

Far Faster Forward: 視覚を用いない状況下での高速楽曲探索インターフェース

青木 秀憲[†] 宮下 芳明^{††}

今日、数百万曲規模の膨大な楽曲ライブラリを対象とした未来型音楽探索インター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音楽が入っていないという実情がある。しかし、この小規模なライブラリから特定の楽曲を「視覚を用いずに」選び出すことは意外に難しい。本稿で提案するインターフェースは、ボタンを押し続けている間に楽曲のサビの冒頭部が次々と高速再生され、離すと選曲ができるものである。これは決して革新的といえるような発想ではなく、大規模なライブラリに対応できる汎用性を持つわけでもないが、サビを切り出す幅や再生速度をチューニングし、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り、初めて使用する人でも従来の5倍速い探索が行えることを実験で明らかにした。熟達によってこの探索速度はさらに向上すると考えられる。

Far Faster Forward: A Fast Nonvisual Interface for Searching Music

HIDENORI AOKI[†] and HOMEI MIYASHITA^{††}

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 think that the more they practice the faster and easier they can browse their music library.

1. はじめに

今日、Napster¹⁾のような定額制音楽配信サービスが普及し始めており、600万曲を越える膨大な楽曲ライブラリの中から自由に音楽を聴くことができる土壌が整いつつある。こうした膨大なライブラリからユーザの好みにあった音楽をみつける作業は、これまでの技術だけでは実現不可能であり、このような問題意識をもとに多くのシステムが提案されている。

類似度に基づき楽曲検索や楽曲推薦を行う研究^{2)~4)}、協調フィルタリングを応用した研究^{5),6)}は21世紀を

境に顕著に発表されるようになった。SD-Jukeboxのミュージックソムリエ⁷⁾のように、人がその楽曲を聴いた印象をマップで表示し、印象の観点で選曲ができるシステムもすでに実用化されている。

インタラクション分野においてもこのテーマの研究は多く、例えば後藤らのMusicream⁸⁾は、楽曲を蛇口から流れ出る水のように表現し、直感的な操作によって手持ちの楽曲に類似する楽曲を取得したり、過去に鑑賞していた音楽を再発見したりすることができるインターフェースを実現している。堀内らは、似ているフレーズを手がかりにして、多くの異なる曲を連続的に楽しむことができる再生システムSong Surfing⁹⁾を開発している。石井らによるIKESU¹⁰⁾は、旬な音楽を聴くためのインターフェースを目標としており、普段聞いていなかった楽曲の混在具合を、いけすの開閉のメタファーで調整することができる。EliasらのMusicRainbow¹¹⁾は、楽曲の類似度によってアーティスト間の距離を計算することで、好みのアーティスト

[†] 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

^{††} 明治大学大学院 理工学研究科 新領域創造専攻 デジタルコンテンツ系
Program in Digital Contents Studies, Programs in
Frontier Science and Innovation, Graduate School of
Science and Technology, Meiji University

を探するためのインタフェースを実現している。

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 Leak¹²⁾ や、音楽があたかも生命体のように実空間を自律的に放浪するノラ音漏れ¹³⁾ といったシステムを開発し、「音楽との出会い」を促進するインタフェースとして提案してきた。様々なシステムこそ乱立しているが、「数百万曲レベルの膨大なライブラリ」が未来型の音楽鑑賞において重要なキーワードであることは間違いないと考えている。

しかしその一方で、現在人々が持つ携帯音楽プレイヤーに目を向けてみると、それほど巨大なライブラリではないことに気づく。図 1 は、2008 年に 5393 人の男女を対象に行われた携帯音楽プレイヤーに関する調査で、携帯音楽プレイヤーに何曲くらい曲が入っているかを尋ねた結果である¹⁴⁾。これをみると、100 曲以下である人たちが 55.5%，200 曲以下が 71.5%，500 曲以下の人たちが 84.8% を占めるということがわかる。また、アメリカで行われた調査¹⁵⁾ でも、携帯音楽プレイヤーの容量は 1000 曲で十分であるという結果が出ている。

