

数値データと音との対応付けを試行するための環境の提案

城 一裕[§], 山本 恭裕[§], 中小路 久美代^{§†}

概要: 本論では, 数値データと音との対応付けを試行するための環境を提案する. 本研究の目的は, 音に関する専門家以外のユーザが, 数値データの理解を目的として音を利用できるようにすることである. 本環境を構築するにあたって, 音表現の生成を目的としてデータを扱う部分と, データ表示を目的として音を生成する部分との 2 つを分離するアプローチをとった. そしてこれらを分離するために, データの数値を音として取り扱えるような数値へと変換する部分を取り入れた. 本論では, これら 3 つの部分の詳細を説明し, ケーススタディとして筆のストロークデータと音との対応付けを 4 種試行したものを示す.

An Environment for Early Stages of Auditory Interaction Design

Kazuhiro JO[§], Yasuhiro YAMAMOTO[§], Kumiyo NAKAKOJI^{§†}

Abstract: We propose an environment for early stages of auditory interaction design. The goal of the project is to help non-sound designers to use sound as an expressive means for understanding streams of various kinds of numerical data. Our approach is to separate the part that handles data from the data that produces a variety of sound representations so that sound variations can be incrementally added without worrying about the data formats to deal with. In order to ensure such domain independency, our architecture introduces the transmission part, which adjust the characteristics of the data to match with the characteristics of the input values that are necessary to produce appropriate sound. This paper explains the details of each part and describes the four cases where we produced sound representation for real-time stroke data.

1 はじめに

我々はこれまでに情報提示手段としての音の利用について研究を行ってきた [城 05, 城 04]. 音を情報提示手段として使うシステムには, 地震震動を音の大きさに伝えるシステム [Hayward 94] や, 歩行リズムを音で知らせる歩行介助システム [高梨 03], パフォーマの動きに伴う加速度の変化を音に変える音楽表現のためのシステム [Jo 07] などがある. このようなシステムを開発するにあたっては, 対象のデータをどのような音に対応づけるか, を決めていく必要がある.

これまでのデータと音とを対応づけるツールには, 既存の音生成用のプログラミング環境上で, 数値の正規化や, 間引き, 補完を容易に行えるようにするもの [Polli 04] や, データを西洋音楽のスキーマに基づいて音に対応づけて再生するもの [Joseph 02,

Walker 03], 科学技術計算用のライブラリ [Matlab] などがある.

また, 既存の音を生成するためのツールの多くは, 楽器ごとの音量を調節したり, 録音された音にエフェクトを加えたり, 音の圧縮率を変更したり, といった最終的なフォームを表現することを主目的として構築されている [中小路 04].

しかし, これらのツールのほとんどは, プログラミング環境そのものや, 西洋音楽に対する専門的な知識を必要とするものであり, 専門家以外の人がデータと音との様々な対応付けを試行することをサポートするものは少ない.

例えば, 紙に描くスケッチでは, 明確なイメージが必ずしも存在しない状態で, まず手が動き, 紙の上に線が描き出される. そこから得られる外在化表現を「見る」ことによって, 何を描きたいか, という内省が進む [中小路 04]. このことと同様に, データと音

[§] 東京大学先端科学技術研究センター
Research Center for Advanced Science and Technology, University of Tokyo
[†] 株式会社 SRA 先端技術研究所
SRA Key Technology Laboratory Inc.

との対応付けの明確なイメージが必ずしも存在しない状態で、様々な対応付けを試行して「聴く」ことによって、データをどのような音として表示したいのか、という内省が進むのでは考えている。

これまでにタンジブルユーザインタフェースやフィジカルコンピューティング、拡張現実感の分野において、いくつかのツールキットが開発されている。これらのツールキットは、無線タグ [Klemmer 04]や、コンピュータビジョン [MacIntyre 04]、アナログセンサー [Greenberg 01]など、各々の分野の基盤となる技術へのアクセスを外部から利用可能な API として共通化し、各々の技術の専門家以外がそれらの技術を利用することを可能にしている。

