

推薦論文

# 極座標表現を用いた3D MIDIの実装と評価

黒岩 謙<sup>1,†1,a)</sup> 宮田 宙和<sup>2</sup> 宇田 隆哉<sup>1</sup>

受付日 2011年7月15日, 採録日 2012年1月13日

**概要:** 近年, 5.1ch サラウンドや 3D 技術を用いた映像, 音声が目まぐるしく注目されている. それにともない様々な規格のサラウンド対応音声ファイルが作られた. その中で MIDI も 2009 年 7 月 23 日にサラウンド化に必要なパラメータが新たに定義された. しかし, 拡張規格として定義されたため, 古い規格や今までのシーケンサで扱うことができない可能性がある. そこで, 我々は聴覚特性を考慮した, サラウンドに必要な定義したパラメータを情報ハイディング技術を用いて埋め込む手法を提案し, 3D MIDI の実装と, その評価を行った.

**キーワード:** 情報セキュリティ, データハイディング, MIDI

## Proposal and Evaluation of 3D MIDI Sound with Expression in Polar Coordinates

KEN KUROIWA<sup>1,†1,a)</sup> HIROKAZU MIYATA<sup>2</sup> RYUYA UDA<sup>1</sup>

Received: July 15, 2011, Accepted: January 13, 2012

**Abstract:** Recently, movies and sounds with 3D technology have been paid attention to. There are various standards for 3D sounds and a new definition for 3D sounds is added to MIDI since July 23, 2009. However, existing MIDI instruments and MIDI sequencers may not read files with that 3D definition since the definition is added to existing MIDI as extension. Therefore, we propose, implement and evaluate another definition in which parameters for 3D MIDI are embedded in a file with information hiding technology in consideration for auditory characteristics. In our proposal, MIDI files with 3D parameters are compatible with existing MIDI instruments and MIDI sequencers.

**Keywords:** information security, data hiding, MIDI

### 1. はじめに

近年, DVD や Blu-ray の普及により, 映画やその他のコンテンツが家庭でも 5.1ch 等のサラウンド環境で利用可能となった. また, 現在, マルチメディアコンテンツの中で特に 3D を用いた映像, 音声が目まぐるしく注目されている. その中でも映像と音声を同時に体感できる 3D 映画の普及によ

り, 今後ますます 3D の普及率が上昇することが期待されている. その結果, サラウンド環境で再生されることを前提とした音声ファイルの需要が高まり, 様々な規格のサラウンド対応音声ファイルが作られた. しかし, 従来から一般的に利用されていた MIDI はサラウンドに対応しておらず, 依然としてステレオ環境のままであった. MIDI を用いたサラウンド楽曲の制作過程では, あらかじめ楽曲で用いる楽器の音を MIDI で録音, 用意しておき, 最終的にミキシングを通して立体的な音を再現している. この方法では, サラウンド化された楽曲の部分的な音の取り直し, 音色

<sup>1</sup> 東京工科大学大学院バイオ・情報メディア研究科  
Graduate School of Tokyo University of Technology,  
Hachioji, Tokyo 192-0982, Japan

<sup>2</sup> 慶應義塾大学大学院理工学研究科  
Graduate School of Science and Technology, Keio University,  
Yokohama, Kanagawa 223-8522, Japan

<sup>†1</sup> 現在, KDDI 株式会社  
Presently with KDDI Corporation

<sup>a)</sup> g2110023ee@mf.teu.ac.jp

本論文の内容は 2010 年 7 月のマルチメディア, 分散, 協調とモバイル (DICOMO) シンポジウム 2010 にて報告され, マルチメディア通信と分散処理研究会主査により情報処理学会論文誌ジャーナルへの掲載が推薦された論文である.

の変更をミキシングと同時にせず、楽器ごとの MIDI データを用意し直さなければならない。つまり、MIDI を最初から用意し直すということである。これでは、非常に手間と時間がかかってしまい、これが MIDI を用いたサラウンド楽曲を制作するうえでの大きな障害となり普及を妨げている。そこで松本らは MIDI をサラウンドに対応させ、ゲームの BGM や効果音として利用できるよう新たにサラウンド化に必要な情報を定義した [1], [2]。その際、既存の MIDI 規格に従うよう情報ハイディング技術を用いて既存の MIDI イベントに対してサラウンド化に必要な情報を埋め込んだ。その後、2009 年 7 月 23 日に MIDI Manufacturers Association<sup>\*1</sup>においてサラウンド再生に必要なパラメータが公開され、MIDI もサラウンド対応音声ファイルとなった。しかし、このパラメータは聴覚特性に基づいて作成されておらず、定義してあるパラメータを生かしきれないという問題点や、新しく定義された規格であるために古い MIDI 機器を用いて再生を試みた際に正常に再生されない可能性があるという問題点がある。

そこで、我々は従来の古い MIDI 形式の中にサラウンド情報を効率良く埋め込める方式を提案し、文献 [2] で示されたデータハイディング手法を用いて、サラウンド再生に必要なパラメータを既存の MIDI ファイルに対して埋め込んだ。この埋め込み手法では、人間の聴覚特性を考慮することで埋め込みビットサイズの圧縮を図っており、MIDI ファイルを無駄に肥大化させることなく効率良くサラウンド情報を埋め込むことが可能となっている。以降、2 章で既存技術をまとめ、3 章で極座標を用いた 3D MIDI について述べる。4 章で提案手法を述べ、5 章で評価する。

## 2. 既存技術

本章では、2.1 節で MIDI のイベントについて、2.2 節で本実装で用いているステガノグラフィ技術について、2.3 節では既存の 3D MIDI について解説する。

### 2.1 MIDI

本節では、MIDI のイベントから、本実装で用いている部分について述べる。

#### 2.1.1 デルタタイム

デルタタイムとは MIDI において次に続くイベントまでの時間を表す情報のことである。トラックの最初のイベントがトラックの先頭に発生する際のデルタタイムや、2 つのイベントが同時に発生するような場合は、0 となる。前のイベントと次のイベントの時間間隔を定義するために、デルタタイムはつねに必要な。また、SMF (Standard MIDI File) ではイベントのデルタタイムを可変長表現で表している。可変長表現とは、先頭 1 ビットをフラグとし

て使用し、実際のデータは残りの 7 ビットで表すものである。先頭ビットのフラグは、後続バイトがある場合には 1、最後のデータバイトは 0 がセットされる。この可変長表現により、わずかな時間から長大な時間までを効率良く表すことができるようになっている。

#### 2.1.2 ステータスバイトとデータバイト

多くの MIDI メッセージは複数のバイトで構成されており、メッセージの種類を表すステータスバイトとデータバイトから構成される。それぞれのバイトの先頭ビットがそのバイトがステータスバイトなのか、データバイトなのかを区別している。ステータスバイトは先頭ビットがつねに 1、データバイトは 0 になる。したがってステータスバイトの値は 80h から FFh までとなり、データバイトの値は 00h から 7Fh となる。

#### 2.1.3 ピッチ・ベンド・チェンジ

ピッチ・ベンドはシンセサイザキーボードにあるホイール、ジョイスティック等のピッチベンド情報を送るメッセージである。ピッチ・ベンドとは音程 (ピッチ) を自由にシフトさせることで、ギターのコッキングや管楽器のしゃくり上げ、スライドといった滑らかな音程変化をつけることを可能にするためのものである。ピッチ・ベンドのメッセージはピッチ・ベンドを示すステータスバイト “En” に続き、データバイトである LSB と MSB の 2 バイト、あわせて 3 バイトで構成されている。ピッチ・ベンド幅は LSB と MSB の 2 バイトを用いて、最大 14 bit のデータの低位 7 bit を LSB へ、上位 7 bit を MSB バイトへ分割して出力する。

## 2.2 SMF ステガノグラフィ

既存研究には、“スタンダード MIDI ファイルステガノグラフィとその能力 [3]”、“表情付けを考慮した SMF ステガノグラフィ [4], [5]”、“発音時間の揺らぎを利用した SMF ステガノグラフィ [6]”、“MIDI ファイルのピッチベンド・チェンジコードを対象とした新たな情報埋め込み方式の提案および評価 [1]”、“音程を変化させる編曲が可能な SMF ステガノグラフィの提案および評価 [7]”がある。

