音 楽 情 報 科 学 31−12
（1999. 8. 8）

自動表情付システム MorPH の実装

中口 孝雄，樋口 直史，真栄城 哲也
ATR 人間情報通信研究所 第6研究室

要旨

自動表情付システム MorPH は，インターネットで公開されているシステムである．表情付けは，システム内に記述されたルールを用いて実行される．入力された SMF データの各トラックは，ブロックに分割され，さらにフレーズに分割される．フレーズの検出は，前後の音との長さの比を順次計算によって行う．また，ギターに関しては，弦へのマッピングを行い，コードやアルペジオの演奏のより忠実な表情付を実現している．ドラムとサクソフォンについては，人間の実演奏データを用いている．初期段階でのユーザーの反応は，特にドラムに関して評価が高い．

MorPH — Musical Performance Humanizer

Tetsuya Maeshiro, Naofumi Higuchi and Takao Nakaguchi
Dept.6, ATR Human Information Processing Res. Labs.

Abstract

MorPH : Musical Performance Humanizer is a system open for internet public. The humanization processes are described in rule forms stored in the system. The tracks of input SMF data are divided into blocks, which are further divided into phrases. The phrase detection is performed by successive calculation of relative duration between previous and next notes. Notes are mapped into six strings of guitar, for accurate humanization of chords and arpeggio parts. Human played data are used for drums and saxophone. Initial responce from the users is affirmative, particularly the drums.

1 はじめに

楽曲の自動表情付けシステムとして，多々発表されている [1, 4, 5, 6, 7, 2, 8]．本報は，我々の自動作曲/編曲システムの研究の一環である自動表情付けシステム MorPH (Musical Performance Humanizer) の表情付けのアルゴリズムについて述べる．MorPH の対象は，ドラムパートを含む楽曲，特にポップス系の楽曲とし，楽譜通りに記述された SMF (Standard MIDI File) データに，音の強弱，発音タイミング，ゴースト音の付加などの処理を行い，表情付された SMF データを出力する．現在対応している楽器は，ドラム，ベース，ギター，そしてサクソフォンである．

システム構成は，図1のようになっている．

第2節では，簡単にシステム構成について述べた後，第3節で表情付の手法について述べる．第4節は考察および結論である．

2 システム構成

MorPH は，サーバー，データベース，そしてクライアントで構成される (図1)．

サーバーは，表情付けの処理と，クライアントとの通信を行う．

クライアントからのデータは，データベースで管理される．収集するデータは，クライアントのホスト名/IP アドレス，アクセスの日付，入力 SMF データ，表情付けのパラメータ等である．

MorPH には，初心者用の “MorPH light 1mg” および細かい設定が可能な “MorPH Rhapsody” の2種類のクライアントが用意されており，インターネットで公開している (http://morph.hip.atr.co.jp)．

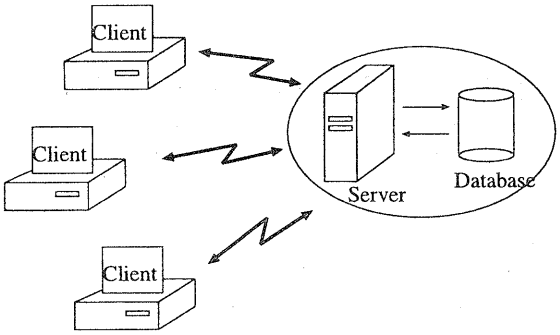

図 1: MorPH システムの構成図

3 表情付

表情付けの処理は，図 2 のように行われる．

3.1 入力 SMF データ

MorPH は，ジャストな SMF データのみを入力データとして受け付ける．これは，入力がフラットなデータでなければ，MIDI データから実際の楽曲が推測できないからである．ここで，ジャストな SMF データとは，音の長さおよび発音時間が，MIDI ファイルの分解能の分数比もしくは整数比であるデータを指す．例えば，分解能が 480 の場合，4 分音符の長さは 480，16 分音符は 120，そして，小節の始めの音符の発音時間 t は，$t\%1920 = 0$ を満たさなければならない．なお，% は，余りを意味する．

3.2 曲の構造抽出

入力 SMF データは，既に楽器毎のトラックに分かれているが，各トラックは，ブロックの集合に，そしてブロックはフレーズの集合に分割される．トラックは，1 小節以上休拍が続くとブロックに分割される．ブロックはフレーズに分割されるが，その手法は以下の通りである．

