音高）と重み (音量) を推定する手法 [15] を提案した。階層的なビート構造（小節）を得るビートトラッキング・小節推定に関する研究としては、テンポ変化に対応できるビートトラッキング手法 [16] が Scheirer によって提案された。

これらの研究は演奏の音声ログと楽曲構造データを用意し、ほぼ全自動で音声ログと楽譜との対応付けを行うものである。それに対して本研究は演奏「練習」の音声ログを対象としているため、会話や雑音などが音声ログ中に多く入っている点で扱いが異なっている。そのため本研究では、演奏練習の録音中にユーザに簡易な操作を行ってもらうことで、複雑な計算を用いないで音声ログを楽譜と対応付ける手法を提案する。

2.3 遠隔非同期な協調作業支援

遠隔で非同期なコミュニケーションや協調作業の支援に関する研究は数多くなされてきた。Goto らは音声データからリズムやメロディなどを抽出し、ユーザに操作可能な形で表示することで能動的な音楽鑑賞が可能なシステム Songle を開発している [17]。また、Heer らはデータ分析のためのシステム sense.us を開発している [18]。sense.us は web ブラウザ上で動作し、アメリカの国勢調査 (census) の統計データを可視化するとともに、それについてグラフィカルな書き込みを用いた議論が行える。データに対するグラフィカルなアノテーション機能や、閲覧したページをストックして自分の意見を投稿する際に簡単にリンクを貼ることができる機能など、スムーズな議論のための多くの工夫がなされている。Phalip らは映像作家と作曲家がネットを介して一つの映像作品を作り上げる作業を支援するシステムを開発している [19]。彼らのシステムは web ベースで動作する動画編集シーケンサであり、編集画面にアノテーションを付加する機能によって遠隔で非同期な協調作業を支援している。また、Farooq らは遠隔に居ながら科学的でクリエイティブな議論をするためのシステム BRIDGE を開発している [20]。BRIDGE は Basic Resources for Integrated Distributed Groupe Environments を意味し、タイムライン・チャット・図示などの機能を持つ仮想的な作業空間を共有し、創造的な議論を支援している。議論の対象のコンテキストに則したアノテーションを付加する研究として、Cadiz らの研究 [21] では web ページ編集者同士のコミュニケーションや web ページの質の向上を目的とし、web ページそのものに対してアノテーションを行うシステムを開発している。また、数百人規模の長期的な実験によってシステムの実用性を客観的に明示している。また Ellis と Groth の研究 [22] では、映像データに対してテキストや画像・音声などといったデータをアノテーションとして付加できるシステムを開発し、創作活動を通して非同期なコミュニケーションを支援している。Lang らはアバターによる視覚と聴覚を介した協調的な web ベースの初心者支援のシステムを開発している [23]。Goldman らは Collabode と呼ばれるコーディングのための web ベースの IDE を開発している [24]。Collabode は数多くの種類の言語において有用性があり、また HTML5 の可能性を拡げたと言える。Marion らは WebGL と WebSocket という 2 つの新しい技術によって、3D データセットに対する高速で簡易な共同可視化作業の手段を提供している [25]。

これらの研究に対し本研究は、演奏に関する議論の支援を行っている。web を介してアノテーション（コメント）の対象となるのは楽譜であり、演奏練習の音声ログである。また、本研究で作成したシステムは Web RTC, Web Audio, Web Workers などの HTML5 の新しい技術を利用している。

第3章　演奏練習の音声ログの活用手法

本研究では演奏練習の音声ログの活用手法として、音声ログの「楽譜への対応付け」・「可視化」・「議論での活用」を提案する。これら 3 つの手法はそれぞれが完全に独立したものではなく、楽譜への対応付けに基づいて可視化し、可視化によって議論への利用を行いやすくしている。なお、本章では各手法のモデルやアルゴリズムを説明するに留め、実際のシステム上の実装は第 5 章にて説明するものとする。

演奏者は演奏練習時に演奏練習の録音をするとともに、システムに簡易な操作を行うことで演奏開始・停止の時刻データ（クリップデータ）を記録する。クリップデータは、「これから○○小節目から演奏する」という情報と「演奏を停止した」という情報を示すもので、システムに表示された楽譜上のタグを選択することで、何小節目からいつ始めるのかまたはいつ演奏を終了したのかを知る手がかりになる。また、システムは予め演奏練習を行う楽曲について楽譜と楽曲構造データを保持しておくものとする。楽譜は画像としての通常の楽譜で、楽曲構造データは小節数・拍子・テンポのまとまりのデータであり、音声ログの楽譜への対応付けに利用する。

3.1　楽譜への対応付け

1 つ目の手法として、演奏練習の音声ログを楽譜へと対応付ける。楽譜への対応付けの概念図を図 3.1 に示す。

対応付けには「楽曲構造データ」、「クリップデータ（演奏開始・停止の時刻データ）」、「音声ログの音量データ」を用いる。楽曲構造データには演奏位置推定に必要なテンポ・拍子・小節数のデータを利用する。これは Lilypond[1]や MusicXML[2]などの音楽記述言語を用いて記述された楽曲構造モデルから取得することを想定している。クリップデータは演奏開始・停止の時刻データであり、演奏練習中に繰り返し行われる演奏それぞれの始まりの小節番号・始まりの時刻・終わりの時刻である。これは演奏練習中にユーザからの簡易な操作によってリアルタイムに取得することを想定している。音声ログの音量データは、録音した音声データから単位時間毎の最大値を求めたものである。これらのデータを組み合わせて音声ログの演奏位置の推定を行い、楽譜と対応付ける。

楽譜への対応付けの流れを図 3.2 に示す。まずクリップデータのうち演奏開始のデータそれぞれから、音量が大きく増加する地点を探す。これはユーザがシステムに演奏開始を伝える操

[1]GNU Lilypond：GNU GPL ライセンスのもとにフリーで公開されている、クロスプラットフォームの楽譜作成ソフトウェア。

[2]MusicXML：XML 形式の楽譜表記のためのオープンなファイルフォーマット。

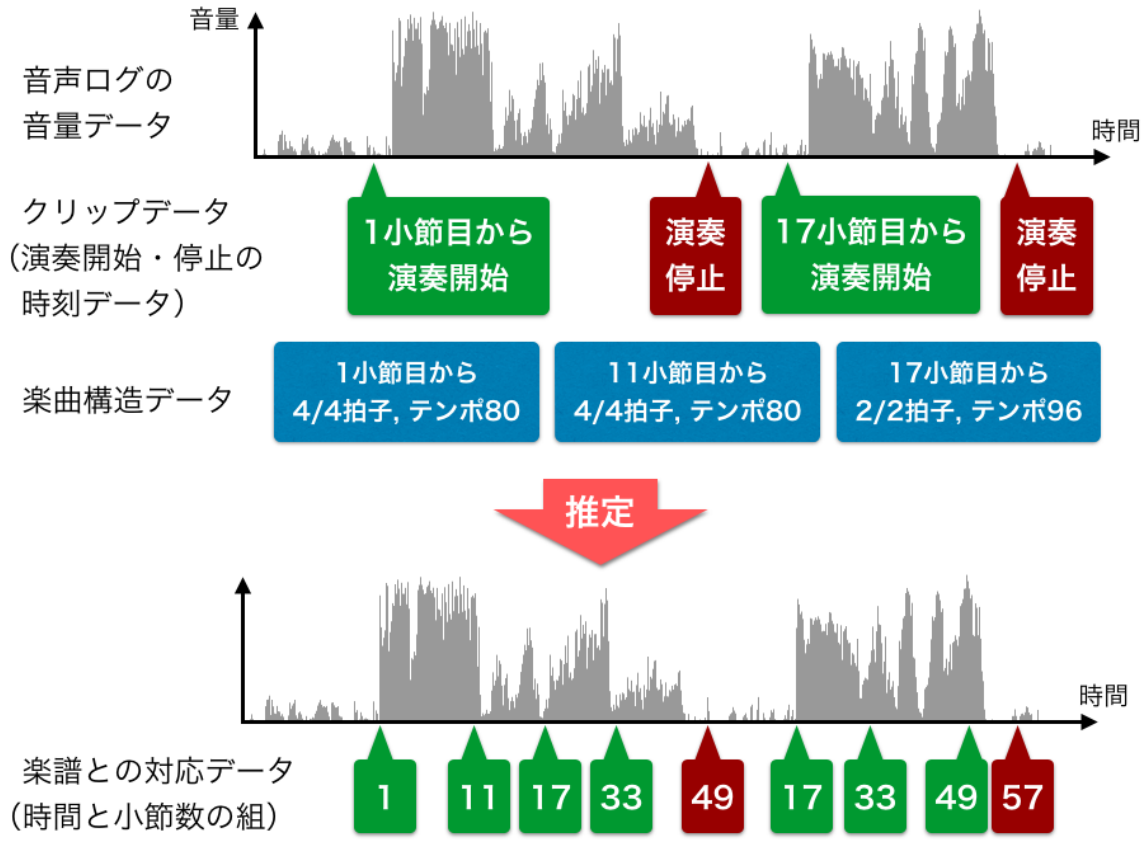

図 3.1: 音声ログの楽譜への対応付けの概念図

作を行ってから、実際に演奏を開始するまでのずれを修正するために行う。この探索は楽曲中で静かなフレーズから演奏を開始する場合には不確実なものとなると考えられるため、時間のしきい値を設けて見つからない場合には保留する。演奏練習において演奏は繰り返し行われるため、他の演奏開始データから実際の演奏開始までの時間の平均を求め、保留した探索に適用する。次に、こうして修正された演奏開始のデータそれぞれから、その次に演奏停止のデータが現れるまでの小節番号を補完する。楽曲構造データから 1 拍あたりの時間（60[秒]をテンポ [毎分] で除算したもの）と 1 小節あたりの拍数を乗算することで 1 小節あたりにかかる時間を求め、次の小説までの時間を求める。これの計算を繰り返し、演奏停止のデータまでを小節番号を埋める。

以上の流れによって音声ログの楽譜への対応付けを行うが、これは必ずしも正確な対応ができるものではない。しかし、あくまで音声ログは最終的に議論のために活用されることを目的としているため、完全に正確な推定を追求する必要はないと考えている。実際にどの程度の正確性であるかは第 6 章の評価実験で詳しく述べる。

本手法の限界について敢えて議論するならば、楽曲によってはテンポの指示が正確な数値

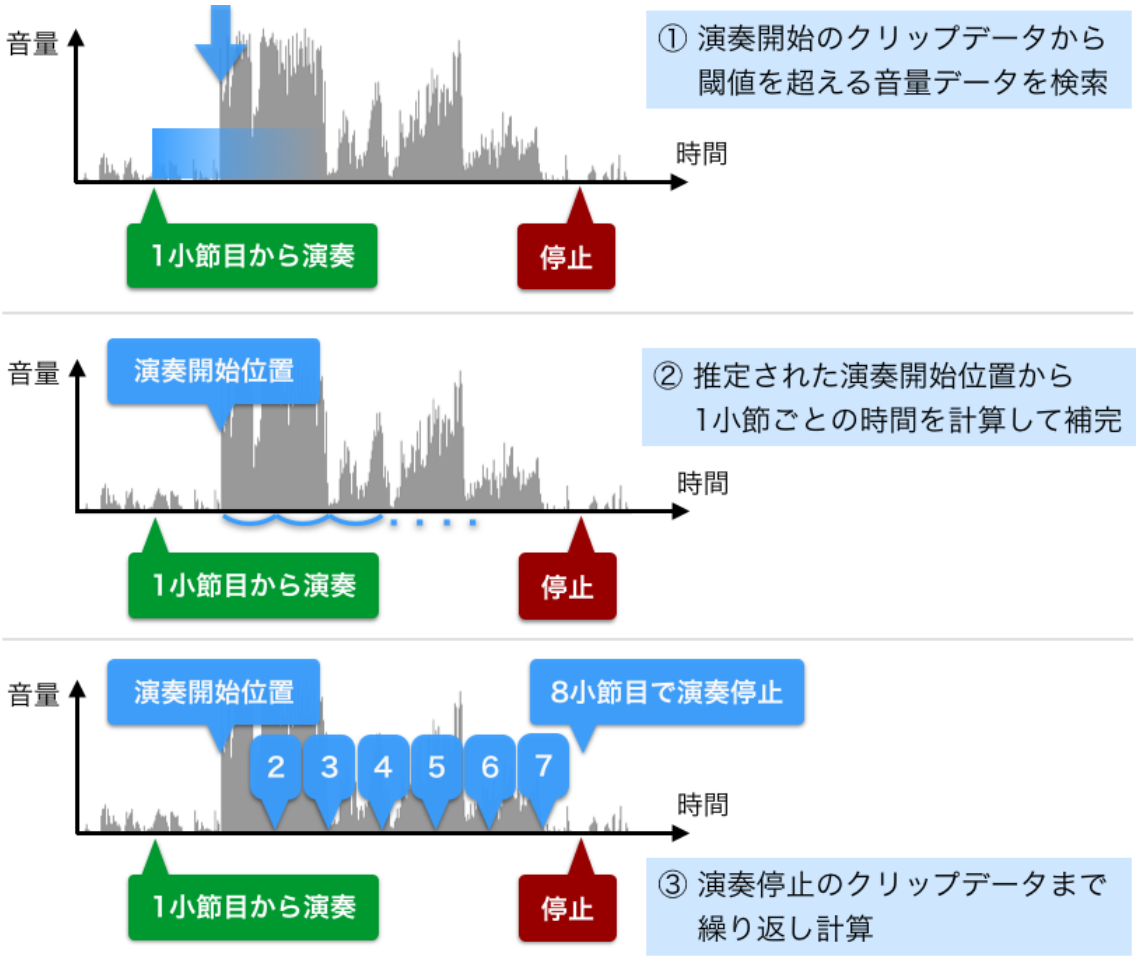

図 3.2: 音声ログの楽譜への対応付けのアルゴリズム

でないもの（Allegro や Andante のような指示など）や、フェルマータ[3]やテンポ・ルバート[4]のあるものには完全には対応することができない。また、テンポの正確な指示があるものであっても演奏者の力量によって、あるいは演奏者自らの解釈によって、本来とは異なるテンポで演奏されることは極めて多い。理想を考えるならば、音声データをリアルタイムで周波数解析しながら楽曲構造データと照らし合わせ、完全自動で楽譜への対応付けを行えることが望ましい。しかし第 2 章で述べたように、多声の演奏と会話が（場合によっては雑音も）連続して入り混じる長時間の音声ログの対応付けを行うことは現在の技術では困難であるため、本研究では上記のようなユーザの簡易な操作を必要とするものとした。他にユーザの簡易な操作によってテンポを取得する方法としては、演奏練習で用いるメトロノームからテンポを取得する方法や、外付けのペダル型入力機器を用いて演奏者に足でテンポを入力してもらう方

[3]フェルマータ：音楽記号の 1 つ。この記号が添えられた音符や休符は楽譜に記された本来の時間よりも長く演奏される。
[4]テンポ・ルバート：演奏指示の 1 つ。一般的にテンポを柔軟に変化させて演奏される。

法なども考えられる。

3.2　可視化

2 つ目の手法として、上述の方法によって導かれた「楽譜との対応データ」を可視化する。この手法では、前述の楽曲構造データから曲の終わりの小節番号を求めた「総小節数」と対応付けで用いたものと同様の「音声ログの音量データ」を用いる。

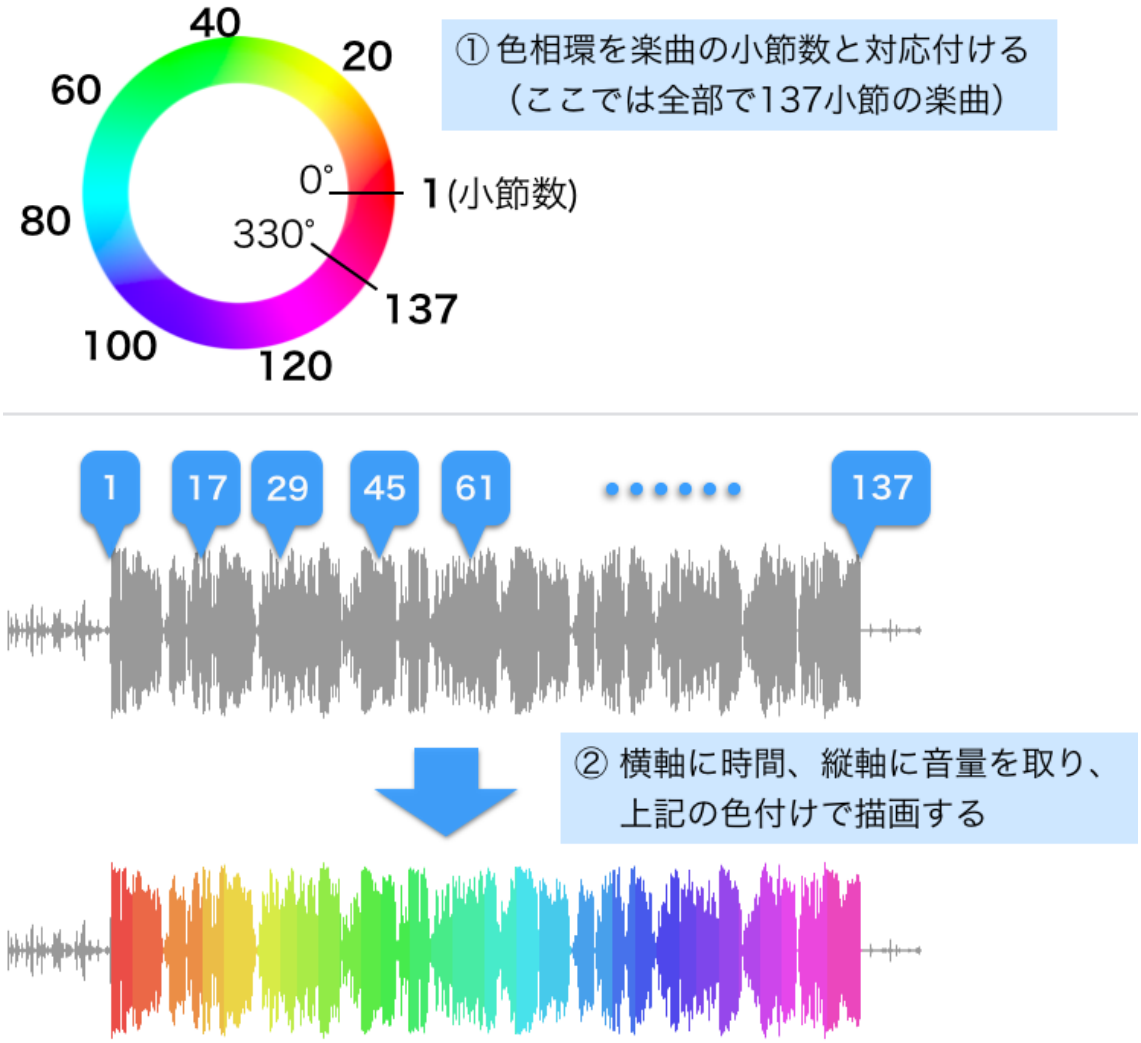

図 3.3: 音声ログの可視化のアルゴリズム

音声ログの可視化の流れを図 3.3 に示す。音声ログの時間の流れに沿って左から右へと描画していく。まず楽譜との対応データから、音声ログの何秒目が楽譜の何小節目にあたるかが割り出される。楽譜（楽曲）の先頭から末尾を色相の変化に対応付け、現在の小節数を総小節