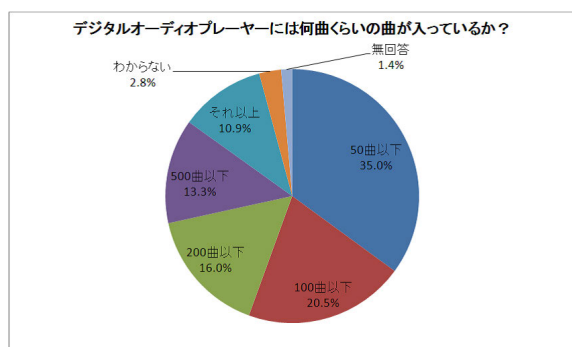


図 1 携帯音楽プレイヤーの楽曲数¹⁴⁾

Fig. 1 the Number of Songs in MP3 Player

このような「数百曲」という、先ほどより 1 万倍規模が小さいライブラリで効率的な楽曲探索を考えると、膨大な未知楽曲からの出会いを目指す前掲のシステムは目的が異なるためうまく機能しないと思われる。かといって、10 曲程度の CD 内ならまだしも、ボタンを数百回連打して楽曲を探すのも負担が大きい。現状で最も適しているのは、ジャンルのカテゴリやアーティスト名で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ということになる。また神原らは、CD ラックに収まる程度の楽曲を大まかに管理・ブラウジングするためのシステム Melting-Sound¹⁶⁾ を開発している。

これらのシステムは全て、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る。しかし、例えば満員電車などのように携帯音楽プレイヤーを取り出すことができない状況や、歩行中・運転中で液晶画面を見られない状況は多い。iPod Shuffle のように表示デバイスをなくし、少ない楽曲数しか入らないメモリ容量に抑え、そこから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。また、視覚的な探索インタフェースも著しく進化し使いやすくなっているが、聴覚コンテンツを探すのに視覚を用いる必然性があるわけではない。円周上に配置した曲を回転させ聞き分けるという、梅本らが提案する一覽再生手法¹⁷⁾ や、「音のみによる楽曲選択インタフェース」として提案されている浜中らの Music Scope Headphones¹⁸⁾ は、まさにこうした理想に立って開発されたシステムであるといえる。

本稿では、「数百曲のライブラリを対象として」「視覚情報を用いずに」「できる限り速く」目的の楽曲を探索するためのインタフェースを考えていきたい。

2. システム

2.1 目標速度

前掲文献¹⁸⁾ で、10 曲の中から好きな曲 1 曲を選択させるタスクの実験が行われている。それによると、梅本らによる前掲のインタフェース¹⁷⁾ では平均して 18.9 秒要してしまっているのに対し、地磁気センサ・傾斜センサ・距離センサを搭載した浜中らのヘッドホンシステムではこれを 15.6 秒と短縮している。

筆者らは、一般的な携帯音楽プレイヤーやメディア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、これと同じタスクを行って見たところ、平均で 28 秒の時間を要した。10 曲ブラウジングするのにかかる時間なので、1 曲あたり 2.8 秒もかかっている計算になる。次曲ボタンを押し、次の曲のイントロが再生され、曲を確認し、また次曲ボタンを押すという一回のプロセスには意外なほど多くの時間がかかっているのである。もちろん、お気に入りの CD アルバム内では、曲順を覚えて速く検索できるだろうが、数百曲規模ではそれが行われないと考えて単純計算すると、例えば 200 曲のライブラリ全体をブラウジングするのに 9 分以上かかる計算になってしまう（加えて 200 回ボタンを連打する疲労も加わる）。

実験で 18.9 秒と 15.6 秒での楽曲選択を実現した楽曲同時提示型のシステムは、次曲ボタンを使う場合（28 秒）と比較すればそれぞれ 1.5 倍速、1.8 倍速の