文献 [3] は、SMF を媒体としたステガノグラフィ方式 (SMF ステガノグラフィ) を提案/発表したものである。この研究では、埋め込み対象のファイルサイズに対して平均約 1 [%]、最大約 4 [%] の埋め込み率を有することを明らかにしている。文献 [4], [5] では、表情付けの施されていない楽音符号に対し、ルールベース表情付けによってテンポへの表情を付加し、あわせて情報を付加する手法について提案されている。この手法によって埋め込み容量の増大、より自然な表情付けの実現が可能になるというものである。文献 [6] は、音楽の実演奏で付加される抑揚や小さな揺らぎ等に見せかけて情報を埋め込むステガノグラフィである。文献 [6] は文献 [5] と同じ手法で埋め込み容量の増加を

<sup>\*1</sup> MIDI Manufacturers Association <http://www.midi.org/>

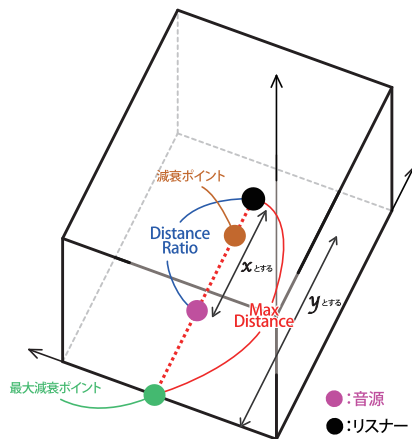


図 1 3D Sound Controllers におけるパラメータの概要  
 Fig. 1 Parameters in 3D Sound Controllers.

行ったものである。文献 [1] は、音程変化を表すピッチ・バンド・チェンジコードに対して秘匿情報を埋め込むというものである。平均埋め込み率は 6.7 [%] となっている。文献 [7] は、音程値に関して編曲を行った際、同時に秘匿情報を埋め込むというものである。これにより、元々ピッチ・バンド・チェンジコードが存在していなかった楽曲に対しても秘匿情報が埋め込めるようになった。

また、MIDIへ埋め込む秘匿情報の活用例として、“Digital Signature Scheme for Flexible Usage of a MIDI File [8]”, “The Efficiency Method of Sanitizable Signature Scheme for SMF [9]” があげられる。最後に、本稿の基礎理論を示した“リアルタイムで位置情報を入力できるサラウンド MIDI シーケンサの開発 [2]” がある。

### 2.3 3D Sound Controllers

本節では、MIDI Manufacturers Association で公開された 3D Sound Controller の定義から、主な部分を抜粋して述べる。図 1 にパラメータの概要を示す。

リスナの位置を図 1 のとおりに設定すると、音源を配置できる空間はリスナの正面、左右、上下それぞれ 1,000 [m] までとなる。平面空間での音源のリスナに対する角度を AZIMUTH ANGLE パラメータで表す。リスナ位置の正面を  $d = 0$ 、リスナ位置の右を  $d = 90$ 、リスナ位置の左を  $d = -90$ 、リスナ位置の背後を  $d = -180$  とし以下の式で Data MSB, Data LSB を決定する。

$$a = -180 + d \times \frac{360}{16384} \quad (1)$$

また、三次元空間での音源のリスナに対する角度、つまり仰角を ELEVATION ANGLE パラメータで表す。リスナと同じ高さを  $d = 0$ 、リスナの真上を  $d = 90$ 、リスナの真下を  $d = -90$  とし、式 (1) で Data MSB, Data LSB を決定する。

また、通常 MIDI におけるボリューム (CC#7)、エクスプレッション (CC#11) の代わりに音の出力レベル

mB (milliBels) を用いて表す。これがリスナとの距離の代わりとなり、GAIN パラメータで表す。最低レベルを  $-163.82$  dB、最大レベルを  $0$  dB とし、Data MSB, Data LSB を決定する。

また、Distance Ratio は音源がリスナから離れた際の減衰量を表している。減衰量に関するパラメータとして、Max Distance, Gain at MAXIMUM Distance, Reference Distance Ratio がある。Max Distance は最大減衰量を表しており、Max Distance に設定された距離以上に音源が離れたとしても、Gain at MAXIMUM Distance に設定された音量以下にはならない。Max Distance は図 1 では  $y$  に相当し、最大減衰ポイントにおける音源の音量が Gain at MAXIMUM Distance に相当する。一方、Reference Distance Ratio は減衰が始まる距離を表しており、図 1 に示した減衰ポイントに相当する値が Reference Distance Ratio である。

### 3. 極座標を用いた 3D MIDI

本章では、本提案で用いる既存技術である文献 [2] で示した手法について論じる。今回、文献 [2] で用いられた提案の実装にあたり、文献 [2] の発表後に MIDI Manufacturers Association で定義されたサラウンドに対応したパラメータをふまえ、新たに極座標表現のサラウンド対応 MIDI に必要なパラメータを再定義した。既存の MIDI 規格には MIDI Manufacturers Association で定義されるまでサラウンドのパラメータを示すデータ構造がなく、サラウンド出力が可能なシーケンサもない。そこで三次元 MIDI に必要なパラメータを MIDI に対する情報ハイディング技術である SMF ステガノグラフィを用い、MIDI 規格から逸脱することなく必要な情報を埋め込む。また、本稿では文献 [2] で用いられている埋め込み手法を用いて実装を行った。以降、極座標表現のサラウンド対応 MIDI (以降 3D MIDI と呼ぶ) を制作する手順を示す。まず、3D MIDI を作成するために情報の埋め込みが可能なシーケンサを起動し、MIDI キーボードを用いて演奏情報を MIDI に記録し、それと同時に音源の位置と動きを指定する。次に、音源と位置と動きに修正があればシーケンサ上で行き、演奏情報の記録が終わったら、サラウンド化に必要な情報を MIDI へ埋め込む。最後にサラウンド化された MIDI を出力してシーケンサを終了する。次にそれぞれの工程での処理を示す。

#### 3.1 演奏情報の記録と再生

まず、本シーケンサを起動し、MIDI キーボードを用いて演奏情報を記録する。それと同時に音源の位置と時間変化による音源の動きを指定する。なお、本シーケンサは一般的に普及しているシーケンサと同等の編集機能を有し、GUI での操作を基本として、音源の位置や動きの指定も GUI を用いて行う。ここでの処理において、サラウンド出

表 1 DirectSound のプロパティ  
Table 1 Propaties of DirectSound.

A : 音源 (Buffer3D クラス)	B : リスナ (Listener3D クラス)
A-1. 位置 : Position プロパティを使用 設定 : $set\_Position(Vector3)$ 取得 : $get\_Position(Vector3)$	B-4. 位置 : Position プロパティを使用 設定 : $set\_Position(Vector3)$ 取得 : $get\_Position(Vector3)$
A-2. 速度 : Velocity プロパティを用いる 設定 : $set\_Velocity(Vector3)$ 取得 : $get\_Velocity(Vector3)$	B-5. 速度 : Velocity プロパティを使用 設定 : $set\_Velocity(Vector3)$ 取得 : $get\_Velocity(Vector3)$
A-3. 向き : ConeAngles, ConeOrientation, ConeOutsideVolume プロパティを使用 設定 : $set\_*(Vector3)$ 取得 : $get\_*(Vector3)$	B-6. 頭部の向き : Orientation プロパティを使用 設定 : $set\_Orientation(* 略 *)$ 取得 : $get\_Orientation(* 略 *)$

表 2 時刻  $t$  で必要なサラウンド情報  
Table 2 Required information for surround music at time  $t$ .

	備考	データサイズ
$(r_t, AZI\theta_t)$	時刻 $t$ における座標 : リスナを中心とした半径 (9 bit), 角度 (9 bit)	18 bit
$ELE\theta_t$	仰角	8 bit
$\vec{r}_t$	半径方向の速度	11 bit
$\vec{\omega}_t$	角速度	11 bit
$E\vec{L}V_t$	仰角の変化率	11 bit
$t$	時刻	8~96 bit

表 3 時刻  $t$  の可変長表記  
Table 3 Variable length field of  $t$ .