数で割ることで音声ログの何秒目を何色で描画するかが決まる。この時、小節数に応じて連続的に変化させるのではなく、楽曲構造の練習番号の小節数ごとに離散的に変化させる。また、演奏していない箇所は彩度を与えず灰色で描画する。縦方向の幅は音声ログの音量データに伴い、音量が大きい部分は幅広く、音量が小さい部分は細く描画される。

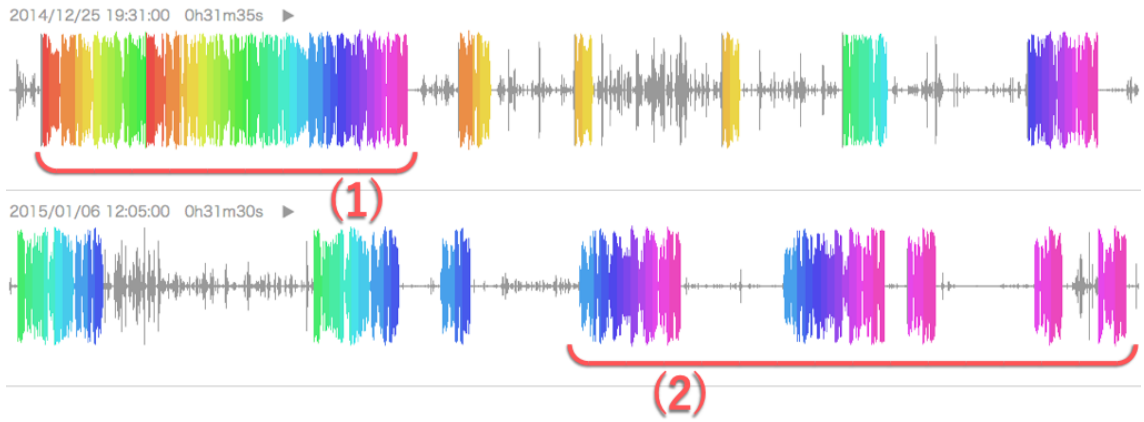

図 3.4: 音声ログの可視化例

以上のような音声ログの楽譜への対応付けと可視化によって、既存の問題点である「音声ログから聴きたい箇所を探し出す手間が大きい」ことを解決し、過去の演奏練習の内容の一覧性を高める。[26] を参考に、楽譜上の小節数に対応する色を音声ログの描画に用いる。具体的な可視化例を図 3.4 に示す。まず図 3.4 の全体として、色付けされた部分では演奏しており灰色の部分は演奏以外の会話などであることがわかる。図 3.4 の (1) では色が赤から始まり色相を回るように変化していき、また 1 度赤に戻ってから色相を 1 周して紫まで到達していることから、途中に繰り返しのある楽曲を最初から最後まで通しで演奏練習したことがわかる。図 3.4 の (2) では、前半は色が青から紫に及び、後半は短い紫が続いていることから、楽曲の中盤から末尾にかけて 2 回演奏した後、末尾の部分を重点的に練習したことがわかる。このように音声ログが可視化されていることで、従来音声ファイルとして共有されていた音声ログをより効率よく活用することができる。

本研究では上記のように色相を赤から紫まで使ったものに加えて、「温度」と「明度」という 2 つの色付け方法を提案する（図 3.5）。「温度」は温度表現に使われる色を用いたもので、基本的には色相と同様であるが色相の範囲をを赤から青までとし、暖色から寒色への変化としてユーザが理解しやすくなるようにした。また、「明度」は明るさを変化させたもので、明るい赤から強い赤を介して黒まで変化するものとした。6.3 の評価実験では、演奏者を対象にこれら 3 つの色付け方法について評価実験を行っている。

予備調査として、可視化を専門に学ぶ大学院生 2 名に上記の色付け方法について意見を求めたところ、Lab 色空間を使うとさらに良いだろうというアドバイスを貰った。Lab 色空間とは、マンセル色相環を「知覚的に均等」になるように変換したものである。知覚的に均等とは、色の値が同じだけ変化したとき、人間がそれを見たときに感じられる変化も等しいこと

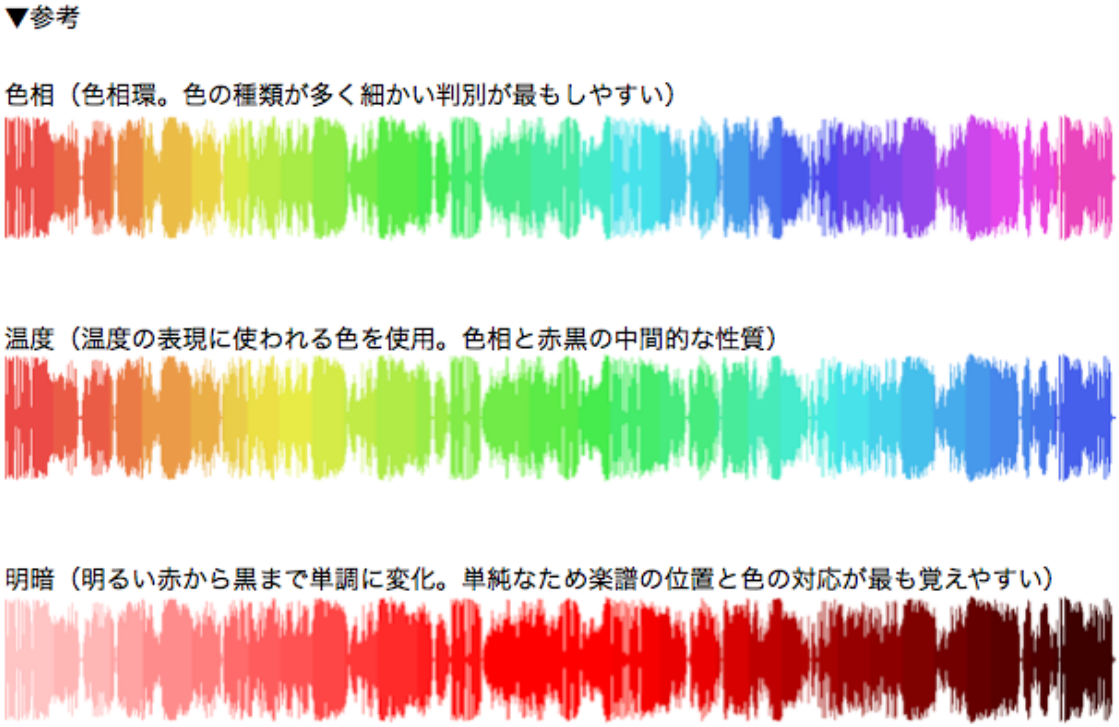

図 3.5: 色相・温度・明暗による色付け方法

を意味する。例えば通常のマンセル色相環では、人間が見ると緑の範囲が他の色に比べて広く変換がゆっくりであるように感じられるが、Lab 色空間を使うことでこの問題が解決される。しかし今回の実装では時間の都合により、通常のマンセル色相環での実装とした。将来的には是非とも Lab 色空間での実装を行いたい。

3.3 議論での活用

3 つ目の手法として、音声ログを演奏に関する議論において活用しやすいようにする。上述の手法によって可視化した音声ログを、先行研究 [1] で提案した楽譜上の特定箇所にコメントすることで演奏に関する遠隔非同期的な議論を行う手法で利用可能にする。

議論での活用の流れを図 3.6 に示す。まず、上述の手法によって可視化された音声ログの中から議論に取り上げたい部分を範囲選択して切り出す。その後、楽譜上の特定箇所にコメントを投稿する際に、その新しいコメントに切り出した音声ログを添付する。添付された音声ログはその場で自由に再生・停止することができる。

以上の流れによって音声ログの一部分に言及しながら議論をすすめることができる。想定される使用方法としては、まずシステムのユーザは音声ログを一人で聴取している中で、演奏者ごとに解釈が異なっている箇所があることに気づく。具体的にはそれは音の長さの違い

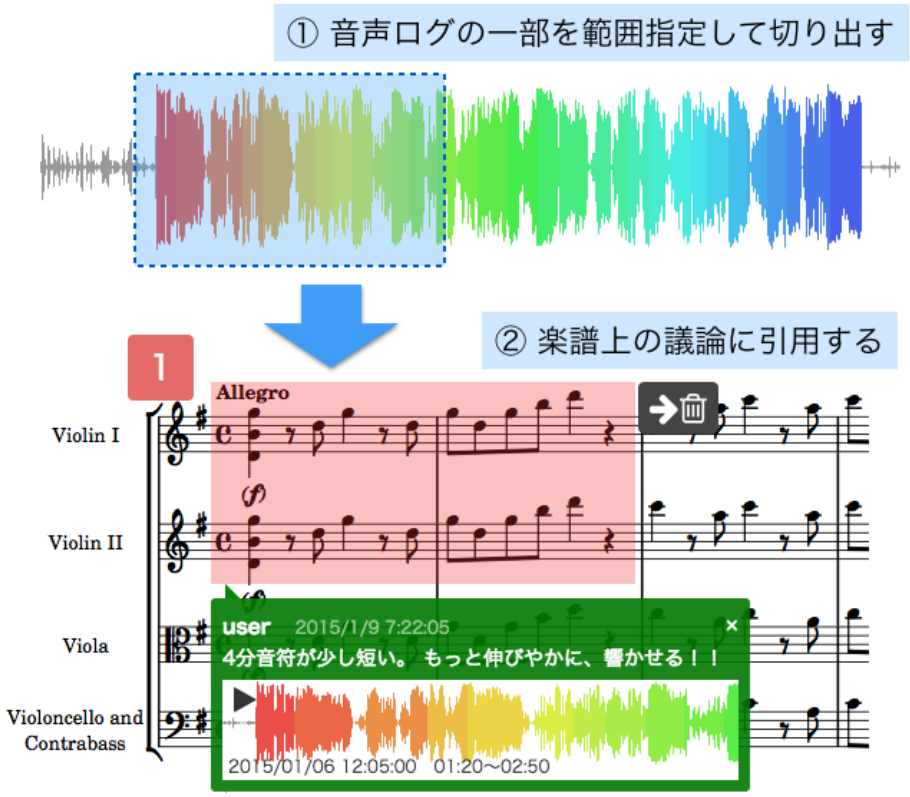

図 3.6: 音声ログの議論での活用

や音の感覚的な「固さ」の違いであり、演奏者間で解釈を統一するべきであるとそのユーザは考える。そこでユーザは音声ログの中からその解釈の違いがわかる部分を切り出し、対応する楽譜上の特定箇所にコメントを切り出した音声ログ付きで投稿する。その後、また別のユーザはそのコメントの文章を読み音声ログを聴くことで、自分の意見を投稿したり次回の練習の課題としたりすることができる。

第4章　Co-Musicatorシステム

本研究で提案する演奏練習の音声ログの活用手法を利用し、演奏者のための遠隔非同期的なコミュニケーションシステム Co-Musicator を作成した。本システムは、演奏練習における音声ログの録音・web を介しての共有・聴取や議論への活用を支援する。ユーザは演奏練習中と演奏練習以外の 2 つの場面からシステムを利用することで、演奏練習の効率化を図ることができる。

4.1　システム概要

システムの概念図を図 4.1 に示す。

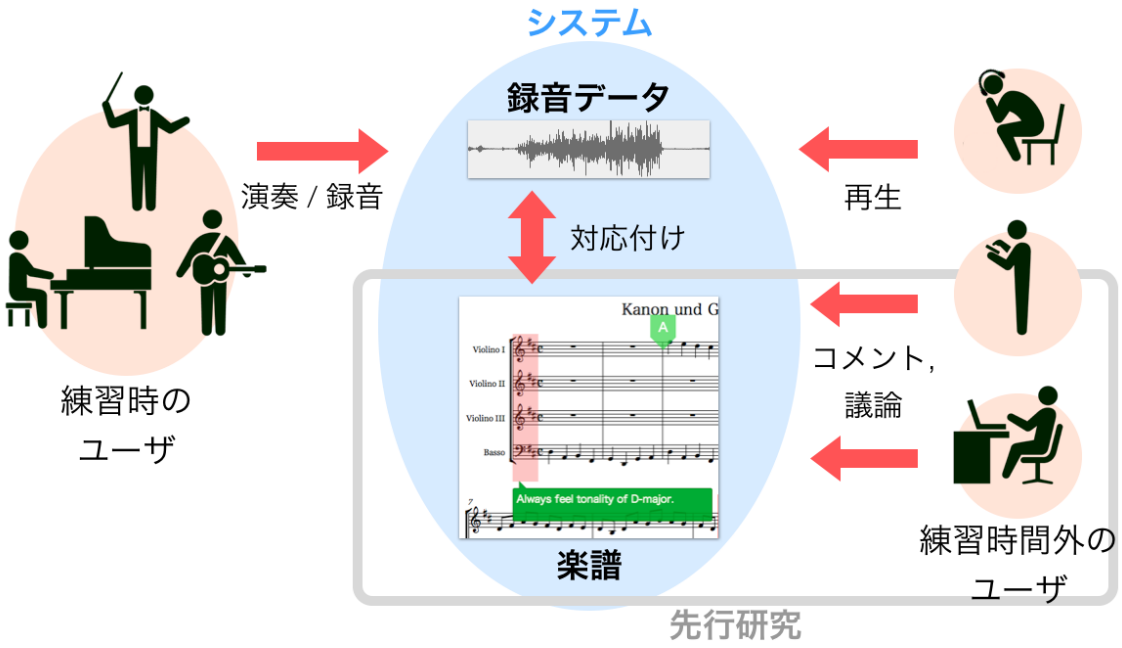

図 4.1: システム概念図

先行研究において開発したシステムでは、演奏練習以外の空いた時間を利用して、楽譜上の特定の箇所にコメントを投稿することで議論を行うことができた。今回作成するシステムでは、演奏練習中と演奏練習以外の 2 つの利用場面がある。演奏練習中においては、ユーザ

の簡易な操作と録音によりシステムが自動で音声ログを楽譜に対応付ける。練習以外の時間においては、ユーザは対応付けによって可視化された音声ログを聴取したり、音声ログを用いた議論を行ったりすることができる。また、練習時間外に行った議論の内容を練習時間中に確認することももちろん可能である。

システムの利用手段として、演奏練習中は少なくとも進行役となる 1 名がタブレット端末あるいはラップトップ PC を介してシステムにアクセスすれば良い。演奏練習以外は演奏者各自が空いた時間を利用して web ブラウジング可能な機器からシステムにアクセスする。システムは web アプリケーションとして動作するため、2 つの場面でどちらも全く同じページにアクセスすることで、練習時間と練習時間外をシームレスに連携することができる。

4.2 画面構成

本システムの画面構成を図 4.2, 図 4.3 に示す。

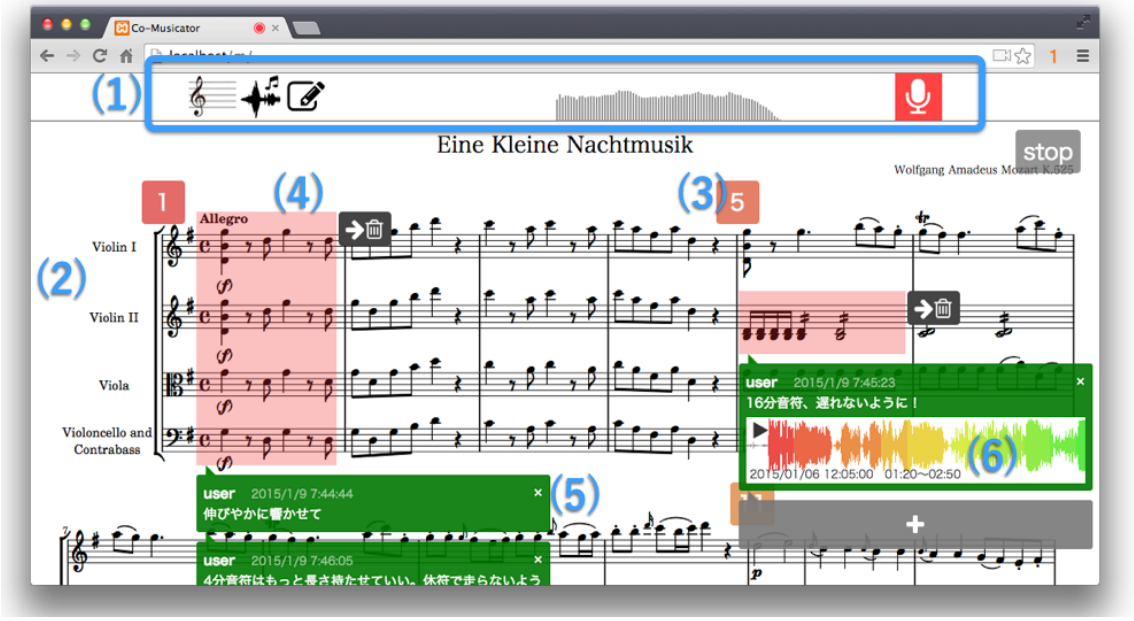

図 4.2: 画面構成：楽譜ビュー

本システムは楽譜ビューと音声ログビューの 2 つの画面を用途に応じて切り替えながら利用する。演奏楽曲のフルスコアの上でコメントのやりとりをすることで議論を進めるための楽譜ビューと、可視化された音声ログが一覧される音声ログビューである。画面上部のナビゲーションメニューから 2 つのビューを切り替えたり、特定の機能を使ったりする。

