

Aibiki: 譜面の適応的自動スクロールによる三味線演奏支援

濱中 敬人^{1,a)} 坂本 大介¹ 五十嵐 健夫¹

概要: 日本の伝統楽器の一つ三味線は、演奏会においては暗譜演奏が慣例となっている。また、頻繁に変化する演奏速度や独特のフォーマットの譜面などの特徴は、暗譜のための練習を難しくしている。そこで本研究では、予め与えられた速度情報に則り譜面を自動スクロールし、演奏者が画面を見るだけで演奏できるシステムを提案する。加えて、本システムではマイク入力された演奏を元に譜本上で演奏箇所の探索を行い、演奏箇所が画面に収まるよう譜面スクロール速度を適応的に調整する。実装後に各機能の性能評価を行った結果、音高推定精度にはまだ改善の余地があるが、その結果を用いた楽譜探索には概ね成功し、ユーザの演奏に合わせた楽譜スクロールが可能になった。またユーザテストにおいては、参加者数は少ないもののこのシステムが演奏練習支援につながるという評価を得られた。

キーワード: 三味線, 演奏支援, 楽譜スクロール, アプリケーション

Aibiki: Supporting Shamisen Performance with Adaptive Automatic Score Scrolling

Abstract: We present a system for supporting users practice on playing the *Shamisen*, a Japanese traditional three-stringed musical instrument. *Shamisen* players are conventionally play music without looking at a musical score in a concert. Likewise, some characteristic problems of *Shamisen*, such as frequently changing tempo of music and a distinctive style to write musical scores, make it harder to practice for memorization. In this research, we created an application that scrolls a musical score automatically in defined speed, and this makes it easier to play the *Shamisen*. There is a gap between defined speed and actual playing speed, the application searches the score and fixes score scrolling speed. After the implementation, we evaluated the performance of application. Although there is room for the improvement in accuracy of pitch detection of lower pitches, the system succeeded in score searching and scrolling the score along the actual performance. The result of the user study showed that this system is useful for practice.

Keywords: *Shamisen*, Support playing the instruments, Score scrolling, Application

1. はじめに

世界には様々な楽器がある。これらは即興的に演奏されることもあるが、既に構成の決まった曲を演奏することも多い。そうした曲は楽譜として記録され、後世の演奏者はそれを読解すれば曲を演奏できる。一方で、これは現代音楽における常識であり、その範疇に入らない音楽も存在する。本研究ではそのような楽器の中でも特に三味線に注目する。三味線音楽には複数ジャンルがあるが、本研究では長唄を扱う。長唄の音楽的な特徴としては、

- (1) 現代音楽に採用されている五線譜のような楽譜ではなく、長唄独特の記法で楽譜が記録される。
 - (2) 楽曲は楽譜に全ての情報が記録されている訳ではなく、口承によって先生から弟子に引き継がれる情報もある。
 - (3) 演奏中に調弦が変わったり、演奏速度が頻繁に変わったりする。
 - (4) 長唄は演奏に10–40分を要する長い音楽だが、演奏会では伝統的に暗譜演奏が推奨される。
- など、三味線・長唄独特の問題点が存在する。特に、(2)については、長唄曲が多数作曲された江戸時代、その伝承方法は譜面ではなく、師匠から弟子への口伝のみであった[1]という背景がある。そのため伝承の過程でいくつもの演奏

¹ 東京大学
The University of Tokyo
^{a)} wobniar7@is.s.u-tokyo.ac.jp

バリエーションが生まれたほか、それらもほとんどが譜にならず、録音として存在するのみである。また、長唄譜は昭和時代に入ってから作られたもので、誤植も多い。

そこで、本研究では Aibiki と名付けた、長唄の暗譜練習支援および演奏支援のための Android タブレット向け譜面の適応的スクロールアプリケーションを提案する。Aibiki で扱う電子的な楽譜では、各小節に打ち込みの速度情報を付加し、Aibiki はそれによって譜をスクロールさせる。実際の演奏速度は必ずしも打ち込みの速度と一致しないため、Aibiki は演奏音から演奏箇所を推定し、楽譜の適切な場所が画面に収まるようにスクロール速度を調整する。また、数多くの演奏バリエーションに対応するために、Aibiki は簡単な譜面編集機能も備えている。