本研究も同様に、音に関する専門家以外のユーザが、数値データの理解を目的として音を利用できるようにすることを目的としている。すなわち、我々の提案する、数値データと音との対応付けを試行するための環境を提供することで、コンピュータ音楽 [Roads, 01]や、音響心理学 [Moore 94]、生態音響学 [Gaver 93]に関する専門知識がなくとも、ユーザがデータ表示のために様々な音を利用することを可能にしようとするものである。

このようなツールキットの作成にあたっては、専門家にインタビューを行ったり [Klemmer 04]、デザイナーと協業する [MacIntyre 04]、実際にそのツールキットを用いてその都度問題を解決していく [Greenberg 01]などのアプローチがある。

これに対して我々のアプローチは、表示するための数値データと、表示のために利用する音表現との独立性を保ち、ユーザが理解の対象とするデータを独立して扱えるように、音表現の生成を目的としてデータを扱う部分と、データ表示を目的として音を生成する部分との2つを分離するものである。そしてこれらを分離するために、データの数値を音として取り扱えるような数値に変換する仕組みを取り入れた。

以下本論では、提案する数値データと音との対応付けを試行するための環境について説明し、ケーススタディとして筆のストロークデータと音との対応付けを試行したものを示す。

2 数値データと音との対応付けを試行するための環境

本論で提案する数値データと音との対応付けを試行するための環境では、音に関する専門家以外のユーザが、データの理解を目的として音を利用できるようにすることを目的としている。そのために、音表現の生成を目的としてデータを扱う部分と、データ表示を目的として音を生成する部分との2つを分離するアプローチをとっている。データを扱う部分では、任意の数値データの表示を行えるよう、データの範囲、個数、時間間隔、には制限を設けず、時間軸に沿った数字列として取得できる全てのデータを取り扱う。また、データ表示を目的として音を生成する部分では、多様な音表現を可能にするよう、ピアノから犬の鳴き声、車のエンジンまで様々な音を、それぞれの音に適したパラメータを用いて操作できるようにしている。

各々のデータは、それぞれが異なる範囲、個数、時間間隔をもっており、各々の音のパラメータも、それぞれ異なる範囲、個数、時間間隔の値を受け付ける。そのため、あるデータのある音に対応付けようとした場合、単にデータの値を音のパラメータに割り当てれば、データを音で表示できる、という訳にはいかない。そこで、我々は、それぞれ異なる範囲、個数、時間間隔をもつデータの数値を、音のパラメータに変換するための仕組みを変換部として提供する。これらの仕組みを用いて、ユーザは任意のデータを任意の音に対応づけることができると考えている。

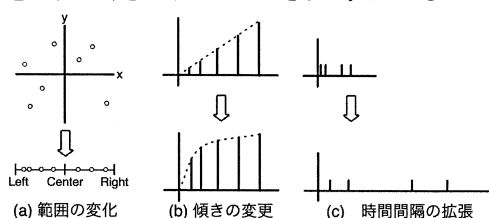


図1: 値の変換

例えば、値の範囲を変化させることで、2次元平面上のx座標を、音の定位に対応づけたり [図1(a)], 値の傾きを変えることで、線形に変化する値を対数尺度をもつ音の高さに対応づけたり [図1(b)], 時間間隔を拡張することで、数ミリ秒間隔で出力される値を、数10ミリ秒間隔に遅らせる [図1(c)], ということが

できる。

図 2 に、データ部、変換部、音生成部、の 3 つの部分から構成される本環境を示す。

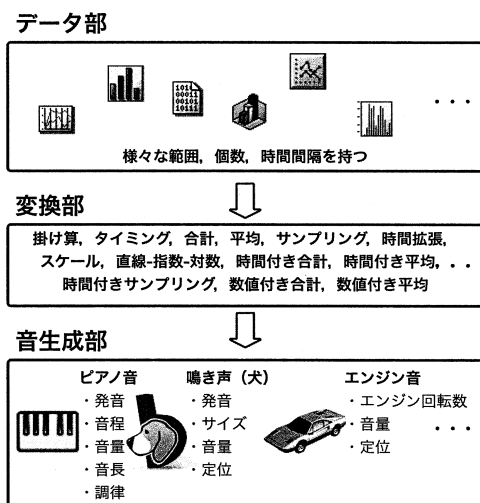


図 2: 数値データと音との対応付けを試行するための環境

データ部では、時間軸に沿った数字列として取得できる全てのデータを取り扱う。データの範囲、個数、時間間隔、には制限を設けていない。ユーザは任意のひとつまとまりのデータを一つのデータセットとして定義することができる。

