

19 事例研究: オートチューンの作成

19.1 概要

オートチューンと呼ばれるエフェクターがある。このエフェクターは、入力の歌声の基本周波数を近い音階の周波数に修正するものである。

このように入力を分析して、効果を変化させるエフェクターは適応型のエフェクターと呼ばれる。そのような効果を実現するための手順は以下の通りであろう。(実際のオートチューンの処理手順とは異なる可能性が高いことに注意。)

- 入力の基本周波数を短時間 (16 分音符以下) のフレームで刻一刻求める。
- 各フレームの基本周波数の近傍の音階を推定する。
- 入力音声の基本周波数が推定された周波数になるように加工する。

ここでは、入力の音色を極端に変化させるような強い効果のオートチューンを想定する。以下、この処理を MATLAB で実装する方法をステップごとに説明する。

19.2 基本周波数の推定

このステップでは、フレームごとに基本周波数を求める。これには、ソースコード 32 が利用できる。

基本周波数は簡便な推定手法の場合、推定誤りが多発する。しかし、今回の処理では、処理対象のメロディーがわかるので、推定すべき基本周波数の範囲が制限できる。この制限によって、ある程度推定誤りを減らせる。

また、子音や無声音、無音の部分は、基本周波数が上手く推定できない場合がある。しかし、このような部分は、そもそも音程が明確でないため、その部分については音程を変化させる必要がない。したがって、そのような部分は、nan 値にしておいて、処理対象としないようにする。

19.3 音階推定

音階に対応する周波数の間隔は一定ではない。例えば、平均律では、ピアノの中央の C 音は 262Hz, D 音は 294Hz, E 音は 330Hz である。

音階と周波数の関係でよく知られているのは、1 オクターブ上昇すると周波数は 2 倍になるということである。この 1 オクターブの間には、12 個の半音が含まれる。1 オクターブの変化が 2 倍というふうに比率として表わされる。そこで、半音ごとの変化も比率で表現し、比率が均等になるように周波数を定める方法が考えられる。このような決め方を平均律とよぶ。

このステップでは、入力の各フレームで推定された基本周波数に最も近い周波数を推定しなければならない。このような手順を考えるためには、まず、周波数の「近さ」を数値的に定義しなければならない。

方法はいろいろありえるが、処理を簡単にするために、まず、半音の音階を整数であらわすことを考える。

例えば、中央の A 音 (440Hz) は A4 と呼ばれることがあるが、この A4 を 0 とする。このときに、全音上の B4 を 2、その半音上の C5 を 3 というように決める。これらを基準に、例えば、入力の基本周波数を実数で表現する方法を考えれば、四捨五入で最も近い音階の基本周波数が推定できる。

19.4 ピッチシフト

入力音声の高さを変化させる処理をピッチシフトと呼ぶ。ピッチシフトの方法にはいろいろある。今回の処理で要求される点を以下にあげる。

- 生成された音声では基本周波数は一定で、指定した周波数にならなければならない。

- 生成された音声では基本周波数のゆらぎは必要ない。

この条件を満たし、なるべく簡単な処理でできる方法としてここでは、wavetable 法による合成手法を採用する。(wavetable 法に関して詳しく知りたい場合は、<http://www.slp.k.hosei.ac.jp/~itou/lecture/2010/ProjectB> を参照のこと)

wavetable 法は、合成音声の素材となる音声データを 1 波長分格納しておき、そのデータの周期を変化させることで、様々な音階の音を生成する。データの周期を簡単に変換する方法としては、resample という関数の利用があげられる。

```
>> [y,fs]=wavread('a-.wav');  
>> y2=resample(y,1,2);  
>> soundsc(y2,fs)
```

このような処理で簡単に高さを 2 倍にすることができる。ただ、この手法では、長さが短くなってしまっているので、それに対処しなければならない。

19.5 関数 auto_tune

ここまでの処理を組み合わせ、auto_tune 関数を作成する。これまで述べてきた手順では、フレームごとの処理を説明した。この処理を組み合わせ、auto_tune 関数を作成するためには、フレームをどの程度ずらすのか、どのように組み合わせるのかを決めなければならない。

最も簡単な方法としては、ソースコード 17 で紹介したハン窓など両側が減衰する窓をかけ、フレーム幅の半分ずつずらす方法がある。

ここまでわかれば、MATLAB で実装できるはずである。