本研究では実際に Aibiki を実装し、その予備的な実験を行った。実験では初学者ではなく、数年程度の演奏経験がある演奏者に対して Aibiki を提示し、そのユーザビリティを評価した。結果、低音の音高推定精度にはまだ改善の余地があるが、その結果を用いた楽譜探索には概ね成功し、速度調整がない場合よりもユーザの演奏に合わせた楽譜スクロールが可能になった。ユーザテストにおいては、参加者数は少ないもののこのシステムが演奏練習の支援につながるという評価を得られた。本稿ではこの研究について報告する。

2. 関連研究

2.1 GIDA_U: 長唄譜記述用データ形式

長唄において、楽譜は譜と呼ばれることが多い。その形式には研精会譜、文化譜、家庭式譜などの種類があるが、本研究では研精会譜をとりあげる。先に挙げた譜は、いずれも五線譜の形式から遠いものである。こうした譜をデータとして記述するため、矢向は 2001 年に長唄の譜のデータ形式 GIDA_U を定義した [2] *1。このデータ形式はあくまで譜の表現・記録のためのものであったため、本研究では GIDA_U に対して以降の節で詳述する演奏速度情報などを付加する等の拡張を行った。

2.2 三味線類楽器の音高推定研究

FFT による音高推定は様々な楽器で行われている [4] が、三味線類楽器については多くは研究が行われてきていない。これは三味線類楽器の特徴として、和音の倍音成分が他の楽器よりも強く現れることが多く、また三味線そのものの個体差が大きいことが考えられる。

和宇慶らは 2001 年に三線の音高推定について研究し [5]、翌年その手法を元に三線の自動採譜装置を設計した [6]。この研究では、三線の音高推定にスペクトル間隔読み法、および倍音スペクトル削除法という二つの手法を提案してい

る。前者は、手法上単音の認識しか行えないが、長唄曲の大部分は単音から構成されているため、本研究でもこの方法で音高推定機構を実装する。三線は三味線の起源とされる楽器で、両者の構造には類似点が多く見られるが、三味線にみられる“さわり” *2 は三線にはみられない。

また、小坂谷らは 2009 年に津軽三味線の自動採譜装置を開発している [7]。この研究では、三味線の皮を取り除き各糸にピックアップマイクを直接取り付けすることで共鳴や雑音を減らしている。音高推定の精度は 70–80% と見積もられている。この研究ではハジキ、スクイといった三味線の特殊奏法 *3 の推定も行えるが、三味線を採譜専用で改造する必要がある点では本研究と異なる。

2.3 他楽器の演奏支援研究

演奏支援の発想に立った研究として、元川らのギターへの演奏支援装置 [8] や Weing らのピアノ学習装置 [9] があげられる。前者は、カメラで取得した映像から指の形を解析し、対応するコード（和音）の記号を Augmented Reality (AR) 技術でカメラ映像に重ね合わせ、和音の演奏を容易にする。後者は、ピアノ全体をスクリーンとし、弾くべき鍵盤や指遣いなどをプロジェクタで真上から投影することで、楽譜が読めない人でも演奏が可能になるシステムである。また、楽譜探索にも先行研究が存在する [10]。これは、MIDI 音源で与えられる入力から楽譜を探索し、対応する伴奏を MIDI で出力するシステムである。この研究では、弾き飛ばしや誤音の挿入に対処するために隠れマルコフモデル (HMM) を利用している。