変換部では、それぞれ異なる範囲、個数、時間間隔をもつデータの値を音のパラメータに変換するための仕組みを提供する。

音生成部では、ピアノから犬の鳴き声、車のエンジン音まで様々な音の選択肢を提供する。我々は、各々の音をサウンドプリミティブとして統一的に扱い、ピアノであれば音程、サイン波であれば周波数、というように各々の音に適したパラメータを定義できるようにしている。ユーザは、データ表示に利用するための音として任意のサウンドプリミティブを選択できる。

このように、本環境を3つの部分にわけ、表示するためのデータと、表示のために利用する音表現を独立に取り扱えるようにすることで、データと音との対応付けの明確なイメージが必ずしも存在しない状態で、様々な対応付けを試行して「聴く」こと、またそのことによって、データをどのような音として表示したいのか、という内省が進むこと、が可能になると考えている。

以下これら 3 つの部分の詳細を説明する。なお、現在本環境は、音生成用のビジュアルプログラミング環境である MaxMSP [Cycling74] 上に構築している。

3 各部分の詳細

3.1 データ部

データ部は、データを時間軸に沿った数字列として取得する。いくつかのデータを一まとまりにしたものをデータセットとして定義できるようにしている。例えばマウス位置を扱うのであれば、x 座標、y 座標、スピード、加速度などを入力データとして、マイクロフォンであれば、音圧、ピークの周波数、などを入力データとして、取り扱うことが考えられる。以下本論では、データセットの例として DiamondTouch から出力されたタイムスタンプ付きの座標値を入力データとして使用した場合で説明する。

DiamondTouch は、接触型の平面入力デバイスである DiamondTouch [Dietz 01] から出力されたタイムスタンプ付きの座標値をもとにしたデータ [表 1] から構成される。

表 1: DiamondTouch のデータ

データ	説明
x	接触点の x 座標。1 - 2048 までの整数値を持つ。
y	接触点の y 座標。1 - 1576 までの整数値を持つ。
distance	現在の接触点と直近の接触点の 2 点間の距離。単位は point。
speed	現在の接触点と直近の接触点の 2 点間の移動速度。単位は point/ms。
size	現在の接触点の接触面積。単位は point ² 。
direction	直近の接触点と現在の接触点とを結ぶ直線の傾き。x 軸正方向を 0 として、時計回りで徐々に大きな値をとる。単位はラジアン。
cos-direction	direction の余弦。
down	接触開始を示す。平面への接触を検知した際に 0 を出力する。
up	接触終了を示す。平面からの離脱を検知した際に 0 を出力する。
down/up	down と up の組み合わせ。平面への接触時に 1、平面からの離脱時に 0 を出力する。

なお、DiamondTouch は、平面への接触を検知している間は、およそ 33ms ごとに down と up を除く全ての値を出力する。

3.2 音生成部

音生成部は、様々な種類の音の選択肢をサウンドプリミティブとして提供する。現在までに以下の 26 個のサウンドプリミティブを用意している [表 2].

表 2: サウンドプリミティブ一覧

• エンジン音	• MIDI ファイルプレイヤー
• 自転車チェーン	• 単旋律
• 鼓動音	• サイン波
• 時計振り子	• ドップラー
• 足音	• 無限音階
• ピアノ音	• 干渉-A
• FM 音	• 干渉-B
• メトロノーム	• ブルースバンド
• ガラスコップ	• ローデータ
• ドアクローズ	• ポインター
• 鳴き声(馬)	• 4ch(循環型)ミキサー
• 鳴き声(犬)	• ノイズフィルタ
• 鳴き声(鷹)	• 水滴

各々のサウンドプリミティブに対して、それぞれの音に適した形で値の変化を音の変化へと対応づけるためのパラメータを定義している。なお、全てのサウンドプリミティブに対して、on/off のスイッチを設けて、音の出力を個別に制御できるようにしている。この中から例えば、値間の時間間隔を表すのであれば鼓動音の速度、ある範囲で変化する値の範囲内でのおおまかな位置を表すのであればピアノ音の音程、連続的な値の変動を表すのであればエンジンのエンジン回転数、値が断続的に来るタイミングを表すのであれば単旋律の発音、周期性を持つ値を表すのであればポインターの基数、値の変化する方向を表すのであれば無限音階の基数、同期した複数の値の変化を表すのであればブルースバンドの各楽器の音量、などを利用することができる。

