ミス検出を備えた歌声音源制作支援システム

加藤 大悟
Daigo Kato
法政大学大学院情報科学部コンピュータ科学科
daigo.kato.2c@stu.hosei.ac.jp

abstract

When making a music, recording is needed many times. Then we also have to listen to them many times. We propose a Computer Aided Vocal Production System with Automatic Defect Detection. This system detects defect in the record and shows users it. addition of loudness and vocalization defect detection to the system is expected this system can reduce a number of listening times. We show that the detection found 74 % of defects from subjective evaluation. It can't detect defects of vocalization because of spectrum's high components is not absolutely absent. However, It shows that Prevention of rusting ears and decreasing the time.

1 はじめに

音楽を制作する際に、レコーディングという手法がある。その中で、歌を録音する場合がある。歌声の録音に必要なこととして、発声練習・録音・録音後の加工等が挙げられる。このような、歌声を制作する全ての過程のことを VocalProduction という。[1] 今回はその中でも、「録音」「コンピング」という過程に着目した。この過程において、基本的に一回だけの録音、俗に言う一発録り出来ない。レコーディングをしたことがある人ならば分かると思うが、一回のみの録音だと「高音が届いていない」「歌い出しが遅い」「低い音が聴こえない」といったミスが生じてしまう。

そこで何度も録音を繰り返し、そこから良い箇所だけをつなぎ合わせるコンピングという作業やミスを補正する必要がある。しかし、連続で歌い続けてしまうと時間がかかってしまう。さらに喉への負担も大きく、休みながら録音してしまう事を考慮すると、より時間がかかってしまう。喉の負担を考慮せずに、休みを入れずに録音したとしても喉の状態が回復せずに良い音源を録音することが出来なくなってしまう。これらのことから VocalProduction における録音の過程は一回で済むものではなく、ある程度の纏まった時間が必要であることが分かる。しかしながら、自分でレコーディングスタジオを持ってもいない限り、レコーディングスタジオを借りる必要があり、時間には制限がある。またコンピングをする場合において、録音した音源を何度も聴き比べる必要があり、この作業はとても時間を要してしまう。その上、何度も似たような音源を聴くことによって耳が「慣れて」しまい、何が上手いのかの判断がつきにくくなってしまう。その為、ミスのない音源の選択は容易だが、そこからどれが一番良いかを選択するのは困難という問題点が存在する。

Supervisor: Prof. Katunobu Itou

そこで、「歌声制作支援システム」を提案する。このシステムは、入力を音声とし、その音源に対してユーザがミスをしている箇所を検出し、どのように修正するかをユーザへ提示する。これによってコンピングで各区間ごとで良い音源を探す際に聴き比べる回数がが少なくなり、耳が「慣れる」事を防ぐことができ、真に良い音源を選択することも期待できる。また短時間でミスのない歌声音源を制作することが可能である。さらには、伴奏や他の作業への時間を増やすことによって、最終的により品質の高い楽曲を制作することが期待できる。

2 音楽制作

そもそも音楽制作におけるミスとはどういうものかを述べる。過去の研究として、エレキギターの演奏練習支援システムの為のミス検出 [2] を行ったものがある。しかし、人の声である歌と楽器であるエレキギターでは考慮すべきものが異なる。歌におけるミスは様々である。例としては 1 章で述べたものや「発声が悪い」等である。他にも細かく分類すると多くのミスがあるが、それを大まかに分類すると以下の四つに分けられる。また正解の基準として、プロの歌手が歌った音源を使用した。正解で誤検出が生じない閾値をそれぞれのミスにて設定した。

2.1 音程のミス

楽譜情報における音符の音高が音楽における「音程」である。歌声において音高の変化の箇所は即座に変わっているわけではなく、短時間ではあるが徐々にに変化している。また、歌声の音高は楽譜情報のように一切の誤差無く保たれていることは少ない。つまり、与えられた正解に対して、音価の初めと終わりにある切り替わり以外の箇所で正解にある程度近ければ問題ない。逆に、与えられた音高に対して、一定以上離れた音高を演奏した時を「音程のミス」とする。具体的には「歌唱者にとって高い音が出ない」・「一定の音を伸ばしている時に、段々高くなってしまう」・「短く、かつ比較的低い音が正確に出せない」等があげられる。

2.2 タイミングのミス

楽譜情報における音符を演奏する瞬間が音楽における「タイミング」である。正解に対してほぼ同じタイミングで歌い出せば問題ない。しかし、与えられた楽譜情報に対して、人間が知覚できるレベルで演奏がズレてしまうことを「タイミングのミス」とする。具体的には「歌い出しが遅れてしまう」・「息が続かず、十分な長さ歌うことができず、途切れてしまう」・「途中の歌詞が早口で噛んでしまう」等があげられる。

