

透かし入り音源による演奏音への実時間情報埋込み

山本 紘太郎 岩切 宗利

g46073@nda.ac.jp iwak@nda.ac.jp

防衛大学校 情報工学科

あらまし ライブパフォーマンスなどのように、コンテンツの生成と流通が同時に生起するような状況では、実時間埋込み機能のない従来の電子透かしは利用できない。そのため、有効な著作権管理のためにはリアルタイムな電子透かし埋込みが必要である。本報告では、電子楽器の音源情報に透かし信号を埋め込み、その電子楽器を用いて演奏を行うことにより、演奏音に対してリアルタイムな情報埋込みを実現する手法を提案し、ソフトウェアPCM音源を用いたシミュレーションにより基礎的な評価を実施した結果について報告する。提案方式を応用すれば、ライブ演奏のような場面でも電子透かしを利用できる。

A real-time information hiding to musical performance using watermarked instruments

Kotaro YAMAMOTO Munetoshi IWAKIRI

g46073@nda.ac.jp iwak@nda.ac.jp

Department of Computer Science, National Defense Academy

Abstract It is necessary to embed watermark in real time for contents protection at situation like live-performance. We investigate that real-time information hiding to musical acoustic signal, which is generated by performance with digital instruments. In this report, a technical proposal is described, that is real-time information hiding method to acoustic signal of musical performance with watermarked instruments. And we evaluated the proposal method by practical system which is implemented on software PCM wavetable. It is considered that watermark may be able to use in situation like live-performance by proposal method.

1. はじめに

コンピュータの高速化やストレージの大容量化、ネットワークの広帯域化によって、さまざまなコンテンツがデジタル化され、ネットワークや電子的な媒体を介して流通するようになった [1]。デジタルコンテンツはその複製が容易なため、コンテンツの不正流通という問題が生じている。不正流通はコンテンツ制作者の著作権を侵すものであり、正常なコンテンツ流通を妨げる問題になる。これを解決するため、情報ハイディング技術を応用した電子透かしが注目されている [2]。

音楽演奏とその演奏情報に対する電子透かしは、これまで多く報告されている [3, 4, 5, 6]。これらは、コンテンツを流通させる前の段階で著作権管理情報を埋め込む。つまり従来方式では、コンテンツの生

成と透かし情報の埋込みは別々の処理であった。

従来の電子透かしは、コンテンツの流通前に情報を埋め込めるような流通の枠組みの中では有効に機能する。しかし、電子透かしを埋め込む前にコンテンツが流通してしまうような場面では、従来方式を用いることができない。そのため、たとえばコンサート等で生演奏を不正に録音された場合、その不正コンテンツの流通を抑制する方法がない。これを解決するためには、演奏者が演奏する段階、あるいは音響信号が出力装置を通過する段階で情報が埋め込まれるのが望ましい。

本研究は、従来方式とは異なるアプローチにより、音響信号にリアルタイムな情報埋込みを実現することを目的とする。そのために、最近の楽曲演奏に用いられる電子楽器 [7] に注目した。本報告では、電子楽器の音源に透かし信号を埋め込むことにより、演

奏音から透かし情報を抽出可能な実時間情報埋込み法を提案する。提案方式によれば、コンテンツ生成と流通が同時に生じりうるような流通形態で起こりうる著作権の侵害に対処できる。つまり、電子楽器の演奏と情報の埋込みを一体化させることで、不正録音された演奏にも透かし情報を埋め込ませることができ、その情報を用いた著作権管理が行える。

まず、第2章では、電子楽器とその仕組みについて述べる。ここでは、特に電子楽器の中でも、現在一般的に用いられているPCM音源について示す。第3章では、透かし入り音源を利用したリアルタイムな情報埋込み法を提案する。第4章では、ソフトウェアPCM音源を用いたシミュレーションを行い、その性能を評価した結果を示す。

2. 電子楽器とその仕組み

電子楽器とは、「電子音源を持ち、電子的に音を作り出し増幅する楽器」である。電子楽器は、楽器と奏者間のインターフェース、音の生成機構、音を発する機構の3つによって構成される。電子楽器は自然楽器と異なり、この3つの機構をそれぞれ個別に設計できるという特性を持つ。ここでは、特に音の生成機構に注目し、一般的な手法について示す。

