画像認識

多重解像度･多重方向解析

佐藤　嘉伸

yoshi@image.med.osaka-u.ac.jp

http://www.image.med.osaka-u.ac.jp/member/yoshi/

→ 日本語ページ → 授業の資料 → 画像認識

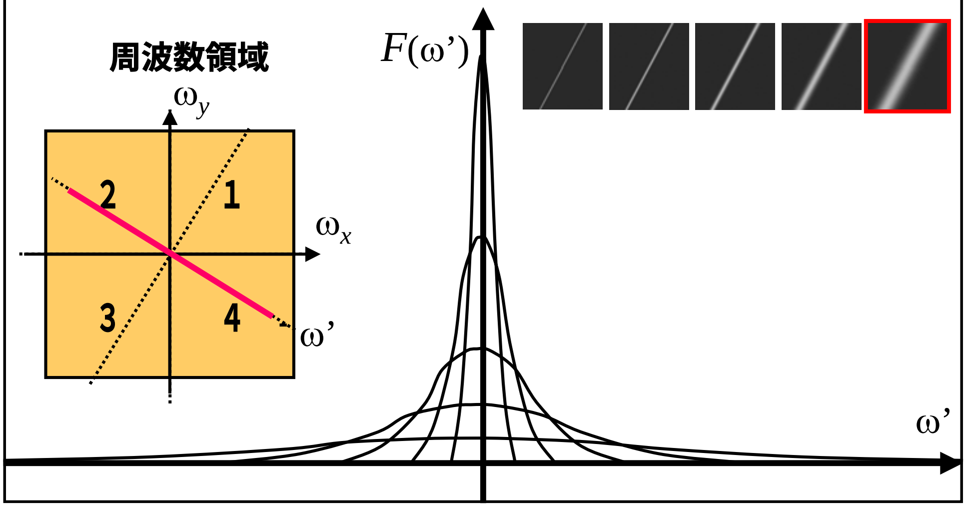
http://www.image.med.osaka-u.ac.jp/member/yoshi/lecture.html