2.3 音量のミス

楽譜には絶対的な音量は定義されていない (ただし、表現として「クレッシェンド」等の音量変化を定義するものはある。)。音楽における音量は人間が実際に感じる音の大きさ、つまり聴

覚的な音の大きさである。基本的に周波数が高ければ高い程人間の耳には「やかましく」聴こえる (女性と男性の悲鳴を思い浮かべてくれると幸いである。)。そこで、実際の音源のパワーではなく、人間の聴覚的な指標を用いる。1 フレーズを通して音量変化が大きすぎない時は問題ない (ただし完全に同一な音量であるのは不自然であることを考慮)。しかし、メロディーの音量全体を通して、音量変化が局所的に大きく変化した場合をミスとする。主にあるミスとしては、「高音の音量が突出してしまい、うるさく聴こえる」・「低音の音量が小さすぎて十分に聴こえない」があげられる。

2.4 発声のミス

発声に関しても楽譜上では定義されない。大声や怒鳴り声を出した時、それらのような声を出した後に喉が痛くなる人が多い。これはそれらの声を出すために極度に喉を絞めていることが原因だと考えられる。つまり喉を人は「がなる」時、喉を絞めており、その状態等を発声が悪いといい、今回は「発声のミス」と定義する。それ以外の状態を発声が悪くない状態とする。

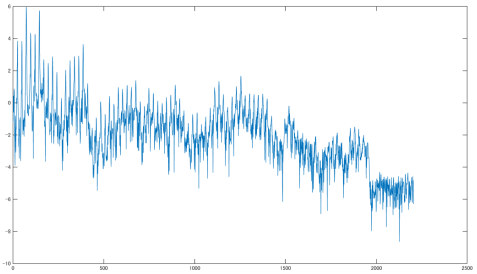

図 1. 許容される発声

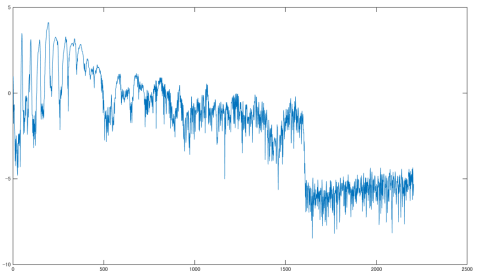

図 2. 乱れている発声

図 3, 図 4 は/a/を発声したもののスペクトルを表示したものである。図 1 は成分が弱いながらも、ピークが等間隔に出ている。しかし、図 4 は後半を見るとピークが等間隔に見えず、ある種の雑音のようなスペクトルになってしまっている。このようにスペクトルが「崩壊」してしまっている発声は不適切な発声と考えられる。予備実験により、5kHz 以下にまで崩壊がある場合、明らかに歌声として不適切な発声になることが分かっている。

2.5 実際の音楽制作

ここに、実際の音楽制作の手法である「コンピング」をシステム化した例を示す。今回、最終的に製作した音楽は一般的な J-POP 程度にミスのない音源である。その為、表現を除いて、ある程度楽譜に忠実に再現する必要がある。そこで問題になってくるのが録音時のミス・聴き直しである。ここでは「コンピング」を行うことを想定している。それにより、ミスの少ない音源を選択する為に、必ず聴き直しが発生する。これを解決する流れを以下に述べる。ユーザによる音声を入力とする。メロディー (音高情報、時間情報) を事前に入力してもらう。入力された情報を元に、許容できない程のミスを検出する。歌ったものに対してユーザが納得しない場合や、修正出来ないレベルのミス (声が裏返る、歌い出しが遅れた際に切り出す有声区間が長すぎる等) がある場合は録音をもう一度行う。どちらの場合にも属さない場合、ユーザに対してミスの箇所をを提示する。検出されたミスの中にユーザが意図的に行ったもの、つまり表現技法の一種であるものが含まれる可能性が考えられる。それらをミスとして検出しない為に、検出箇所に対してミスでないものをユーザに選択してもらう。ミスでないものが除かれたら、最後に検出されたミスをどのように修正すれば良いかをユーザへと提示する。またユーザが多数録音している場合、コンピングを提案する。その際にユーザがコンピングを望んだ時、各音源に対して最もミスが少ない音源を推薦する。さらに修正を施した場合の音声と聴き比べてユーザの好みの方を選択してもらう。

