テキスト (応用編)

1.画像の拡大･縮小

1．1　画像サイズを整数倍に拡大する

画像サイズの拡大法とは、サイズの拡大に伴う新たな画素の決定法(補完法)である。いま、画像を縦に L 倍、横に M 倍に拡大(ただし L、M 共に正の整数)する場合について考える。

もっとも単純な方法として、Fig 1.1 に示すように原画の各画素を単純に L×M の領域に並べる方法がある。同図(b)は、この操作を画像の断面から見たものである。この方法を零時ホールド法(アップ･サンプリング法)と呼ぶ。

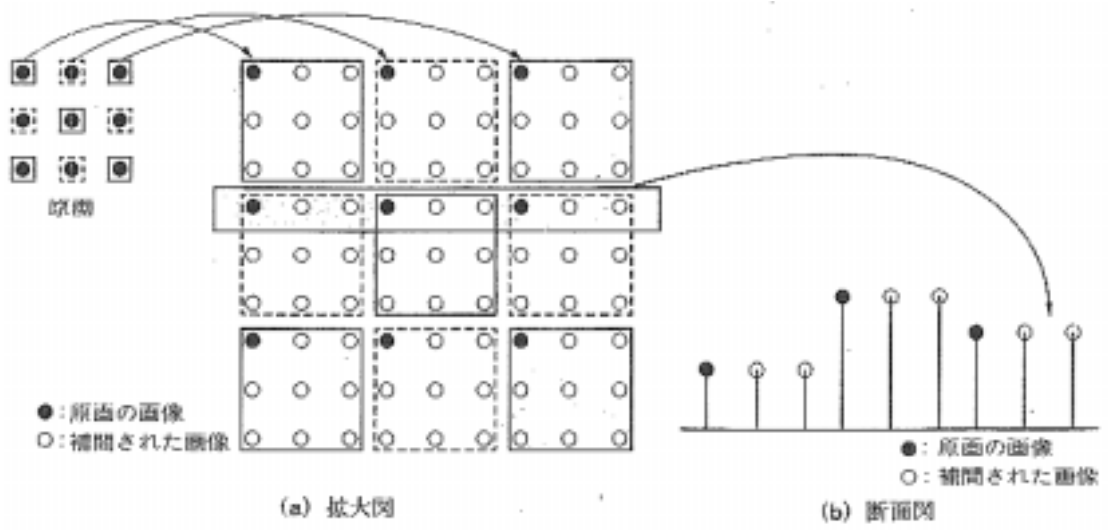


Fig 1.1 零次ホールド法(L = M = 3)

1．2　画像サイズを整数分の 1 に縮小する

画像サイズの縮小は、サイズの縮小に伴う余分な画素の消去である。いま、画像を縦に 1/L、横に 1/M するように縮小(ただし、L、M 共に正の整数)する場合について考える。

最も簡単な方法は、Fig 1.2 に示すように、原画の L×M の各領域から 1 画素を単純に選び出し、画像を再構成する方法である。この方法をダウン･サンプリング法(あるいは、サブ･サンプリング法)という。