→ 画像認識

多重解像度解析

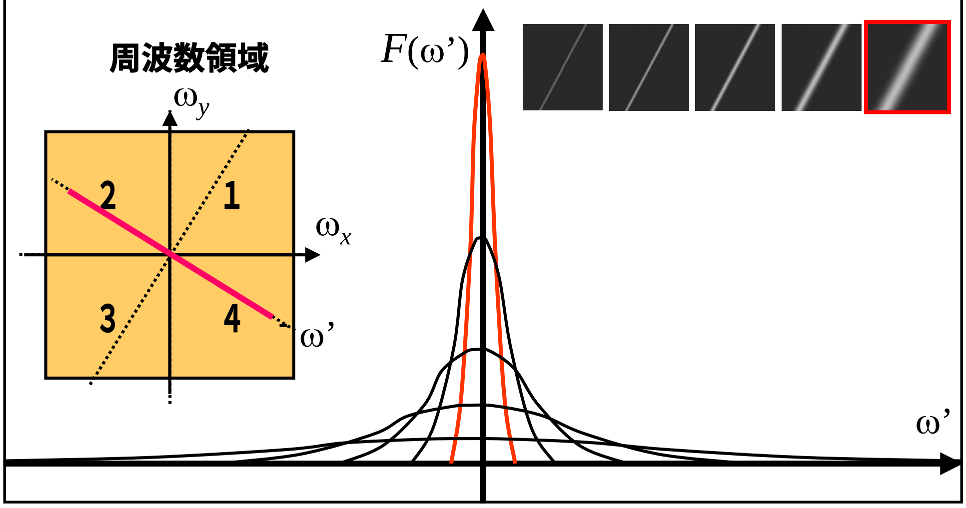
多重解像度解析

特定の解像度（周波数の範囲）のみに反応するような画像解析を行う。
複数の解像度に対してこの処理を行うことを、多重解像度解析と呼ぶ。



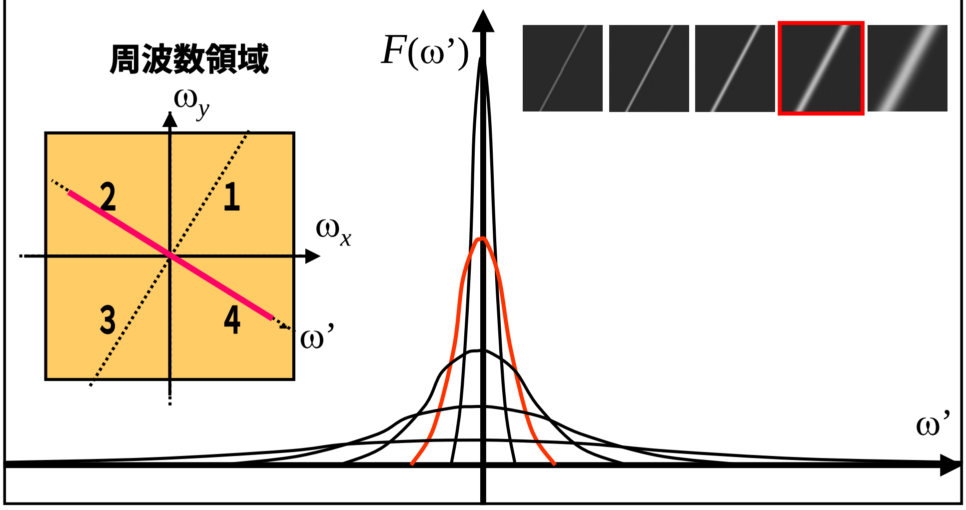
多重解像度解析

特定の解像度（周波数の範囲）のみに反応するような画像解析を行う。
複数の解像度に対してこの処理を行うことを、多重解像度解析と呼ぶ。



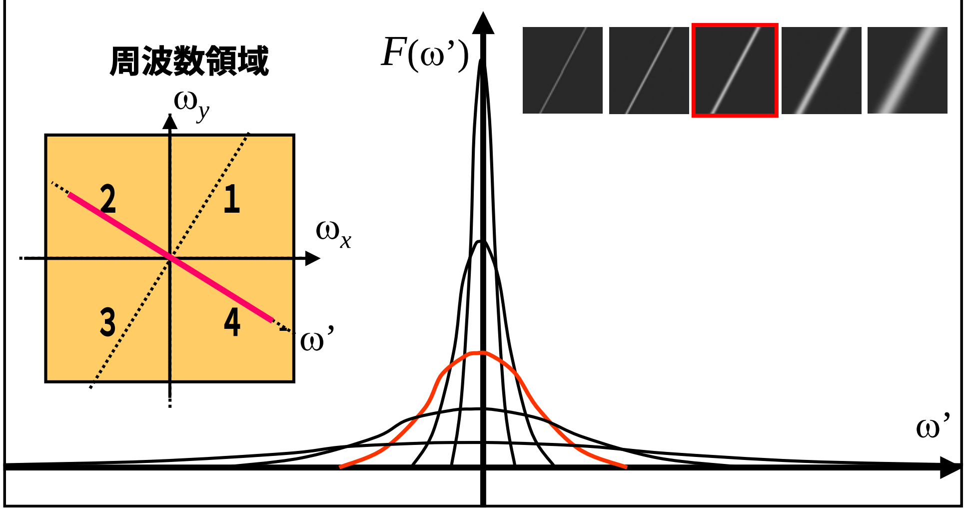
多重解像度解析

特定の解像度（周波数の範囲）のみに反応するような画像解析を行う。
複数の解像度に対してこの処理を行うことを、多重解像度解析と呼ぶ。



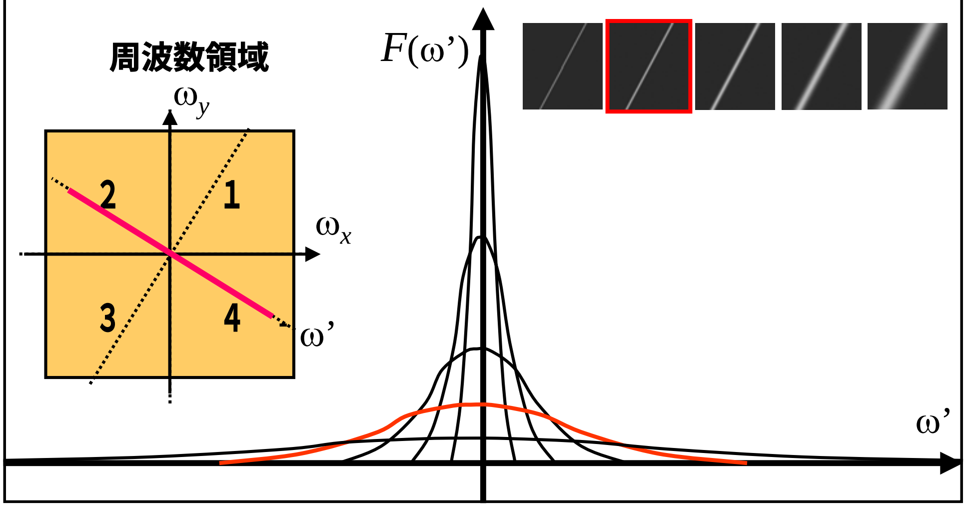
多重解像度解析

特定の解像度（周波数の範囲）のみに反応するような画像解析を行う。
複数の解像度に対してこの処理を行うことを、多重解像度解析と呼ぶ。



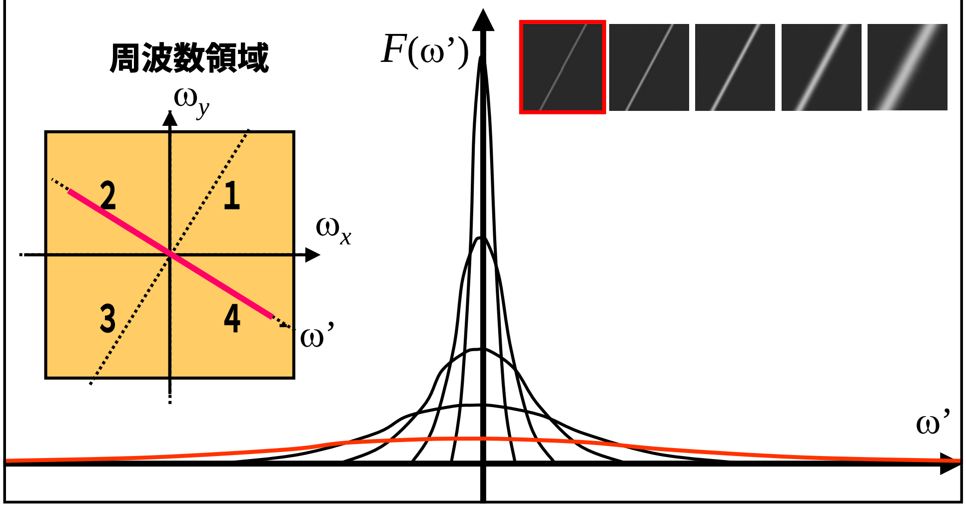
多重解像度解析

特定の解像度（周波数の範囲）のみに反応するような画像解析を行う。複数の解像度に対してこの処理を行うことを、多重解像度解析と呼ぶ。



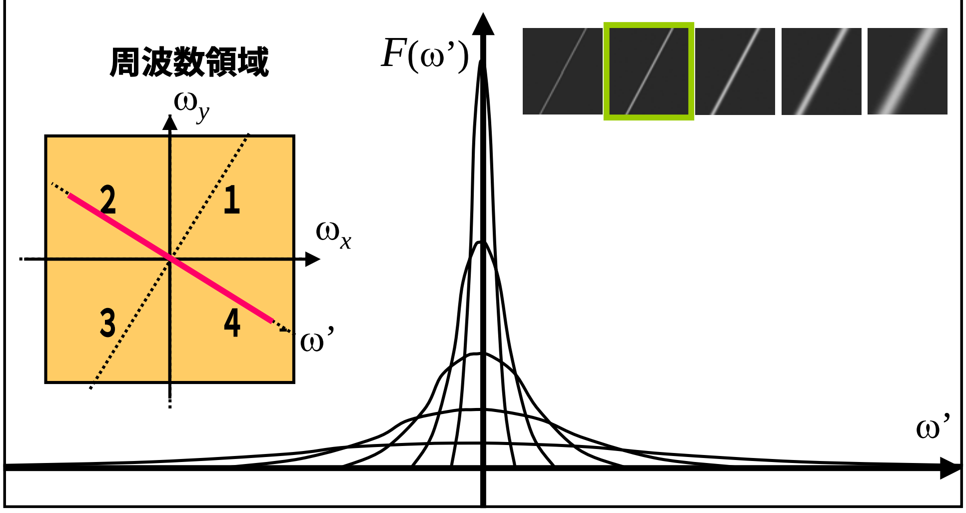
多重解像度解析

特定の解像度（周波数の範囲）のみに反応するような画像解析を行う。複数の解像度に対してこの処理を行うことを、多重解像度解析と呼ぶ。



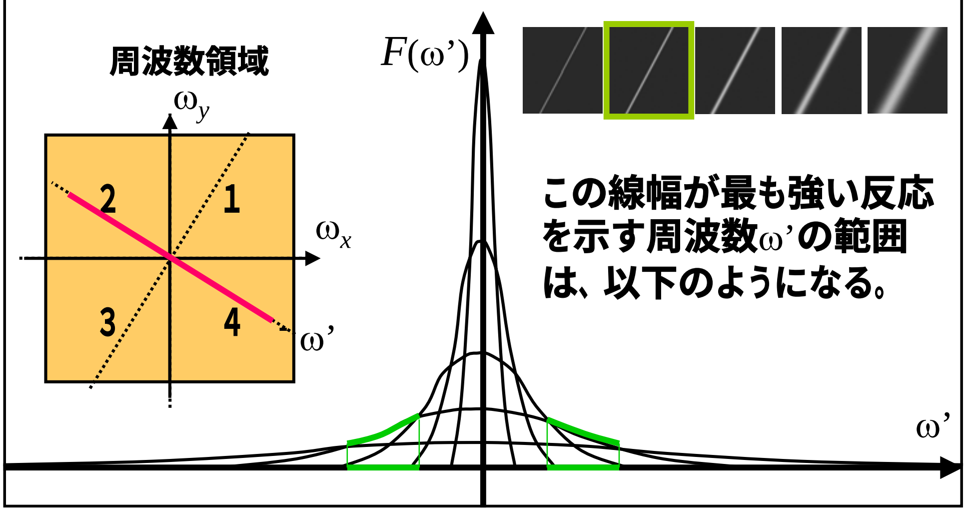
多重解像度解析

特定の解像度（周波数の範囲）のみに反応するような画像解析を行う。複数の解像度に対してこの処理を行うことを、多重解像度解析と呼ぶ。



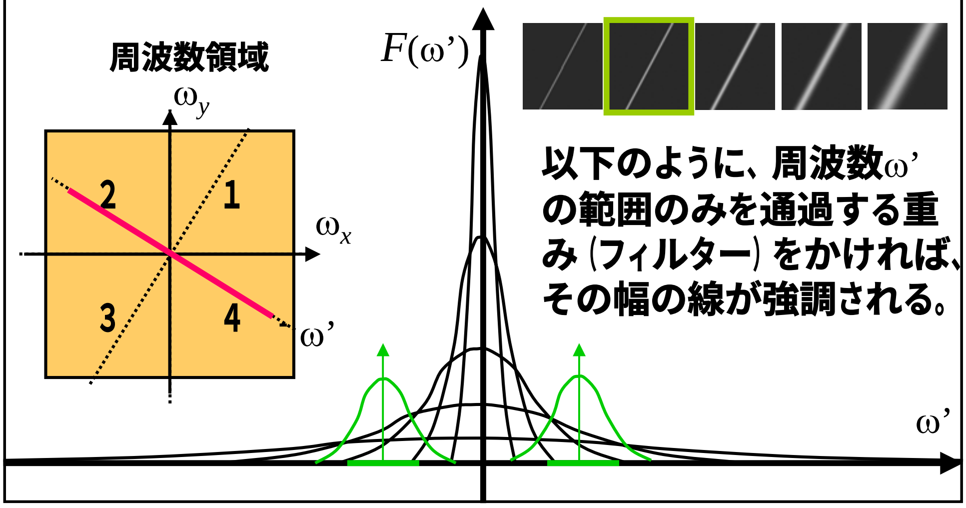
多重解像度解析

特定の解像度（周波数の範囲）のみに反応するような画像解析を行う。複数の解像度に対してこの処理を行うことを、多重解像度解析と呼ぶ。



多重解像度解析

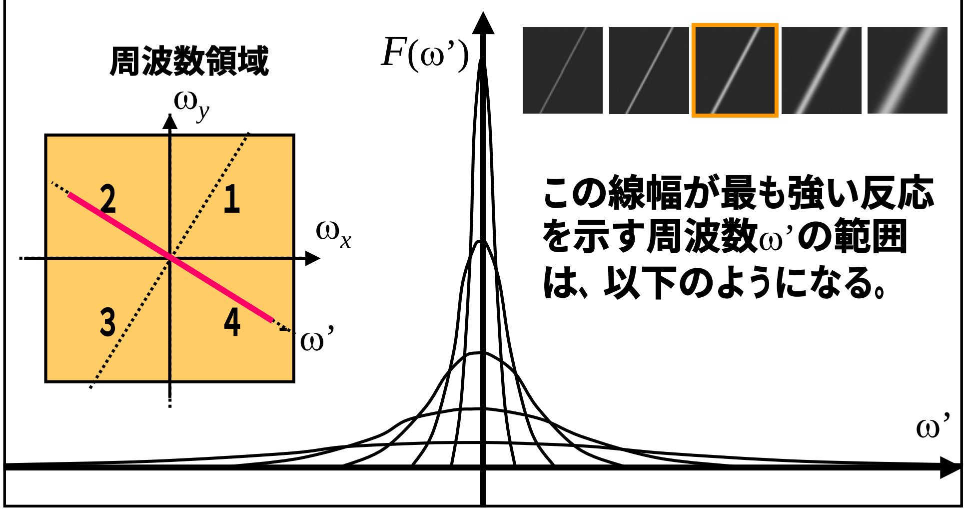
特定の解像度（周波数の範囲）のみに反応するような画像解析を行う。複数の解像度に対してこの処理を行うことを、多重解像度解析と呼ぶ。



