

携帯型音楽プレイヤーのためのコンテンツ再生・配信方式

井上 亮文 * 天野 玲奈 * 備瀬 翔平 * 市村 哲 * 松下 温 *

e-mail: akifumi@cs.teu.ac.jp

本稿ではDJ(Disc Jockey)の知識を利用した音楽の再生方法と、それに基づくデータ配信サービスについて述べる。通常の再生方法では1曲単位で音楽を再生する。これに対しDJは2枚のレコードを巧みに操り、2つの曲を混ぜながら連続的に再生する。まず、実際のDJがミキシングした375曲の分析を行い、どのように曲をつなげるのかの分析を行った。次に、分析の結果得られた知識・技術を組み込んだプロトタイプシステムRAIS-Mixerを実装した。被験者に自動ミキシングした曲を聴いてもらいアンケートをとったところ、実際のDJに近い「Cool」なミキシングも実現できることがわかった。

Playback and Distribution Methods for Digital Audio Players

Akifumi Inoue *, Reina Amano *, Sho-hei Bise *, Satoshi Ichimura *
and Yutaka Matsushita *

In this paper, new playback and distribution methods for digital audio players are presented. Normal method plays the track one by one. On the other hand, a DJ mixes two separate tracks into one consecutive track. First, an analysis was conducted on 375 tracks to obtain how they mixed those tracks. Then a prototype system “RAIS-Mixer” was developed based on the knowledge by the analysis. From the experimental results, the RAIS-Mixer was confirmed to be able to mix tracks like actual DJs.

1 はじめに

mp3に代表される音楽データの圧縮技術はここ数年で急速に普及してきた。同時に、音楽鑑賞をささえる記憶媒体はレコード、カセットテープ、CD、MDと移り変わってきた。最近ではフラッシュメモリやHDDなどもますます小型・大容量化している。この相乗効果により、ユーザは大量の曲を持ち運んで聴くことができるようになった。いわば、いつでもどこでも大量の音楽を視聴可能な環境にある。

ここで、プレイヤーの機能に注目してみると、レコードやテープといったアナログ媒体の時代では曲の再生、停止、早送り、巻き戻しのみが可能であった。CDやMDのようなデジタル媒体になってからは、これらに加えてイントロ再生、瞬時の頭だし、ランダム再生などが可能になった。

こうしてみると、CDが登場して以降は記録媒体やオーディオ機器のサイズこそ大きな変化を遂げている

が、プレイヤーの再生機能にはほとんどといってよいほど変化が見られない。状況に適した曲の推薦 [1] など、曲順や選択に関してはいくつかのアプローチがあるが、コンテンツそのものに手を加える再生手法に関しては、音質改善などのイコライザを除いてほとんど考慮されていないのが現状である。

本稿では、音楽に接する時間が増えつつあるユーザに対して新たな楽しみを提供することを目的とし、DJ(Disc Jockey)の知識・技術に注目した音楽再生方法を提案する。また、この再生方法を実現するための配信サービスのあり方についても言及する。

以下、2章では現状の再生方法、3章では提案方式、4章ではプロトタイプシステムRAIS-Mixer、5章では評価と考察について述べ、6章をまとめとする。

2 関連技術

2.1 再生方法

記録媒体がデジタル化されたことで、アナログ時代にはできなかった再生方法がいくつか登場している。代

*東京工科大学 コンピュータサイエンス学部
School of Computer Science,
Tokyo University of Technology

表的なものとして、プレイリスト再生とクロスフェードエフェクトが挙げられる。

プレイリスト再生は、曲名がリストされたファイル（プレイリスト）をプレイヤーが読み込み、その順番にしたがって再生を行う。ユーザはこのリストを自由に編集することが出来る。テープなどでも自分の好きな曲を好きな順序で編集することができたが、プレイリストはエディタやマウスによるドラッグで気軽に編集できるなど変更の手間が容易である。プレイリストによる通常の連続再生の様子を図1に示す。図上部に現在再生中の曲 T_n 、次に再生予定の曲 T_{n+1} の全体を示す。図下部は両者の切り替わり部分を、横軸に時間、縦軸にボリュームをとって拡大したものである。プレイリスト再生は単に2つの曲を続けて再生しているだけであり、 T_n の末尾（曲の終わり）と T_{n+1} の冒頭（次曲のはじまり）の合間に若干の無音区間が存在する場合もある。

