

## 視覚を用いない状況下での 高速楽曲探索インタフェースの設計と検証

青木 秀 憲<sup>†1</sup> 宮下 芳 明<sup>†2</sup>

今日、数百万曲規模の楽曲ライブラリを対象とした音楽探索インタフェースが多く提案されているが、一般には携帯音楽プレイヤーに数百曲しか保持しないという実情がある。この小規模なライブラリを対象として、特定楽曲を「視覚を用いずに」選択することは困難である。本論文で提案するインタフェース FFF は、ボタン押下間に楽曲のサビ冒頭部を連続的に高速再生し、ボタンの解放によって選曲を行うものである。サビ区間の幅や再生速度を調整し、ユーザの操作遅延を補正する機構を設けることにより、初めて使用するユーザでも従来の 5 倍速い探索が行えることを実験で明らかにした。また熟達によりこの探索速度はさらに向上することも実証した。

### Design and Evaluation of a Fast Nonvisual Interface for Searching Music

HIDENORI AOKI<sup>†1</sup> and HOMEI MIYASHITA<sup>†2</sup>

Nowadays many seeking system for music are developed, especially for million-song libraries, however in fact most people have several hundred songs in their mp3 players. Even in that kind of small song library it takes some time to find out the song without use of visual sense. In this paper, we introduce Far Faster Forward system, that plays beginning part of chorus section one right after the other in double speed, and the user can select the song by releasing the FFF button. We adjusted the time period of the chorus part, the speed, and the latency of the user, and finally subjects of the evaluation experiment succeeded in finding songs approximately five times faster than the traditional model. We also found that the more they practice the faster and easier they can browse their music library.

### 1. はじめに

Napster<sup>1)</sup> に代表される定額制音楽配信サービスが普及し始めている今日、数百万曲という規模の楽曲ライブラリから音楽を聴取する土壌が整いつつある。こうした背景を受け、膨大なライブラリからユーザの嗜好に適合する音楽を提供する研究が多く開発・提案されている。類似度に基づき楽曲検索や楽曲推薦を行う研究<sup>2)-4)</sup>、協調フィルタリングを応用した研究<sup>5),6)</sup> も多く、SD-Jukebox におけるミュージックソムリエ<sup>7)</sup> は楽曲の印象を 2 次元地図で表示しており、印象の観点で選曲する技術もすでに実用化されている。

インタフェース研究の領域でも、後藤らの Musiccream<sup>8)</sup> は、楽曲を蛇口からの流水として表現し、直感的な操作で所有楽曲に類似した楽曲を取得したり、過去に鑑賞した音楽を再発見したりできるインタフェースを実現している。堀内らは、類似フレーズを糸口に異なる楽曲を連続的に楽しめる再生システム Song Surfing<sup>9)</sup> を開発している。石井らによる IKESU<sup>10)</sup> は、旬な音楽を聴取するインタフェースを目標とし、未聴取の楽曲の混在比率を調整する。Pampalk らの MusicRainbow<sup>11)</sup> は、楽曲の類似度からアーティスト間距離を算出し、嗜好に合うアーティスト探索を実現している。筆者らも音漏れによって音楽との出会いを促進するヘッドホン Music Leak<sup>12)</sup> や、音楽が仮想生命体として実空間を放浪するノラ音漏れ<sup>13)</sup> を開発・提案してきた。

しかし現在、人々が携帯する音楽プレイヤーにはこの規模でのライブラリが保存されているわけではない。図 1 は 2008 年に実施された 5,393 人の男女を対象にした調査<sup>14)</sup> で、携帯音楽プレイヤーに何曲保存されているかを尋ねた結果である。100 曲以下が 55.5%、200 曲以下が 71.5%、500 曲以下で 84.8% を占めている。米国での調査<sup>15)</sup> でも、携帯音楽プレイヤーの容量上限が 1,000 曲で十分との結果が報告されている。

以上のように、前述の大規模ライブラリより 1 万倍規模が小さい数百曲のライブラリから効率的な楽曲探索を考えると、膨大な未知楽曲からの推薦とは目的が異なるため適さず、階層化されたフォルダ内を移動する方法や、アルバムジャケットをめくる iPod の CoverFlow のようなインタフェースが現状では適していることになる。また神原らは、CD ラックに収まる分量の楽曲数を大まかに管理・ブラウジングするためのシステム Melting-Sound<sup>16)</sup> を

<sup>†1</sup> 明治大学大学院理工学研究科基礎理工学専攻情報科学系

Computer Science Course, Graduate School of Science and Technology, Meiji University

<sup>†2</sup> 明治大学理工学部情報科学科

Department of Computer Science, Meiji University

開発している。

これらのシステムはすべて液晶ディスプレイ等での視覚情報提示と組み合わせて実現されているが、満員電車のように携帯音楽プレイヤーを取り出せない状況や、歩行中・運転中で液晶画面を見ることができない状況は多い。iPod Shuffle のように表示デバイスを省略し、少ない楽曲数のメモリ容量に抑え、ランダムな選曲によって楽曲を鑑賞するスタイルを提案するプレイヤーもある。視覚を用いずに楽曲を選択するシステムとしては、円周上に配置した曲を回転させ聴き分ける梅本らの一覽再生手法<sup>17)</sup> や、「音のみによる楽曲選択インタフェース」として提案された浜中らの Music Scope Headphones<sup>18)</sup> がある。

