音 楽 情 報 科 学 44－6
（２００２．２．１６）

パターン認識手法を用いたオーディオ符号化に関する研究

松井　唯史　　甲藤　二郎

早稲田大学大学院理工学研究科

{matsui,katto}@katto.comm.waseda.ac.jp

あらまし　インターネットや携帯電話を利用した音楽配信サービスが盛んに行われており、より少ない情報量で音楽を伝送することが求められている．しかしながら、現在の MP3 に代表されるオーディオ符号化技術では、その音質を保つことが限界に達しつつある．一方、MPEG ビデオと比較して、MPEG オーディオでは、時間軸方向の相関を利用した冗長削減の割合が非常に小さいと考えられる．今回、特に DTM によって製作された音楽に関しては、同じパターンが何度も繰り返される可能性が高いことに注目し、そのような同一パターンを DP マッチングを用いて音楽ソースの中から探索し、重複するパターンの情報量を削減するアルゴリズムについて提案する．これまで提案してきた非可逆圧縮に加え、類似パターンの差分信号と DP パス情報を記録する可逆圧縮にも言及する．

キーワード　オーディオ符号化, パターンマッチング, DP, 音楽配信

A Study on Audio Compression Using Pattern Recognition Approaches

Tadashi Matsui　　Jiro Katto

Graduate School of Science and Engineering, Waseda University

{matsui,katto}@katto.comm.waseda.ac.jp

Abstract　Music distribution service on the Internet or the cellular phone networks is deployed quickly, and music transmission with the fewer amount of information is required. However, with the audio coding technology such as MP3, maintaining the tone quality is reaching a limit. On the other hand ,as compared with MPEG video, MPEG audio does not necessarily exploit temporal correlation for data compression in a sufficient manner. Especially, in case of the music produced by DTM ,it is observed that the same pattern is repeated more frequently. This paper, therefore, proposes usage of DP matching in order to detect the same patterns in sound sources, and to reduce total amount of sound data compared with classical compression schemes. In addition to lossy approaches, we also refer to a loss-less approach by recording differences of similar patterns and DP path information.

Key words　audio coding, pattern matching, dynamic programming, music distribution

1. はじめに

近年、インターネットを用いた音楽配信サービスの普及にはめざましいものがある．しかし、MP3 に代表される既存のオーディオ符号化方式では、ダイヤルアップなどの狭帯域な接続環境では、ダウンロードしながらリアルタイムに音楽を楽しむことは難しく、更なる冗長な情報を削減することが求められている．その一方では、ADSL などの広帯域な接続環境が普及してきた背景もあり、今後オーディオ信号のロスレス符号化への需要も増加してくるものと考えられる．

MPEG ビデオの符号化手法と比較して、MPEG オーディオでは時間軸方向の相関を利用した冗長削減の割合が非常に小さい．しかし、DTM (Desktop Music)で作られた音楽など、特定の音楽ジャンルによっては類似したパターンが何度も繰り返される音楽ソースが存在する．本稿では、そのような類似パターンを音楽ソースの中から探索し、重複するパターンを完全に削減するロッシー方式と、類似パターンの差分情報のみを記録するロスレス方式の二通りのアルゴリズムについて提案する[1][2]．その概念図を図 1 に示す．

図 1　概念図

2. オーディオ・パターンマッチング

2.1 オーディオ・パターンマッチングの概要

本稿で提案するオーディオ・パターンマッチング方式の特長は、音楽ソースの中から類似パターンを探し出し、繰り返されるパターンは 1 度しか記録しないか、あるいはその差分情報のみを記録することにある．前者の場合、各パターンは必要に応じて更に MP3 などで圧縮することができるようになっており、既存の符号化技術と共存することが可能なのも大きな特長である．ファイルヘッダは、通常のオーディオファイルと違い再生順序情報を持っており、再生アプリケーションでは、その情報をもとに各パターンを途切れ目のないよう再生するため、原信号との聴感上の差は感じられないようになっている．

2.2 ファイルフォーマット

提案手法のために設定したファイルフォーマットは、ヘッダに符号化フォーマットの情報、再生順序情報、各パターンのサイズの各情報を持っている．再生アプリケーションは、まず符号化フォーマットの情報に従って符号化データをデコードし、サイズの情報から各パターンを取り出し、それを再生順序情報に従って再生する．再生順序情報は、ゲームの BGM など、無限にループする音楽を記述することもできるようになっている．ファイルフォーマットの詳細を図 2 に示す．

