

平成 12 年度

学士学位論文

ウェーブレット変換を用いた音声符号化

Speech coding using wavelet transform

1010370 今井 大

指導教員 福本 昌弘

2001 年 2 月 5 日

高知工科大学 情報システム工学科

要 旨

ウェーブレット変換を用いた音声符号化

今井 大

現在のインターネットや携帯電話で行われている音声符号化の目的のひとつとして、音声情報の冗長な部分を除去することによる情報の圧縮が挙げられる。情報を圧縮することで、限られた帯域や伝送速度で多くの情報を伝送することができる [1]。現在、最も使用されている情報圧縮の技術は JPEG などに用いられている離散コサイン変換 (DCT) である。その他に、離散フーリエ変換 (DFT)，ウェーブレット変換などがある。本研究では、時間周波数解析方法であるウェーブレット変換を用いた音声符号化の方式を示し、その特性を評価する。

キーワード ウェーブレット変換，離散コサイン変換 (DCT)，聴覚特性，マスキング効果

Abstract

Speech coding using wavelet transform

Dai Imai

As one of the purposes of the speech coding has been done with the present the Internet and the cellular phone. The compression of the information by getting rid of the lengthy part of the voice information it is given. By compressing information, much information can be transmitted in the limited obi stage, and the transmission speed. At present, the technology of the information compression being used most is the discrete cosine transform (DCT) being used for JPEG and so on. There are a discrete Fourier transform (DFT), wavelet transform, and so on additionally. The form of the speech coding which wavelet transform that time frequency was how to analyze it was used for is shown, and that character is evaluated by this research.

key words wavelet transform, discrete cosine transform (DCT), Hearing characteristic, masking effect

目次

第 1 章	序論	1
1.1	本研究の背景と目的	1
1.2	本論文の概要	2
第 2 章	音声情報圧縮	3
2.1	まえがき	3
2.2	音声	4
2.3	音の性質	5
2.4	聴覚特性	7
2.5	標本化(サンプリング)	8
2.6	量子化	9
2.7	エントロピー符号化	10
2.8	直交変換符号化	10
2.8.1	離散フーリエ変換(DFT)	11
2.8.2	離散コサイン変換(DCT)	11
第 3 章	ウェーブレット変換	13
3.1	まえがき	13
3.2	ウェーブレット変換	14
3.2.1	ウェーブレット解析	14
3.2.2	ウェーブレット変換	14
3.2.3	連続ウェーブレット変換	15
3.3	離散ウェーブレット変換	16
3.3.1	ハールウェーブレット	17

3.3.2	信号の分解と再構成	18
3.3.3	ドベシイの直交ウェーブレット	18
第4章	ウェーブレット変換を用いた音声符号化	20
4.1	まえがき	20
4.2	離散ウェーブレット変換と逆変換	20
4.3	ウェーブレット変換した信号の音声符号化	20
第5章	結論	23
5.1	離散ウェーブレット変換と離散コサイン変換についての検討	23
5.2	結論	23
謝辞		24
参考文献		25
付録A	離散ウェーブレット変換(DWT)プログラム	26
付録B	離散コサイン変換(DCT)プログラム	32

図目次

2.1	冗長部分の圧縮	3
2.2	DCT を用いた情報圧縮のモデル	4
2.3	音波	5
2.4	音量と音程 1	6
2.5	音量と音程 2	6
2.6	心理聴覚特性の例	7
2.7	アナログ信号	8
2.8	標本化された信号	8
2.9	量子化された信号	9
3.1	ハールの直交ウェーブレット	18
4.1	入力信号	21
4.2	ウェーブレット変換した信号	22
B.1	入力信号	37
B.2	DCT した信号	37

第1章

序論

1.1 本研究の背景と目的

現在のインターネットや携帯電話などで行われている音声符号化の目的のひとつとして、アナログ信号をデジタル信号に変換して、さらに時間領域と周波数領域に変換することで、音声情報の特性を見つけ、その特性を用いて音声情報の冗長な部分を割りだし、除去することによる音声情報の圧縮が挙げられる。携帯電話などでは、音声情報を圧縮し、限られた帯域や伝送速度で人間にとて有益な情報を多く伝送することが音声の品質の改善につながっている。現在、最も使用されている音声符号化の直交変換の技術は JPEG などに用いられている離散コサイン変換 (DCT) である。その他に、離散フーリエ変換 (DFT), ウェーブレット変換などがある。

しかし、現在の携帯電話は、話す際に相手側の音声は聞こえるものの、その背景（臨場感）は今一つ伝わらない。例えば、携帯電話を使用している時に、相手側の音声は聞こえるが、その背景で流れている音楽などを聞き分けるのは困難である。これは、人間の発する音声を第一に考えて作られているため、その他の音はある程度聞こえなくなってしまっていいという考え方から、背景の音などは音声情報の圧縮の際に除去されたり、粗くしてしまっている。

今以上の臨場感を味わうためには、限られた帯域や伝送速度で、人間にとて有益な情報のみを多く圧縮し伝送する必要があるため、音声符号化の直交変換の技術に目をつけた。

そこで本研究では、時間周波数解析方法であるウェーブレット変換を用いた音声符号化の方式を示し、その特性を評価する。

1.2 本論文の概要

本論文の概要は以下の通りである。

まず第2章においては、音声の特徴や人間の聴覚特性などについて述べ、本研究のもととなっている従来の音声情報圧縮についての方式を説明する。

次に第3章においては、本研究で使用するウェーブレット変換について述べ、実際にウェーブレット変換で音声信号を変換し、その特徴から圧縮方法を示す。

最後に、本研究より得られた結果を第5章に結論として記す。

第 2 章

音声情報圧縮

2.1 まえがき

現在の MD や携帯電話などのデジタル情報機器は、音声信号を高度な符号化、圧縮を行い、MD などの記録媒体に記録されたり、情報ネットワークなどを通じて伝送されている。このために、まず信号を帯域通過フィルタを使い、入力信号の制限処理（入力信号で、人間の可聴周波数領域は約 20kHz であるが、電話信号の場合は 3.4kHz を信号入力とする）を行う。音声信号の入力信号の制限処理とは、周波数成分をカットすることである [1]。例えば、人間の聴覚特性を考え、人間の聴覚にあまり影響を与えない音を取り除き、データ量を減らしていくというやり方がある。これは、人間の耳には聞こえない音を取り除いていく作業で、人間の耳に聞こえない低音域や高音域をカットしてしまうというものである。つまり、取り除いても本来の情報に影響のない冗長な部分を取り除くことである（図 2.1）。

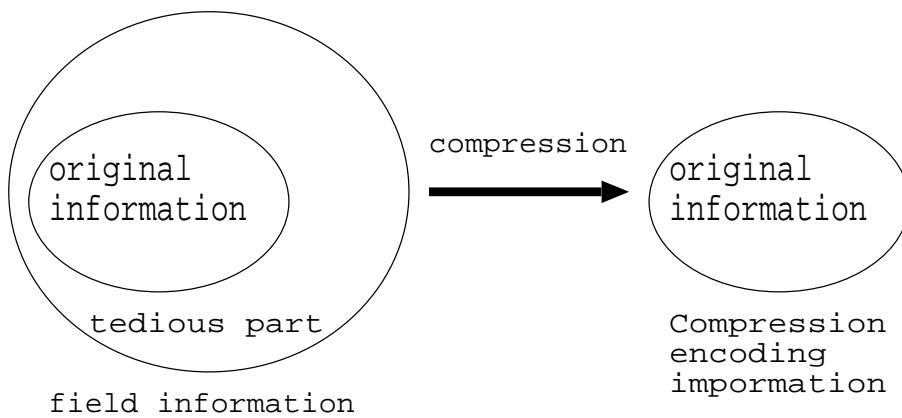


図 2.1 冗長部分の圧縮

次に適切な周期でのサンプリング処理を行う。電話信号の場合は 8kHz でサンプリングし、各サンプルデータを 8 ビット・データとするため 64kbps が音声通信の基本とされている。これは、デジタル信号処理の基本となるシャノンの定理(原信号周期の 1/2 以下の周期でサンプリングした信号は、完全に復元することが可能である)に基づいている [1]。