2.1 音源情報の生成

音源情報の生成法は次の3つに大別される。

- (1) PCM (Pulse Code Modulation)
自然楽器の音をデジタルサンプリングし、入力に応じて読み出し、音程と音強に応じ変調を加え出力する方法である。
- (2) アルゴリズムによる音響合成
音響信号処理の要素をソフトウェア的なアルゴリズムとして合成する方法である。ユニットジェネレータと呼ばれる基本要素を用いてさまざまなアルゴリズムを構成する。加算合成、減算合成、非線形変換等の方法がある。
- (3) 物理モデルによる音響合成
自然楽器などの音響振動を物理的なモデルとして構成し、そのままリアルタイムシミュレーションにより音響を生成する方法である。

これらのうち、PCMは録音された実楽器の音響信号を用いる。それに対してアルゴリズム、物理モデルによる方法では、モデルに従って音響信号を合

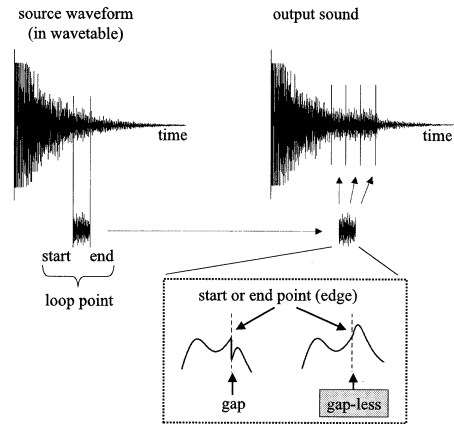


図 1: ルーピング処理

成する。現在、商用音源として主流となっているのはPCMである。次にPCM音源の音響出力方法について述べる。

2.2 PCM音源のサンプリング合成

PCM音源では、あらかじめ自然楽器の音をデジタル録音し、音源内に保存する。そしてインターフェースを介して入力された発音の指示に対し、対応する波形を読み出して出力する。これがPCM音源の基本的な仕組みである。しかし、たとえばピアノの演奏であれば、各鍵盤に対して、強さの異なる複数の音源波形が必要となる。また、音の長さが一定であるとは限らず、演奏者の打鍵、離鍵動作に応じて出力を変化させる必要がある。ペダルによる音色の変化も考慮しなければならない。これらを考慮すると、そのデータ量は膨大となる。

そのため、通常、商用のPCM方式の音源では、保存領域の削減のためルーピングやピッチシフトが施される。次に、これらの処理について示す。

2.2.1 ルーピング

ルーピングとは、キーボードで演奏されるサンプリング音の長さを伸ばす処理である。デジタルサンプリングされ、保存される音源波形は一定の長さしか持たない。しかし電子楽器は、入力に対し消音されるまで継ぎ目なく音を出力し続けなければならない。そのため、サンプリング音に繰返し区間を定め、それを繰り返すことで長い音の出力を可能にする。図1は、wavetable上の波形にルーピングを施し、

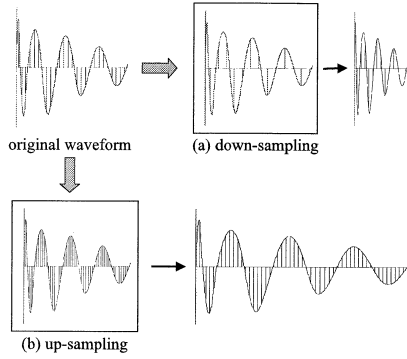


図 2: ピッチシフト処理

長い音を出力する処理を示したものである。このとき、繰り返し区間の境界で継ぎ目にずれ (gap) が生じないように注意しなければならない。

2.2.2 ピッチシフト

発音される全ての音程に対して音源波形を保存できるだけの記憶容量が確保できない場合、3~4 半音置きに波形を保存し、ピッチシフト処理によって必要な高さの音を合成する。ピッチシフトには、周波数領域ピッチシフトと時間領域ピッチシフトがある。次に、時間領域ピッチシフトの1つである、サンプリングレート変換によるピッチシフトについて示す。

