時間軸シャフリングによる周波数シフト法*

5Z－8

山内祥暢†　大村智宏‡　千種康民§
東京工科大学¶

1　まえがき

音楽情報処理においてFFTを使用する場合，低い周波数の識別に必要な周波数解像度よりFFTの周波数解像度が劣る場合がある．その場合，FFTの前に音楽信号をn倍して必要な解像度を満足させる必要がある．

既にデジタイズされた入力信号の周波数をn倍したい場合，最も単純な方法は，一定間隔おきにデータを間引き，空いた部分を何らかの方法で埋めることである．この研究は，その「空いた部分の埋め方」について3つの方法を比較したものである．

2　周波数のn倍法

それでは，3つのn倍法について説明する．これから例として示す波形は計算して作成したものではないので多少正確さに欠けるが，実際には原信号に対するサンプリングレートを多くとるので理想のn倍が可能である．

2.1　間引いた後繰り返す方法

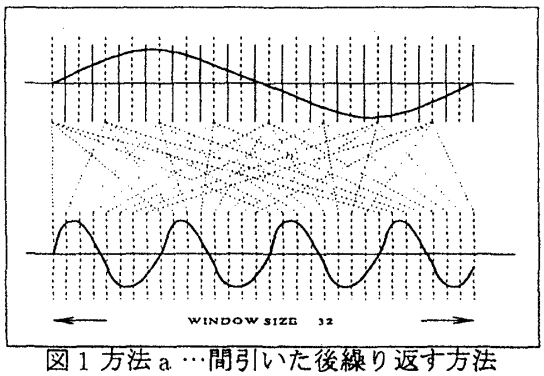

図1 方法a …間引いた後繰り返す方法

4倍を例に説明すると，4つに3つを間引くと4倍になるわけだが，そのままでは処理単位ウインドウ内のデータの数が1/4になってしまう．そこで，間引いた後のものをウインドウ内に4つ並べることで埋めてやる方法である．

*Frequency Multiplier using Time-Domain-Shuffling.
†Yoshinobu YAMAUCHI
‡Tomohiro OHMURA
§Yasutami CHIGUSA (E-mail chigusa@cc.teu.ac.jp)
¶Tokyo Engineering University, Japan

2.2　並べかえる方法

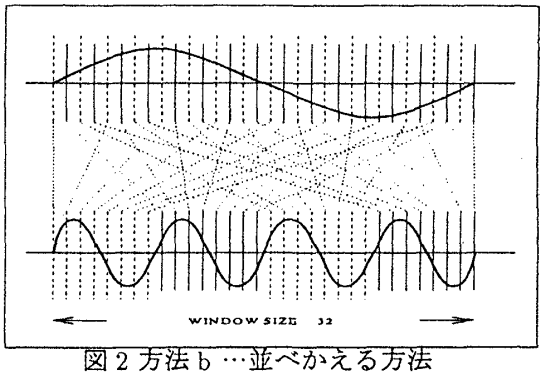

図2 方法b …並べかえる方法

同じく4倍を例に説明すると，処理単位ウインドウ内で，4で割り切れるもの，1余るもの，2余るもの，3余るものに分け，それを順に並べることで4倍化する方法である．

2.3　折り返しながら並べかえる方法

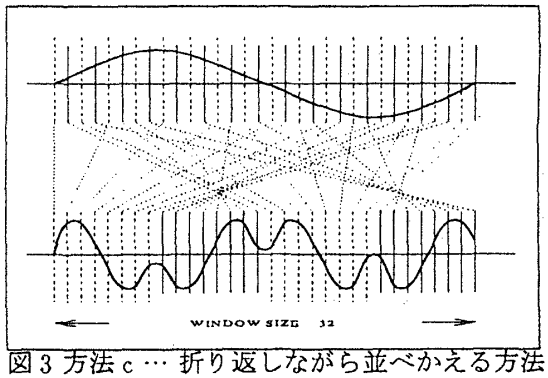

図3 方法c … 折り返しながら並べかえる方法

これは少々複雑になるが，処理単位ウインドウ内で，4で割り切れるもの，1余るもの，2余るもの，3余るものに分け，これらを昇順，降順，昇順，降順に並べることで4倍する．

3　3つの方法の比較

3.1　正弦波での比較

これら3つの方法のうち，どれが最も良いかを調べるために，FFTを用いて次のような実験を行なった．

まず，入力波の周波数を F[Hz]，サンプリングレートを 8192[Hz]，並べかえ処理のウインドウサイズを 128 とする．その F をそれぞれの方法で 4 倍した後，ウインドウの両端の不連続な部分を取り除くためにハニング窓を掛ける．[1]