次に、入力信号の制限処理を終えた後、情報圧縮処理を行う。情報圧縮処理には、標本化(サンプリング)、量子化、変換符号化、およびエントロピー符号化を組み合わせた処理を行うとされている。これは、変換符号化に離散コサイン変換(DCT)を用いた情報圧縮のモデルの図である(図 2.2)。

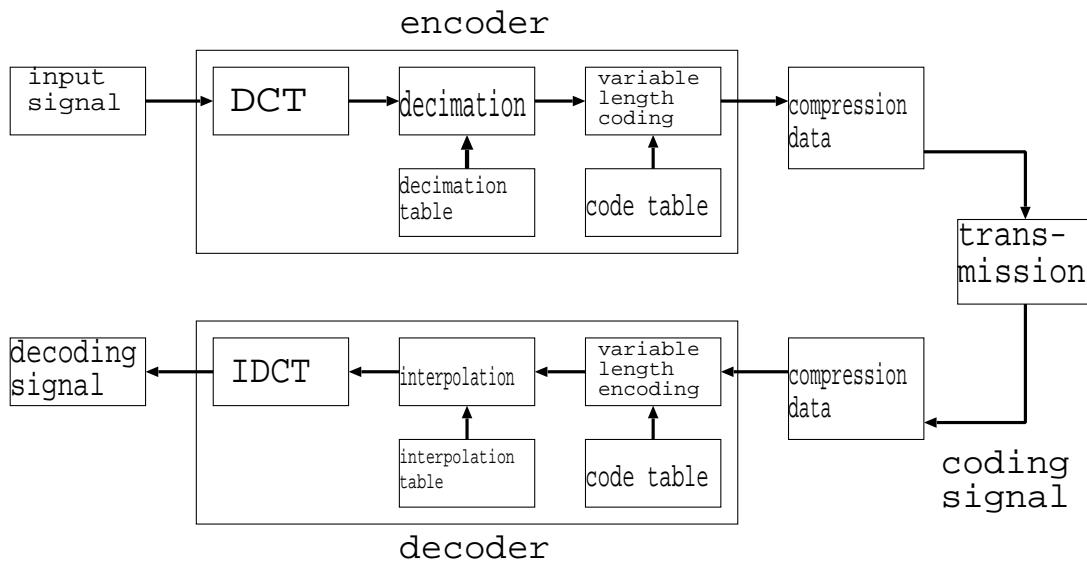


図 2.2 DCT を用いた情報圧縮のモデル

次に、音声の符号化について考えるため、音声について述べる。

2.2 音声

音声は、音波と言われる空気中を伝わる振動のことである。ここでいう振動というのは、空気の密度の変化のことで、このことから音は空気中を伝わる疎密波ともいえる。疎密波と

2.3 音の性質

いうのは、空気が圧縮された密と空気が薄まって圧力が小さい疎が動くような波である。疎密波は、波の進行方向と同じ向きに振動する波のため、縦波とも呼ばれている。ただし、縦波は視覚的に表現することが難しいため、通常は縦波を横波、つまり波の進行方向に対して垂直に変化する波として表している。これがいわゆる音の波形と呼ばれているものである[2]。(図 2.3)

音は音源から発生して空気中を伝わり、耳に届くことによって、音を聞くことができる。この空気中を伝わる音の性質や人間の聴覚特性によって、さまざまな聞こえ方(認識)となる。

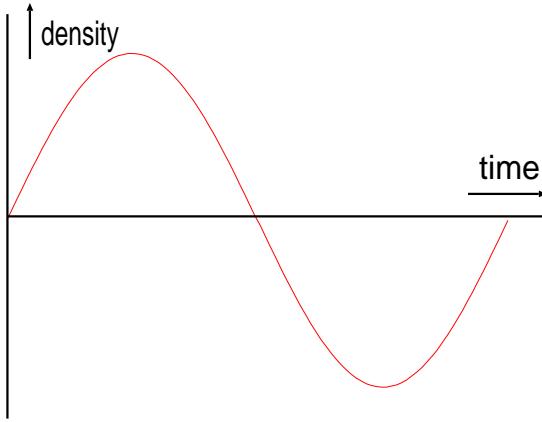


図 2.3 音波

2.3 音の性質

音には、音の三大要素といわれているものがある。それは、音量、音程、音色である。音量は音波の振幅の大きさで、振幅が大きいほど音量が大きいとなる。音程は、音波の周波数(1秒間に上下する波の回数)によって決まる。周波数が低いと低い音、周波数が高いと高い音となる。音色は、音に含まれる周波数成分の違いや、時間の経過による音量、音程の変化などがつくっているとされている[3]。

例として、振幅が 2、周波数 4Hz の波(図 2.4)と振幅が 1、周波数 2Hz の波(図 2.5)の

2.3 音の性質

波の図を以下に示す。

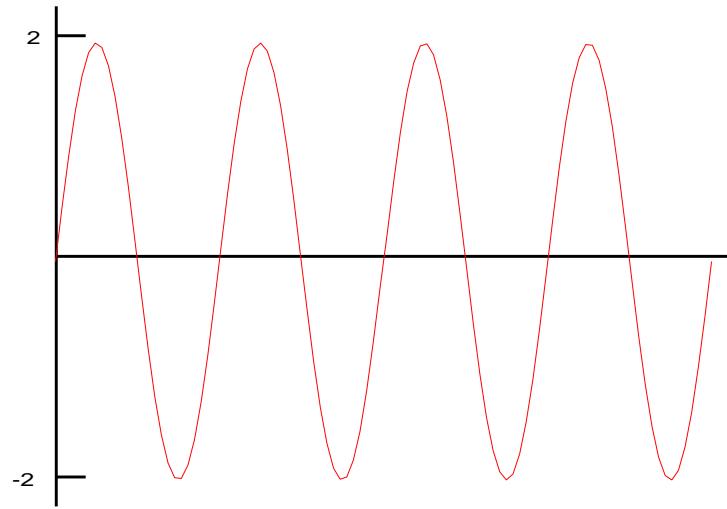


図 2.4 音量と音程 1

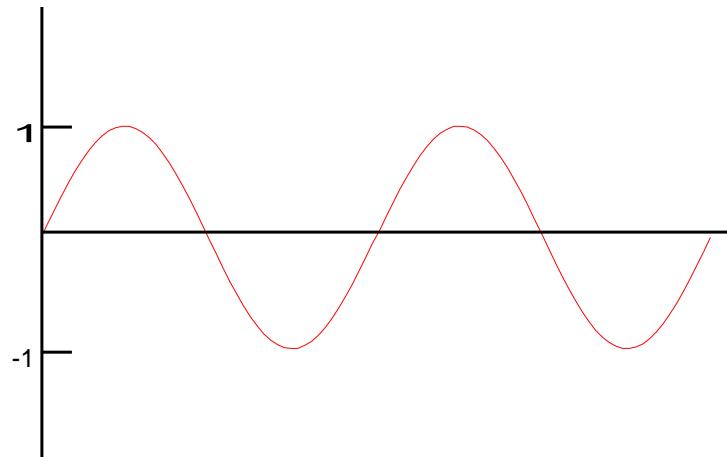


図 2.5 音量と音程 2

(図 2.4) の波の方が (図 2.5) の波より音量が大きく、音程は高くなる。

2.4 聴覚特性

人間の耳は、 $20\text{Hz} \sim 20\text{kHz}$ までの音しか基本的には聞くことはできない。しかし、これには個人差があり、サンプル音の音圧レベルや年齢、体調などによってもかなり変化する。また、静寂時の最小可聴レベル（聞くことができる音圧の最小値は）、周波数が低い部分と高い部分ではかなり限界値が高くなる。つまり、低い音、高い音はある程度大きい音でないと聞くことができないということである。また、ある強い音が存在すると、その周辺の弱い音は聞こえないという心理聴覚特性というものがある。例えば駅のプラットホームで普通に話をしていても、電車がはいってくると電車の大きな音にかき消されて、話が聞こえにくくなる現象がある。これはマスキング効果と呼ばれている[6]。また、ある強い音が存在すると、その回りの弱い音が聞こえなくなるという現象は、大きな音が発生した後だけでなく、その直前でも起こっている。つまりマスキング効果は、周波数軸だけではなく時間軸でも起こる。

