

5

画像の周波数領域処理

5.1 空間周波数

画像の周波数について考えてみる。

実行例 5.1

```
>> I3=imread('lighthouse.png');  
>> G3=rgb2gray(I3);  
>> figure; plot(G3(160,:))  
>> hold on  
>> plot(G3(400,:))  
>> plot(G3(600,:))
```

`rgb2gray` は、RGB 画像をグレースケール画像に変換する関数である。このグレースケール画像の横方向の画素値の変化を縦軸を画素値、横軸を x 座標として、 $y = 160, 400, 600$ のところをそれぞれ、青、赤、黄色でプロットしている。このグラフを見ると、空 ($y = 400$ の大部分を占める) や灯台の塔の部分は、値の変化が小さい。また、領域のへり (エッジとよぶ) の部分では、画素値が急激に大きく変化する (塔と空の境の部分など)。芝生の部分は値が細かく変化する。

これらの信号の空間周波数スペクトルは次のようになる。

実行例 5.2

```
>> plot(log(abs(fft(G3(160,:))))))
>> hold on
>> plot(log(abs(fft(G3(400,:))))))
>> plot(log(abs(fft(G3(600,:))))))
```

このグラフの横軸は周波数である。時間の変化ではなく、空間的な変化なので、空間周波数とよぶ。今回の場合、画像の横幅が 480 点なので、周期が 480 点の場合、1 となる。このスペクトルでは、直流成分が一番大きくなっている。ここで処理した画素値は正の値しか取らないため、信号の平均値 (平均的な明るさ) が直流成分となる。芝生のような細かい模様は、高い空間周波数に関係する。空や塔など大きな領域は、低い空間周波数に関係する。このような側面から見ると、画像もフーリエ変換を用いて様々に処理できる。

5.2 2次元フーリエ変換

画像は 2 次元の信号なので、空間周波数は 2 次元となる。実際の多くの画像では、多くの周波数成分を含んで複雑なので、まず、単純な場合で空間周波数を考える。次の画像の 2 次元の空間周波数を計算すると次のようになる。



図 5.1 垂直周波数を持つ画像

プログラム 5-1 (2次元のスペクトル)

```
1 >> fs=64;
2 >> x=0:1/fs:1;
3 >> x=x(1:64)';
4 >> G= repmat(uint8(100*(sin(2*pi*5*x)+1)),1,64);
5 >> imtool(G,'InitialMagnification','fit');
```

```
6 >> subplot(3,1,1); plot(x,G(:,1))
7 >> subplot(3,1,2); stem(abs(fft(G(:,1))))
8 >> S=fft(G);
9 >> subplot(3,1,3); stem(abs(S(:,1)))
10 >> S2=fft(S.').';
11 >> figure; stem3(abs(S2))
```

`stem`、`stem3`などはMATLABの関数である。4行目で、列方向の正弦波を生成し、それを行方向に繰り返すことで図5.1を生成している。つまり、この画像は、全ての列は同じ内容となる2次元配列であり、正弦波の周期にあわせて明るさが変化する。7行目は正弦波のスペクトルを描画しており、縦方向に5周期分の成分が確認できる(全体を1秒と考えれば5Hzの位置に対応する)。この正弦波の周期は $64/5 = 12.8$ 画素である。8行目は2次元配列に対して`fft`を計算している。この場合、第1次元の方向にフーリエ変換を行う。つまり、列方向にフーリエ変換を行い、それを行方向全体に適用することを意味する。つまり、正弦波をフーリエ変換した結果がSの列ベクトルとして保存されている(9行目)。10行目では、Sを転置(`.'`)してフーリエ変換し、さらに転置した結果をS2に保存している。つまり、行列の行方向にフーリエ変換を行い、それを列方向に繰り返すことを意味する。この画像の場合は、最初の列方向のフーリエ変換が全ての列で同一となるため、行方向には、値が一定なので直流成分しか存在しない。このように、(1次元)fftをまず縦に行い、次に横にfftを行うことで2次元のスペクトルを求める。この処理を2次元フーリエ変換とよぶ。

