Vol.2010-MUS-86 No.23
2010/7/30

音楽理論 GTTM に基づく 木構造を用いたメロディ生成手法

西田 智[†1]　浜中 雅俊[†1]
平田 圭二[†2]　東条 敏[†3]

本稿では，曲の雰囲気や特徴を変えずに，楽曲構造に基づきメロディ中に新たな音を追加して，メロディを生成する手法について述べる．本手法の特長は楽曲構造を利用したことであり，その結果，生成されたメロディは元のメロディの雰囲気を忠実に反映することができた．提案手法では，まず，メロディを音楽理論 GTTM に基づき分析し，タイムスパン木を獲得する．次に，得られたタイムスパン木に，既存の曲のタイムスパン木を参考にして，新たな枝を追加することで，音数の増加した新たなメロディを生成する．実験の結果，簡約したメロディから楽曲構造に沿って具体的なメロディを生成できることを確認した．

Melody Generation method using tree structure of GTTM

Satoru Nishida,[†1]　Masatoshi Hamanaka, [†1]
Keiji Hirata, [†2]　and Satoshi Tojo[†3]

This paper presents a melody generation method which adds new notes to a given melody with its flavor and characteristics retained. The main advantage of our method is that musical structures are employed to keep the similar flavor of an original melody in a generated melody. First, we analyze an original melody based on music theory GTTM and acquire a corresponding time-span tree. Next, we add a new branch and a note, referring to the time-span tree of an original melody. As a result, we obtain a new melody in which a new note has been added. We conduct an experiment, which demonstartes that our method can generate an instantiated melody from a simple melody, unchanging its basic musical structure.

1. はじめに

本研究では，元々の曲の雰囲気や特徴を大きく変えずに，メロディ中に新たな音を追加して，新たなメロディを生成できるシステムの構築を目指す．曲の雰囲気や特徴といった要素は複数の音により成り立ち，音楽初心者は雰囲気や特徴を具体的に捉えることは難しい．また，曲の雰囲気や特徴を捉えることができても，雰囲気などを崩さないような音を決めるのは，作曲経験のないユーザにとって困難である．そこで，メロディ生成システムがユーザの入力したメロディに音を追加することで，作曲経験のないユーザでもメロディを容易に完成させることが可能となる．

本稿では，ユーザが与えたメロディを分析し，その楽曲構造木に，既存の曲の楽曲構造木を参考に新たな枝を追加して，音数を増加させていくメロディ生成手法を提案する．本研究で用いる楽曲構造木はメロディの音を重要な音と装飾的な音を区別したもので，入力されたメロディに装飾的な音を追加することで，メロディの雰囲気や特徴を残しつつ，新たなメロディの生成ができる．これによって，ユーザは音数が少ない未完成なメロディから，より音数の多い具体的なメロディを完成させることができる．また，ギターソロのような細かく変奏されるメロディを，元のメロディの特長やニュアンスなどを崩すことなく生成することも可能となる．

これまで，ユーザの入力に基づいてメロディを生成する手法は様々な研究がなされており，対位法や機能和声法などの音楽理論を元にメロディを生成する手法や，既存の曲を学習して構築した確率モデルに基づきメロディを生成する手法が提案されてきた．しかし，これらの生成手法は，音を音高や音価などの表層的な要素のみで扱いメロディを生成するため，ユーザの入力したメロディと楽曲構造が変わってしまう問題があった．

これまでのメロディ生成の代表的な研究として，Pachet の Continuator が挙げられる [1]．Pachet は，メロディを木構造のマルコフ連鎖によりモデル化し，音高の遷移確率を学習することで，ユーザが入力したメロディに続く次の音高を決定する手法を提案した．入力されたメロディと合致するメロディが学習データにない場合に，入力されたメロディの最初の音を削りメロディの検索範囲を緩和しながら検索することで，スパースネスの問題を解決している．しかし，音高を遷移確率に基づき決定するため，楽曲構造の異なるメロディが生成されることがあった．また，音価はユーザの入力や学習データに基づき決定するため，本来強拍であるべき拍点に発音がなく，強拍とな

[†1] 筑波大学大学院システム情報工学研究科
University of Tsukuba, Graduate School of Systems and Information Engineering
[†2] NTT コミュニケーション科学基礎研究所
NTT Communication Science Laboratories
[†3] 北陸先端科学技術大学院大学
Japan Advance Institute of Science and Technology