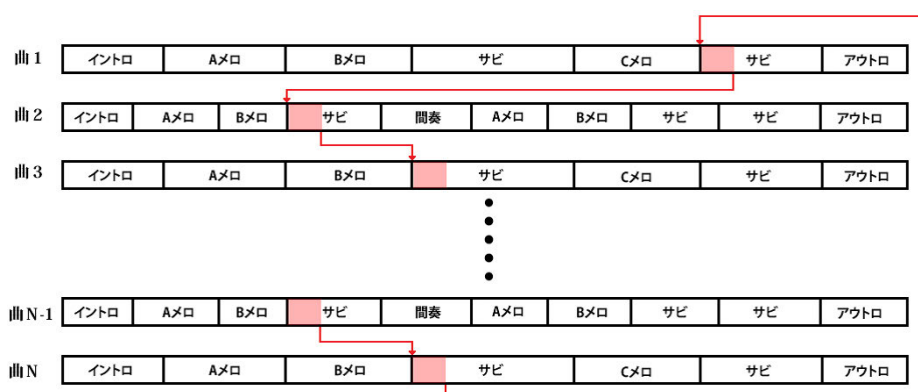


図 2 FFF システムの基本コンセプト

Fig. 2 the Basic Concept of Far Faster Forward (FFF) System

高速化に成功しているといえる。しかし、200 曲のライブラリに単純に換算すると、それでも 5 分以上かかることになり、まだ現実的な探索速度とはいえない。視覚情報を用いれば 30 秒ほどでみつけれられるはずなのに、「聴覚情報だけで」楽曲を探すのは段違いに難しいということなのだろう。

つまり、視覚情報を用いず、かつ実用的な速度で楽曲を探せるインターフェースを実現したいのであれば、少々速くなったぐらいで喜ぶことはできず、従来よりも何倍も速い実績を出さなければならないと考えられる。よって本稿では、数倍の速度向上が実現できることを目標とした。そのためには、探索にとって不必要な時間をできる限り削るようなシステムデザインを行わなければならないのである。

2.2 設計ポリシー

楽曲探索の観点から「次曲ボタン」よりも優れたボタンがあるとすれば、それは後藤の SmartMusicKIOSK¹⁹⁾ で実装されている「サビ出しボタン」であると考えられる。なぜなら、冒頭からその楽曲を聞くよりも、サビから聞いたときの方がより瞬時にその楽曲が何かを判別できると期待されるからである。よって、「サビ出しボタン」の亜種として、次曲のサビに飛んで再生する「次サビボタン」があれば、冒頭から再生する「次曲ボタン」より効果的に働くに違いない。

また、ボタンを毎回連打するのではなく、図 2 のようにボタンを押しっぱなしにしていれば次々とサビの一部が連続再生され、ボタンを離せばその楽曲が再生されればよいのかもしれない。本稿で提案する Far Faster Forward (以下 FFF) は、この設計ポリシーに基づいている。

ボタンを押しっぱなしで連続再生するのは、単にボタン連打の疲労を軽減するためだけではない。(1) サ

ビを聴く → (2) それが何の曲か (探している曲と一致するか) を判断する → (3) 次曲のサビを聞く、という段階的なプロセスではなく、もしかしたら人間は (2) の判断を行いながら (3) 次曲のサビを聞くことができるのではないかと仮定したからである。

たとえば通訳を行う人は、聞いた言葉を翻訳しながら、次に話される言葉を同時に聞いている。翻訳完了を待たずに次の情報が流入するという一見オーバーフローを起こしそうな状況にあっても、人間はパイプライン処理としてそれをこなすことができている。同時通訳のメカニズム解明は現在進行中の課題であり²⁰⁾、その並列処理にはワーキングメモリ²¹⁾ が関与していると考えられているようだが、いずれにせよ同時通訳で働いている脳のメカニズムを生かせば、マルチタスクな重なりを少し生じさせることもできるのであり、それによって連続するシングルタスクを高速に処理することが可能だと考えた。

このシステムにおいて、サビ区間をできる限り狭くし、再生速度をできる限り上げれば、さらなる高速化が期待できる。また、ボタンを離れたときに行き過ぎてしまうのを防ぐために、遅延補正を行って一発で目的の楽曲に到達できるよう工夫を行った。