MATLABでは、2次元フーリエ変換は`fft2`を用いて計算する。

プログラム 5-2 (fft2 使用例)

```
1 >> Z2=fft2(G);
2 >> figure; stem3(abs(Z2))
3 >> figure; stem3(abs(fftshift(Z2)))
```

プログラム5-1では、音声信号のときと同様に直流成分が端に位置するようにプロットした。しかし、画像の場合は、通常、直流成分が中央に来るように

プロットする。3行目の `fftshift` を用いると直流成分が中央になるように位置をずらしている。

ただし、中央といっても、この場合は、サイズが偶数なので、注意を要する。

実行例 5.3

```
>> Z2shift=fftshift(Z2);  
>> stem(Z2shift(:,33))
```

このグラフからもわかるように、`Z2shift(33,33)` が中心である。この要素は、この画像の直流成分を表す。

5.3 周波数領域でのフィルタ処理

時間領域のフィルタ処理は、周波数領域ではかけ算となる。したがって、時間領域で畳み込みを用いて計算した LPF など、周波数領域のかけ算として実現できる。

プログラム 5-3 (周波数領域の 1 次元 LPF)

```
1 fs=2000;  
2 t=0:1/fs:1;  
3 y=sin(2*pi*440*t)+sin(2*pi*660*t);  
4 fftlen=64;  
5 overlap=32;  
6 cutoff=500;  
7 cind=cutoff/(fs/fftlen);  
8 ylen=length(y);  
9 F=zeros(1,fftlen);  
10 F([1:cind+1 end-cind+1:end])=1;  
11 rind=1;  
12 ry=zeros(1,ylen);  
13 while rind <= ylen-fftlen  
14     r=rind:rind+fftlen-1;  
15     S=fft(y(r).*hann(fftlen)');  
16     ry(r)=ry(r)+ifft(S.*F);  
17     rind=rind+(fftlen-overlap);  
18 end
```

これは、`cutoff` で指定された遮断周波数以下の成分が残るように設計した LPF である。9 行目の `F` が周波数領域のフィルタ係数で、そのまま通す周波数成分に対しては 1 を設定し、通さない成分に対しては 0 を設定している。そのフィルタ係数を 16 行目で要素ごとの乗算 (`.*`) で適用している。

画像データに関しても同様の考え方でフィルタを実現できる。グレースケール画像に対する HPF の例を示す。

プログラム 5-4 (画像の HPF 処理)

```
1 I=imread('gantrycrane.png');
2 G=rgb2gray(I);
3 imtool(G,'InitialMagnification','fit')
4 [h,w]=size(G);
5 fftsize=2^nextpow2(max(size(G)));
6 F=ones(fftsize);
7 wlen=20;
8 ctr=fftsize/2+1;
9 F(ctr-wlen:ctr+wlen,ctr-wlen:ctr+wlen)=0;
10 figure; mesh(F)
11 Z=fftshift(fft2(G,fftsize,fftsize));
12 figure; mesh(log(abs(Z))); view(0,90)
13 fG=uint8(abs(ifft2(fftshift(Z.*F))));
14 imtool(fG(1:h,1:w),'InitialMagnification','fit')
15 imtool(fG(1:h,1:w)*4,'InitialMagnification','fit')
16 imtool(fG*32,'InitialMagnification','fit')
```

ここで使われている `mesh`、`nextpow2`、`ifft2`、`view` は MATLAB の関数である。`fft` はサイズを 2 のべき乗にした方が計算効率がよいので、4 行目で、サイズを元のサイズ以上で最も小さい 2 のべき乗の数を計算している。8 行目で HPF を設計している。9 行目のグラフを見ると、`F` は、中央部 (低周波数部分) が正方形の範囲で 0 となり、他の部分が 1 となっている。厳密には中央部を円形に処理すべきであるが、この例では簡易的に正方形で処理している。12 行目は 2 次元スペクトルを描画している。2 次元スペクトルのグラフは 3 次元になる。ここで描画されるグラフは、スペクトログラムと同様に 3 次元のグラフを上から見たもので、強い部分は明るく、弱い部分を暗くして表現している。低周波に対応する中央部分を除くと、強い (明るい) 成分は、中心を通る 4 本