らずに生成されるメロディのリズムが崩れてしまうという問題もあった．

また，もう一つの代表的な研究として，David Cope の Experiments in Musical Intelligence (EMI) が挙げられる[2]．EMI は，作曲家に特有の”signature”という構造を分析し，複数のメロディを収めたデータベースから適切なメロディの断片を検索し，ある音楽文法に従って接続することでメロディを生成する手法を提案した．EMI は，メロディを分析することで，特有の作曲家らしさを反映したメロディの生成を可能にした．しかし，接続に用いる音楽文法は音価や音高などの表層的な要素により決定されるものであるため，楽曲構造が崩れたメロディが多く生成される問題があった．また，EMI に用いられた音楽文法は EMI 専用に構築されたものであり，他の生成手法のための再構築や拡張が容易に行うことできないという問題もある．

これらより，メロディの分析結果を利用したメロディ生成は有効であると考えた．これまで，我々は Fred Lerdahl と Ray Jackendoff により提唱された音楽理論 GTTM (Generative Theory of Tonal Music) [3]に基づく楽曲構造を用いてメロディを生成する手法を研究してきた．GTTM の分析により得られる楽曲構造は，音楽に関して専門知識のある聴取者の直感を形式的に記述した階層的な楽曲構造である．GTTM の楽曲構造は，図 1 に示される 3 つの楽曲構造があり，メロディのまとまりを表すグルーピング構造，メロディのリズムを表す拍節構造，そして，音の構造的な重要さを表すタイムスパン木である．これらの楽曲構造は隣り合う音との音高差などにより分析されるため，メロディ全体の構造とメロディ内の音を同時に扱うことができる．

音楽理論 GTTM に基づくメロディ内の音の分割手法として，浜中らは，GTTM の拍節構造で最も拍点の強さが高くなる拍点で音を分割する手法を提案した[4]．また，分割後の音高について，分割後の 2 音のうち，発音点の拍が弱い方の音の音高を，和音の進行などを記述した TPS (Tonal Pitch Space)[5]により算出される安定度が最も高い音高に変更する手法も提案した．この手法は，GTTM の拍節構造や TPS を考慮して細分化するため，生成されるメロディが．しかし，分割後の楽曲構造を考慮していないため, 入力メロディと分割後のメロディの楽曲構造が変わってしまう可能性があった．

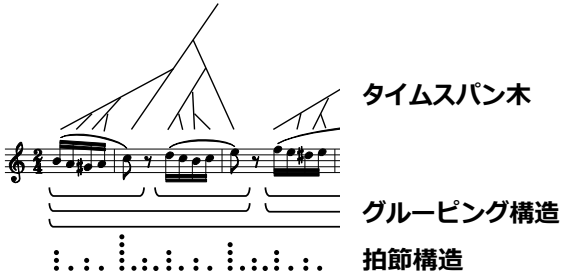

図 1　GTTM に基づく分析により得られる楽曲構造

この問題を解決するために，本研究では，入力されたメロディを音楽理論 GTTM により分析し，楽曲構造の一つであるタイムスパン木を用いる．入力されたメロディのタイムスパン木に新たな枝を追加することで，生成前のメロディのタイムスパン木の構造を変えることなく，装飾的な音を追加した新たなメロディを生成することができる．そして，追加する新たな枝の決定手法として，音楽家が分析した既存のメロディのタイムスパン木のデータから，現在のタイムスパン木に追加できる最適な枝を検索することで，外れた音などがない自然なメロディを生成することができる．

2. 音楽理論 GTTM のタイムスパン木の特長とメロディ生成

本研究では，楽曲構造を分析する理論として音楽理論 GTTM を用い，GTTM の楽曲構造のタイムスパン木を用いる．このタイムスパン木は，グルーピング構造と拍節構造の結果より，メロディを構造的に重要な音と装飾的な音とを区別し，重要な音がより幹となる木構造である．タイムスパン木は，グルーピング構造と拍節構造から得られるため，メロディとリズムも扱うことができる楽曲構造である．

また，このタイムスパン木には，楽曲を階層的なタイムスパンに分割し，各タイムスパンを楽曲構造的に重要な音に簡約化することができる性質がある．図 2 にはメロディとそのメロディのタイムスパン木が示されている．このメロディをタイムスパン木の簡約レベル A より下にある枝の音を省略すると簡約されたメロディ A が得られる．さらに簡約レベル B より下にある枝の音を省略すると，メロディ A よりも簡約されたメロディ B が得られる．このように，タイムスパン木の枝を取り除き幹を残すことで，メロディから装飾的な音が取り除かれ重要な音のみを残したメロディへと簡約することができる．このタイムスパン木の「枝の音を省略することで重要な音のメロディに簡約される」という特徴を逆に利用し，現在のタイムスパン木に新たに枝を追加していくことで，音数を増加させながら，重要な音から装飾的な音へと段階的に具体的なメロディを生成することが可能となる．