1 音目から順次見ていき，前後の音の長さとの比を計算する．その際，音の発音の長さと，次の音の発音時間までの長さの両者に対してそれぞれ計算する．フレーズの区切りの判定には，次の発音時間までの長さの絶対値と相対値，そして音の発音の長さの相対値を用いる．ここで，発音の長さを d, 次の音の発音時間までの長さを p とし，d_i のように添字 i は，音 i を表すとする．判定に用いる値は，

$$d_{i+1}/d_i \tag{1}$$

$$p_{i+1}/p_i \tag{2}$$

$$p_i/d_i \tag{3}$$

である．これらの値の組合せが基準値を越えれば，フレーズの区切りとする．

また，メロディーやソロの部分には，ベロシティの設定が必要であるが，式 1 から式 3 の値に加えて，音程の変化，音の長さ，そしてフレーズの始め，中間，終りの位置関係を用いる．

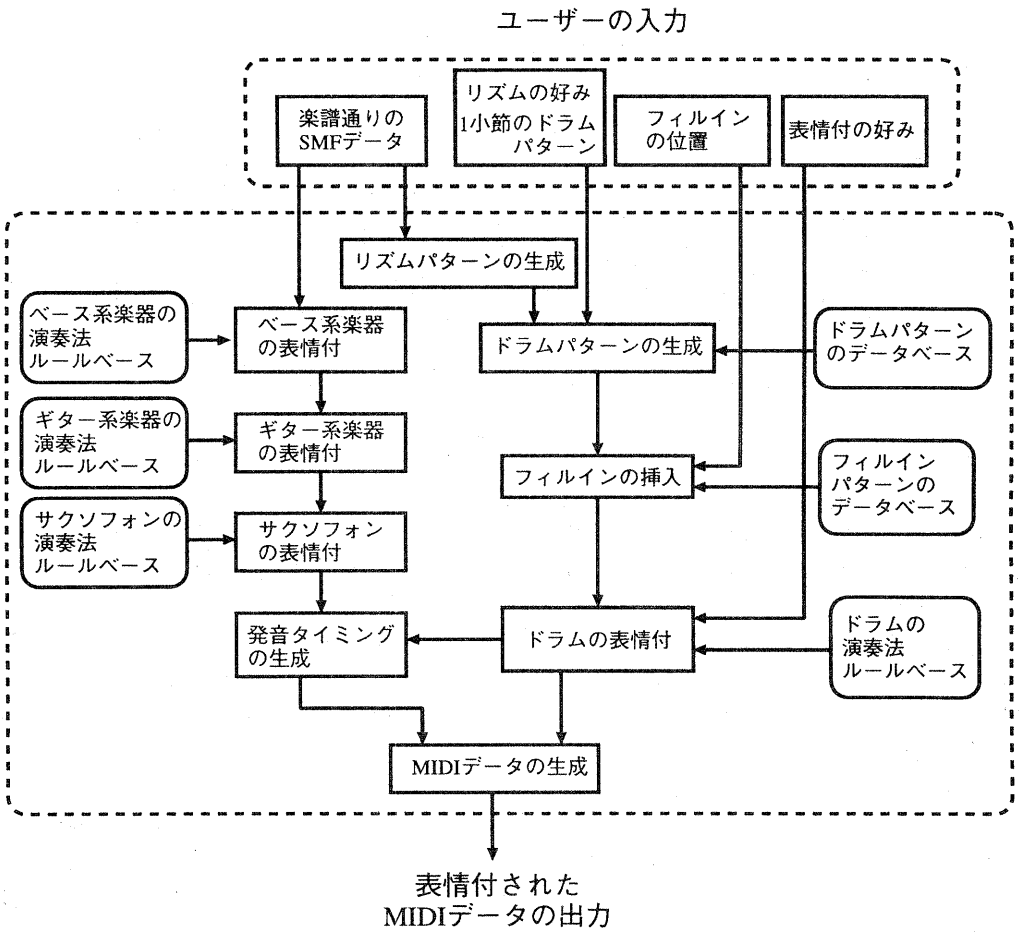

図 2: MorPH の処理のブロック図

3.3 サクソフォン

サクソフォンの処理は，ノート・ベロシティおよびエクスプレッションを行う．エクスプレッションの値は，実際に人間がウインドコントローラを演奏したデータを用いている．処理はフレーズ単位で行い，フレーズの始め，中間，終り，そして短い音の連続，次の音の音程などから，エクスプレッションのカーブを生成する．図 3 は，生成結果である．