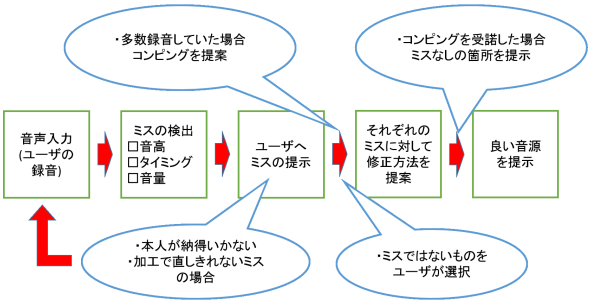

図 3. システムのイメージ図

3 ミスの自動検出

3 章にて述べてきた 4 つのミスを検出して提示することが出来れば、コンピングをする時も含めて、録音を聴き直す回数の減少が期待できると考えられる。また前提として、有声区間と無声区間の違いがはっきり区別でき、かつ音声が歪まないようにユーザによってマイクへの入力レベルを調整してもらう。それに伴って、無声区間に極力音が入らないように録音時に聴く伴奏が入力されないように、また歌わない箇所ではマイクから出来るだけ離れてもらう。

3.1 音程

2.1 節の定義に従う。今回、ミスとして判定される一定以上離れる場合を 1/4 音以上とした。1/4 音以上ズレると、音楽経験が少なからずある者であればその差異に気付くからである。これらのことから、事前にユーザによって入力された音高情報を元に検出。音声入力された歌声の音高を自己相関法によって推定 [3][4][5]。また、1 フレームを 960 点数とし、フレームシフトの点数を 48 点 (約 1ms) とした。(ビブラートによる推定誤差の防止の為) 推定結果には一次元のメディアンフィルターを掛けることにより、誤り値を減らす。推定結果と本来の音高を Hz から cent に変換。Hz で比較してしまうと、低周波でのミスが検出されてにくいため、音階間の値が一定になる cent を使用する。変換したのち、二つを比較し、設定された閾値 (半音=50cent) 以上にズレていた場合にミスとして検出する。また、その音高がある区間の推定結果の平均値を求め、その値も検出

に使用する。検出されたミスの本来の音高からのズレを Hz で示した上で、それを修正できるかどうかを判定する。

図 4 は実際に音程のミスを検出していた図である。赤線が正しい音高、黒線が自己相関法によって推定された音高、マゼンタ線がミスとして検出された箇所、青線が各区間 (音価) の平均値、緑線が平均値で見た時のミスである。最初の二つの区間で音が届いていない所がミスとして検出されている。また、一番右の伸ばしで段々低くなっている箇所も、平均値を用いないことで検出されていることが分かる。

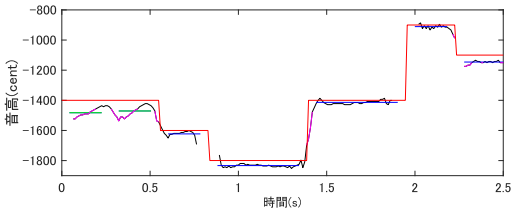

図 4. 音程のミスの検出のイメージ

3.2 タイミング

2.2 節の定義に従う。今回、ミスとして判定される歌い出しが一定以上早く、又は遅く歌い出してしまった時のズレ時間を 0.1ms 以上とした。筆者が実際に音源を制作する際に、ミスだと感じられる時間である。これらのことから、スペクトル包絡 (14 次元のメル周波数ケプストラム係数; 以降 mfcc とする) を計測し、差分をとることによって歌い出しのタイミングを計測する。楽曲において、基本的に音符一つに対して一つの音節を当てる。(例外はあり) つまり、発音が変わる箇所が音符の変わり目である。そこで、スペクトル包絡は発音に深く関与しているので、スペクトル包絡の違いで検出することが可能である。修正方法の提案をする際には、前述で検出された区間及び、本来歌われるべき区間をユーザに提示。

図 5 は実際にタイミングのミスを検出した図である。一行目は、青線が音声データの波形、赤線が楽譜情報による正解のタイミング、黒線が実際に検出されたタイミングである。2 行目は、青線がフレームごとのスペクトル包絡の違い (Δ mfcc)、赤線が検出されたタイミングのスペクトル包絡の違いを数値化して表示したものである。上下で比較すると、左から 2 番目のタイミングで、Δ mfcc の値が大きいところが本来のタイミングからズレていることが分かる。つまり、入力データでの音韻変化が本来の情報からズレていることが分かる。