一方クロスフェードとは、連続して再生される2曲のボリュームを同時に操作するエフェクトである。図2にクロスフェードによる連続再生の様子を示す。特徴として、 T_n の末端数秒前からボリュームをフェードアウトさせると同時に、 T_{n+1} の冒頭からボリュームをフェードインさせている。通常の連続再生と異なり、両者が重なり合いながら（overlapping time）入れ替わる。Windows Media Player[2] や iTunes[3] といった代表的な再生ソフトにはクロスフェード再生機能が標準で備わっており、重ね合わせ時間を何秒にするかなどの設定も可能である。

クロスフェードは本来はバラバラの2曲に一体感をもたらす一方で弊害もある。楽曲の構成を意識せず、末端と冒頭を機械的にミキシングしているため、 T_{n+1} の冒頭部分にMCや重要なメロディラインが録音されている場合、その部分が聴き取れなくなってしまう。雰囲気まったく異なる2曲をクロスフェードすると、視聴者は違和感を覚えることもある。

2.2 配信方法

音楽コンテンツに関する配信サービスとしては、曲に関連するメタデータを配信するものと、コンテンツそのものを配信するものに分かれる。

メタデータを配信するものの代表に CDDDB サービス[4]が挙げられる。CDDDBには世界中のCDのアルバムタイトルと各トラックの曲名が記録されている。音楽データをエンコードしようとPCにCDを挿入するとプレイヤー上にタイトルと曲名が自動で表示されるが、

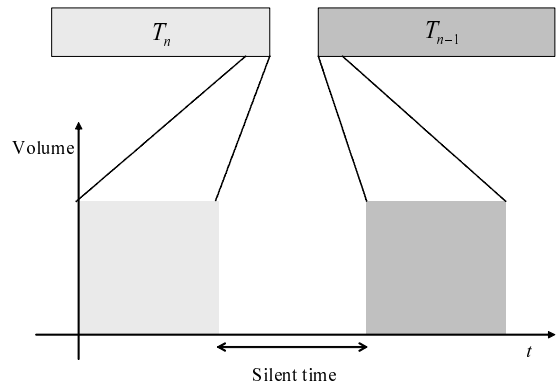


図1: プレイリストによる通常の連続再生

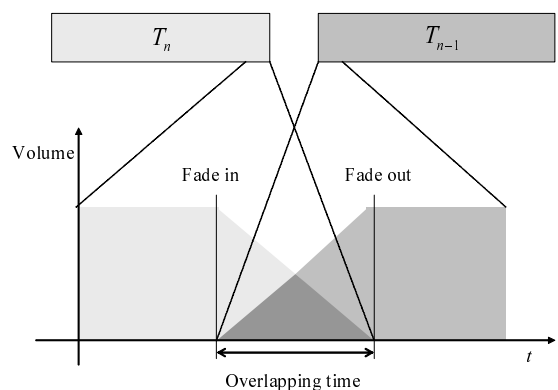


図2: クロスフェードによる連続再生

これはプレイヤーがCDのTOCフィールドから曲の再生時間に関する情報を読み込み、これをもとにCDDDBのデータと照合を行うためである。しかもこのCDDDBのデータは、ユーザのフィードバックにより日々増加している。

コンテンツを配信するものとしてはiTMSに代表されるオンラインストアが挙げられるが、最近ではメタデータとコンテンツを組み合わせたポッドキャストのようなサービスも始められている。ポッドキャストとは、RSS Ver'2.0を公開しているサイトから音声コンテンツを取得し、iPod等の携帯音楽再生機器に取り込むことをいう。ユーザ側では、登録したコンテンツに更新があった場合にのみクライアント側へダウンロードを行い、非同期に視聴を行うことができる。

上記の配信は、いずれもユーザのデータ管理を補助するものである。本研究が扱う再生方法そのものに関わるサービスは、個人レベルでプレイリストを公開する程度で、ほとんど行われていないのが実情である。

3 提案方式

以上の背景を踏まえ、本研究では様々な曲を流すDJの技術・知識に注目した。本章ではまずDJを分析し、その知識に基づいた音楽の再生方法について述べる。次にこの方法を実現するためのデータ配信について述べる。

