

平成 30 年度 特別研究報告書

MIDI 情報からファミコン音楽への
変換手法の提案

龍谷大学 理工学部 情報メディア学科

T150480 楠田一樹

指導教員 三好 力 教授

内容梗概

ファミコンの音源を取り入れたファミコン音楽はチップチューンという音楽ジャンルに含まれており、ファミコン音楽のピコピコ感が好きな現代の若者の心も掴んでいる。しかし、自分の好きな楽曲のファミコン音楽アレンジが無い場合は自分で作らなければならず、困難である。ファミコン音楽へ変換を行う既存技術は音の数を減らさず音色のみをファミコン音源に変換したり、リズムやテンポ、音の長さなどがぐちゃぐちゃでファミコン音楽らしくない問題点がある。

そこで、本研究では、ファミコン音楽らしい楽曲の定義である4つの音に絞り、既存楽曲をファミコン音楽らしい楽曲に変換することができる新しいアルゴリズムを提案し、既存技術の問題点を解決することを目的とし実験を行った。結果、既存技術の問題点の1つの正確な変換については解決出来たが、MIDIで定義されている矩形波ではファミコン音楽を表現することは難しく、改善が必要であることが分かった。

目次

第1章.はじめに.....	1
第2章.既存技術.....	3
2.1.「ピコピコンブ」.....	3
2.2.「Retro Music Editor」.....	4
2.3.MIDI.....	4
第3章.提案手法.....	8
3.1.概要.....	8
3.2.アルゴリズム.....	10
第4章.実験と考察.....	12
4.1.実験概要.....	12
4.2.実験結果.....	12
4.3.既存技術との比較結果.....	13
4.4.考察.....	20
第5章.おわりに.....	21
謝辞.....	22
参考文献.....	23

第1章 はじめに

ファミコンの音源を取り入れたファミコン音楽はチップチューンという音楽ジャンルに含まれており、日本のみならず海外のアーティストも自分たちの楽曲にファミコン音楽を取り入れている。このファミコン音楽は、当時ファミコンでゲームをしていた人だけでなくファミコン音楽のピコピコ感が好きな現代の若者の心も掴んでいる。しかし、自分の好きな楽曲のファミコン音楽アレンジが無い場合は自分で作らなければならない、その作業は非常に困難なものになっている。

ファミコン音楽とは、ファミリーコンピュータ内部の音源・ファミコンカートリッジ内の拡張音源で構成された楽曲のことを示し、音楽ジャンルとしてはチップチューンに含まれる。また、そのファミコン音楽を構成するファミコン音源とは、ファミリーコンピュータに CPU 内蔵されている RP2A03 が搭載している音源のことであり、PSG(Programmable Sound Generator)音源という音を作り出す電子回路の一種とも呼ばれている。搭載されている音色として、矩形波(パルス波)2音、三角波1音、ノイズ1音の4つの音色が搭載している。また、RP2C33、SUNSOFT 5B、MMC5 などファミコンカートリッジによって様々な拡張音源があり、ファミリーコンピュータの持つ音源をファミコンカートリッジに内蔵されている拡張音源で拡張することで、カートリッジごとに様々なBGMや効果音を出すことが出来る。

基本的にファミコンのCPUに内蔵されている音源は、上記で述べた4つの音色で構成されており、同時にさせる音の数も規格上4つまでとなっている。一概にファミコン音楽にこれといった定義はないが、本論文では矩形波2音、三角波1音、ノイズ1音の4つの音色の音源で構成された楽曲をファミコン音楽と定義する。

既存楽曲からファミコン音楽への変換の際は、それらの条件を踏まえた上で変換しなければいけない。この変換の処理プロセスをプログラムによって行うことが出来れば、より容易に既存楽曲からファミコン音楽への変換を行うことができるのではないかと考える。

本研究では、既存楽曲の音の数を減らし、残された音をファミコン音源へ変換しファミコン音楽を作るプログラムに関して、C言語を用いて、既存楽曲からメロディやリズムのずれがなく、ファミコン音楽らしい楽曲の定義である4つの音に絞り、既存楽曲を忠実にファミコン音楽で再現するファミコン音楽らしい楽曲に変換することができる新しいプログラムを提案し、既存技術の問題点を解決することを目的とする。

いくつもの音の数から成る既存楽曲をファミコン音楽の定義に当てはまるよう音の数を減らし、残された音をファミコン音源へと変換しファミコン音楽を完成させる処理を行うアルゴリズムを提案する。本研究では、MIDIファイルからファミコン音源への変換を対象とした。MIDIファイルは、主旋律や副旋律、伴奏などが一つの音声として記述されているMP3やWAVなど音声ファイルではなく、それらの楽曲情報が楽

譜のように記述され、音声ファイルに比べると処理が容易である。また、既存技術との変換結果を比較するため、既存楽曲の原曲と本研究の提案アルゴリズムでの処理楽曲と既存技術での処理楽曲を聴き比べてもらうアンケート調査を行い、結果を検討する。

本論文では、第 2 章でファミコン音楽の定義と既存技術の概要と問題点、MIDI ファイルの構造について述べ、第 3 章では本研究の提案するプログラムのアルゴリズムの概要を述べる。第 4 章では、提案プログラムの処理結果について述べ、既存技術との比較をするためのアンケート調査の結果とその結果についての考察を述べる。そして最後に第 5 章で本論文のまとめと今後の課題について述べる。

第2章 既存技術

2.1 「ピコピコンブ」

ピコピコンブとは MP3 や AAC、FLAC、WAV、MIDI など様々な楽曲ファイルをピコピコ音源風に、つまりファミコン音楽に変換することができる Android アプリである[1]。また、変換の際に、三角波、短形波、ノコギリ波など様々な音色に設定することができ、変換後は AAC ファイル、WAV ファイルで保存することができる。

しかし、実際に変換してみると、主旋律のメロディは変換されているが、主旋律のハモリのメロディや伴奏が途切れている箇所や、リズムがところどころ合っていない。この原因は、再生されている音声ファイルの音波を解析し、検出を行っていると考えられるので、主旋律のメロディは認識できるものの、伴奏やハモリがあまり検出することができないと考えられる。また、音の数は減らせてはいるものの、6音7音同時に音を鳴らしている箇所もあるためファミコン音楽らしくないと考えられる。

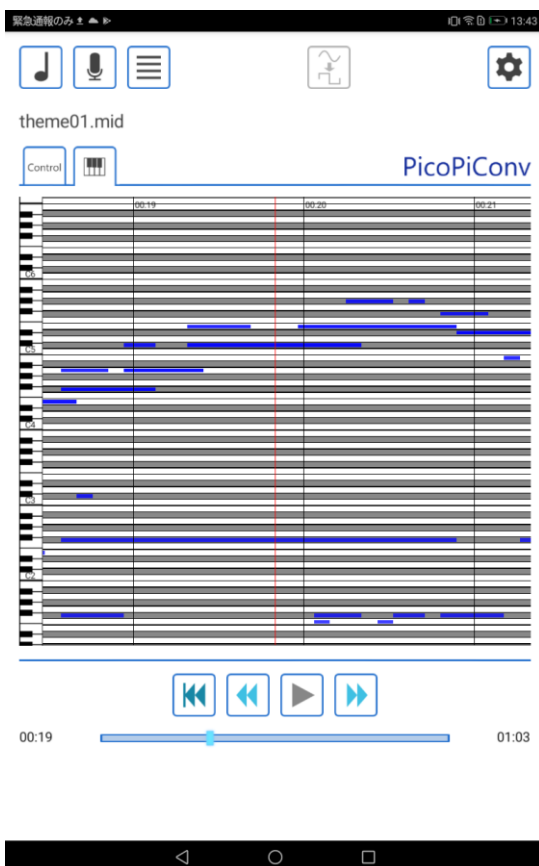


図 1:ピコピコンブ(楽曲を読み込んだ際の画面)