以下に、構築したサウンドプリミティブの中から特徴的なものをいくつか説明する。

エンジン音プリミティブ

エンジン音プリミティブは、エンジン回転数、音量、定位の 3 つのパラメータを持っている。エンジン回転数は 0.0 以上の全ての値を受け付け、値が大きくなるにつれてエンジンの回転数が増していくような音がする。値とエンジンの回転数との間に明確な等式はないが、値が 0.0 から 200.0 の範囲にあるとエンジンのような音として聞こえる。音量は、0.0 から 1.0

までの値を受け付け、値が大きくなるにつれて音量が増加する。定位は、-1.0 から 1.0 までの値を受け付け、値が大きくなるにつれて音像は左から右に移動する。値が 0 のときは音は中央から聞こえる。同様なプリミティブに、自転車チェーンの音を生成する自転車チェーンプリミティブがある。

ピアノ音プリミティブ

ピアノ音プリミティブは、発音、音程、音量、音長、調律の 5 つのパラメータを持つ。発音は、全ての値を受け付け、値の変化を検出すると、ピアノ音を生成する。音程は、0 から 127 までの整数値を受け付け、12 音階に基づき、音の高さを決定する。小数値を入力した場合は四捨五入される。60 が通常の 88 鍵盤のピアノの中央のドの音(C4)と同じである。値が大きくなるにつれ、音の高さが増す。音長は 0. 以上の値を受け付け、値が大きくなるにつれて、ピアノ音の持続時間が長くなる。単位はミリセカンドである。調律は、0 か 1 の値を受け付け、下記の設定に基づき、ピアノの音階を決定する。

0 = 12 音階, 1 = ハ長調

ハ長調では、ハ長調の音階に適合するように、音程に入力された値を丸め込む。黒鍵にあたる値が入力された場合は全て直下の白鍵の値に置き換えられる。音量はエンジン音プリミティブと同様である。同様に値の変化を検出すると音を生成するものに、FM 音プリミティブ、ガラスコッププリミティブ、ドアクローズプリミティブ、鳴き声プリミティブ、ドップラープリミティブがある。

メトロノームプリミティブ

メトロノームプリミティブは、テンポ、周波数、音長の 3 つのパラメータを持っている。テンポは 0.0 以上の値を受け付け、値が大きくなるにつれて音(サイン波)を生成する間隔が長くなる。単位はミリセカンドである。周波数は 0.0 以上の値を受け付け、サイン波の周波数を決定する。単位は Hz である。人の可聴域はおおよそ 20-20000Hz までであるが [小橋 69], それ以上もしくはそれ以下の値も受け付ける。音長はピアノ音プリミティブと同様である。単位はミリセカンドである。音の重なりを防ぐためには、音長の値はテンポの値よりも小さくする必要がある。

MIDI ファイルプレイヤープリミティブ

MIDI ファイルプレイヤープリミティブは、テンポ、

音量、リセット、ファイル読み込み、の4つのパラメータを持ちMIDIファイルを再生する。テンポは、0.0 以上の値を受け付け、値が大きくなるにつれて、再生するファイルのテンポが長くなる。単位は BPM(Beats Per Minute)である。リセットは全ての値を受け付け、値の変化を検出するとファイルの再生箇所を最初に戻す。なおファイルが最後まで再生された場合も、ファイルの再生箇所は最初に戻る。ファイル読み込みをクリックすることで、MIDIファイルの読み込みを行う。ユーザは任意の MIDI ファイルを再生することができる。音量はエンジン音プリミティブと同様である。

3.3 変換部

変換部は、データの数値を音のパラメータに変換するための仕組み(以下「アダプタ」と呼ぶ)を提供する。現在までに以下の 13 個のアダプタを開発している [表 3]。

表 3: アダプター一覧

・掛け算	・直線-指数-対数
・タイミング	・時間付き合計
・合計	・時間付き平均
・平均	・時間付きサンプリング
・サンプリング	・数値付き合計
・時間拡張	・数値付き平均
・スケール	