場所を示す離散的な値である探索結果を利用して、速度を調整しつつ連続的に楽譜をスクロールするということが本研究の新しい点である *4。

3. ユーザインタフェース

3.1 長唄譜の表示形式

2.1 節で取りあげたように、今回実装するアプリケーションでは長唄譜の表現に「研精会譜」を用いる。図 1 に研精会譜の一例を示す。ここには主に三味線用の音符（図中 A）、唄の歌詞と音高（B）、調弦情報（C）と速度情報（D）が記載されている。三味線の音符は上から下、右から左へ記述される。速度に関する情報は“ノツテ”“シメル”など相対的で曖昧なものが多く、また実際の変化に対して記載量が少ない（実際にはもっと頻りに速度が変わる）ことが特徴である。今回、譜をスクロールさせるにあたり、音源

*2 二の糸または三の糸で特定の音高の高さの音を奏でると、弾いていない一の糸が共鳴する現象。三味線の三本の糸は、音の低い方から一、二、三の糸と名前がついている。

*3 ハジキは糸を撥で叩かず左手指で弾く奏法、スクイは撥で糸を下からすくう奏法で、ともに演奏に変化をつけるものである。

*4 楽譜探索の研究は数多くあるが、それを利用した譜面スクロールに焦点を当てた研究は、我々の知る限りはない。

*1 同形式は翌年菊地らによって改善が図られている [3]。

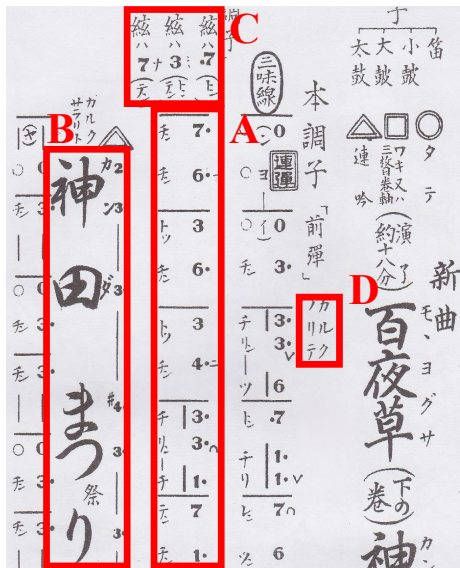


図 1 研精会譜の例 *5

Fig. 1 Kenseikai-Fu example

調弦 変更	減速			
	(<x0.7)	(<x0.8)	(<x0.9)	(<x1.0)
本調子 ○ 0	○ 0 BPM 120	○ 0 BPM 120	○ 0 BPM 120	○ 0 BPM 120
㊦ 6	㊦ 6 BPM 80	㊦ 6 BPM 90	㊦ 6 BPM 100	㊦ 6 BPM 110
ニ上り ㊦ 4	㊦ 4 BPM 120	㊦ 4 BPM 130	㊦ 4 BPM 140	㊦ 4 BPM 150
㊦ 3	㊦ 3 BPM 120	㊦ 3 BPM 130	㊦ 3 BPM 140	㊦ 3 BPM 150
加速				
	(>x1.0)	(>x1.1)	(>x1.2)	(>x1.3)
○ 0	○ 0 BPM 120	○ 0 BPM 120	○ 0 BPM 120	○ 0 BPM 120
㊦ 6	㊦ 6 BPM 120	㊦ 6 BPM 130	㊦ 6 BPM 140	㊦ 6 BPM 150
㊦ 4	㊦ 4 BPM 120	㊦ 4 BPM 130	㊦ 4 BPM 140	㊦ 4 BPM 150
㊦ 3	㊦ 3 BPM 120	㊦ 3 BPM 130	㊦ 3 BPM 140	㊦ 3 BPM 150

図 2 アプリケーション上での速度・調弦変化の視覚化

Fig. 2 Speed and tuning visualization

を参考におおよその速度変化情報を譜データの中に加えている。アプリケーション上では譜面を表示する際、前の小節から速度が変化する際には図 2 のように色をつけて目立つようにしている。