2.2 「Retro Music Editor」

Retro Music Editor(以下 RME)は自分でファミコン音楽を作ることができるエディターである[2]。また、MIDI ファイルを読み込むことも可能であり、読み込んだ MIDI ファイルの音源をファミコン音源に変換することができる。変換した音源は、WAV ファイルで保存することが可能になっている。

この RME で変換したファミコン音楽は前述のピコピコンブとは違い、リズムのずれもなく、正確にすべての音をファミコン音源に変換できるものの、全ての音をファミコン音源に変換してしまうので音の数が多く、音が何重にも重なって聞こえ、ファミコン音源らしくないと考えられる。また、読み込んだ曲によってはテンポが遅くなった状態のものもあり、自分でテンポを合わせないといけない点もある。

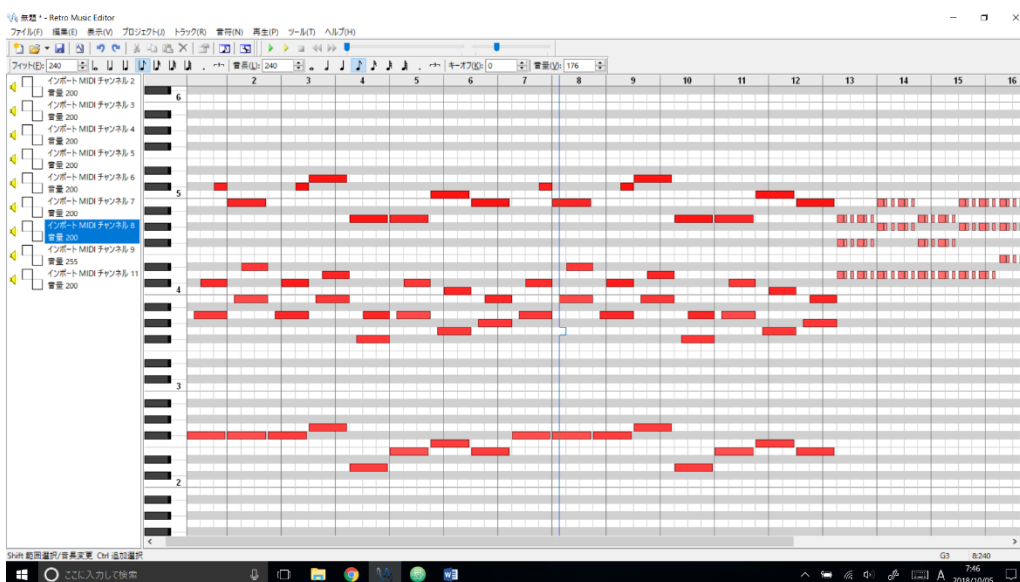


図 2:RME(楽曲を読み込んだ際の画面)

2.3 MIDI

MIDI(Musical Instrument Digital Interface)とは、日本の MIDI 規格協議会と国際団体の MIDI Manufacturers Association により策定された電子楽器の演奏データを機器間でデジタル転送するための世界共通規格である。MIDI 規格上のデータの送受信は、すべて MIDI メッセージで行われており、8 ビットで構成されている[3]。

MIDI メッセージはチャンネルメッセージとシステムメッセージの 2 種類に分けられる。チャンネルメッセージは、特定の MIDI チャンネルに作用するメッセージで用途に応じてチャンネルボイスメッセージとチャンネルモードメッセージの 2 種

類が用意されている。以下に MIDI メッセージをカテゴリ化した図を示す。

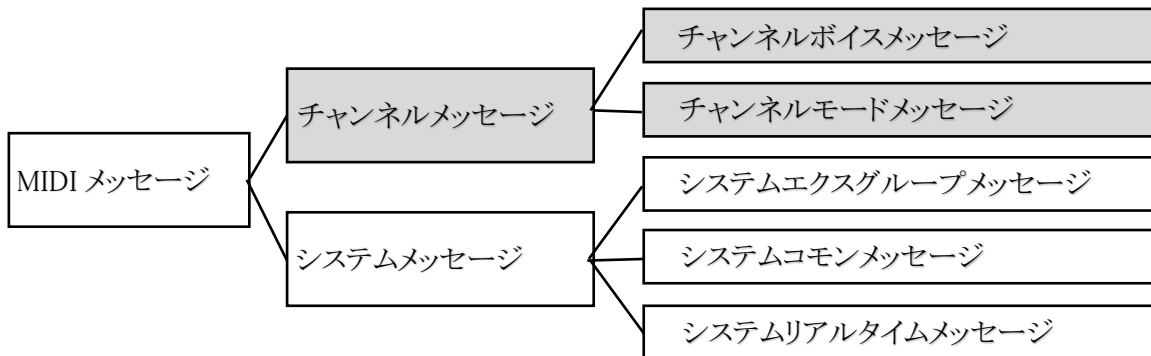


図 3: MIDI メッセージのカテゴリ

中でも、チャンネルボイスメッセージは発音/消音や音色など演奏情報を持っており、ノートオン(発音)、ノートオフ(消音)、コントロールチェンジ(音色調整)などの様々な種類があり、ステータスバイトとデータバイトの 2 種類に分けられる。ステータスバイトではノートオンやコントロールチェンジなどを定義しており、データバイトはステータスバイトで定義されたものの、内容や任意の数値を指定している。また、通常メッセージの種類を示すステータスバイト×1 バイトと、その値を示すデータバイト×1ⁿ数バイトから構成されており、そのメッセージの種類によっては、ステータスバイトのみで構成されるものや、データバイト部分が数 K バイトもの長さを持っているものもある。次項に、例として音階 C を鳴らした時のノートオンメッセージのステータスバイト、データバイトの構成を示す。

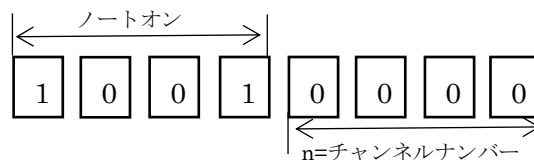


図 4: 音階 C を鳴らした時のノートオンメッセージのステータスバイト

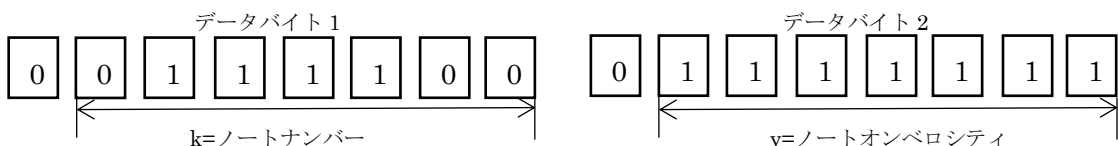


図 5: 音階 C を鳴らした時のノートオンメッセージの 2 つのデータバイト

図 4 より、ノートオンメッセージの上位 4 ビット 1001 はメッセージの種類を表す部分で固定となっており、下位 4 ビット 0000 は 1~16 の MIDI チャンネルという MIDI が持つチャンネルのそれぞれの番号を表す可変値である。この方式に