3.1 DJとは

DJはクラブやディスコなどのライブスペースにおいて、聴衆を飽きさせることなく曲を聴かせ、躍らせる役割を持つ。そのテクニックや手法は、専門とする音楽ジャンルにより様々なものがあるが、代表的技術の1つにミキシングがある。

ミキシングでは、ターンテーブル（レコードの再生機器）とミキサーを使いながら2枚のレコードを同時に操作する。そしてターンテーブルの回転数を変更して2つの曲のテンポを合わせ、一方の曲からもう一方の曲へと違和感なくつなぎ、2つの音源を連続的に再生する。ミキシングと、CDやPCにおける通常の再生方法では、曲の接続部分に大きな違いがある。

図3は、DJによるミキシングでの接続部分を表している。曲の間に無音部分は存在せず双方が混ざりあっている時間帯が存在すること、ボリュームのフェードイン・アウトを行うことはクロスフェードと同等である。違いは、曲のどの時点でボリューム操作を行うかである。クロスフェードでは、 T_n の末端部分数秒と、 T_{n+1} の先頭部分数秒をオーバーラップさせる。これに対しDJによるミキシングでは、 T_n の途中部分でオーバーラップを開始し、 T_{n+1} の途中部分へつなげていく。ミキシングにはDJならではのノウハウがあり、技術的な問題も含め初心者には難しい。

このようなDJの役割や技術は、ユーザが大量の曲の中から特定の曲やプレイリストを繰り返し聴くのは異なる新鮮な感覚を与える可能性がある。

3.2 分析

DJの知識・技術を獲得するため、実際のDJがミキシングした曲の分析を行なった。用意したサンプルは375曲である。以下、図3をもとに分析結果について述べる。

T_n の再生が終了し、次の曲に完全に切り替わるのは、サビの終わり、は曲の開始から一番初めにボーカルが入る「歌いだし」がほとんどであった。ミキシ

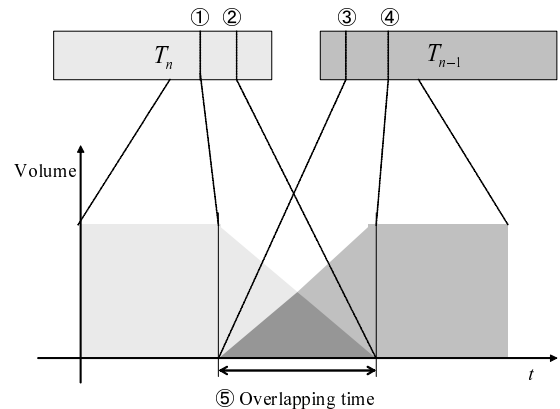


図3: DJのミキシングによる連続再生

ング部における T_n と T_{n+1} の前後関係を調べたところ、全体の77%に相当する289ヶ所が「声から声」へ接続されていた。「声から曲」は12%(45ヶ所)、「曲から声」は8.5%(32ヶ所)、「曲から曲」は2.4%(9ヶ所)だった。

T_n と T_{n+1} の両曲がミキシングされている区間の長さは、主観評価で測定した結果12.32秒となった。一般的なプレイヤーのクロスフェード時間初期値(3~5秒)と比べて比較的長い時間オーバーラップされていることがわかる。

メジャーキー(明るい)からマイナーキー(暗い)へというようなミキシング時の雰囲気の違いに関しては、明暗明暗という極端な転換は見られなかった。プレイリストがマイナーキーで始めると、大抵はマイナーキーの曲が連続して再生されていた。

5人のDJにヒアリングを行ったところ、ミキシングする2曲のテンポであるBPM*の差がなくなるよう調整することが大事だという意見で一致した。DJがミキシングした200曲あまりに対して、マウスボタンを押して手動でBPMを測定した場合(フリーウェア「BTC」利用)、自動で測定した場合(シェアウェア「TRACTOR DJ」利用)のBPM前後差の度数分布を図4に示す。手動の結果からわかるように、ほとんどの場合BPMの前後差が5%以内に抑えられている。自動測定では誤検出があり分布幅が広がっているが、それでも前後差10%以内に集中した。