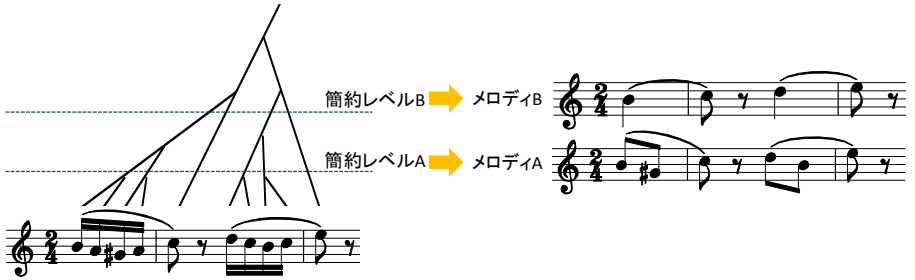

図 2　メロディの簡約例

また，メロディの簡約は，新たに枝を追加するときの既存のメロディとの検索においても役立つ．本研究では，音を追加していくことでメロディを生成するため，入力メロディの長さは既存の曲と同じであっても，音数が既存の曲の音数よりも少ない可能性は十分にある．このような場合，入力メロディと既存曲のメロディはその簡約レベルが異なっているが，タイムスパン木を用いない従来の検索では，レベルを無視して音数を一致させて比較したり，別途に音数を減らしたメロディを用意する必要がある．しかし，タイムスパン木を用いることで，簡約レベルを考慮した検索ができ，適宜簡約することで別途にメロディを用意する必要がない．

本研究では，クラシック曲から切り出した 8 小節の長さの 100 個のメロディを分析したタイムスパン木のデータを用いて，入力されたメロディのタイムスパン木に追加する最適な音の検索する．入力メロディのタイムスパン木に一致するタイムスパン木を検索する場合，その木構造の形だけでなく，各枝に割り当てられた音の音高，音価などの要素が一致しなければならない．しかし，今回用いる 100 個のメロディデータから，入力メロディのタイムスパン木にマッチするタイムスパン木を検索する場合，次の問題点が挙げられる．

課題 1：入力メロディのタイムスパン木のスパースネス問題

本研究では，枝の追加のために用いているタイムスパン木のデータは少なく，ユーザが入力したメロディのタイムスパン木に完全にマッチするタイムスパン木が，システムのデータから得られない可能性がある．従来，このようなスパースネス問題を解決する方法として，Pachet の Continuator では検索に用いるメロディの音数を減らすなど，検索範囲を緩和する方法が用いられてきた．

解決法 1 ：部分木を用いた検索

本研究では，このスパースネス問題を解決するために，検索範囲の緩和するため，入力メロディのタイムスパン木の部分木を用いて再検索する．これにより，入力メロディのタイムスパン木と一致する既存の曲のタイムスパン木をより確実に得ることが可能となる．本研究では，検索に用いる最小の部分木を，1 本の幹と 1 本の枝で構成される木構造を最小の部分木とする．

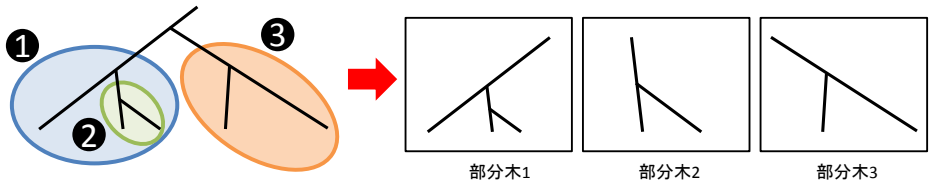

図 3　タイムスパン木とその部分木

課題 2：完全一致するタイムスパン木のスパースネス問題

課題 1 の解決策により，部分木を含めたタイムスパン木を検索する．しかし，部分木を含めても完全にマッチするタイムスパン木が 1 つも検索されなかった場合も考えられる．過去のメロディ生成の研究では，音高のみの一致に基づいて音を生成する方法や，入力された音高列の代わりに音高差を用いる方法があった．

解決法 2：音価のずれを考慮した検索

本研究ではこのような場合，音価のずれを考慮して，木構造・音高・出現順序が一致するタイムスパン木の検索を行うこととした．タイムスパン木の構成要素の内，木構造の形と音高とその出現順序は，生成される音の音高に影響を与え，不自然なメロディになってしまう可能性がある．一方で，音価のずれはリズムに影響を与えるが，リズムの多少のずれについては，人間は許容できると我々は考えた．これらにより，音価のみが異なっているが，木構造の形と音高および音高の出現順序が一致しているタイムスパン木を検索し，マッチする木を得ることとした．