3.2 ユーザインタフェース設計

図 3 に実際のアプリケーションのスクリーンショットを示す。画面最上部には曲名が表示されており、ユーザはここをタップして読み込む譜を選択することができる*6。画面の右側を大きく占めているのが譜面表示エリアであり、自動スクロールする他、手動でも上下にスクロールさせることが可能である。譜面表示エリアの中央上寄りに位置す

*5 長唄新稽古本:節付音譜並三味線譜入(第五十七編)新曲百夜草(下の巻)神田祭・新曲春遊び©1960 吉住小十郎

*6 幾つかの譜データはあらかじめ入っているが、その他後述の編集機構でユーザが編集、保存した譜も読み込むことが可能である。

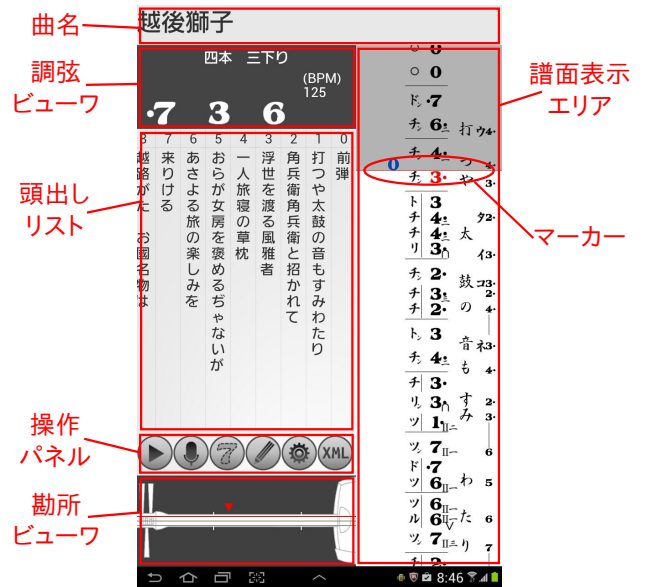


図 3 Aibiki のスクリーンショット

Fig. 3 Screen shot of Aibiki

る赤いラインは以降マーカーと呼ぶことにする。画面左側には上から順に調弦ビューワ、頭出しリスト、操作パネル、勘所ビューワがある。調弦ビューワは現在の曲のキーと調弦、速度を表している(調弦と速度は現在マーカーがある小節のもの)。ここをタップすれば曲のキーを変更可能である。頭出しリストは長い譜をスクロールする手間を省くために実装した。勘所ビューワは、現在マーカーがある音符を演奏するために三味線のどの糸のどこを押さえればよいかを表示する。

操作パネル上には 6 つのボタンがある。左から順に、つぎのような機能がある。

- 譜面の自動スクロールの開始/停止。
- マイク入力の有効/無効の切り替え。
- 暗譜補助機能の有効/無効の切り替え。有効にすると、マーカーより下の音符(つまりこれから演奏する音符)が見えなくなる。
- 譜面編集機能の有効/無効の切り替え。有効にすると、頭出しリストのある場所に図 4 のような譜面編集メニューが現れる。小節の挿入/削除や音符の変更などが行えるが、譜の形式が違うだけで既存の譜面編集アプリケーションとそれほど違いはないため、紙面の都合上この機能の詳細は省略する。
- 設定画面を開く。マイク認識の閾値調整などが行える。
- 現在表示している譜面データを XML 形式で書き出す。一番簡単な例として、例えばユーザは次のように Aibiki を使用できる:

- (1) 曲名表示欄をタップして譜を読み込む。
- (2) 曲のキーをセットする。デフォルトでは四本(三味線の最低音 C3)にセットされている。
- (3) マイク入力を有効にする。