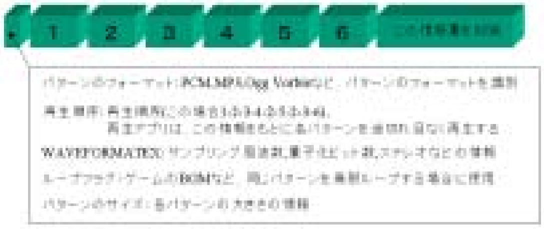

図 2 ファイルフォーマット

2.3 オーディオ・パターンマッチングの効果

オーディオ・パターンマッチングは、ファイルフォーマットさえ決めることができれば、波形編集ソフトなどを用いて手動で分割を行い、異なるパターンのみを抜き出してくることによってファイルを作成することも可能である．そのような手法を用い、市販されている音楽がオーディオ・パターンマッチングによってどの程度情報を削減できるかの予備実験を行ってみた．その結果を表 1 に示す．

	音楽 A	音楽 B	音楽 C	音楽 D	音楽 E	音楽 F
元の PCMファイル	55,364,820	47,359,429	41,196,620	53,395,748	59,455,252	65,973,644
パターンマッチング後	52,219,469	42,010,129	32,766,645	40,906,615	42,352,549	38,708,426
情報削減率[%]	5.69	11.0	20.4	23.2	28.5	41.3
パターンマッチング+MP3	4,736,413	3,889,681	2,956,932	3,710,184	3,886,196	3,513,736
平均ビットレート[kbps]	121	114	102	90.3	94.0	75.2

情報削減率,平均ビットレート以外の単位はbytes

表 1 予備実験の結果

音楽 A～C は一般的な日本のポップス、音楽 D～F はインストゥルメンタルの部分が多く、同じパターンが繰り返される確率がより高いと考えられるダンス系の音楽である．この予備実験により、A～C では 5%から 20%、D～F では 23%から 41%の情報を削減できることが確かめられた．また、主観評価ではあるが、この予備実験による音質劣化は認められなかった．

この予備実験の結果に加え、パターンマッチングによって整理されたオーディオデータを、更にMP3を用いて圧縮した結果も併せて示している．MP3は128kbpsでエンコードを行ったが、その平均ビットレートはいずれも128kbpsを下回っており、パターンマッチングの効果が生かされていることが確認された．

3. 符号化アルゴリズム

3.1 音楽分割アルゴリズムの概要

では、2.3で述べたような予備実験と同様のことを信号処理のみで実現するためには、どのようにすればよいか．筆者らは、これまで次のようなアルゴリズムを提案してきた[1][2]． 図3に、そのフローチャートを示す．

図3 符号化アルゴリズムのフローチャート

まず、音楽信号を適当な長さに分割する必要がある．ここで、MP3のようにある一定のサンプル数で音楽を分割してしまうと、分割されたブロックは音楽として意味のない長さになり、類似パターンを抽出することが困難になる．また、再生時にはパターンの途切れ目が不自然な音楽になってしまう．

これを回避するために、音楽信号にローパスフィルタを適用し、バスドラムなどのリズムを刻んでいる低音成分を抽出する[3]．次に、抽出された成分のピークを探索し、ピーク間のサンプル数に従って忠実に音楽ソースを分割する．このようにして分割された音楽信号は、一小節などといった音楽として意味のある長さになり、類似パターン抽出の難しさや再生時の不自然さは解消される．

次に、得られた各ブロックの中から、比較的サンプルの近いブロック同士に対してのみパターンマッチング処理を行う．パターンマッチング処理が終了したら、類似ブロックが連続している箇所同士を類似パターンと決定し、整理されたパターンと再生順序情報をファイルとして出力する．以上のアルゴリズムを図示したものを図4に示す．