2.3 サビ区間 W のチューニング

前節のシステムで高速化を行うためには、再生するサビ区間 W をできる限り狭くする必要がある。しかし、これを狭くしすぎると今度は何の曲かを判別できなくなり効率を下げってしまう。そこで、この区間をどこまで狭くすれば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 秒間のバリエーションで 20 曲の楽曲のサビ区間を連続再生し、それぞれの楽曲が判別できるかを確認してもらった。その結果、被験者・楽曲によっては 0.5 秒の

ときに判別できないものがいくつかあり、逆に区間を1秒間、2秒間とした場合は、ほぼ100%判別することができた。そこで、楽曲判別が可能で最小なサビ区間を、1秒間と設定することにした。

2.4 再生速度 V のチューニング

前節で設定したサビ区間を高速再生することができれば、1曲当たりの再生時間をさらに短くすることができる。そこで、この再生速度 V を最適化するために、予備実験を行った。前節で決定した1秒間ずつ取り出したサビ区間を連続再生するにあたり、20曲を2倍速、3倍速、4倍速で再生した。被験者は3名である。なお、高速再生すると通常はピッチが上がってしまうが、その分逆方向にピッチシフトして補正した。このため、ピッチが上がりすぎて聞き取りにくくなるようなことはない。

実験の結果、4倍速では被験者・楽曲によって聞き取れないケースがあった。また、3倍速は聞き取ることこそできるものの、被験者の疲労度合が大きいことを確認した。よって、比較的負担が少なかった2倍速を再生速度として採用することにした。

2.5 レイテンシー L のチューニング

以上のように、1秒間のサビ区間を連続で倍速再生するような最適化を行うと、1曲当たり0.5秒で次々と楽曲が流れていくこととなる。この状態でターゲットの楽曲を認識した瞬間にボタンを離すと、ひとつ行きすぎた楽曲が再生されてしまう現象が起こった。つまり、認識してからボタンを離すまでの間に遅延が発生するのである。この遅延時間がある程度一定であれば、その補正を行うことで正確な楽曲選択が可能になる。そこで、ミリ秒単位で遅延時間を測定できるプログラムを作成し、被験者3名に対して、楽曲を認識したと同時にボタンを離してもらって予備実験を5楽曲に対して行った。

レイテンシー L の算出は以下のように行った。図3のようにサビ区間 W で連続して曲が次々に提示され時刻 t_n でターゲットとなる楽曲が再生され、時刻 t_{OFF} でボタンが離されたとき、 L はターゲット楽曲が再生されているちょうど半分の地点から t_{OFF} までの距離、すなわち

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

とした。こうすることで、たとえ曲によって遅延時間が増減したとしても、それが $\pm W/2$ 以下であれば補正值 ($t_{OFF} - L$) はターゲット楽曲の中に入るわけである。

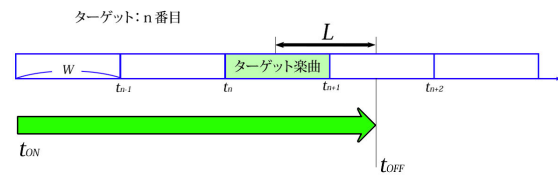


図3 レイテンシー L の算出
Fig.3 Calculation of the Latency (L)

予備実験結果をみると、各個人によって500msから800ms程度までの差があったため、この L については固定せず、被験者ごとに最適化して用いるのがよいと考えた。なお、5回の測定のうち最小値と最大値の差を求めると、どの被験者も100ms程度に収まっており、 W (500ms) よりも十分小さいため、高い精度で遅延補正ができると予想される。