一覽再生手法は、提示された項目の選択によってタスクを確実に実行するインタフェースの一事例であり、また Music Scope Headphones は特定のパートの音を探しながら演奏を聴く鑑賞方法を提案するものであるが、文献 18) ではこの 2 システムを比較し、選曲方法としての有効性を評価する実験が行われている。10 曲中から好きな楽曲 1 曲を探索するタスクにおいて、一覽再生手法では平均して 18.9 秒要するのに対し、地磁気センサ・傾斜センサ・距離センサを搭載したヘッドホンシステムでは 15.6 秒に短縮している。筆者らは、一般的な携帯音楽プレイヤーに搭載されている次曲の冒頭部に移動するボタン（以下、次曲ボタン）を用い、同一条件での実験を行った。その結果は平均 28 秒であったが、10 曲ブラウジングするのに要する時間であるため、1 曲あたり 2.8 秒要している計算になる。次曲ボタンを押下 次曲のイントロを再生 曲を確認 再び次曲ボタンを押下...というプロセスに 3

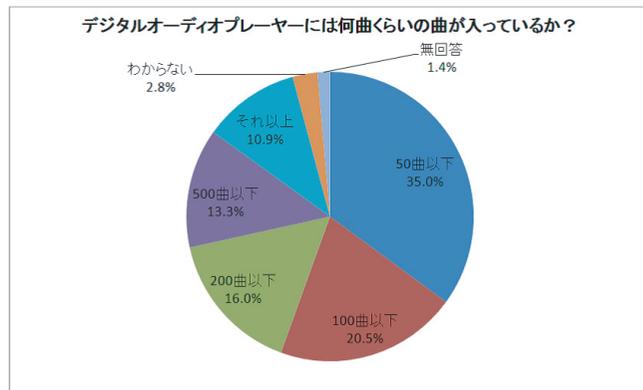


図 1 携帯音楽プレイヤーの楽曲数<sup>14)</sup>  
Fig. 1 The number of songs in MP3 player.

秒近くが費やされていることになる。つまり 200 曲のライブラリ全体をブラウジングする場合は 9 分以上要する計算になる。

実験で 18.9 秒と 15.6 秒での楽曲選択を実現した文献 17), 18) のシステムは、次曲ボタンを使う場合 (28 秒) と比較して各々 1.5 倍速, 1.8 倍速の高速化に成功したことになる。しかし、これを同様に 200 曲のライブラリに換算すると、1 曲を探し出すのに 5 分以上を要する計算となる。本論文では、こうした現状をふまえ、数百曲の既知楽曲ライブラリを対象として、視覚情報を用いずに、できる限り早く目的楽曲を探索するためのインタフェースを提案するものである。

## 2. 提案システム

図 2 は、提案システム Far Faster Forward (以下 FFF) の概念図である。ボタン押下の間、各楽曲のサビ部分が連続で高速再生され、ボタンが解放されるとユーザによる遅延を考慮して目的楽曲が再生される。

後藤の SmartMusicKIOSK<sup>19)</sup> で実装されている「サビ出しボタン」は、現在試聴中の楽曲のサビ冒頭部に移動するものであるが、FFF は試聴中の楽曲の「次の楽曲のサビ冒頭部」に移動し、高速にサビを再生し、一定時間経過するとさらに次曲のサビ冒頭部に移動、という循環を継続する。ボタン解放は楽曲決定を意味し、ユーザの遅延を考慮してその楽曲を推定、その楽曲の冒頭部に移動したうえで通常速度の再生を開始する。筆者らは、この FFF システムにおいてサビ区間を狭くし、再生速度を上げることで、ライブラリ全体をブ

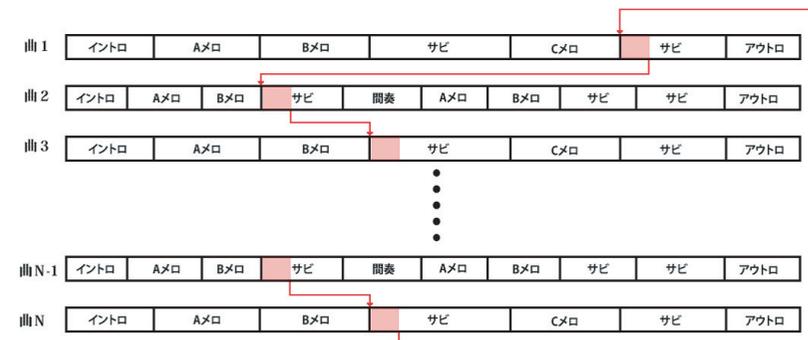


図 2 FFF システムの基本コンセプト  
Fig. 2 The basic concept of Far Faster Forward (FFF) System.