3.4 ギター

ギターの処理は，(1) ソロ，(2) コード，そして (3) アルペジオの 3 種類に分けられる．ここでは，コードおよびアルペジオの処理に使われる，コードもしくはアルペジオの構成音の弦へのマッピングについて述べる．

コード，アルペジオともバッキングであり，どちらも実際の演奏時には複数の音が同時に発音している．コードの場合は，ストロークによって発音時間に微妙なずれが発生するため，どの音

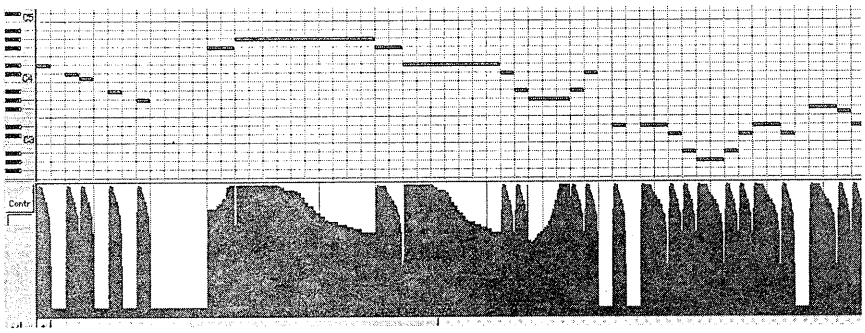

図 3: サクソフォンの処理によるエクスプレッション・カーブ

がギターのどの弦で弾かれるかを計算する必要がある．そうすることで，途中の弦がミュートによって発音しない場合でも，より現実に近い発音時間を計算することができる．一方，アルペジオの場合，入力される SMF データ (楽譜) には単音ずつで記述されていても，実際には弦を止めないため，音が重なって聞こえる．どの音がいつまで鳴るかを計算するには，弦へのマッピングが必要である．

コードの場合，構成音は発音時間が同じ音であるため，弦へのマッピングは比較的楽であるが，アルペジオの場合には，時間的に離散しているため，構成音を集める処理が必要となる．その処理には，2 種類の判定方法を用いている．1 つは，小節の変わり目で区切るものであり，もう 1 つは，次に述べる弦へのマッピングができなくなった時に区切るものである．

弦へのマッピングは動的計画法 [3] を用いて行う．評価関数は，指の移動量に相当する各音のタブの位置の範囲である．音を 6 弦の可能な全ての位置に順次マッピングしていき，範囲が一番狭いマッピングを選択する．この手法を用いて，実際に使われているコードと弦の対応表でテストしたところ，全ての弦のマッピングを推測できた．このことと，アルペジオはコードを分解したものであることから，本手法は有効であると判断した．また，コードの場合には，処理速度の向上を目的として，コードの弦へのマッピング用に，2125 種類のコードと弦のマッピングデータをシステム内に保持しており，対応する項目が無かった場合に，動的計画法を用いる．

一方，アルペジオとソロの判別もギターの処理には必要である．MorPH では，ブロックに分割する際，そのブロックがソロかバッキングかを判別する．ブロックに含まれる音が発音の長さで計算して 99%以上単音であるならばソロとし，それ以外をバッキングとする．ソロと判断したブロックがアルペジオであるかの判断は，ブロックの長さが 2 小節より長く，かつ複数のフレーズに分割できなければ，アルペジオと判断する．テストしたデータに関しては，この判別方法は有効である．

3.5 ベース

ベースの表情付けは，ハンマリング，ダウンスライド，ビブラート，そしてゴースト音の付加である．これらは，ルールとして記述してある．ダウンスライドとビブラートは，フレーズの終りの長い音で，次の音との音程差に大きければ，ダウンスライドを用いる．また，ゴースト音の付加は，基本的には拍毎だが，前後のリズムパターンを見ている．ハンマリングに関しては，同様にリズムパターンを見ているが，基本的には 16 分音符以下の連続した音に対して処理する．