また、BPM変換の際には、曲の音程を変えずにスピードのみを変更する必要がある。単純にサンプリングデータを伸縮すると、再生速度を遅くすれば音程が低く、速くすれば高くなってしまい、視聴者が違和感を覚えてしまうからである。

*Beat Per Minute. 1分あたりの四分音符の数を表す。

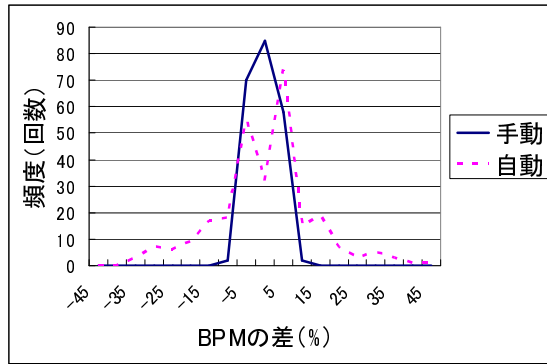


図 4: BPM 前後差の度数分布

3.3 DJ の知識に基づく連続再生

以上の分析結果より，本稿では以下のような連続再生方法の実現を目指す．

1. 手動・ランダムを問わずプレイリストを作成する．
2. 曲の再生を開始する．
3. サビが終わり約 10-12 秒前から次の曲とのミキシングが開始される．
4. ミキシングされる曲の BPM は，前曲の BPM と等しくなるよう調整される．その際，原曲の音程を保つように伸張する．
5. ミキシングが完了するのは次曲の歌いだし地点となるようにする．

3.4 データ配信

3.3 節のような連続再生を実現するうえで，各楽曲のサビの開始・終了といった構成と BPM が必要となる．本稿では，このようなミキシングに必要な各種パラメータを CDDDB のようなデータベースで共有し，メタデータとしてネットワーク配信することを想定する．

パラメータをクライアント側で自動分析するものもある．その例としてサビ区間検出方法 RefraiD[5] では，楽曲中でサビが繰り返し登場することに注目している．処理としては，ある時間区分における C～B 各 12 音階のパワースペクトル分布を表した 12 次元クロマベクトルを特徴量として求め，このクロマベクトルが類似する区間を繰り返しとして推定する．また，音響信号に対して BPM を自動で認識・追跡する手法も存在する [6]．

しかし類似度計算は楽曲全体にわたって必要な上，転調[†]を考慮して同様の計算を 12 回繰り返すなど比較的処理が重い．また，検出精度は約 8 割であり，BPM 認識ソフトも 3.2 節で述べたようにまったく異なる値が算出される場合がある．本稿が扱うミキシング再生用パラメータとして利用するには難しい．

以上より，本稿ではすべてのユーザが同様の処理を行う必要性は薄いと考え，自動分析手法はあくまで補助的な位置づけとする．現在の CDDDB におけるアルバム情報のように，コンテンツ制作者もしくは最初のユーザが新規データを追加していく形を想定する．

4 プロトタイプシステム

4.1 RAIS-Mixer 概要

提案方式を実現するプロトタイプ RAIS-Mixer を開発した．その概要を図 5 に示す．

まず，プレイリスト作成部で再生曲のリストを作成する．これは自身のライブラリの中からランダムに生成しても良いし，自由に編集も可能である．

次に，リストのデータを読み込み，ボリュームや速度変換の処理をするためのバッファに一定長ずつ読み込む．RAIS-Mixer は，このバッファをミキシング用に 2 つ保持している点に特徴がある．

バッファ上のデータはミキサー部に書き込まれ，音声出力される．ミキサー部では，曲の再生位置を常に監視している．再生位置が適切な点に達した時点でもう一方のバッファからデータを読み込みミキシングを開始する．BPM やサビなどの関連情報はネットワーク経由で取得し，必要に応じて速度変換やボリュームのフィルタ加えることで DJ ライクな再生を実現する．

4.2 再生部分詳細

再生の流れの詳細を図 6 に示す．RAIS-Mixer では，ミキシング用データを蓄えるバッファが 2 つ存在する (バッファ 1，バッファ 2)．

まず (1) で，プレイリストから楽曲のデータを読み込み再生を開始する．この時，サビや BPM の関連情報も読み込む．1 曲目の再生開始時は，バッファ上で何の処理も行われない．