より、ステータスバイトを見るだけで、メッセージの種類とMIDIチャンネルが何番かがわかるようになっている。MIDIチャンネルの2進数表記は少し異なるため、下記の表に示す。図4の場合、チャンネルは1になる。

表1:MIDIチャンネルの2進数表記対応表

MIDIチャンネル	2進数	MIDIチャンネル	2進数
1	0000	9	1000
2	0001	10	1001
3	0010	11	1010
4	0011	12	1011
5	0100	13	1100
6	0101	14	1101
7	0110	15	1110
8	0111	16	1111

図5では、ノートナンバーは音階を表しており、ノートオンベロシティは音の強さ、つまり鍵盤を押すときの速さ(強さ)を表している。ノートナンバーは中央Cの音階を60として0~127までの範囲を、ノートオンベロシティは1~127の範囲を表している。

しかし、ここでMIDIメッセージの内容を0と1の2進数で表す方法は少しわかりにくいので、これ以降は代わりに0~Fまでの16進数で表していく。よって、図4、5のノートオンメッセージのステータスバイト、データバイトは16進数であらわすと以下ようになる。

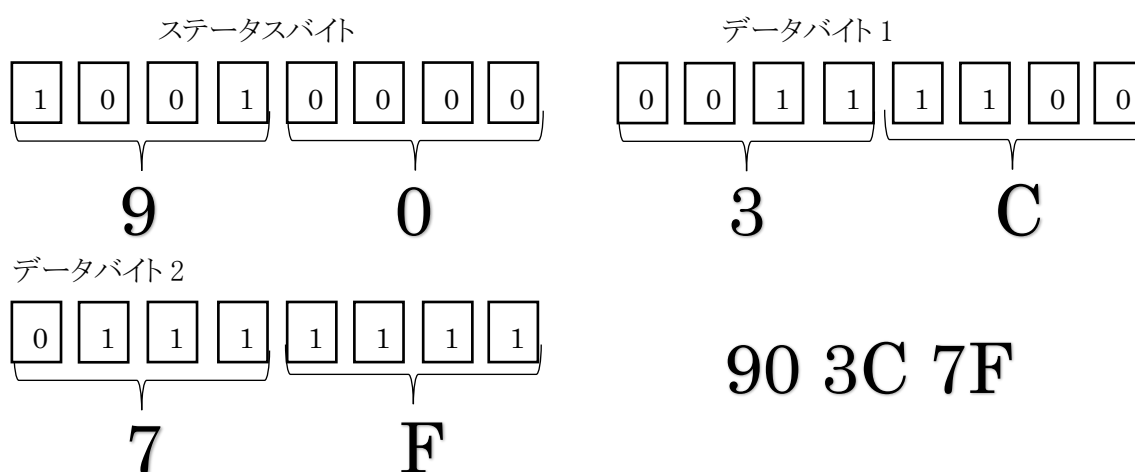


図6:16進数でのMIDIメッセージ表記

この方法を使い、ノートオン、ノートオフメッセージを可変変数を表す小文字英字で表すとノートオンは「9n kk vv」、ノートオフは「8n kk vv」と表せる。

そして、MIDI データの全体としては、ヘッダ部とトラックデータ部からなり、トラックデータ部は前述のステータスバイト(システムコマンド)、データバイト(音の高さ、音の長さ、音の強さ)の各情報単位を2進数8ビット、つまり16進数2桁を組としたデータ列からなる。

第3章 提案手法

3.1 概要

一般的に、楽曲はファミコン音楽の定義である4音にとどまらず、数々の音の数で構成されている。この楽曲を4音にするにはいかにして元の楽曲の構成を崩さずに音の数を減らすのかが重要になってくる。

音楽は3つの要素、リズム(律動)、メロディ(旋律)、ハーモニー(和声)で構成されると言われている。その中で、ハーモニーはメロディとの相対的な倍音関係を構成しており、メロディをより際立たせる役割を持っている。また、このハーモニーは和音とも呼ばれる。和音とは高さが異なる複数のピッチクラスの楽音が同時に響く音のことである。参考文献[4]では、「広く和音の定義を考えると、「2つ以上の音が同時に鳴ったもの」ということになろう」(p.3)と和音の定義がされており、本論文でもこれを定義する。また、3つのピッチクラスから成る和音を三和音、4つのピッチクラスから成る和音を4和音などと呼ぶ。

本研究で提案するアルゴリズムは、この和音の一番音階が低い音の根音(ルート音)だけを抜き取り、和音を単音にするアルゴリズムを提案する。この処理を行う理由としては、根音は和音が構成する音の中で基準の音であり、これを削って単音にしてしまうと、楽曲の音程のバランスがずれてしまうからである。この和音の中の根音を残し、他の音を削り、単音とすることで楽曲のバランスを崩さないようにすることが出来るのではないかと考える。

また、ファミコン音楽で和音を表現する手法として、分散和音(アルペジオ)がある。分散和音とは、和音を構成する全ての音を何度かに分けて単音で鳴らすことで和音のような音を表現する手法である。和音を分散和音にすることに関しては、本研究のアルゴリズムには加えず、アンケート調査の際に、手動で作成した分散和音処理を加えた楽曲と本研究提案アルゴリズムと既存技術で処理した楽曲を聴き比べてもらい、本研究の提案アルゴリズムに加え、分散和音についても検討し、考察する。

MIDI ファイルで和音を表す羅列は「9n(トラックチャンネル) kk(音階) vv(ベロシティ) 9n ll vv 9n mm vv」といったノートオンが連続で並ぶ羅列になっており、この羅列の後にノートオンで発生している音を消すノートオフが並ぶ。この和音の構成を考慮し、変換処理を行う。