サンプリングレート変換は、時間領域でのリサンプリングによって実現する。ピッチをあげる場合には、より低いサンプリングレートでリサンプルし、元のサンプリングレートで出力すればよい。これは変更するサンプリングレートにしたがって不要なサンプルを読み飛ばせばよい。これをダウンサンプリングとも呼ぶ。同様に、ピッチを下げる場合には、より高いサンプリングレートでリサンプルすればよい。これを実現するには、現存するサンプルとサンプルの間を補完して、新たなサンプルを挿入する。これをアップサンプリングとも呼ぶ。

図 2-(a) はピッチを2倍にするときの、図 2-(b) はピッチを半分にするときのリサンプリングをそれぞれ表している。

3. 提案方式

PCM 音源はその機構上、音源波形が持つ周波数成分を保持しつつ音響として出力する。ここでは、これを利用した実時間情報埋込み法を提案する。従来

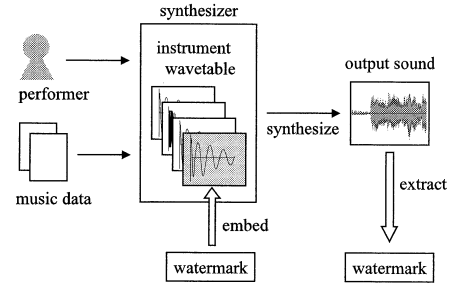
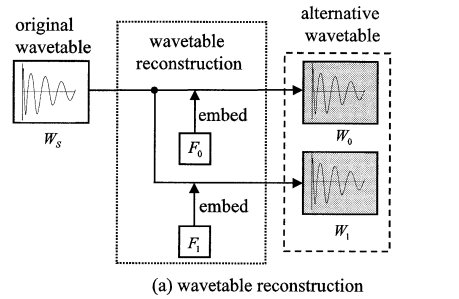
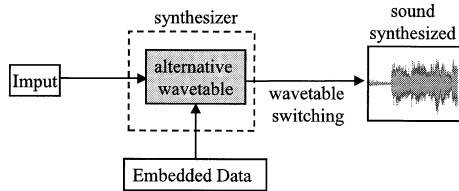


図 3: 提案方式



(a) wavetable reconstruction



(b) switching output control

図 4: 波形選択法

方式では、図 3 のように録音された音声に対して情報を埋め込んでいたが、本研究では異なるアプローチを採用する。すなわち、音源情報にあらかじめ透かし信号を埋め込み、その音源による演奏により演奏音に透かし情報を挿入する。このように、演奏音響にリアルタイムで情報を埋め込めるという手法である。

3.1 波形選択法

音源情報の波形 W_s に対し、特定の周波数信号 F_0 を埋め込んで波形 W_0 を生成し、演奏を行う。すると、再生音からは F_0 を観測できる。同様に、図 4-(a) のように W_s に対して周波数信号 F_0, F_1 をそれぞれ埋め込んだ W_0, W_1 を用意し、それらを出力

の時点で切り替えられるように音源を設計する。すなわち、再生音には F_0 , F_1 が含まれることになる。このとき、図 4-(b) のように、埋込み情報 E によって波形 W_E を選択するようにすれば、音源波形の切り替えによって情報を表現できることになる。これが波形選択法である。

3.1.1 サンプルング合成による透かし信号への影響

波形選択法による情報埋込みを実現するためには、音源からの出力音声から、埋込まれた信号を正しく検出できなければならない。つまり、第 2 章で述べたルーピングやピッチシフトの影響をあらかじめ考慮して埋込み法を設計する必要がある。

ルーピングの、ループ部分には同じ周波数成分が含まれる。そのため、もし音源波形全体に信号 F_r を埋め込むと、ループ部分の長さによっては正しい信号抽出の難しい場合がある。ループポイントは図 1 のように、発音の後半部分で指定されるため、音響のもっとも強い先頭部分の周波数成分は影響を受けない。そのため、 F_r を音源波形の先頭部分に埋め込めば、ルーピングによる影響を考慮しなくてもよい。