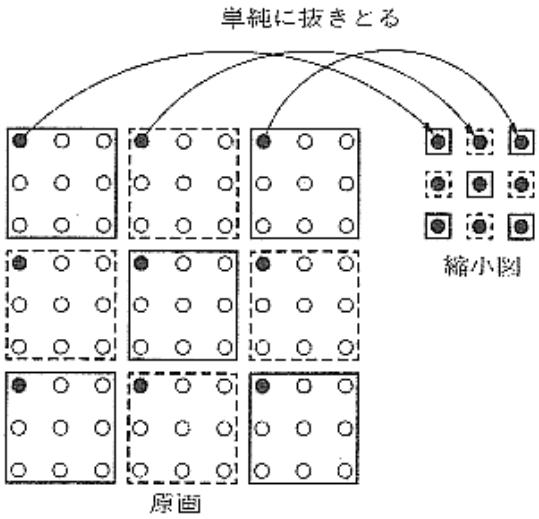
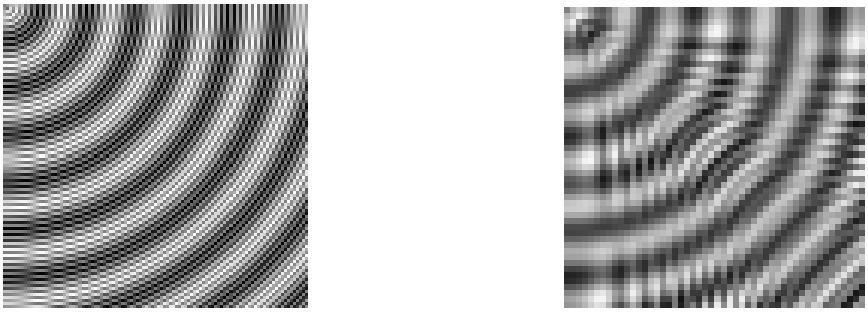


Fig 1.2 ダウン・サンプリング法 (L=M=3)

Fig 1.3 は、ダウン・サンプリング法により縮小した例である。(a)の縞模様に比べ、(b)では元の縞模様に加え別の模様(モアレ縞)が現れているのが分かる。これは、画素数を単純に低減したことにより、高周波数成分が折り返した(エリアシングが生じた)結果である。ダウン・サンプリング法は、単純な方法であるが、このようなエリアシングの発生を回避することはできない。



(a)元画像　　(b)元画像を 1/2 に縮小したものを拡大

Fig 1.3 画像の縮小における問題点

これは縮小を行う前にあるフィルタを通すことによって避けることが出来るが、それは次の「2. 周波数特性」で述べることにする。

MATLAB では、画像の拡大･縮小に’imresize’という関数を使用する。

$$X_{new} = imresize(X_{old}, SCALE, 'bilinear', argument)$$

ただし、X_{new}：拡大･縮小後の画像　　X_{old}：元画像

SCALE：縮尺比　　Argument(引数)：今回は 0 を与えておく

また、任意のピクセルサイズに拡大･縮小するには次のようにする。

$$X_{new} = imresize(X_{old}, [\text{Vertical size}, \text{Side size}])$$

ただし、Vertical size：縦のピクセル数　　Side size：横のピクセル数

練習問題 1

サンプル画像を MATLAB で取り込み、任意の大きさに拡大･縮小し、表示させるプログラムを MATLAB の M ファイルで作成せよ。また、縮小した画像をウィンドウ上で拡大し、元の画像と比較せよ。

2. 画像の周波数特性

2. 1 ディジタル信号処理システムと信号

Fig 2.1 は、アナログ入出力ディジタル新語処理システムである。システムによっては、直接にディジタル信号をインターフェースする場合もある。いずれにしても、ディジタル信号処理システムで扱う信号は、もともとアナログ信号である場合が多い。

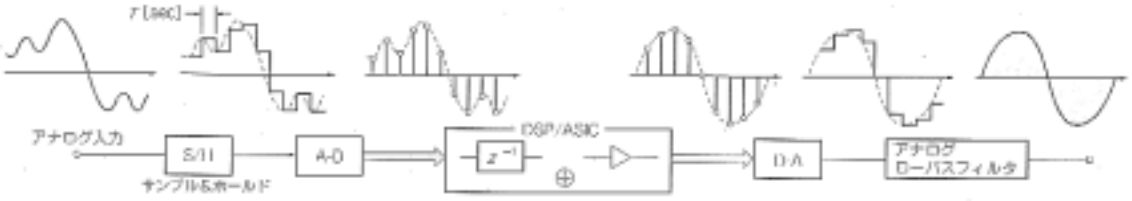


Fig 2.1 アナログ入出力ディジタル信号処理システム

処理対象のアナログ信号は、まずサンプル&ホールド回路と A-D コンバータで T[sec]ごとにその電圧値が2進数に変換される。

このように量子化されたディジタル信号は、DSP(ディジタル・シグナル・プロセッサ)や ASIC で処理内容別にアルゴリズムに基づいて処理される。ここがいわゆる「信号処理」と呼ばれる部分であり、ディジタルフィルタ、高速フーリエ変換(FFT)や周波数シフトなどがその代表的なアルゴリズムである。