以下のように、周波数ω'の範囲のみを通過する重み（フィルター）をかければ、その幅の線が強調される。

多重解像度解析

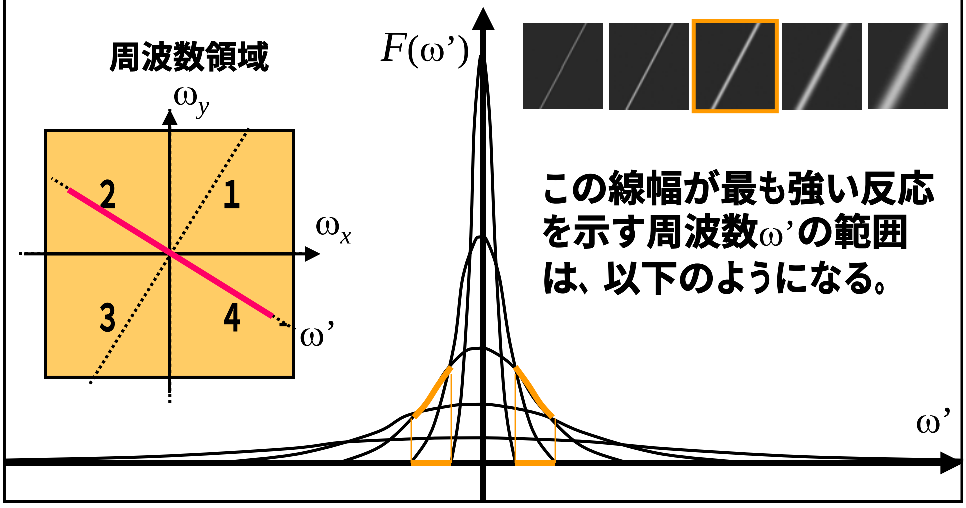
特定の解像度（周波数の範囲）のみに反応するような画像解析を行う。複数の解像度に対してこの処理を行うことを、多重解像度解析と呼ぶ。



この線幅が最も強い反応を示す周波数ω'の範囲は、以下のようになる。

多重解像度解析

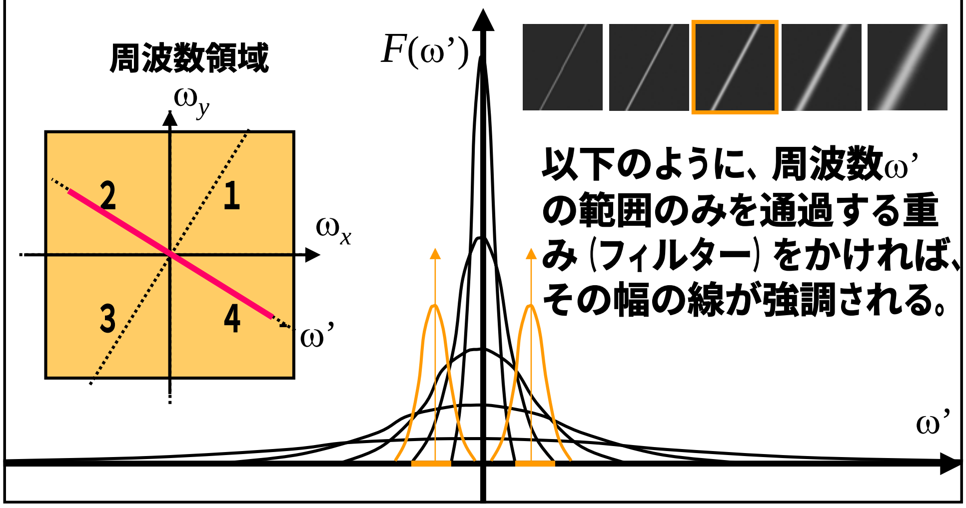
**特定の解像度（周波数の範囲）のみに反応するような画像解析を行う。
複数の解像度に対してこの処理を行うことを、多重解像度解析と呼ぶ。**



この線幅が最も強い反応を示す周波数ω'の範囲は、以下のようになる。

多重解像度解析

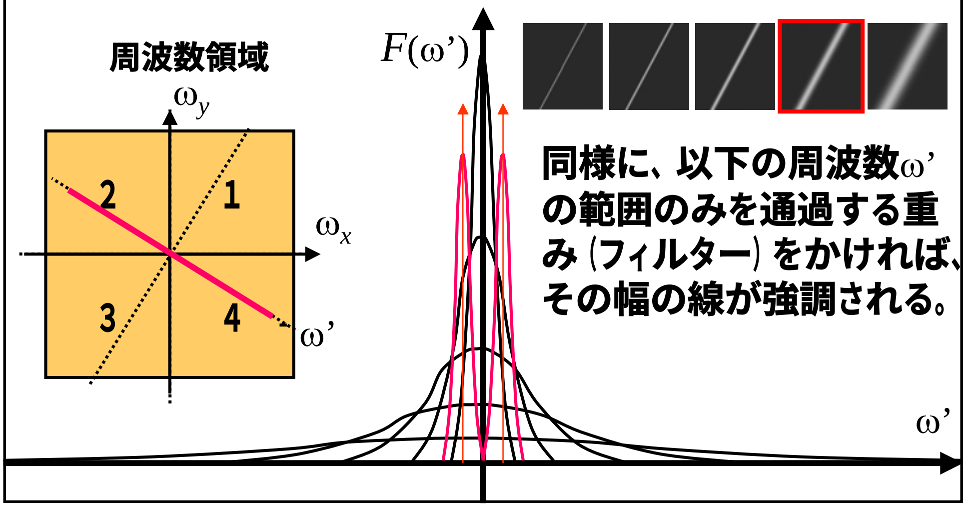
**特定の解像度（周波数の範囲）のみに反応するような画像解析を行う。
複数の解像度に対してこの処理を行うことを、多重解像度解析と呼ぶ。**