ユーザは、複数のアダプタを組み合わせ、データの数値を音のパラメータに変換することができる。本節では、各々のアダプタの特徴を説明し、具体的な使用例を DiamondTouch と前述のサウンドプリミティブを例にとり示す。

掛け算アダプタ

掛け算アダプタは、基数と乗数の 2 つの入力を持ち、それらを掛け合わせた結果を出力する。基数、乗数とも全ての値を受け付ける。例えば、DiamondTouch の x 座標の値の変化を、ピアノ音プリミティブの音程の変化に対応付けて聴きたい場合は、基数に 1 から 2048 までの範囲で変化する整数値である x を入力し、常数に 0.06201172 (127/2048) を入力すれば、ピアノ音プリミティブの音程が受け付ける、0 から 127 までの範囲で変化する値を得ることができる。

タイミングアダプタ

タイミングアダプタは、スタートとストップの 2 つの

入力を持つ、スタート、ストップとも全ての値を受け付け、両者に入力した値の間の時間間隔を出力する。出力の単位はミリ秒である。例えば、DiamondTouch への接触時間を、メトロノームプリミティブのテンポに対応付けて聴きたい場合、down をスタート、up をストップに入力し、その出力をテンポに割り当てることで、DiamondTouch 表面に触れてから離れるまでの経過時間に沿って、メトロノームプリミティブのテンポを変化させることができる。

時間拡張アダプタ

時間拡張アダプタは、基数と倍数、リセットの 3 つの入力を持つ。基数は全ての値を受け付け、倍数は 1. 以上の値を受け付ける。基数に入力した値の時間間隔を、倍数に入力した値にそって拡張する。リセットは全ての値を受け付け、値を入力すると、時間間隔の拡張を停止する。例えば、DiamondTouch の speed の変化をエンジン音プリミティブのエンジン回転数に対応づけてゆっくりと聴きたい場合は、基数におよそ 33ms 毎に出力される speed、倍数に 33 を入力し、その出力をエンジン回転数に割り当てることで、およそ 1 秒間に 1 回の割合でエンジン音の回転数を変化させることができる。

スケールアダプタ

スケールアダプタは、基数、入力(最小値)、入力(最大値)、出力(最小値)、出力(最大値)の 5 つの入力を持ち下記の式にそって値の範囲を変換する。

$$\text{出力} = (\text{基数} - \text{入力(最小値)}) * (\text{出力(最大値)} - \text{出力(最小値)}) / (\text{入力(最大値)} - \text{入力(最小値)}) + \text{出力(最小値)}$$

例えば、DiamondTouch の x 座標の値の変化を、エンジン音プリミティブの定位の変化に対応付けて聴きたい場合は、基数に x を入力し、入力(最小値)に 0. , 入力(最大値)に 2048, 出力(最小値)に -1. , 出力(最大値)に 1. を入力することで、エンジン音プリミティブの定位が受け付ける、-1.0 から 1.0 までの範囲で変化する値を得ることができる。

直線-対数-指数アダプタ

直線-対数-指数アダプタは、基数と傾きの 2 つのパラメータを持つ。基数は 0. から 1. までの値を受け付ける。傾きは、直線、対数、指数、の 3 つの選択肢を持ち、選択された傾きにそって、基数に入力した値を出力する。例えば、DiamondTouch の y 座標の値の変化を、エンジン音プリミティブの音量の変化に

対応付けて聴きたい場合は、まず掛け算アダプタで、y座標の値の範囲を0. から1.に変更した後に、その出力を基数に入力し、傾きとして対数を選び、その出力を音量に割り当てることで、y座標の値の変化に沿って、エンジン音プリミティブの音量を聴感上の感覚の尺度に近い対数軸 [Roads 01]に沿って変化させることができる。

合計アダプタ

合計アダプタは、基数とリセットの2つの入力を持ち、基数に入力した値の合計を出力する。基数、リセットとも全ての値を受け付ける。リセットに値を入力すると出力は0となる。例えば、DiamondTouchの表面に触れている間の移動距離の合計を、ピアノ音プリミティブの音の長さに対応付けて聴きたい場合は、distanceを基数に、downをリセットに入力し、その出力を音長に割り当てることで、DiamondTouchに触れた際の一回ごとの移動距離の合計を、ピアノ音の鳴っている時間として聴くことができる。

