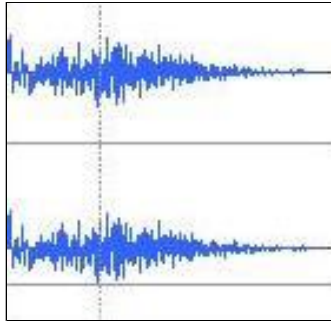


INCLUDEの第一回
公開日:04/4/30

第一回:音声圧縮に挑戦!



こんにちは、皆さん。いよいよ始まった当コーナー**INCLUDE**。INCLUDEは主に、話題になっている工学技術に、不定期に取り組んでいくページです。実際に記事で作成したプログラミングのプロジェクトもダウンロードできるので、実際に触ってみてください。

さて、早速本題に入りましょう。今回は「音声圧縮」というものに取り組みます。

1: 音声圧縮とは何か？

音声圧縮とは、音楽CDなどの**デジタル**の音楽を、何とかしてでも**データサイズを減らす**ことで…と、かなり適当にいつちやいましたねえ。ううんと…何とかするんですよ(笑)!

圧縮という言葉の、本来の意味は**つぶすこと**。コンピュータで言われる**圧縮ファイル**なんていうのも、データをつぶすことですよ？でも…どうやって？

データなのに手で「ぐしゃあっ!」とつぶしたりはできませんよね？皆さんがファイルを圧縮したりするときは**圧縮ソフト**といわれるソフトに、ファイルを放り込んだりするわけです。そうすると、変な青いバーが、ぐによーっと右に伸びていって、「終わりましたよん」なんていって、デスクトップなんかには、ぽん!と、圧縮できたファイルを作ってくれますよね。

音声圧縮、すなわち、何とかして音声データのサイズを減らす方法も、同様に圧縮ソフトにファイルを入れれば実現できるわけです。が…この方法ではいささか問題があるんですよ。

音声データのファイルも、文書とかのファイルと全く同じような手続きで圧縮されるわけなんです。そのとき、文書などの単純なファイルは、圧縮ソフトで十分小さくできるんです。なんせ中身が単純だから。

でも、音声ファイルは、中身が決まっています。

たとえば、文書だったら、ひらがな・カタカナは50種類ちょっとだし、漢字だってせいぜい数千文字しか使いませんよね。だから、できあがる文書って言うのは、全体を見ると、違う文書でも随所に”こんにちは”が出てきたり”春眠暁は…云々”などの似たような内容が…言い換えると似た

ような0101001010・・・が並んだりするわけです。

ところが！音声は違います。1秒たったら「あっ！」って叫ばないといけない、だとか英語しかしゃべっちゃだめだとか、そんなうるさい決まり事はありませんよね。でも文書には50音以外の平仮名なんて出てきません。それは決まりごとです。

人の声を見ても、自分と全く同じ人など2人といません。同じ言葉を読させても、人によって聞きづらかったり、甲高い声だったりかすれてたりするのだから、データが複雑なのもうなずけると思います。

当然、データの中身もテンでばらばら。文書などを圧縮するソフトって言うのは、決まり事が少ない程、うまく圧縮できなくなるんです。言い換えると、データのサイズも減りにくい。圧縮したのに10MBの音声ファイルが40キロバイトしか減らなかった、なんてことはよく聞くことです。

で、今回挑戦するのは、そんな音声の圧縮に強い方法。と話を進める前に、もう1つ言っておかないといけないことがあります。

圧縮には実は2つ、区別があるんです。圧縮の方法ではなく、**圧縮しても中身が変わらないかどうか**という区別です。

一つには、**可逆圧縮**というもの、もう一つは**非可逆圧縮**と言うものです。

前者は、圧縮後のデータが**絶対に元のデータと同じ**です。圧縮したあと、**解凍**をするわけですが、解凍したデータは元のデータと全く同じです。

あ、そうそう。圧縮データを取り出す作業・・・つまり、圧縮したデータをもとのデータにもどす事を解凍と言います。組みかえられたデータの復元作業というわけですね。ちなみに、この圧縮解凍という作業の呼び方はいろいろあります。例えば、動画データの圧縮解凍には、俗にエンコード・デコードという言葉が使われています。

で、後者の非可逆圧縮。こちらは、圧縮後のデータが**絶対元に戻らない**・・・つまり、解凍したら元のデータと**中身が違ってしまふ**。解凍して元の文書などと比べると中身が変わっちゃう。

後者は元に戻らないから役に立たないように思われましたか？ところが、一概にそうとは言えないんですよ。

実は後者の場合、今回取り組む**音声圧縮**などに絶大な効果を発揮します。上手あいやりかたを用いると、音声データなどを20分の一ぐらいにできるんですよ。一方、可逆圧縮を選ぶと、どうやっても2分の一ぐらいが限度だったりします。