[†]楽曲中において，その曲が最初に始まった調とは別の調へ変わること．現代音楽でも頻繁に用いられている．

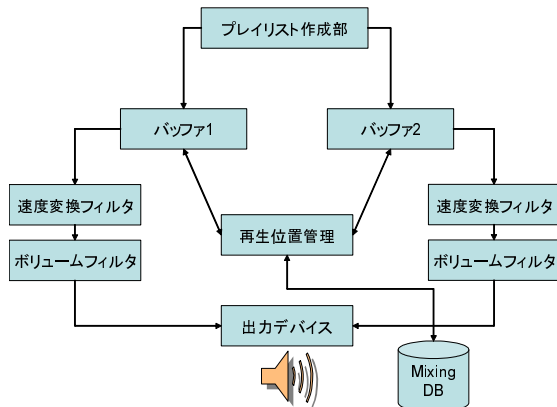


図 5: RAIS-Mixer 概要

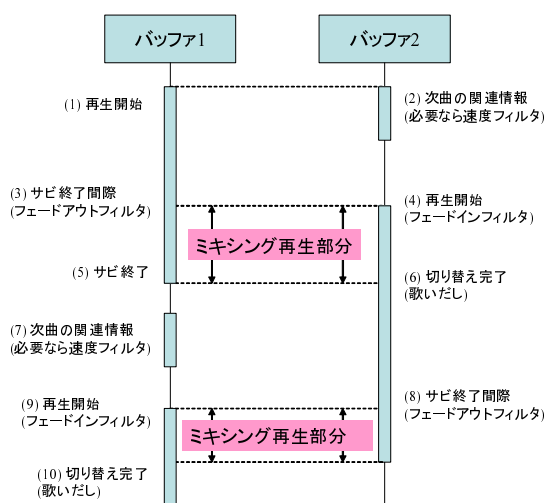


図 6: 再生時の流れ

この直後の (2) では、ミキシングに備えて次曲の BPM を調べる。現在の曲の BPM との差が 5% 以上ならば、以降の再生で速度変換用のフィルタを適用する。

(3) のサビの終了直前に差し掛かった時点で、もう一方の再生を開始する。この時、双方のバッファにはボリュームフェード用フィルタが適用される。

(5) でサビが終わり、一曲目の再生が終了する。(3) ~ (6) で重なり合っている箇所がミキシング再生部分に相当する。完全に曲が切り替わる時点 (6) は、もう一方の曲の歌いだしになっている。(6) 以降は操作を入れ替え、連続的に再生を行う。

4.3 配信データ

ミキシングに関連する情報は、曲名などの一般的な情報以外に、次の項目を扱うこととする。これらをイ

ンターネット経由で取得する。

BPM：楽曲の速度。この項目で速度差を比較し、差が 30% 以内であれば変換を行う。これ以上差がある場合は速度を極端に変更することになり原曲とイメージが変わってしまうため、通常の連続再生を行う。

サビ位置：楽曲中で盛り上がるサビの終了。ミキシング時にビートが一致するように、サビのボーカルが終了する小節の終わり時刻とする。

歌いだし位置：楽曲中で最初にボーカルが始まる部分。サビ位置と同様に、ビートが一致するように、小節の開始時刻とする。

4.4 実装

実装は Windows マシン上で VisualC++ で行った。音声再生部分には DirectSound を用いた。今回は実験のため、対応する音声フォーマットは 44.1kHz の wav のみとなっているが、これは今後容易に拡張可能である。

再生速度変換フィルタでは、音程を変更せずに時間軸を操作する TSM (Time Scale Modification) として SOLA (Synchronized Overlap and Add) を用いた。SOLA は音質はそれほど高くないが計算量が比較的軽く、バッファに読み出されたデータに対してほぼリアルタイムに処理を適用することができる。

5 議論

RAIS-Mixer を使用して、15 人の被験者を対象にアンケート評価実験を行った。評価項目は以下の 3 点であり、5 段階評価を行なった。5 を一番良い、1 を悪いとし、各項目の平均点を求めた。また、視聴してもらったときに気づいた点などを自由形式で記述してもらった。