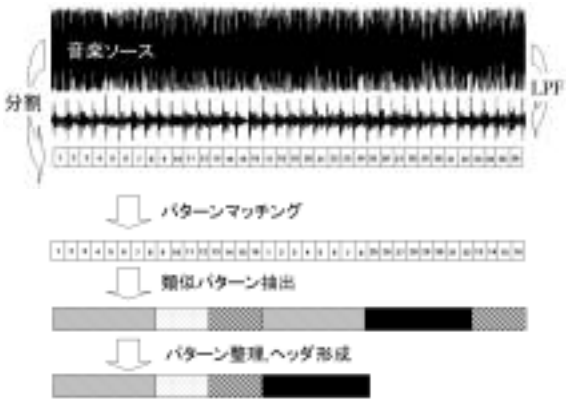

図4 符号化アルゴリズム

3.2 パターンマッチング手法の概要

オーディオ信号のパターンマッチングを行う際、問題となるのが信号に生じた時間的な伸縮である．これは、人間の聴覚がオーディオ信号の時間的な伸縮に対して鈍感であるために、伸縮した信号であっても聴感上は類似に聴こえることがあるため、単純に差分をとるだけのマッチングでは不十分だからである．特に、アナログ的な手法で編集された音楽は、信号に時間的伸縮が生じていることが多い．

本研究では、パターンマッチング手法としてDP(Dynamic Programming)マッチング法を導入した[4]．これは、変形した二つの信号間のマッチングに特化したアルゴリズムで、音声認識や指紋照合などの分野で広く用いられている．DPマッチングは、単純に差分をとっていくだけではなく、サンプルの挿入・削除の処理を行うことで、変形した信号に対しても効率的なマッチングを行う。単純に差分をとっていく移動マッチングとDPマッチングとの違いを図示したものを図5に示す．この手法により、二つのオーディオ信号間に生じた時間的な伸縮が吸収され、より精度の高いマッチングを行うことができると考えられる．通常の差分波形を図6に、DPを用いた場合の差分波形を図7にそれぞれ示す．

二つの信号間の距離をg(i,j)、局所距離をd(i,j)とするとき、DPマッチングは次の漸化式で定義される．

初期条件：$g(0,0)=0 \quad g(i,0)=g(0,j)=\infty$ for $i=1,2,\cdots,I$ $j=1,2,\cdots,J$

$i=1,2,\cdots,I$, $j=1,2,\cdots,J$

$$g(i,j)=\min\begin{cases} g(i-1,j)+d(i,j) \\ g(i-1,j-1)+2d(i,j) \\ g(i,j-1)+d(i,j) \end{cases}$$

オーディオ信号の場合、d(i,j)は二つの信号のサンプルの差の絶対値などとし、最後に計算されるg(I,J)を局所距離の加算回数で正規化した値を求める．通常、その値が最小となる組み合わせを類似パターンの組み合わせとしてマッチング結果を得る．

また、DP マッチングでは、どのようなルートが最小の距離となるかを表すパス情報が得られる．DP マッチングの漸化式を見ると、常に 3 つの経路の最小値を求めていく式になっているが、どの経路が最小だったかの情報がこのパス情報に相当する.ロスレス符号化を行う場合、このパス情報を保存することが重要となる．

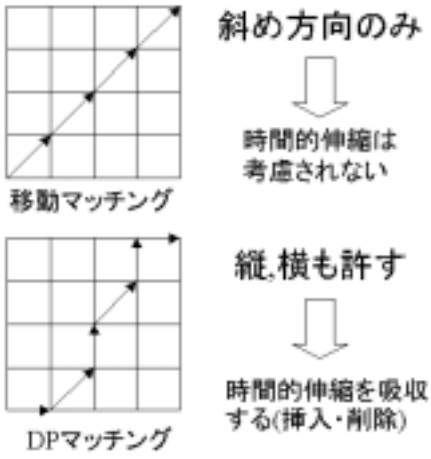

図 5　移動マッチングと DP マッチングの違い

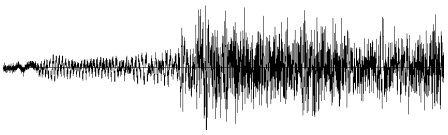

図 6　通常の差分波形

図 7　DP を用いた差分波形

3.3　パターンマッチング実験

3.1 の手法を用いて三種類の音楽から抽出された、それぞれ 20 個のブロックを用いて実験を行った．これらのブロックは、ビート検出で区切られた 20000 サンプル程度の大きさであり、ここでは 10 通りの類似した組み合わせを含んでいる．このとき、全ての組み合わせは 20C2=190 通りとなり、その全ての組み合わせについて DP マッチングを行う．ある一つのブロックに対して、もう一つのブロックとの距離が最小となる組み合わせを類似と判断し、それ以外の組み合わせは非類似と判断する．その結果が聴覚による判断と一致したものを正解とした．その正解率を表 2 に示す．