2. 2 離散時間信号とは

ディジタル信号処理で扱う信号 $x(nT)$ は、Fig 2.2 に示すようにアナログである連続信号 $x(t)$ をサンプル間隔 T(sec)でサンプリングした信号である。

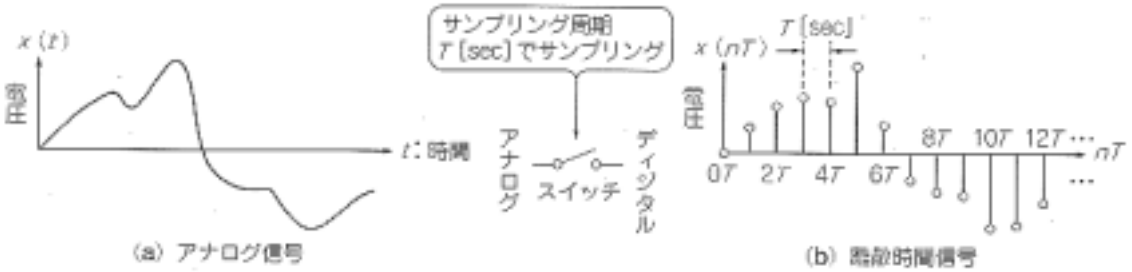


Fig 2.2 連続時間信号と離散時間信号

$x(nT)$ は離散時間信号あるいはディジタル信号と呼ばれ、$x(t)$ の時刻 $t = nT, n = 0,1,2\mathrm{L}$ における電圧値である。また、$f_s = 1/T$ をサンプリング周波数と呼ん

でいる。

実際のシステムでは、Fig 2.1 のようにサンプル&ホールド回路によりアナログ信号をサンプリングして $x(nT)$ を得ている。さらに A-D コンバータにより、$x(nT)$ の電圧値を 2 進数に量子化する。

2．3 フーリエ変換

フーリエ変換(Fourier Transform)は、信号の周波数領域における情報を与える数学的手法である。まず、連続時間信号のフーリエ変換から述べる。

(1) 連続時間信号のフーリエ変換と逆変換

アナログ信号すなわち連続時間信号 $x_a(t)$ のフーリエ変換 $X_a(j\Omega)$ は、以下のように定義されている。

$$X_a(j\Omega) = \int_{-\infty}^{\infty} x_a(t) e^{j\Omega t} dt \tag{2.1}$$

ただし、t:時間、Ω=2πf、f:周波数

フーリエ変換 $X_a(j\Omega)$ は周波数スペクトルと呼ばれており、信号 $x(t)$ の周波数 Ωにおける振幅スペクトル $|X_a(j\Omega)|$、位相スペクトル $\angle X_a(j\Omega)$、パワースペクトル $|X_a(j\Omega)|^2$ などの情報を与える。