課題 3：マッチした枝が複数存在する問題

現在のメロディのタイムスパン木にマッチするタイムスパン木を検索した場合，そのタイムスパン木にマッチするタイムスパン木がデータから複数個検索される可能性がある．また，課題 1 の解決法により，メロディのタイムスパン木とその部分木を用いてマッチするタイムスパン木を検索するため，1 つのメロディのタイムスパン木から，複数個のタイムスパン木がマッチし，新たな枝の追加方法が複数個考えられる場合がある．

解決法 3：検索されたタイムスパン木の順序付け

本研究で用いるデータは少ないため，乱数により生成する音を決定すると，ある曲独自の音が決定される可能性がある．そこで，本研究では新たに追加する枝が一意的に定まる方法をとる．枝を一意的に定めるために，マッチした音のタイムスパンの長さ，マッチしていない音の音価の差の総量，タイムスパン木のタイムスパンの長さ，マッチした枝の出現頻度の 4 つを評価基準として順位付けを行うこととし，最もよい枝をメロディに新たに追加する枝とした．

課題 4： 同率 1 位となる複数の枝

課題 3 の解決策により，検索されたタイムスパン木の複数の枝をタイムスパンの長さなどに基づいて新たに追加する最適な枝を決定する．しかし，既存曲のタイムスパン木のデータ数が少ないため，同率 1 位の枝が複数算出される場合がある．たとえば，図 4 の検索対象のメロディと完全一致する木が 2 本検索され，そのどちらの出現頻度も最も高い値であった場合，この 2 つの枝を順位付けした場合，同率 1 位となり，追加する枝を 1 本に決めることができなくなってしまう．

Vol.2010-MUS-86 No.23
2010/7/30

解決法 4：1 つ先の枝の追加を予想した検索

本研究ではこのように同率 1 位となる候補の枝が複数存在する場合，候補となっている枝をメロディに追加したと仮定し，さらにもう 1 回枝を追加する場合を予測した先読み検索を行う．そして，それぞれの枝で先読み検索した結果さらに追加される枝同士を比較・順位付けを繰り返し，追加する枝を 1 つに決定する（図 4）．

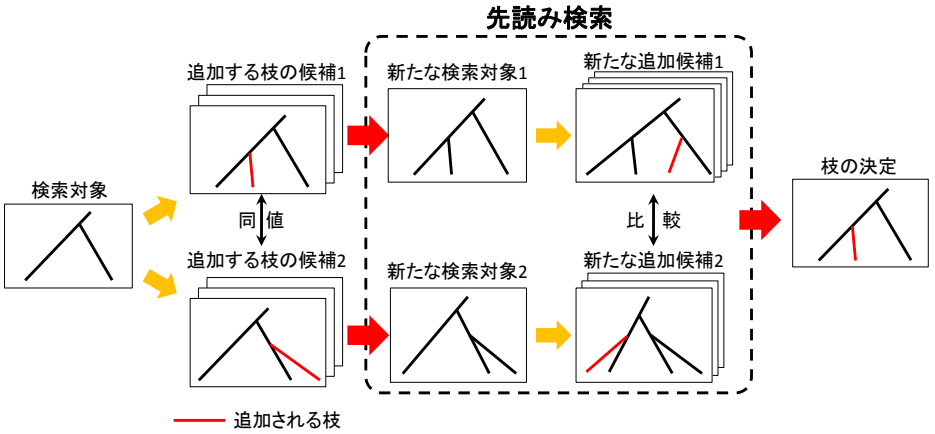

図 4　先読み検索

3.　タイムスパン木を用いたメロディ生成手法

我々が提案するメロディ生成手法の全体図を図 5 に示す．まず，ユーザはメロディを入力としてシステムに与える．システムは，入力されたメロディを分析し，メロディのタイムスパン木を獲得する．さらに，そのタイムスパン木の部分木を獲得する．

次に，入力されたメロディのタイムスパン木とその部分木に完全一致するタイムスパン木を，音楽家が分析した既存曲のタイムスパン木のデータから検索する．このとき，部分木を含め木構造・音高・音高の出現順序・音価の全てが完全に一致するタイムスパン木が 1 つも見つからなかった場合は，入力されたタイムスパン木とその部分木の木構造・音高・音高の出現順序が一致するタイムスパン木の検索するそして，検索された枝の出現頻度や検索クエリのタイムスパン木全体のタイムスパンの長さなどを用いて，一致した各枝のスコアを算出する．そして，最大のスコアを持つ枝を新たに追加する枝とする．このとき，最大のスコアを持つ枝が複数存在する場合は，先読み検索を行い追加する枝を 1 つに絞り込む．