また、ピッチシフトはリサンプルングによって元の周波数分布全体を、その形状を保持したまま上下させる。仮に 5~6 半音の間でピッチをシフトさせた場合、そのシフト幅は録音された波形の上下各 3 半音となる。すると、平均律の楽器であればそのシフト幅は元の周波数 F_r の $2^{-\frac{1}{4}} \sim 2^{\frac{1}{4}}$ 倍程度である。仮に 2 種類の信号を埋め込んでも、その周波数の間隔が十分であれば、ピッチのシフト幅をあらかじめ考慮することにより出力音声から別々の信号として抽出できる。

3.2 波形選択法の処理手順

波形選択法では、音源側にあらかじめ透かし信号入り音源波形を保持させ、演奏と同時に、埋込み情報による透かし波形の出力制御を行う。

3.2.1 透かし入り音源波形の生成

2 種類の波形を用意し、1 ビットの情報を表現する際の音源情報への信号付加処理を示す。本研究では、基礎的な検討のため、特定の低周波成分に透かし信号を埋め込むことにした。

Step 1. 透かしを埋め込む波形 W_S が L_I 個のサンプルを持ち、そのサンプルングレートが R_S ,

サンプルングデータが $x_k(k = 0, \dots, L_I - 1)$ であるとする。フレームサイズ $N = 0.2 \times R_S$ とし、離散フーリエ変換 (DFT: Discrete Fourier Transform) により周波数スペクトル f_p を求める。

$$f_p = \sum_{k=0}^{N-1} x_k e^{\frac{2\pi i}{N} pk} \quad (p = 0, 1, \dots, N-1). \quad (1)$$

式 (2) 中の i は虚数単位を表し、 x_k と f_p はともに複素数である。DFT のフレームサイズを 0.2 秒で行うのは、音源波形のアタックレベルに透かし信号を埋め込むことでルーピングによる影響をなくすためである。

Step 2. 特定の周波数信号 F_0 [Hz] ($F_0 \leq 20$) を埋め込む。 F_0 , N , R_S を用いて次式により埋め込み対象スペクトル m を決定する。

$$m = \lfloor \frac{F_0 \times N}{R_S} \rfloor. \quad (2)$$

周波数スペクトル f_m を

$$|f_m| = \frac{1}{10} \max(|f_p|). \quad (3)$$

により置き換える。このとき、 $\max()$ は f_p の最も強いスペクトルを返す関数である。

Step 3. 得られた f_p を逆 DFT (IDFT: Inverse DFT) によってサンプルングデータ x'_k に変換する。

$$x'_k = \frac{1}{N} \sum_{p=0}^{N-1} f_p e^{\frac{2\pi i}{N} pk} \quad (k = 0, \dots, N-1). \quad (4)$$

これを波形 W_0 とする。

Step 4. 同様に、 W_S へ周波数信号 F_1 [Hz] ($F_1 \leq 20$, $F_1 \neq F_0$) を埋め込み、 W_1 とする。このとき、演奏音では F_0 , F_1 が式 (1) の幅を持つことに注意し、それぞれを識別できる十分な間隔を確保する。

3.2.2 透かし情報による出力波形制御

提案方式の波形出力処理について示す。 F_0 [Hz] の信号を含む波形を W_0 , F_1 の信号を含む波形を W_1 とする。電子楽器のインターフェース機構で入力を受け取る都度、埋込み情報 E を 1 ビット読み込む。次のように出力波形 W_E を選択する。

$$W_E = \begin{cases} W_0 & (E = 0 \text{ の場合}) \\ W_1 & (E = 1 \text{ の場合}) \end{cases} \quad (5)$$

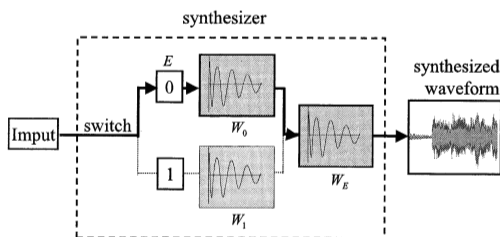


図 5: 透かし情報による出力制御 ($E = 0$)