ラウジングする時間を短縮していくことができると考えた．よって本論文では，提案システムの有効性を示すだけでなく，サビ区間や再生速度の設定について，熟達プロセスも含めて検証する．

### 3. 実装

システムの実装には Cycling'74 社の Max/MSP を使用し，ソフトウェアとして Windows PC で動作させた（図 3 左）．各機能ボタンは専用のハードウェアは設けずに，キーボードのテンキーに割り当てて実現している．図 3 右は，インタラクション 2009 での発表<sup>20)</sup>で，アイマスクを着用し視覚を用いずにデモンストレーションを行っているところである．

FFF システムは 4.5 インチ液晶の PC (SONY VAIO Type UX) 上で動作するため，Bluetooth ヘッドホンで音楽を伝送し，無線テンキーユニット (Logicool Cordless Number Pad for Notebooks) と組み合わせることで，PC の存在を意識させることなく携帯プレイヤーとしての使用が可能になっている．iPod Touch の無線テンキー化アプリケーション (BALMUDA design NumberKey<sup>21)</sup>) によって iPod 上で動作しているように振る舞わせることも可能である．なお本論文の実験ではヘッドホンおよびキーボードともに有線接続で統一して実施している．

#### 3.1 サビ区間の初期値設定

FFF システムでサビ区間を狭めることで原理的に探索時間を短縮できるが，過度に行えばユーザはその楽曲を判別できず効率低下する．そこでこの区間をどこまで狭めると認識が困難となるかを被験者 3 名による予備実験で測定した．0.5 秒間，1 秒間，2 秒間の条件

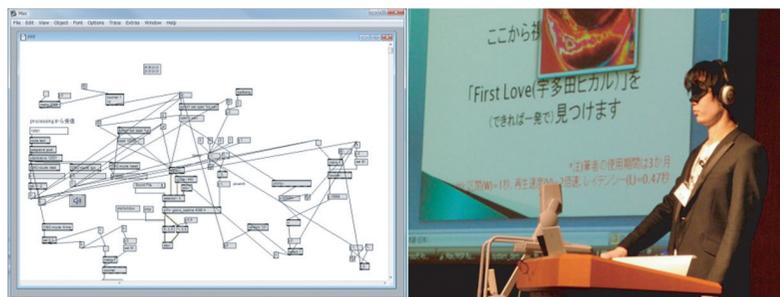


図 3 FFF システムのスクリーンショットとデモ風景

Fig.3 Screenshot of the FFF System and demo scene.

で 20 楽曲のサビ区間を連続再生するタスクを 5 回実施し，各々の楽曲が判別可能か確認させた．その結果，被験者や楽曲によっては 0.5 秒のときに判別できない事例があり，区間を 1 秒間，2 秒間とした場合は，確実に判別することができた．よって楽曲判別可能で最小なサビ区間の初期値を 1 秒間と設定した．

#### 3.2 再生速度の初期値設定

前節と同様に，再生速度を上げれば 1 曲あたりの再生時間を原理的にさらに短縮できる．そこで再生速度の最適値を設定するために予備実験を行った．

前節の予備実験と同じ被験者に対し，1 秒のサビ区間を 20 楽曲連続再生し，それぞれ 2 倍速，3 倍速，4 倍速で聴取させるタスクを 5 回実施した．なお高速再生を行うと音響ファイルのピッチは上がってしまうが，逆方向にピッチシフトすることで補正している．

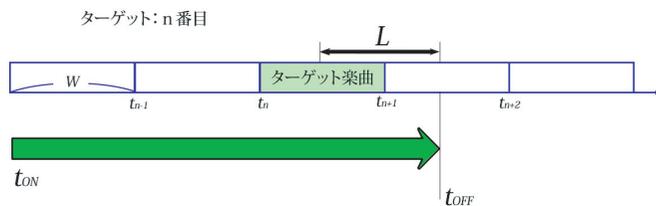
実験の結果，4 倍速では被験者や楽曲によって聞き取れない事例があった．また 3 倍速は聴取可能なものの被験者が疲労を訴えた．よって比較的負担が少ない 2 倍速を再生速度の初期値として採用した．

#### 3.3 遅延補正值の初期値設定

3.1, 3.2 節によって決定した初期値を用い，1 秒のサビ区間を連続で倍速再生すると，曲あたり 0.5 秒で次々と楽曲が流れ，目的楽曲を認識した瞬間にボタンを離しても適切に楽曲が選択できない現象が生じた．これは，認識してからボタンを離すまでの間に遅延が発生することに起因する．遅延時間がある程度一定であれば，補正を行うことで正確な楽曲選択が可能になるため，ミリ秒単位で遅延時間を測定できるプログラムを Cycling'74 社の Max/MSP によって作成し，前節と同一の被験者に対して，5 楽曲に対して認識と同時にボタンを解放するタスクを 5 回実施した．このときの遅延  $L$  の算出は以下のように行った．図 4 のようにサビ区間  $W$  で連続的に楽曲が提示され，目的楽曲の再生時刻  $t_n$  を，ボタン解放時刻を  $t_{OFF}$  とすると，遅延  $L$  は目的楽曲の中間地点から  $t_{OFF}$  までの距離，すなわち

$$L = t_{OFF} - t_n - \frac{W}{2}$$

とした．この計算により，曲ごとに遅延時間が増減しても  $\pm W/2$  以下ならば補正值 ( $t_{OFF} - L$ ) はターゲット楽曲の中に入る．予備実験結果では各個人によって 500 ms から 800 ms 程度までの差がみられたため，この遅延  $L$  については被験者ごとに最適化して用いるのがよいと考えた．なお，遅延  $L$  の被験者ごとの最小値と最大値の差は 100 ms の範囲に収まり，これが  $W$  (500 ms) よりも十分小さいことから，高精度で遅延補正ができると推量できる．

図 4 レイテンシ  $L$  の算出Fig. 4 Calculation of the latency ( $L$ ).