最後に，決定された枝を現在のタイムスパン木に追加することで，元のメロディに

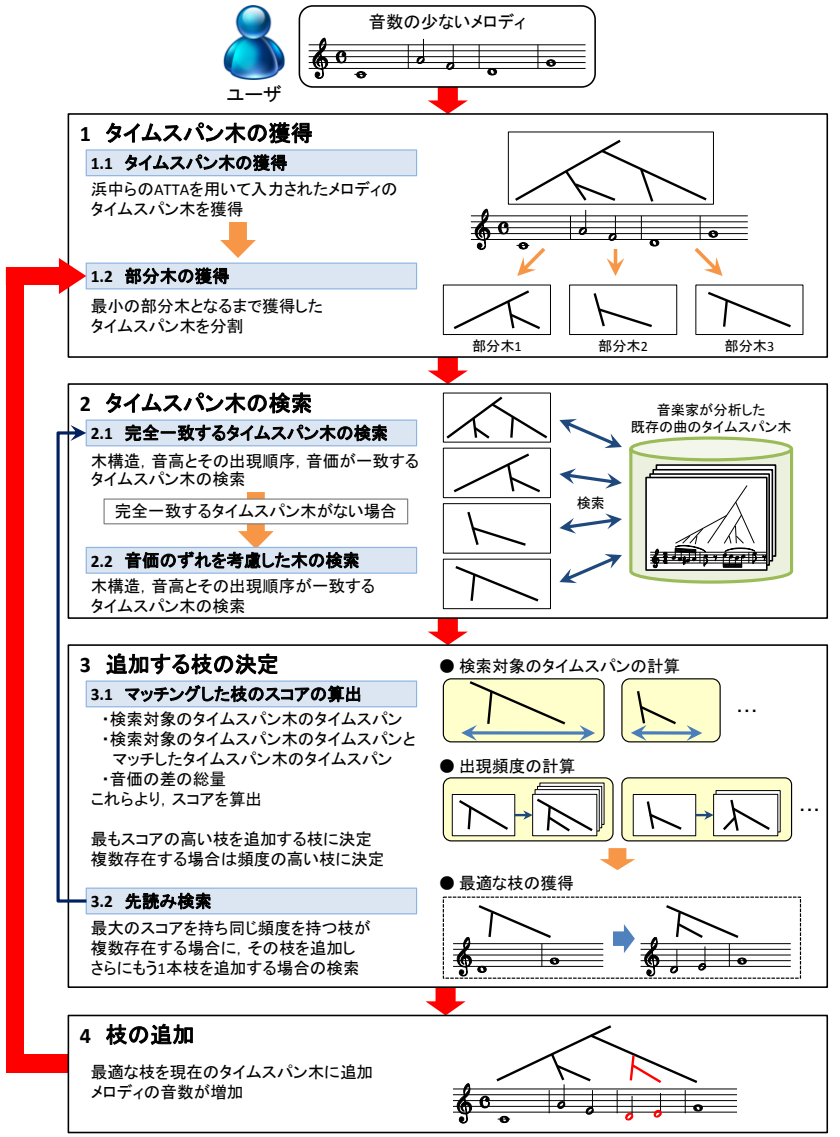

図 5　生成手法の全体図

Vol.2010-MUS-86 No.23
2010/7/30

新たに 1 音追加されたメロディが生成される．この流れを繰り返すことで，音数の少ないメロディを 1 音ずつ増加させ，より音数の多いメロディを生成する．

システムの入出力には MusicXML を用いる．MusicXML は XML 形式の楽譜表記フォーマットで，多くの楽曲編集ソフトウェアで使用できる．本研究で用いるタイムスパン木のデータは Time-spanXML という XML 形式で記述されており，MusicXML と要素が互いに結合している．また，MusicXML には調号などの楽譜情報を記述している要素があるため，入出力に MusicXML を用いることは提案する手法に有効である．

3.1 タイムスパン木の獲得

まず，入力されたメロディを浜中らの提案した ATTA により分析し，入力メロディのタイムスパン木の獲得する．ATTA は音楽理論 GTTM をコンピュータ用に改良した exGTTM を実装し，タイムスパン木を獲得できるシステムである．ATTA については，文献[6]に詳細が述べられているため，ここでは省略する．

ATTA により得られたタイムスパン木から部分木を獲得する．本研究のメロディ生成は，手法である．そのため，部分簡約により得られる部分木を考慮しない．よって，幹に最も近い接点から分割を繰り返し，2 音の木構造がなくなるまでタイムスパン木の分割を繰り返す．