このように定めた W_E を用いて音を合成する。図 5 は、 $E = 0$ のときの出力制御を示したものである。

3.2.3 再生音からの情報抽出手順

Step 1. 再生音の波形 W_P が L_P 個のサンプルを持ち、そのサンプリングレートが R_P 、サンプリングデータが $x_k (k = 0, \dots, L_P - 1)$ であるとす。フレームサイズ N 、DFT 間隔を T_I 秒として、DFT により $t \times T_I$ 秒での周波数スペクトル f_p を求める。

$$f_p = \sum_{k=O_t}^{O_t+N-1} x_k e^{2\pi i p k} \quad (6)$$

$$(p = O_t, O_t + 1, \dots, O_t + N - 1)$$

$$O_t = T_I \times R_P \times t$$

このとき N は埋め込んだ周波数信号が識別できる分解能を確保できる大きさにする。

Step 2. $n = \lfloor \frac{20 \times N}{R_S} \rfloor$ として、 $0 \leq p < n$ に関して次式を満たすピーク S_t を求める。

$$S_t = \lfloor \frac{p \times R_P}{N} \rfloor \quad (7)$$

$$(p : f_p - f_{p-1} > 0 \text{ かつ } f_{p+1} - f_p < 0)$$

S_t が複数存在する場合、スペクトルの強度が最大の要素を採用する。

Step 3. W_P の終端まで、Step 1. と Step 2. を繰り返し、 S_t を求める。

Step 4. 同じ値を持つ S_t が 0.2 秒以上連続している区間を取り出し、次式により埋込み情報 E を抽出する。

$$E = \begin{cases} 0 & (0.84 \times F_0 < S_t < 1.2 \times F_0) \\ 1 & (0.84 \times F_1 < S_t < 1.2 \times F_1) \end{cases} \quad (8)$$

なお、このとき S_t に幅を持たせているが、この幅は信号を埋め込んだ音源の最大ピッチシフト幅から適切に定める。

4. 試作システムの実装と評価

提案方式を実電子楽器に実装する場合には、埋込み情報に基づいて出力波形を切り替える機構を音源に組み込む必要がある。本研究では、ソフトウェア PCM 音源を用いたシミュレーション実験を行った。

4.1 実装方法

システムの実装には、ソフトウェア PCM 音源である TiMidity++^{*1} を用いた。音源情報として freepats^{*2} で配布している GUS Patch (Gravis UltraSound Patch) を利用した。その音源波形に対し人の可聴域外の周波数 F_0 、 F_1 に透かし信号を埋め込み、 W_0 、 W_1 とした。これらを別々の楽器として TiMidity++ の音源チャンネル C_0 、 C_1 に登録した。

試作システムの処理を次に示す。

(1) 演奏情報の入力

SMF (Standard MIDI File) [8] を入力する。

(2) 音源による波形生成

Step 1. 埋込み情報に基づき SMF に楽器 C_0 、 C_1 へのプログラムチェンジ^{*3} を挿入し、Modified SMF を生成する。

Step 2. TiMidity++ に Modified SMF を渡し、出力させる。

(3) 音声出力と保存

TiMidity++ の出力を音声波形として出力し、保存する。この処理はデジタル録音 [9] と同等である。

波形の出力制御は (2) 音源部の Step 1. で行う。つまり、発音指示であるノートメッセージの直前に、埋込み情報に応じてプログラムチェンジを挿入することにより情報を埋め込む。

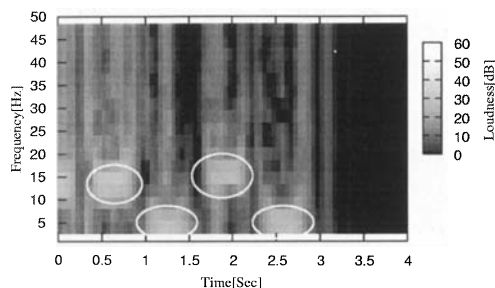
本研究では、ピアノの音源情報^{*4} に $F_0 = 5$ 、 $F_1 = 15$ の信号を埋め込んだ。また基礎的な検討のため、入力 SMF には、ピアノのみによる、和音や音の重複のない楽曲を使用した。