図 4.3: 画面構成：音声ログビュー

4.2.1 ナビゲーションメニュー

図 4.2, 図 4.3 の (1) がナビゲーションメニューである。ナビゲーションメニューは楽譜ビューと音声ログビュー共通で、常に画面上部に表示されている。

詳細な構成を図 4.4 に示す。(1) 楽譜ビューボタンと (2) 音声ログビューボタンは、クリックまたはタップすることでそれぞれ楽譜ビュー・音声ログビューへと切り替わる。(3) 新規コメントエリアボタンは楽譜ビューのときにのみ表示され、クリックまたはタップすることで新規コメントエリア作成モードへと切り替わり、楽譜上をドラッグすることで新規コメントエリアを作成する。(4) 音声ログ切り取りボタンは音声ログビューのときにのみ表示され、クリックまたはタップすることで音声ログ切り取りモードへと切り替わり、音声ログ上をクリックまたはタップすることでコメントに含めるための音声ログを切り取る。(5) 音声スペクトルにはマイクからのリアルタイムな入力を取得し、周波数領域として表示する。ユーザはここを見ることで音声が取得できているかを知ることができる。(6) 録音ボタンは演奏練習の開始時と終了時に録音を開始・終了するためのボタンで、クリックまたはタップすることで録音の on/off が切り替わる。録音中は図 4.4 下のようにナビゲーションメニュー全体が赤くなり、ユーザは現在システムが録音中かを判別することができる。

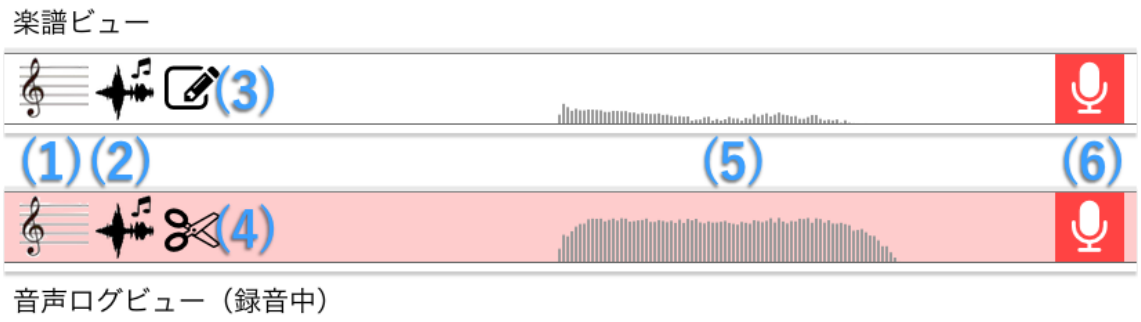

図 4.4: ナビゲーションメニュー

4.2.2 楽譜ビュー

図 4.2 の (2) が楽譜ビューである。楽譜ビューでは演奏楽曲のフルスコア（この図では Mozart 作曲の Eine Kleine Nachtmusic）が縦方向に連なり、その上に重畳されるようにして様々な情報が表示される。(3) は練習番号のタグであり、第 3 章で述べた可視化の手法と同様に楽曲の総小節数で除算することで色が決定されている。録音中にクリックまたはタップすることで演奏開始時刻・小節を記録したクリップデータが追加される。クリップデータは録音開始時刻からの時間とクリックまたはタップされた練習番号（小節数）から構成される。図 4.2 右上の stop ボタンも、クリックまたはタップすることで演奏停止時刻を記録したクリップデータが追加される。(4) の赤い半透明の矩形はコメントエリアであり、コメントを投稿するための楽譜上の範囲を意味している。クリックまたはタップすることで、そのコメントエリアに属しているコメントとコメントエリアの削除ボタンの表示/非表示が切り替わる。削除ボタンはクリックまたはタップすることでそのコメントエリアとそのコメントエリアに属する全てのコメントを削除することができる。(5) の緑色の吹き出し状の矩形はコメントであり、コメントエリアに属することで楽譜上の位置を特定したメッセージを交わすことができる。コメントはユーザ名, そのコメントが投稿された日時, 削除ボタン, テキストメッセージ, 音声ログから構成されている。(6) の色鮮やかな絵はコメントに含まれた音声ログであり、クリックまたはタップすることでその場で音声ログを再生・停止することができる。コメントに含まれた音声ログは可視化された波形の上に、再生・停止ボタン, その音声ログが録音開始された日時, 音声ログ全体の中のどの部分かを示す分と秒が表示されている。

4.2.3 音声ログビュー

図 4.3 のナビゲーションメニューを除いた部分が音声ログビューである。音声ログビューでは操作用のタグの下に音声ログが縦方向に連なる。(7) は表示切り替え用のタグであり、音声ログの拡大・縮小に使用する灰色の虫眼鏡マークのものと音声ログの頭出しに使用するカラフルな数字のものがある。システム起動時、音声ログビューの音声ログは 1 ピクセルあたり 2 秒分の音声ログのデータ密度であるが、虫眼鏡のタグをクリックまたはタップすることで 1

ピクセルあたり 8 秒から 0.25 秒までの範囲で表示を切り替えることができる。拡大するほどに細かい時間の指定ができるので詳細な聴取などに向いており、縮小するほどに全体を俯瞰できるので練習内容の確認などに向いている。

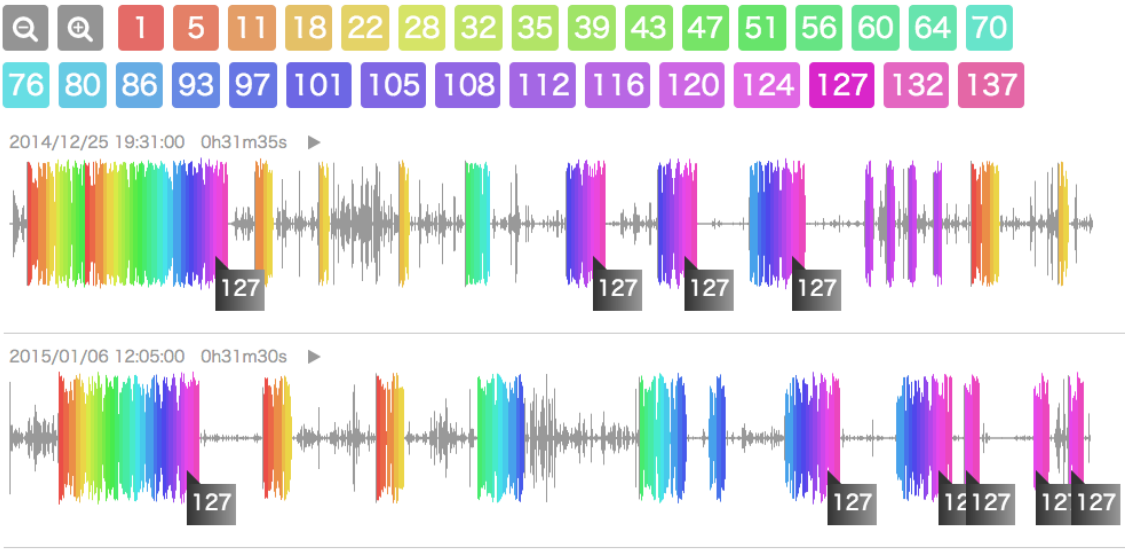

図 4.5: 音声ログ上に表示されたタグ

また、カラフルに色付けされた数字のタグは楽譜ビューの練習番号タグ（図 4.2(3)）と同様の数値・色であり、クリックまたはタップすることで音声ログ上の対応する箇所にタグの表示がなされる（図 4.5）。このタグはクリックすることでそこから再生することができ、またドラッグすることで対応付けがずれていた場合に修正することができる。(8) は音声ログであり、第 3 章で述べた可視化の手法によって時間に対する音量の変化と楽譜との対応が表示される。個々の音声ログはその音声ログが録音を開始した日時と再生・停止ボタンが付いている。音声ログ上をマウスホバーすることで音声ログの時間が追従表示され、クリックまたはタップすることでその時間から再生することができる。また、上述の音声ログの拡大操作によって画面から入りきらなくなった場合は、個別の音声ログが横スクロールできるようになる。

4.3 システム利用方法

本システムは演奏練習時と演奏練習外の 2 つの場面を想定して作成されている。それぞれについてユーザがどのようにシステムを操作し、それによってどのような問題が解決されるのかを説明する。

4.3.1 演奏練習時

演奏練習時、ユーザはタブレット端末やラップトップ PC などから web 上に公開された本システムにアクセスする。演奏練習という場面での操作性の観点から、大型のタブレット端末を台や机に置いて利用することが望ましい。端末内蔵のマイクでは音質に満足できない場合や、そもそも端末にマイクが内蔵されていない場合には外付けのマイクを別途準備し、接続しておく必要がある。システムは起動時に楽譜・コメントエリア・コメント・音声ログを読み込み、画面に表示する。すでに議論を行った場合にはこの時点で議論の内容を確認することができる。また、すでに演奏練習の音声ログを取った場合にはその音声を聴取することができる。

演奏練習の開始時、ユーザはナビゲーションメニューの録音ボタンを 1 度押して録音を開始する。そして音を出す演奏を始める前に、かならずシステム上で開始位置のタグをタップする。また、演奏が止まるたびに stop タグをタップする。この操作によってクリップデータが作成され、録音の終了後に音声ログの楽譜への対応付けを行うことができる。最も、最終的な推定されたクリップデータはあとから編集が可能なため、万一タグのタップを忘れることがあっても問題はない。演奏練習の終了時、ユーザは再度録音ボタンを押して録音を終了する。これにより MP3 ファイルの生成と音声ログの楽譜への対応付けが実行され、即座に音声ログビューに可視化された音声ログが追加される。なお、演奏練習は数十分から 1 時間程度を想定しており、練習の合間に休憩などが入る場合に別々に録音するか繋げて録音するかはユーザの好みによるところである。

コメントエリアの作成やコメントの投稿は、基本的には演奏練習外に行われることを想定しているが、演奏練習時であっても当然利用することができる。例えば、練習中に議論の余地があると判断された箇所にコメントエリアだけ作成しておいて、あとで演奏練習外にコメントしやすくするといった使い方ができる。または、演奏練習時の話し合いで決定された事項を忘れないように、すぐにコメントとして投稿しておくことも有効であろう。演奏練習中であっても録音を一旦停止して、その場で音声ログを聴取することも可能である。

4.3.2 演奏練習外

演奏練習外の時間に、ユーザはそれぞれが相手いる時間を利用してシステムにアクセスする。システムはユーザの使い慣れた PC 環境、すなわちマウスとキーボードによる使用を想定している。スマートフォンやタブレット端末などのタッチデバイスでもシステムにアクセスすることはできるが、新規コメントエリアの作成にドラッグ操作が必要なため、これらの端末からでは新規コメントエリアの作成が利用できない。また、ブラウザのバージョンによってはレイアウトが崩れたり音声を再生できなかったりする可能性がある。コメントの閲覧や既にあるコメントエリアへの投稿は可能である。

システムを起動し、ユーザはコメントのチェックや音声ログの聴取を行う。投稿されたコメントや演奏練習時の記憶から、ユーザは課題があると思う箇所にコメントエリアを作成したりコメントを投稿したりする。また、他人のコメントへの返信を行うことで議論を深める。音

声ログの中で課題を発見した場合は、音声ログの切り取り操作を行い新たに投稿するコメントに含めることで、より問題の共有がしやすくなる。過去の音声ログを俯瞰することで、思いの外練習回数が少ない箇所があった場合にはコメントでそれを提起し、次回以降の演奏練習で取り上げる方針を示すこともできる。

録音機能は基本的に演奏練習時に使用される想定だが、演奏練習外にも使うことができる。対面環境での議論では、実際に楽器で演奏してみせたり声に出して歌ったりすることで音楽の表現を伝えることがよくある。そのため遠隔非同期的なテキストベースの議論ではその点に限界があったが、本システムの録音機能はそこにひとつの解を与える。ユーザ個人が演奏練習外にシステムを利用している際に、1 人での演奏や歌を音声ログとして残し、コメントに含めることで表現の選択肢を拡げることができる。

第5章　実装

5.1　システム構成

システムの構成を図 5.1 に示す。

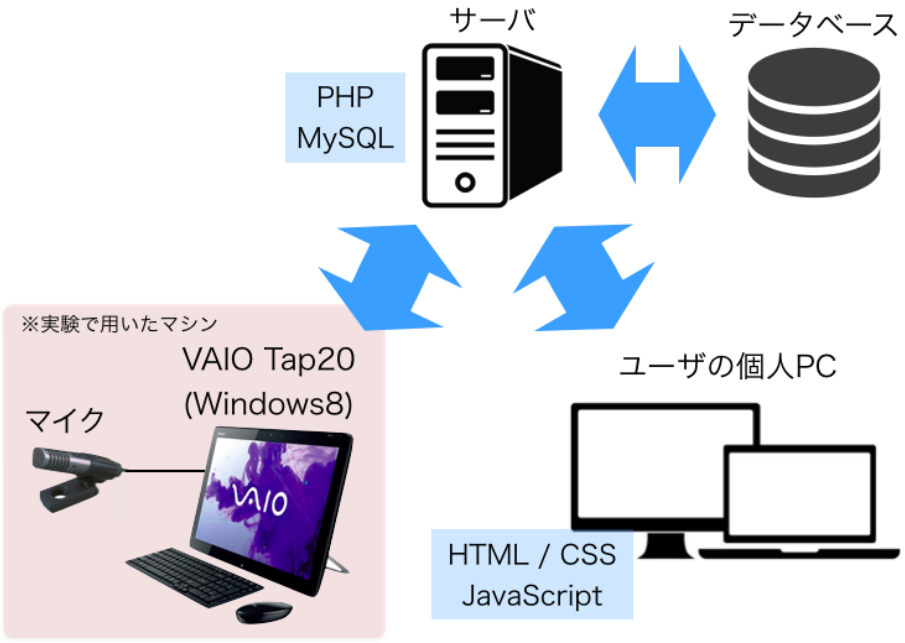

図 5.1: システム構成図

システムはクライアントサイドとサーバサイドに分かれており、クライアントサイドは HTML/CSS, JavaScript で、サーバサイドは PHP, MySQL を用いて実装されている。ユーザの持つ web ブラウジング可能な機器からアクセスされ、演奏練習時にはデバイス内蔵あるいは外付けのマイクロフォンから音声を取得する。録音が完了すると音声のバッファは WAV 形式を介して MP3 ファイルへと変換され、サーバ上に保存される。ユーザがシステムにアクセスすると、サーバからコメント・音声ログに関するデータが読み込まれて表示される。新たにコメントエリアを作成したりコメントを投稿したりすると、Ajax 通信によりこれらのデータはリアルタイムでサーバに送信される。音声ログの描画やマイクからの音声の取得、MP3 へのエンコードのための非同期処理など、いわゆる HTML5 と呼ばれる新しい技術を用いて実装されている。そのため本システムは 2015 年 1 月現在、一部のモダンブラウザ（PC 版 Google

Chrome, Firefox, Opera など）以外ではレイアウトが崩れたり一部の機能が利用できなかったりすることがある。しかし HTML5 は 2014 年 10 月 28 日に W3C によって正式に勧告され、今後の web の標準となっていくことが予想されるため、本システムを利用可能な環境の制限は無くなっていくと考えられる。

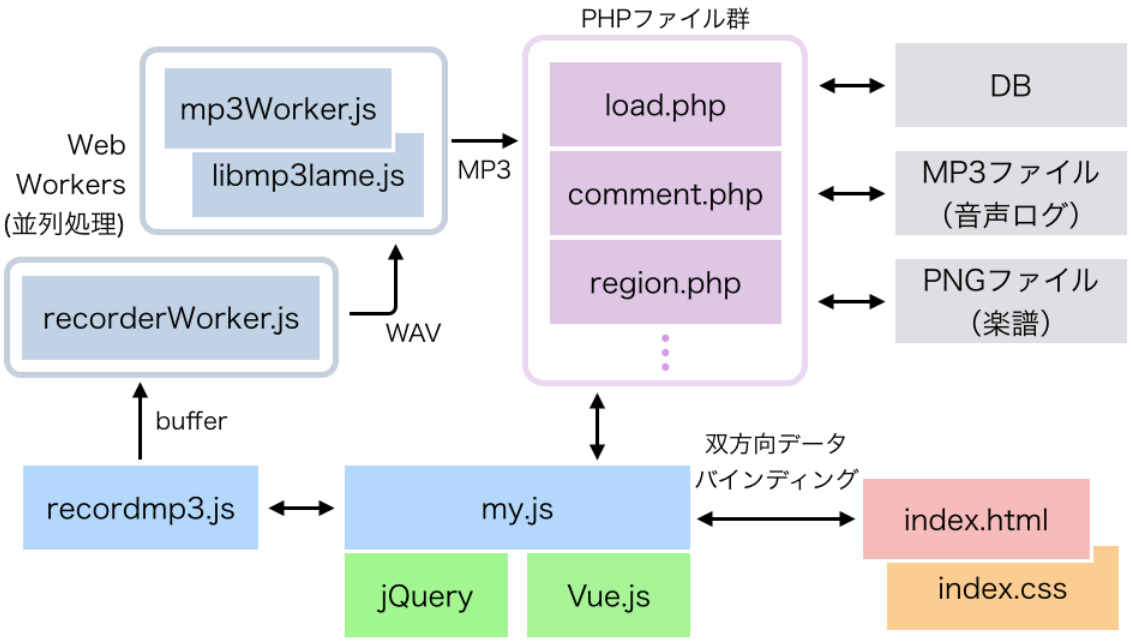

図 5.2: ソフトウェア構成図

次に、システムのソフトウェア構成の詳細を図 5.2 に示す。まず、ライブラリとして jQuery, Vue.js, Recordmp3js を使用した。jQuery[1] は DOM 操作のインタフェースを提供するライブラリで、JavaScript のライブラリとして最も有名であると行っても過言ではない。Vue.js[2] はインタラクティブなウェブインタフェースを提供する軽量な MVVM モデル（Model View ViewModel）型のフレームワークで、完結な記法で強力なデータバインディングが可能である。Recordmp3js[3] はブラウザ上で録音を行い、MP3 ファイルを生成するライブラリである。Recordmp3js はインタフェースを提供する recordmp3.js, JavaScript で非同期処理を実現するための Web Worker とやりとりを行う mp3Worker.js, Web Worker 本体である recorderWorker.js, MP3 への変換を行う libmp3lame.js から構成されており、このうち recordmp3.js, mp3Worker.js, recorderWorker.js は筆者によってビットレートの調整などの修正を施した。Vue.js の Vue インスタンスを my.js 内で生成し、index.html ではタグに v-repeat 属性を与えることで my.js のデータを表示する。index.html 内の要素のマウスイベントは v-on 属性によって my.js のメソッドが呼び出される。今回、楽曲構造データは JavaScript オブジェクトとして使用できるものを予め用意した。ま