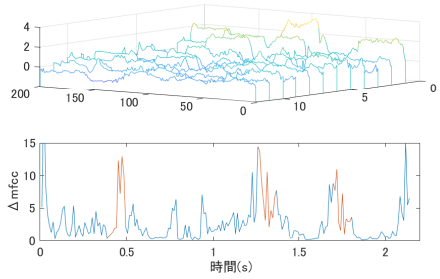

図 5. タイミングのミスの検出

3.3 音量

2.3 節の定義に従う。しかし、フレーズ内で局所的に大きな音量変化 (大きい、小さい) してしまった場合はミスである。また、音の大きさを測るものとして、ラウドネスというものが存在する [6][7][8]。ラウドネスは ISO532 で規格化されており、聴覚特性を考慮して計算される感覚量である。実際に聴こえる音量なので、短時間パワーよりも有効であると考えられる。今回ミスとして判定される変化量は 4LKFS である。こちらもタイミングと同じく、実際に音源制作する際にミスと感じられた変化量となっている。入力された歌声から、事前入力情報を用いてメロディーの一音ごとの区間を切り出す。それぞれの区間に対してラウドネスを計測し、前後の区間で閾値以上に異なる場合にミスと検出する。また、区間の周波数に応じてある程度値が異なることも許容することとし、その事を考慮して修正を提案する。

図 6 は実際に音量のミスを検出しているものである。線はそれぞれ、黒線が実際に計測されたラウドネス。青線が区間ごとのラウドネスの平均値、マゼンタ線がフレーズ全体のラウドネスの平均値、赤線がミスとして検出した箇所。赤線に着目すると、他の区間に比べてマゼンタ線からの距離が遠い。一つ目は小さすぎるのでフレーズに埋もれてしまい、二つ目は逆に大きすぎて浮いてしまっていることが分かる。

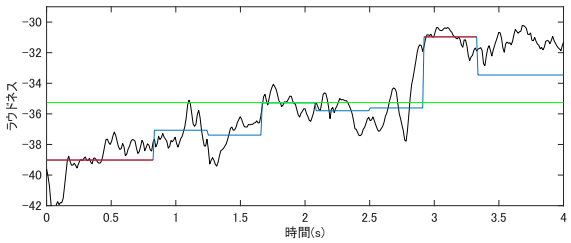

図 6. 音量のミスの検出

3.4 発声

2.4 章でも述べた通り、発声にはスペクトルが深く関係している。そこで今回はスペクトル重心を「崩壊度」と定義して、発声の状態を検出する。スペクトル重心が。予備実験により、スペクトル重心の値が 40 以下だと発声が悪いと決定した。各フレームにおいて高速フーリエ変換を行い、スペクトルを求める。求められたスペクトルからスペクトル重心を計測。そのスペクトル重心と予備実験により決定した閾値との差分が大きかった時、ミスとして検出する。

図 7 は実際に発声のミスを検出している図である。青点がスペクトル重心の値、赤線が発声が悪いと判断された値である。実際に赤線で表示されている区間はとても強くがなっている。

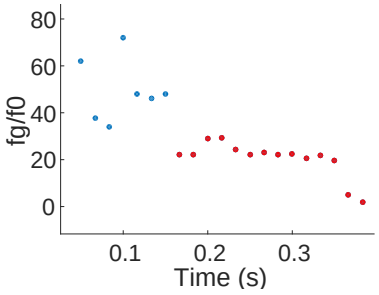

図 7. 発声のミスの検出

3.5 コンピング

音楽制作には様々な手法がある。その中でも本研究は特に音源を聴く回数が多い「コンピング」に着目している。コンピン

グとは、修正を使わずに多数の録音された音源から良いところをつなぎ合わせることでミスがない音源を作ることである。この手法はユーザが一定以上の回数録音をやり直した場合、またはミスの修正なしで歌を完成させたい人に適している。この機能を実装するために、まずミスの種類や度合によって各ミスフレームに対してスコア付けを行う。今回はミスの度合が大きい程スコアが高くなるので、低い順に順位を付けていく。これを各セグメント (2 音が入る程度の長さ) 毎に行っていき、それぞれのセグメントでの 3 位までの音源をユーザへ提示する。

4 評価

4.1 実験

以上で実装したミス検出機能の性能評価実験を行った。まずはミス検出機能の性能評価実験について述べる。対象者は歌唱経験がある 20 代の若者 4 人である。被験者に曲調に極端な偏りが出ないように、「テンポが遅い・速い」「音程が低い・高い」を組み合わせた計 4 種類の曲が均一な数になるよう歌う曲を選んでもらった。被験者に選んだ曲のワンフレーズを歌ってもらい、被験者と筆者でミスと思われる箇所とミスの種類を決定。決定されたミスに対して、検出された数を検出率とした。