時刻 $t$ の bit 長	ビット列の中身	時刻 $t$ のデータサイズ
0~7	1*****	8 bit
8~14	0***** 1*****	16 bit
14~84	0***** 0***** ... 1*****	24 bit~96 bit

力と音源の動きの管理は DirectX の DirectSound を用いて行う。DirectSound におけるサラウンド音響は 3D サウンドと呼ばれ、3D 空間での音源とリスナの位置・速度・向きを、直交座標の 3 本の座標軸、x 軸、y 軸、z 軸の値で表す。以降、この直交座標を Vector3 とする。本シーケンサは、3D サウンドの Buffer3D、Listener3D クラスを用いてサラウンド再生を行う。表 1 に、それぞれのクラスで用いる 6 つのプロパティを示す。

表 1 の\*は音源の向きを表す 3 つのプロパティ名、\*略\*には Vector3 とは異なるベクトル表記の座標が入る。音源とリスナそれぞれに対して位置、速度、向きを指定できるが、本シーケンサでは、リスナの位置は仮想空間の中央で、直交座標 (0,0,0) として固定、速度は 0、頭部の向きは正面に固定する。つまり、Listener3D クラス B-4~B-6 の Vector3 はすべて初期値のまま固定する。音源の向きは、静止しているときはつねにリスナのいる方向、移動しているときは進行方向に向くよう設定する。ConeAngles と ConeOutsideVolume の値は初期値を利用する。

次に、DirectSound を用いるうえで 3D MIDI に必要な

情報 (サラウンド情報) を示す。1 回の音源移動につき、表 2 のパラメータを必要とする。

本提案で用いる音源の動きの表現は座標 A から座標 B へ移動する際の軌跡を細かく座標データとして保持せず、ある時刻  $t$  における座標と速度から、音源の軌跡を補完する方法をとる。時刻  $t$  は、シーケンサで音源を移動させた際の時間であり、 $t_0$  における座標は音源の初期位置とし、 $t_0$ ,  $\vec{r}_t$ ,  $\vec{\omega}_t$ ,  $E\vec{L}V_t$  は数値 "0" とする。 $t_1$  は初めて音源が移動する際の曲の再生開始時からの経過時間を表し、 $t_2$  の間、座標  $(r_0, AZI\theta_0)$  から毎時刻  $r_0$ [m/s] だけ半径が変化、毎時刻  $\omega_0$ [m/s] だけ角度が変化、毎時刻  $E\vec{L}V_0$ [m/s] だけ仰角が変化する。以降、 $t_i$  の間、座標  $(r_{i-1}, \theta_{i-1})$  から毎時刻  $r_{i-1}$ [m/s] 半径が変化、毎時刻  $\omega_{i-1}$ [m/s] 角度が変化する。表 2 で示したサラウンド情報は音源を静止状態から移動させるたびに追加する。時刻の単位はテンポ/分解能の値とする。時刻  $t$  は MIDI のフォーマットであるデルタタイムと同様のデータ構造とし、先頭 1 ビットをフラグとし、時刻  $t$  が 8 bit を超えた場合、表 3 で示す値とする。

DirectSound では x, y, z の軸座標で表されるが、音が

リスナの周囲を巡回する表現を考慮し、データ表現の効率化のために角度を使用している。座標を表すための角度  $AZI\theta_t$  は、1 周 360 度なので、1 度刻みで表し、9 bit とする。リスナの正面に音源がある場合  $AZI\theta = 0$ 、左にある場合  $AZI\theta = 270$ 、右にある場合  $AZI\theta = 90$ 、背後にある場合  $AZI\theta = 180$  とする。リスナを中心とした半径  $r_t$  は、浮動小数点で 9 bit。うち、10 を基底として指数部 2 bit 使用し、範囲は  $10^{-3} \sim 10^0$ 、数値部 7 bit とする。単位はメートル。音源の高さを表す仰角  $ELE\theta_t$  は、リスナの頭と同じ高さの場合  $ELE\theta = 90$ 、足元の場合  $ELE\theta = 0$ 、真上の場合  $ELE\theta = 180$  とし、1 度刻みの 9 bit とする。仰角の値が 90 以外の場合、音源の座標は仰角  $ELE\theta_t$  と半径  $r_t$  を用いて音源の高さと距離を算出し、その後  $AZI\theta$  の値だけ回転させる。半径方向の速度  $\vec{r}_t$  と角速度  $\vec{\omega}_t$  は、浮動小数点で 11 bit。うち、符号部 1 bit、10 を基底として指数部 3 bit を使用し、範囲は  $10^{-7} \sim 10^0$ 、数値部 7 bit とする。単位はそれぞれ、 $\vec{r}_t$  [m/s]、 $\vec{\omega}_t$  [rad/s]。  $r_t$ 、 $\vec{r}_t$ 、 $\vec{\omega}_t$  の最大値は 127 とする。座標を表現する際のリスナを中心とした半径と半径方向の速度、角速度の精度が違うのは、音源の移動時間が長い場合、時間が経過するにつれて座標の精度が落ちることを防ぐため、速度の精度を高め、座標の精度を落とさないようにしている。また、2 点間のみで補完するため、リスナから近い場所を巡回する場合と遠い場所を巡回するときで補完する際の誤差に違いが生じてしまうことが考えられる。そこで、リスナを中心とした半径  $r$  を対数表記にすることで誤差を小さくする方法をとる。これにより必要となるサラウンド情報のデータ容量を節約でき、なおかつ音源のあらゆる動きを再現することが可能である。

しかし、表 2 で示した埋め込むサラウンド情報は人間の聴覚特性を考慮せずに決定していたため、聴覚特性に合わせたサラウンド情報を再定義し、埋め込むことでより埋め込むデータ量を減らすことができるのではないかと考えられる。

### 3.2 MIDI へのサラウンド情報の埋め込み

ここでは、3.1 節で示したサラウンド情報を、演奏情報が記録された MIDI に埋め込む。埋め込むデータを  $EmbData$  とすると、

$$EmbData = \{t_0, r_{t_0}, AZI\theta_{t_0}, ELE\theta_{t_0}, \vec{r}_{t_0}, \vec{\omega}_{t_0}, \\ EL\vec{V}_{t_0}, t_1, r_{t_1}, AZI\theta_{t_1}, \dots\}$$

という連続したデータになる。

また、埋め込まれた情報がサラウンド情報であることを表すために、まず、ヘッダとして専用の ID:S3DM (Steganography 3D MIDI) の文字列を ASCII コードで表現した 4 byte とすべての  $EmbData$  を接続したデータのハッシュ値  $H(EmbDataA)$  (SHA1 で生成された 20 byte) をバイナリ値として埋め込む。これにより、

同じ埋め込みアルゴリズムで別のデータが埋め込まれていたとしても、埋め込まれているデータのハッシュ値と  $H(EmbDataA)$  が一致しない場合は埋め込まれている情報は本方式のサラウンド情報ではないと判断し、3D での再生をしないものとしている。このように、ヘッダとサラウンド情報を埋め込むことにより、仮にサラウンド情報以外のデータが埋め込まれていた場合であっても互換性に影響を与えずに利用することができる。

これらのヘッダやサラウンド情報は次に示す方法を用いて MIDI に埋め込まれる。まず、埋め込み対象となる MIDI イベントはピッチ・ベンド・チェンジコードの音程値とする。複数の楽器で構成される MIDI の場合、 $n$  番目の楽器のサラウンド情報は  $n$  番目のチャンネルの音程値に対してのみ埋め込む。つまり、それぞれの楽器の移動情報はそれぞれの楽器の演奏情報に対してのみ埋め込まれることに留意されたい。また、実際に 3.1 節で記録した際に発生したピッチ・ベンド・チェンジコードを用いるのではなく、各チャンネルの演奏が終了した後、各チャンネルの最後に  $EmbData$  を埋め込んだ音程値のピッチ・ベンド・チェンジコードを埋め込み容量に達するまで追加する。

次に、ピッチ・ベンド・チェンジコードの追加と  $EmbData$  の具体的な埋め込み処理を示す。

#### 3.2.1 音程値への埋め込み

音程値は 14 bit で表される。つまり、1 回のピッチ・ベンド・チェンジコードで  $EmbData$  を 14 bit ずつ埋め込むことが可能である。追加するピッチ・ベンド・チェンジコードを  $Pitch_n$  とし、それぞれのデルタタイムを 0 とする。次に、埋め込む際の手順と埋め込み処理を示す。