## 4. 実験 1：短期使用による速度差

### 4.1 実験内容

提案システム FFF を日常的な音楽鑑賞の道具として使用している筆者らの実感として熟達による速度向上が予想されたため、本論文における評価として、(1) 使い始めにおいて速い探索が可能であること、(2) 継続使用によって探索速度がさらに向上することの 2 点を実証する必要があると考えた。そのため、短期使用と長期使用に焦点を当てた実験をそれぞれ実施し、本章と次章で述べる。

まず本章では未経験者が初めて提案システムを用いる場合（短期使用）での有効性について検証するため、どれだけ素早く目的楽曲を探索可能か計測する実験を行う。実験タスクとしては 120 楽曲のライブラリから指定の 3 楽曲を見つけ出すまでの時間を計測した。楽曲数を 3 楽曲に設定したのは、1 曲だけの探索タスクの場合に、それがライブラリ内の前方に配置されるのと後方に配置されるのとで測定結果に大きな差が生じることによる。連続して 3 曲探索するタスクは日常的な音楽プレイヤーの操作としては不自然であるが、測定結果におけるばらつきを抑えられると考えた。

### 4.2 実験状況

実験に際し、まず前章の遅延計測と同様に、5 楽曲に対して認識と同時にボタンを解放するタスクを 5 回実施して被験者ごとの遅延平均を算出し、これを遅延補正值として登録した。

実験にあたっては、提案システムを含む 5 つのバリエーション（後述）をソフトウェアとして実装した。どのシステムについても練習は行わず、未経験者が初めて使ったときの効果を測定した。実験内容の把握やインターフェースの取扱いに対する慣れの差をなくすため、システムの使用順序は（被験者ごとに異なる）ランダムで行った。各システム使用後には、そのつど感想をインタビューした。5 つのシステムでの被験が終わるたびに 5 分程度の休憩

を入れた。

使用する楽曲ライブラリは、1990 年から 2007 年までの年間シングルヒットチャート<sup>22)</sup> から全被験者に既知の楽曲 120 曲を選び構築した。ライブラリは探索のたびにランダムな順序で並ぶよう制御されるが、乱数生成については同一のシードを用いているため被験者ごとに同じ順序となっている。サビの位置については SmartMusicKIOSK のような自動サビ検出機構は用いず、120 曲に対して手動でサビ開始時刻を登録して用いている。

この 120 楽曲のライブラリから 3 楽曲を見つけ出すタスクを 5 回行う実験とした。システムに 5 つのバリエーションがあることから、インタビューを含めると 1 人あたりの総被験時間はおよそ 3 時間となり、熟達プロセスが起こる前の探索速度測定としては適切な被験時間であると考えている。また、被験者の疲労の意味でもこれ以上長時間の実験は精度を下げると判断した。

被験者は 20 代の男性 6 名であり、ノイズキャンセリングヘッドホンで実験を行った。本実験に関してシステムから視覚的なフィードバックは行われぬ。探索すべき楽曲が見つからない場合のみ中止することも可能である旨のインストラクションを行っている。

探索に用いるシステムのバリエーションは、以下の 5 つである。

#### システム A（次曲ボタン）

通常の携帯音楽プレイヤーの「次曲ボタン」をモデルとしたものである。システム A を基準にすれば、一般的な携帯音楽プレイヤーと比較した速度向上率を見積もることが可能になる。「次曲ボタン」のほかに「再生」「停止」「早送り」「巻き戻し」「前曲ボタン」も用意した。

#### システム B（早送りボタン）

カセットテープのインターフェースをモデルとしたものである。120 楽曲が一行に連なりテープに記録されていると仮定して「早送り」「巻き戻し」「再生」「停止」による探索を行う。断片をつなぎ合わせた離散的なシークエンスである提案システム FFF に対し、楽曲を切り取らず接合したシークエンスが支持される可能性もあるためこのバリエーションを用意した。

#### システム C（次サビボタン）

SmartMusicKIOSK で実装されている「サビ出しボタン」と同様に、次曲のサビ冒頭にジャンプする「次サビボタン」を実装したものである。システム A の「次曲ボタン」と比較することで、楽曲冒頭とサビ部分のどちらがどの程度音響的なサムネイルとして有効に機能するかを検証することができる。「次サビボタン」以外には「(冒頭からの)再生」「停止」「前曲ボタン」を補助手段として用意した。

表 1 評価実験の結果 (単位: 秒)  
Table 1 Result of the evaluation experiment (sec).