図 4 譜面編集メニュー
Fig. 4 Score edit menu

(4) 自動スクロールを開始させ、三味線を演奏する。マイク入力有効時は、何か音が入力されると Aibiki はその音高を推定し結果を譜面表示エリアに出力する。

4. アプリケーションの実装

4.1 FFT を用いた音高推定

音高推定の実装には Java の FFT ライブラリ JTransforms[11] を利用する。今回は三味線の奏でうる最低音を A2 (110Hz) とし、次に高い音は A#2 (約 116.54Hz) であるため、最低でも 6Hz 刻み以上の周波数分解能があればよいことになる。今回はサンプリング周波数 22050Hz、データ長 8192 として約 2.69Hz の周波数分解能を得る*7。楽器の出す単音には普通基音 f_0 と倍音 $2f_0, 3f_0, \dots$ が含まれるが、三味線の出す音には基音のスペクトルがほとんど含まれず、代わりに倍音スペクトルが高次まで豊かに出る。したがって、周波数スペクトル上でこの倍音スペクトルの間隔を測れば基音が測定できることになる*8。Aibiki は入力音を FFT し続けるが、その結果から音高推定するのは入力音量が閾値を上回った場合だけである。

4.2 楽譜の適応的スクロール

音高推定によって、Aibiki は演奏された音符列を得る*9。そして、元々持っている楽譜の音符列と比較を行い、演奏箇所を探索する。比較には Smith-Waterman のアルゴリズムを用いる。演奏された音と譜面上の音との比較スコアは表 1 に示す。長唄曲には 2-3 小節単位の繰り返しが多数あるため、探索範囲はマーカーがある小節の前後 4 小節にと

*7 Google API Guide[12] によれば、全 Android 端末に対応するサンプリング周波数は 44100Hz のみだが、本研究で利用したタブレット (Galaxy Tab 10.1 LTE/Android 4.0.4) は 22050Hz での録音に対応している。

*8 周波数スペクトル上で、もっとも強いスペクトルの 10% 以上の強度があり、かつ前後 100Hz 内でもっとも強い周波数のみを倍音スペクトルと判断する。

*9 最大 12 個までで、それ以降は古い音から消去される。

表 1 演奏音と譜面上の音との比較スコア

Table 1 Matching score

評価	例 (演奏音 - 譜)	スコア*10
完全一致	C4 - C4	+20
半音違い、および 1 または 2 オクターブ違い	C4 - C3, C4 - C#4	+13
全音違い、および半音と 1 または 2 オクターブ違い	C4 - D4, C4 - C#3	+5
その他	C4 - F5, F3 - A4	-10

どめている。

楽譜探索の後、Aibiki はスクロール速度の調整を行う。各小節に打ち込みの速度で譜面をスクロールしつつ、探索結果がマーカー手前にあれば減速し、先にあれば加速するというのが基本の考え方である。画面上で、探索結果の始点、終点、マーカーの y 座標を y_b, y_e, y_m とすると、加減速を決めるパラメータ p は

$$p = \frac{d(y_e - y_m) - (y_b - y_m)}{y_e - y_b}$$

によって与えられる。 $p > 1.0$ の時は 1 段階加速、 $p < -0.5$ の時は 1 段階減速する。本実装では $d = 9$ とし、これはマーカーが探索結果を 9:1 に内分するとき $p = 0$ になる、すなわちそれを目指してスクロール速度調整が行われることを表す。Aibiki は速度の補正值 q *11 を持ち、実際のスクロール速度 v はこの q と打ち込みの速度 b を用いて

$$v = b(1 + q)$$

と計算される。 v, b の単位は BPM である。

4.3 Aibiki で使用している長唄譜のデータ構造

長唄譜は XML 形式で記述されている。2.1 節で述べたように、記述形式は先行研究の GIDA_U[2], [3] を元としているが、幾つか変更点がある：

- モジュールや要素の追加。デフォルトの演奏速度を示す要素を小節要素に付加している。元の長唄譜には具体的な速度情報はないので、プロの演奏音源を参考に今回独自に定義した。また、頭出しのためのモジュールも実装した。
- 幾つかの要素を属性に変更。GIDA_U においては、“楽譜”モジュール中に多数の要素があり、読み込みにかかる時間が長くなる。例えば、三味線の“音符”要素一つは“糸”“勘所”“口三味線”要素各一つを必ず含む。こうした要素をすべて“音符”要素の属性とすることで、読み込み時間を半分ほどに短縮した。