(図 2.6)

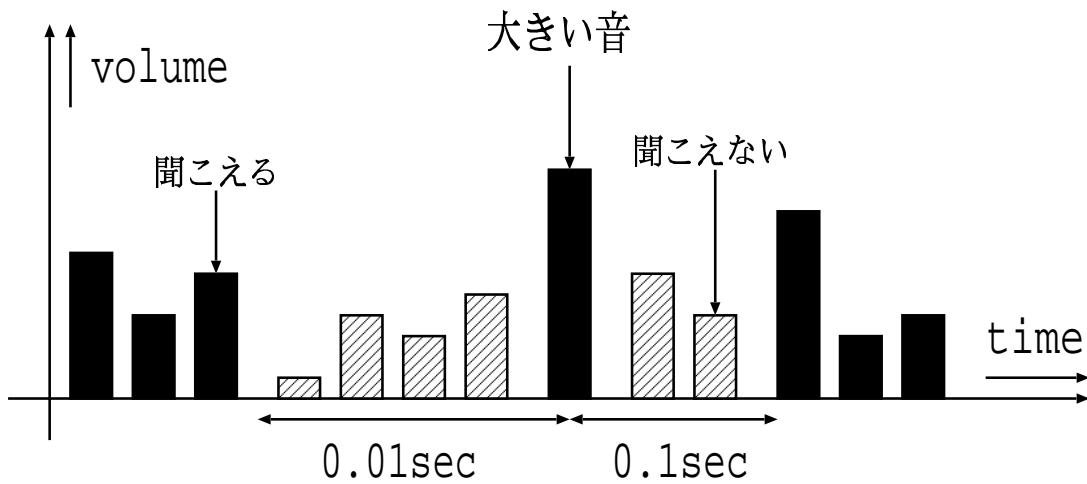


図 2.6 心理聴覚特性の例

2.5 標本化(サンプリング)

標本化は、アナログ信号から時間的に離散的な信号を取り出すことである [4]. (図 2.7) の連続信号に周期 T のインパルス列をかけることによって、サンプリングされた (図 2.8) の離散時間信号を得ることができる。標本化周波数(サンプリングレート)とは、1秒間に行うサンプリングの数でその数を増やす程、標本化の精度が上がるといわれている。たとえば CD では 44.1kHz でサンプリングされる。これは一秒間に 44,100 回サンプリングが行われていることを示す。信号が速い変化をする(高い周波数成分を持ち、音程が高い)場合、サンプリングレートはそれに合わせて設定しないと信号は正しくとらえられない [4].

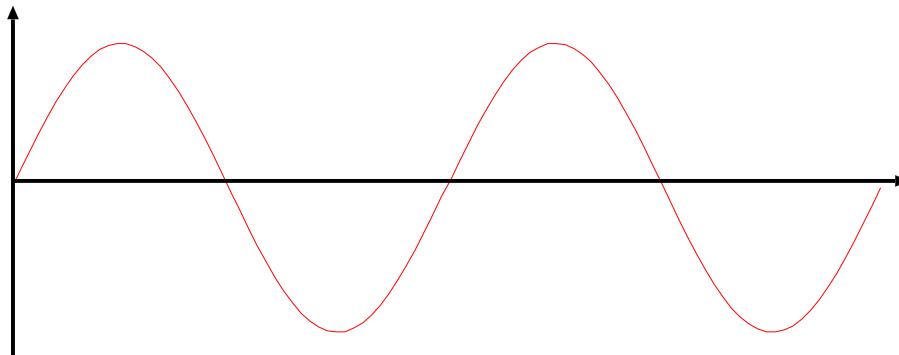


図 2.7 アナログ信号

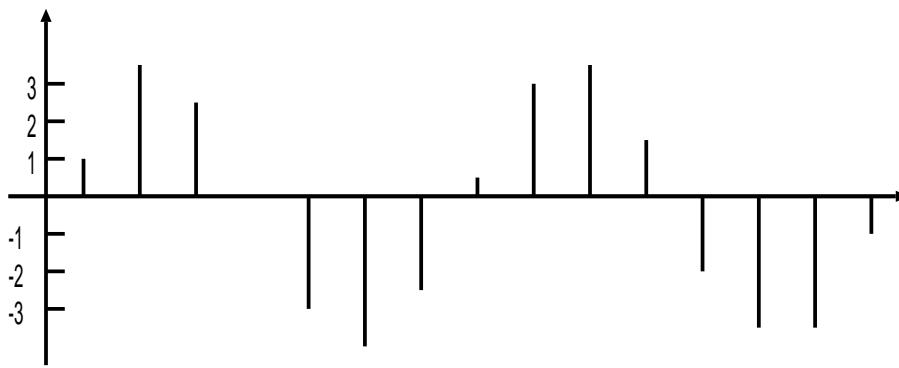


図 2.8 標本化された信号

2.6 量子化

量子化とは、サンプリングされた各時点での音圧を数値化することをいう [1]。つまり入力信号を有限長のディジタル値に変換することである。例えば、サンプル当たり 5 ビットは、各サンプルが 0 から 31 までの 32 種類のうちのひとつをとる。CD は 16 ビットで量子化が行われる。サンプルのビット数が大きければ大きいほど、可能な値を分割するステップ数(ビットレート)を増やす必要がある。ビットレートを増やすことで、サンプルの精度は高くなり、より原音に近くなる。量子化と標本化は、アナログからデジタルの変換を行う A/D 変換器とその逆の D/A 変換器によって行われる。A/D 変換器は符号化器、D/A 変換器は復号化器として用いられる。

(図 2.9) は、(図 2.8) の標本化された信号を量子化したものである。

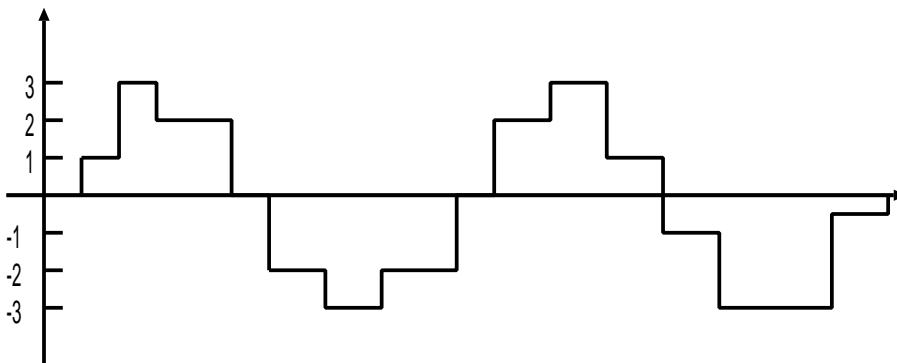


図 2.9 量子化された信号

量子化は、標本化された信号の振幅を、その近くの離散的な振幅に置き換えているため、量子化後のデータには、実際の入力値と量子化値との間に量子化誤差が発生する。これは、デジタル信号では基本的に振幅を表すための桁数が制限されているため、有限な桁数に丸め(切り捨て)なければならない。