3. 実 験

本システムの有効性を評価し、どれだけ素早く目的の楽曲を選択できるかどうかを調べる実験を行った。実験には1990年から2007年度の年間シングルヒットチャート²²⁾から120曲を選び、楽曲ライブラリを構築した。この120曲の中から3曲を指定し、それを3曲とも見つけるまでの時間を測定した。ライブラリは探索のたびにランダムな順序で並ぶように制御されている。1曲だけを探索するタスクの場合、それがライブラリの前のほうに配置されたときと後ろのほうに配置されたときでは測定結果にかなりの差が出てしまう。連続して3曲探索するというのは日常的な音楽プレイヤーの操作としてはやや不自然だが、このほうが測定結果におけるばらつきを抑えられると考えた。探索に用いるシステムには、以下の5パターンを用意した。

システム A (次曲ボタン)

このシステムは、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。このシステム A を基準にすれば、一般的な携帯音楽プレイヤーと比べてどれだけ速くなったかを見積もることができ、2章のように \times 倍速といった速度表現が可能になる。

携帯音楽プレイヤーをモデルとしているため、「次曲ボタン」の他に「再生」「停止」「早送り」「巻き戻し」「前曲ボタン」も用意されている。

システム B (早送りボタン)

このシステムは、カセットテープのインタフェースをモデルとしたものである。数百曲が一列に連なってテープに記録されているとして、その「早送り」「巻

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	give up	give up	give up	give up	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

き戻し」「再生」「停止」によって探索を行う。

このモデルを比較対象とした理由は、FFFのような断片をつなぎ合わせた離散的なシークエンスに対し、楽曲を切り取ることなくつないだ自然なシークエンスが支持される可能性もあるからである。

システム C (次サビボタン)

これは、2.2 節で述べた後藤の SmartMusicKIOSK[19] で実装されている「サビ出しボタン」のように、次の曲のサビにジャンプする「次サビボタン」をモデルとしたものである。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、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しているかを測定することができる。「次サビボタン」以外には、「(冒頭から)再生」「停止」「前曲ボタン」を補助手段として用意した。

システム D (FFF イントロ Ver.)

このシステムは、提案する FFF システムでありながら、再生する区間をサビではなく楽曲のイントロ部分としたバージョンである。冒頭の無音部分を省いたイントロの 1 秒間を取り出し、連続して倍速再生している。この 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。なお、遅延補正は行っている。

システム E (FFF サビ Ver.)

これは提案システム 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ものである。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。

実験に際しては、最初に 5 楽曲を用いたレイテンシー測定を行った。これはシステム D と E に反映される。また探索すべき楽曲がどうしても見つからない場合はギブアップ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。どのシステムについても練習はさせず、未経験者が初めて使ったときにどれだけのパフォーマンスが出るのかを測定することとした。実験内容の把握やインターフェースの取り扱いに対する慣れの差をなくすために、システムの使用順序はラン

ダムで行った。各システム使用後には、その感想をインタビューした。

システムの実装には Cycling'74 社の Max/MSP を使用し、Windows パソコンで動作させた。各機能ボタンはテンキーにアサインして実現している。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探しデータベース化したものを用いている。

なお、当然ながらどのシステムについても視覚的なフィードバックは行われない。被験者は 20 代男性 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。

4. 結果と考察

実験結果を表 1 に示す。まず、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、どの被験者にとっても (システム B のギブアップを除けば) 一番探索に時間がかかっており、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ギブアップしてしまっている。シークバーのような視覚的フィードバックがないため、このシステムだと自分がライブラリのどのあたりにいるのかもわからず、曲をいくつかスキップしてしまったのかどうかもわからない。そのため、こまめに早送りと再生を繰り返さなければならず、探索するモチベーションを著しく下げているようであった。

「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。やはり楽曲のイントロで判断するよりも、サビで判断するほうが早く楽曲を見極められるようだ。ただし被験者 3 は、1.7 倍速にとどまっている。この被験者は、今回のライブラリと普段聴いている音楽がかなり近いようで、「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。被験

者 2 も同様な理由でどのシステムでも他の被験者よりも比較的早く楽曲をみつけている。

イントロ部分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 倍速ほどのパフォーマンスをあげている (1%水準で有意)。この数値は、2 章の換算にならえば、Music Scope Headphones のような楽曲同時提示型システムとほぼ同等な速度向上率である。