ただし！非可逆圧縮は、残念ながら**元には戻りません！**

だから、元に戻らなくてもいいなら非可逆圧縮を大いに活用すればいいわけですよ。

だってねえ、**あいまいでもいいじゃない？！** 音声なんてね、遠くから聞くだけで小声が聞き取りにくくなったりするんですよ？後ろで話してる人のざわつきとか、ものすごく高い音程とか、身近な曲とかには結構、人にとってどうでもいい部分があるわけ。

要は、適材適所なんですね。文書など、元の内容と後の内容の不一致が困るデータには絶対元に戻る保障のある可逆圧縮を使えばいいわけだし、見かけの形など、**曖昧でも良いもの**、言い換えると無駄の多いもの・・・例えば、写真データみたいに、見た目がある程度きれいなら細部はある程度元に戻らなくても良いものなどには、非可逆圧縮を使えばいい。

今回の音声圧縮は、人間にとって聞こえない、聞けなくてもいい部分・・・つまり、無駄が多いデ

一タの圧縮になるわけだから、後者の方法をとるわけです。

例えば、今もっとも有名な音声圧縮「MP3」、MICRSOFTが推し進める「WMA」、MDの「ATRAC」など、圧縮方法にはいろいろな種類があります。

どれも基本は、人間には聞きとれない部分を切り取ってしまう、と言うものです。超音波と呼ばれる音を切り取ってしまうわけです。

今回は、こういう有名な音声圧縮には特許の問題があって、かつてに技術公開をするのもよろしくないという問題もありましたし、とりあえずは導入として、オリジナルの圧縮方法を紹介する事にしました。

圧縮っていうのは、**数学が命**です。特に、非可逆圧縮は、数学的な理論を組み合わせで無駄に見える部分を切り取ること・・・つまり**人間の曖昧さ**を考慮しデータサイズ削減に大いに利用すること、また圧縮・解凍のスピードが目的にあっているかも考えなくてはならないことなど、非常に注文の多い作業です。これに付随して高性能な圧縮アルゴリズムほど、中身が複雑になる傾向があります。

今回は導入ですから、まずは簡単に、数学的な部分として、**中央値**と言うものを、圧縮に利用する事に見てみましょう。

2:実際に圧縮をする

ではでは。圧縮に入る前にまず、中央値の説明をしましょう。

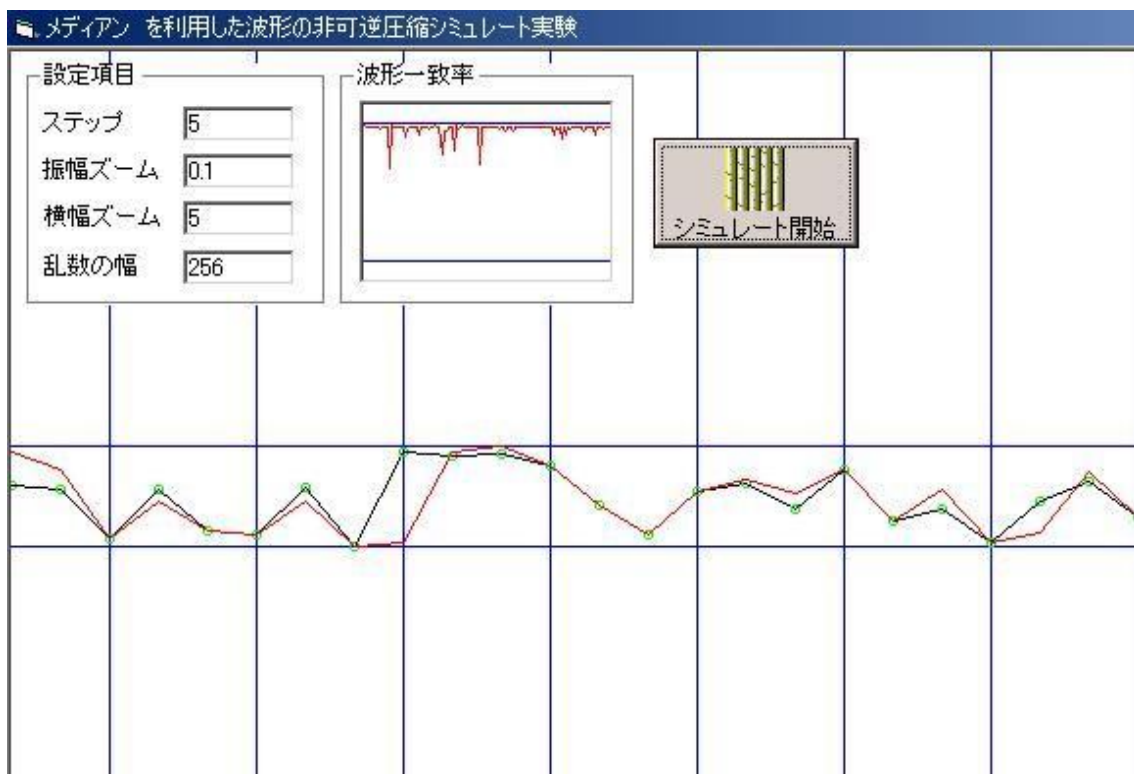
中央値とは、ある数字の集まりを小さい順に並べていった時、ど真ん中の大きさを持つ値を言います。