以下のように、周波数ω'の範囲のみを通過する重み（フィルター）をかければ、その幅の線が強調される。

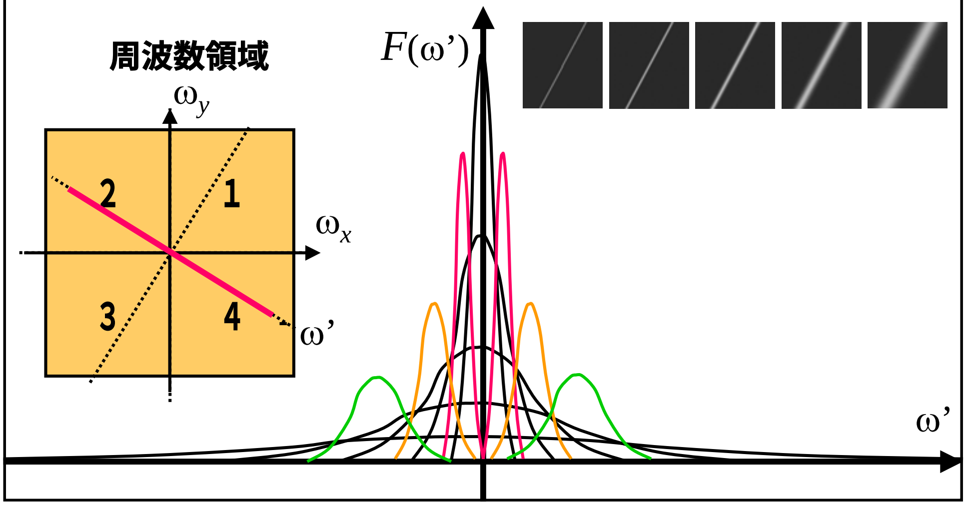
多重解像度解析

特定の解像度（周波数の範囲）のみに反応するような画像解析を行う。
複数の解像度に対してこの処理を行うことを、多重解像度解析と呼ぶ。



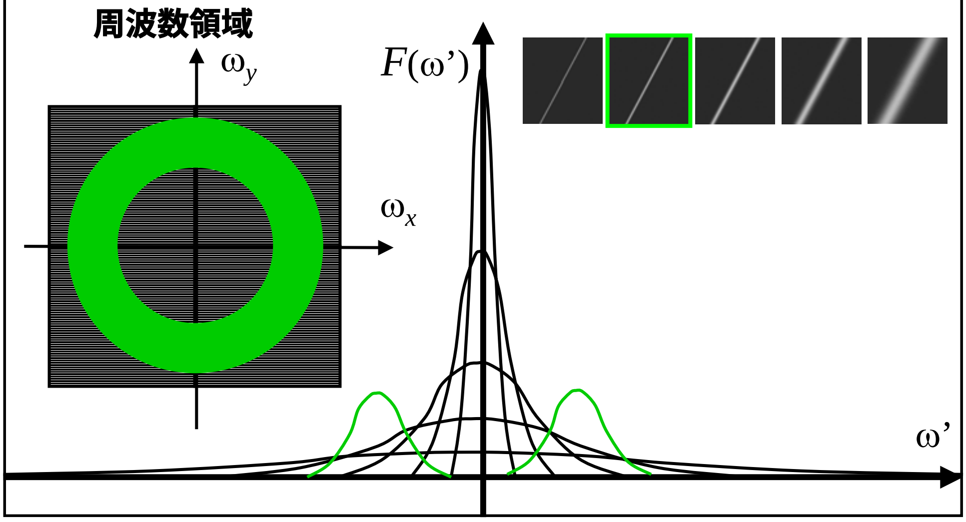
多重解像度解析

特定の解像度（周波数の範囲）のみに反応するような画像解析を行う。
複数の解像度に対してこの処理を行うことを、多重解像度解析と呼ぶ。



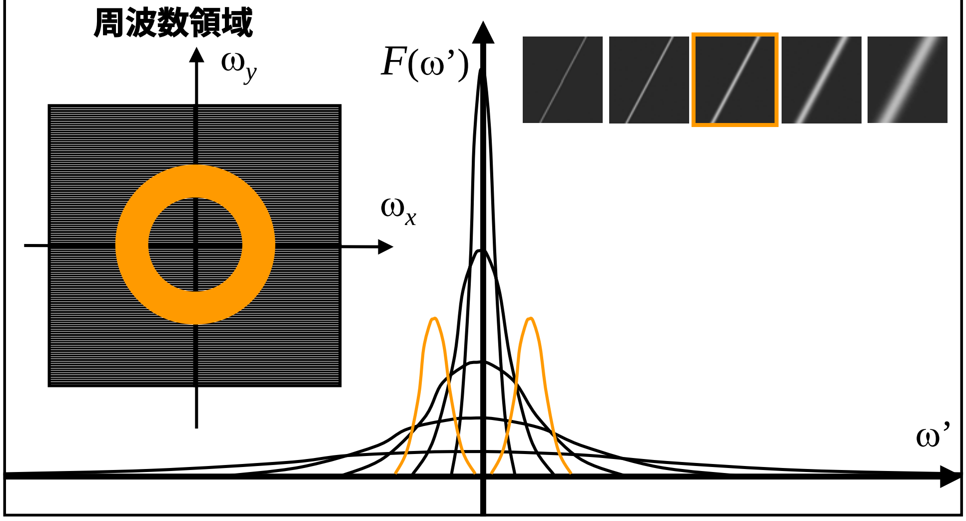
多重解像度解析：2次元画像

特定の解像度（周波数の範囲）のみに反応するような画像解析を行う。
複数の解像度に対してこの処理を行うことを、多重解像度解析と呼ぶ。



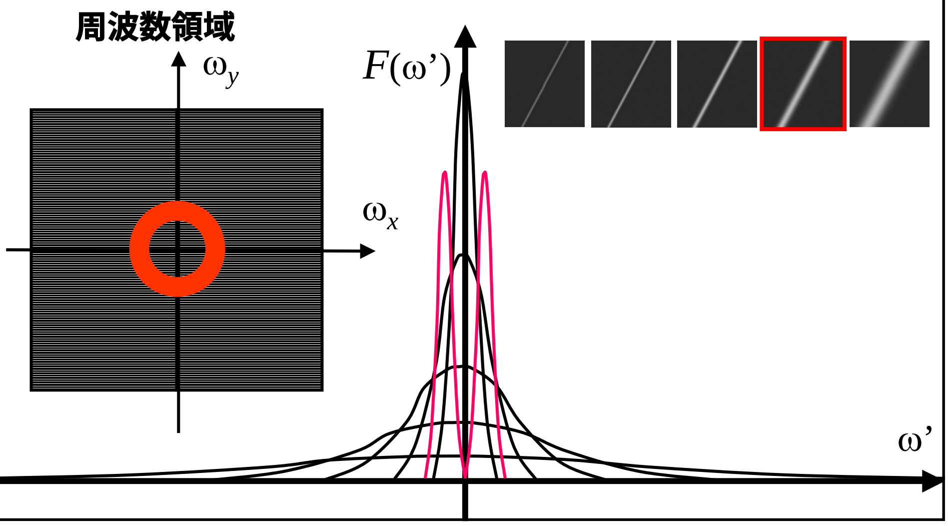
多重解像度解析：2次元画像

特定の解像度（周波数の範囲）のみに反応するような画像解析を行う。
複数の解像度に対してこの処理を行うことを、多重解像度解析と呼ぶ。



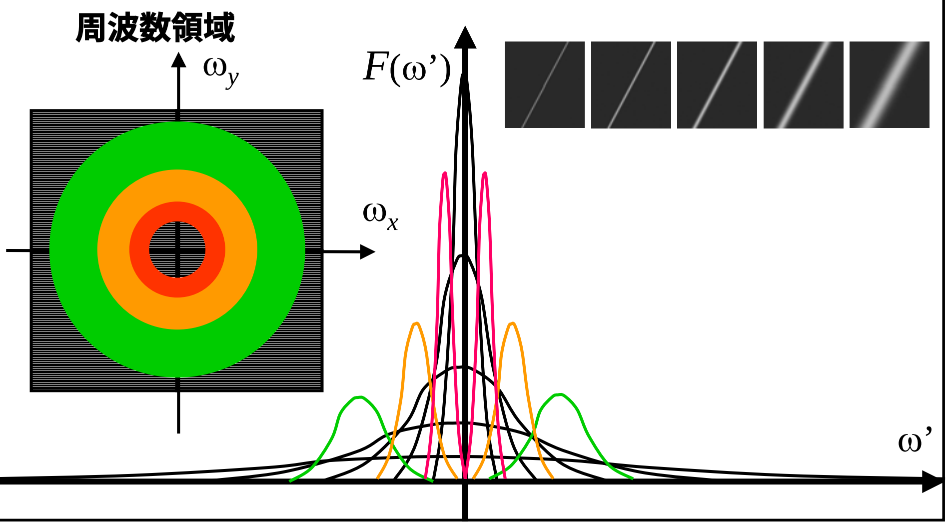
多重解像度解析：2次元画像

特定の解像度（周波数の範囲）のみに反応するような画像解析を行う。
複数の解像度に対してこの処理を行うことを、多重解像度解析と呼ぶ。



多重解像度解析：2次元画像

特定の解像度（周波数の範囲）のみに反応するような画像解析を行う。
複数の解像度に対してこの処理を行うことを、多重解像度解析と呼ぶ。

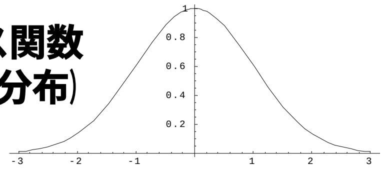


Gabor (ガボール) 関数
Gabor (ガボア) 関数

1次元Gabor (ガボール) 関数

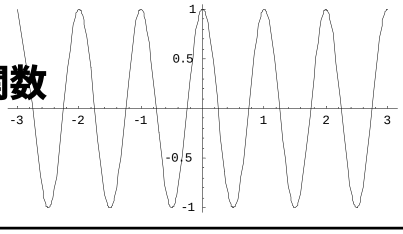
- ガウス関数と三角関数の積をとることにより、三角関数を有限範囲で減衰させる。

ガウス関数
(正規分布)



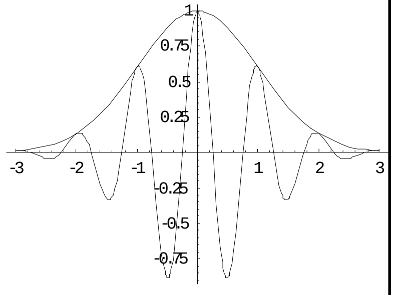
×

三角関数



＝

Gabor (ガボール) 関数

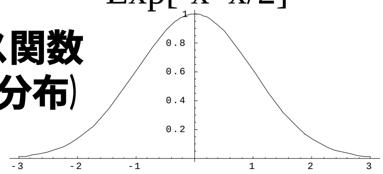


1次元Gabor (ガボール) 関数

- ガウス関数と三角関数の積をとることにより、三角関数を有限範囲で減衰させる。

Exp[-x*x/2]

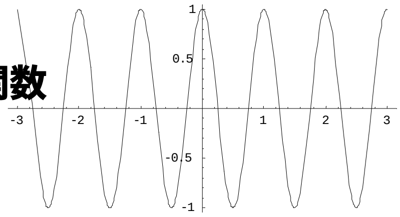
ガウス関数 (正規分布)



×

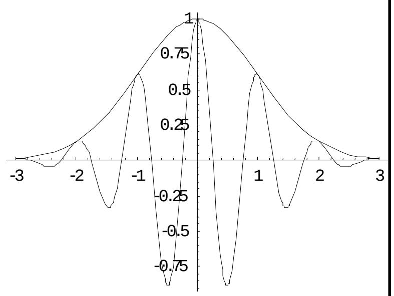
Cos[2Pi*x]

三角関数



=

Gabor (ガボール) 関数

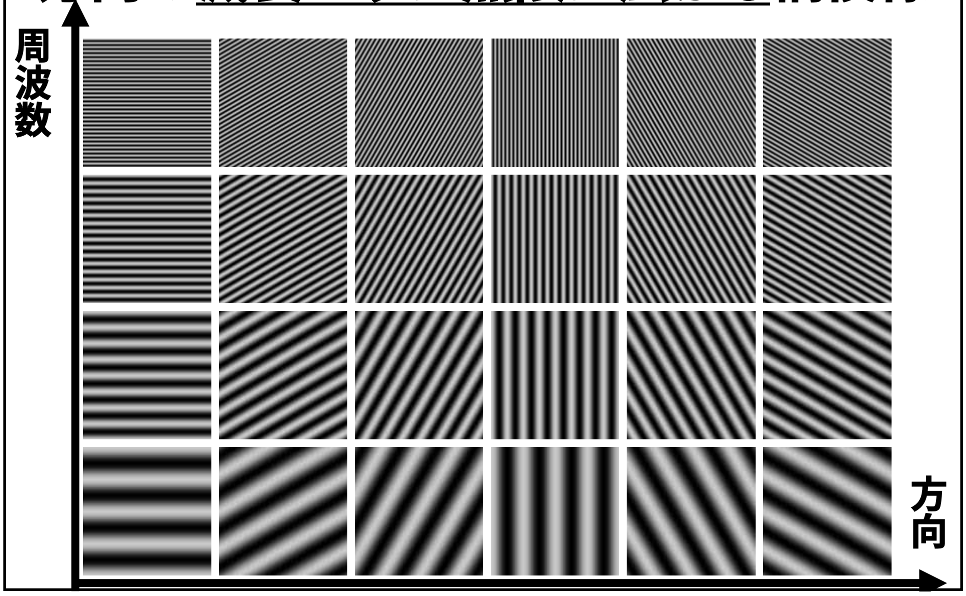


Exp[-x*x/2]*Cos[2Pi*x]