[1] jQuery：http://jquery.com/
[2] Vue.js：http://vuejs.org/
[3] Recordmp3js：https://github.com/nusofthq/Recordmp3js

た、データベースの読み書きのための PHP ファイルとして、システム開始時に読み込みを行う load.php, コメントエリア及びコメントの生成と削除のためのもの, 音声ログの音量データ生成のためのもの, MP3 ファイルをサーバ上に保存するためのものがある。これらは my.js から jQuery の ajax メソッドによって JSON 形式でデータを受け取り、SQL 文を発行する。

次に、システムのデータ構造について説明する。まず音声ログのデータには、録音が開始された日付を表す文字列, 音声ファイルが何秒間であるかの数値, 音量データの数値の配列, 音声ログの楽譜との対応付けデータの配列, 音声ファイルのパスの文字列が記憶される。音声ログの楽譜との対応付けデータは、音量データの配列のインデックス, 演奏中か演奏停止かを表す真偽値, 楽譜上の小節番号の 3 つのデータからなるオブジェクトの配列である。コメントエリアのデータには、そのコメントエリアが属する楽譜のページ番号, その楽譜のページ内でのコメントエリアの左上端の x 座標と y 座標, コメントエリアの幅と高さが記憶される。コメントのデータには、そのコメントが属するコメントエリアの ID, コメントを投稿したユーザ名, コメントのテキスト内容, コメントに含まれる音声ログの ID, その音声ログの開始位置が記憶される。以上のデータにはユニークな番号である ID, そのデータが生成された時刻のタイムスタンプ情報を持つものとする。楽曲構造データには、楽曲のタイトルと作曲者, 楽曲全体の小節数, 練習番号データの配列から構成される。練習番号データは、小節番号, その小説番号が出現する楽譜上でのページ番号, そのページ内でのその小節の x 座標, システム上に表示される文字（小節番号の数値そのものあるいはアルファベット）, その小節からの拍子とテンポを保持する。その他に、ファイルとしてサーバ上に保持されるものとして、楽譜の PNG 画像データ, 音声ログの MP3 データがある。

5.2 音声の取得・録音

録音の実現には本システムの中でも特に新しい技術が多く使われている。getUserMedia/Stream API は、ウェブカメラやマイクロフォンにアクセスするための API で、本システムで用いられている HTML5 API の中で最も新しいものである。そのため 2015 年 1 月現在、本システムの利用には getUserMedia/Stream API をサポートしているブラウザ（PC 版 Google Chrome, Firefox, Opera など）が必要である。Web Audio API は、音声を処理・合成するための web アプリケーション向けのハイレベルな JavaScript API である。Source, Destination, GainNode などのインスタンスを生成・接続し、HTML5 の Audio 要素と組み合わせることによってブラウザ上で音声を再生することができる。Web Workers API は JavaScript を並列計算させるための API で、バックグラウンドのスレッドで処理を実行することができる。この API は本システムではバッファデータを WAV 形式にエンコードし、更に MP3 形式に変換するという非常に重い処理を並列処理させるために用いられている。Worker の宣言と生成を別々の JavaScript ファイルで行い、postMessage メソッドと onmessage ハンドラの定義により非同期な処理が可能となる。

録音の流れを図 5.3 に示す。システムの状態により録音中（演奏練習中）である場合に限り、Audio ノードの onaudioprocess イベントが発火し、Worker に inputBuffer を乗せたメッセージ

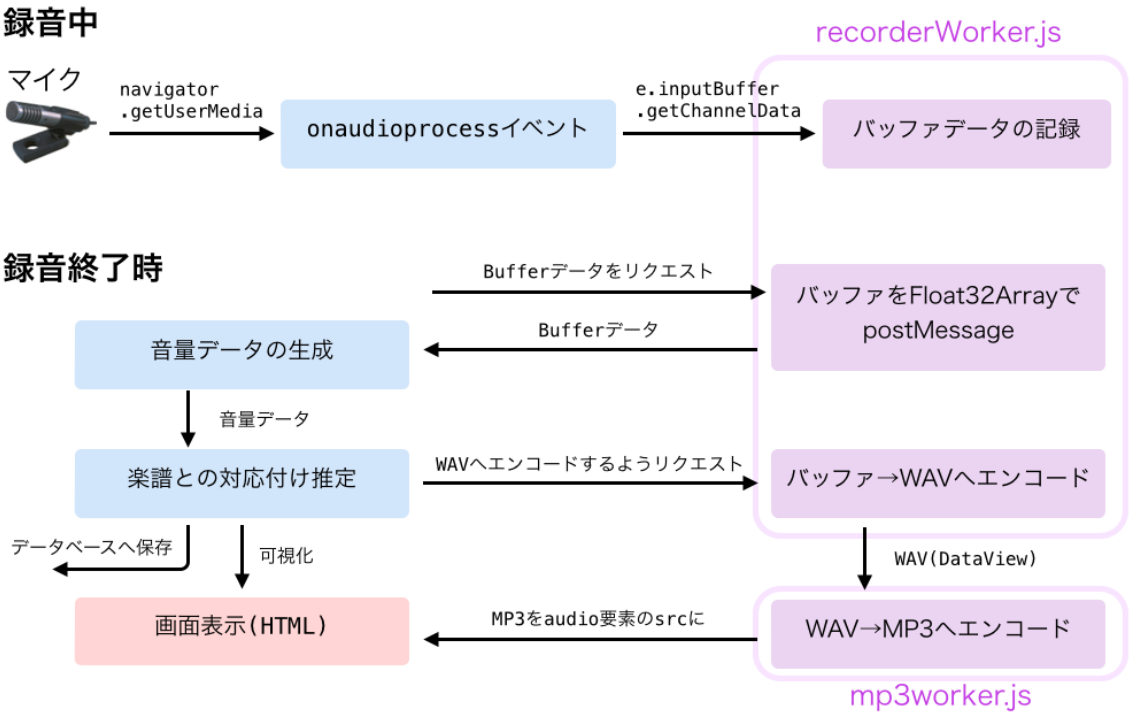

図 5.3: 録音の流れ

が送られる。Worker は受け取ったバッファデータを随時配列にプッシュしていく。録音が終了したら、Worker へバッファの配列を返すようメッセージを送り、Worker はそれを受けてバッファの配列を返す。Worker から受け取った配列はコールバック関数へ渡され、まず音量データの生成が行われる。音量データは 0.25 秒ごと（バッファは 44,100Hz で取得されているため、0.25 秒は 11025 個の数値になる）に、その単位時間内で最大と最小の数値の差を求め、音声ログビューに表示される canvas の高さに正規化されるよう 60 を乗算することで生成される（バッファデータは-1〜1 の範囲なので、最大値と最小値の差は最大で 2。音声ログビューに表示される canvas の高さは 120 なので、60 を乗算する）。0.25 秒ごととしたのは、データベースへの記録や JavaScript 内での処理を軽くするためや、音声ログの描画の際に 1 秒を 4pixel で表示すればユーザはストレス無く聴きたい箇所から聴けるであろうと考えたためである。次に、生成された音量データ, ユーザによるクリップデータ, 楽曲構造データから音声ログの演奏位置推定・対応付けが行われる。推定が完了次第、音声ログデータとして JavaScript 内の変数へのプッシュと Ajax 通信によるデータベースへの書込みが行われる。そして、HTML の audio 要素の音声ソースに最終的に出力された MP3 ファイルを設定する関数をコールバック関数として、Worker がバッファから WAV 形式へのエンコード、更に WAV 形式から MP3 形式へのエンコードを行う。最後に MP3 ファイルを Ajax 通信によって upload.php へ送り、サーバ上に MP3 ファイルが保存される。

5.3　演奏位置推定・楽譜への対応付け

演奏位置の推定には「楽曲構造データ」、「音声ログの音量データ」、「クリップデータ」を用いる。それぞれの JavaScript オブジェクト状態のデータの記述を図に示す。楽曲構造データは、楽曲のタイトルと作曲者, 楽曲全体の小節数, 練習番号データの配列から構成されている。音声ログの音量データは、1 以上の浮動小数点の配列となっている。これは前節で説明したとおり、0.25 秒ごとの音量を表している。クリップデータは、音声ログの音量データのインデックスに対応する数値, 楽譜上での小節番号, 演奏開始であれば true/演奏停止であれば false の真偽値から構成されている。

楽曲構造データ

```
var einekMusicStructure = {
    title: "Eine Kleine Nachtmusik",
    composer: "Wolfgang Amadeus Mozart K.525",
    totalBar: 137,
    letters: [
        { bar: 1, page: "01", pos: 110, letter: "1", timeSignature: 4, tempo: 132 },
        { bar: 5, page: "01", pos: 720, letter: "5", timeSignature: 4, tempo: 132 },
        { bar: 11, page: "02", pos: 735, letter: "11", timeSignature: 4, tempo: 132 },
        { bar: 18, page: "03", pos: 565, letter: "18", timeSignature: 4, tempo: 132 },
        { bar: 22, page: "04", pos: 215, letter: "22", timeSignature: 4, tempo: 132 },
        { bar: 28, page: "05", pos: 0, letter: "28", timeSignature: 4, tempo: 132 },
```

音量データ

```
var volumes0106 = [9.68,91.51,2.99,1.56,7.88,6.68,6.03,1.44,2.56,5.63,1,1,1.13,4.95,4.52,
```

クリップデータ（対応付け推定済み）

```
var trueClip1225 = [
    {index: 0, bar: -1, play: false },
    {index: 98, bar: 1, play: true },
    {index: 125, bar: 5, play: true },
    {index: 169, bar: 11, play: true },
    {index: 222, bar: 18, play: true },
    {index: 250, bar: 22, play: true },
    {index: 294, bar: 28, play: true },
    {index: 324, bar: 32, play: true },
    {index: 347, bar: 35, play: true },
    {index: 376, bar: 39, play: true },
    {index: 404, bar: 43, play: true },
    {index: 432, bar: 47, play: true },
    {index: 460, bar: 51, play: true },
    {index: 497, bar: 1, play: true },
    {index: 526, bar: 5, play: true },
    {index: 570, bar: 11, play: true },
```

図 5.4: 楽曲構造データ, 音声ログの音量データ, クリップデータの JavaScript 表現

推定を行う estimate メソッドは上記 3 つのデータを引数に取り、推定されたクリップデータを返り値に持つ。まずクリップデータのうち play が true であるものの index から、音量データを順方向に走査する。音量データは最小値が 1、最大値が 120 なので、本システムでは 25%の値である 30 を超えるものが現れるまで index を増やしながら検索するものとした。検索範囲はクリップから 15 秒間である 60 アイテムとした。これはシステム上で練習番号をタップして

から、一言伝えながらであっても長くても 15 秒以内には演奏を開始するであろうと予想したためである。こうして 30 を超えるものが現れたら、そのクリップの index をその位置に変更する。この時点でクリップはタップした時刻ではなく演奏を開始した（音が鳴り始めた）時刻になっていることが期待される。次に、この index が変化したクリップの小節番号から、次の小説までの時間を逐次足していく。次の小説までの時間は、楽曲構造データを用いて (現在の練習番号の拍子数) × (60/現在の練習番号のテンポ) で求めることができる。小節が進み、次の楽曲構造データのこの計算を繰り返し行い、play が false のクリップデータの index になるまで練習番号の補完が行われる。

5.4 可視化

音声ログの可視化には、上述の方法によって導かれた「楽譜との対応データ」、「楽曲の総小節数」、「音声ログの音量データ」を用いる。楽譜との対応データは前節で述べたものと同様、音声ログの音量データのインデックスに対応する数値, 楽譜上での小節番号, 演奏開始であれば true/演奏停止であれば false の真偽値から構成されている。楽曲の総小節数は楽曲構造データの totalBar の値そのものである。音声ログの音量データも前節同様、0.25 秒ごとの音量を表した 1 以上の浮動小数点の配列である。また、描画には HTML5 の Canvas API を利用している。Canvas API は web ページ上でプログラミング可能な図形の描画を行うための API で、リアルタイムのアニメーションや 3D の描画も可能である。

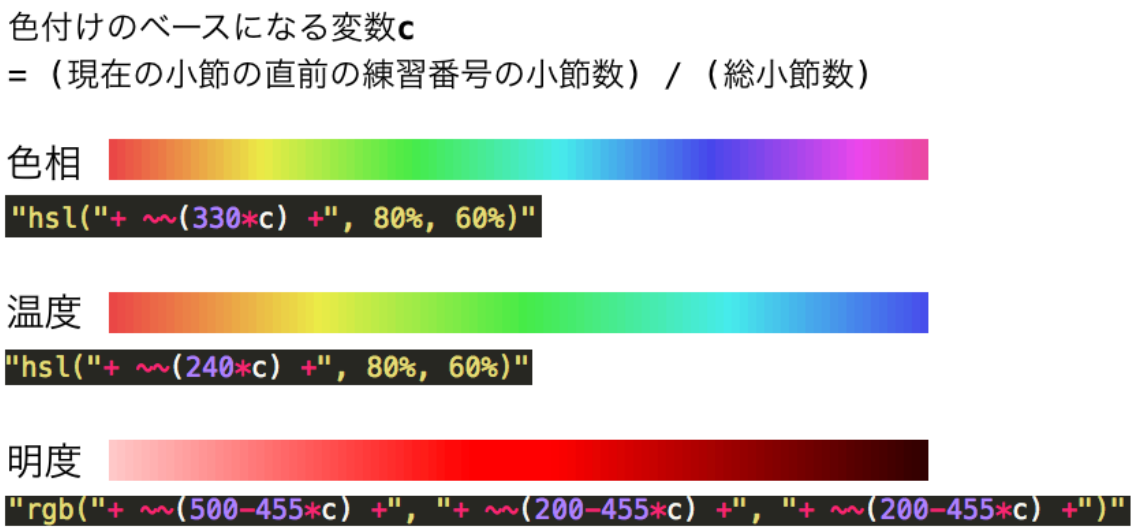

図 5.5: 3 種類の色付け方法

描画を行う drawCanvas メソッドは上記 3 つのデータと「canvas の DOM 要素」,「1 ピクセルあたりの音量データ数」,「オフセット」を引数に取り、返り値はない（canvas にその場で描画する）。1 ピクセル辺りの音量データ数は、音声ログビューに表示される場合は 1,2,4,8,16,32

の 6 段階の値が、楽譜ビューのコメント内に表示される場合は 1 で固定の値が来ることを想定している。オフセットは楽譜ビューのコメント内に表示される際にオプションで与える値（指定がない場合は 0）で、描画の開始 index を指定することができる。まず canvas の DOM 要素から getContext メソッドにより 2D コンテキストを取得し、clearRect メソッドにより canvas を白紙化する。ここから for 文で canvas の x 軸方向に 1 ピクセルずつ描画していく。オフセットの位置から開始して、1 ピクセル辺りの音量データ数ごとに音量データの最大値を求める。ここで音声ログをより見やすくするための工夫として、log 関数により音量の少ない部分の変化を大きく、音量の大きい部分の変化を小さくしている。最後にその音量データの最大値を canvas の DOM 要素の高さで正規化する（音声ログビューに表示される場合は 120、楽譜ビューのコメント内に表示される場合は 60）。この値は音声ログビューの波形の振幅として使用される。次に色を決定するため、現在の音量データの index と楽譜との対応データから現在の小節数を計算する。連続的に色を変化させる場合は現在の index 前後にある楽譜との対応データから、index の差から小節数の差を求め、総小節数で除算したものに 330 を乗算することで色相が求められる（図 5.5）。離散的に色を変化させる場合は現在の index 直前にある楽譜との対応データから、現在の小節数を総小節数で除算したものに 330 を乗算することで色相が求められる。ここでは色相の値は 0〜330 となるが、これは 360 まで使ってしまうと色の識別能力が低いユーザにとっては楽曲の最初の赤なのか、楽曲の最後の赤なのか判別しづらくなってしまう恐れがあったためである。また、次章で行った可視化に関する評価実験で用意した色付けの方法として、色相のほかに「温度」と「明度」を用いたものがある。「温度」は温度表現に使われる色を用いたもので、基本的には色相と同様であるが色相の値を 0〜240 とし、暖色から寒色への変化としてユーザが理解しやすくなるようにした。また、「明度」は明るさを変化させたもので、明るい赤から強い赤を介して黒まで変化するよう、RGB=(255,200,200) から (45,0,0) までの値を動く（図 5.5）。現在の index 直前にある楽譜との対応データの play が true であればその色で、false であれば演奏していないことを表すため灰色（#888）に canvas context の fillStyle をセットする。そして x 軸の幅は 1、y 軸の高さは上で求めた振幅で、上下方向にセンタリングされた位置になるよう fillRect メソッドによって canvas に描画される。

5.5 楽譜上での議論

楽譜上での議論には、楽譜・コメントエリア・コメント・音声ログなどの概念があるため、それぞれの実装について説明する。

楽譜ビューの HTML 構造を図 5.6 に示す。システムの楽譜はページごとに分割された PNG 画像ファイルを、Vue.js の v-repeat プロパティによって HTML に繰り返し描画している。楽譜は img 要素に描画され、全ての楽譜のページは img 要素を覆い被せるように canvas 要素が配置されている。これは、楽譜上のドラッグ操作によるコメントエリアの作成の際に、リアルタイムでユーザに範囲のフィードバックを与えるためのものである。それぞれの楽譜のページごとにそのページに属する練習番号のタグやコメントエリアである div 要素がフィルター付きの v-repeat プロパティによって生成される。これらのタグとコメントエリアは Vue インス