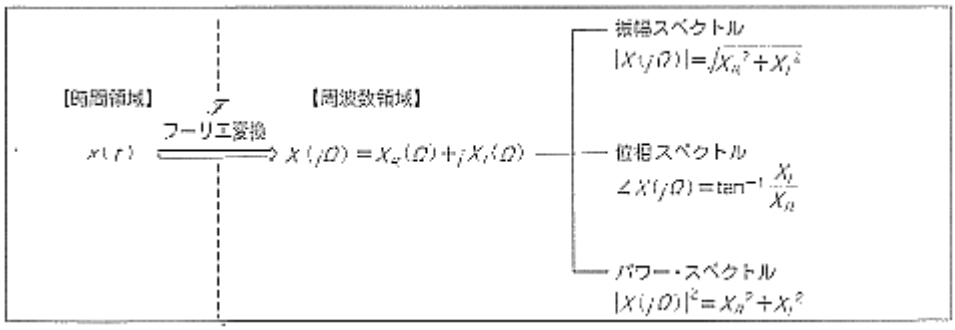


Fig 2.3 フーリエ変換から得られる情報

$x(t)$ のフーリエ変換 $X_a(j\Omega)$ を、次のように表すことにする。

$$X_a(j\Omega) = f\,|\,x(t)\,| \quad \text{または} \quad x(t) \overset{f}{\leftrightarrow} X(j\Omega) \tag{2.2}$$

一方、$X_a(j\Omega)$ から逆に $x(t)$ を求める算法は逆フーリエ変換と呼ばれ、次式で与えら

れる。

$$x_a(t) = \frac{1}{-2\pi}\int_{-\infty}^{\infty} X_a(j\Omega)e^{j\Omega t}d\Omega \tag{2.3}$$

(2) 離散時間信号のフーリエ変換

離散時間信号 $x(nT)$ のフーリエ変換 $X(j\omega)$ は、離散時間フーリエ変換と呼ばれ、以下のように定義されている。

$$X(j\omega) = \sum_{n=-\infty}^{\infty} x(nT)e^{-j\omega_n T} \tag{2.4}$$

一方、$X(j\omega)$ から逆に $x(nT)$ を求める算法は逆フーリエ変換は次式で与えられる。(2.4)式の導出については付録に記載する。

$$x(nT) = \frac{1}{\omega_s}\int_{-\omega_s/2}^{\omega_s/2} X(j\omega)e^{j\omega_n T}d\omega \tag{2.5}$$

$X(j\omega)$ は、周波数軸上でサンプリング周波数 ω_s ごとにそのスペクトルが繰り返される。

つまり、$X(j\omega)$ は ω_s を周期とする周期関数である。周期関数はフーリエ級数で表すことが可能であるため、(2.5)の逆変換が求められるのである。

(3) 2 次元フーリエ変換

2 次元フーリエ変換(Two-dimensional Fourier transform)は、以下のように定義されている。

$$X(\omega_1,\omega_2) = \sum_{n_1=-\infty}^{\infty}\sum_{n_2=-\infty}^{\infty} x(n_1,n_2)e^{-j\omega_1 n_1}e^{-j\omega_2 n_2} \tag{2.6}$$

その逆変換は

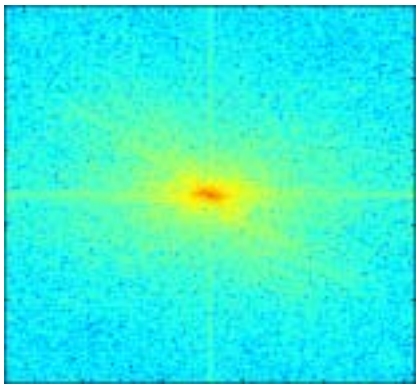
$$x(n_1,n_2) = \frac{1}{(2\pi)^2}\int_{\omega_1=-\pi}^{\pi}\int_{\omega_2=-\pi}^{\pi} X(\omega_1,\omega_2)e^{j\omega_1 n_1}e^{j\omega_2 n_2} \tag{2.7}$$

である。ここで、ω_1、ω_2 はそれぞれ n_1、n_2 軸の空間周波数(spatial frequency)といい、2 次元フーリエ変換により離散空間信号が連続周波数領域の周波数スペクトルに変換される。

実際には、主に処理を高速化した高速フーリエ変換(FFT : Fast Fourier transform)が信号解析その他に使用されている。

では、実際に画像のフーリエ変換を行ってみよう。Fig 2.4 は標準画像 lenna とその 2 次元高速フーリエ変換(2DFFT)を行ったものである。

(a) 元画像



(b) パワースペクトル

Fig 2.4 2 次元フーリエ変換

パワースペクトルは中央部分が DC(直流)成分で、端の方へ行くほど周波数が高くなり、その強度は青→緑→黄色→赤の順に増加する。(b)では、DC 成分がもっとも強く、また左上から右下にかけてスペクトルの成分が出ている(緑色)ことがわかる。