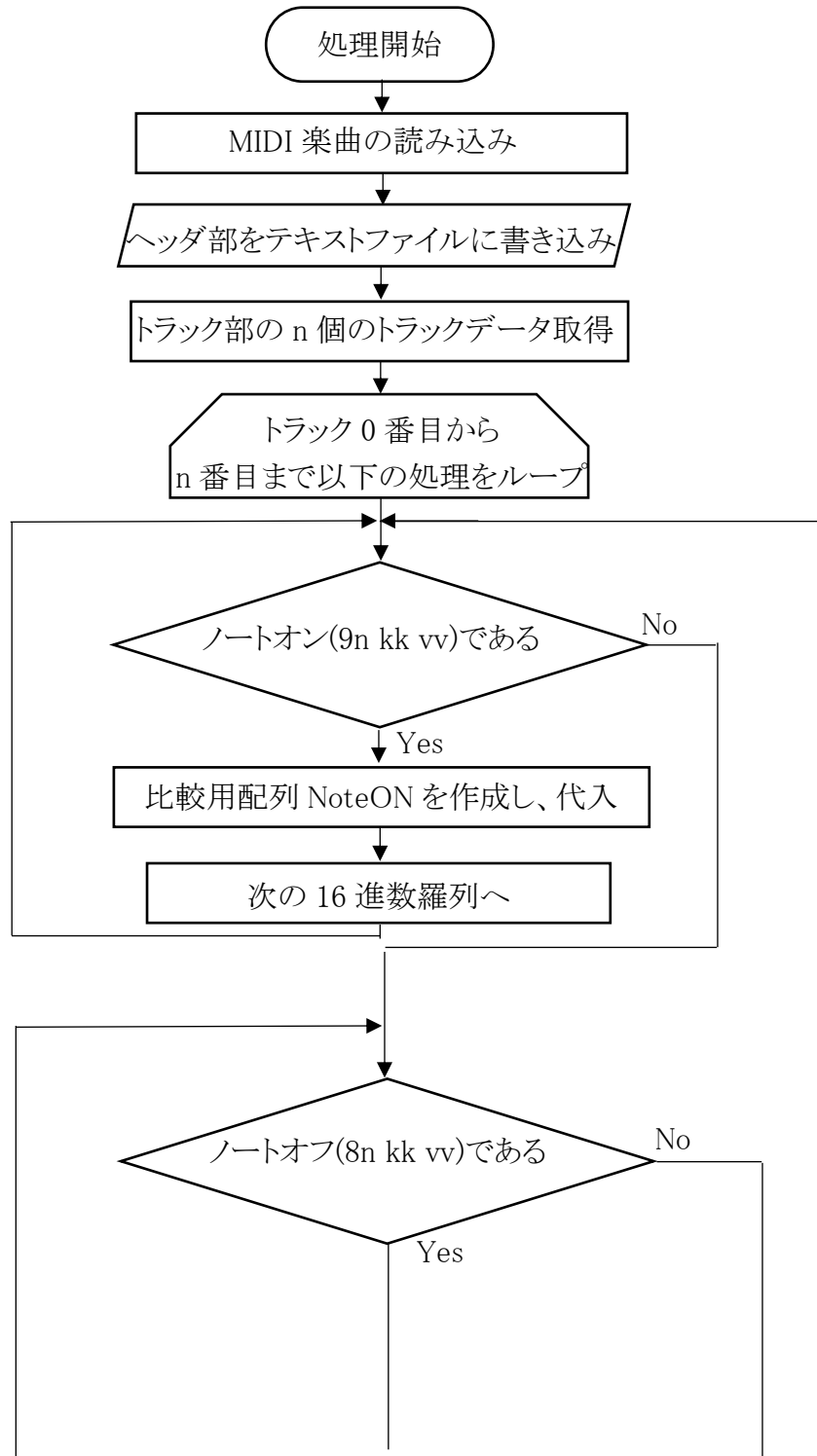
まず楽曲情報を読み込み、ヘッダ部の情報をテキストファイルに書き込む。その後、トラックデータ部を取得し、1つ1つのトラックデータに音色変換(プログラムチェンジ)と単音変換とサイズ調整のループ処理を行う。まず、ノートオン、ノートオフの羅列を共に個別の比較用配列に代入していき、それぞれの根音だけを

取り出した情報羅列を根音用配列に代入していく。ノートオン、ノートオフの羅列だけ処理を行いたいのので、波長など他の羅列情報に処理はせず、根音用配列に代入していく。トラックデータの終了部分を確認したら、トラックデータ部のヘッダ情報をテキストファイルに記述していく。その際、音を減らした状態でサイズを減らす前の情報にしていると破損してしまうため、減らした音の数だけサイズを減らすように適正なサイズ情報を記述する。1つのトラックのトラックデータ部のヘッダ情報を飛ばし、最初に根音抜き出し処理を行う理由はこのためである。そして、音色の情報は、「Cn pp」で表記されており、この pp の部分が音色を表しており、その音色は MIDI 音源の General MIDI(以下 GM)で 128 種類の楽器の音色が定義されている。矩形波と三角波に変更をするため、81 番のシンセ・ソロ音色 1(矩形波)、82 番のシンセ・ソロ音色 2(鋸状波)に変更を行うよう記述し、根音用配列に代入されている根音の情報をヘッダ情報に続き記述していく。

全てのトラックで書き込み処理を終えたら既存曲のヘッダ部と抜き出された音の情報が 16 進数羅列で書き込まれたテキストファイルを出力し、そのテキストファイルをバイナリ化し、MIDI ファイルで書き出す。和音を単音にするこの処理のアルゴリズムを次項の図 7 に示す。

3.2 アルゴリズム

4.1 の概要で述べたアルゴリズムを以下の図 7 に示す。



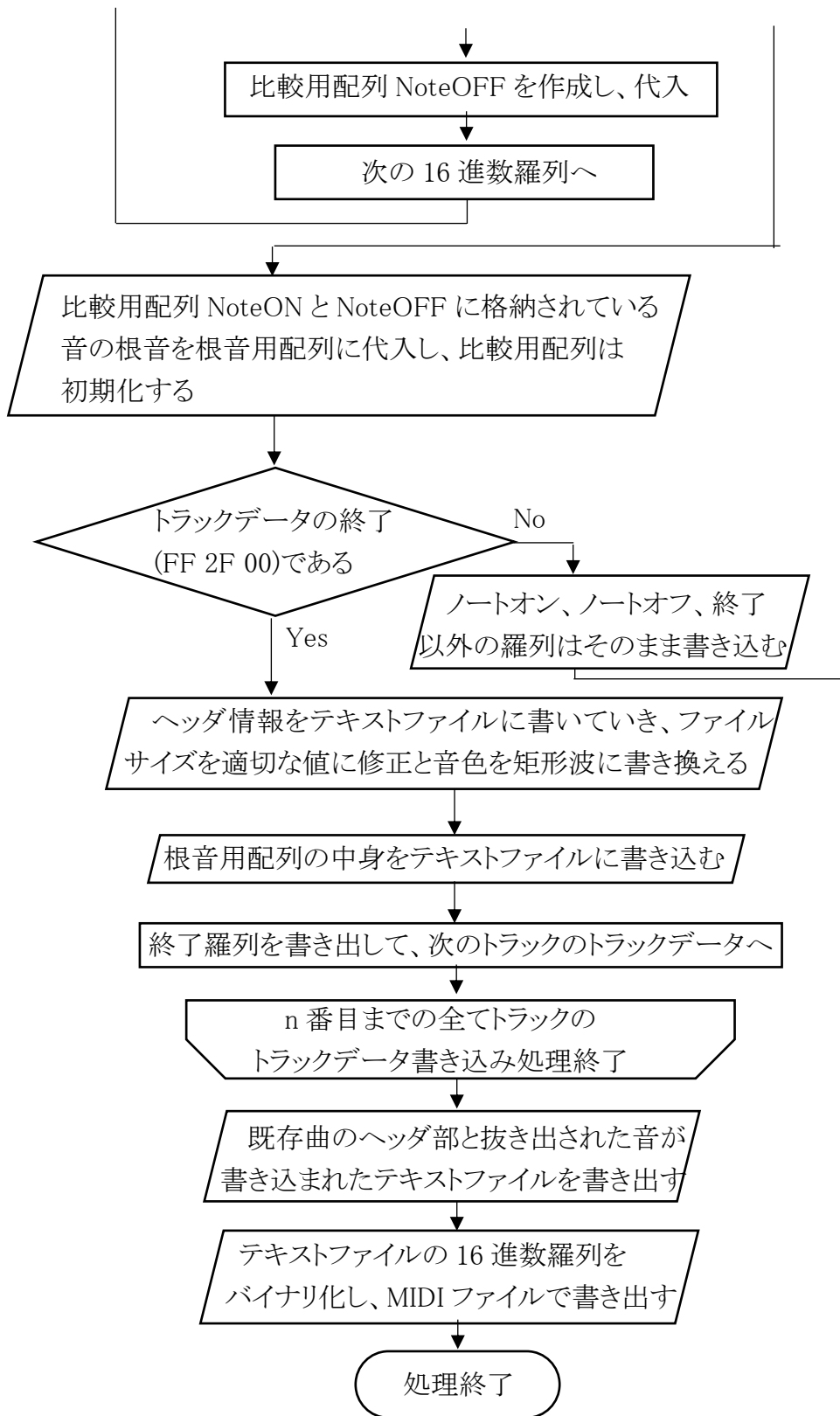


図 7: 提案アルゴリズムのフローチャート

4 章 実験と考察

4.1 実験概要

本章では本研究で提案する変換プログラムの結果と考察について述べる。実験で用いた MIDI 楽曲はフリー音源配布サイト「魔王魂」[10]からダウンロードしたそれぞれジャンルの異なる MIDI 楽曲 5 曲である。この実験を以下の環境で行った