3.2 タイムスパン木の検索

まず，メロディのタイムスパン木とその部分木のそれぞれにマッチするタイムスパン木を音楽家が分析した既存の曲のタイムスパン木のデータから検索する．2 章で述べたように，タイムスパン木の検索では，木構造の形，音高，出現順序，音価のマッチングを行う．ただし，調やオクターブが異なっていても，新たに生成される音の音程は変わらないと考えたため，既存のメロディの調やオクターブを入力メロディに合わせて音高を比較する．メロディの転調は MusicXML 内に記述されている調号の要素をもとに既存の曲を転調させる．

そして，検索によりマッチングした既存の曲のタイムスパン木から追加する枝の候補を決定する．新たに追加する枝の候補は，入力メロディのタイムスパン木にマッチしていない枝の内，最も簡約レベルの高い枝を候補とする（図 6）．このとき，最も高い簡約レベルでマッチしていない枝が複数本存在する場合，全ての枝を候補として記憶する（図 7）．

入力メロディのタイムスパン木およびその部分木と完全に一致するタイムスパン木が 1 個も見つからなかった場合，木構造の形，音高，出現順序が一致するタイムスパン木の検索を行う．この検索では，音価のマッチング以外は，完全一致する木の検索と同様に，既存の曲のメロディの調とオクターブを入力メロディに合わせて検索を行い追加する枝の候補を決定する．

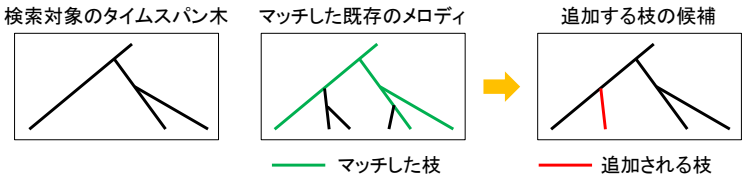

図 6　枝の追加

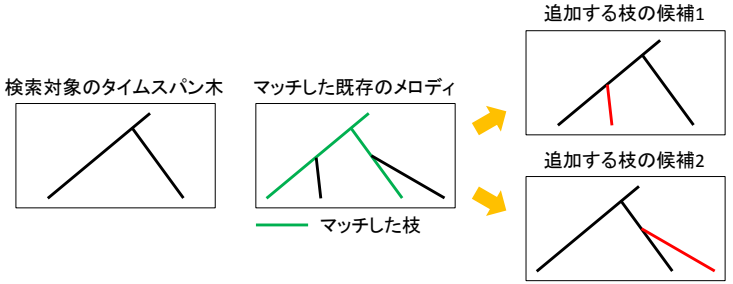

図 7　複数の枝の追加

3.3 枝の決定

検索により複数の枝が候補として挙げられる．これらの候補となった枝から最もよい枝を 1 つ見つけるため，に基づきスコア付けを行い，最良な枝を決定する．

まず，検索に用いた木のタイムスパンの長さを算出する．部分木を検索に用いた場合，用いられた部分木の高さがメロディのタイムスパン木の高さに近いほど，その部分木はメロディをより表現している．そこで，部分木とメロディのタイムスパン木の高さの近さを，部分木のタイムスパンの長さによって評価する．部分木は分割を重ねるごとにタイムスパンの長さは短くなる．よって，検索に用いられた部分木のタイムスパン ts_{sub} とメロディのタイムスパン木のタイムスパン ts_{melody} から，部分木の高さの近さ p_b を次式により求める：

$$p_b = ts_{sub} / ts_{melody} \qquad (1)$$

また，音価の差を考慮して検索した場合は，マッチした木と検索に用いた木の音価やタイムスパンの長さが異なっている．そこでまずは，検索に用いられた木のタイムスパンとマッチしたタイムスパン木のタイムスパンがどの程度異なっているかを求める．新たに追加する音の音価は木の中の他の音から相対的に決定されると考え，絶対的なタイムスパンの差ではなく相対的な差を用いることとした．よって，検索に用い

られた木のタイムスパンタイムスパン ts_{sub} とマッチしたタイムスパン木のタイムスパン ts_{match} の比 r_b をスコアとして用いる：

$$r_b = \begin{cases} ts_{sub}/ts_{match} & ts_{match} \geq ts_{sub} \\ ts_{match}/ts_{sub} & ts_{sub} < ts_{match} \end{cases} \quad (2)$$

さらに，音価の差の総量を計算する．音価の差もタイムスパンの差と同様に相対的であると考え，まず，検索に用いたタイムスパン木にタイムスパンの長さを合わせる．その後，検索に用いた木の音価 ds とマッチした木の音価 dm の差を音符ごとに求め，木全体の差 g_b を算出する．木全体の差 g_b は，各音の音価の差が小さいほど大きい値を持つように，各音の差を総乗する式とした：

$$g_b = \prod_i \left(1 - \frac{|ds_i - dm_i|}{ds_i}\right) \quad (3)$$

この 3 つを用いて，新たに追加する枝のスコア S_b を次式で定義する：

$$S_b = p_b \cdot r_b \cdot g_b \quad (4)$$

このスコアが最も高い枝を新たに追加する枝とする．このとき，最も高いスコアを持つ枝が複数ある場合，それぞれの枝の出現した頻度を求め，最も高い頻度を持つ枝を追加する枝と決定する．