2.7 エントロピー符号化

符号化データの発生確率の偏りを利用した可変長符号化で、圧縮率がデータ圧縮の限界であるエントロピーに漸近する符号化法 [1]。つまり出現頻度の多いものに短い符号を割り当てる事である。エントロピー符号化は、情報圧縮符号化の最終段階で適用するのが常とされている。エントロピー符号化の例としては、代表的なものとしてハフマン符号がよく用いられる。その他にもランレンジス符号化、算術符号などがある [7]。

ハフマン符号化とは、発生頻度に応じて符号の長さを可変とする効率の良い符号化であるが、発生頻度に偏りがないデータに対しては意味がないとされている。つまり文字列データでいえば、等しい文字に対して、できるだけ同じビット列の符号を割り当てる事によって、元のビット数を減らすというものである [7]。

ランレンジス符号化とは、ファクシミリのように信号が白黒の2値画像の場合や、その他の場合でも同じ信号が続く場合には、同じ信号の連続の長さを情報として符号化することにより、効率化を図る方式である [6]。

算術符号とは、情報を0と1の出現確率に応じて小数を対応させ、その小数点以下の部分を符号とするものである。たとえば、対応する小数が0.10011101なら、10011101を圧縮後の符号にするというものである [7]。

2.8 直交変換符号化

直交変換とは、音声信号を周波数成分に変換する方式である。直交変換では変換後もエネルギーが保存され、逆変換による完全再生が可能である。変換符号化は、信号領域をある関数によって変換し、べつの領域へ写像ことである。直交変換の方式には、JPEGで採用されている離散コサイン変換(DCT)を始め、離散フーリエ変換(DFT)、ウェーブレット変換などがあるが、情報圧縮に適用可能な直交変換の方式で、実質的に有効なのはDCTであると言われている。

2.8.1 離散フーリエ変換 (DFT)

離散フーリエ変換は、音声の信号を解析するための技術で、音声の信号の波の特徴を正確に知るために用いる。ディジタルの時系列信号（時間領域によるインパルス列）を周波数分析する場合に用いることが多い。それは、連続でない波では信号そのものが離散的な数値列として表されているので、周波数スペクトルもそれに対応した離散的な数値列として表現されるように、離散フーリエ変換 (DFT:Discrete Fourier Transform) が用いられる。

有限長時系列 信号 $x[n]$ は、離散フーリエ変換 (DFT) で次式で表される。

$$X(k) = \sum_{n=0}^{N-1} x(n) \exp\left(-j \frac{2\pi kn}{N}\right), k = 0, 1, \dots, N-1 \quad (2.1)$$

離散フーリエ変換 (DFT) によって、得られた周波数特性（振幅特性と位相特性）から、元の時間信号を推定し直すのが、離散的フーリエ逆変換 (IDFT) であり、次式で表される。

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) \exp\left(j \frac{2\pi kn}{N}\right), n = 0, 1, \dots, N-1 \quad (2.2)$$

2.8.2 離散コサイン変換 (DCT)

音声・画像データを周波数成分に変換する直交変換の 1 つ、Discrete Cosine Transform を省略して、DCT と呼ばれることがある。DCT は、現在最も広く情報圧縮アルゴリズムとして、採用されているものである。DCT を使ったデータ圧縮では、音声の N 点を直交変換し、周波数成分を示す係数を得る。これと量子化を組み合わせることで高い圧縮率と比較的劣化の少ない音声を実現している。DCT では相関の強い信号は、低周波数領域に信号の電力が集中する傾向がある。従って DCT 係数でも電力が集中する係数と集中しない係数が存在する。DCT を用いたデータ圧縮とは、この電力集中を応用して、電力の集中する領域に多くのビット数を、電力が集中しない領域では少ないビット量を割り当てることで行う。その最も単純な方式が、高周波領域の一部が 0 に近いことを利用して、0 値近似してしまう（すなわちこの部分のデータを切り捨てる）ことで行う方法である。復元する場合は、切り捨てた部分に 0 の値を挿入して IDCT を行う。

入力ベクトルを x とした時の、 N 点離散コサイン変換は、

$$\begin{aligned} X(k) &= \frac{2}{N} c_k \sum_{n=0}^{N-1} x(n) \cos \left[\frac{(2n+1)k\pi}{2N} \right] \\ c_k &= \begin{cases} \frac{1}{\sqrt{2}}, k = 0 \\ 1, k \neq 0 \end{cases}, k = 0, 1, \dots, N-1 \end{aligned} \quad (2.3)$$

で定義される。また、 N 点逆離散コサイン変換 (IDCT) は、

$$\begin{aligned} x(n) &= \sum_{k=0}^{N-1} c_k X(k) \cos \left[\frac{(2n+1)k\pi}{2N} \right] \\ c_k &= \begin{cases} \frac{1}{\sqrt{2}}, k = 0 \\ 1, k \neq 0 \end{cases}, n = 0, 1, \dots, N-1 \end{aligned} \quad (2.4)$$

で定義される。

DCT 方式は、直交変換の中で最も効率が高い方式として知られている。また DCT 方式は、MPEG、JPEG、H.261 などの国際標準規格にも採用されている。

第3章

ウェーブレット変換

3.1 まえがき

いろいろな波形の周波数成分を調べるのに、今までフーリエ級数展開やフーリエ変換を使ってきたが、これはある波を沢山の正弦波の和の形で表わすというものだった。ところが正弦波は、負の無限大から正の無限大まで限りなく続く定常的な波であるため、フーリエ変換ではさまざまな困難が生じます。

- フーリエ変換を行うとき、元になる波形と正弦波との積の積分計算(内積)を、負の無限大から正の無限大までしなければならず、計算に大変時間がかかる。
- 波形の一部分を変えただけでも全体の周波数成分が変わってしまう。

などが挙げられる [11]。このような困難を解消するために考えられたのがウェーブレット解析である。ウェーブレットは (Wavelets) は、三角関数や対数関数のような特定の決まった関数ではなく、局在する波を表すさまざまな関数の総称(局所的な参照波)である。つまり、ウェーブレットと呼ばれるさざなみのような振動した小さい波のひとたまりを考え、元の波を、正弦波のような定常的な波ではなく、沢山のウェーブレットの和の形に表わすものである。語源は、波 (wave) と小さな (lets) の複号語であり、時間と周波数の小さなさざ波のことである [11]。

本章では、このウェーブレットを用いた音声符号化の方式を示す。

3.2 ウエーブレット変換

3.2.1 ウエーブレット解析

さざなみのような振動した小さい波 $\psi(t)$ (マザーウエーブレット) を考える。さざなみであるから、ある幅で極在している必要がある。そこで、このさざなみは 2 乗可積分であると仮定する。

(実数 R 上で定義された関数 ψ のうち、次の条件を満たすものを 2 乗可積分関数という)

$$\int_{-\infty}^{\infty} |\psi(t)|^2 dt < \infty$$

この波を時間軸に a 倍する操作と時間軸に沿って b だけシフトする操作により、さまざまなスケール a と位置 b をもったさざなみ状の関数の集合

$$\psi_{a,b}(t) = \frac{1}{\sqrt{a}} \psi \left(\frac{(t-b)}{a} \right) \quad (3.1)$$