OS :Ubuntu 17.0.4 , Windows10

言語:C

エディタ:Atom 1.26.1 x64

また、変換した MIDI 楽曲の状態をピアノロールで分かりやすく表示するために、Windows10 にて「Domino」[11]という MIDI シーケンサを用いた。

4.2 実験結果

実験を行った結果、5 曲のうち、2 曲は再生の際に破損ファイル扱いとなり変換に失敗したが、残りの 3 曲は変換に成功した。変換に成功した楽曲 3 曲のうち 1 曲のピアノロールの一部の比較画像を以下に示す。

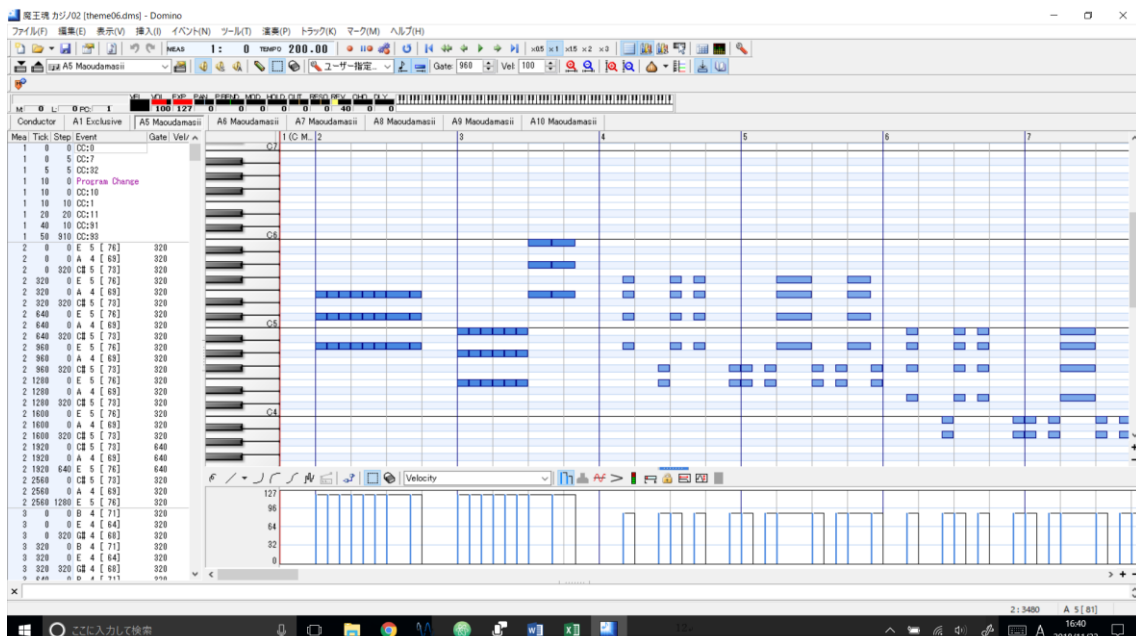


図 8.1 変換前のピアノロール画面

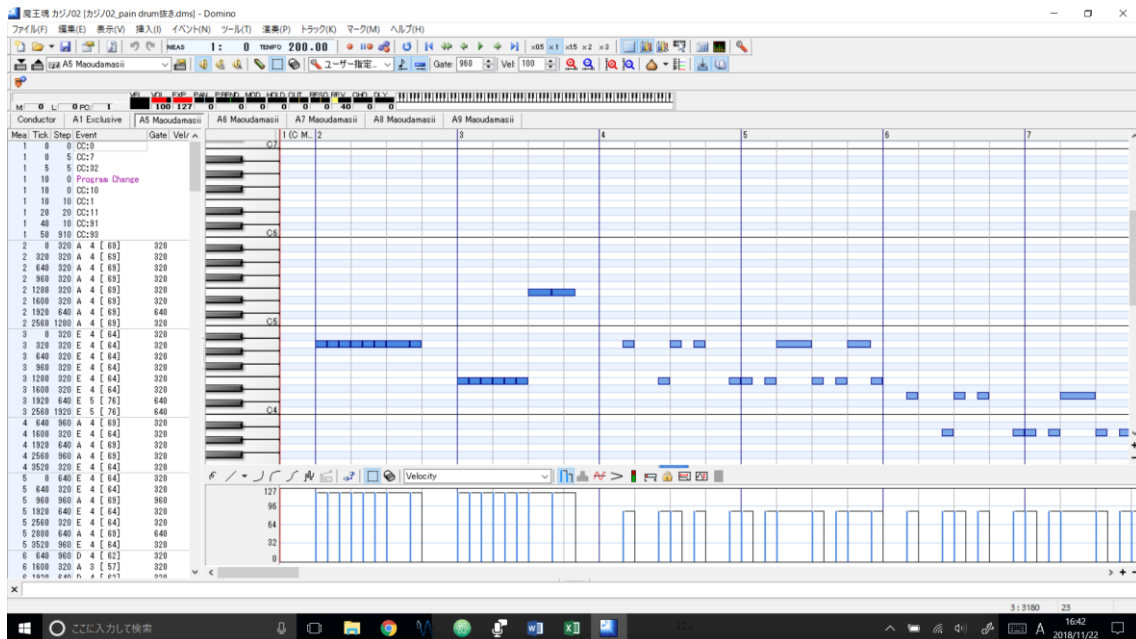


図 8.2 変換後のピアノロール画面

和音の部分は根音だけが残り、処理が成功していることが分かった。処理に失敗した原因については 4.4 考察で述べる。

4.3 既存技術との比較結果

先ほど述べた実験で変換に成功した MIDI 楽曲 3 曲について、本研究で提案したアルゴリズムと分散和音処理と既存技術である「ピコピコンブ」と「RME」で変換した楽曲について比較した結果について述べる。

比較をするための方法として 20 代の男性 10 人女性 2 人にアンケート調査を行った。ファミコン音楽に興味があるかどうかの質問から始め、変換処理を行ったソフトの名前は伏せ、本研究提案アルゴリズムと和音部分に分散和音処理をしたものとピコピコンブと RME で処理を施したそれぞれの楽曲と原曲を聴き比べてもらい、どのソフトの変換処理結果の楽曲が一番ファミコン音楽らしいか、一番原曲に近い(原曲からの変換であると判別できるか)、どの変換処理が好みかを 1~5(1:全く思わない、2:やや思わない、3:よくわからない、4:やや思う、5:とても思う)までの 5 段階評価で回答してもらった。次項にアンケートの質問内容と結果を示す。また、No は、実験した順番を表す。性は 1 が男子、2 は女子を表す。