最後に、提案システムであるシステム E だが、平均では最速となっており、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 分かかっていた探索が 1 分で済んでしまう 8 倍速という大幅な向上を見せており、「すぐ楽曲がみつけれられてとても嬉しい」という感想を述べていた。システム C のときに「イントロだけで十分楽曲を判別できる」と述べていた被験者 3 は、ここでもサビ出しによる効果が薄く、結果としてシステム D とシステム E がほぼ同等な速度となった。システム C と E の差 (すなわちボタンを押しっぱなしにすると連続して提示することの差) は 1%水準で有意、システム D と E の差 (イントロではなくサビを用いることによる差) は 5%水準で有意となっている。

以上のように、本稿で提案する FFF システムは有意に探索の効率化をはか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。特筆すべきなのは、これが初めてシステムを使った未経験者のデータであるということである。彼らは、次々と楽曲のサビが 0.5 秒ずつ連続再生されるインタフェースにややとまどいつつ実験を行っており、ときには補助ボタンを使って試行錯誤しながら目的の楽曲に到達している。しかし、これが熟達してくれば動作もより効率的になるので、5 倍速などと言わず、10 倍速近い向上も望めるのではないかと考えている。現に、本稿の第一筆者は普段の音楽鑑賞でもこのインタフェースを使用しているが、補助キーを使うことはなく、ワンアクションでほぼ確実に目的の楽曲に到達することができるようになってきている。120 曲のライブラリから 1 曲探すタスクなら、期待値として 30 秒で探索できる計算になる。これくらいの速度だと、「視覚を用いる楽曲探索」よりも早く楽曲を見つけれられるケースも出てくるだろう。

本実験ではレイテンシー L のみ個人ごとに最適化した。再生速度 V や提示区間 W をさらに細かく調整する機構を設ければ、ユーザにもなじみ、使い込むほど洗練されていくインタフェースになるに違いないと考えている。この方法だと、疲れているときには

再生速度を下げるなど、よりユーザにやさしいインタフェースに発展させることもできる。これまで速度にこだわって議論を進めてきたが、あくまで楽曲探索は音楽鑑賞のための手段であって、必要以上にユーザに負担をかけない柔軟性を持たせることはその解決につながると考えている。

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、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被験者もいれば、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。この改良については今後検討していきたいと思っている。

「次々と思いが蘇る」という変わった感想もあった。実験で用いたライブラリの特性も大きく影響していると思われるが、様々な音楽が走馬燈のように流れていくのを聞いていると、たしかにそのような感覚を理解することはできる。美崎らの「記憶する住宅²³⁾」は、個人が撮りためた膨大な写真ライブラリを住宅内でスライドショーしつづけることによって記憶想起活動を活性化するシステムであるが、本システムも似たような効果をもっているのかもしれない。

最後になったが、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ことを述べておきたい。実はこの感想は、速度向上だけを念頭においてデザインを行ってきた筆者らにとっては全く予想していないものであった。曰く、このインタフェースが「音のスロットマシン」のように感じられ、早く曲をみつめてやろう、思った場所でぴたり止めてやろう、という気持ちになるようで、はからずもエンタテインメント性が高いシステムになっているようだ。

また、レイテンシー補正の効果によって、ユーザがここだと思った地点でちゃんと楽曲が再生されるため、それに驚いたという被験者もいた。つまり本システムは「(意外に) 思い通りに止められるスロットマシン」という見方をすることができ、そこに楽しさを感じられたのかもしれない。

「熟達するほど速く探索が行えるインタフェース」に向上心を刺激するゲーム性とエンタテインメント性が加わっていたというのは、筆者らにとって願ってもいないことであった。この二つの要素の相乗効果によって、筆者らが願っていた「高速化」はさらに加速するに違いないからである。