を作成する。式 (3.1)において、さざなみ $\psi(t)$ を時間軸方向に a 倍すると同時に振幅方向に $1/\sqrt{a}$ 倍しているが、これは、 $\psi_{a,b}(t)$ の 2 乗積分値、すなわちエネルギーを一定にするための便宜上の操作である [10]。

3.2.2 ウエーブレット変換

これらのさざなみ $\psi_{a,b}(t)$ と信号 $f(t)$ の内積

$$W_{\psi}(t)(a, b) = \int_{-\infty}^{\infty} f(t) \psi \left(\frac{(t-b)}{a} \right) dt \quad (3.2)$$

を求めたとする。パラメータ a, b は、それぞれスケール、位置を表しており、時系列に対しでは、おおよそ周波数、時刻にそれぞれ対応させることができる。したがって、さざなみ状の関数と時系列の内積は、各時刻での周波数特性を求めるために似ている。すなわち、時系列の時間-周波数解析のようなことを行えることになる [9]。これら平行移動パラメータ b と拡大縮小動パラメータ a を適切に決定し、 $\psi((x-b)/a)$ と置換することで、パターンの局所的な性質を調べる。また式 (3.2) のような変換をウエーブレット変換という。ウェーブレッ

ト変換は離散コサイン変換同様、音声、画像データを周波数成分に変換する直交変換の1つである。

ガボール変換では、時間の分解能（時刻の刻みを細かくする）と周波数の刻みが粗くなり、その逆もいえる。しかし、ウェーブレット変換ではこの問題を、高周波領域を解析するには短い基底を用いて時間分解能を上げ、低周波領域を解析するには長い基底を用いて時間分解能を上げることで解決している。これは、基底関数を一つの核関数により定義し、この核関数（マザーウェーブレット）の伸張と移動によって実現できる。すなわち、前に述べたようにマザーウェーブレットを $\psi(t)$ とすると、基底関数は $\psi_{a,b}(t) = (1/\sqrt{a})\psi((t-b)/a)$ となる。 a, b は伸張、移動パラメータである [10]。このウェーブレットの原点は、ガボールの短時間フーリエ変換と呼ぶ時間周波数解析方法であり、ガボール変換と呼ばれ、今日でも位相情報を保存するウェーブレット変換として良く用いられている [9]。ガボールウェーブレットは、

$$\psi(t) = \int_{-\infty}^{\infty} \frac{1}{2\pi} e^{-\frac{t^2}{4}} (e^{-2\pi t} - e^{2\pi t}) dt \quad (3.3)$$

と表せ、ガウス関数を窓関数として用いている。

3.2.3 連続ウェーブレット変換

マザーウェーブレット $\psi(x)$ による、関数 $f(x)$ の連続ウェーブレット変換は、

$$(W_\psi f)(b, a) = \frac{1}{\sqrt{|a|}} \cdot \int_{-\infty}^{\infty} \overline{\psi\left(\frac{x-b}{a}\right)} \cdot f(x) dt \quad (3.4)$$

で与えられる。 $\psi((x-b)/a)$ が信号 $f(x)$ の部分に似ているときは、積分の値は大きくなる。

逆に、似ていないときは、正と負の両方の領域に値が存在し、積分の値はほぼ 0 となる。

ウェーブレット $\psi((x-b)/a)$ は、マザーウェーブレット $\psi(x)$ を b トランスレーション（平行移動、シフト）し、 a スケール（拡大縮小）したものである。スケールパラメータ a に対応して $\psi(x)$ の幅が a 倍になることから、 $1/a$ が周波数に対応していることが分かる。すなわち、 a が大きいと基底関数は広がり、小さいと狭まる。つまり、大きい a が低周波に、小さい a が高周波に対応する。マザーウェーブレット ψ およびスケールパラメータ a 、トラン

3.3 離散ウェーブレット変換

スレーションパラメータ b を適切に選ぶことにより、信号 $f(x)$ の局所的様子を調べることができます。

逆に、ウェーブレット変換から元の信号 $f(x)$ を復元することもできる。つまり、逆ウェーブレット変換が存在し、次のように定義される。

$$f(x) = \frac{1}{C_\psi} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} (W_\psi f)(b, a) \cdot \frac{1}{\sqrt{|a|}} \psi\left(\frac{x-b}{a}\right) \frac{1}{a^2} da db \quad (3.5)$$

ここで、右辺が定義できるためには、次のアドミッシブル条件が満たさなければならない。

$$C_\psi = \int_{-\infty}^{\infty} \frac{|\widehat{\psi}(\omega)|^2}{|\omega|} d\omega < \infty \quad (3.6)$$

なお、 $\widehat{\psi}$ は ψ のフーリエ変換である。しかし、実際においては以下の条件式が代用されることがある。

$$\int_{-\infty}^{\infty} \psi(x) dx = 0 \quad (3.7)$$

この式は $\psi(x)$ が振動的であることを意味する。

ウェーブレット変換の定義式では、積分範囲は $-\infty$ から ∞ となっているが、マザーウェーブレットはサポートコンパクト（範囲が限られている）なので、マザーウェーブレットの幅の範囲分だけ積分すれば良い。ただし、マザーウェーブレットの幅は、スケールパラメータ a によって変化する [9]。

3.3 離散ウェーブレット変換

ウェーブレット変換は、パラメータ a, b が連続である限り、いかなるウェーブレットに対しても直交性を満足しないことが示されている。直交変換を実現するには、パラメータ a, b を離散化することを考えなければならない。ここで、離散化にあたり、2進分割をとり、 $a = 2^j, b = 2^j k$ とすると、式 (3.1) の連続ウェーブレットは次のように離散化され、これを $\psi_{j,k}$ で表す。

$$\psi_{j,k}(t) = 2^{-\frac{j}{2}} \psi(2^{-j}t - k) \quad (3.8)$$

3.3 離散ウェーブレット変換

ここで, j, k はそれぞれ a, b に対応する正数である. b に対応するのは $2^j k$ であり, 移動量は基底関数の広がりに伴う. つまり, 長い基底関数の移動は大きく, 短い基底関数では移動は小さい. また, この $\psi_{j,k}(t)$ の形式で 2 乗可積分関数空間 l^2 の直交基底を構成できるので直交基底のウェーブレットが実現できる.

マザーウェーブレット $\psi(x)$ による, 関数 $f(x)$ の離散ウェーブレット変換 DWT(Descret Wavelet Transformation) は,

$$d_k^{(j)} = 2^j \cdot \int_{-\infty}^{\infty} \overline{\psi(2^j x - k)} \cdot f(x) dx \quad (3.9)$$

で与えられ, その逆変換は,

$$f(x) = \sum_j \sum_k d_k^{(j)} \psi(2^j x - k) \quad (3.10)$$

で与えられる [10]. 連続ウェーブレット変換では, アドミッシブル条件を満たす関数であれば, どのような関数でもマザーウェーブレット ψ として用いることができる. しかし, 離散ウェーブレット変換では, ψ は基底関数となるものでなければならない. しかも, 応用上はコンパクトなサポートも持つ関数 (関数値が 0 でない区間が有限である関数) が望ましい [10]. 次に, 直交基底のウェーブレットで最もシンプルだといわれているハールウェーブレット (Haar wavlet) について述べる.

3.3.1 ハールウェーブレット

ハールウェーブレットは, 式で表すと,

$$\psi(t) = \begin{cases} 1 & (0 \leq t < \frac{1}{2}) \\ -1 & (\frac{1}{2} \leq t < 1) \\ 0 & (otherwise) \end{cases} \quad (3.11)$$

であり, 図で表すと (図 3.1) である.