1. ミキシングでの曲の混ざり具合
2. 本システムの面白さ
3. 従来の再生方法との差

結果、(1) 3.3, (2) 3.8, (3) 4.2 となった。本システムは全体として肯定的な評価を得られた。(1) が他の 2 つに比べて若干低い理由として、プロトタイプの性能の問題が挙げられる。実装に用いたライブラリの関係上、再生中に音が途切れてしまうことが稀にあった。

自由記述部分でもこの点を指摘した被験者が存在した。しかし、プロトタイプでもミキシングが決まったケース（俗に言う Cool）では、自由記述にて「まるで本物の DJ ようだ」との高い評価を得ることができた。

一方、手動・ランダムで生成したリストではミキシングに出来・不出来が存在したようだ。今回は明らかに出来なかった前後の曲の雰囲気が影響している可能性もある。例えば静かな曲の後に激しい曲のように雰囲気が大きく変わる場合でも一概に不適切とは言えず、時にはうまくいく場合もある。現段階では、これら前後関係をシステムティックに判断するには至っていない。

この点に関しては、ミキシングに必要なデータをダウンロードするだけでなく、上手くいった場合の結果をアップロードし、ネットワーク上で広く共有するサービスがあればよいと考えられる。例えば携帯プレイヤーで視聴中、予想外に Cool なミキシングに遭遇した場合は「Cool ボタン」を押し、前後の曲情報を記録しておく。そして次回プレイヤーをクレイドルなどに接続した際、その結果が自動的にアップロードされ知識として共有される。いわば、deli.cio.us のようなソーシャルブックマークサービスに似た、ソーシャルミキシングリストサービスである。

6 おわりに

本稿では、DJ の知識に注目した新たな再生方法を提案した。また、その再生方法を実現するためのデータ配信、サービスモデルに関しても言及した。本手法を用いることで、ユーザに対して新たな音楽の楽しみ方を提供することができる。プロトタイプを用いた評価では概ね肯定的な評価を得ることができた。また、本物の DJ に近いミキシングも可能なことを確認した。

今後の課題としては、ミキシング以外にも DJ の知識・技術の導入が挙げられる。DJammer[7, 8] では、再生中にレコード盤を操作してノイズのような音を出し演出を加えるスクラッチングという技術を加速度センサーなどで実現している。このようなユーザが音楽に対してインタラクションを加えることを検討する必要がある。

また、DJ の技術の根幹は「ビートを合わせる」ことである。我々の身の回りには、音響的にも視覚的にも様々なビートが遍在している [9]。プレイヤーが携帯性を帯びているということは、これらビートに接する機会も当然多くなる。今後はこのような外部のビートに対して同期をとる仕組みも検討していきたい。

参考文献

- [1] 梶克彦, 平田圭二, 長尾確. 状況と嗜好に関するアノテーションに基づくオンライン楽曲推薦システム. 情報処理学会研究報告 2004-MUS-58, pp. 33–38, 2004.
- [2] Microsoft Windows Media Home.
URL: (<http://www.microsoft.com/japan/windows/windowsmedia/default.msp>).
- [3] iPod + iTunes.
URL: (<http://www.apple.com/jp/itunes/>).
- [4] Gracenote CDDB.
URL: (<http://www.gracenote.com/gn-japan/cddb.html>).
- [5] 後藤真孝. Smartmusiciosk: サビ出し機能付き音楽試聴機. 情報処理学会論文誌, Vol. 44, No. 11, pp. 2737–2747, nov 2003.
- [6] 後藤真孝, 村岡洋一. 音響信号を対象としたリアルタイムビートトラッキングシステム — コード変化検出による打楽器音を含まない音楽への対応 —. 電子情報通信学会論文誌 D-II, Vol. J81-D-II, No. 2, pp. 227–237, feb 1998.
- [7] Mat C. Hans and Mark T. Smith. A wearable networked mp3 player and “turntable” for collaborative scratching. In *IEEE International Symposium on Wearable Computers*, pp. 138–145, oct 2003.
- [8] Mat C. Hans and Mark T. Smith. Interacting with audio streams for entertainment and communication. In *Proc. of the 11th ACM international conference on Multimedia*, pp. 539–545, nov 2003.
- [9] 長嶋洋一. 音楽的ビートが映像的ビートの知覚に及ぼす引き込み効果. 芸術科学会論文誌, Vol. 3, No. 1, pp. 108–148, mar 2004.