		被験者 1	被験者 2	被験者 3	被験者 4	被験者 5	被験者 6	平均
次曲	A	518.8	193.3	216.0	264.1	352.3	399.2	323.8
早送り	B	give up						
次サビ	C	118.6	80.7	125.5	63.4	87.4	96.9	95.4
イントロ FFF	D	361.1	93.2	88.6	198.9	245.1	187.3	195.7
サビ FFF	E	64.5	54.7	94.1	56.1	59.1	67.9	66.1
遅延補正值 $L$		0.75	0.64	0.53	0.48	0.50	0.48	

#### システム D (FFF 冒頭部使用)

提案システム FFF の枠組みを用いて再生する区間をサビではなく楽曲の冒頭部分としたバージョンである。楽曲冒頭の無音部分を省いた 1 秒間を取得し、これを連続して倍速再生する。この機能以外に、「再生」「停止」「次曲ボタン」「前曲ボタン」を補助手段として用意した。遅延補正は行っている。

#### システム E (提案システム FFF)

提案システム FFF であり、楽曲のサビ区間 1 秒間を連続して次々と倍速再生し、ボタンを離すと再生を行う。遅延補正は行っている。FFF ボタン以外に「再生」「停止」「次曲ボタン」「前曲ボタン」を補助手段として用意した。

#### 4.3 実験結果

実験結果、および各被験者の遅延補正值を表 1 に示す。携帯音楽プレイヤーの「次曲ボタン」をモデルとしたシステム A は、システム B の中止を除けば、最も探索に時間を要しており、平均で 5 分程度であった。以後のシステムの評価についてはこの探索時間を基準として速度向上率を算出する。

カセットテープをモデルとしたシステム B については、全員が指定楽曲を見つけ出すことができず中止する結果となった。

「次サビボタン」のシステム C については、かなり探索時間が短くなっており (1%水準で有意)、「次曲ボタン」のシステム A と比較すると平均して 3.4 倍速を実現している。ただし被験者 3 は 1.7 倍速にとどまっている。

楽曲冒頭部を次々と高速提示するシステム D については、システム A と比較して 1.7 倍速ほどの効果をあげている (1%水準で有意)。

提案システムであるシステム E は、平均で最速となっており約 5 倍速を実現している (1%水準で有意)。特に被験者 1 は、8 分要していた探索が 1 分で完了する 8 倍速という大

幅な向上がみられた。ただし被験者 3 は 2.3 倍速にとどまっており、システム D のほうがやや速い結果となっている。

システム C と E の差 (すなわちボタン連打と押下継続というインタフェースの差) については 1%水準で有意、システム D と E の差 (楽曲冒頭部使用とサビ使用の差) は 5%水準で有意となっている。

#### 4.4 被験者の感想

被験者にこのシステムを便利だと思うか、使いたいと思うかを尋ねたところ、とても便利なのでぜひ使いたいという感想を得ることができた。筆者らが想定していたように満員電車の中で使いたいと述べた事例や、フットスイッチや歯を食いしばる圧力で操作して両手が自由な状態で選曲してみたいという提案もあった。提案システムに関して、多くの被験者からこのインタフェースが「楽しい」という感想が聞かれた。スロットマシンゲームのように感じられ、早く曲をみつきたい、思った場所で止めたい、という気持ちを起こさせるという。また、遅延補正の効果に驚いたという被験者もいた。

#### 4.5 考察

以上のように、本論文で提案する FFF システムは有意に探索の効率化を図ることに成功しており、平均 5 倍速で探索が行えることが明らかになった。システム D と E の差が開かなかった被験者 3 は、実験に用いたライブラリとふだん聴いている音楽がかなり近いと指摘したうえで、イントロだけでも十分に楽曲を判断できるという感想を述べていた。このように、楽曲冒頭部だけで曲の判別ができるほどライブラリに慣れている場合は、サビを用いずに楽曲冒頭部を用いるシステム D でも同じ効果を得ることができ、被験者 3 のデータはその結果であると考えられる。

この実験 1 は、初めてシステムを使った未経験者が、使い始めにおいてどれだけ速い探索が可能かを検証したものである。被験者を観察すると、次々と楽曲の一部が連続再生されるインタフェースに、ややとまどいながら実験を行っている様子が見えられた。補助手段として用意したボタンを使った試行錯誤があったため、熟達によって動作が洗練されれば、さらなる速度向上が期待できる。そこで、次章では長期使用による速度差について実験を実施する。

### 5. 実験 2: 長期使用による速度差

#### 5.1 実験内容

3 章においてサビ区間  $W$ , 再生速度  $V$ , 遅延  $L$  の 3 つのパラメータの初期値を  $W \quad V$

$L$  の順で決定したが、再生速度と秒数を組み合わせて測定する方法や、何倍速が適切か ( $V$ ) を決定した後にその速度による遅延時間 ( $L$ ) を測定し、その遅延時間をもとにサビ区間  $W$  を決定する方法も考えられる。また、この初期値設定プロセスにおける刻み幅は荒く、さらなる高速化のためにはより詳細な調整が必要だと思われる。

そこで本章では、3章で決定した  $W \cdot V \cdot L$  の値を起点とし、ユーザが自由に調整できるインターフェースを用いた長期実験によって、提案システムの熟達を検証するとともに、より理想的なパラメータ設定について調査することにした。またエラーレートの減少についても評価する。

## 5.2 実験条件

本実験においては実験 1 と異なる被験者 5 名を用いた。使用する楽曲ライブラリは実験 1 と同一の 120 曲で構成されるライブラリとしたが、実験 2 の被験者にとって未知楽曲が含まれていたため、まずは被験者にランダム再生で 1 週間聴取させ全曲を完全な既知楽曲とした。

実験に際しては、まず遅延補正値の取得について実験 1 と同様に計測を行った。この後システム A (次曲ボタン) とシステム E (提案システム) のみについて、実験 1 とまったく同じ条件での測定を行った。このとき、システムは指定ミスの回数 (探し出す曲ではない曲を 5 秒以上再生した場合を指定ミスとした)、次曲・前曲ボタンの押下回数も履歴として記録した。