表2 パターンマッチングの正解率

	移動マッチング	DP マッチング
音楽1	97.4%	99.5%
音楽2	95.3%	96.3%
音楽3	94.7%	100%

3.4　ロッシーモード

オーディオ・パターンマッチングは、ロッシーモードとロスレスモードの二つのモードを定義している．ロッシーモードは、類似したパターンが存在した場合、そのパターンを一度だけ記録し、残りは完全に削除してしまい、記録したパターンを何度も使いまわすモードである．高い圧縮率が得られたり、MP3 などの他のロッシー符号化と共存できるといったメリットがある反面、パターンマッチングを誤った場合の音楽への影響が大きく、手作業による修正を完全に排除することは難しいモードである．

3.5 ロスレスモード

一方、ロスレスモードは音質を全く劣化させないモードで、類似パターン同士の場合は差分情報を記録するため、誤ったマッチングをしたとしても元の信号は完全に復号される．ロッシーモードでは元信号のみを記録したが、ロスレスモードの場合は元信号に加え、パス情報と差分情報を記録する．差分情報の符号化には既存のオーディオ信号用ロスレス符号化を、パス情報の符号化にはランレングス符号化をそれぞれ用いる．WaveZIP[5]や Monkey's Audio[6]などといった既存のオーディオ信号用ロスレス符号化は、過去のサンプルか

ら次のサンプルを予測する線形予測と L,R チャネルのどちらかに情報量を偏らせるステレオ符号化を用いて符号化を行っており、パターンマッチングは用いていない[7]. そのため、パターンマッチングと組み合わせても、その効果は生かされると考えられる. 線形予測とステレオ符号化の概念図を図 8,9 に示す. パス情報の符号化にランレングス符号化を用いるのは、類似した信号同士のマッチングほど、パス情報が斜め方向(挿入・削除を行わない)になる確率が高くなるためである. ある信号を A、その類似信号を A'とするとき、ロッシーモードとロスレスモードの違いを図 10 に示す.

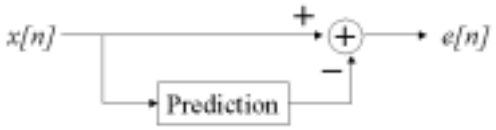

図 8 線形予測

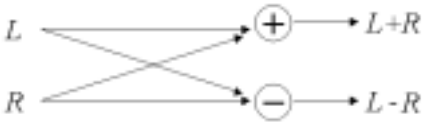

図 9 ステレオ符号化

図 10　ロッシーモードとロスレスモード

3.6　ロスレス符号実験

3.3 で用いたサンプル音楽 3 を用いてロスレス符号化の実験を行った. その結果を表 3 に示す. 組み合わせるロスレス符号化として、オーディオ信号用のロスレスコーデックとして定評のある Monkey's Audio を用いて実験を行った[6]. このアプリケーションは、処理に要する時間が非常に短いうえ、他のコーデックに比べて圧縮率も高い. Winamp のためのプラグインも用意されており、再生する環境も整っている優れたコーデックである. 表では、例えば 1-11(L)とは、ブロック番号 1 番と 11 番の組み合わせの L チャネルでの結果を示している. 符号化後の括弧内に示されているパーセンテージは、元の PCM ファイルを 100%とした場合の符号化後の割合を示している. 既存のコーデックのみを用いて符号化した場合に比べ、差分情報を記録する方式では、約 7%程度の圧縮率の向上を確認することができた.

	元の PCM ファイル	パスの情報量（圧縮前）	パスの情報量（圧縮後）	Monkey's Audio のみで符号化	差分情報とパスを符号化（手書方式）
1-11(L)	69424	17131	211	56808883.0M	52091076.1M
1-11(R)	69424	17204	252	56920883.2M	52080076.1M
2-12(L)	69112	17113	238	56804882.3M	51278075.3M
2-12(R)	69112	17085	208	56732882.4M	51348075.4M
3-13(L)	69104	17131	264	56544883.0M	51888076.2M
3-13(R)	69104	17084	199	56429882.9M	51515075.6M