スコアが最も高く，出現頻度が同じ枝が複数本存在する場合，候補となっている枝それぞれにおいて，候補の枝をメロディに追加したと仮定し，さらにもう 1 回枝を追加する場合の処理を行う．枝をさらに追加した場合の処理の結果，得られるさらに追加する枝のスコアと頻度を候補の枝の新たなスコアと頻度として比較する．比較の手順は通常の枝の追加方法と同様に，まずスコアを比較し，最も大きいスコアを持つ枝を追加する枝とする．スコアの最大値を持つ枝が複数存在する場合は，頻度の比較を行い，最も高い頻度を持つ枝を追加する枝とする．この追加を想定したときの処理を，枝が 1 本に決定されるまで繰り返す．

3.4 枝の追加

獲得された最適な枝を現在のタイムスパン木に追加し，音数の増加したメロディをユーザに提示する．さらに枝の追加する場合は 3.1 の部分木の獲得に戻り，3.2 の検索と 3.3 のスコア算出を行い，音数を 1 音ずつ増加させるメロディ生成を繰り返す．

4. 実験

提案手法の有効性を示すため，提案手法に基づき音数の少ないメロディを入力し，音数が増加したメロディを生成する．まず，検索データにない既存の曲のメロディを，メロディ自身のタイムスパン木に従い簡約したメロディを入力とする．そして，入力されたメロディを提案手法により元のメロディと同じ音数になるまで分割を行う．この生成の過程をし，生成されるメロディが音楽的に正しいかを調べた．

4.1 実験方法

実験の入力メロディには，Beethoven 作曲の歓喜の歌の始めの 8 小節を用いた．図 8 に歓喜の歌のメロディとタイムスパン木を示す．これを，中の簡約レベルで簡約したメロディを入力として用いる（図 9）．入力メロディは 8 音であり，歓喜の歌は 28 音であるため，のメロディから 20 音追加する実験を行う．

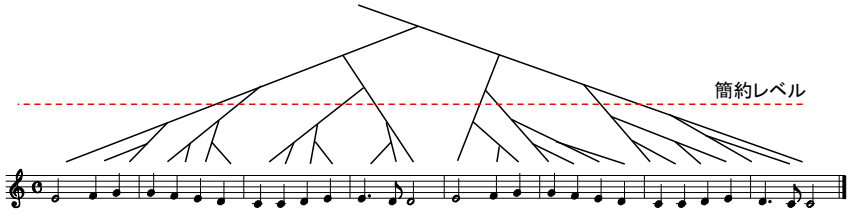

図 8　Beethoven の歓喜の歌とタイムスパン木

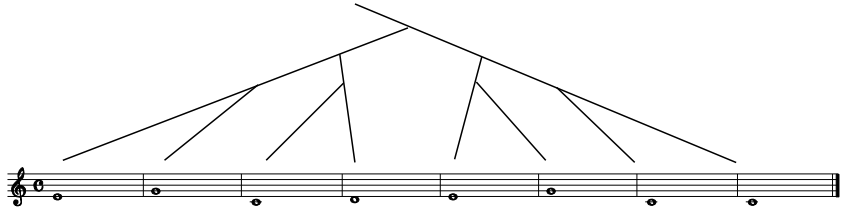

図 9　実験で入力に用いたメロディ

なお，提案手法では，タイムスパン木の獲得に ATTA を用いるが，現在の ATTA は正確なタイムスパン木を得ることが難しい．入力メロディのタイムスパン木が異なっている場合は，正しくメロディ生成が行われないと考えられる．そこで，今回の実験では，人間が分析したタイムスパン木を用いて，枝を追加するメロディ生成を行い，タイムスパン木に新たな枝を追加する手法の有効性を検証する．

4.2 結果

8 音に簡約されたメロディを元のメロディと同じ音数である 28 音のメロディに生成されていく過程を図 10 に示す.

序盤に生成された音は，音が強拍にあたる拍点に置かれ，音高は調や和声に沿った音となっている．すなわち，生成された音は，拍節や和声に沿ったメロディの中で重要な音であることを示している．また，音数が増えていくにしたがって，次第に裏拍にあたる拍点に音が割り当てられていき，音高は音と音の間を埋めるように定められていった. これらの生成された音は, メロディの間を繋げるような装飾的な音である.