表 2 アンケートの質問内容

以下の質問に5段階評価で回答してください。
Q1:あなたはファミコン音楽に興味がありますか
楽曲Aについて、①～④の変換処理はファミコン音楽らしいかお答えください
Q2.1:①の変換処理はファミコン音楽らしいですか
Q2.2:②の変換処理はファミコン音楽らしいですか
Q2.3:③の変換処理はファミコン音楽らしいですか
Q2.4:④の変換処理はファミコン音楽らしいですか
楽曲 A について①～④までの変換処理は原曲に近いかお答えください
Q2.5: ①の変換処理は原曲に近いですか
Q2.6: ②の変換処理は原曲に近いですか
Q2.7: ③の変換処理は原曲に近いですか
Q2.8: ④の変換処理は原曲に近いですか
Q2.9:①～④までのどの処理があなたの好みか、一つお答えください
楽曲 B について、①～④の変換処理はファミコン音楽らしいかお答えください
Q3.1:①の変換処理はファミコン音楽らしいですか
Q3.2:②の変換処理はファミコン音楽らしいですか
Q3.3:③の変換処理はファミコン音楽らしいですか
Q3.4:④の変換処理はファミコン音楽らしいですか
楽曲 B について①～④の変換処理は原曲に近いかお答えください
Q3.5: ①の変換処理は原曲に近いですか
Q3.6: ②の変換処理は原曲に近いですか
Q3.7: ③の変換処理は原曲に近いですか
Q3.8: ④の変換処理は原曲に近いですか
Q3.9:①～④までのどの処理があなたの好みか、一つお答えください
楽曲 C について、①～④の変換処理はファミコン音楽らしいかお答えください
Q4.1:①の変換処理はファミコン音楽らしいですか
Q4.2:②の変換処理はファミコン音楽らしいですか
Q4.3:③の変換処理はファミコン音楽らしいですか
Q4.4:④の変換処理はファミコン音楽らしいですか
楽曲 C について①～④までの変換処理は原曲に近いかお答えください
Q4.5: ①の変換処理は原曲に近いですか
Q4.6: ②の変換処理は原曲に近いですか
Q4.7: ③の変換処理は原曲に近いですか
Q4.8: ④の変換処理は原曲に近いですか

Q4.9:①～④までのどの処理があなたの好みか、一つお答えください

表 3.1 アンケート結果 1

No	歳	性	Q1	Q2.1	Q2.2	Q2.3	Q2.4	Q2.5	Q2.6	Q2.7	Q2.8	Q2.9
1	22	1	4	5	5	4	2	5	5	4	1	1
2	21	2	4	4	4	5	2	4	5	5	1	3
3	22	2	1	3	3	4	4	5	4	4	2	3
4	22	1	1	3	3	4	5	4	3	2	1	1
5	22	1	5	4	4	3	5	2	2	2	1	4
6	21	1	4	3	2	4	5	4	4	3	1	2
7	22	1	5	2	1	1	3	1	1	1	1	1
8	22	1	3	2	3	4	3	5	4	4	1	3
9	22	1	1	4	4	3	5	2	2	3	1	2
10	22	1	5	4	3	3	2	4	4	3	2	2
11	21	1	5	5	5	4	4	4	5	4	3	2
12	23	1	2	3	4	4	5	3	2	1	3	3

表 3.2 アンケート結果 2

No	歳	性	Q3.1	Q3.2	Q3.3	Q3.4	Q3.5	Q3.6	Q3.7	Q3.8	Q3.9
1	22	1	4	4	3	2	5	5	4	1	1
2	21	2	5	5	5	1	4	5	5	1	2
3	22	2	2	2	4	3	4	4	4	2	3
4	22	1	3	3	4	4	3	3	4	1	1
5	22	1	3	3	4	4	4	4	2	1	1
6	21	1	4	4	3	1	4	4	3	1	3
7	22	1	5	5	3	4	4	3	2	1	1
8	22	1	2	2	4	3	5	5	3	1	3
9	22	1	4	4	3	1	3	4	3	1	2
10	22	1	4	4	2	2	4	4	3	2	2
11	21	1	5	5	4	4	4	5	5	3	2
12	23	1	2	2	4	5	4	4	3	2	3

表 3.3 アンケート結果 3

No	歳	性	Q4.1	Q4.2	Q4.3	Q4.4	Q4.5	Q4.6	Q4.7	Q4.8	Q4.9
----	---	---	------	------	------	------	------	------	------	------	------

1	22	1	2	2	2	1	4	4	4	1	1
2	21	2	3	5	5	1	2	5	5	1	3
3	22	2	2	2	5	3	2	2	4	1	3
4	22	1	3	3	5	4	3	3	4	1	1
5	22	1	4	5	4	2	5	4	3	2	3
6	21	1	2	2	4	3	4	2	4	1	3
7	22	1	4	3	2	4	3	3	2	1	4
8	22	1	4	3	4	2	5	5	4	1	1
9	22	1	4	5	5	1	2	2	1	1	3
10	22	1	5	5	3	2	4	4	3	2	2
11	21	1	5	5	4	4	4	4	5	2	3
12	23	1	3	3	4	5	3	2	2	1	4