の直線である。この成分は、水平、垂直、斜めの 2 本の計 4 本のエッジに対応している。13 行目で F をかけることでフィルタ処理を実現している。14 行目は画像を明るく表示するために、全画素を 4 倍している。元の画像と比較すると、どの部分が高周波成分かわかる。

また、ここでは、画像サイズより大きなサイズでフーリエ変換を行なっている。16 行目ではフーリエ逆変換で生成された画像全体を表示している。14,15 行目では必要なところだけを取り出している。

空間周波数領域の処理で特定の空間周波数を取り除くことも可能である。例えば、周期性のノイズが混入している画像を考える。画像では縞状のノイズに見えるが、このようなノイズはスペクトル上では、特定の位置に出現する。したがって、その部分を弱めるようなフィルタリングを行なえば除去できる。人工的に周期性のノイズを付加して、そのノイズを空間周波数領域でのフィルタリングで除去するプログラムを次に示す(ここで使われる hp は**章末問題【7】**で作成する)。

————— **プログラム 5-5** (周期性ノイズの除去) —————

```

1  G=imread('cameraman.tif');
2  [h,w]=size(G);
3  fftsize=h;
4  imtool(uint8(hp)+mean(mean(G)), 'InitialMagnification', 'fit')
5  x1=G+uint8(hp);
6  imtool(x1, 'InitialMagnification', 'fit')
7  Z=fftshift(fft2(x1,h,h));
8  figure; mesh(log(abs(Z)));
9  A=ones(h);
10 A(145,129)=0;
11 A(113,129)=0;
12 figure; mesh(log(abs(Z.*A)));
13 G2=uint8(abs(ifft2(fftshift(Z.*A))));
14 imtool(G2, 'InitialMagnification', 'fit')

```

hp は、**プログラム 5-1** を参考に 16 画素を周期とする垂直周波数を持ち、サイズが G と同じになる画像である。正弦波の振幅は 16 とせよ。5 行目では、この模様を元画像に加えて周期性ノイズを模擬している。正しく生成できれば次

のような画像となる。



図 5.2 周期性ノイズを付加した画像

このようなノイズが含まれている画像のスペクトルでは、対応する周波数成分にピークが現われる。10 行目のスペクトルを y 軸方向から見ると、ピークがゼロ周波数 (129,129) から 16 画素離れた (113,129) と (145,129) に観察できる (図 5.3)。