ハール関数により生成されたウェーブレットは, 正規直交性を満たし, かつ局在性も満たしている. しかし, ウェーブレット関数系が不連続であるため, 展開の収束が遅く, 限定された場合にしか実用にはならない [11].

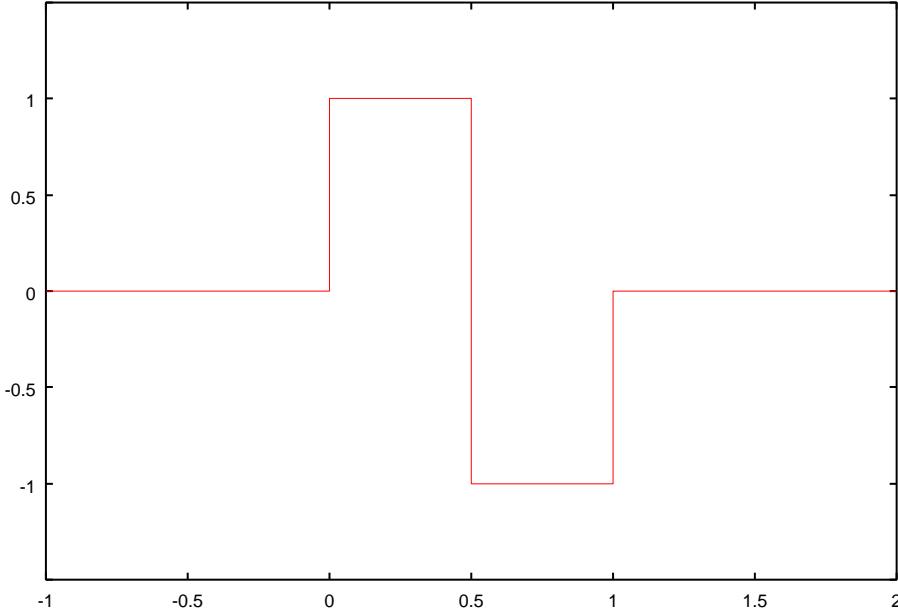


図 3.1 ハールの直交ウェーブレット

3.3.2 信号の分解と再構成

まず、信号を、スケーリング関数と呼ばれる関数の一次結合で近似する。スケーリング関数の一次結合を近似関数と呼ぶ。また、近似の精度をレベルと呼び、レベル 0 が最も精度の高い近似となる。レベルが大きくなるに従い、粗い近似となる。任意のレベル j の近似関数と、それより 1 レベル粗い $j+1$ レベルの近似関数との差は、ウェーブレットの一次結合である。信号は、レベル 1 から、任意のレベルまでのウェーブレットと、スケーリング関数との一次結合の和で表される [9]。

3.3.3 ドベシイの直交ウェーブレット

ドベシイのウェーブレットおよび、そのスケーリング関数は形が複雑で、既知の関数では表現できない。ドベシイのウェーブレットでは、自然数 N を用いて、ウェーブレットの特徴づけを行っている。

ドベシイのウェーブレットには、以下の特徴もある。

- ウェーブレットおよびスケーリング関数が、時間軸上で長さが $2N - 1$ の範囲に局在している。つまり、このウェーブレットおよびスケーリング関数はコンパクトサポート（範囲が限られている）を持つ。
- ウェーブレットでは、0次から、 $N - 1$ 次までのすべてのモーメントが0である。

ここで、 $N - 1$ 次モーメントまでが0であることをしきで表すと、式(3.13)のようになる。

$$\int_{-\infty}^{\infty} x^n \psi(x) dx = 0 \quad (n = 0, 1, \dots, N - 1) \quad (3.12)$$

ドベシィのウェーブレットはハールのウェーブレットとは異なり、直接 $\psi(t)$ が与えられるのではなく、 $2N$ 個の離散数列のみが与えられる。

$$w_k^{(j)} = \int_{-\infty}^{\infty} f(t) \overline{\psi_{j,k}(t)} dt = \langle f, \psi_{j,k} \rangle \quad (3.13)$$

式(3.13)のように内積で求めるには、まず、繰り返し計算から $2N$ 個の離散数列からウェーブレットを近似する精度の高い関数を求め、次に、数値計算により、信号とその関数の内積を求めることになる[9]。

また、ドベシィのウェーブレットは、 $2N$ 個の離散数列から直接、ウェーブレットの展開係数を求めることができることと、演算量を減らすことができるため、本研究では、ウェーブレットの直交基底にドベシィのウェーブレットを用いることにする。

第4章

ウェーブレット変換を用いた音声符号化

4.1 まえがき

本章では、3章で述べた離散ウェーブレット変換の基底にドバシィのウェーブレットを用いて実際に変換と逆変換を行い、その特性からウェーブレット変換を用いた音声符号化の方法を示す。

4.2 縮散ウェーブレット変換と逆変換

図(4.1)の入力信号を縮散ウェーブレット変換と縮散ウェーブレット逆変換した信号を図(4.2)に示す。

4.3 ウェーブレット変換した信号の音声符号化

ウェーブレット変換した信号について、人間の聴覚特性とマスキング効果を利用した音声符号化の方法を示す。

人間の耳は、通常 $20\text{Hz} \sim 20\text{kHz}$ までの音しか聞くことはできない。よって、これよりも高い周波数成分を除去してしまってもほとんど影響はないといわれている[6]。また、静寂時の最小可聴レベル(聞くことができる音圧の最小値)は、周波数が低い部分と高い部分で

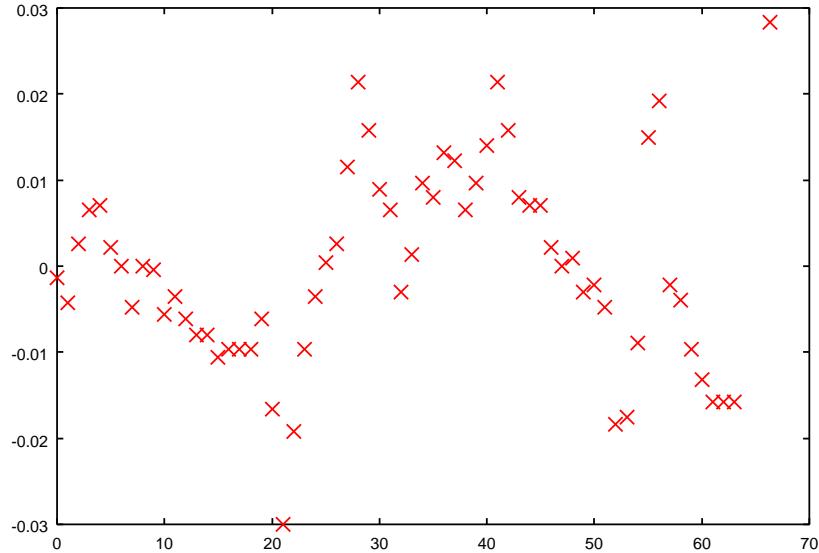
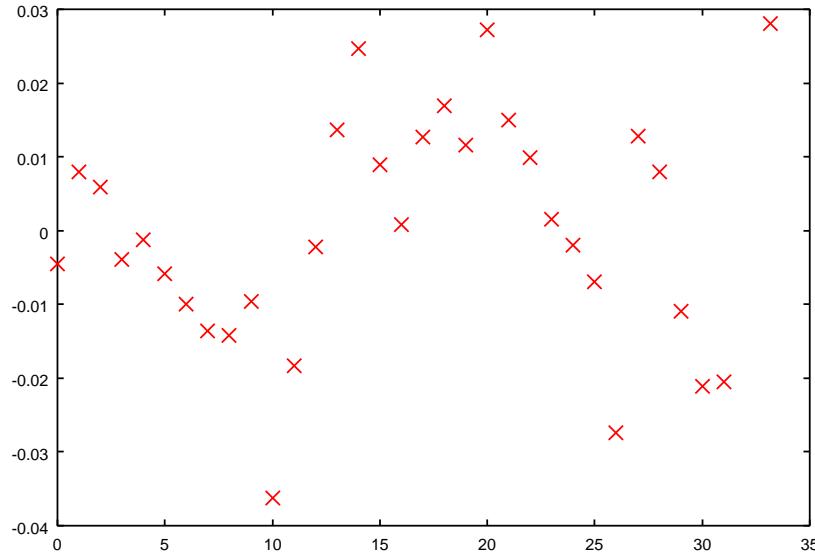