平均アダプタ

平均アダプタは、基数とリセットの2つの入力を持ち、基数に入力した値の平均を出力する。基数、リセットとも全ての値を受け付ける。リセットに値を入力すると出力は0となる。例えば、DiamondTouchの表面に触れている間の移動速度の平均を、メトロノームプリミティブのテンポに対応付けて聴きたい場合は、speedを基数に、downをリセットに入力し、その出力をテンポに割り当てることで、DiamondTouchに触れた際の一回ごとの移動速度の平均の変化を、メトロノームのテンポの変化として聴くことができる。

サンプリングアダプタ

サンプリングは、基数と分母の2つの入力を持つ。基数は全ての値を受け付け、分母は1以上の整数値を受け付ける。基数に入力した値を、分母に入力した値に基づき間引いて出力する。例えば、例えば、DiamondTouchの表面への接触を10回に1度、ピアノ音プリミティブに対応づけて聴きたい場合には、基数にdown、分母に10を入力し、その出力を発音に割り当てることで、10回の接触に付きに1度ピアノ音を鳴らすことができる。

時間付きアダプタ群

時間付き平均アダプタ、時間付き合計アダプタ、時間付きサンプリングアダプタは、それぞれ、前述

の平均アダプタ、合計アダプタ、サンプリングアダプタに、時間の入力を付け加えたものである。与えられた時間に一度値を出力する。

数値付きアダプタ群

数値付き平均アダプタ、数値付き合計アダプタは、それぞれ、前述の平均アダプタ、合計アダプタに、平均数、合計数の入力を付け加えたものである。与えられた数値分の、平均、合計を出力する。

4 ケーススタディと考察

これまでに、3.1で説明したDiamondTouchを用いた筆のストロークデータと音との対応付けを試行してきた。4本の絵筆を用意し、DiamondTouchの盤上に配置し画用紙に絵筆で描くストローク [図3]のデータを、それぞれ、エンジン音、サイン波、MIDIファイル再生、FM音の4種類のサウンドプリミティブと対応づけるツールを試作した。試作したツールでは、ユーザがストロークの描き方を変えると、それに応じて各々のサウンドプリミティブのパラメータが変化し、鳴る音が変わる。以下に、ストロークデータと4種類のサウンドプリミティブとの対応付けに関して、どのようなデータを選択し、どのような変換を行い、どのようなパラメータに対応付けたかを説明する。



図3: 絵筆でストロークを描くところ

4.1 筆先の速度とエンジン音

筆を速く動かすと、エンジン音の回転数が上がっていきような音ができるような対応を作った。実際に紙に絵筆で描いてみたところ、速度であるspeedの範囲はおおよそ0.0から10.0あたりであることがわかった。そこで、この対応付けでは、掛け算アダプタを用いて、値を100倍にして、エンジン音の回転数に割り

当てた。さらに、描く方向に応じて音の定位が変化するように、直近の接触点と現在の接触点とを結ぶ直線の傾きの余弦である $\cos\text{-direction}$ を定位に割り当てた。 $\cos\text{-direction}$ の値は-1.0 から 1.0 の範囲で変化するが、定位も-1.0 から 1.0 までの範囲の値を受け付けるため、ここではデータの変換は行っていない。また、down/up をスイッチに割り当てることで、描いているときだけ音が出力されるようにした。この対応付けを図 4 に示す。

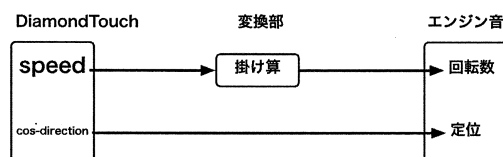


図 4: 筆先の速度とエンジン音の対応付け

4.2 筆先の位置とサイン波

右から左に描くにつれ音が高くなり、下から上に描くにつれ音が大きくなるように、筆の絶対 x 座標をサイン波プリミティブの周波数、絶対 y 座標を音量に対応づけた。この対応付けでは、筆先の絶対座標に基づいてサイン波の周波数と音量を変化させるために、図 5 に示すように変換部を用いた。