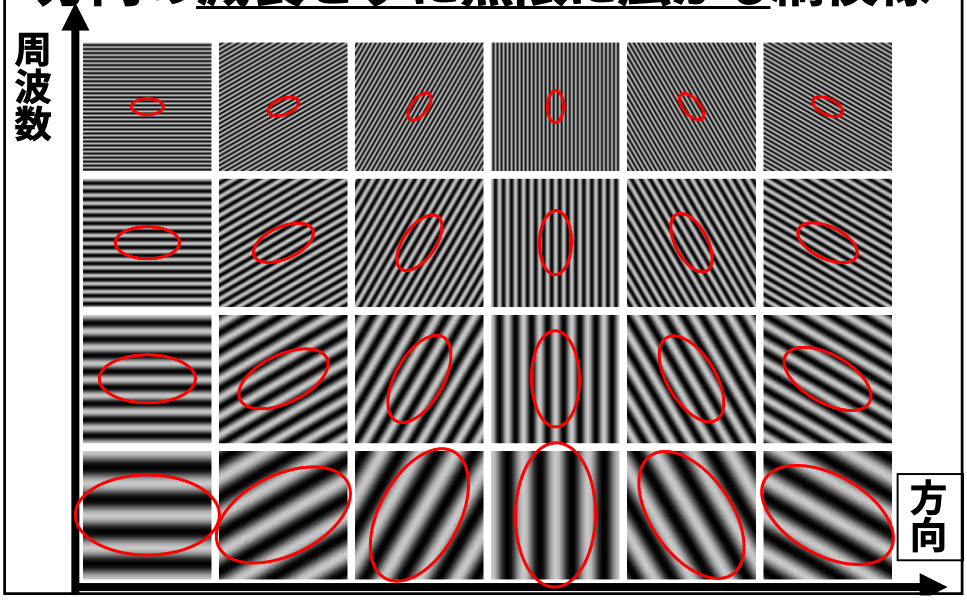
Mathematica によるGabor関数のプロット

- 1次元
  - ガウス関数 Plot [{Exp[-x*x/2]}, {x,-3.0,3.0}]
  - 三角関数 Plot [{Cos[2Pi*x]}, {x,-3.0,3.0}]
  - ガボール関数 Plot [{Exp[-x*x/2]* Cos[2Pi*x]}, {x,-3.0,3.0}]
- 2次元
  - ガウス関数 DensityPlot [{Exp[-(x*x+y*y)/2]}, {x,-3.0, 3.0}, {y,-3.0,3.0}, PlotRange->{-1,1}, PlotPoints ->100, Mesh -> False]
  - 三角関数 DensityPlot [{Cos[2Pi*x]}, {x,-3.0,3.0} ,{y,-3.0,3.0}, PlotRange->{-1,1}, PlotPoints ->100, Mesh -> False]
  - ガボール関数 DensityPlot[{Exp[-(x*x+y*y)/2]* Cos[2Pi*x]},{x,-3.0,3.0} , {y,-3.0,3.0}, PlotRange->{-1,1}, PlotPoints ->100, Mesh -> False]

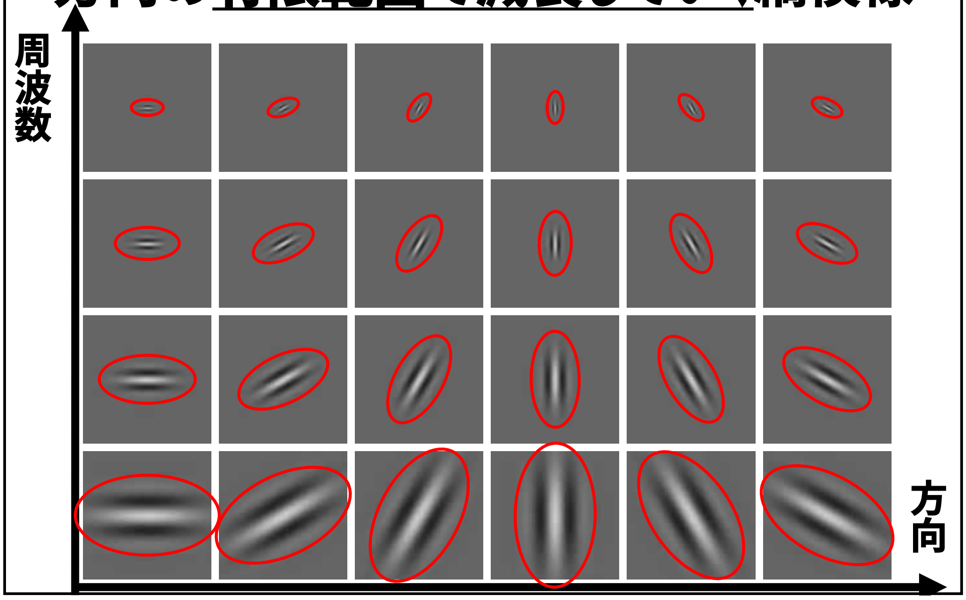
2次元三角関数 ＝ 様々の幅(周波数)・方向の減衰せずに無限に広がる縞模様



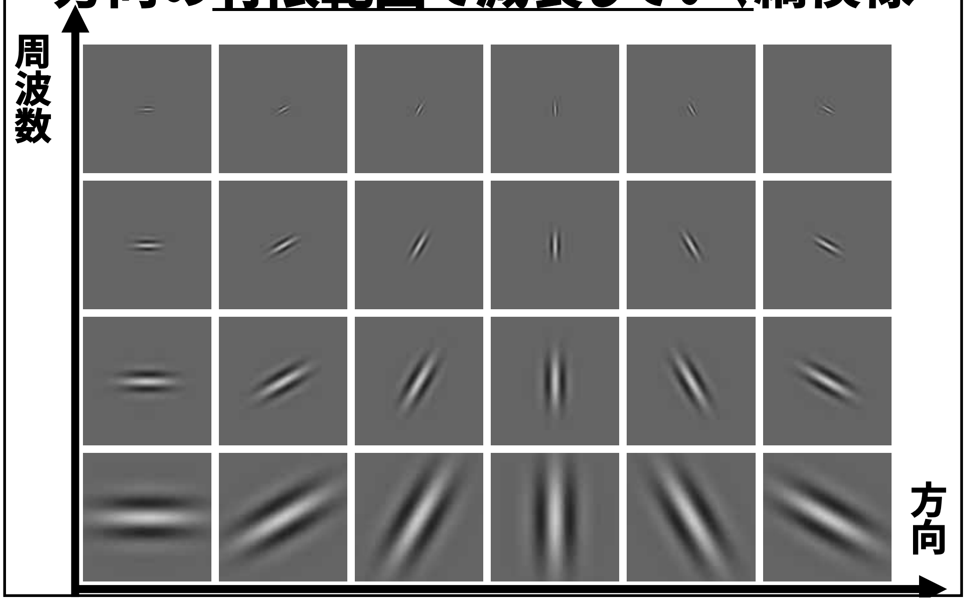
2次元三角関数 ＝ 様々の幅(周波数)・方向の減衰せずに無限に広がる縞模様



2次元Gabor関数 ＝ 様々の幅(周波数)・方向の有限範囲で減衰していく縞模様



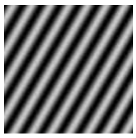
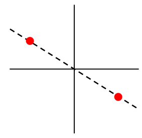
2次元Gabor関数 ＝ 様々の幅(周波数)・方向の有限範囲で減衰していく縞模様



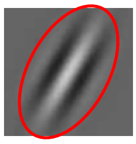
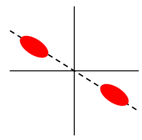
不確定性原理

- 空間領域（画像）
- 周波数領域（フーリエ変換）

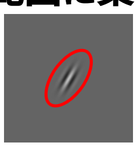
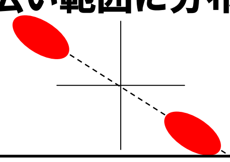
無限に広がる ⟺ 一点に集まる

広い範囲に分布する ⟺ 狭い範囲に集まる

狭い範囲に集まる ⟺ 広い範囲に分布する

Mathematica による2次元フーリエ変換：Gabor関数

http://www.image.med.osaka-u.ac.jp/
member/yoshi/ouec_lecture/image_recognition/

http://www.image.med.osaka-u.ac.jp/
member/yoshi/lecture.html　　画像認識をクリック

- 2次元Gabor関数画像 の “gabor_xx_yyy.pgm” をダウンロードする。(xx はしま模様の周期（幅）、yyyはしま模様の角度を表す。)
  - デスクトップにファイルを置いた場合、ファイルパスは、MACでは、
  - /Users/w学籍番号/Desktop/……./bar_data0.txt
  - MACでファイルパスを知る方法
    1. プルダウンメニュー　入力　－＞　ファイルパスの取得
    2. Terminal にフォルダをおく
- Mathematica でダウンロードした画像を表示する。
- Mathematica で2次元波形画像のフーリエ変換を行う。Gabor関数の周期（幅）（中心周波数）と角度が変わるとフーリエ変換はどのように変わるかを確かめよ。