^{*1} <http://sourceforge.net/projects/timidity/>

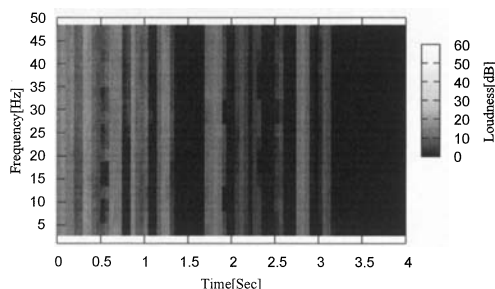
^{*2} <http://freepats.opensrc.org/>

^{*3} MIDI チャンネルに楽器を割り当てる制御符号

^{*4} freepats に含まれる 000_Acoustic_Grand_Piano.pat



(a) analysis result of embedded waveform



(b) analysis result of original waveform

図 6: 音源出力結果 ($F \leq 50$)

4.2 実験結果と考察

埋め込みを施した出力音声の周波数分析結果を図 6-(a) に、埋め込みを施さない出力音声の周波数分析結果を図 6-(b) にそれぞれ示す。出力音声のサンプリングレートは 44100Hz、量子化ビット数は符号付き 16 ビットであり、周波数分析はフレームサイズ $N = 16384$ 、0.05 秒間隔の DFT により行った。図 6-(a) の白円で示した部分に、埋め込んだ信号が確認できる。

信号を埋め込んだ音源波形による演奏音からは、埋め込んだ周波数信号が検出でき、正しく情報が抽出できた。このことから、電子楽器演奏における実時間情報埋め込みと再生音からの抽出が可能であることが分かる。提案方式ではルーピングやピッチシフトの影響を考慮しているため、どのような楽器に対しても、単音であれば十分な情報の抽出が可能であると考えられる。以上のように、基礎的ではあるが提案方式の有効性を確認した。

5. おわりに

本報告では、電子楽器の音源情報への透かし信号埋め込みによる演奏音への実時間情報埋め込み法を提案

した。音源波形に対し単一の透かし信号を埋め込み、再生時に埋込み情報に基づく波形出力制御を行うことにより、演奏動作と情報埋め込みを一体化できることを確認した。

提案方式では、再生音が出力装置を通過する際には情報が埋め込まれている。そのため、生演奏などの場面における著作権管理や、情報配信などの用途に応用できると考えられる。

本報告の評価では、演奏音は直接デジタル録音されている。しかし、スピーカ等を介したアナログ録音にも対応できることが望ましい。これらに対する評価および検討は今後の課題である。

参考文献

- [1] 財団法人 デジタルコンテンツ協会 : デジタルコンテンツ白書 2006, 財団法人デジタルコンテンツ協会 (2006).
- [2] 松井甲子雄 : 電子透かしの基礎, 森北出版 (1998).
- [3] Gruhl, D., A. Lu, and W. Bender : Echo Hiding, *Springer Lecture Note in Computer Science*, vol.1174, pp.7-21(1996).
- [4] Bender, W., D. Gruhl, and N. Morimoto : Techniques for data hiding, *IBM Systems Journal*, vol.35, no.3/4, pp.131-136(1996).
- [5] 井上大介, 松本勉 : スタンダード MIDI ファイル ステガノグラフィとその能力, 情報処理学会論文誌, Vol. 43, No. 2, pp.2489-2501(2002).
- [6] 岩切宗利, 山本紘太郎, 関根健一郎, 松井甲子雄 : 電子演奏の半雑音化と音源符号への電子透かし, 情報処理学会論文誌, Vol. 43, No. 2 pp.225-233(2002).
- [7] Roads, Curtis., 青柳龍也, 後藤真孝: コンピュータ音楽-歴史・テクノロジー・アート, 東京電機大学出版局 (2001) .
- [8] 社団法人 音楽電子事業協会 : MIDI 1.0 規格書, リットーミュージック (1998).
- [9] Bruse, Claudius. : デジタル・レコーディングの全知識, リットーミュージック (2003).