##### Phase.1 $EmbData$ の分割

14 bit 単位で、 $EmbData$  を分割する。

このとき、分割した  $EmbData$  を  $DivEmbData1_y$  ( $0 \leq y < \frac{EmbData.length}{14}$ ) とする。

なお、 $EmbData.length$  は  $EmbData$  のデータサイズである。

##### Phase.2 $Pitch_n$ への埋め込み

それぞれの楽器のチャンネルの先頭から順番に、

```
01:for(i=0;i<y;i++){
02:   $Pitch_i=DivEmbData1_i$ ;
03:}
```

となるようにピッチ・ベンド・チェンジコードの音程値を変更する。

楽器が複数ある場合、それぞれのチャンネルで独立して埋め込み処理を行う。 $n$  チャンネルあれば上で示した埋め込み処理を  $n$  回行う。次に 3.3 節でサラウンド情報を含んだ MIDI としてファイル出力する。

### 3.3 MIDI の出力

ここでは、3.2 節でサラウンド情報が埋め込まれた MIDI

を、ファイル出力する。サラウンド情報は独自に定義した MIDI イベントを用いていないため、本シーケンサを用いた場合はサラウンド MIDI として、また、既存のシーケンサや再生アプリケーションを用いた場合は既存の MIDI として再生される。以降、3.4 節でサラウンド化された MIDI の再生方法を述べる。

### 3.4 サラウンド MIDI の再生

すでに本シーケンサによってサラウンド情報が埋め込まれた MIDI の再生、制作中の MIDI の再生には DirectSound の 3D サウンド機能を用いる。以降、再生時の処理を示す。

#### Step.1 サラウンド情報の抽出

それぞれの楽器のチャンネルのすべてのベロシティと音程値からサラウンド情報を抽出し、プログラム内部に保持する。

#### Step.2 楽器数を把握

まず、サラウンド MIDI のヘッダチャンク内に格納されているチャンネル数を読み込む。次に、チャンネル数分、サラウンド MIDI を分割し単チャンネルの MIDI ファイルを作成する。つまり、楽器の数だけサラウンド MIDI を作成する。

#### Step.3 サウンドバッファの作成

チャンネルごとに分割されたサラウンド MIDI からサウンドバッファを作成する。

#### Step.4 サウンドバッファのパラメータ設定

サウンドバッファに以下のパラメータを設定し、スピーカから再生する。リスナの位置を仮想空間の中央で固定、速度 0、頭部の向きは正面に固定する。つまり、Listener3D クラスの値はすべて初期値のまま固定する。

また、音源が静止しているときはつねにリスナのいる方向へ向いているよう設定する。次に、サラウンド情報から時刻  $t$ 、時刻  $t$  における座標、仰角、半径方向の速度、角速度、仰角の変化率を取り出し、直交座標へ変換する。表 1、A-1set\_Position(Vector3) の x 軸、y 軸、z 軸には、時刻  $t$  における座標と仰角から算出し、その結果を代入。A-2set\_Velocity(Vector3) の x 軸、y 軸、z 軸には、1 つ前の時刻にいた位置から現在の位置に向かうベクトルの値を、A-3set\_ConeOrientation(Vector3) の x 軸、y 軸、z 軸には、半径方向の速度、直交座標に変換した角速度、仰角の変化率を用い、3 つのベクトルを加算した値を代入する。

#### Step.5 再生中のパラメータ再設定

再生中は、100 ms ごとに半径方向の速度、角速度、仰角の変化率から座標を再計算し、サウンドバッファのパラメータを更新する。また、音源が停止した場合、set\_Velocity(Vector3) の x 軸、y 軸、z 軸には 0 を代入し、移動を停止する。

## 4. 聴覚特性を考慮した効率の良いサラウンド情報埋め込み手法の提案

本稿では、MIDI Manufacturers Association で定義されたサラウンドパラメータや文献 [2] で示されたサラウンド情報をもとに、人間の聴覚特性を考慮し、パラメータに必要なビット数を少なくすることで効率化された新しいサラウンド情報を提案する。また、文献 [2] で提案されている手法を用いて、サラウンド再生に必要なパラメータを既存の MIDI ファイルに対して情報ハイディング技術の一種である、SMF ステガノグラフィを用いて埋め込みを行っている。このため、新しい定義である 3D Sound Controllers では古い規格にしか対応していない MIDI 機器において再生できない可能性や、今までのシーケンサでは読めない可能性があるといった問題点を生じることなく、3D MIDI の実現が可能であると考えられる。効率化された新しいサラウンド情報を定義するために、聴覚特性に対する実験および評価を行い、聴覚特性に基づいた新しいサラウンド情報を提案する。人間の耳は頭部の左右に付いているため、聴覚特性上音の発信源を認識しやすい向きとしにくい向きが存在する。本提案手法では、この聴覚特性を考慮し、角度に対するビットの割当てを最小限にとどめ、MIDI ファイルが無駄に肥大化しないよう効率化を図っている。具体的には、4.1 節で評価した聴覚特性をふまえてビットの割当てを決定しているため、ビットの配置についての詳細は 4.2 節を参照されたい。

また、聴覚特性を考慮したサラウンド情報を定義しているため、3D 音響環境においては、ITU-R BS.775-1 で推奨されているとおりにスピーカが設置されていることを本稿の前提としている。しかし、各家庭等の設置場所によっては、3D 音響システムがそれぞれ固有の形態をとらざるをえないこともあり、ITU-R BS.775-1 で推奨されているとおりにスピーカが設置されていることになるよう、アンブ側にスピーカ設置位置補正の仕組みが組み込まれているものもある。

### 4.1 聴覚特性の評価

人間の聴覚特性を測定し、効率の良いサラウンド情報を検討するために、実楽器を用いて楽器がどの位置で鳴っているのかをリスナがどの程度把握できるのかを計測し、評価を行った。実楽器を用いたのは、MIDI 音源を用いて実験した場合どの MIDI 音源を使うかによって音の鳴り方が変わってしまうが、実楽器以上に精巧な音を出せる MIDI 音源は存在しないと考えられるためである。

まず評価方法と評価の手順を示す。測定は次の方法で行った。各方向において、音源を被験者が特定できるかどうかを測定するために、被験者から 5[m] はなれた地点で楽器を発音させた。このときの音源について図 2 に示す。

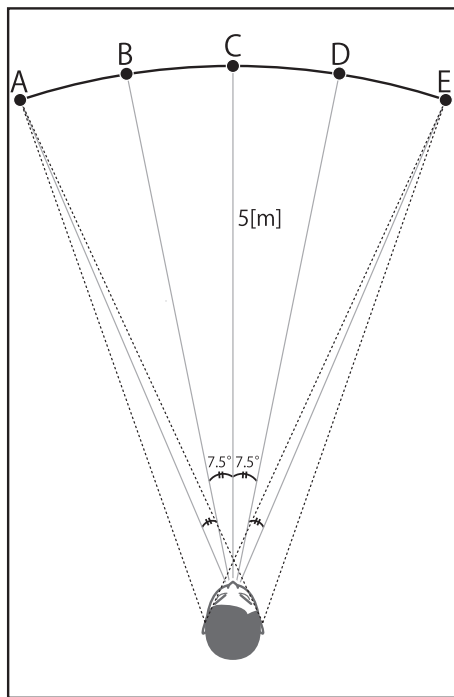


図 2 評価における音源の位置

Fig. 2 Musical performance point for evaluation.

表 4 音色の発生に用いた楽器

Table 4 Musical instruments for evaluation.