次に、1 週間システム E を使用し、このライブラリを日常的に聴取させた。その間、サビ区間  $W$  と遅延補正値  $L$  は 0.1 秒刻みで、再生速度  $V$  は 0.1 倍速刻みで変更可能とし、ユーザに自由に変更させた。この変更履歴はファイルに記録される。この 1 週間においては、ノイズキャンセリングヘッドホンを用いず、ふだんの音楽試聴に使用しているヘッドホンやイヤホンを使用してもらうようにした。これによって日常的な使用に即したパラメータ設定を最終的に取得することを試みた。

1 週間後の実験で、システム E (提案システム) について実験 1 と同じ条件での測定を行い、指定ミスの回数、次曲・前曲ボタンの押下回数も記録した。このときの  $V$ 、 $L$ 、 $W$  は被験者が設定した状態のままで行った。

## 5.3 実験結果

実験結果を以下に示す。使用時間は被験者 1 が累計 10 時間、被験者 2 は累計 7 時間、被験者 3-5 が累計 10 時間であった。表 2 は、実験 1 と同様、楽曲の探索に要した時間であり、システム A (次曲ボタン) とシステム E (提案システム)、および 1 週間後のシステム

表 2 探索時間 (平均単位: 秒)  
Table 2 Searching time (average sec).

	被験者 1	被験者 2	被験者 3	被験者 4	被験者 5
次曲	321	225	220	318	278
FFF	56	59	93	74	54
遅延補正値 $L$ (初期値)	0.40	0.45	0.48	0.48	0.48
FFF (1 週間後)	60	49	87	35.2	33.3
遅延補正値 $L$ (最終値)	0.48	0.45	0.65	0.65	0.50

表 3 指定ミスの回数 (合計)  
Table 3 Number of mistakes (total).

	被験者 1	被験者 2	被験者 3	被験者 4	被験者 5
次曲	11	3	0	0	2
FFF	1	0	0	0	0
FFF (1 週間後)	0	0	0	0	0

表 4 次曲ボタンが押された回数 (合計)  
Table 4 Number of pushing forward button (total).

	被験者 1	被験者 2	被験者 3	被験者 4	被験者 5
FFF	4	0	3	0	0
FFF (1 週間後)	0	0	0	0	0

表 5 前曲ボタンが押された回数 (合計)  
Table 5 Number of pushing rewind button (total).

	被験者 1	被験者 2	被験者 3	被験者 4	被験者 5
FFF	1	11	2	0	0
FFF (1 週間後)	0	1	0	0	0

E を示している。表 3 は、その際の指定ミスの回数 (合計) であり、表 4 は次曲ボタンが押された回数 (合計)、表 5 は前曲ボタンが押された回数 (合計) を示している。

また、1 週間の継続使用の間に、全被験者がサビ区間  $W$ 、遅延補正値  $L$ 、再生速度  $V$  をどのように変更したかという履歴を示したのが以下の図 5、図 6、図 7 である。

以下、被験者ごとに結果を検証する。被験者 1 は、初回実験において 5.7 倍速で FFF による探索を行い、指定ミスが 1 回、次曲・前曲ボタンの押下回数は 5 回であった。これが 1 週間の使用後、探索時間はほぼ同様だが、指定ミスと次曲・前曲ボタンの押下回数がゼロとなっている。被験者 1 のパラメータ変更履歴を見てみると、 $W$  および  $L$  を初期状態から

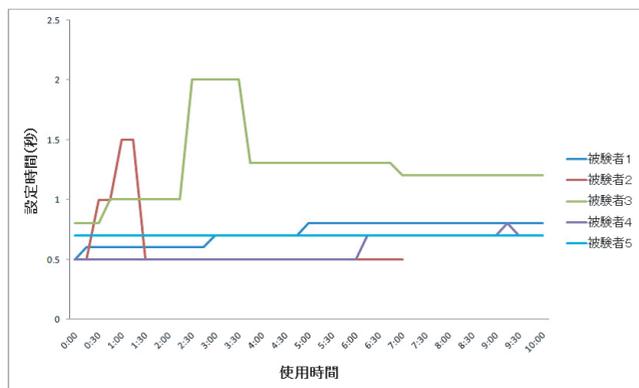


図 5 全被験者におけるサビ区間  $W$  の変更履歴  
Fig. 5 Archival record of  $W$  (Subject 1, 2, 3, 4, 5).

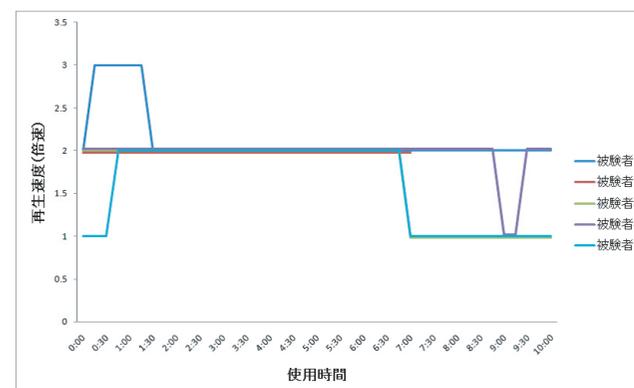


図 7 全被験者における再生速度  $V$  の変更履歴  
Fig. 7 Archival record of  $V$  (Subject 1, 2, 3, 4, 5).