*10 スコアは実験的に定めたもので、今後最適化を必要とする。

*11 初期値は 0 で、加減速の度に値が -0.25 から 0.25 の間で 0.025 ずつ変化する。

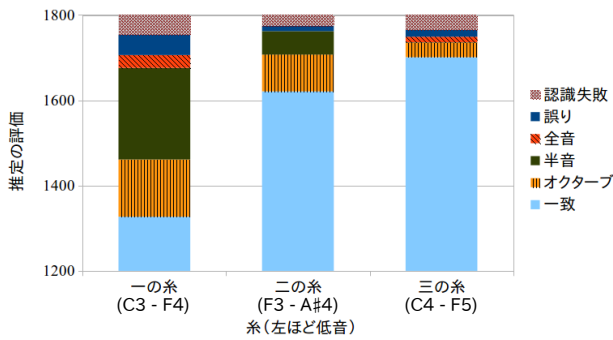


図 5 音高推定の評価結果 *12

Fig. 5 Evaluation of pitch detection

表 2 評価の詳細

Table 2 Scale of grading

評価	推定した音	例 (推定 - 正解)
一致	演奏とまったく同じ音高	C4 - C4
オクターブ	1または2オクターブ高い(低い)音高	F3 - F4
半音	半音高い(低い),あるいはその1オクターブ上(下)の音高	C4 - C#4
全音	全音高い(低い),あるいは半音違いの2オクターブ上(下)の音高	C4 - D4
誤り	上のどれにも当てはまらない	C4 - F5
認識失敗	演奏を認識できなかった	-

5. アプリケーションの評価

今回実装したアプリケーションについて、性能評価およびユーザテストによる評価を行った。

5.1 システムの性能評価

5.1.1 音高推定機構

三味線の3本の糸それぞれで、18種類の異なる音高を100回ずつ、計1800回鳴らし、その推定結果を集計した。演奏はBPM130前後(秒間2回強)で行った。集計結果を図5に示す。3つのグラフはそれぞれの糸に対応し、横軸は鳴らした音高(右ほど高い)、縦軸は推定結果の評価を示す。表2に各評価の詳細を示す。

この結果から、低い糸を出す糸ほど推定の精度が落ちていることが分かった。また半音違いの推定は低い音でよく起こっており、これはFFTの周波数分解能と低音域における隣の音との間隔が近いこと由来するとみられる。

5.1.2 適応的譜面スクロール機能

続いて、適応的譜面スクロール機能の性能評価を行った。速度調整ありの場合となしの場合で、同一の録音を入力と

*12 3本の糸ともに、1200回は“一致”が得られたので、その部分は視認性のために省略した。

し、譜面スクロールの様子を観察した。この結果、速度調整を行うと、行わない場合より演奏に追従したスクロールが可能になった。ただし「加速して演奏を追い抜き、減速して今度は遅れ、また加速し」という挙動を繰り返し、速度が不安定になってしまうこともあった。また、演奏が探索範囲外に出た時は速度調整不能となってしまうが、一般的な利用方法の範囲内であれば問題とならないと考えられる。

5.2 ユーザテスト

ユーザテストの参加者は大学生、および大学院生3名で、年齢は21-25歳(平均23歳)、男性1名女性2名。全員三味線演奏経験者である。テスト開始前に、スマートフォンの利用状況と三味線演奏歴についての事前アンケートを行った。参加者全員がスマートフォンを所持していたが、タブレット端末は所持していなかった*13。三味線演奏歴は2-5年(平均3.67年)であった。今回は適応的楽譜スクロールと譜面編集の2機能をテストし、所要時間は一人当たり75-90分であった。