これらにより，本手法では，タイムスパン木に沿って重要な音から装飾的な音へと生成が行われることが確認できる．また，今回生成された音は，全て調から外れておらず，Beethoven の歓喜の歌の雰囲気や特徴を大きく崩していないと言える.

5. おわりに

本稿では，入力されたメロディのタイムスパン木に，新たな枝を追加することで，入力メロディより音数の増加したメロディを生成する手法を提案した．また，提案手法では，新たに追加する枝を既存の曲のタイムスパン木の出現頻度などから決定することにより，新たに追加される音が一意的に定まる．そして，提案手法の有効性を検証するために，既存の曲をそのタイムスパン木に基づき簡約したメロディから，メロディの生成を行った．メロディ生成の結果から，タイムスパン木に沿って雰囲気や特徴を崩さないメロディの生成が可能であることを確認した.

今後は，この部分に音を追加したい，このような音を追加したいなどの，ユーザの細かい要望に応えることができるメロディ生成手法について検討する．また，新たに追加する枝をタイムスパン木のではなく幹に追加することで，ユーザの入力したメロディに続くメロディの生成や予測・補完する手法についても検討する予定である.

参考文献

[1] Pachet, F: Thecontinuator: Musical Interaction With style, In Proc. of ICMC 2002, pp. 211-218 (2002).
[2] David Cope.: Experiments in Musical Intelligence, A-R Editions, Inc. (1996).
[3] Lerdahl, F., and R. Jackendoff.: A Generative Theory of Tonal Music., MIT Press (1983).
[4] 浜中雅俊，平田圭二，東条敏: Melody Extrapolation in GTTM Approach, In Proc. of ICMC 2009, pp. 89-92 (2009).
[5] Lerdahl, F.: Tonal Pitch Space, Oxford University Press (2001).
[6] 浜中雅俊，平田圭二，東条敏:”ATTA:exGTTM に基づく自動タイムスパン木獲得システム”, 情報処理学会研究報告 2005-MUS-61-4, Vol. 2005, No. 82, pp. 19-26 (2005).

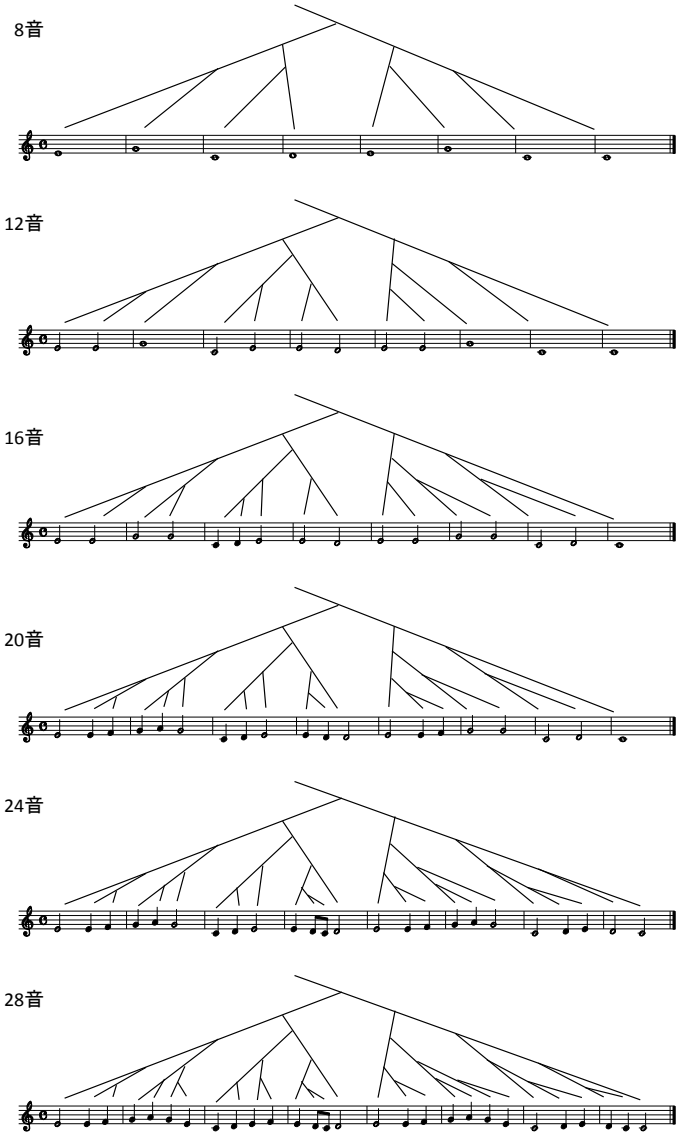

図 10 メロディ生成実験の過程