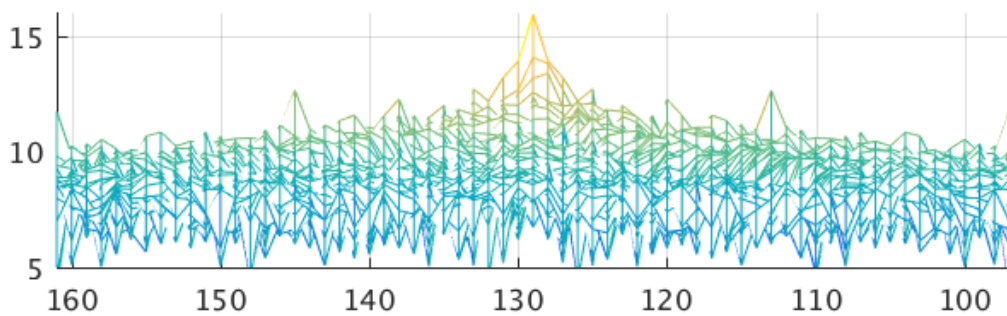


図 5.3 周期性ノイズを付加した画像のスペクトルの中心部

そこで、11 行目から 13 行目で、その周波数を 0 であるようなフィルタを作成し適用している。結果を見ると、ノイズがある程度軽減されていることがわかる。

5.4 周波数領域での画像の拡大

空間周波数領域では他にもいろいろな処理が可能である。ここでは、画像を2倍の寸法に拡大する処理を紹介する。

プログラム 5-6 (画像の拡大)

```
1 G=imread('cameraman.tif');
2 [h,~]=size(G);
3 fftsize=h;
4 Z=fft2(G,fftsize,fftsize);
5 ZZ=fftshift(Z);
6 targetsize=fftsize*2;
7 A=zeros(fftsize*2);
8 ctr=targetsize/2+1;
9 r=ctr-fftsize/2+1:ctr+fftsize/2;
10 A(r,r)=ZZ;
11 fG=ifft2(fftshift(A));
12 imtool(uint8(abs(fG)));
13 imtool(uint8(abs(fG))*4);
```

7行目で元の2倍のサイズの2次元配列Aを用意し、全ての要素を0で初期化している。10行目でこのAの中央部(つまり低周波部)に元の画像のスペクトルを代入している。11行目ではAを逆フーリエ変換でグレースケール画像に戻している。12行目の出力でわかるように、このように作成すると画像は暗くなる。そこで、13行目では4倍することで、明るさを元に戻している。

同様の処理で、音声ファイルも(1次元で)2倍に拡大することができる。

章 末 問 題

- 【1】 プログラム 5-1 の画像 G を転置して縦縞の画像を作成し、2次元フーリエ変換を行え。結果のどの部分が何を表しているかを説明せよ。
- 【2】 プログラム 5-4 の `wlen` を2種類の別の値に変更して実行せよ。それらの結果を比較して画像の2次元周波数に関して考察せよ。
- 【3】 プログラム 5-4 を参考に画像のLPFのプログラムを作成せよ。その上で、2

枚以上の画像を探し、その画像に対して、いくつかのパラメータで実行せよ。それらの結果から、2次元周波数の分布について考察せよ。

- 【4】 **プログラム 4-1** の s の高い方の成分が残るような HPF を周波数領域で実装せよ。
- 【5】 3種類の正弦波を加えた音を作成せよ。その上で、真ん中の高さの正弦波だけを取り除くフィルタ (BPF) を周波数領域で実装せよ。
- 【6】 画像の BPF のプログラムを作成し、BPF の効果を説明するのに適している画像を探し実行せよ。
- 【7】 **プログラム 5-5** に使われる周期的な模様 hp を、**プログラム 5-1** を参考に作成せよ。ただし、サイズは、`cameraman.tif` と同じで、正弦波は、16 画素を周期とする垂直周波数を持ち、振幅は 16 となるようにせよ。
- 【8】 画像ファイル `stripe.png`[†] の周期性ノイズを除去せよ。
- 【9】 適当なグレースケール画像を元の画像より暗くせよ。
- 【10】 **プログラム 5-6** の 9 行目から 10 行目を **プログラム 5-7** で置き変える。このプログラムは何倍に拡大されるか？ このプログラムの結果を **プログラム 5-6** の結果と比較して考察せよ。

————— **プログラム 5-7** (変更部分) —————

```

r=ctr-fftsize/4+1:ctr+fftsize/4;
ctr2=fftsize/2+1;
r2=ctr2-fftsize/4+1:ctr2+fftsize/4;
A(r,r)=ZZ(r2,r2);
    
```

- 【11】 **プログラム 5-6** を元の 4 倍に拡大されるように変更し、**プログラム 5-6** とは異なる画像を拡大してみよ。
- 【12】 **プログラム 5-6** を参考に、音声を周波数領域で 2 倍に拡大するプログラムを作成せよ。このプログラムの効果を説明するのに適当な音を選んでこのプログラムがどのような処理を行なっているかを説明せよ。

[†] 画像ファイルは授業支援システムからダウンロードしておくこと。