MATLAB では、画像の 2 次元高速フーリエ変換を行うにはまず画像をグレースケール化し、その後に 2 次元フーリエ変換を行う。画像のグレースケール化には'rgb2gray'という関数を使用する。

$$X_{gray} = \mathrm{rgb2gray}(X_{old})$$

ただし、X_{gray}：グレースケール化した画像　　X_{old}：元画像

次に、画像の 2 次元高速フーリエ変換には'fft2'という関数を使用する。

$$X_{2DFFT} = \mathrm{fft2}(X_{gray})$$

ただし、X_{2DFFT}：パワースペクトル

X_{gray}：グレースケール化した画像

Fig 2.4 の(b)のように DC 成分を中央に持ってくるため、’fftshift’と言う関数を用いる。

$$X_{shift} = \mathrm{fftshift}(X_{2DFFT})$$

ただし、X_{shift}：シフト後のパワースペクトル　　X_{2DFFT}：パワースペクトル

最後に、パワースペクトルの表示は以下のように行う。

$$\mathrm{colormap(jet(64)),imagesc(log(abs(}X_{shift}\mathrm{)));colorbar}$$

ただし、X_{shift}：シフト後のパワースペクトル

練習問題 2

サンプル画像を MATLAB で取り込み、2 次元高速フーリエ変換の後パワースペクトルを表示するプログラムを MATLAB の M ファイルで作成せよ。また、正方形の領域に各自で模様を描き、そのパワースペクトルを観察・比較せよ。

練習問題 3

vcapg で取り込んだ画像を bitmap で保存し、その画像を 2 次元高速フーリエ変換の後パワースペクトルを表示するプログラムを MATLAB の M ファイルで作成せよ。

2．3　アップサンプリングとダウンサンプリング

1．1及び1．2で説明したアップサンプリング及びダウンサンプリングを、周波数領域において考えてみる。アップサンプリング、ダウンサンプリングの 1 次元におけるイメージを Fig 2.5、Fig 2.6 に示す。

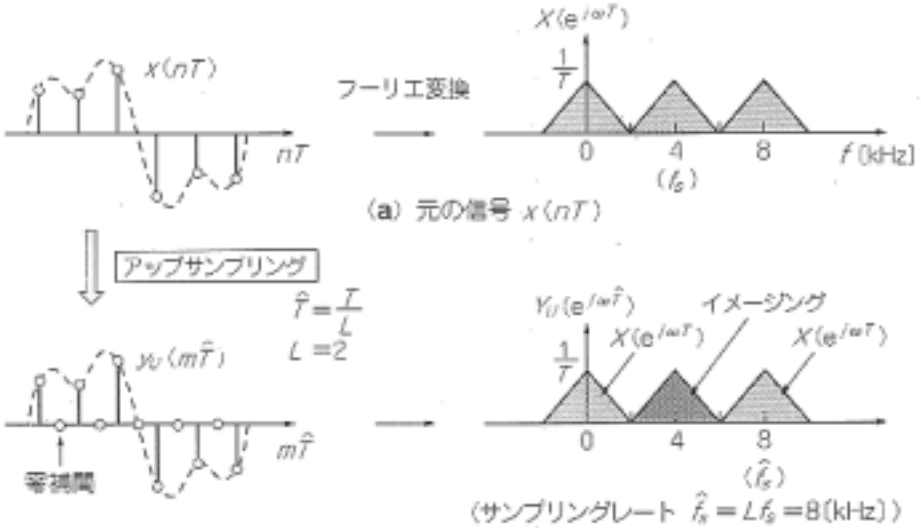


Fig 2.5 アップサンプリング (L=2)