また、A、B、C の楽曲ごとにファミコン音楽らしいかという観点において、既存技術の RME と提案アルゴリズムの評価点数での平均と標準偏差の図を以下に示す。

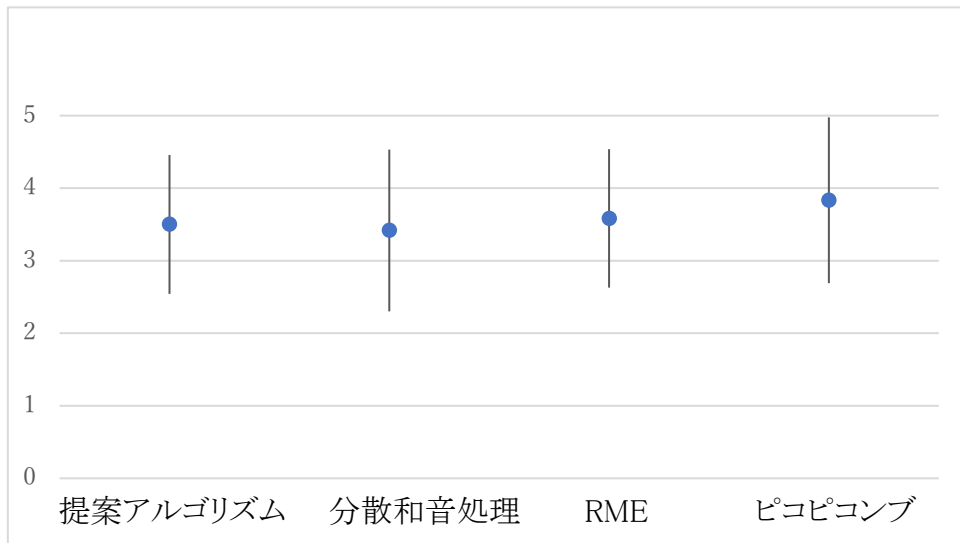


図 10.1 MIDI 楽曲 A におけるファミコン音楽らしいかの平均と標準偏差

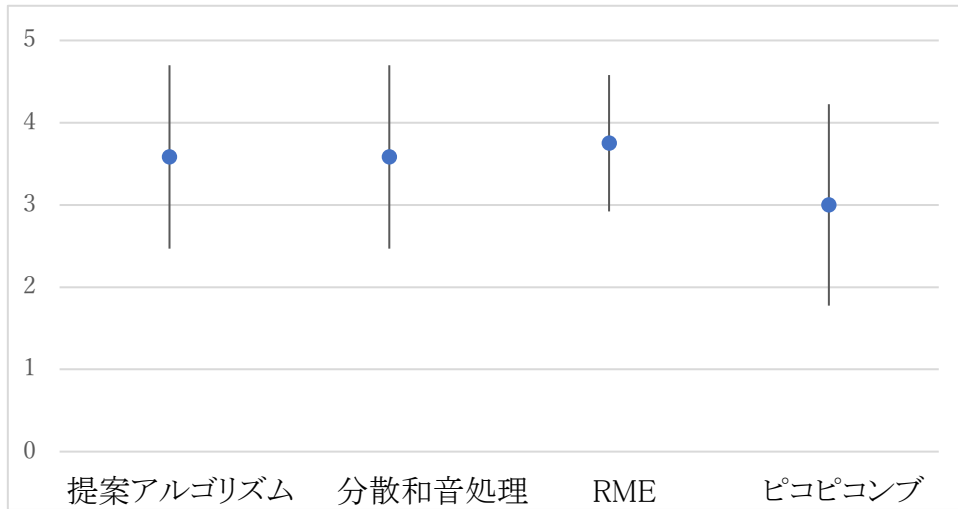


図 10.2 MIDI 楽曲 B におけるファミコン音楽らしいかの平均と標準偏差

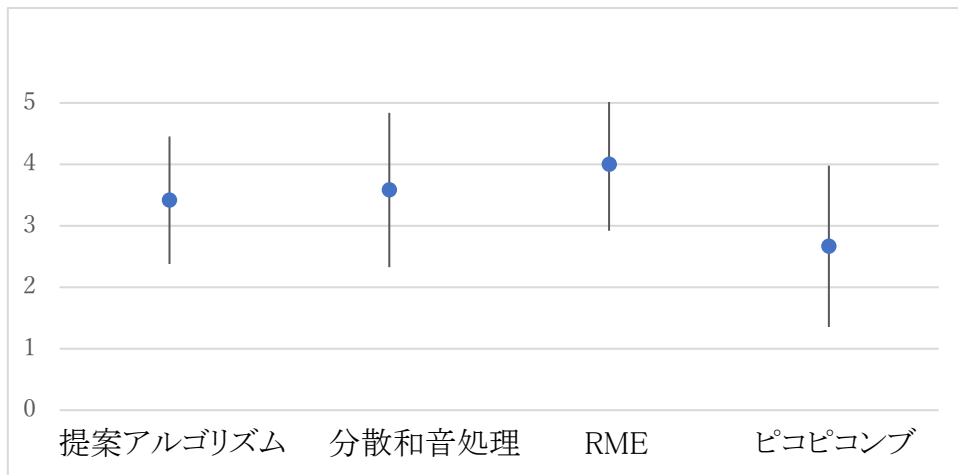


図 10.3 MIDI 楽曲 C におけるファミコン音楽らしいかの平均と標準偏差

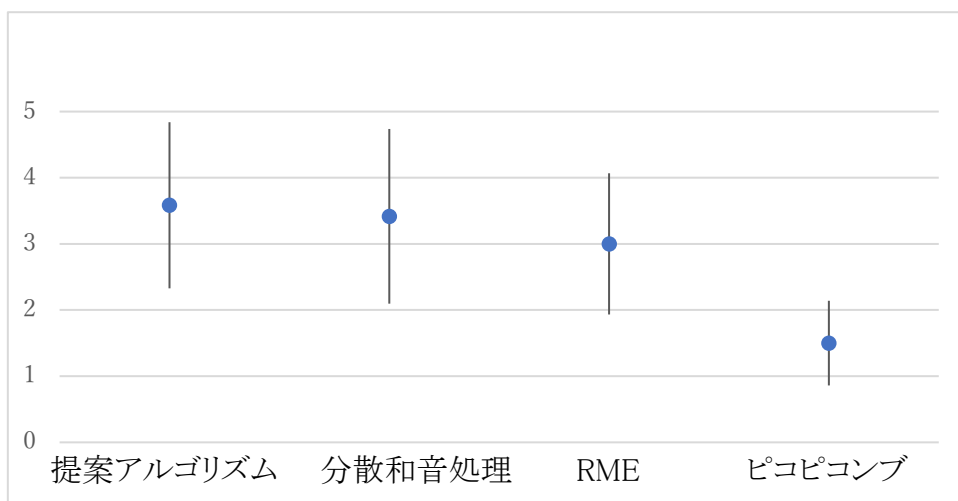


図 10.4 MIDI 楽曲 A における原曲に近いかの平均と標準偏差

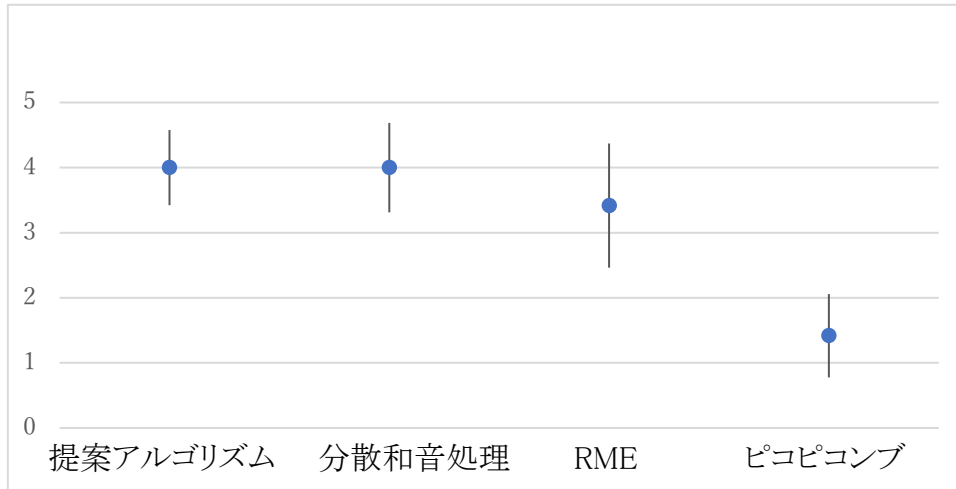


図 10.5 MIDI 楽曲 B における原曲に近いかの平均と標準偏差

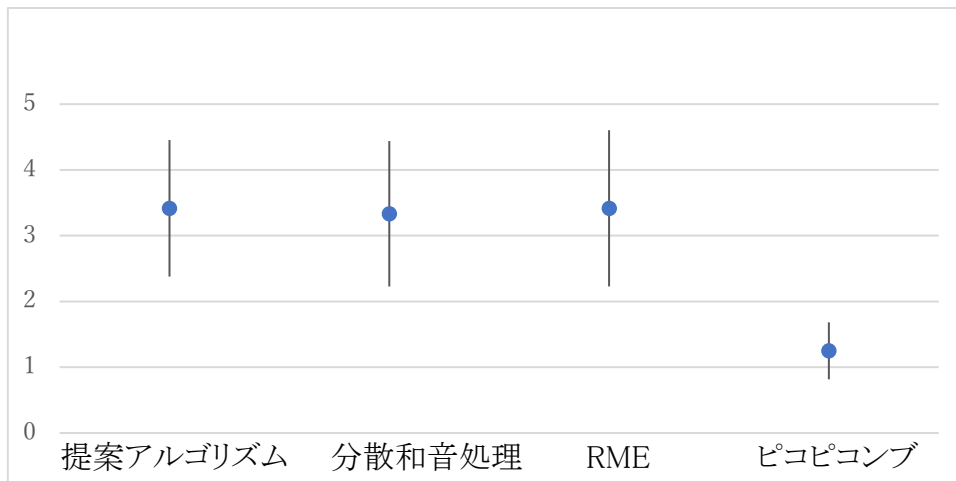


図 10.6 MIDI 楽曲 C における原曲に近いかの平均と標準偏差

次に、提案アルゴリズムとの比較を行うため、T 検定を行った。以下に T 検定の結果を示す。

表 2.1 MIDI 楽曲 A におけるファミコン音楽らしいかの T 検定

	t 値	有意確率(両側)
提案アルゴリズム - 分散和音処理	0.432	0.674
提案アルゴリズム - RME	0.248	0.809
提案アルゴリズム - ピコピコンブ	0.491	0.633