楽器名	音階	周波数 [Hz]
トランペット	A4	440
トロンボーン	A2	110
フルート	A5	880
ファゴット	A2	110
ヴィオラ	A4	440
チェロ	A1	55

なお、表 2 のパラメータでは仰角も設定できるが、今回仰角を再現する環境が用意できなかったため、平面環境での実験とした。また、各方位で左右に音源が移動した際に判断できるか測定するため、A 点から E 点まで左右 30 度の発音範囲を設定したうえで被験者に向けて発音を行い、被験者に音源のある方向を判断させ、発音地点に対し左方向に誤差が出たか、右方向に誤差が出たかを各方位で 10 回ずつ測定した。なお、4.1 節内では便宜上、極座標表記を用いず角度は度数表記とする。次に評価を行った空間の環境と、使用した楽器、音源、被験者の情報を示す。

a. 評価場所 全無響室<sup>\*2,\*3</sup> (広さ: 横幅 5.30 [m] \* 縦幅 7.20 [m] \* 高さ 3.50 [m])

b. 使用した楽器 表 4 で示されている楽器をオーケストラの方の協力を得て演奏

このとき、楽器の高さは演奏者の身長により異なるが、指向性を考慮し被験者の方を向いて演奏するものと

<sup>\*2</sup> 日本工学院八王子専門学校コンピュータ&テクノロジーセンター無響室

<sup>\*3</sup> 水平方向だけではなく、垂直方向も吸音材で覆われている無響室

した。そのため、楽器の向きは楽器の種類によって異なる。

c. 被験者 被験者は 5 名で、被験者 A, B, C が 20 代の男性、被験者 D, E が 20 代の女性である。

被験者は同一の椅子に着座した状態で評価を行った。同一の椅子を用いているため、被験者の座高によって耳の高さは異なっている。また、右耳と左耳の感覚は被験者によって個人差があるものとしている。

次に、表 5、表 6 に各楽器における前方からの音に対する聴覚特性実験における特定誤差と、その偏差を、表 7、表 8 に各楽器における側方からの音に対する聴覚特性実験における誤差と、その偏差を、表 9、表 10 に各楽器における後方からの音に対する聴覚特性実験における特定誤差と、その偏差を示す。表 5 から表 10 において、回数は実験回数を、各セルの数字は角度を表している。

#### 4.2 提案するビット配置

まず、現在埋め込んでいるサラウンド情報の妥当性を、実楽器における誤差の偏差を表 5 から表 10 に示した結果から考える。前方からの音に対しては全体的に高い精度で認識されていることが表 5、表 6 より分かる。しかし、誤差が少なからず生じており、一番正答率の高い被験者で偏差は 2.27、一番正答率の低い被験者では 5.85 と一番認識しやすい前方からの音であっても、各被験者の間で大きな差が生じていることが分かる。側方からの音に対しては表 7、表 8 に示したとおり、誤差が大きく、音源位置の認識が困難であることが分かった。このとき、誤差の偏差は 9.47~14.42 と前方の 3~4 倍程度の数字になっている。後方からの音に対しては前方よりも認識が困難であるが、側方より音源位置の認識は可能であることが表 9、表 10 より分かる。このとき、誤差の偏差は 6.47~10.28 とバラつきが大きく、前方に近い認識が可能である楽器と、側方からの音源と同様に認識が困難な楽器が存在していることが分かった。これらのことから、どの方向からの音であっても 1 度刻みで表現する必要はないのではないかと考えられる。そのため、サラウンド情報で用いるビット数を削減し、効率の良い埋め込みを行うために、ビット配置を表 11 に示すように再定義することを提案する。

従来手法では角度に 9 bit を割り振り、1 度を 1 bit とし、1 bit 刻みで表現していたが、提案手法では 5~7 bit を割り振っている。これは、表 12 に示した評価より導出された誤差の標準偏差を考慮している。このとき、各被験者の偏差と、5 名の平均と正答率の高い上位 40%、上位 20% の偏差を導出した。この結果から、5 名の平均を基に 1 bit あたりの角度を決定した場合と、上位 20% の結果を考慮して決定した場合では、差が大きくなってしまおうと考えられる。そのため、埋め込む情報量を変化させることで、誤差が生じないようにサラウンド情報を作成するのか、埋

表 5 被験者・楽器別の誤差 (前方 1)  
Table 5 Margin of errors in listening (front case1).

回数	楽器名						
	トランペット	トロンボーン	フルート	ファゴット	ヴァイオラ	チェロ	
被験者 A	1	-15	0	-15	-15	0	0
	2	-15	0	0	0	0	-15
	3	-15	0	0	0	0	0
	4	0	0	0	0	0	0
	5	0	0	0	0	0	0
	6	0	-15	-15	0	0	0
	7	-15	-15	0	0	15	0
	8	0	0	0	0	0	0
	9	0	0	0	0	0	0
	10	-15	0	0	0	0	0
被験者 B	1	0	2	1	-3	15	5
	2	5	-3	-3	-4	15	0
	3	0	0	0	0	2	0
	4	5	0	0	5	0	0
	5	0	5	-15	4	10	2
	6	0	5	-12	4	0	3
	7	0	0	-2	-2	5	0
	8	0	0	0	0	0	3
	9	0	10	0	-5	10	0
	10	-10	3	6	-2	0	-10
被験者 C	1	10	0	0	0	0	0
	2	0	0	0	0	0	0
	3	0	0	0	4	0	0
	4	-15	0	0	0	0	0
	5	0	5	0	0	0	0
	6	0	0	0	0	-5	-5
	7	0	0	0	0	0	0
	8	0	0	5	0	0	0
	9	0	0	0	0	0	5
	10	0	0	0	0	0	0

め込む情報量を極力減らしてサラウンド情報を作成するの  
かを作成者が決定できるようにすることでどのようなケー  
スにあっても対応させることができると考えられる。これ  
らのことを考慮し、表 13 に提案するサラウンド情報にお  
ける 1bit あたりの角度を示す。また、表 12 に示したと  
おり、提案手法では割り振ることで埋め込む情報量を削減  
しつつも、再現度の高い 3D MIDI が実現できると考えら  
れる。

## 5. 評価

本章では、本提案の有用性について評価する。まず、  
5.1 節で極座標表現の妥当性について論じ、5.2 節において  
聴覚特性に基づくサラウンド情報の有用性について論ずる。

### 5.1 極座標表示の妥当性

本節では、独自に定義したサラウンド情報の極座標表現  
の妥当性と、3D Sound Controller に対する優位性につい

て考察する。まずサラウンド情報の極座標表現の妥当性  
を考える。直交座標で表す場合、ある時刻  $t$  における座標  
( $x, y, z$ )、速度 ( $\dot{x}, \dot{y}, \dot{z}$ ) が必要となる。音源が直線的に移動  
する場合は、これらの 2 つのパラメータで軌跡を表現でき  
る。しかし、人間の聴覚に大きな影響を与えようとする  
とき、その楽器がリスナの周囲を旋回するように動かすこ  
とが、以下の理由で効果的であると考えられる。

リスナの周囲を楽器が旋回する動きが、直線的な動きよ  
りも人間の聴覚に与える影響が大きい理由について、図 3、  
図 4 を用いて説明する。人間が音の向きを感じる理由は、  
音源から左右それぞれの耳までの音の伝達時間に差がある  
ことである。つまり、音源から右耳と左耳までの距離の差  
によって、人間は音の向きを感じている。音の波長および  
波形にも依存するため、任意の距離で任意の波長および波  
形の音に対してすべて適用されるわけではないが、おおよ  
その場合において説明する。人間が鉛直方向手前に頭頂部  
を向け、 $y$  軸方向に視線を向けているとする。左右の耳の



表 6 被験者・楽器別の誤差 (前方 2)  
Table 6 Margin of errors in listening (front case2).

回数		楽器名					
		トランペット	トロンボーン	フルート	ファゴット	ヴィオラ	チェロ
被 験 者 D	1	-20	-5	-5	-5	15	-3
	2	-10	0	0	-5	0	0
	3	5	-5	0	-15	0	10
	4	-10	3	3	-5	0	10
	5	0	0	0	-5	6	0
	6	-7	0	-5	0	3	10
	7	5	-6	0	0	0	5
	8	0	-13	3	-12	-2	5
	9	-5	6	2	0	0	-8
	10	-1	2	7	10	12	0
被 験 者 E	1	0	-1	0	8	0	-15
	2	0	1	0	0	5	0
	3	-2	-6	-5	10	-3	-5
	4	3	0	0	-15	2	2
	5	-2	-5	-4	-3	-1	-15
	6	-12	1	-3	3	0	-1
	7	0	2	-7	0	0	0
	8	3	0	0	-5	-9	-4
	9	-3	-6	2	-10	10	1
	10	0	0	-6	0	4	7
標準偏差		6.80	4.45	4.58	4.03	4.39	4.18

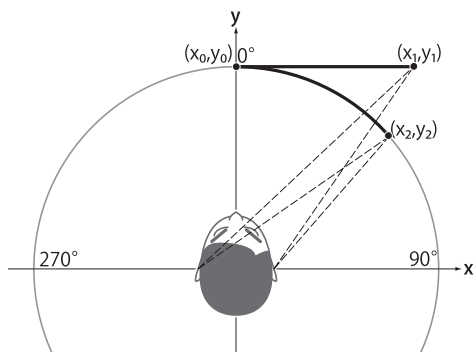


図 3 音源から左右の耳までの距離 (1)

Fig. 3 Distance between sound source and each ear (1).