1 2 4 8 12 14 15 18 20

例えば、上の数字の場合、**12**がそれにあたります。右からの個数と左からの個数が一緒ですよ？それが中央値です。もし全体の個数が偶数個なら、両端から真ん中の2つの数字の平均が中央値になります。

さて、問題はこれからです。この中央値、「メディアン」といつたりするわけですけど、どうすれば音声圧縮に役立つのか？それを考えてみましょう。

みなさん、下図をご覧ください。これは、音声の波形をシミュレートしたものです。



図のように、まず音声データを3個の点ごとに区切ってみます。ちなみに、図自体は30個ごとに縮小して表示しています。そして、区切った3つのデータ(数字)の中で、中央値を探します。

そして、3つの中の中央値だけを左から順に記録していくわけです。これだけで、データ量が三分の一になりますよね？すなわち、データが3つで24バイトなんですが、この中で記録するのは一個の中央値…つまり8バイトだけになります。

ところで、この中央値のかわりに、3つの平均値を記録したらどうでしょう？平均値をとっても全体の大まかな波形は記録できます。いや、なにも平均を取らなくても3つごとに2つを間引いてもいいのです。

しかし、3つのデータ…つまり数字に、大小のばらつきが多い場合、平均をとると音の歪む場所が出てきたりしますし、音の大きくない所まで音が大きくなったりして、スピーカから聞こえる音がうるさすぎてしまう場合も出てきます。

平均をとらない場合…間引く場合は、これらのことがおきませんが、逆に点を打つ機会が減るわけです。もともと周波数が決まってる音を出したのに、間引いてしまっただけでは周波数が下がってしまいます。間引く周波数のことをサンプリングレートといいます。これを下げることは、ビデオの3倍モードにしたようなものです。

中央値のいいところは、**極端に周りより大きかったり、小さすぎるものが選ばれない**ので、平均をとって記録するよりは、元データとの誤差の発生が減る事です。またある手法でもっと音を浴することもできます。

理由:平均をとるデータにばらつきが無い時はメディアンも平均値に近づきます。そうでない時はばらつきがあるのでメディアンが有効です。どちらの場合も、平均値よりメディアンが誤差に対して有効なのです。音声圧縮は誤差の少ないほうがいい音に聞こえますよね？ですから、誤差が少ない事はいいことなのです。

ここで、中央値以外に中央値より大きい数字を、もうひとつ余分に記録すると、残った数は必然的に一番小さい値と決まります。中央値と一番大きい値との差を、中央値から引くともう一つの値はもとの値と近い値になります。つまり、圧縮後に**大方**推測できるわけです。ただどね、圧

縮データからの最小値の推測時には誤差の発生が起こりますよ、さすがに・・・。

3: 音声を圧縮して聞いてみた

さて、実際シミュレートしてみた。そしたらね・・・全然だめでしたよ。音量を上げすぎてスピーカが音割れしているような感じです。データは間引かれるので、画像の解像度が極端に低いときに気持ち悪くなってくるのと似ています。また、ちょっとバグってるのかな・・・データの膨張が起こります。

シミュレート図の小さい波形が、青い下の線を越えると、音割れになってしまいます。ちなみに図の小さい波形は、実データとの誤差率です。下の線が誤差200パーセント線です。上の赤線に近いほど、誤差は小さくなります。

超えないはずなのに・・・

4: 記事を書き終えて・・・

今回の方法は、あまり音声圧縮には向かなかったようです。しかし、今後も音声圧縮にチャレンジしていきますので、またご覧ください。

ダウンロード

さて、記事に使ったプロジェクトを公開します。

[VisualBasic6.0プロジェクト・・・圧縮波形シュミレーション](#)

メール等の受付

当サイトの管理人は、MORIOです。

質問やご要望、ご感想、苦情などは、メールで受け付けております。以下のアドレス宛に送ってくださいませ。

master@morik.net

form 2006/1/9