それを，FFT に掛けた時に現われるスペクトルを X_f とすると，式 (1) の値が $4 \times F$ に近いほど正確に 4 倍されているといえる．

$$\frac{\sum_{f=1}^{4096}(f \times X_f)}{\sum_{f=1}^{4096} X_f} \Rightarrow 4 \times F\,? \qquad \cdots\cdots(1)$$

このことを利用して，3 つの方法を 200〜600[Hz] を 50[Hz] 刻みで比較した．

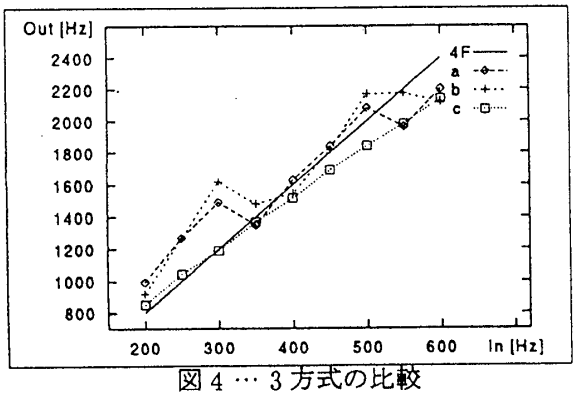

図 4 … 3 方式の比較

4F → 入力 F の 4 倍の周波数（理想値）．
a → 繰り返す方法．
b → 並べかえる方法．
c → 工夫して並べかえる方法．

直線 4F に近いものほど正しく 4 倍されているわけである．もちろんウインドウ単位で FFT に掛けることによる誤差は 0 ではないが，縦軸のレンジから考えればほとんど無視できる大きさであった．

それぞれ，ほぼ 4 倍されてるが，方法 a,b はとんでもない値になる部分がある．それに対して方法 c は非常に緩やかで直線に近い．

3.2 複雑な波形での比較

次に、正弦波より複雑な波形で比較する。

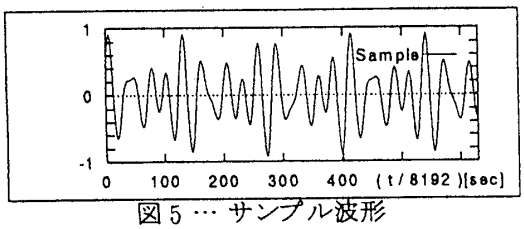

図 5 … サンプル波形

[1] ウインドウの端のいずれかが 0 でない場合，そのことによる影響が FFT の結果に混入する．ウインドウ内での操作による影響を比べるにはこれを取り除く必要がある．

この波形をそのまま FFT に掛けた時の式 (1) の値は 260 なので，4 倍した時の周波数の理想値は 1040 となる．

方法（ウインドウサイズ 128）	結果
間引いた後繰り返してうめる方法	1157.00
並べかえる方法	1053.28
折り返しながら並べかえる方法	1077.92

方法（ウインドウサイズ 64）	結果
間引いた後繰り返してうめる方法	1209.81
並べかえる方法	1372.50
折り返しながら並べかえる方法	1009.43

方法（ウインドウサイズ 32）	結果
間引いた後繰り返してうめる方法	1085.30
並べかえる方法	1231.71
折り返しながら並べかえる方法	1037.85

表 1 … ウインドウサイズによる精度

複雑な波形の場合はどの方法でも n 倍が難しいようである．しかしこの場合も，比較的安定して良い結果を出すのは，折り返しながら並べかえる方法 c であった．

4 むすび

方法 a,b はともかく，方法 c は入力が正弦波なら広い周波数帯域で安定した結果を出し，図 5 のような波形であっても，他の方法より良い結果であった．

最初にも述べたが，ここで挙げた方法は原信号に対するサンプリングレートを多くとるほど波形を正しく n 倍する．最適なサンプリングレート，最適なウインドウサイズを入力に対して適用し，ある程度の補正を行なうことのできるような用途に対しては有効な方法であると思われる．

参考文献

[1] 小畑秀文，“CAI ディジタル信号処理”，コロナ社

[2] 小島啓彰，高嶋　貢，“楽音ピッチ同定のためのアナログ前処理”，信学論，A Vol.J78-A No.11 pp.1497-1500 (1995)