距離の差が 0 となるのは  $y$  軸上に音源がある場合であり、左右の耳の距離の差が最大となるのは  $x$  軸上に音源がある場合である。まず、 $(x_0, y_0)$  に音源があり、その音源が等距離の軌跡を描き  $(x_1, y_1)$  または  $(x_2, y_2)$  に移動したとすると、頭部を中心に旋回起動を描く軌跡のほうが、音源から左右の耳までの距離の差が大きくなる。ただし、図 3 では直線的な動きの例が  $x$  軸方向に沿う向きであったため、さらに図 4 を用いて、音源が直線的に斜めに移動する場合との比較説明を行う。音源が  $(x_0, y_0)$  から  $(x_3, y_3)$  に移動する場合は、移動距離からみると直線的に動くほうが短い。この例では、音源から左右の耳まで距離の差が同じだけ変化した際に、直線的な動きのほうが移動距離は短いことになるが、そのままの軌道で音源が動き続けた場合、た

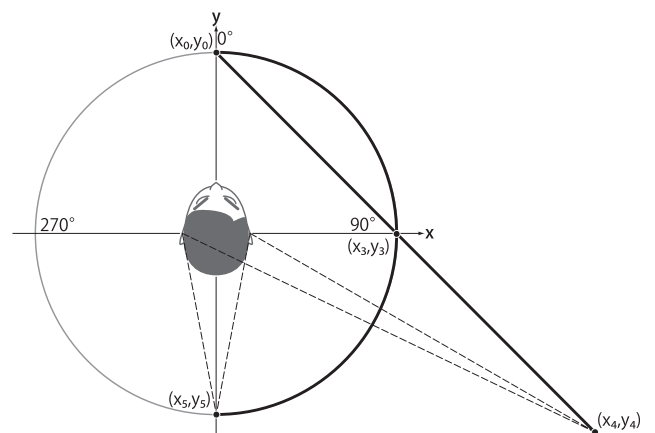


図 4 音源から左右の耳までの距離 (2)

Fig. 4 Distance between sound source and each ear (2).

たとえば  $(x_4, y_4)$  と  $(x_5, y_5)$  では、 $(x_3, y_3)$  の位置からの音の向きの変化は、旋回軌道のほうが大きくなる。以上より、音源が、加速度が急激に変化しない動きをするのであれば、リスナの周囲を旋回するように楽器を動かすほうが、直線的に動かすよりも効果的に人間に音源の向きの変化を伝えることができるといえる。旋回軌道を表現するには、直角座標よりも極座標のほうが適しているため、本稿や MIDI Manufactures Association では、MIDI の 3D 表現に極座標を採用している。

次に、旋回軌道を描く音源の位置を表現するには、直角座標よりも極座標を用いたほうが少ない点で正確に描け

表 7 被験者・楽器別の誤差 (側方 1)  
Table 7 Margin of errors in listening (side case1).

回数		楽器名					
		トランペット	トロンボーン	フルート	ファゴット	ヴァイオリン	チェロ
被験者 A	1	-15	15	30	0	0	0
	2	-15	0	0	-15	-30	0
	3	15	15	-45	-15	0	15
	4	0	15	-30	-15	-15	-45
	5	15	-15	0	-15	-30	0
	6	15	0	15	-15	-45	0
	7	0	45	0	-15	-15	-15
	8	15	0	0	15	-15	30
	9	-15	-15	30	-30	-15	30
	10	15	-15	-30	-15	15	15
被験者 B	1	-5	0	15	-10	5	20
	2	12	0	-15	-10	0	10
	3	-15	0	0	-15	-15	-5
	4	10	0	5	10	15	-15
	5	0	-3	5	0	-5	-10
	6	0	10	0	-5	15	-5
	7	3	0	11	-10	0	0
	8	0	0	0	9	-10	-15
	9	5	-4	4	-5	10	-15
	10	11	5	15	12	5	-5
被験者 C	1	10	0	15	0	60	0
	2	30	0	0	0	0	0
	3	15	0	15	0	0	0
	4	15	-10	0	-15	-15	-5
	5	0	-10	0	-30	0	0
	6	5	0	0	0	0	15
	7	0	0	-15	0	0	0
	8	15	10	-15	0	-15	0
	9	10	-15	0	0	-15	-15
	10	5	0	0	30	15	0

る理由を説明する。説明を簡潔かつ簡略にするため、半径 1 [m] の真円上を、1 秒間に 1 周の等速度で音源が巡回運動する例について、図 5 を用いて説明する。まず、各座標系では次のパラメータをそれぞれ次に示す順番で記述するものとする。

- 極座標系 (上向きを  $0^\circ$ ) ( $r, \theta, v_r, v_\theta, a_r, a_\theta$ )

$r$ : 円の半径  
 $\theta$ : 角度  
 $v_r$ : 半径方向の速度  
 $v_\theta$ : 角速度  
 $a_r$ : 半径方向の加速度  
 $a_\theta$ : 角加速度

- 直角座標系 ( $x, y, v_x, v_y, a_x, a_y$ )

$x$ :  $x$  座標  
 $y$ :  $y$  座標  
 $v_x$ :  $x$  方向の速度

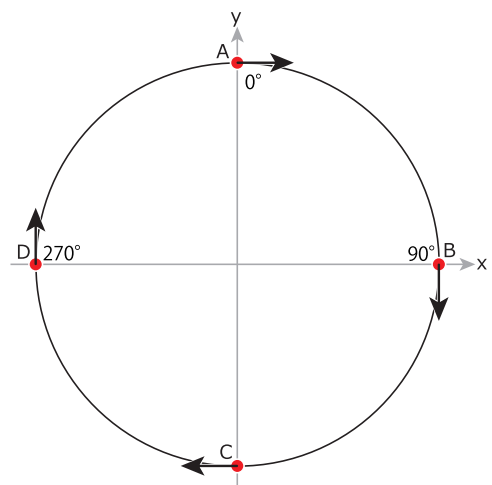


図 5 音源の移動例  
Fig. 5 Example of moving sound source.

表 8 被験者・楽器別の誤差 (側方 2)  
Table 8 Margin of errors in listening (side case2).

回数	楽器名						
	トランペット	トロンボーン	フルート	ファゴット	ヴィオラ	チェロ	
被験者 D	1	5	10	-10	0	5	2
	2	0	25	0	15	15	0
	3	-8	0	15	-5	15	-10
	4	5	20	2	3	10	30
	5	7	13	-15	0	10	0
	6	30	0	-5	10	20	0
	7	0	25	0	5	0	15
	8	10	10	-5	-13	5	-7
	9	15	20	0	10	-5	15
	10	-5	-25	3	5	15	30
被験者 E	1	0	-7	-15	-7	0	15
	2	2	2	-1	0	0	0
	3	5	-15	0	-15	-5	-15
	4	0	-9	1	-22	5	5
	5	-15	-10	2	10	-3	0
	6	0	10	5	-15	-4	-5
	7	15	5	10	-12	0	0
	8	0	6	-15	-10	-55	-8
	9	-10	4	0	5	15	-9
	10	-4	-10	-19	15	-15	0
標準偏差	9.47	10.21	11.74	10.74	14.42	12.04	

$v_y$ :  $y$  方向の速度  
 $a_x$ :  $x$  方向の加速度  
 $a_y$ :  $y$  方向の加速度

なお、円周  $2\pi$  [m] ( $2\pi r$  であるため) を 1 秒で進むので、円の接線方向の速度は  $v = 2\pi$  [m/s], 円の中心方向に向かう加速度は  $a = \frac{v^2}{r} = 4\pi^2$  [m/s<sup>2</sup>] となっている。