図 4.1 入力信号



はかなり限界値が高くなる。つまり低い音、高い音はある程度大きい音でないと聞くことができないという特性がある。この特性を用いて、聞こえない部分の周波数成分を除去してしまうことや、聞こえにくい部分の周波数成分を粗くしたりして情報量を減らす。

また、500Hz～1kHz の人間が聞き取りやすい低周波数領域には信号の電力が集中する傾向がある。この電力集中を応用して、電力の集中する領域に多くのビット数を割り当てる、電力が集中しない領域では少ないビット数を割り当てる。こうして情報量を減らす。

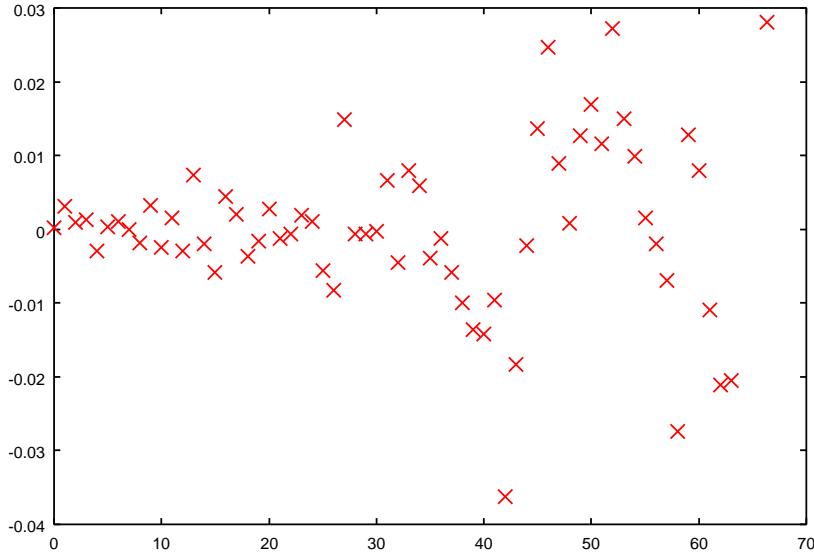


図 4.2 ウェーブレット変換した信号

マスキング効果とは、ある強い音が存在すると、その周辺の弱い音は聞こえなくなるというものである。これは、大きな音が発生した後だけでなく、その直前でも起きている。つまりマスキング効果は、周波数軸だけではなく時間軸でも起こる。このマスキング効果を応用して、強い音が存在している周辺の弱い音を除去することや強い音により少しだけマスキングされた音に関しては、その差分を求ることによって情報量を減らす。マスキング効果を用いることで、符号長が短くなり、処理が簡単になる。

以上のような方式を示す。

第 5 章

結論

5.1 離散ウェーブレット変換と離散コサイン変換についての検討

ある同一の音声信号について、離散ウェーブレット変換と離散コサイン変換についての検討を両方についておこなったが（付録 B に示す）、離散コサイン変換 (DCT) した信号の方がどちらかといえば、その特性がはっきりと出ていた。今回は、基底のウェーブレットがあまり合っていなかったため、うまくいかなかった。しかし、離散ウェーブレット変換 (DWT) は、変換するための基底を自由に選べるため、それぞれの音声信号に適した基底を見つけることで、さまざまな特性が挙げることができる。

5.2 結論

本研究では、ある音声信号について離散ウェーブレット変換を実際に行い、その結果から、聴覚特性とマスキング効果を利用するという音声情報圧縮の方式を示した。また、離散ウェーブレット変換は信号によって、適した基底を用いることが大事であるということを示した。

謝辞

本研究を行うにあたり、ご指導、ご助言を頂いた福本昌弘講師に深く感謝いたします。また、本論文のご審議をしてくださる島村和典教授、菊地豊助教授に深謝いたします。最後に、日頃から他面にわたくってご協力を頂いた福本研究室の諸氏に感謝いたします。

参考文献

- [1] 藤原 洋 編, マルチメディア情報圧縮, 共立出版株式会社, 2000
- [2] 鹿野 清宏, 中村 哲, 伊勢 史郎, 音声・音情報のデジタル信号処理, 昭晃堂, 1997
- [3] トランスナショナル カレッジ オブ レックス, フーリエの冒険, ヒッポファミリークラブ, 1988
- [4] 貴家 仁志, マルチレート信号処理, 昭晃堂, 1995
- [5] 中野 宏毅, 山本 鎮男, 吉田 靖夫, ウェーブレットによる信号処理と画像処理, 共立出版株式会社, 1999
- [6] 電子情報通信学会 編, デジタル信号処理ハンドブック, オーム社, 1993
- [7] 越智 宏, 黒田 英夫, 図解でわかる 画像圧縮技術, 日本事業出版社, 1999
- [8] 寺田 文行, 中村 直人, 釈氏 孝浩, 松居 辰則, 情報数学の基礎, サイエンス社, 1999
- [9] 中野 宏毅, 山本 鎮男, 吉田 靖夫 ウェーブレットによる信号処理と画像処理, 共立出版株式会社, 1999
- [10] 新井 康平, ウェーブレット解析の基礎理論, 森北出版株式会社, 2000
- [11] 白井 支朗, 信号解析, オーム社, 1998

付録 A

離散ウェーブレット変換 (DWT) プログラム

ここでは、4.2で示したドベシィのウェーブレットを基底にした、1次元離散ウェーブレット変換、離散ウェーブレット逆変換のC言語によるプログラムを示す。

```
/*離散ウェーブレット変換 ヘッダーファイル*/\n\n#define L 64      /*1次元信号の長さ*/\n#define L2 32     /*L/2*/\n#define K 4       /*数列p_kの長さ*/\n#define PI 3.14159265358979\n\n#ifndef max\n#define max(A, B) ((A) > (B) ? (A) : (B))\n#endif\n\nvoid fwt1d (\n    double s0[],\n    unsigned int s_len,\n    double p[],\n    double q[],
```

```

        unsigned int sup,
        double s1[],
        double w1[]
    );

void ifwt1d (
    double s1[],
    double w1[],
    unsigned int s_len,
    double p[],
    double q[],
    unsigned int sup,
    double s0[]
);

void main(void);

/*1 次元高速ウェーブレット変換メインプログラム
functions used:
fwt1d : 1 次元高速ウェーブレット変換
ifwt1d : 1 次元高速ウェーブレット逆変換*/

```

```

#include <stdio.h>
#include <math.h>
#include "fwt.h"

```

```

void fwt1d(), ifwt1d();

void main(void){

    double s0[L];           /*入力信号*/
    double s0r[L];          /*再構成信号*/
    double s1[L2], w1[L2];  /*分解信号*/
    double p[K] = {0.482962913145, 0.836516303738,
                   0.224143868042, -0.129409522551}; /*ドベシイの数列 p_k (N=2)*/
    double q[K];            *ドベシイの数列 q_k (N=2)*

    int i;
    int sup = K, s_len = L;

    FILE *fp;
    fp=fopen("sp1","r");
    for (i = 0; i < L; i++){
        fscanf(fp,"%lf",&s0[i]);
    }
    fclose(fp);

    for(i = 0; i < sup; i++){
        q[i] = pow(-1,i)*p[sup-i-1];
    }

    printf("入力信号 \n");
}