5.2.1 適応的楽譜スクロール機能

適応的楽譜スクロール機能のテストの所要時間は最大60分であった。参加者にAibiki起動直後の状態のタブレット端末を渡し、その後は以下の手順でテストを行った:

- (1) 曲を選択する。曲(の抜粋)は5つ用意した。
- (2) Aibikiを説明なしでしばらく自由に操作してもらう。その後、必要ならば機能の説明(マイク入力の切り替え、譜面スクロールの開始についてのみ)を行う。
- (3) Aibikiに慣れるのも兼ねて、三味線を最大15分練習してもらう。説明していない機能を利用したり、操作について質問するのは参加者の自由である。
- (4) 暗譜練習(分からない所だけ譜を見て弾く)の形式で、Aibikiを使いつつ三味線を演奏してもらう。
- (5) 事後アンケートに回答してもらう。いくつかの回答についてはさらに口頭質問も行った。

事後アンケートの結果を表3に示す。“%”は好意的評価(7段階Likert尺度で5以上)の数を示す。結果として、3人全員が対話的譜面スクロールが必要だと感じたことが分かった。一方、(その速度で弾くのだとしても)スクロールが速すぎて目で追えないことがあったという意見もあり、紙の譜ではありえない“音符が動く”譜はすぐには馴染みにくいことが分かった。スクロール開始がボタン操作であるため、押してから三味線を弾き始めるまで時間がかかるといった意見も出た。また質問7にあるように、Aibikiは暗譜練習を補助しようと評価されたが、正しく弾けたか評価するシステムがあるとよいとの意見も得られた。

*13 1名Kindleを所持していたが、今回はタブレットとしてはカウントしなかった。

表 3 アンケート結果 (譜面スクロール機能)

Table 3 Results of questionnaire (score scroll function)

#	質問	平均	標準偏差	%
1	私は、対話的スクロール速度調整機能が必要だと思う。	6.00	0.82	3/3
2	私はスクロールが速すぎて、目で追えないことがあった。	4.00	2.16	2/3
3	私は、Aibiki は演奏中の場所を表示できていたと思う。	5.33	0.94	2/3
4	私は、速度変化を色で可視化したのは便利だと思う。	6.00	0.82	3/3
5	私は、Aibiki は直感的に操作できたと思う。	4.67	1.25	2/3
6	私は、Aibiki は三味線を演奏しながらでも操作できると思う。	4.67	1.70	1/3
7	私は、Aibiki は暗譜練習の助けになると思う。	6.33	0.94	3/3

表 4 アンケート結果 (譜面編集機能)

Table 4 Results of questionnaire (score editing function)

#	質問	平均	標準偏差	%
1	私はこの機能を直感的に利用できたと思う	5.00	1.63	2/3
2	私は自信を持って操作できたと思う	4.67	1.70	1/3
3	私はこの機能の利用に技術的サポートが必要だと思う	3.33	1.25	1/3
4	私はデジタル端末で譜を編集して記録できるのは便利だと思う	5.00	1.41	2/3
5	私はこの機能で長い譜を入力するのは時間がかかると思う	5.33	0.47	3/3

5.2.2 譜面編集機能

譜面編集機能のテストは、アプリケーション上で譜面を表示した状態のタブレット端末と、その譜の元になった紙の譜を渡し、参加者に譜データを修正してもらう、という形式で行い、所要時間は 15-30 分であった。アプリケーション上の譜には音高間違いが 1 つ、小節の抜けが 1 つ、補助記号間違いが 2 つの計 4 つの間違いがあり、この情報は最初に参加者に伝えた。結果、全員が 4 つの間違いを 5 分以内で発見、修正できた。事後アンケートの結果を表 4 に示す。3 人中 2 人がこの機能の操作を直感的と回答した。一方、予想できたことだが長い譜をこの機能で入力するには時間がかかるという意見が得られ、ユーザが任意の楽譜で Aibiki を使うためには、譜面の入力方式をもっと検討する必要があることが分かった。