2次元Gabor関数画像のフーリエ変換サンプル

2次元Gabor関数画像の入力と表示

2次元波形画像入力、画像表示、3次元表示

```
g =
Import[“d:/presen/oecu_game_lecture/gabor_images/gabor_
08_120.pgm”];  gabor08120 = g[[1,1]]-100;
ListDensityPlot[gabor08120, Mesh->False, PlotRange->All];
ListPlot3D[gabor08120, PlotRange->All];
```

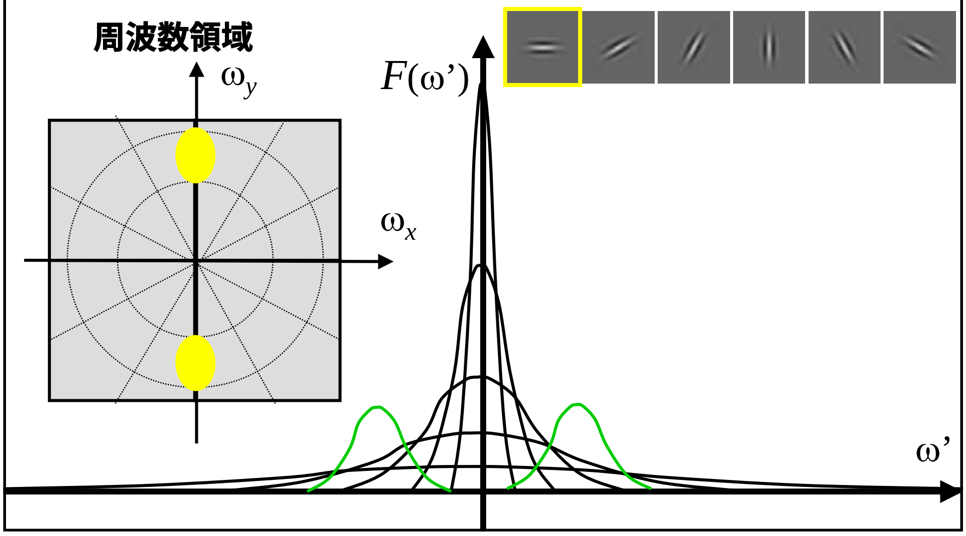
2次元Gabor関数画像のフーリエ変換と表示

フーリエ変換、画像表示、3次元表示

```
fgabor08120 = Fourier[gabor08120];
ListDensityPlot[Abs[fgabor08120], Mesh->False,
PlotRange->All];
ListPlot3D[Abs[fgabor08120], PlotRange->All];
```