5. おわりに

本稿では、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、ボタンを押し続けている間に楽曲のサビ冒頭部が次々と高速再生され、離すと選曲ができるシステム、Far Faster Forwardを提案した。このシステムデザインは、決して革新的といえるような発想ではないし、大規模なライブラリに対応できる汎用性を持つわけでもない。しかし、サビを切り出す幅や再生速度をチューニングし、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5倍速い探索が行えるところまで効果を高めることに成功した。この高速化は熟達によってさらに向上すると考えられるが、システムに内在するゲーム性がこれに貢献できるのではないかと展望した。

本稿では、サビ区間 W 、再生速度 V 、レイテンシー L の3つのパラメータを $W \rightarrow V \rightarrow L$ 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、何倍速が適切であるか (V) を決定した後にその速度による遅延時間 (L) を測定し、その遅延時間を元にサビ区間 W を決定する方法も考えられる。

そこで追加実験では、本稿で決定した $W \cdot V \cdot L$ の値を初期設定値ととらえ、それらをユーザが自由に調整できるインタフェースを用いた長期実験によって、熟達によってどのようにパラメータが変化していくかを調査する予定である。実験の際には、ノイズキャンセリングヘッドホンを用いず、通勤・通学といった普段の音楽試聴スタイルの中で調整してもらうことでより現実に即したパラメータ設定を取得しようと考えている。また本稿では速度中心に議論してきたが、エラーレートの減少についても評価していきたい。

また本稿で行った実験において各システム間に十分な水準の有意差を導出したと考えているが、被験者のバックグラウンドや探索曲の知名度も鑑みた検証をさらに行いたいと考えている。

システムの改良点としては、足や歯による操作によって「視覚も手も用いない」楽曲探索インタフェースへの可能性も模索したい。アルバム単位・アーティスト単位の探索への対応や、ジャンルやアーティストを読み上げ、それらを選択して階層的に下がっていくインタフェースとの比較や連携についても考えていきたい。さらに、ユーザごとに細かなチューニング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と考えている。今回の実験で発見された副次的効

果であるエンタテインメント性や向上心の刺激についても、それをより高めるようなデザインを考えていきたい。まだまだ長い道のりであることは承知しているが、いつの日かは視覚を用いる楽曲探索インタフェースを越えることを目標として積極的な研究開発を行っていきたい。

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、本稿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) Elias Pampalk: A MATLAB toolbox to compute music similarity from audio, Proceedings of International Conference on Music Information Retrieval (ISMIR2004), pp. 254-257, (2004).
- 3) Thomas Soding and Alan. F. Smeaton: Evaluating a music information retrieval system - TREC style, Proceedings of International Conference on Music Information Retrieval (ISMIR2002), pp. 71-78, (2002).
- 4) George Tzanetakis and Perry Cook: Musical genre classification of audio signals, IEEE Transactions on Speech and Audio Processing, Vol. 10, pp. 293-302, (2002).
- 5) William W. Cohen and Wei Fan: Webcollaborative filtering: ecommending music by crawling the Web, Proceedings of the 9th international World Wide Web conference on Computer networks/the international journal of computer and telecommunications netowrking, Vol. 33, pp. 685-698, (2000).
- 6) Alexandra Uitdenbogerd and Ron van Schyndel: A review of factors affecting music recommender success, Proceedings of 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, 藪田俊行, 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 RD, 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) Elias Pampalk Masataka Goto: MusicRainbow: A New User Interface to Discover Artists Using Audio-based Similarity and Web-based Labeling, in the Proceedings of the 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.amazonet.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) 船山仲他, 水野 的, 荻阪 満里子. 研究資料としての同時通訳データ, 日本言語学会 第128回大会ワークショップ, (2004).
 - 21) 荻阪満里子. ワーキングメモリ-脳のメモ帳. 新曜社, (2002).
 - 22) Music Rank ランキング総合ファンサイト <http://www.musicvprogram.com/index-2.html>
 - 23) 美崎薫, 河野恭之, 「記憶する住宅」～55万枚のデジタルスキャン画像の常時スライドショー・ブラウジングによる過去記憶の甦りの実際, 情報処理学会インタラクシオン2004 論文集, pp.129-136, (2004).