この例においては、比較のため、極座標系では点 A の位置のみパラメータを記録し、直角座標系では点 A, B, C, D の 4 点のパラメータを記録する。各点のパラメータは次のようになる。

- 極座標系  
 A (1, 0, 0,  $2\pi$ , 0, 0)
- 直角座標系  
 A (0, 1,  $2\pi$ , 0, 0,  $-4\pi^2$ )  
 B (1, 0, 0,  $-2\pi$ ,  $-4\pi^2$ , 0)  
 C (0, -1,  $-2\pi$ , 0, 0,  $4\pi^2$ )  
 D ( $-1$ , 0, 0,  $2\pi$ ,  $4\pi^2$ , 0)

直角座標系においては、 $x$  軸方向および  $y$  軸方向の加速度がそれぞれ等加速度であると仮定して、この円を 1 周する時間を均等に 20 分割して円周上に 16 点を補完した。極座標系においては、中心点への重力加速度および周回軌道の加速度がそれぞれ等加速度であると仮定して、この円を 1 周する時間を均等に 20 分割して円周上に 19 点を補完し

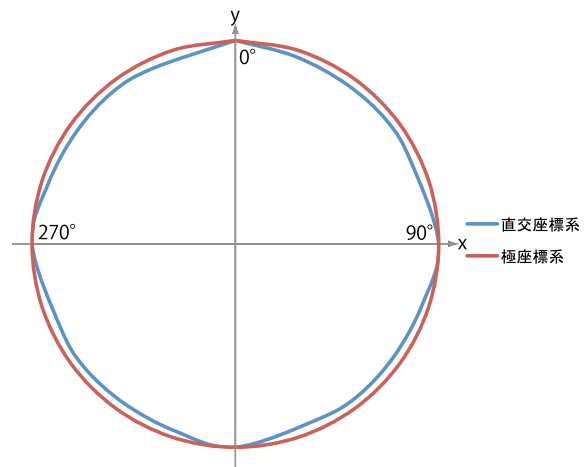


図 6 極座標系と直角座標系の比較

Fig. 6 Compare polar coordinates system rectangular coordinate.

た。補完された点を含み、それぞれの座標系で 20 点をつないだ様子を図 6 に示す。今回の例では、真円にしたため極座標系における各点の平均誤差は 0 [m], 直角座標系における各点の平均誤差は 0.031 [m] となった。この誤差に対し、極座標系では 2 点 (点 A を 2 回含む必要があるため)、直角座標系では 5 点 (点 A を 2 回含む必要があるため) で起動を表現していることから、ほぼ同じ軌道を描くために、極座標系は直角座標系と比較して 40% のデータ量で済んでいることが分かる。

表 9 被験者・楽器別の誤差 (後方 1)  
Table 9 Margin of errors in listening (back case1).

回数	楽器名						
	トランペット	トロンボーン	フルート	ファゴット	ヴァイオラ	チェロ	
被験者 A	1	0	0	0	-15	0	0
	2	0	-15	-15	-15	0	-15
	3	0	0	0	0	-15	0
	4	0	-15	0	15	0	15
	5	0	0	0	-15	0	0
	6	0	15	15	-15	0	0
	7	0	0	0	0	0	0
	8	0	0	0	-15	0	0
	9	-15	-15	-15	0	15	0
	10	0	0	0	-15	0	0
被験者 B	1	-10	15	1	-15	0	3
	2	0	-3	6	0	2	0
	3	-10	0	18	-4	0	-25
	4	10	0	10	0	10	10
	5	-2	10	1	-2	7	-10
	6	0	-3	15	0	0	5
	7	2	0	8	-10	0	0
	8	-10	0	-5	0	-3	5
	9	4	12	12	-5	-3	0
	10	-6	0	-6	4	-2	0
被験者 C	1	0	-5	0	0	0	0
	2	0	0	0	0	0	0
	3	0	0	0	0	0	10
	4	0	5	0	15	-5	-5
	5	-15	0	0	0	0	0
	6	0	0	15	0	0	15
	7	0	0	0	0	0	-5
	8	0	8	15	0	0	0
	9	0	0	0	0	15	0
	10	-15	0	0	0	-10	-15

なお、本稿では音源の加速度が急激に変化しないことを前提に座標系の議論をしているが、音源の加速度が急激に変化する場合には、極座標と直角座標のどちらが効率良く表現できるかについては一概にいえない。究極的には、MIDIの世界では実楽器を使用する必要がないため、音源が瞬時に（移動時間0秒で）位置を大きく変えるような表現をすることも可能である。しかし、MIDI Manufactures Associationによる3D MIDIの定義でも極座標が採用されていることから、本稿と同様、実楽器がリスナの周囲を巡回する場合に、少ないデータ量で効率良く動きを表現できるようにすることを想定していると思われる。

5.2 聴覚特性に基づくサラウンド情報の有用性

次に、聴覚特性に基づくサラウンド情報を定義することによりどの程度の情報量が削減できたのか、図7に示す。図7では、楽器の速度が若干変化していくことを想定し、従来手法と提案手法でサラウンド情報を埋め込んだ例を示

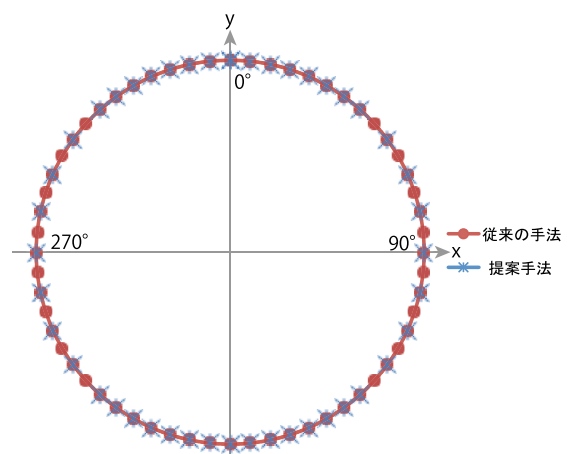


図7 サラウンド情報の埋め込み例  
Fig. 7 Example of information for surround.

した。このとき、従来手法では楽器の速度変化を想定し、円周を60分割し、6度ごとにサラウンド情報を埋め込んだ例を示している。また、提案手法として新しく定義したサ

表 10 被験者・楽器別の誤差 (後方 2)  
Table 10 Margin of errors in listening (back case2).

回数	楽器名						
	トランペット	トロンボーン	フルート	ファゴット	ヴィオラ	チェロ	
被 験 者 D	1	10	10	-5	0	15	-5
	2	0	0	-15	-5	0	0
	3	0	-10	0	5	-2	-10
	4	0	15	10	-25	10	30
	5	10	-5	5	0	5	0
	6	0	3	10	10	3	0
	7	5	0	-5	-10	0	15
	8	15	8	10	0	16	-7
	9	0	5	-10	5	6	-10
	10	7	3	5	-10	-5	3
被 験 者 E	1	0	-4	15	-15	15	-15
	2	-5	0	4	-5	0	-5
	3	5	-15	0	0	-4	-30
	4	15	0	0	55	-19	2
	5	-15	10	-15	20	-15	-21
	6	0	10	0	15	0	15
	7	15	-5	15	-13	-5	5
	8	2	0	15	15	2	-15
	9	-15	30	11	15	-7	3
	10	-16	5	5	12	1	24
標準偏差	6.62	7.36	7.86	9.72	6.47	10.28	

表 11 提案するサラウンド情報  
Table 11 Proposal information for surround music.