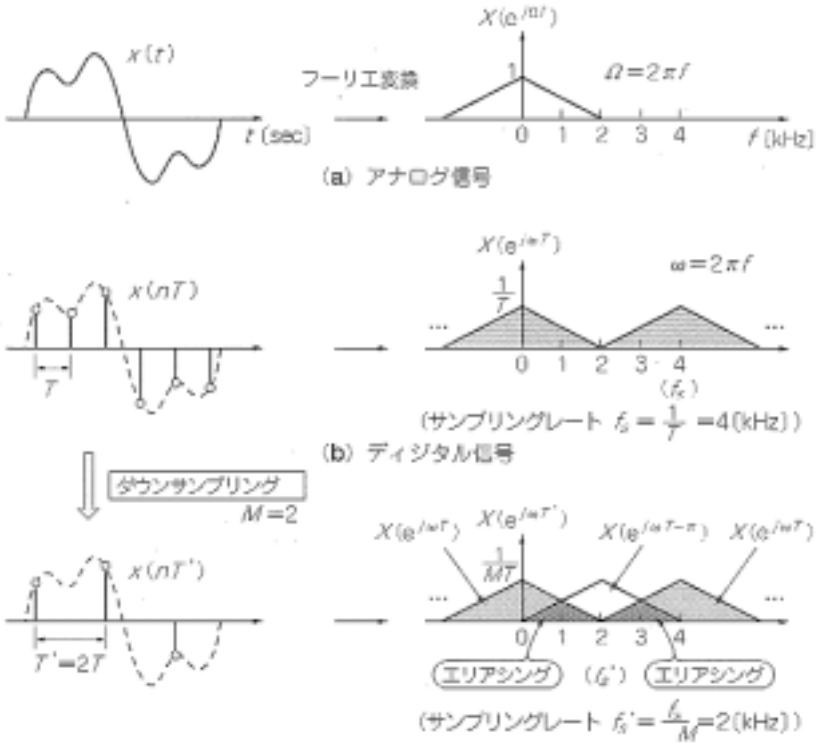


Fig 2.6 ダウンサンプリング (M=2)

単純に信号を増やしたり間引いたりするだけではイメージングやエリアシングが生じてしまう。特にエリアシングは Fig 1.3 (b)のように画像に悪影響を及ぼすためダウンサンプリングの前にローパスフィルタ(LPF)を通して高周波成分のカットを行う(Fig 2.7)。

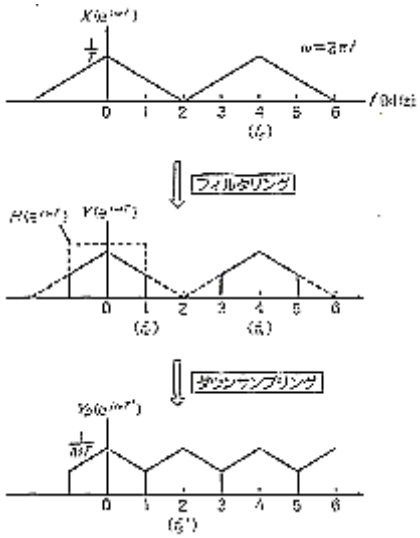


Fig 2.7 ローパスフィルタによる高周波成分のカット及びダウンサンプリング

MATLAB では画像の縮小の場合、'imresize'の引数の 0 を省略することによってローパスフィルタを通すことができる。ただし、画像の拡大においては各自でフィルタを設定し、適用しなければならない。

練習問題 4

サンプル画像を MATLAB で取り込み、拡大/縮小の後 2 次元高速フーリエ変換を行い、そのパワースペクトルを表示するプログラムを MATLAB の M ファイルで作成せよ。

練習問題 5

練習問題 4 において、画像にローパスフィルタを適用し、2 次元高速フーリエ変換を行うプログラムを作成し、フィルタの適用前と適用後の画像を比較せよ。