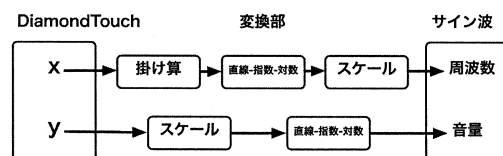


図 5: 筆先の位置とサイン波の対応付け

4.3 筆先の速度とMIDIファイル再生

筆を速く動かすとテンポが速くなり、ゆっくり動かすとテンポが遅くなるように、筆先の速度を MIDI ファイル再生プリミティブのテンポに対応づけた。再生するファイルとしてはバッハのピアノ曲を選択した。この対応付けでは、図 6 に示すように変換部を用いた。

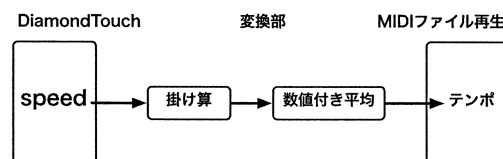


図 6: 筆先の速度と MIDI ファイル再生の対応付け

4.4 ストロークを描いた時間、距離とFM音

ゆっくりと長いストロークを描くと、長くかつ高い音が鳴り、いそいで短いストロークを描くと、短くかつ低い音が鳴るように、一つのストロークを描いた時間と FM 音プリミティブの音長、描いた距離を周波数、に対応づけた。この対応付けでは、図 7 に示すように変換部を用いた。

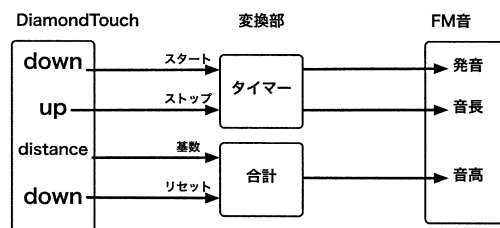


図 7: ストロークを描いた時間、距離と FM 音の対応付け

4.5 考察

本論では、ストロークを描く速度をエンジン音の回転数や MIDI ファイル再生のテンポに割り当てることで、絵筆の動きの緩急を音で表すことを試してみるための環境を報告した。本環境を利用することで、例えば書道の筆の運びやゴルフのスイングといった人の動きを、音に対応づけて聴かせるツールを比較的に、サウンドデザインの専門家でなくとも作れることを目指している。

音を取り扱うには、ラップトップコンピュータに内蔵されているスピーカと、外部接続型のモニタースピーカの、2種類の出力デバイスを用いて音を確認したが、内蔵スピーカでは聴き取りづらい、エンジン音の定位の変化や、サイン波の音量の違い、FM 音の低い音を、モニタースピーカでははっきりと聴き取ることができた。このような、出力デバイスに関わる、音像定位、ダイナミックレンジ、周波数の再生帯域、といった要素は、今後の研究課題となる。

提案した環境の応用例として、例えば、パフォーマンスの描いた絵にあわせて音楽が生成される、というようなある種の楽器的な用途を考えることもできる。ただし、その場合には、科学者はデータをいかにわかりやすく表示するか、に重きを置くが、音楽家はデータと音との関係をいかに直接的ではなく聞かせるか、に重きを置く [Gibson 06] ということが指し示すように、各々のサウンドプリミティブの内部での音の

細かなチューニングが必要だと考える。

5 おわりに

本論では、音に関する専門家以外のユーザが、数値データの理解を目的として音を利用できるために、データと音との対応付けを試行するための環境を提案した。本環境の構築にあたっては、表示するためのデータと、表示のために利用する音表現との独立性を保ち、ユーザが理解の対象とするデータを独立して扱えるように、音表現の生成を目的としてデータを扱う部分と、データ表示を目的として音を生成する部分との2つを分離するアプローチをとった。そしてこれらを分離するために、データの数値を音のパラメータに変換する仕組みを取り入れた。

このデータの数値を音のパラメータに変換するための仕組みとして、13個のアダプタについて説明した。環境の構築に際し、我々が実際に本環境上でデータと音との対応付けを試行することで、どのようなアダプタがデータの数値を音のパラメータに変換する為に必要なのかは、継続して検討している。