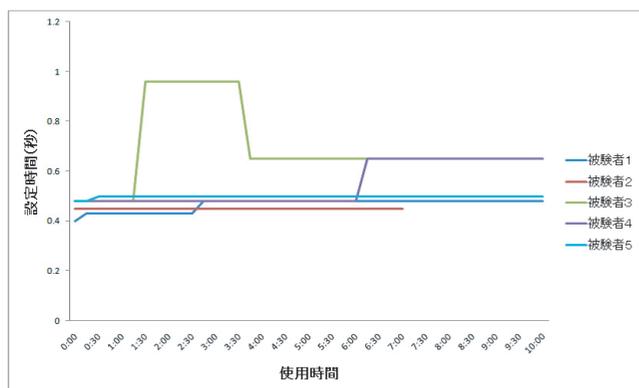


図 6 全被験者における遅延補正值  $L$  の変更履歴  
Fig. 6 Archival record of  $L$  (Subject 1, 2, 3, 4, 5).

長めに修正し、より確実に選択ができるように設定変更を行っている。

被験者 2 について見てみると、初回実験については 3.8 倍速で FFF による探索を行っており、前曲ボタンを 11 回も押下している。これが 1 週間の習熟後は 1 回の押下にとどまり、探索時間も 10 秒短縮し、4.6 倍速へと向上している。被験者 2 の  $W$  および  $L$  の変更履歴を見ると、一時期には幅  $W$  を大幅に上げているものの、最終的には初期設定と同じパ

ラメータに戻して設定している。

被験者 3 は、次曲ボタンを用いたシステム A での探索が速く、指定ミスもなかったため、初回実験についての FFF による速度向上が 2.4 倍速という計算になっているが、1 週間の使用によってそれを 2.5 倍速に向上させている。次曲ボタンを押す回数が 3 回から 0 回に、前曲ボタンを押す回数も 2 回から 0 回となっている。変更履歴を見てみると、様々な設定を試しながら最終的には  $W = 1.2$  秒、 $L = 0.65$  秒という設定に行き着いている。再生速度  $V$  については最終的に 1 倍速を選択している。

被験者 4 は、初回実験についての FFF による速度向上が 4.3 倍速であるが、1 週間の熟達によって 9.0 倍速を実現している。指定ミス・次曲ボタン・前曲ボタンの回数は初回も 1 週間後もゼロである。変更履歴を見ても大幅な修正は行っていない。

被験者 5 のデータでは、初回実験で 5.1 倍速、1 週間の熟達によって 8.3 倍速という速度向上がみられる。指定ミスも初回実験で 2 回あったものが 1 週間後の実験ではなくなっている。なおこの被験者については、履歴上も  $W$  と  $L$  はいっさい変更していないが、最終的な再生速度  $V$  については等速再生を選択している。

#### 5.4 考 察

以上の実験結果から、長期使用によって探索時間の短縮、指定ミスの減少、次曲・前曲ボタンの押下回数の減少がみられた。実験 1 で示した初回使用時の効果だけではなく、提案システム FFF は熟達によってさらなる速度向上を見込めることが示されたと考えている。

本実験からはサビ区間  $W$ ・遅延補正值  $L$  のパラメータについて各個人による調整が必要であった。また、一部の被験者にとっては等速の再生が好まれることも判明した。熟達プロセスを念頭に入れたインタフェースデザインとして、これらのパラメータ調整機構を検討したほうが良いのかもしれない。

被験者のインタビューから、運転中は使用不能である指摘を得たが、同様に、歩行中の使用も困難であると考えられる。

## 6. おわりに

本論文では、数百曲規模の楽曲ライブラリを対象として、視覚を用いずに楽曲を探索するためのインタフェースについて考察を行った。そして、ボタンを押下し続ける間に楽曲のサビ冒頭部が次々と高速再生され、ボタン解放によって選曲を行うシステム、Far Faster Forward を提案した。サビを切り出す幅や再生速度を調整し、さらにユーザの操作の遅延を補正する機構を設けることによって、初心者でも従来の 5 倍速の探索が行えるところまで効果を高めることに成功した。またこの高速化が熟達によってさらに向上することも調査し、長期使用実験によって探索時間の短縮、指定ミスの減少、次曲・前曲ボタンの押下回数の減少がみられ、さらなる速度向上がなされることを確認した。被験者の感想からは、向上心を刺激するゲーム性、およびエンタテインメント性に関わるものがみられた。この要素は、熟達プロセスを促進するうえで効果的なのではないかと筆者らは考えている。

提案システムでの選曲には一定の集中が必要であるため、もたらず疲労感についても今後の検討項目である。実験 2 の感想にもあるとおり、現状では運転中・歩行中の使用に耐えうるものではない。またアルバム単位・アーティスト単位の探索や階層化、ユーザごとに細かい調整ができるシステム面の改良、楽曲をよりみつけやすくする「順序」を導くアルゴリズムについても考察していきたい。

謝辞 本研究の一部は、明治大学科学技術研究所の重点研究課題「デジタルコンテンツ学の確立とその知見に基づく創造/享受メディアの研究開発」の助成を受けた。また、本論文の着想を得るにあたり、貴重なご意見をいただいた産業技術総合研究所/筑波大学大学院の後藤真孝氏に感謝いたします。