```

```

for(i = 0; i < L; i++){
    printf("%7.4f ",s0[i]);
}

printf("\n");

fwt1d(s0,L,p,q,sup,s1,w1);

printf("DWT した信号 \n");
for(i = 0; i < L; i++){
    printf("%7.4f ",s1[i]);
}

printf("\n");

printf("DWT した信号 \n");
for(i = 0; i < L; i++){
    printf("%7.4f ",w1[i]);
}

printf("\n");

ifwt1d(s1,w1,L/2,p,q,sup,s0r);

printf("再構成信号\n");
for(i = 0; i < L; i++){
    printf("%7.4f ",s0r[i]);
}

```

```

    printf("\n");
}

void fwt1d (
    double s0[],           /*入力信号*/
    unsigned int s_len,     /*入力信号長*/
    double p[],             /*スケーリング数列 pk*/
    double q[],             /*ウェーブレット数列 qk*/
    unsigned int sup,        /*数列長*/
    double s1[],            /*分解信号*/
    double w1[])            /*分解信号*/
{
    unsigned int n, k;
    unsigned int index;

    for (k = 0; k < s_len/2; k++) {
        s1[k] = 0.0;
        w1[k] = 0.0;
        for (n = 0; n < sup; n++) {
            index = (n+2*k)%s_len ;
            s1[k] += p[n] * s0[index];
            w1[k] += q[n] * s0[index];
        }
    }
}

```

```

void ifwt1d (
    double s1[],          /*分解信号*/
    double w1[],          /*分解信号*/
    unsigned int s_len,   /*信号長*/
    double p[],           /*スケーリング数列 pk*/
    double q[],           /*ウェーブレット数列 qk*/
    unsigned int sup,     /*数列長*/
    double s0[]           /*再構成信号*/
)
{
    unsigned int n, k;
    unsigned int index, ofs;

    ofs = max(1024,s_len);      /*index が負にならないための補正值*/
    for (n = 0; n < s_len; n++) {
        s0[2*n+1] = 0.0;
        s0[2*n] = 0.0;
        for (k = 0; k < sup/2; k++) {
            index = (n-k+ofs) % s_len ;
            s0[2*n+1] += p[2*k+1] * s1[index] + q[2*k+1] * w1[index];
            s0[2*n] += p[2*k] * s1[index] + q[2*k] * w1[index];
        }
    }
}

```

付録 B

離散コサイン変換 (DCT) プログラム

ここでは、2.8.2で説明した、1次元離散コサイン変換(DCT)、離散コサイン逆変換(IDCT)のC言語によるプログラムを示す。

```
/*離散コサイン変換 ヘッダーファイル*/\n\n#define L 64      /*1次元信号の長さ*/\n#define L2 32     /*L/2*/\n#define PI 3.14159265358979\n\n#ifndef max\n#define      max(A, B)      ((A) > (B) ? (A) : (B))\n#endif\n\nvoid dct1d (\n    double s0[],\n    unsigned int s_len,\n    double xk[]\n);
```

```

void idct1d (
    double s0[],
    unsigned int s_len,
    double xk[]
);

void main(void);

/*1 次元離散コサイン変換, 逆変換メインプログラム

functions used:
dct1d : 1 次元離散コサイン変換
idct1d : 1 次元離散コサイン逆変換*/



#include <stdio.h>
#include <math.h>
#include "dct1.h"

void dct1d(),idct1d();

void main(){

double s0[L]; /*入力信号*/
double s0r[L]; /*再構成信号*/
double xk[L]; /*DCT した信号*/

int i;
}

```

```
int s_len = L; /*入力信号長*/
```

```
FILE *fp;
```

```
fp=fopen("sp1","r");
```

```
for (i = 0; i < L; i++){
```

```
    fscanf(fp,"%lf",&s0[i]);
```

```
}
```

```
fclose(fp);
```

```
printf("入力信号 \n");
```

```
for(i = 0; i < L; i++){
```

```
    printf("%7.4f ",s0[i]);
```

```
}
```

```
printf("\n");
```

```
dct1d(s0,L,xk);
```

```
printf("DCT した信号\n");
```

```
for(i = 0; i < L; i++){
```

```
    printf("%7.4f ",xk[i]);
```

```
}
```

```
printf("\n");
```

```
idct1d(s0r,L,xk);
```

```

printf("再構成信号\n");

for(i = 0; i < L; i++){
    printf("%7.4f ",s0r[i]);
}

printf("\n");

}

void dct1d (
    double s0[],
    unsigned int s_len,
    double xk[])
{
    unsigned int n, k;
    double ck,x;

    for (k = 0; k < s_len; k++)
    {
        if (k == 0)
            ck = 1/sqrt(2);
        else
            ck = 1;

        for (n = 0,x = 0; n < s_len; n++)
        {
            x += cos((2*n+1)*k*PI/(2*s_len))*s0[n] ;
        }
    }
}

```

```

xk[k] = 2*ck*x/s_len;

}

}

void idct1d (
    double s0r[],
    unsigned int s_len,
    double xk[])
{

    unsigned int n, k;
    double ck;

    for (n = 0; n < s_len; n++)
        s0r[n] = 0;

    for (k = 0; k < s_len; k++)
    {
        if (k == 0)
            ck = 1/sqrt(2);
        else
            ck = 1;

        for (n = 0; n < s_len; n++)
{

```

```

s0r[n] += cos((2*n+1)*k*PI/(2*s_len))*ck*xk[k];
}

}

```

結果を図で表すと、以下のようになる。

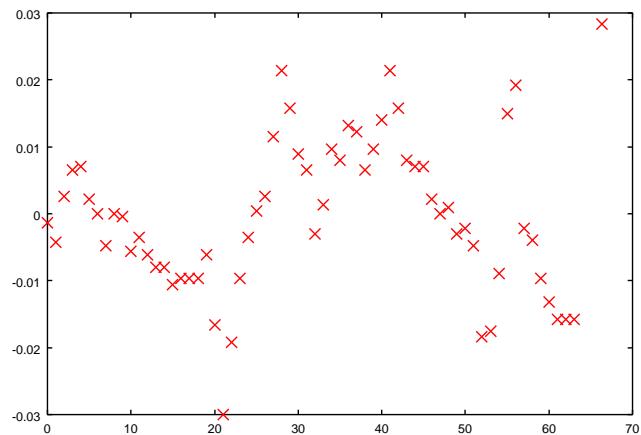


図 B.1 入力信号

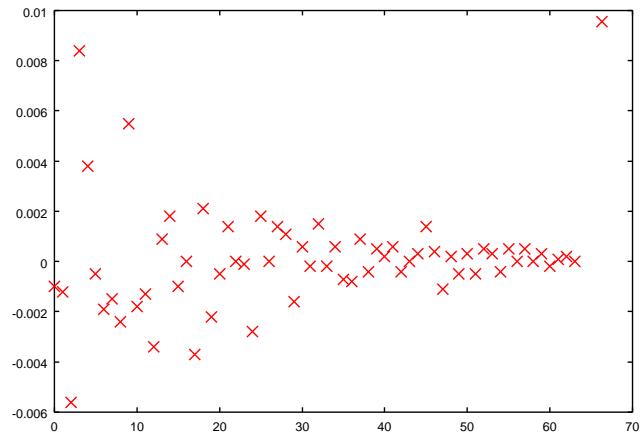


図 B.2 DCT した信号