表 3 パス情報と差分情報の情報量

3.7　実装例

2.2 で定めたファイルフォーマットを再生するアプリケーションの実装を Windows 上で行った. 実装した再生アプリケーションの様子を図 11 に示す. 機能的には再生・一時停止・停止のみのシンプルなものであるが、同じパターンを繰り返し使用している様子が視覚的に読み取れるようになっている. 2.2 で定めた独自フォーマット(*.apm)のほか、通常の Windows PCM (*.wav),MPEG-1 レイヤ 3 (*.mp3),Ogg Vorbis (*.ogg)も再生することができる. 更に高機能で操作性を良くするためには、Winamp などの有名なオーディオ再生アプリケーション向けのプラグインとして実装を行っていく必要がある. 今回実装したアプリケーションは、[8]に記された URL にて、音楽のサンプルとともにダウンロードすることが可能となっているので、興味のある方はテストして頂きたい.

図 11　実装した再生アプリケーション

4. 高速化に関する考察

DP マッチングに要した処理時間について考察する. これまでに述べたように、DP は時間的伸縮の生じやすいオーディオ信号のマッチングにおいて非常に効果的であるが、処理に要する時間が大きいという欠点がある.

今回実装したオーディオ信号用の DP マッチングプログラムでは、20000 サンプル前後のブロック同士を DP マッチングするのに、30〜60 秒程度を要する. 1 回のマッチングに要する時間自体は待てないほどではないが、通常一曲の音楽は数百個のブロックに分割されるため、一曲全体を処理するとなると相当な時間を要する.

SIMD 命令を用いてアセンブラレベルで CPU に最適化を施したり、ネットワークを利用して複数のコンピュータに並列処理させたりといったハードウェアの力を借りる方法もあるが、アルゴリズム的にもいくつか改善する必要がある.

一つの方法として、DP よりも処理時間の少なくて済む、DP 以外の処理から得られた前処理情報から類似・非類似をおおまかに分類し、類似と判断された組み合わせにのみ DP 処理を行う手法が考えられる. おおまかに分類するだけであれば、DP による差分信号を記録するロスレスモードにも適用可能である. 前処理情報としては、

・包絡線などの波形の形状情報
・ FFT や Wavelet 変換による周波数情報
・ブロックのサンプル数

などが挙げられる. これらは、時間領域 DP よりもはるかに少ない処理で結果を得られるため、これらの情報からおおまかな分類結果を得ることができれば、飛躍的に処理速度を上げることができると考えられる.

もう一つの方法としては、過去の DP の結果を常に記憶しておき、これまでで最小の DP の結果を越えてしまった時点で DP 処理を途中で打ち切るといった方法も効果的があると考えられる.

以上を考慮し、現在提案アルゴリズムの高速化について検討を進めている.

5. おわりに

オーディオ・パターンマッチングにより、同じパターンが繰り返し現れる音楽では、既存の符号化と共存させながら情報量を削減できることが確認された. また、差分情報を記録することにより、ロスレス符号化の場合であっても、予測符号化などを用いる従来のオーディオ用ロスレス符号化よりも高い圧縮率を得られることも確認できた. 今後の展望として、ビートを全く刻んでいない音楽への応用や DP マッチングの処理の高速化などに取り組んでいく予定である.

参 考 文 献

[1]松井唯史,甲藤二郎:”オーディオ・パターンマッチングに関する一検討”,2001 年 電子情報通信学会総合大会,D-14-11.

[2]松井唯史,甲藤二郎:”DP マッチングを用いたオーディオ・パターンマッチングの特性改善”,情報処理学会第 63 回全国大会,2Q-05.

[3]後藤真考,村岡洋一:”音楽音響信号を対象としたビートトラッキングシステム-小節線の検出と打楽器音の有無に応じた音楽的知識の選択”,情報処理学会音楽情報科学研究会報告 97-MUS-2-18,Vol.97,No.67,July 1997.

[4]中川聖一 著:”パターン情報処理”, 丸善.

[5]Gadget Labs WaveZIP: http://www.gadgetlabs.com/.

[6]Monkey's Audio: http://www.monkeysaudio.com/index.html.

[7]M.Hands,R.W.Schafer:“Lossless Compression of Digital Audio” IEEE Signal Processing Magazine, pp.21-32, July.2001.

[8]オーディオ・パターンマッチングのサイト: http://www.katto.commwaseda.ac.jp/~matsui//audiopm/index.html.