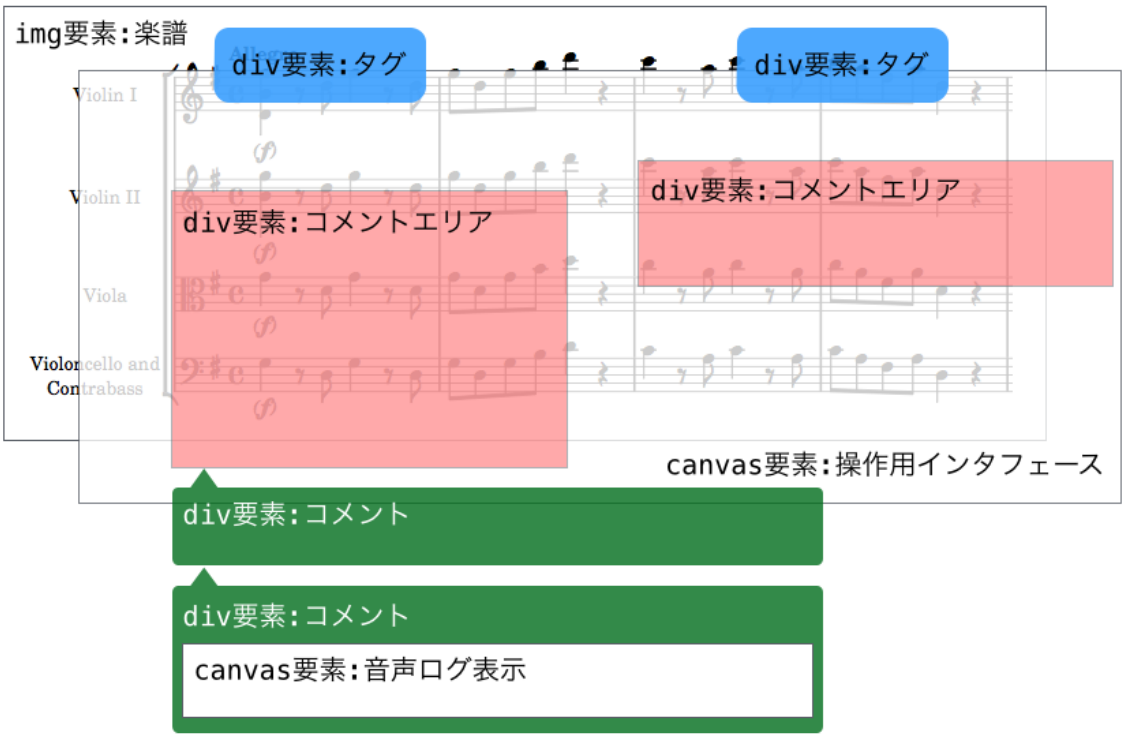

図 5.6: 楽譜ビューの HTML 構造

タンスのモデルと直に結びついており、HTML の style プロパティにモデルの値をバインディングすることで、表示の大きさや位置を決定づけている。各コメントエリアそれぞれ、属する各コメントが div 要素として v-repeat プロパティとフィルタによって繰り返し生成される。このコメントの div 要素は CSS によって吹き出しのような形状になっており、ユーザの投稿したメッセージであることが理解しやすい見た目になっている。v-if プロパティを利用してコメントに音声ログが含まれる場合のみ、可視化のための canvas 要素や日時を表示するための p 要素などを表示している。

コメントエリアを作成する際、ユーザはナビゲーションメニューからコメントエリア作成ボタンをクリックする。v-on プロパティによってイベントが発火し、body 要素を暗くし jQuery の on メソッドにより楽譜上の canvas 要素に mousedown イベントをバインドする。マウスが押された瞬間に、マウスが押された canvas 要素にドラッグ中に指定範囲をリアルタイムで表示するための mousemove イベント, 指定範囲を確定するための mouseup イベント, 範囲指定をキャンセルするための mouseout イベントが同時にバインドされる。mousedown, mouseup によって一定以上の大きさ（縦横 4 ピクセル以上とした）の範囲が指定されたら、Vue インスタンスのモデルの regions にページ番号, xy 座標, 高さと幅を記したオブジェクトを push する。regions モデルが変更されたタイミングで、予め設定しておいた$watch イベントが発火し、コメントエリアの再描画が行われる。

コメントを投稿するには、全てのコメントエリアの末尾にあるプラスマークの描かれた吹出しをクリックする。クリックすると、メッセージを入力するためのテキストボックス, 音声ログを含めるか選択するためのチェックボックス, 投稿のための post ボタンが表示される。このチェックボックスは、予め音声ログビューで音声ログの切り出しを行っておかないと非アクティブ状態のままとなる。post ボタンをクリックすると日時, テキストメッセージ, チェックボックスにチェックが入っている場合には切り出しの対象である音声ログの id と開始時間を含むオブジェクトが Vue インスタンスのモデルの regions[n].comments に push される。コメントエリア作成時と同様、regions に変更が加わったため$watch イベントが発火し、コメントエリアとコメントの再描画が行われることでユーザはコメントが投稿されたことがわかる。投稿されたコメントは×ボタンを押すことでアラートが表示された後、OK を選択することで削除することができる。

続いて音声ログの切り出しについて説明する。音声ログを議論で取り上げるべき理由として考えられるものは「音程のずれ」や「タイミングのずれ」、または人によって音符の長さが違うなどといった「演奏解釈のずれ」などである。いずれも楽曲の中では長くても数十秒程度の局所的なものであるため、切り出す音声ログにあまり長さは必要ないと考えた。そのため切り出しの範囲選択は開始点と終了点を決めるものではなく、長さを固定して開始点のみ選択する方法を取る。今回の実装では 90 秒とし、コメント内に表示する際には 1 秒 4 ピクセルとして描画するものとした。コメント内の音声ログは再生ボタンをクリックすることでその切り出した範囲の最初から音声を聴取することができる。また、音声ログはマウスホバーすることで音声の時間が表示され、クリックすることで自由な位置から再生することができる。なお、表示の範囲は 90 秒のみだが、内部的には演奏練習の音声ファイルを参照しているため範囲を超えて聴き続けることは可能である。

第6章　評価

本章では、提案するアプローチの有効性を評価するため、実装したシステムを用いて行った 3 つの実験について述べる。3 つの実験は、それぞれ「音声ログの楽譜への対応付けの推定」、「音声ログの楽譜への対応付けの可視化」、「演奏練習と議論におけるシステムの有用性」に関して有用性を評価するために行った。最初の節では、3 つの実験に使用した演奏練習のデータについて述べる。

6.1　評価に用いた演奏練習データ

6.2〜6.4 の実験に用いるため、実際の音楽活動を行う演奏者にシステムを使いながら演奏練習を行ってもらった。4 名× 2 グループが、それぞれのグループで 2 回ずつ演奏練習を開催した。楽曲は W.A.Mozart 作曲の Eine Kleine Nachtmusik より第 1 楽章を使用。1 回あたり 30 分間、システム上でクリップデータ作成のための操作を行いながら自由に練習を行ってもらった。これにより音声（MP3 ファイル）, 音量データ, クリップデータのセットを得た。ただし、これらのデータは合計 4 セット得られるはずであったが、片方のグループの 2 回目の演奏練習ではユーザが操作を間違え録音が途中で中断されてしまったため、その以外の 3 回分の音声ログとクリップデータを 6.1, 6.2 の実験に使用した。演奏練習の詳細な状況設定や環境については 6.3 節で詳しく述べる。

6.2　音声ログの楽譜への対応付けの推定

本システムの機能である音声ログの楽譜への対応付けの精度を調査するため、実際の演奏練習の音声とクリップデータを用いて評価を行なった。

6.2.1　実験内容

上述の演奏練習で得られた音声とクリップデータを用いて、実際の演奏位置とどの程度ズレが生じるのか調査した。楽曲のテンポは最初から最後まで一定の Allegro であり、システムに与える楽曲構造データのテンポとしては、NIKKO[1], YAMAHA[2], SEIKO[3]など多くの振り子

[1]NIKKO：http://nikkoseiki.com/
[2]YAMAHA メトロノーム：http://jp.yamaha.com/products/musical-instruments/keyboards/accessories/metronomes/
[3]SEIKO メトロノーム：http://www.sii.co.jp/music/

式メトロノームの目盛りにならい Allegro=132 とした。差異を比較するための実際の演奏位置データとして、筆者が音声を聴取し人力で作成した理想のクリップデータを用いた。

6.2.2 結果と考察

結果の全体像を図 6.1, 図 6.2, 図 6.3 に示す。

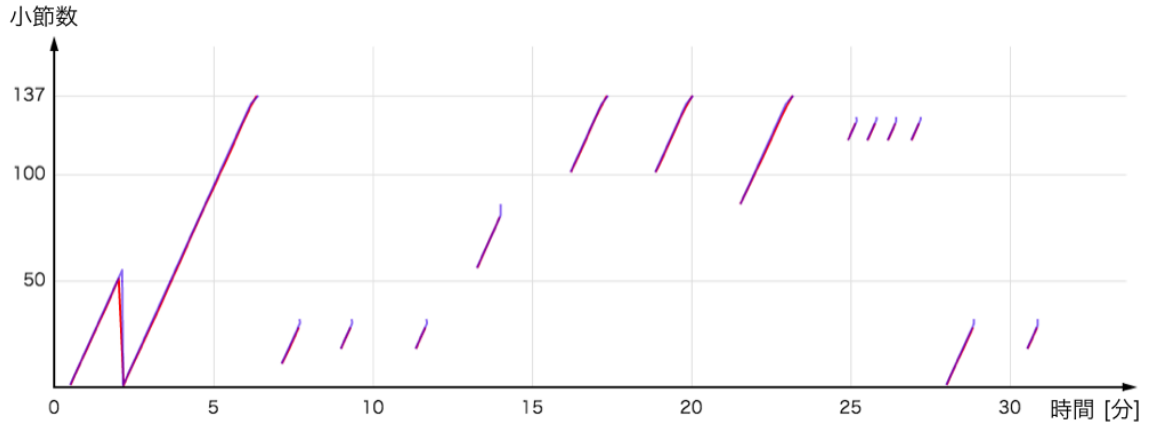

図 6.1: システムが推定した対応付け（青）と実際の対応（赤）：音声ログ A

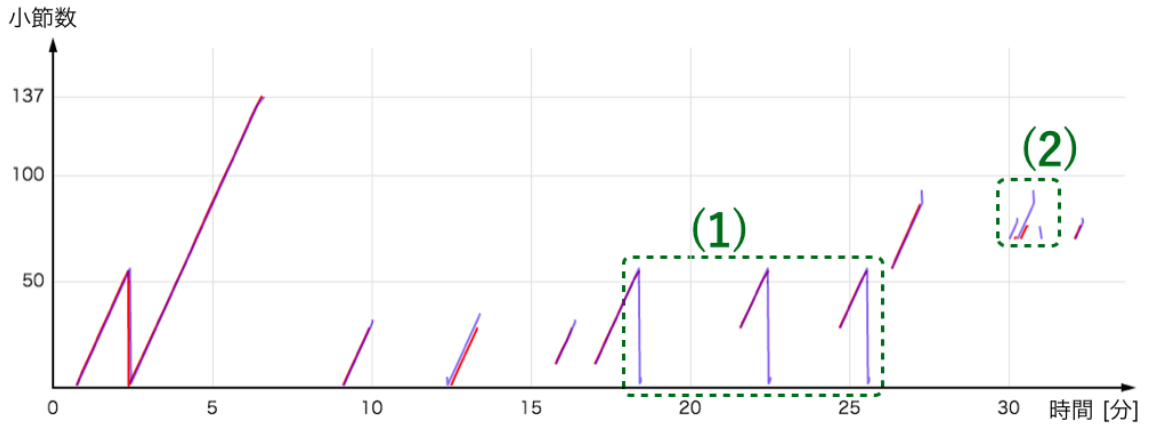

図 6.2: システムが推定した対応付け（青）と実際の対応（赤）：音声ログ B

図では横軸に演奏練習の経過時間、縦軸に演奏の小節数を取り、赤い線が実際の楽譜との対応、半透明の青い線がユーザのクリップデータからシステムが推定した楽譜との対応を表している。提案した手法によって概ね正しい推定ができていることがわかるが、実際の対応と大きな差異が出てしまった部分について説明する。図 6.2 の (1) は、実際の対応では 56 小節目付近で演奏が止まるが、推定した対応では 1 小節目に返ってしまっている。これは 55 小節目に繰り返し記号があるためであり、また演奏が止まる直前の差異であることから、実際

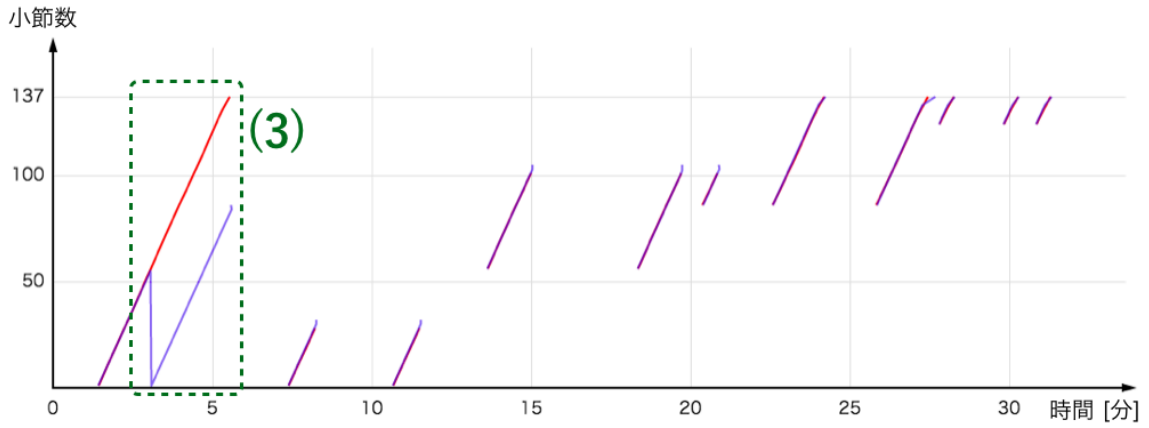

図 6.3: システムが推定した対応付け（青）と実際の対応（赤）：音声ログ C

にシステムで音声ログを聴取する際には問題にならないと考えている。同図の (2) は、実際に演奏されていない部分に推定が入ってしまっている。これはユーザが、システム上で練習番号をタップして演奏を開始したものの、そのフレーズの入りを間違えたため即座に演奏を止め、再度演奏を開始したりしたためである。特に (2) において赤線に比べて青線が長く続いている部分は、ユーザが演奏停止の際に押すべき stop ボタンを押し忘れているために発生した差異である。図 6.3 の (3) は、実際には繰り返し無しで演奏されているところを、繰り返しを勘定に入れて推定してしまっている。これはユーザが時間短縮のために繰り返し無しで通した（通す=楽曲全体を通しで演奏する）ためであるが、システムのひとつの限界を示している。正しく推定するためには繰り返しが有効かどうか演奏前にユーザに入力してもらう必要があるが、繰り返しが複数ある曲もあり全てを入力してもらうのは好ましくない。システムとしては寧ろ、あとで上述のような差異を簡単に修正できるインタフェース設計にするか、あるいは音量などのデータから自動で繰り返しを判定して推定を行うべきであろう。

システムが推定した演奏位置と、実際の演奏位置との差異のヒストグラムを図 6.4 に示す。図では横軸に推定した演奏位置の実際の演奏位置との差異、縦軸にクリップ数を取り、演奏練習の A, B, C それぞれが赤, 緑, 青に対応している。全ての推定されたクリップデータの 69.2%が ± 1 秒の範囲にあり、95.4%が± 2 秒の範囲にある。データがマイナス方向に偏っているが、これはユーザの操作によるクリップデータは必ず演奏開始よりも時間的にマイナス方向にあり、推定はそのクリップデータを起点にプラス方向に行われるためであると考えられる。すなわち、ユーザがシステム上の練習番号をタップしてから演奏を開始するまで、声や呼吸などのノイズが推定の閾値を超える場合に起こりうる。逆にプラス方向のデータは、演奏の最初が小さい音量で始まるために推定の閾値を超えるのが遅れる場合であると考えられる。また、図 6.4 において-8.75 秒から-7.5 秒の範囲にあるデータは、ユーザがシステム上の練習番号をタップした瞬間に声を出していたため（タップしたその場で推定の閾値を超えたため）、大幅に手前に推定される結果となった。これは推定のアルゴリズムを、単純に閾値を超えた瞬間にするのではなく、数秒間の平均値を取るなどすることで精度を高めることができるか