またシステムの性能評価実験も行った。今回はコンピングにおける聴き比べによる問題に着目している為、システムもコンピングを使用する前提で構築した。対象者は歌唱経験がある 20 代の若者 4 人である。被験者に好みの 3 曲のワンフレーズを 5 回歌ってもらい、自力でコンピングを行ってもらう。その際に選択された音源を正解とする。その後システムでコンピングを行い、それぞれのミス度合いをスコア付けする。それぞれのスコアから順位を決め、正解音源の順位を合計したものを評価尺度として用いる。

4.2 結果

表 1 はミス検出機能の性能評価実験の結果である。表 1 より、タイミングと発声のミスの検出率が少々低くなってしまった。全体としては 0.74 のミス検出率があることが分かった。

まず、音量のミスで「伸ばしの途中で局所的に小さくなる」場合が検出できなかった。これはメディアンフィルターでも対処しきれない外れ値があったことが原因だと考えられる。これは各区間内でも局所的な変化の検出を行うことで解決できると考えられる。

次に、発声のミスの検出が著しく低くなってしまった原因として、「発声の悪いスペクトルは、高周波成分にピークが出ていなくても成分自体はある」ということがあげられる。これはピークが等間隔でない性質を利用して、ピーク間隔を計測して、その値も用いることで解決できると考えられる。

表 1. ミス検出の性能

音程のミス	0.87
タイミングのミス	0.69
音量のミス	0.74
発声のミス	0.65
全体	0.74

表 2 はシステムの性能評価実験の結果である。表 2 より、7 割の音源は聴き直しをしなくても選択されることが分かった。従って、自力でコンピングを行う時と比べ、明らかに少ない回数の聴き直しで音源を制作することが出来る。これは先に示したミス検出及び、ミスのスコア付による選択が影響していると考えられる。しかし、今回は 5 回の録音でコンピングを行っており、録音回数が 7 回を超えた場合は、聴き直しの回数が多すぎてしまうことが考えられる。

表 2. システムの性能

被験者 1	11 個/15 個
被験者 2	10 個/15 個
1 と 2 の合計	21 個/30 個

5 おわりに

本研究では、音楽制作における最も回数が必要な「録音」と「コンピング」という過程に着目し、繰り返しの録音による聴き直しを減らす為のシステムを提案した。従来のカラオケ採点システムに加えて音量と発声のミスも検出することによって自動ミス検出機能を作成した。作成されたミス検出機能の評価実験を行い、0.74 の検出率の性能を持つことが分かった。また、スコア付けを用いたコンピングのシステムを構築した。構築されたシステムの性能評価実験により、7 割の音源は聴き直しをしなくても選択されることが分かった。これらにより、音楽制作の手法の一つの「コンピング」での「耳の慣れ」という問題点が解決される可能性を示した。

今後の課題として、アルゴリズムの改善でより検出率を高められる可能性がある。また、現在想定されているシステムは DTM ソフトウェア外での動きを想定しているため、必ず二つのソフトウェアを動かすという不便さが生じてしまう。今後、既存の DTM 自体にこのシステムを組み込む、または事前に組み込んだ新たな DTM ソフトウェアを作成することで解決できると考えられる。

参考文献

[1] Matthew Weiss, The Complete Guide to Vocal Production, 2018.July.30,https://theproaudiofiles.com/vocal-production/

[2] 下尾 波輝, 矢谷 浩司, エレキギター演奏におけるミスの自動検出, 2018 情処全文, 1 号, pp131-132, 2018

[3] 竹内 英世, 保黒 政大, 梅崎 太造, カラオケ採点用の高分解能ピッチ抽出法, 電学論 C, 129 巻, 10 号, pp1889-1901, 2009

[4] 竹内 英世, 保黒 政大, 梅崎 太造, 人の主観評価に近いカラオケ採点法, 電学論 C, 130 巻, 6 号, pp1042-1053, 2010

[5] 鈴木 久嬉, ピッチ抽出
の今昔, 音響誌, 56 巻, 2 号, pp121-128, 2000

[6] 曽根 敏夫, 鈴木 陽一, ラウドネス, 音響誌, 44 巻, 10 号, pp1042-1053, 1988

[7] 栗原 信義, 高橋 信夫, ラウドネスレベルメータの開発, Vol.55, no.3, pp364-371, March, 2002

[8] 黒住 幸一, 新しいラウドネス曲線に基づいたラウドネスレベルメータの開発, 日本音響学会講演論文集, pp495-496, 2004