現在は、テスト用のデータとしてDiamondTouchを用いている。今後、他のデータを用いてデータと音との対応付けを試行することで、さらなるアダプタの充実はほかりたい。

6 謝辞

本環境の構築に関して助言をいただいた西中芳幸氏、浅田充弘氏、および本論文の執筆にあたりコメントをいただいた白井良成氏に感謝する。本研究は文部科学省科学研究費基盤研究(A)(1)16200008のもとに実施されたものである。

参考文献

- [Cycling74] Cycling74, MaxMSP, <http://www.cycling74.com/>
- [Dietz 01] P. Dietz, and D. Leigh, DiamondTouch: A Multi-User Touch Technology, In Proceedings of UIST 2001, pp.219-226 (2001)
- [Gaver 93] W. W. Gaver, What in the World Do We Hear?: An Ecological Approach to Auditory Event Perception. Ecological Psychology, 5, pp.1-29 (1993)
- [Gibson 06] J. Gibson, sLowlife: "Sonification of Plant Study Data," Leonardo Music Journal 16, pp.42-44 (2006)
- [Greenberg 01] S. Greenberg, C. Fitchett, Phidgets: easy development of physical interfaces through physical widgets. In Proceedings of the 14th Annual ACM Symposium on User interface Software and Technology, UIST '01, ACM Press, New York, NY, pp. 209-218 (2001)
- [Hayward 94] C. Hayward, Listening to the earth sing, In G. Kramer (Ed.), Auditory display: Sonification, audification, and auditory interfaces, Reading, MA: Addison Wesley, pp.369-404 (1994)
- [Jo 07] K. Jo, Design transition of an instrument: in the case of sound performance project "aeo", Leonardo Music Journal No.17, (2007) (to be published)
- [Joseph 02] A. Joseph, S. K. Lodha, MUSART: Musical audio transfer real-time toolkit, In Proceedings of International Conference on Auditory Display (2002)
- [Klemmer 04] S. R. Klemmer, J. Li, J. Lin, and J. A. Landay, Papier-Mache: toolkit support for tangible input. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '04, ACM Press, New York, NY, pp.399-406 (2004)
- [MacIntyre 04] B. MacIntyre, M. Gandy, S. Dow, and J. D. Bolter, DART: a toolkit for rapid design exploration of augmented reality experiences. In Proceedings of the 17th Annual ACM Symposium on User interface Software and Technology, UIST '04, ACM Press, New York, NY, pp.197-206 (2004)
- [Mathworks] Mathworks, Matlab, <http://www.mathworks.com>
- [Moore 94] B. C. J. Moore, 聴覚心理学概論, 誠信書房 (1994)
- [Polli 04] A. Polli, DATAREADER: a tool for art and science collaborations. In Proceedings of the 12th Annual ACM international Conference on Multimedia MULTIMEDIA '04, ACM Press, New York, NY, pp.520-523 (2004)
- [Roads 01] C. Roads, コンピュータ音楽, 東京電機大学出版局 (2001)
- [Walker 03] B. Walker and J. Cothran, Sonification Sandbox: a graphical toolkit for auditory graphs, In Proceedings of the Ninth International Conference on Auditory Display ICAD2003, pp.161-163 (2003)
- [小橋 69] 小橋豊, 音と音波, 裳華房 (1969)
- [城 05] 城一裕, 山田和明, 中小路久美代, 問題の可聴化表現における音の属性と解きやすさの関係, 情報処理学会ヒューマンインタフェース研究会, IPSJ-SIGHI-114, pp.83-90, November (2005)
- [城 04] 城一裕, 山本恭裕, 中小路久美代, 音を伴う身体パフォーマンスのための聴覚インタラクションのモデルの提案, 情報処理学会ヒューマンインタフェース研究会, IPSJ-SIG-HI, Vol.2004, No.74, pp.27-32, July (2004)
- [高梨 03] 高梨 豪也, 三宅 美博, 共創出型介助ロボット "Walk-Mate"の歩行障害への適用, 計測自動制御学会論文 文集, 39(1), pp.74-81 (2003)
- [中小路 04] 中小路久美代, 山本恭裕, 創造的情報創出のためのナレッジインタラクションデザイン, 人工知能学会論文 誌, Vol.19, No.2, pp.154-165 March (2004)