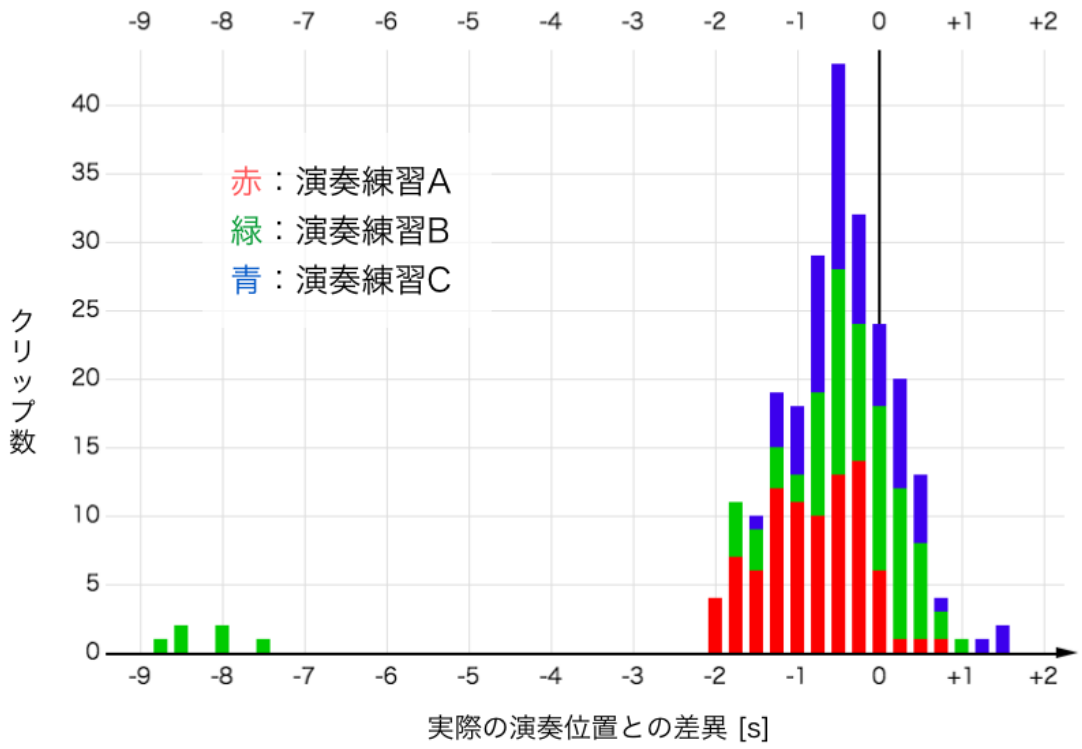

図 6.4: システムが推定した演奏位置と実際の演奏位置との差異

もしれない。

今回、実際の演奏練習 3 回分のデータからは非常に良い推定の結果が得られたと思うが、これは楽曲や演奏団体に大きく依存した結果である可能性がある。つまり、今回用いた楽曲である Eine Kleine Nachtmusik は、全曲を通してテンポが Allegro 一定であり、推定には非常に好都合な楽曲であったと言える。第 3 章で述べたように、楽曲によってはアッチェレランド・リタルダンドなどの加速・減速やフェルマータなどの休止など、不確実性の高い要素が含まれる。また、Allegro の実際のテンポをメトロノームの目盛りから 132 拍毎分としたが、これも偶然今回のユーザがそれに近いテンポで演奏練習を行っただけとも言える。Allegro とは「軽快に」などと訳されるように、決まったテンポがあるわけではなく演奏団体によってそのテンポは大きく左右される。以上のことから幅広い楽曲の推定を達成するには、ソフトウェア的にメトロノーム機能をつけたり、足踏みペダルをユーザに踏んでもらったり、あるいは音楽情報処理の技術を用いてビートトラッキングしたりするなど、何らかの方法でテンポ情報を取得する必要があるであろう。

6.3　音声ログの楽譜への対応付けの可視化

音声ログの楽譜への対応付けの可視化によって、音声ログの中から聴きたい箇所を探しだすのに役立つかどうか、またどのような色付けの方法が最も好まれるかを調査するため、実際の演奏者を対象とした調査を行った。

6.3.1　実験内容

実験は被験者ごとに個別に行なった。手順として、まず被験者に演奏練習の録音を有効活用するためのシステムを作っていることを説明し、音声ログの楽譜への対応付けの可視化手法（縦軸は音量、横軸は時間、色は楽譜上の小節数に対応しているということ）について説明した。この説明では図 6.5 のようなシステム上の可視化表現を見せながら、「例えば曲の冒頭部分は赤色に対応しているから赤色の部分は冒頭部分を練習していることがわかる」といった具体例を示しながら説明した（6.1 節で述べた演奏練習データをこの説明に用いた）。可視化された図は音声ファイルと結びついており、システム上で音声ログをクリックすると音声を再生することができることも説明した。次に、色付けの方法として色相の全体を使ったもの・温度の表現を模倣したもの・明るさの変化のみを使ったもの（図 3.5）、3 種類それぞれについて被験者間でカウンターバランスを取った順序で長所と短所などの特徴を説明した。この際、演奏練習 3 回分の音声ログをそれぞれの色付けの方法によって可視化されたものを MacBook Air 11 インチに表示しながら説明した。そしてアンケートに回答してもらった。

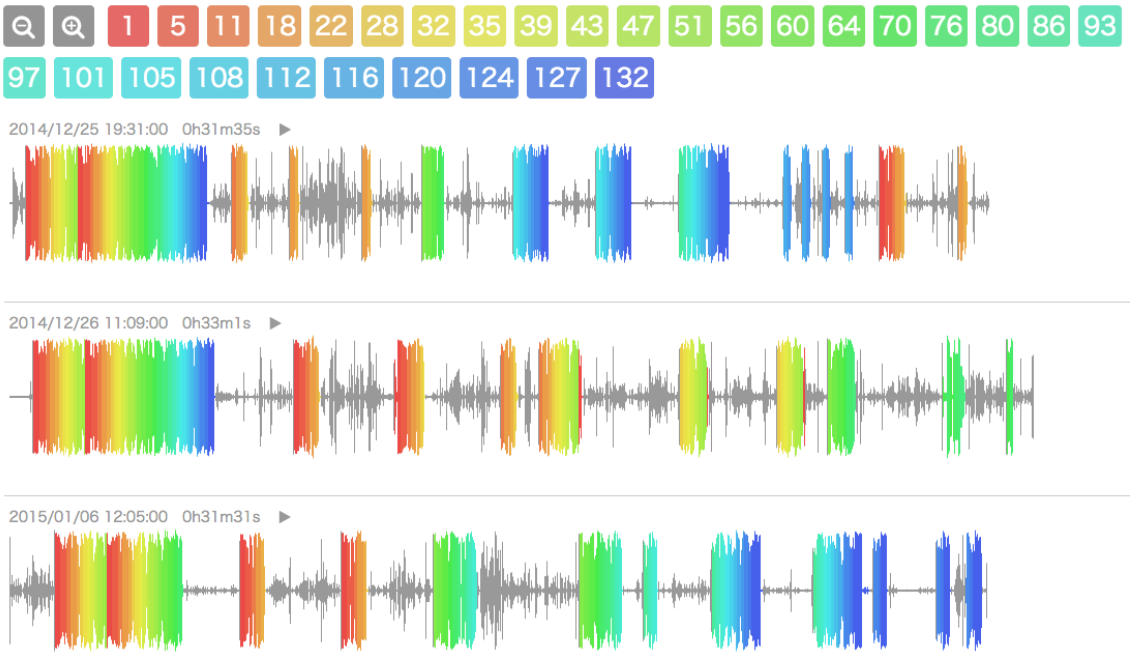

図 6.5: 説明に使用した音声ログのシステム上の可視化表現

アンケートの内容は以下のとおりである。5 段階評価は「1:全くそう思わない」から「5:非常にそう思う」までの 5 段階から 1 つを選択してもらった。

- 聴きたい箇所を探すのに役立つ（5 段階）
- 楽譜上で特定の箇所をどのくらい練習してきたかを確認するのに役立つ（5 段階）
- 色付けされた音声ログを普段から利用したい（5 段階）
- 最も好きな色付けの方法（色相・温度・明暗 から選択）とその理由（記述）

被験者は音楽団体に属し演奏練習の録音を普段から利用しており、3 年以上の楽器演奏経験を持つ大学生・大学院生 12 名（20〜23 歳、男性 6 名、女性 6 名）であった。また 12 名のうち、「色相」を知っていると答えたのは 4 名、色覚異常を持っていると答えたのは 1 名であった。

6.3.2　結果と考察

アンケート結果を図 6.6 に示す。

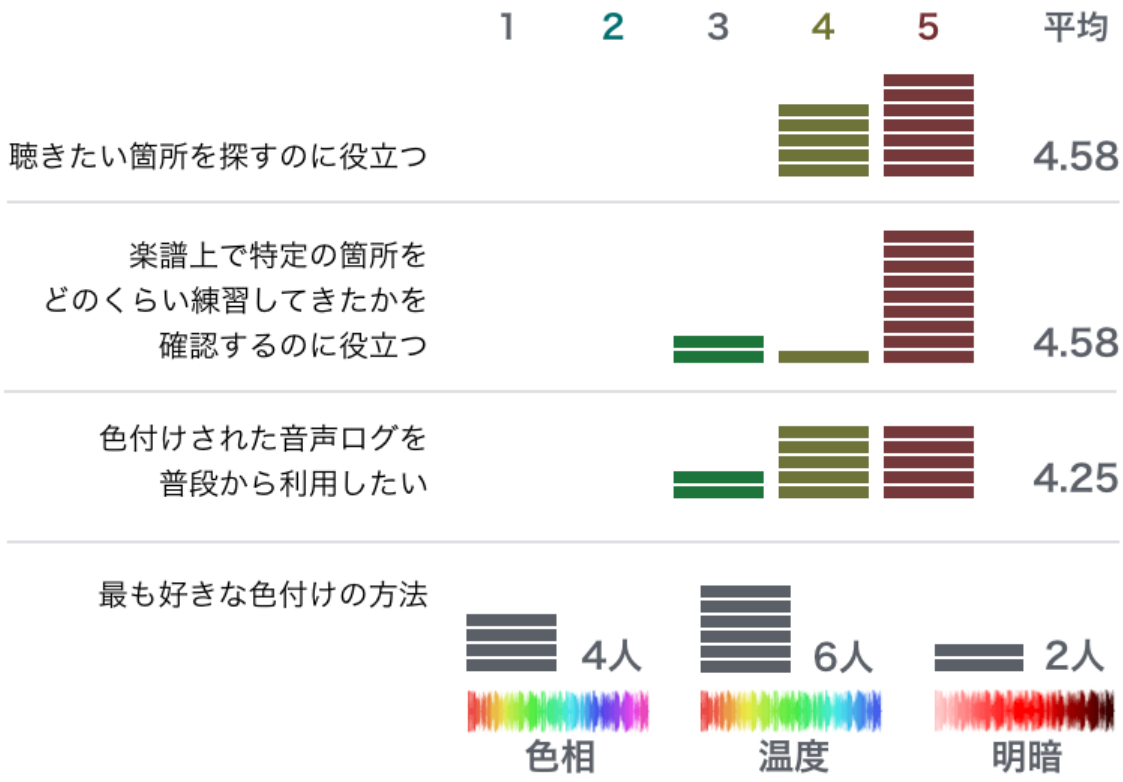

図 6.6: アンケート結果：音声ログの楽譜への対応付けの可視化

3 つの評価項目の平均値はそれぞれ 4.58, 4.58, 4.25（小数点第 3 位四捨五入）と、極めて高い数値を得ることができた。また、実験の説明において「色は小節に対応付けされており、色の付いていない灰色の部分は会話など演奏していないことを表している」ということを説明した際に、多くの被験者がポジティブな反応を示していたことが印象深かった。このことから、本研究で提案した音声ログの楽譜への対応付けの可視化手法は、音声ログの有効活用に非常に効果的であると言える。その他、「このようなシステムが普段の合奏やセク練（=セクション練習。弦楽器・木管楽器・金管楽器などそれぞれで集まって行われる練習のこと）にあれば、助かるし効率的だと思いました」という意見も得ることができ、音声ログの可視化のみであっても十分な有用性と需要があることが分かった。

最も好きな色付けの手法については、12 名のうち 4 名が色相, 6 名が温度, 2 名が明暗と回答した。このうち、色相を知っていると答えたのは 4 名で、内訳は色相 2 名, 温度 1 名, 明暗 1 名であった。選択の理由として、色相は「一番判別がしやすく、聴きたい部分を探すのに便利だと思うから」「慣れてしまえば細かい小節の判定がつく色相のほうが良い。長い曲になるほど細かい色分けが欲しくなると思う」などが挙げられた。温度は「色相と温度でまよったが、色が多すぎてもかえって見にくい印象を受けた」「色によって時間が簡単に分かったほうが良いが、明暗は判別しにくいと感じました」「色相と明暗の長所を併せ持っていて、パッと分かりやすいと思います」などの理由であった。明暗は「覚えやすい。他は直覚的に覚えにくく、都度参照が必要」「ぱっと見でどの辺りか個人的にわかりやすい。あまり色が多いと対応させるのが少々面倒」とのことであった（いずれも原文のまま引用）。この結果から、システムの色付けは温度をデフォルトにしつつ、ユーザ各自が好みに合わせて色相や明暗にも切り替えることができることが望ましいことが分かった。

6.4 演奏練習と議論におけるシステムの有用性

本システム利用することで、音声ログの管理・聴取がやりやすくなるかどうか、また音声ログを用いた議論がどのようにして行われるかを調査するため、実際の演奏者を対象とした評価実験を行った。

6.4.1 実験内容

本実験の大まかな流れとして、(1) 対面環境におけるシステムを利用した練習、(2) 遠隔非同期環境におけるシステムを利用した議論、(3) アンケート及びインタビューでの評価、の 3 段階に分けられる。すべての段階で、被験者は一般的な弦楽四重奏の構成（ヴァイオリン 2 名・ヴィオラ 1 名・チェロ 1 名）を 1 つのグループとして進行した。被験者は 3 年以上の楽器演奏経験を持つ大学生・大学院生 8 名（20〜24 歳、男性 4 名、女性 4 名）であった。なお、この実験の (1) と (3) で得られた音声, 音量データ, クリップデータを 6.2, 6.3 の実験に用いた。

(1) 対面環境におけるシステムを利用した練習

被験者のグループを演奏練習のできる環境に集め、最初の 15 分程度で実験の目的とシステムの練習時における使い方を説明した。

図 6.7: (1),(3) における演奏練習の環境

その後 10 分間程度を楽器準備・音出しの時間に充て、更に 30 分間システムを利用しながら楽曲の演奏練習を行った（楽曲は W.A.Mozart 作曲の Eine Kleine Nachtmusik より第 1 楽章を使用。一般的な演奏時間は前半繰り返し有り・後半繰り返し無しで約 5 分半。被験者には予めこの楽曲の演奏練習をすることを伝えてあり、個々人での譜読み練習は済んでいる段階を想定)。基本的には被験者の日常的な練習と同様に、椅子に着席して半円状になり、各自が各パートの紙の楽譜を譜面台に置いていた（図 6.7)。1st ヴァイオリンを担当する被験者を演奏練習の進行役とし、VAIO Tap20 を譜面台の隣に配置してシステムを操作してもらった（図 6.7 下部, 図 6.8)。被験者が行ったシステム上での具体的な操作は、演奏練習全体の開始/終了時の録音開始/終了ボタンのタップと、音出しを始める前/終えた後に楽譜ビュー上の対応するタグをタップしてもらうことであった。演奏練習中は常に筆者が近くで様子を観察しており、数回写真を撮影した。

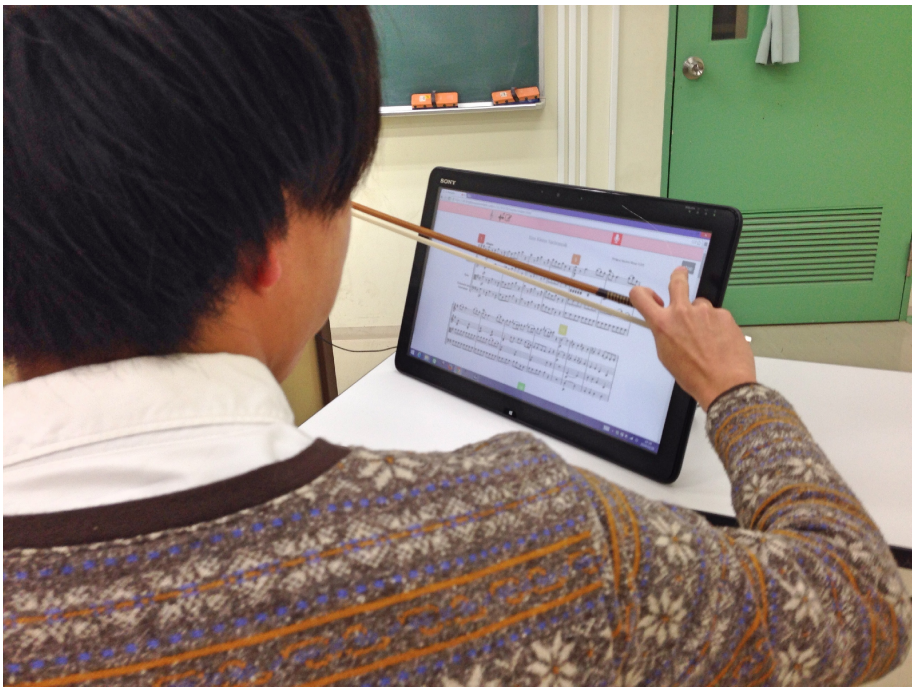

図 6.8: (1),(3) におけるシステムの利用環境

演奏練習の終了後に、再び 15 分ほど使ってシステムの議論時における使い方と留意点について説明した。留意点に関しては下記の (2) に述べる。

(2) 遠隔非同期環境におけるシステムを利用した議論

(1) の演奏練習の音声ログがシステム上で利用可能な状態で、1 週間ほどかけてシステム上で議論を行ってもらった。被験者は各自の都合の良い時間帯に、各自の web ブラウジング可能なデバイスを用いてシステムを利用した。留意点として予め被験者には、システム上での議論は演奏会に向けて演奏の質を高める意識を持って望んでほしいこと、活発にシステムを利用し議論を進めた者に対して歩合制の報奨を与えることを伝えた。また、基本的には被験者各自が最も使い慣れている PC・マウス・キーボード環境での利用を推奨したが、スマートフォンやタブレットなどのタッチデバイスでもコメントの閲覧や返信は可能である旨を伝えた。

(3) 対面環境における 2 度目の練習

(2) の議論の期間の終了後再び被験者のグループを集め、(1) の演奏練習と同様に 10 分程度を楽器準備・音出しの時間に充て 30 分間楽曲の演奏練習を行った。練習の環境は基本的には (1) と同様であるが、(2) にて行った議論の内容を確認しながら進めることが可能であった（図 6.9）。