表 2.2 MIDI 楽曲 B におけるファミコン音楽らしいかの T 検定

	t 値	有意確率(両側)
提案アルゴリズム - 分散和音処理	error	error
提案アルゴリズム - RME	0.378	0.712
提案アルゴリズム - ピコピコンブ	1.216	0.250

表 2.3 MIDI 楽曲 C におけるファミコン音楽らしいかの T 検定

	t 値	有意確率(両側)
提案アルゴリズム - 分散和音処理	0.692	0.504
提案アルゴリズム - RME	1.246	0.239
提案アルゴリズム - ピコピコンブ	1.517	0.157

表 2.4 MIDI 楽曲 A における原曲に近いかの T 検定

	t 値	有意確率(両側)
提案アルゴリズム - 分散和音処理	0.804	0.438
提案アルゴリズム - RME	2.028	0.067
提案アルゴリズム - ピコピコンブ	5	0

表 2.5 MIDI 楽曲 B における原曲に近いかの T 検定

	t 値	有意確率(両側)
提案アルゴリズム - 分散和音処理	1	0.339
提案アルゴリズム - RME	1.735	0.111
提案アルゴリズム - ピコピコンブ	9.940	0

表 2.6 MIDI 楽曲 C における原曲に近いかの T 検定

	t 値	有意確率(両側)
提案アルゴリズム - 分散和音処理	0.248	0.809
提案アルゴリズム - RME	0	1
提案アルゴリズム - ピコピコンブ	8.006	0

4.4 考察

まず変換に失敗した原因について述べる。破損した理由としては、それぞれ破損した楽曲ファイルには共通した音を出した部分があった。階段状に音を出す部分である。この音を出す情報を16進数で表すと、「9n(ノートオンでnはトラックチャンネル)kk(音階)vv(音の強さ)ww(波長情報)9n ll vv ww 8n(ノートオフでnはトラックチャンネル)kk vv 8n ll vv」となり、本研究で提案するアルゴリズムで処理を行うと「9n kk vv ww ww 8n kk vv」となり、波長情報が余分に残ってしまうことで、破損してしまったと考えられる。

次に4.3で行った比較実験結果について述べる。図10.1、10.2、10.3、表2.1、2.2、2.3のファミコン音楽らしさの結果より、楽曲B、Cに関してはピコピンプよりファミコン音楽らしさを出せたということがわかる。しかし、RMEより劣っていることがわかる。これは、提案アルゴリズムは音色をGMで定義されている矩形波にしており、この音に比べるとまだファミコンらしさが足りないと考えられる。図10.4、10.5、10.6、表2.4、2.5、2.6の原曲に近いかの結果からは、どの楽曲に関してもピコピンプより評価は高く、表の結果からも有意水準0.05を満たしていないため有意性があることがわかる。RMEと比較を行うと僅かに提案アルゴリズムのほうが評価は高く、楽曲A、Bにおいて差があることがわかる。よって、既存技術に比べ、提案アルゴリズムは原曲からの変換だとわかるように出来たのではないかと考える。表2.2のerrorは提案アルゴリズムと分散和音処理の評価がすべて同じ点数であったためエラーが出た。

最後に分散和音について述べる。今回の実験で行った分散和音処理において、和音部分は8分音符単位で交互に音を鳴らすよう手動で処理を行った。8分音符を採用した理由としては16分音符以上になると曲のテンポによっては速くて、ドゥルドゥルという音が高速で鳴ってしまう点があったためである。アンケート調査の結果は、図10.1～10.6の図を見る限りではあまり差がないことが分かる。この理由はどちらの音色もGMで定義されている矩形波であるためではないかと考えられる。しかし、曲によって分散和音は合わないとの意見があった。3曲目の暗い曲調の楽曲には分散和音がマッチしているという意見があり、図10.3より、分散和音処理のほうがファミコン音楽らしいことがわかる。一方で1曲目の楽曲の分散和音は何かリズムがずれた感じがするという意見や、必要ないのではないかという意見があった。これらの意見、結果により、分散和音は処理を行う楽曲の曲調やテンポによって、必ずしも和音を分散和音にしなければならないということではないことがわかった。

第5章 おわりに

本研究では、既存楽曲からファミコン音楽への変換を目的にアルゴリズムを提案し、検討した。

また、提案アルゴリズムで変換した楽曲と既存技術で変換した楽曲を比較するためのアンケート調査を行い、既存技術の問題点については解決出来た。しかし、既存技術との比較アンケートを行うことで、MIDI で定義されている GM の矩形波ではファミコン音楽を表現することは難しいということが分かり、改善が必要であることが分かった。

今後の課題を以下に述べる。上記で述べた矩形波に関することと本研究が提案するアルゴリズムだと、MIDI に 10 個のトラックが内蔵されている場合、10 個の単音になってしまうため、楽曲を構成する上で必要であるトラックのみを抜き出し、変換を行う必要がある。また、ファミコン音楽で使用されている手法の分散和音を用いるために、楽曲の曲調やテンポを判別し、適切な分散和音処理をする必要もある。

謝辞

本論文を作成するにあたり、多くのご指導、ご助言を頂きました三好力教授に厚く御礼申し上げます。また、議論・実験に協力して下さった三好研究室の皆様や学友の皆様に心から感謝致します。

参考文献

- [1] 株式会社プニグラム ピコピコンブ
<https://www.punigram.co.jp/products/punibarom/index.html>
- [2] Freeware Place
<http://freewareplace.web.fc2.com/>
- [3] 高橋信之, コンプリート MIDI ブック, 株式会社リットーミュージック, 2005 年
- [4] 三上次郎, “和音進行のデータベース化と和声分析におけるその活用の一考察”
長崎大学教育学部紀要. 人文化学. vol.62,p.11-27; 2001 2001-03-25
<http://naosite.lb.nagasaki-u.ac.jp/dspace/handle/10069/5787>
- [5] 森本智志, “和音進行における音楽的期待の計算論的研究”奈良先端技術大学院大学情報科学研究科. NAIST-IS-DD761027
https://library.naist.jp/mylimedio/dllimedio/showpdf2.cgi/DLPDFR013166_P1-93
- [6] C 言語で MIDI(SMF)データを読んでみる
<http://torasukenote.blog120.fc2.com/blog-entry-104.html>
- [7] SMF の初歩
<http://www.geocities.co.jp/SiliconValley-SanJose/8132/>
- [8] SMF(Standard MIDI Files)の構造
<https://sites.google.com/site/yyagisite/material/smfspec>
- [9] 8bit ゲーム音楽の作り方。仕様を理解し、正しく打ち込みましょう。
<https://wingless-seraph.net/wp/2017/09/09/>
- [10] 全曲無料・フリー音楽素材/魔王魂
<https://maoudamashii.jokersounds.com/>
- [11] TAKABO SOFT/Domino
<http://takabosoft.com/domino>
- [12] T 検定参考 PDF
<http://mizumot.com/handbook/wp-content/uploads/b93393073fa7698302d96237f6f86c63.pdf>