3.6 ドラム

MorPH の 1 つの特徴は，ドラムパートの自動生成である．MorPH は，一拍分のスネアドラムとキックドラムの全ての組合せを構造化して内部に持っており，類似パターン，そしてリズムが細かい/粗いパターンに関連付けられている．また，類似パターンは，スネアまたはキックドラムが鳴っている位置は同じだが，鳴っているドラムの種類が異なる場合を指す．パターン A がパターン B より細かいとは，パターン B にスネアもしくはキックドラムが 1 つ加えられている場合であり，より粗いとは，その逆である．さらに，フィルインのパターンも同様であり，類似/細かい/粗いパターンを関連付けている．フィルインの場合には，タムやシンバル等，使える楽器が増えるが，全ての組合せは持っておらず，使えないパターンは主観的に判断して除いてある．こうすることで，部分毎に推測された楽曲のドラムパターンに基づいて，より細かい，粗い，もしくは類似のドラムパターンの生成が可能である．また，ゴースト音として，16 部音符および 32 部音符のパターンが追加される．ゴースト音の追加も，段階的に設定できる．

また，出力 SMF データに加えられるドラムパターンの生成は，(1) ユーザーのドラムパターンが与えられている場合と，(2) 与えられていない場合の 2 通りがある．いずれの場合も，リズムパターンを生成してから，スネアやキックなどの割当を行う．(1) の場合，ユーザの指定する 1 小節分のドラムパターンと，楽曲から推測されるドラムパターンから，両者の共通項を取る．内部に保持している類似/細かい/粗い関連パターンを用いて，ユーザーのドラムパターンの特徴を保持しながら，曲にあったパターンを生成する．また，そのような変更をしないようにもできる．この場合，ユーザーのドラムパターンが単純に追加され，そこにハイハットのアクセントや発音タイミング等が生成される．一方，(2) の場合は単純に，ベースパートから推測したドラムパターンを基に，ドラムパートを生成する．

4 考察および結論

フレーズの認識は，単にメロディーの区切りだけではなく，ギターのソロとアルペジオの判別に使えるため，有効であると考えられる．判別実験の対象としたアルペジオには，同じ長さの音が連続するパターンだけではなく，リズム的な要素も含むパターンも含まれており，メロディーと他の演奏パターンとの判別にも使えると思われる．今後，インターネット経由で送られてくるユーザーの SMF データを対象に，さらに実験を重ねていく．一方，ギターの音符の弦へのマッピングは，他の弦楽器にそのまま応用できるものであり，現在，ベースを対象に実装中である．ドラムに関しては，基本的な表情付は終了したと考えており，今後，MorPH のルール編集機構の実装とともに，特定の演奏家の個性，特にドラムのソロの実現を予定している．その際，現在システムに組み込まれているドラムパターンの類似性による関連関係と，ソロのパターンがそれらにどう位置付けられるかを検討する予定である．インターネットで公開した目的は，広範囲なユーザーの評価と実験データの収集であるが，特に後者が表情付けの実装に関して，有効であると考えられる．

参考文献

[1] 青野 裕司，片寄 晴弘，井口 征士 (1997) “重回帰分析を用いた演奏表現法の抽出”，情報処理学会 **38**:1473-1481.

[2] T. Hoshishiba, S. Horiguchi and I. Fujinaga (1996) "Study of expression and individuality in music performance using normative data derived from MIDI recordings of piano music", *Proc. ICMPC*, 465-470.

[3] S. E. Dreyfus and A. M. Law (1977) *The Art and Theory of Dynamic Programming*, Academic Press, New York.

[4] A. Friberg and J. Sundberg (1986) "A Lisp environment for creating and applying rules for musical performance", *Proc. ICMC*, 1-3.

[5] 福岡 俊之, 片寄 晴弘, 井口 征士 (1990) "音楽解釈システム MIS における自動演奏生成過程について", 情報処理学会 全国大会, 1585-1586.

[6] 片寄 晴弘, 福岡 俊之, 井口 征士 (1990) "音楽解釈システム MIS における演奏ルールの抽出について", 情報処理学会 全国大会, 1587-1588.

[7] 平賀 瑠美 (1998) "演奏の表情付け", コンピュータと音楽の世界, 共立出版.

[8] G. Widmer (1993) "Understanding and learning musical expression", *Proc. ICMC*, 102-108.

[9] G. Widmer (1994) "Learning expression at multiple structural levels", *Proc. ICMC*, 102-108.