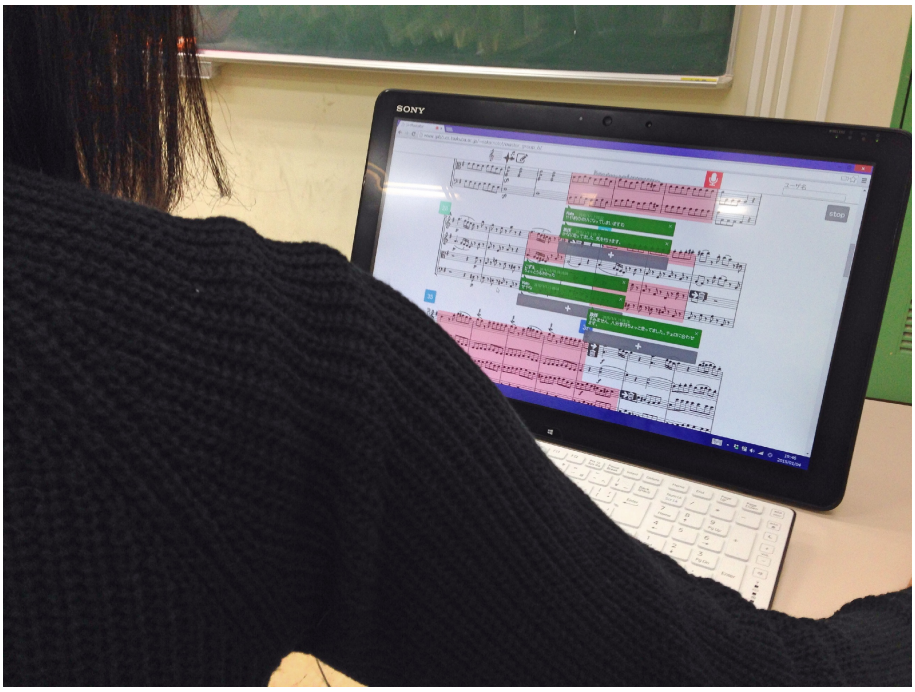

図 6.9: (3) において過去の議論内容を確認する被験者

演奏練習の終了後、アンケートに回答してもらった。その後 15 分ほど使って、被験者 4 名と筆者による座談会形式で意見や感想を述べてもらった。アンケートと座談会の内容は以下のとおりである。

- アンケート（「1:全くそう思わない」から「5:非常にそう思う」までの 5 段階評価。理由があれば併記してもらった）
  - 練習中のシステム上での練習番号のタップ操作は煩わしくなかった
  - 練習中のシステム上での練習番号のタップ操作はわかりやすかった
  - コメントが楽譜上の位置に紐づいていることは議論に役立った
  - 新しいコメントを投稿する際の楽譜上の位置の指定方法はわかりやすかった
  - 音声ログが聴きたい箇所から聴けることは議論に役立った
  - 音声ログの一部を切り出し、コメントに含めることは議論に役立った
  - 音声ログの一部を切り出し、コメントに含める操作はわかりやすかった
  - 音声ログ・スコア・議論の場が統一されていることは良いことだ
  - このシステムを自分自身の音楽活動・演奏練習に取り入れたい
- アンケート（自由解答）

- 良いと感じた点
- 改善するべきだと感じた点
- システムを使った感想・その他の意見など

- 座談会のテーマ
  - 議論してみた感想・実際使ってみたいかどうか
  - システムに加えてほしい機能
  - 音声ログの更なる活用方法 など

6.4.2 結果と考察

得られたデータとして、アンケート結果（5 段階評価及び自由記述）, 座談会で得られた意見・感想, システムの試用を通じて得られたコメントがある。まず、アンケート結果を図 6.10 に示す。

全体として概ね良い評価を得ることができた。特に「コメントが楽譜上の位置に紐づいていること」「音声ログが聴きたい箇所から聴けること」「音声ログ・スコア・議論の場が統一されていること」は高い評価を得ていることから、本研究のアプローチの有効性が示された。すべての被験者が「システムを自分自身の音楽活動・演奏練習に取り入れたい」と考えている（4 か 5 の評価を選択している）ことから、本システムの実用性と需要は十分であると言えるであろう。反面、特に低い評価を得たのが「音声ログの一部を切り出し、コメントに含める操作」であり、今回は音声ログの一部を切り出してコメント投稿の際に含めるかどうかを選択する方法で実装したが、UI や操作の流れ自体を改善する必要があると言える。「音声ログの一部を切り出し、コメントに含めることは議論に役立った」かどうかは平均 3.75 とまずまずの成績であったが、上記の操作方法を改善したり楽譜やコメントとの結びつき方を変えたりすることで、より高い評価が得られるようになると考えられる。また、「練習中のシステム上での練習番号のタップ操作」についても平均は 3.63 であるが 2 を選択した被験者もいたため、タップ操作よりも良い入力方法を模索するべきだと言える。

良いと感じた点で挙げられた意見として、「みんなが離れている場所で、同じスコア上で議論できるのはとても良かった。」「自分の時間があるときにゆっくりきいて、意見を共有できることがいいと思った。」「楽譜に意見が書いてあるので問題の箇所を重点的に練習できた。」（いずれも原文のまま引用。以下同様）など、演奏に関して遠隔非同期的な議論を行ったことがない者であっても、本システム上での議論は好意的に受け入れられた。また、「聴きたいところだけ取り出して聴けることがいいと思った。」「録音が簡単。再生する時に場所が見やすい（色がグラデーションになっている）」など、可視化された音声ログについても良い評価を得ることができた。

反面、悪いと感じた点・改善するべきだと感じた点として、「パソコン環境がないときが多いので、スマホとかでも出来るようになるとさらに良いと思った。」という意見があり、これは現状の実装ではコメントエリアの作成にドラッグ操作が必要であり一部の機能が現状のス

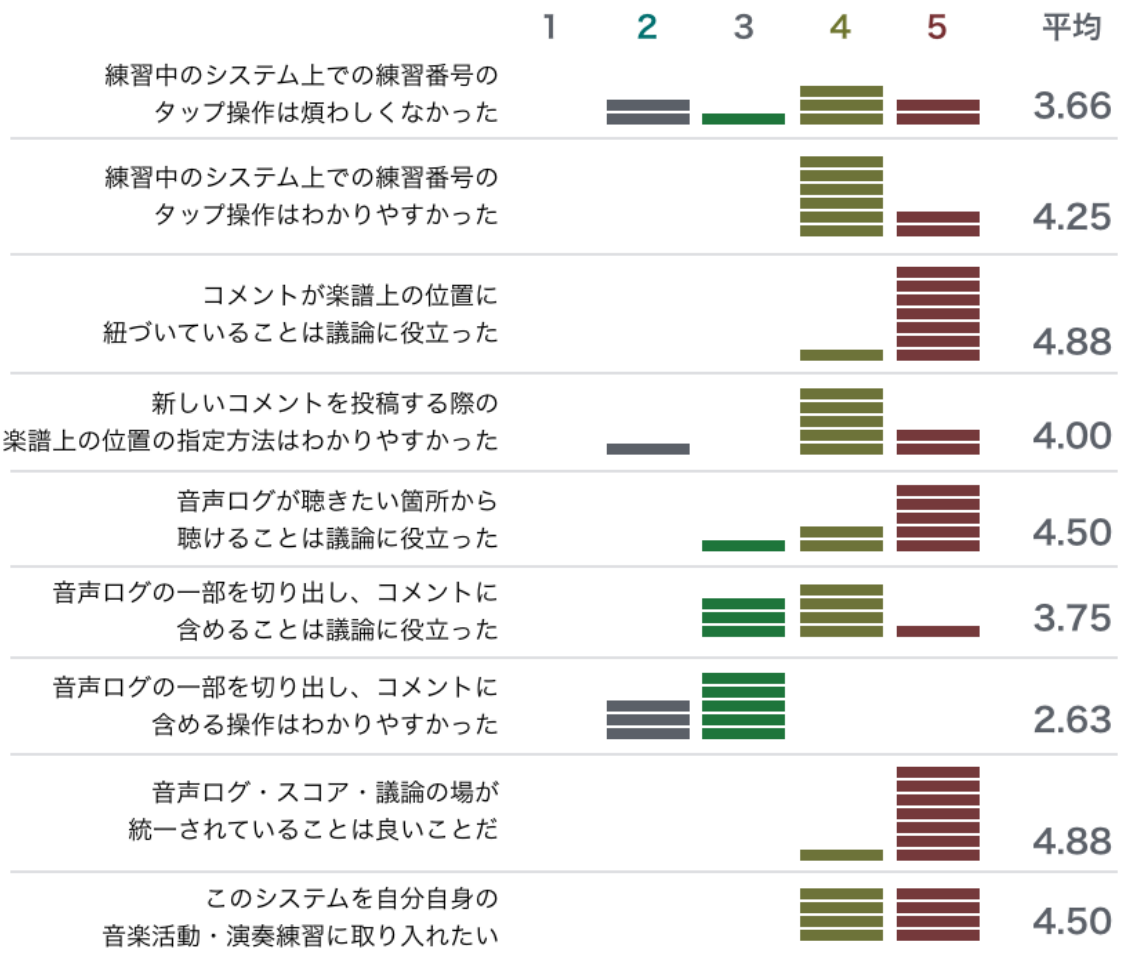

図 6.10: アンケート結果：演奏練習と議論におけるシステムの有用性

マートフォンの web ブラウザに対応していないためであった。演奏者が空き時間を利用して議論する、という本研究のコンセプトからは、むしろスマートフォンでの利用をメインに考えても良かったのかもしれない。また、「『こうしたい』ということを、言葉で説明しなければならない」という意見があり、これは対面での議論であれば実際に楽器で演奏したり歌ったりして表現することができるが、本システムではテキストで表現しなければならない、という限界を示している。しかし、これはユーザが個別に録音を行いその音声ログをコメントに引用することで、システム上の議論でも実際に楽器で演奏したり歌ったりして表現することが可能になると思われる（(2) の遠隔非同期環境における議論ではシステムの録音機能をオフにしていた）。その他、「コメントりれきみたいなのがほしい。新しいコメントがないかすべてのコメント箇所を開かないといけない。」「音声再生ページと、楽譜書き込みページが同一画面 or 分割で表示できるとより使いやすい（音声ページから楽譜に戻ると一番上に戻ってしまう！）」「音声を再生しているときにどこを再生しているのかわからない　再生かしょを示す

バーがほしい」など、欲しい機能やより良い発展に関する意見を多く得ることができた。これらの意見は、今後の研究においてシステムを発展させる際に参考にさせていただく。

システムを使った感想・その他の意見などの項目では、「新しい試みでとても面白かった。アンサンブルでも是非活用してみたい。」「なかなか時間が合わない人にとって、良いシステムだと思う。」「集まる時間がない中でそれぞれの意見を楽譜と一緒に共有できたので練習をするときに効率よく進められると思った。」「練習の間隔が広いとき、なかなか会えない人とアンサンブルするときに使いやすい。」などの感想が得られた。本研究の狙いである、空いた時間を利用した議論による演奏練習の効率化は、概ね達成することができたと言えよう。また、この項目でも欲しい機能として「ページジャンプなどがあると嬉しいかも。いちいちスライドしていくのが大変」「LINE 等の SNS で他の人に「これどう思う？」みたいな感じでシェアできたりするといいなと思いました。」などのアイディアを得ることができた。

次に、アンケートの後に行った座談会で得られた意見として特徴的なものを挙げていく。まず「文字だけで議論を行うのには限界がある。対面での議論であれば楽器で弾いたり歌ったりすることで、演奏の微妙なニュアンスを伝えることできる」という意見を得た。これは本実験の悔いの残る点であるが、(2) の遠隔非同期的な議論のフェイズではユーザのデバイス間での差異をなくすためにシステムの録音機能をオフにしていた。録音機能を演奏練習以外の時間にも使えるようにすることで、個人の楽器演奏や歌を音声ログとして残し、それをコメントに含めることで対面環境に近い意思伝達が実現できると考えている。また、「楽譜を MIDI でもいいので再生できるといい」という意見を得た。楽曲の演奏練習の初期段階ではざっくりとした曲のイメージを掴むために聴いてみたい、というものであった。「録音のデシベル値を表示することで、p（ピアノ）や f（フォルテ）の音量の幅に参考にできるのではないか」という意見も挙げられた。通常の演奏練習ではこのあたりの音の強弱は演奏者の間隔に委ねられているが、敢えて機械的に値を測定することで間隔と実際とのすり合わせを行い、演奏の自信にもつながるのではないかとのことであった。

また、「今回は演奏練習中には 1st ヴァイオリンの演奏者のみがシステムを利用できた（フルスコアが閲覧できた）が、全員が iPad などの端末を用意してみている画面を共有できると素晴らしい」という意見を得た。この実験で得られたひとつの知見として、演奏者はフルスコアを常に閲覧できる環境にあるということそれだけでも得られる恩恵が大きいというものがあった。そのためシステムを利用して常にフルスコアを閲覧できるようにし、演奏者間で画面の共有を実現すれば高い有用性を発揮できるであろう。「今回は四重奏の形式での議論だったが、むしろ（7,80 人規模の人数の）合奏での利用に真価を発揮するのではないか」という意見も得ることができた。オーケストラでは、演奏者の数が多いため個々人が思い思いの解釈で演奏するわけにはいかず、全体で意識や解釈を統一する必要がある。そのため本システムのような web を介して意見を共有できるものは、議論よりも意思の周知に威力を発揮することになるであろう。その他「テンポの歪みを検出して見れると良い」「音程のずれを指摘して欲しい」「スマートフォンで利用したい」など、システム拡張の方向性について多くの示唆を得た。また、「色付けされた音声ログを見て、意外と音を出している時間が少ないことに気づいた」といった感想も得た。

次に、システム上で行われた議論や得られたコメントについて考察する。両グループ合計で 33 のコメントエリアと 68 のコメントを得た。

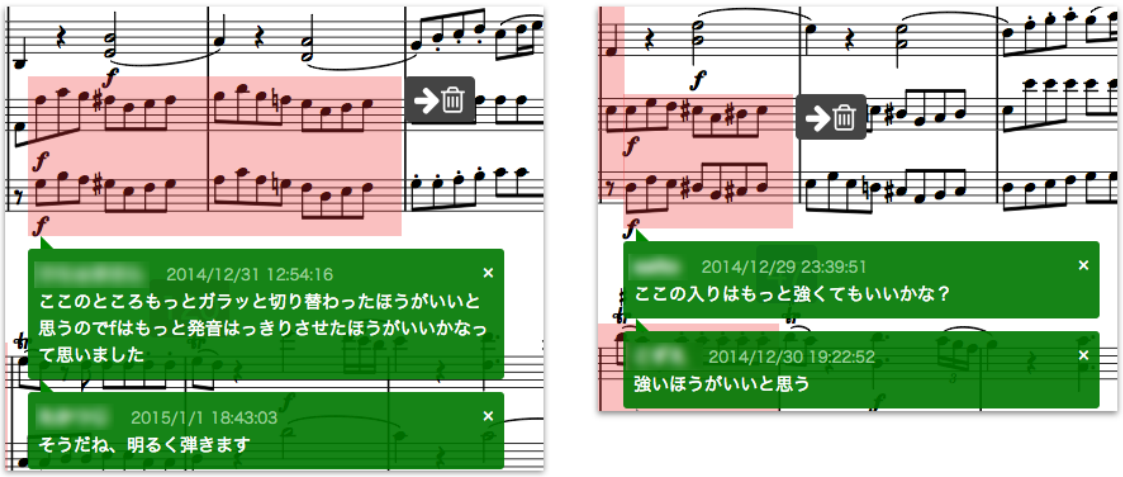

図 6.11: 実験で得られたコメント：短い議論

図 6.11 は実験で得られたコメントの中でも、スムーズに結論が導かれたものである。左右それぞれ別々のグループのものであり、どちらも楽譜上の同じ箇所についての議論であった。ここは直前まで p（ピアノ）で落ち着いた雰囲気であったものが、この小節の 2 音目からヴィオラとチェロの音量が f（フォルテ）に急に上がる部分である。「ガラッと切り替わ」る,「発音はっきり」,「入りはもっと強く」など、ここから演奏の勢いを変えることを意識している点は両グループとも同じであった。しかし、図 6.11 左はチェロの奏者の意見に対して同じフレーズを演奏するヴィオラ奏者が同意の意見を投稿しているが、図 6.11 右はチェロ奏者の意見に対して 1st ヴァイオリン奏者が客観的に意見を述べている点では、議論の形式は異なっていると言える。実際にそのフレーズを演奏する者だけではなく、全体を巻き込んだ議論が可能であるという例であると言える。

図 6.12 は実験で得られたコメントの中でも、特にコメント数が多く議論が長くなったものである。図 6.12 左はフレーズが区切れて改めて p（ピアノ）の静かな雰囲気で始まるフレーズであった。ここでは「強弱について」,「テンポについて」,「音色について」それぞれの意見が出されているが、各コメントの方向性は一貫しており、このフレーズから雰囲気を変えて演奏するという全体の意識がある中で細かい注意点などを共有している様に見受けられる。一方で図 6.12 右は sf（スフォルツァンド：特に強く、の意）と p（ピアノ）の処理について、三者三様の意見が投稿されている。最初のユーザは「sf の付いている音符のみ強くする」という解釈、次のユーザは「sf は次の p まで有効なので 4 つの音符を強くする」という解釈、最後のユーザは「sf からデクレッシェンド（徐々に弱く）して p に到達する」という解釈である。ここではシステム上での議論によって解釈のずれが明らかになり、次回の練習で取り上げる方針を決めるに至っている。システムによって演奏練習が効率化できたひとつの例であると言えよう。