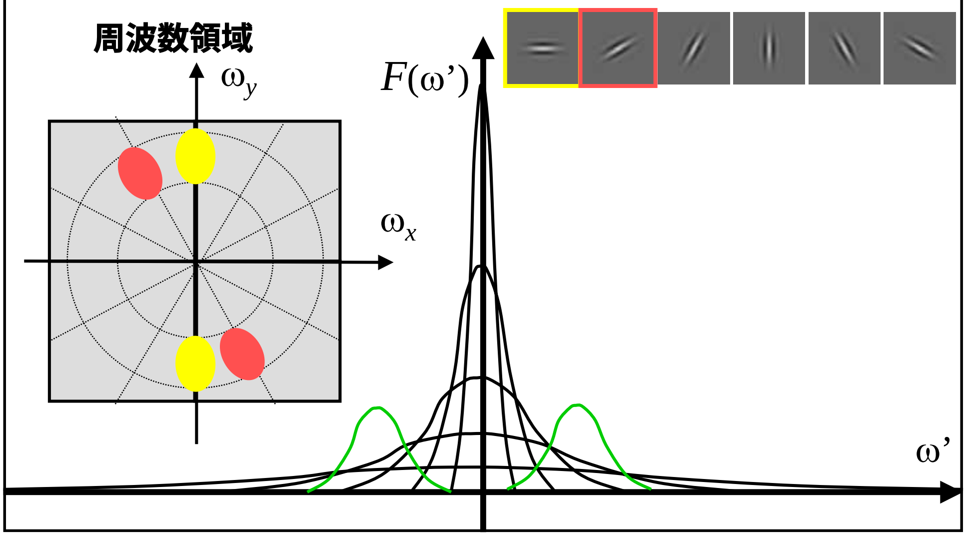
Gabor関数のフーリエ変換

特定の方向(の範囲)のみに反応するような画像解析を行う。複数の方向に対してこの処理を行うことを、多重方向解析と呼ぶ。



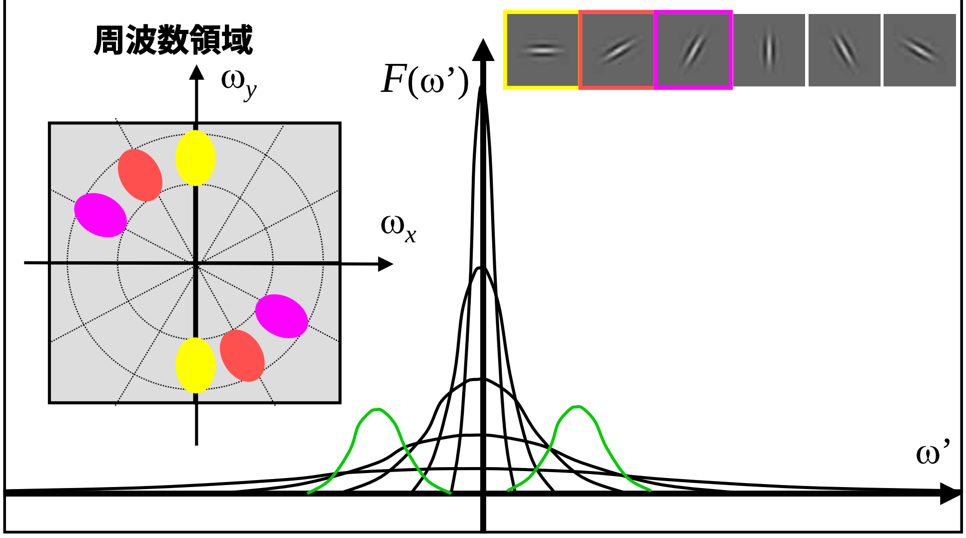
Gabor関数のフーリエ変換

特定の方向（の範囲）のみに反応するような画像解析を行う。複数の方向に対してこの処理を行うことを、多重方向解析と呼ぶ。



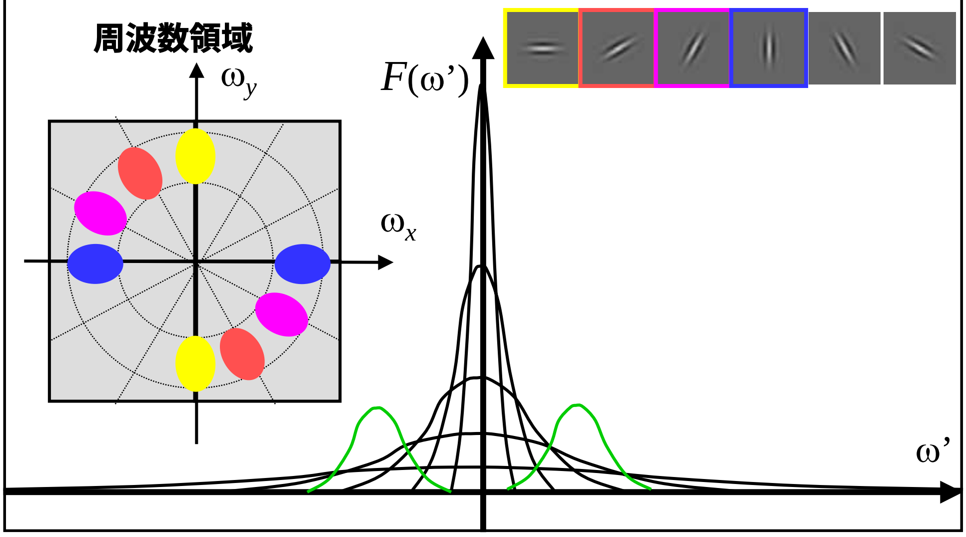
Gabor関数のフーリエ変換

特定の方向（の範囲）のみに反応するような画像解析を行う。複数の方向に対してこの処理を行うことを、多重方向解析と呼ぶ。



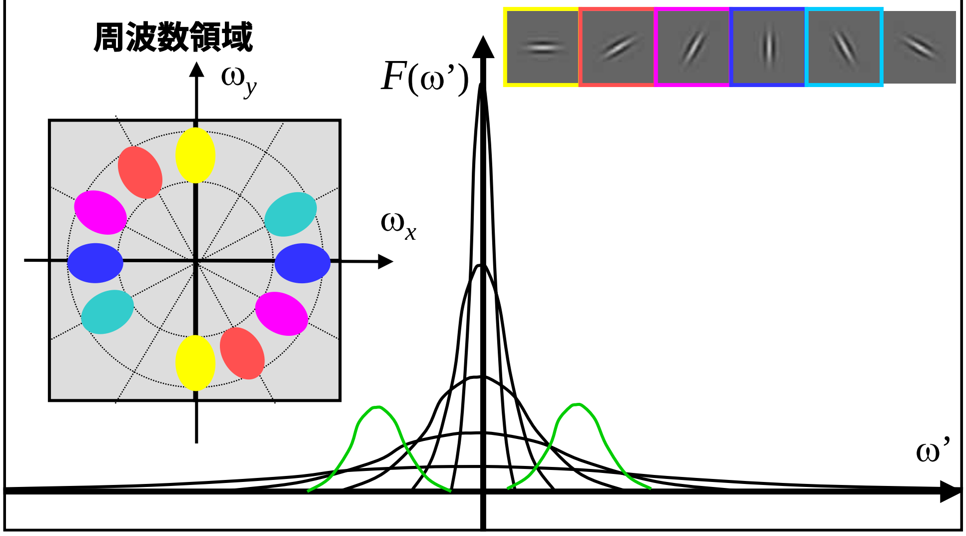
Gabor関数のフーリエ変換

特定の方向（の範囲）のみに反応するような画像解析を行う。複数の方向に対してこの処理を行うことを、多重方向解析と呼ぶ。



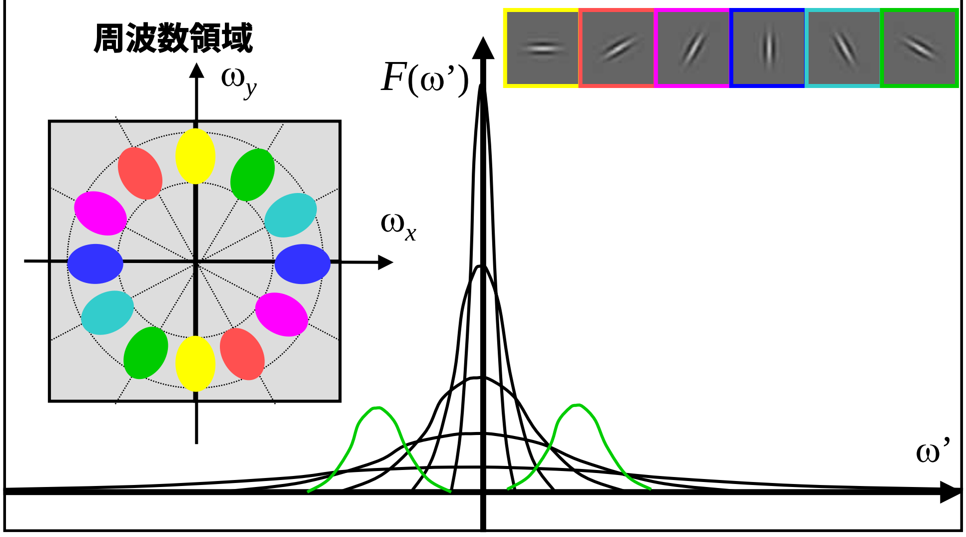
Gabor関数のフーリエ変換

特定の方向（の範囲）のみに反応するような画像解析を行う。複数の方向に対してこの処理を行うことを、多重方向解析と呼ぶ。



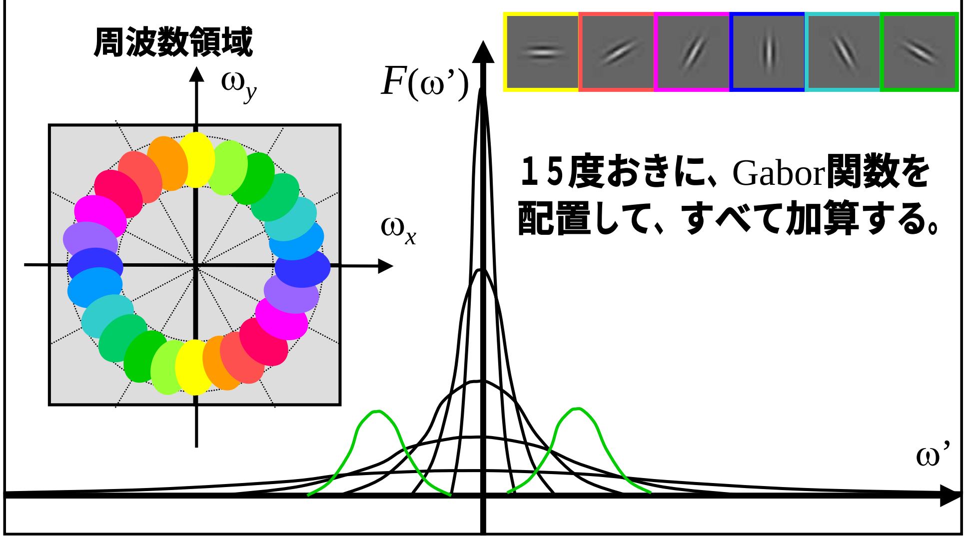
Gabor関数のフーリエ変換

特定の方向（の範囲）のみに反応するような画像解析を行う。複数の方向に対してこの処理を行うことを、多重方向解析と呼ぶ。



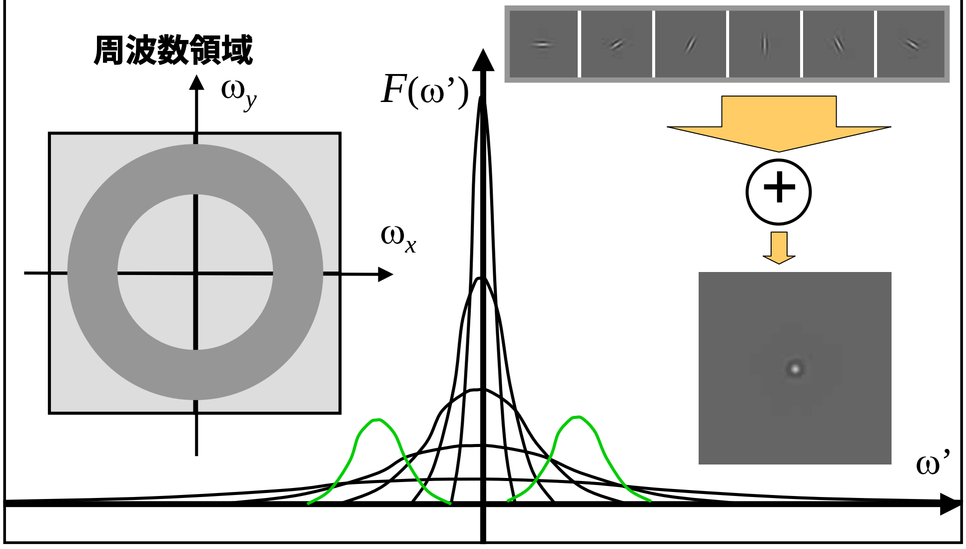
Gabor関数のフーリエ変換

特定の方向（の範囲）のみに反応するような画像解析を行う。複数の方向に対してこの処理を行うことを、多重方向解析と呼ぶ。



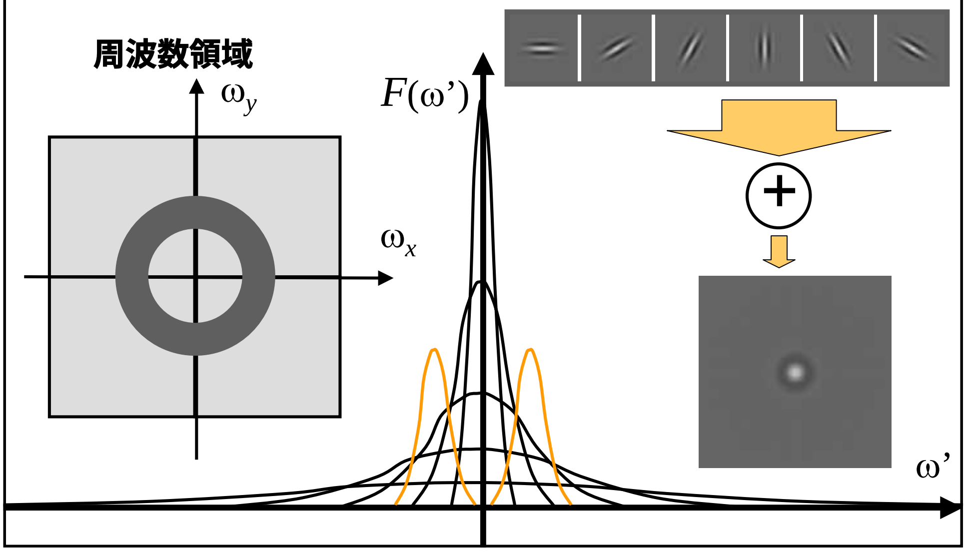
多重解像度解析

すべてを加算すると、方向に関係なく、特定の解像度（周波数の範囲）のみを強調するような処理を行うことができる。



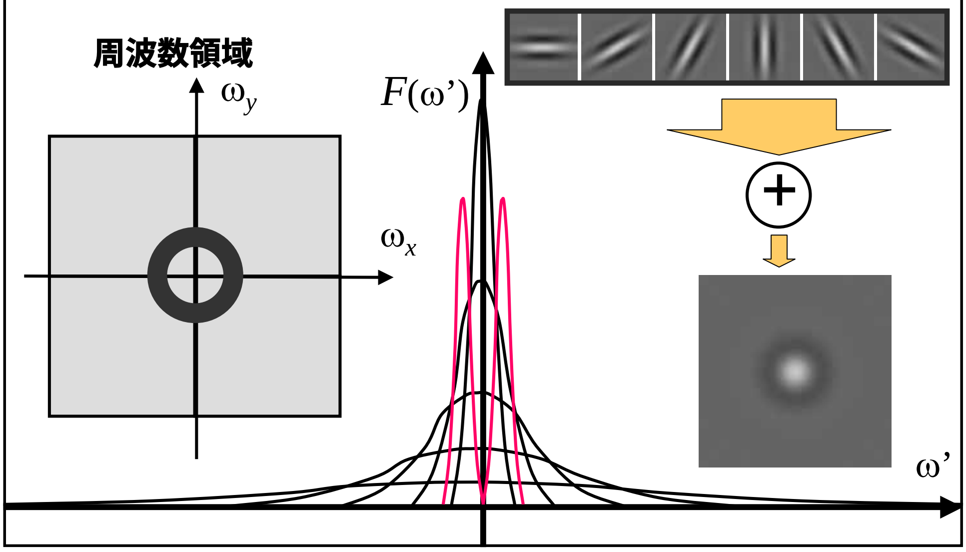
多重解像度解析

より低周波数の解像度（より大きな対象）のみを強調する。



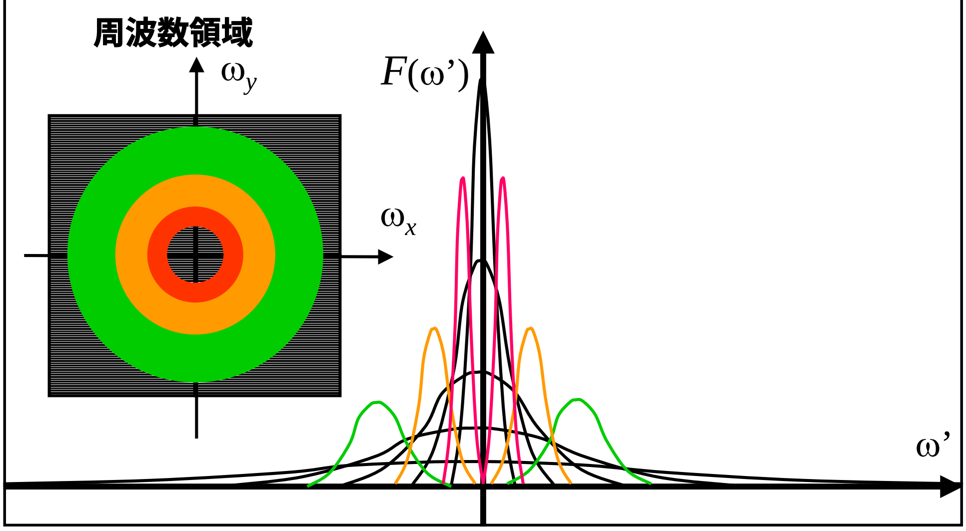
多重解像度解析

さらに低周波数の解像度（より大きな対象）のみを強調する。



多重解像度解析

特定の解像度（周波数の範囲）のみに反応するような画像解析を行う。
複数の解像度に対してこの処理を行うことを、多重解像度解析と呼ぶ。



Mathematica による2次元フーリエ変換：Gabor関数

- ある幅（周期）における全方向2次元Gabor関数画像
  - “gabor_08_000.pgm”
  - “gabor_08_015.pgm”
  - “gabor_08_030.pgm”
  - “gabor_08_045.pgm”
  - “gabor_08_060.pgm”
  - “gabor_08_075.pgm”
  - “gabor_08_090.pgm”
  - “gabor_08_105.pgm”
  - “gabor_08_120.pgm”
  - “gabor_08_135.pgm”
  - “gabor_08_150.pgm”
  - “gabor_08_165.pgm”

をダウンロードする。1つ前のスライドと同様に、入力･表示せよ。

- 以上の2次元Gabor関数画像をすべて加算し、cgabor_08という変数名にして、表示せよ。されにそれをフーリエ変換し、表示せよ。

2次元Gabor関数画像の全方向加算のフーリエ変換サンプル

2次元Gabor関数画像の加算と表示

2次元波形画像入力、画像表示、3次元表示

```
cgabor08 = gabor08000 + gabor08015 + gabor08030 +
……..+ gabor08150 + gabor08165
ListDensityPlot[cgabor08, Mesh->False, PlotRange->All];
ListPlot3D[cgabor08, PlotRange->All];
```

全方向加算された2次元Gabor関数画像のフーリエ変換と表示

フーリエ変換、画像表示、3次元表示

```
fcgabor08 = Fourier[cgabor08];
ListDensityPlot[Abs[fcgabor08], Mesh->False,