6. おわりに

本研究では、三味線の暗譜学習支援を目的としたアプリケーションである Aibiki を提案し、その性能評価を行った。Aibiki の対話的スクロール速度調整は、打ち込みの速

度でのスクロールを基準としながらも、実際の演奏との間に生じる速度のずれに対応し、ページをめくる必要のないスムーズな演奏を可能にした。また暗譜練習の助けになるという目的については、ユーザーテストからは好意的な意見が得られた。アプリケーションの実装面においては、更なる機能拡充や音高推定・探索精度の向上などが今後の課題となると思われる。

謝辞 本研究を進めるにあたり、技術的アドバイスを頂いた柏野邦夫氏、永野秀尚氏 (NTT コミュニケーション科学基礎研究所)、およびユーザーテストの参加者 3 名 (伊藤篤義氏、中山みこと氏、および石田美緒氏) に感謝の意を表す。

参考文献

- [1] 杉浦 聡: 5 記譜法と口三味線, まるごと三味線の本 (田中悠美子, 野川美穂子, 配川美加, 編), 青弓社, chapter 5, pp. 232-239 (2009).
- [2] 矢向正人: XML を用いた長唄譜のデータ表現, 情報処理学会研究報告 [音楽情報科学], Vol. 2001, No. 103, pp. 43-48 (2001).
- [3] 菊地宏一郎, 矢向正人: GIDA-U (XML を用いた長唄譜のデータ形式): スキーマのモジュール化, 情報処理学会研究報告 [音楽情報科学], Vol. 2002, No. 123, pp. 15-21 (2002).
- [4] Lee, N., Duan, Z. and Smith III, J. O.: Excitation signal extraction for guitar tones, *International Computer Music Conference (ICMC)* (2007).
- [5] 和宇慶琢磨, 山城 毅, 渡久地實: 倍音スペクトル削除法による三線音の音高判別, 琉球大学工学部紀要, Vol. 61, pp. 85-89 (2001).
- [6] 和宇慶琢磨, 山城 毅, 渡久地實: 倍音スペクトル削除法を用いた三線演奏の自動採譜, 琉球大学工学部紀要, Vol. 63, pp. 57-62 (2002).
- [7] 小坂谷壽一, 小玉成人, 川守田礼子: 伝統音楽 (津軽三味線) 保存用自動採譜装置に関する研究, 電子情報通信学会技術研究報告. EA, 応用音響, Vol. 108, No. 491, pp. 37-42 (2009).
- [8] Motokawa, Y. and Saito, H.: Support System for Guitar Playing using Augmented Reality Display, *Mixed and Augmented Reality, 2006. ISMAR 2006. IEEE/ACM International Symposium on*, pp. 243-244 (2006).
- [9] Weing, M., Röhlig, A., Rogers, K., Gugenheimer, J., Schaub, F., Könings, B., Rukzio, E. and Weber, M.: P.I.A.N.O.: Enhancing Instrument Learning via Interactive Projected Augmentation, *Proceedings of the 2013 ACM conference on Pervasive and ubiquitous computing adjunct publication*.
- [10] 中村栄太, 武田晴登, 山本龍一, 齋藤康之, 酒向慎司, 嵯峨山茂樹: 任意箇所への弾き直し・弾き飛ばしを含む演奏に追従可能な楽譜追跡と自動伴奏, 情報処理学会論文誌, Vol. 54, No. 4, pp. 1338-1349 (2013).
- [11] Wendykier, P.: JTransforms - Piotr Wendykier, available from <https://sites.google.com/site/piotrwendykier/software/jtransforms> (accessed 2014-02-15).
- [12] Google: AudioRecord | Android Developers, available from <http://developer.android.com/reference/android/media/AudioRecord.html> (accessed 2014-02-15).