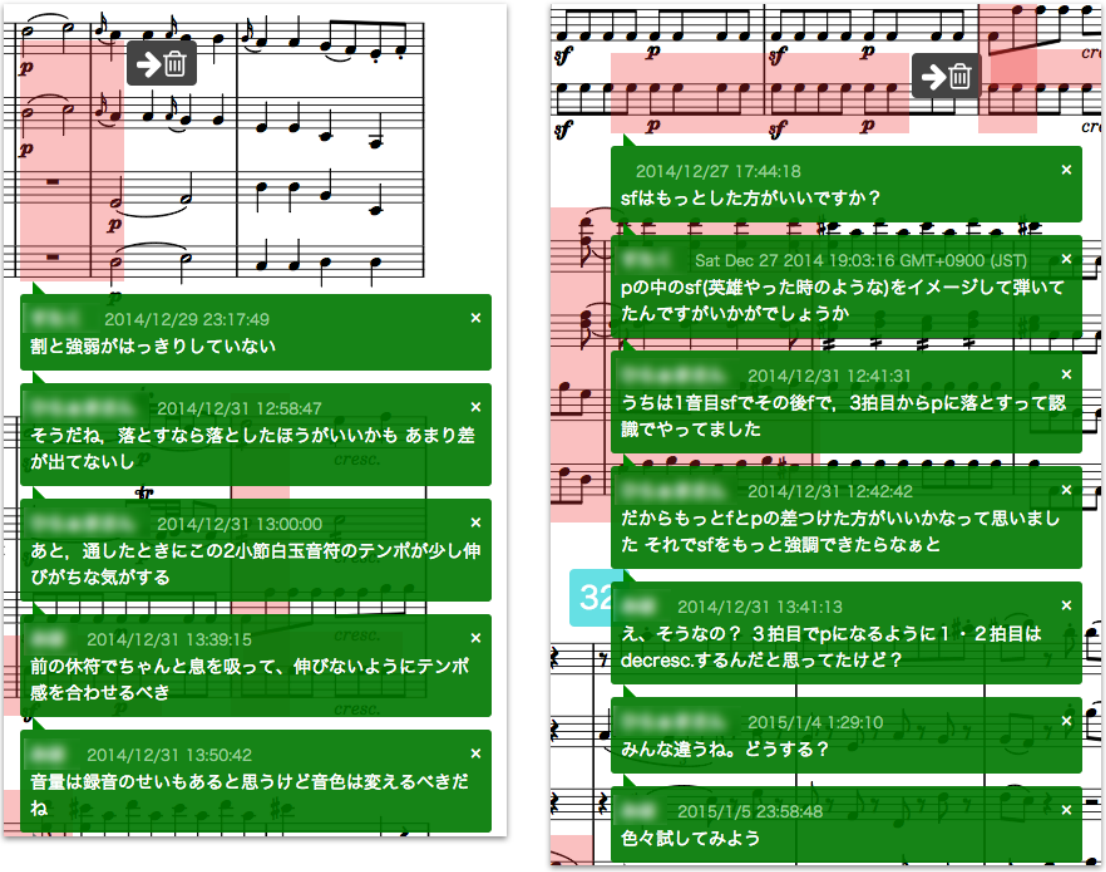

図 6.12: 実験で得られたコメント：長い議論

図 6.13 は実験で得られたコメントの中で、切り出した音声ログを含めたものである。今回の実験では残念ながら操作性が悪かったせいか、この 2 つのコメントしか音声ログ入りのコメントを得ることができなかった。図 6.13 左は、音声ログを聴取して本来アクセントの付いていない音符にアクセントが付いているかのように聴こえたユーザが、その部分の音声ログを含むコメントを投稿することで注意を促している。指摘されたユーザも癖であることを自覚しており、指摘にしたがい「そうならないように気をつける」と応答している。図 6.13 右は、弓順（弦楽器の演奏において、弓をアップボウかダウンボウのどちらで弾くか）に関して音声ログを交えて問題提起している。これは 1 度目の演奏練習中に弓順について議論された箇所であり、音声ログを聴くことで更に議論の続きを行っている。一般的にアップボウは気を抜くと入りが弱くなり「cresc.（徐々に大きくなる）気味」になりがちであるため、それに注意するようコメントが交わされている。

図 6.14 は実験で得られたコメントの中で、休符に対してコメントエリアが作成されていることが特徴的であった。コメントが「（楽曲）全体」に対する内容であることと、コメントしたユーザがチェロ奏者であったことからこのような絶妙な箇所にコメントエリアを作成した

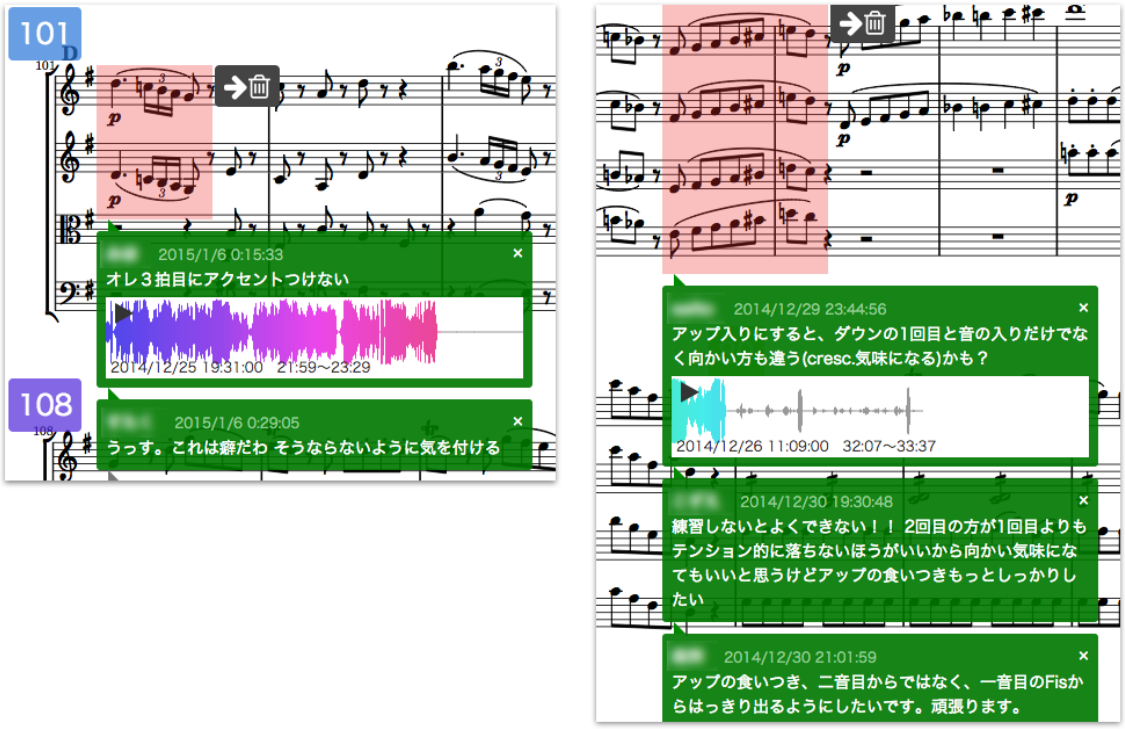

図 6.13: 実験で得られたコメント：音声ログを含むコメント

ことが推測される。真意は不明だが、続く 2 人のユーザはコメントエリアの箇所について意見を述べているため、最初のユーザは楽曲全体に対して議論したかったのに特定の箇所についての意見が返されて齟齬が発生していた可能性がある。この対策としては、楽曲全体やそれに準ずる広い範囲（コーダから楽曲の最後まで全般、第 1 主題が出てくる部分、など）に対するコメントの投稿方法を容易する必要があるだろう。具体的な案としては、システムの左右の余白を使う, 広範囲のコメント用のスペースを別途設けるなどの拡張が考えられる。

次に、(1),(3) の対面環境での演奏練習において観察していて明らかになった点について考察する。演奏練習において、「どこから始めようか」「15 くらいからやりたい」「じゃあ（システム上では 15 小節目にタグが無かったため）11 から」というやり取りがあった。瑣末なやり取りではあるが、演奏練習がシステムに縛られたひとつの例である。演奏練習時においては、システムはあくまで補助的な役割を果たすべきであって、自由な演奏練習を妨げるべきではないと考えている。上述の例では、例えばすべての小節にタグを用意すれば問題なかったが、弱起（フレーズが小節の途中から始まること）や小節内にフェルマータ（音符や休符を長く伸ばす）のある楽曲は必ずしも小節の先頭から演奏するわけではないため、ユーザは操作に困惑してしまうかもしれない。この問題の解決策としては、ユーザが演奏を開始する際にタグではなく楽譜上の自由な位置を選択する方法が考えられる。

また、両グループともに 2 回目の演奏練習ではシステム上のコメントを見ながら練習を進めていた。システム上のコメントエリアを上から順にタップしてコメントを表示し、議論し

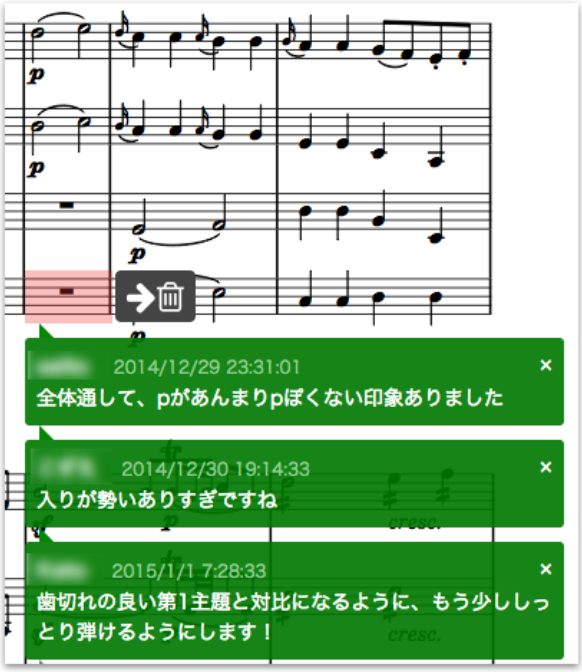

図 6.14: 実験で得られたコメント：休符に対するコメントエリア

た内容を確認したり、システム上では結論が出せなかった議論について実際の演奏を交えながら練習したりしていた。これはシステムの当初の狙い通りであり、遠隔非同期的に議論を行っておくことで対面環境での演奏練習の効率を高め、スムーズに進行することが可能になったと言えよう。

以上の実験結果から総合的に得られた新たな課題として、大きな編成の曲や演奏時間の長い曲ではシステムの利便性が低下するおそれがある。被験者実験では楽譜は 4 パートだけであり、また曲の長さも 137 小節と比較的短いものであったため、全曲を一度に表示してスクロールする仕様に大きな問題はなかった。しかし例えばフルオーケストラの曲では楽譜は 20 段ほどになり、500 小節（演奏時間は 20 分以上）を超える曲も珍しくない。解決策としては上述の被験者から得られたアイディアであるページング機能も考えられるが、多くのページを辿るのは非常に手間のかかる作業である。そこで、楽譜をページごとにサムネイル表示しコメントの投稿履歴などを一覧できるページの実装を考えている。通常サイズの楽譜ビューの前にひと画面挟むことで、楽曲全体を一望でき新着のコメントを探す手間も短縮することができる。

第7章　結論

本研究では、複数人数で行う演奏練習の音声ログを活用する手法として、楽譜との対応付け・可視化・議論への活用を提案した。楽譜との対応付けは、予め用意した楽曲構造データと演奏練習中のユーザ操作による演奏開始・停止の時刻データであるクリップデータから、音声ログのどの時点が楽譜上のどの箇所を演奏しているかを推定する。音声ログの可視化は、上述の楽譜への対応付け情報から楽曲を先頭から末尾まで一律な色彩の変化に充てることで、音声ログの視覚的な一覧性を大きく向上する。議論への活用は、先行研究において提案した楽譜上の特定の箇所にコメントすることで議論をすすめる手法に、音声ログの一部を切りだして含めることができる。これら 3 つの機能を備えた web ベースのシステム Co-Musicator を開発した。本システムでは演奏練習時に演奏練習の録音とユーザによる簡易な操作を行ってもらい、演奏練習以外の時間にユーザ各自が空いた時間を利用して、可視化された音声ログを聴取したり一部を切り出して議論の対象にしたりすることができる。本システムはいわゆる HTML5 を呼ばれる多くの新しい技術を利用して開発され、web のみで音声の取得と録音・音声ファイルのエンコードと保存・音声ログのインタラクティブな可視化を実現した。本研究の提案手法と本システムの有用性を評価するため、3 種類の評価実験を行った。音声ログの楽譜との対応付けの推定に関する実験では、概ね正しい推測ができたものの、楽曲構造データの利用や推定アルゴリズムについて新たな課題を発見した。音声ログの可視化に関する実験では、可視化による利便性の向上と可視化に対する需要について非常に高い評価を得ることができた。演奏練習と議論におけるシステムの試用実験では、システムの有用性と需要について特に高い評価が得られ、今後のシステムの発展に対して多くの示唆を得ることができた。今後の発展として、演奏練習時のユーザインタラクション設計の見直し、本システムへの要望の強い機能の実装、システムのサービスとしての公開などを検討している。

謝辞

本論文を執筆するにあたり、指導教員である高橋伸准教授には丁寧なご指導とご助言を賜りました。3 年間の研究生活において、研究の方向性に関するアドバイスやシステムの設計上のアイディアを頂きました。研究のプロフェッショナルとして、本質を見抜く眼差しと本質を突くお言葉で正しい導きを与えて下さりました。2 度の国際会議での登壇発表の経験や卒業論文・修士論文の執筆を果たすことができたのも、高橋先生のお力添えあってのことと思っております。ここに心よりの感謝の意を申し上げます。また、田中二郎教授、三末和男准教授、志築文太郎准教授、嵯峨智准教授、Simona Vasilache 助教にも、全体ゼミでの発表練習や春合宿を通して大変貴重なご意見・ご協力を頂きました。心より御礼申し上げます。

インタラクティブプログラミング研究室の皆様には日常生活の中で様々な示唆を頂きました。特にユビキタスチームの皆様や同期の皆様には、日常的な研究活動において数多くのご意見を頂きました。研究活動に付随する研究生活全般においても充実した時間を過ごすことができたのは、皆様のおかげであると確信しております。

そして、筑波大学管弦楽団の皆様は研究テーマや研究内容などを決めるにあたり力強い支えとなりました。音楽が好きで、音楽の中に身を置いてきた大学生活の後半戦、研究においても音楽に関わり続けることができたのはこの上ない幸福でした。聞き込み調査や被験者実験では多くの方に興味を持って時間を割いていただき、本当に恵まれた環境であったと実感しております。

本論文は、学問の点においても趣味の点においても、私の大学・大学院生活の集大成であります。この 6 年間に関わった全ての家族、先生方、友人に、心より感謝いたします。本当に、ありがとうございました。

参考文献

[1] T. Sakamoto, S. Takahashi and J. Tanaka. A Method for Discussing Musical Expression between Music Ensemble Players using A Web-Based System, Proceedings of the 15th International Conference on Human-Computer Interaction (HCI International2013), LNCS 8007, Pages 730-739, Las Vegas, USA, July 21-26, 2013.

[2] P. Bellini, P. Nesi, and M. B. Spinu. 2002. Cooperative visual manipulation of music notation. ACM Transactions on Computer-Human Interaction (TOCHI) Volume 9 Issue 3, September 2002, Pages 194-237.

[3] A. A. Sawchuk, E. Chew, R. Zimmermann, C. Papadopoulos, and C. Kyriakakis. 2003. From remote media immersion to Distributed Immersive Performance. In Proceedings of the 2003 ACM SIGMM workshop on Experiential telepresence (ETP '03). ACM, Pages 110-120.

[4] D. Akoumianakis, G. Vellis, I. Milolidakis, D. Kotsalis, and C. Alexandraki. 2008. Distributed collective practices in collaborative music performance. In Proceedings of the 3rd international conference on Digital Interactive Media in Entertainment and Arts (DIMEA '08). ACM, Pages 368-375.

[5] K. Hirata, S. Matsuda, K. Kaji and K. Nagao: Annotated music for retrieval, reproduction, and exchange. Proc. ICMC 2004 (2004).

[6] M. Goto, R. Neyama and Y. Muraoka: RMCP: Remote music control protocol ― design and applications ―. Proc. ICMC 1997, Pages 446―449 (1997).

[7] K. A. Lim and C. Raphael. 2009. InTune: a musician's intonation visualization system. In SIGGRAPH 2009: Talks (SIGGRAPH '09). Article 37, 1 page.

[8] M. Gurevich. 2006. JamSpace: a networked real-time collaborative music environment. In CHI '06 Extended Abstracts on Human Factors in Computing Systems (CHI EA '06). ACM, Pages 821-826.

[9] 松原 正樹, 諏訪 正樹, 斎藤 博昭. 2012. インタラクティブな楽譜色付けによるオーケストラスコア理解支援システム. 人工知能学会論文誌, Vol.27, No.5, Pages 281-295.

[10] piaScore : http://piascore.com/

[11] M. Goto: A real-time music scene description system: Predominant-F0 estimation for detecting melody and bass lines in real-world audio signals. Speech Commun. (2004).

[12] A.P. Klapuri. Multiple fundamental frequency estimation based on harmonicity and spectral smoothness. IEEE Trans. Speech Audio Process., 11, 804—816 (2003).

[13] M. Davy and S.J. Godsill. Bayesian harmonic models for musical signal analysis. Bayesian Stat., 7, 105—124 (2003).

[14] A.T. Cemgil, B. Kappen and D. Barber. Generative model based polyphonic music transcription. Proc. WASPAA 2003, pp. 181—184 (2003).

[15] H. Kameoka, T. Nishimoto and S. Sagayama. Extraction of multiple fundamental frequencies from polyphonic music using harmonic clustering. Proc. ICA 2004, pp. I-59—62 (2004).

[16] E.D. Scheirer. Tempo and beat analysis of acous- tic musical signals. J. Acoust. Soc. Am., 103, 588—601 (1998).

[17] M. Goto, K. Yoshii, H. Fujihara, M. Mauch, and T. Nakano. Songle: A Web Service for Active Music Listening Improved by User Contributions. In ISMIR (2011). Pages 311-316.

[18] J. Heer, F. B. Viegas, and M. Wattenberg. Voyagers and voyeurs: supporting asynchronous collaborative information visualization. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '07). ACM, Pages 1029-1038.

[19] J. Phalip, E. A. Edmonds, and D. Jean. Supporting remote creative collaboration in film scoring. In Proceedings of the seventh ACM conference on Creativity and cognition (C&C '09). ACM, Pages 211-220.

[20] U. Farooq, J. M. Carroll, and C. H. Ganoe. Supporting creativity in distributed scientific communities. In Proceedings of the 2005 international ACM SIGGROUP conference on Supporting group work (GROUP '05). ACM, Pages 217-226.

[21] J. J. Cadiz, A. Gupta, and J. Grudin. Using Web annotations for asynchronous collaboration around documents. In Proceedings of the 2000 ACM conference on Computer supported cooperative work (CSCW '00). ACM, Pages 309-318.

[22] S. E. Ellis and D. P. Groth. 2004. A collaborative annotation system for data visualization. In Proceedings of the working conference on Advanced visual interfaces (AVI '04). ACM, Pages 411-414.

[23] H. Lang and W. Minker. A collaborative web-based help-system. In Proceedings of the 2nd International Conference on Web Intelligence, Mining and Semantics (WIMS '12). ACM, Article 60, 5 pages.

[24] M. Goldman, G. Little, and R. C. Miller. Collabode: collaborative coding in the browser. In Proceedings of the 4th International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE '11). ACM, Pages 65-68.

[25] C. Marion and J. Jomier. Real-time collaborative scientific WebGL visualization with WebSocket. In Proceedings of the 17th International Conference on 3D Web Technology (Web3D '12). ACM, Pages 47-50.

[26] Ware, Colin. Information visualization: perception for design. Elsevier, 2013.