## 参 考 文 献

- 1) Napster. <http://www.napster.jp>
- 2) Pampalk, E.: A MATLAB toolbox to compute music similarity from audio, *Proc.*

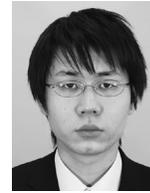
- International Conference on Music Information Retrieval (ISMIR2004)*, pp.254–257 (2004).
- 3) Soding, T. and Smeaton, A.F.: Evaluating a music information retrieval system – TREC style, *Proc. International Conference on Music Information Retrieval (ISMIR2002)*, pp.71–78 (2002).
- 4) Tzanetakis, G. and Cook, P.: Musical genre classification of audio signals, *IEEE Trans. Speech and Audio Processing*, Vol.10, pp.293–302 (2002).
- 5) Cohen, W.W. and Fan, W.: Webcollaborative filtering: Ecommending music by crawling the Web, *Proc. 9th international World Wide Web conference on Computer networks/the international journal of computer and telecommunications networking*, Vol.33, pp.685–698 (2000).
- 6) Uitdenbogerd, A. and van Schyndel, R.: A review of factors affecting music recommender success, *Proc. International Conference on Music Information Retrieval (ISMIR2002)*, pp.204–208 (2002).
- 7) <http://panasonic.jp/support/software/sdjb/prod/v6/v6le/ordinary/main05.html>
- 8) 後藤孝行, 後藤真孝: Musicream: 楽曲を流してくっつけて並べることのできる新たな音楽再生インタフェース, 日本ソフトウェア科学会第 12 回インタラクティブシステムとソフトウェアに関するワークショップ (WISS 2004) 論文集, pp.53–58 (2004).
- 9) 堀内直明, 園田俊行, 田中浩司, 田中淳一, 永沢秀哉, 莪山真一: Song Surfing: 類似フレーズで音楽ライブラリを散策する音楽再生システム, PIONEER R&D, Vol.17, No.2 (2007).
- 10) 石井隆昭, 望月有人, 星野剛史, 堀井洋一: IKESU: 「旬」な音楽を聴くための収集型ミュージックプレーヤー, 日本ソフトウェア科学会第 13 回インタラクティブシステムとソフトウェアに関するワークショップ (WISS 2005) 論文集 (2005).
- 11) Pampalk, E. and Goto, M.: MusicRainbow: A New User Interface to Discover Artists Using Audio-based Similarity and Web-based Labeling, *Proc. ISMIR International Conference on Music Information Retrieval* (2006).
- 12) 青木秀憲, 宮下芳明: Music Leak: 音漏れを聴く, 新しい音楽の楽しみ方, 第 15 回インタラクティブシステムとソフトウェアに関するワークショップ (WISS2007), pp.105–106 (2007).
- 13) 青木秀憲, 篠原祐樹, 宮下芳明: ノラ音漏れ, 第 16 回インタラクティブシステムとソフトウェアに関するワークショップ (WISS2008), pp.155–156 (2008).
- 14) [http://www.amazon.com/petit\\_pdf/enq-25.pdf](http://www.amazon.com/petit_pdf/enq-25.pdf)
- 15) <http://japan.cnet.com/news/tech/story/0,2000056025,20065658,00.htm>
- 16) 神原啓介, 安村通晃: MeltingSound: なめらかなオーディオブラウジング, ヒューマンインタフェースシンポジウム 2003 論文集, pp.817–820 (2003).
- 17) 梅本あずさ, 柴尾忠秀, 水口 充, 浦野直樹: 音声提示型インタフェースの実装と評価, 第 7 回インタラクティブシステムとソフトウェアに関するワークショップ (WISS'99),

pp.83-92 (1999).

- 18) 浜中雅俊, 李昇姫: サウンドスコープヘッドフォン, 日本バーチャルリアリティ学会論文誌, Vol.12, No.3, pp.295-304 (2007).
- 19) 後藤真孝: SmartMusicKIOSK: サビ出し機能付き音楽試聴機, 情報処理学会インタラクシオン 2003 論文集, pp.9-16 (2003).
- 20) 青木秀憲, 宮下芳明: Far Faster Forward: 視覚を用いない状況下での高速楽曲探索インタフェース, インタラクシオン 2009 論文集, pp.79-86 (2009).
- 21) <http://www.balmuda.com/jp/laboratory/numberkey/>
- 22) Music Rank ランキング総合ファンサイト.  
<http://www.musicvprogram.com/index-2.html>

(平成 21 年 4 月 21 日受付)

(平成 21 年 11 月 6 日採録)



青木 秀憲 (正会員)

1984 年生まれ. 2009 年明治大学大学院理工学研究科基礎理工学専攻情報科学系修了. 同年 KDDI (株) 入社. ヒューマンコンピュータインタラクシオンの研究に従事.



宮下 芳明 (正会員)

1976 年生まれ. 2006 年北陸先端科学技術大学院大学知識科学研究科博士後期課程修了. 同大学科学技術開発戦略センター研究員を経て, 2007 年より明治大学理工学部情報科学科に着任, 2009 年より准教授, 現在に至る. 博士 (知識科学).