	備考	データサイズ
Bit $x$	角度の bit 数 (1 回のみ利用)	2 bit
$(r_t, AZI\theta_t)$	時刻 $t$ における座標: リスナを中心とした半径 (9 bit), 角度 (5-7 bit)	16 bit
$ELE\theta_t$	仰角	8 bit
$\vec{r}_t$	半径方向の速度	11 bit
$\vec{\omega}_t$	角速度	11 bit
$E\vec{L}\dot{V}_t$	仰角の変化率	11 bit
$t$	時刻	8~96 bit

ラウンド情報を用いて聴覚特性を考慮した埋め込みを行った例を示した。提案手法では、表 12 に示したとおり、横方向においては人間の聴覚特性上区別が難しいため、表 13 で示したサラウンド情報を利用し埋め込む情報量を削減している。この例では、図 7 より提案手法は 26%の情報量削減を実現しており、聴覚特性に基づくことで効率の良い埋め込みが実現できたと考えられる。削減された分のデータを補完して再生する際、音源の位置が正確な位置からずれていたとしても、表 12 より、被験者の側方にある音源の位置は正確に特定しにくく、ずれによる影響は少ない。逆に、同一のデータ量で埋め込みを行う場合、本提案方式では被験者の前方や後方に多くの音源位置の情報を割り当てられるため、従来方式より正確に音源の位置を表現できる。

## 6. おわりに

本稿では、昨年度提案したリアルタイムで位置情報を入力できるサラウンド MIDI シーケンサの開発 [2] の実装を行い、設定したパラメータが妥当なものであるか評価を行った。3D MIDI の定義が、MIDI Manufacturers Association によって 3D Sound Controller として公表されたが、本提案では独自のサラウンド情報の実装例として、MIDI データハイディングの技術を応用し、サラウンド情報を埋め込んでいる。これにより、本来サラウンドに対応していない MIDI ファイルにサラウンド化に必要な情報を埋め込むことができ、既存の MIDI 機器、シーケンサ、再生装置を変更することなく利用することができる。また評価の結果より聴覚特性を考慮することで、3D Sound Controller のパ

表 12 方位・被験者別の誤差における平均偏差  
Table 12 Average margin of errors in listening.

方位	被験者	トランペット	トロンボーン	フルート	ファゴット	ヴァイオラ	チェロ	平均
前方	被験者 A	9.60	6.00	6.00	4.50	4.50	4.50	5.85
	被験者 B	3.87	3.52	5.97	3.38	5.95	3.82	4.42
	被験者 C	5.68	1.50	1.50	1.20	1.50	2.24	2.27
	被験者 D	7.35	5.21	3.44	6.57	5.50	5.82	5.65
	被験者 E	7.50	6.00	6.00	4.50	4.50	4.50	5.50
	平均	6.80	4.45	4.58	4.03	4.39	4.18	4.74
	上位 40%	4.78	2.51	3.74	2.29	3.73	3.03	3.34
	上位 20%	5.68	1.50	1.50	1.20	1.50	2.24	2.27
側方	被験者 A	13.08	17.81	24.00	11.22	16.43	21.00	17.26
	被験者 B	7.78	3.79	8.38	9.18	9.54	10.91	8.26
	被験者 C	8.50	6.80	9.49	14.15	21.00	6.87	11.14
	被験者 D	10.32	14.43	7.68	7.67	7.35	13.57	10.17
	被験者 E	7.68	8.24	9.16	11.48	17.79	7.85	10.37
	平均	9.47	10.21	11.74	10.74	14.42	12.04	11.44
	上位 40%	9.05	9.11	8.03	8.42	8.44	12.24	9.22
	上位 20%	7.78	3.79	8.38	9.18	9.54	10.91	8.26
後方	被験者 A	4.50	9.00	8.08	10.06	6.71	6.71	7.51
	被験者 B	6.42	6.25	7.72	5.33	4.04	9.33	6.51
	被験者 C	6.00	3.28	6.00	4.50	5.92	7.75	5.57
	被験者 D	5.27	6.88	8.50	9.54	6.71	11.76	8.11
	被験者 E	10.91	11.38	9.01	19.19	8.96	15.87	12.55
	平均	6.62	7.36	7.86	9.72	6.47	10.28	8.05
	上位 40%	6.21	4.77	6.86	4.91	4.98	8.54	6.04
	上位 20%	5.64	3.28	6.00	4.50	5.92	7.75	5.57

表 13 結果別角度の表現手法

Table 13 Proposal azimuth angle for surround.

角度の範囲	1 bit あたりの度数		
	平均	上位 40%	上位 20%
315-45 (前方)	4	3	2
45-135 (右方)	11	9	8
135-225 (後方)	8	6	5
225-315 (左方)	11	9	8

ラメータより少ないビット数で効率の良いサラウンド情報の埋め込みができたと考えられる。本稿では、一例として文献 [2] のデータハイディング手法を用いて実装を行ったが、本手法に適用可能なデータハイディング手法はこれにとどまらず、他のデータハイディング手法を用いて行うことも可能である。ただし、文献 [3], [4], [5], [6] で述べられているステガノグラフィの手法では、オリジナルの MIDI の演奏情報を聴覚上認知可能な程度に変更しているため、文献 [2] で述べられているような、オリジナルの MIDI の演奏情報を変更しないデータハイディング手法を選択する必要がある。

参考文献

- [1] 松本圭祐, 宮田宙和, 宇田隆哉: MIDI ファイルのピッチベンド・チェンジコードを対象とした新たな情報埋め込み方式の提案および評価, 情報処理学会 DICO2008 論文集, pp.1763-1769 (2008).
- [2] 松本圭祐, 宮田宙和, 宇田隆哉: リアルタイムで位置情報を入力できるサラウンド MIDI シーケンサの開発, 情報処理学会 DICO2009 論文集, pp.1198-1205 (2009).
- [3] 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.8, pp.2489-2501 (2002).
- [4] 山本鉦太郎, 岩切宗利: テンポへの表情付けを考慮した楽音符号への情報ハイディング, 電子情報通信学会技術研究報告 WIT, 福祉情報工学, Vol.106, No.144, pp.13-18 (2006).
- [5] 山本鉦太郎, 岩切宗利: 表情付けを考慮した SMF ステガノグラフィ, 情報処理学会論文誌, Vol.47, No.8, pp.2724-2732 (2006).
- [6] 山本鉦太郎, 岩切宗利: 発音時間の揺らぎを利用した SMF ステガノグラフィ, 情報処理学会 CSS2007 論文集, pp.133-138 (2007).
- [7] 松本圭祐, 上田英司, 宮田宙和, 宇田隆哉: 音程を変化させる編曲が可能な SMF ステガノグラフィの提案および評価, 情報処理学会 CSS2008 論文集, pp.911-916 (2008).
- [8] Miyata, H., Uda, R. and Matsushita, Y.: Digital Signature Scheme for Flexible Usage of a MIDI File, *IASTED International Conference NCS2006*, pp.346-351 (2006).
- [9] Miyata, H., Uda, R. and Okada, K.: The Efficiency Method of Sanitizable Signature Scheme for SMF, *AINA 2009*, pp.495-502 (2009).

## 推薦文

本論文は、MIDI を拡張しサラウンドに対応させる手法の提案を行っている。提案手法は、3D MIDI の実現に MIDI ステガノグラフィを応用することで、従来の 3D Sound Controller よりも少ない情報量で実現できることを示した点が非常に有効な方法である。また、将来のマルチメディアアプリケーションの開発やコンテンツの制作において有用性が高いと考えられる。

(マルチメディア通信と分散処理研究会主査 勝本道哲)



### 黒岩 謙

2010 年東京工科大学コンピュータサイエンス学部卒業。2012 年同大学大学院バイオ・情報メディア研究科コンピュータサイエンス専攻博士前期課程修了。2009 年情報処理学会 DICOMO2009 シンポジウムにおいて優秀論文賞，2010 年情報処理学会 DICOMO2010 シンポジウムにおいてヤングリサーチ賞，ならびに優秀論文賞を受賞。現在，KDDI 株式会社勤務。



### 宮田 宙和

2006 年東京工科大学工学部情報工学科卒業。2008 年同大学大学院バイオ・情報メディア研究科コンピュータサイエンス専攻博士前期課程修了。現在，慶應義塾大学大学院理工学研究科開放環境科学専攻後期博士課程在学中。情報セキュリティの研究に従事。



### 宇田 隆哉 (正会員)

1998 年慶應義塾大学理工学部計測工学科卒業。2000 年同大学大学院理工学研究科計測工学専攻前期博士課程修了。2002 年同大学院理工学研究科開放環境科学専攻後期博士課程修了。博士 (工学)。現在，東京工科大学コンピュータサイエンス学部講師。ネットワークセキュリティの研究に従事。2002 年 IFIP/SEC 2002 Best Student Paper Award 受賞。電子情報通信学会会員。