PlotRange->All];
ListPlot3D[Abs[fcgabor08], PlotRange->All];
```

2次元Gabor関数画像の全方向加算のフーリエ変換サンプル

全方向2次元Gabor関数画像（周期8、pgm）の入力とフーリエ変換

```
g = Import["/…/…/…/gabor_08_120.pgm"];
gabor08120 = g[[1,1]]ー100;
fgabor08120 = Fourier[gabor08120];
```

方向120度の場合

以上を全12方向 gabor_08_000.pgm(gabor08000), gabor_08_015.pgm(gabor08015), gabor_08_030.pgm(gabor08030), …………. , gabor_08_150.pgm(gabor08150), gabor_08_165.pgm(gabor08165) **について行う。**

2次元Gabor関数画像の全方向加算とフーリエ変換

```
cgabor08 = gabor08000 + gabor08015 + gabor08030 +
……..+ gabor08150 + gabor08165
fcgabor08 = Fourier[cgabor08];
```

表示

```
ListDensityPlot[Abs[fcgabor08], Mesh->False, PlotRange->All];
ListPlot3D[Abs[fcgabor08], PlotRange->All];
```

2次元Gabor関数画像の全方向加算のフーリエ変換サンプル

全方向2次元Gabor関数画像（周期8、pgm）の入力・加算・フーリエ変換

```
g = Import["/…/…/…/gabor_08_000.pgm"];
gabor08000 = g[[1,1]]ー100;
g = Import["/…/…/…/gabor_08_015.pgm"];
gabor08015 = g[[1,1]]ー100;
g = Import["/…/…/…/gabor_08_030.pgm"];
gabor08030 = g[[1,1]]ー100;
        :
g = Import["/…/…/…/gabor_08_150.pgm"];
gabor08150 = g[[1,1]]ー100;
g = Import["/…/…/…/gabor_08_165.pgm"];
gabor08165 = g[[1,1]]ー100;
cgabor08 = gabor08000 + gabor08015 + gabor08030 + ……..+
gabor08150 + gabor08165
fcgabor08 = Fourier[cgabor08];
ListDensityPlot[Abs[fcgabor08], Mesh->False, PlotRange->All];
ListPlot3D[Abs[fcgabor08], PlotRange->All];
```

2次元Gabor関数画像の全方向加算のフーリエ変換サンプル

04, 08, 16, 32 の各幅 (周期) の全方向Gabor関数画像をダウンロードし、全方向加算した2次元Gabor関数画像をつくる。

cgabor04.pgm
cgabor08.pgm
cgabor16.pgm
cgabor32.pgm

Mathematica 演習：特定幅の線の認識

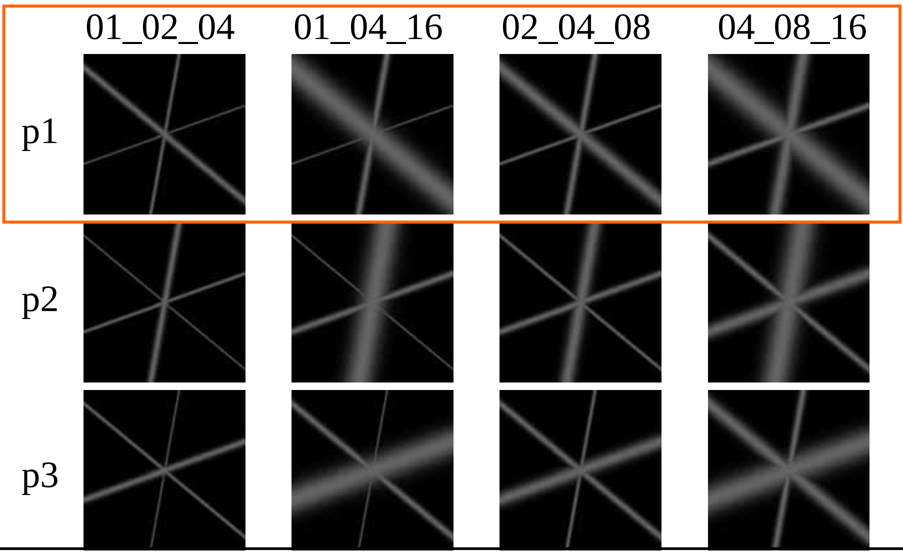
http://www.image.med.osaka-u.ac.jp/member/yoshi/ouec_lecture/image_recognition/

http://www.image.med.osaka-u.ac.jp/member/yoshi/lecture.html　画像認識をクリック

- 3本の複数幅の線画像 "mgline_xx_yy_zz_pi.pgm" をダウンロードする。(xx,yy,zz は線の幅、iは線の配置パターン番号を表す。)
  デスクトップにファイルを置いた場合、ファイルパスは、MACでは、
  /Users/w学籍番号/Desktop/……./bar_data0.txt
  MACでファイルパスを知る方法
  1　プルダウンメニュー　入力　−＞　ファイルパスの取得
  2　Terminal にフォルダをおく
- Mathematica でダウンロードした画像を表示する。
- 全方向加算したGaborフィルタを用いて、特定の線幅のみ強調するような画像変換を行え。

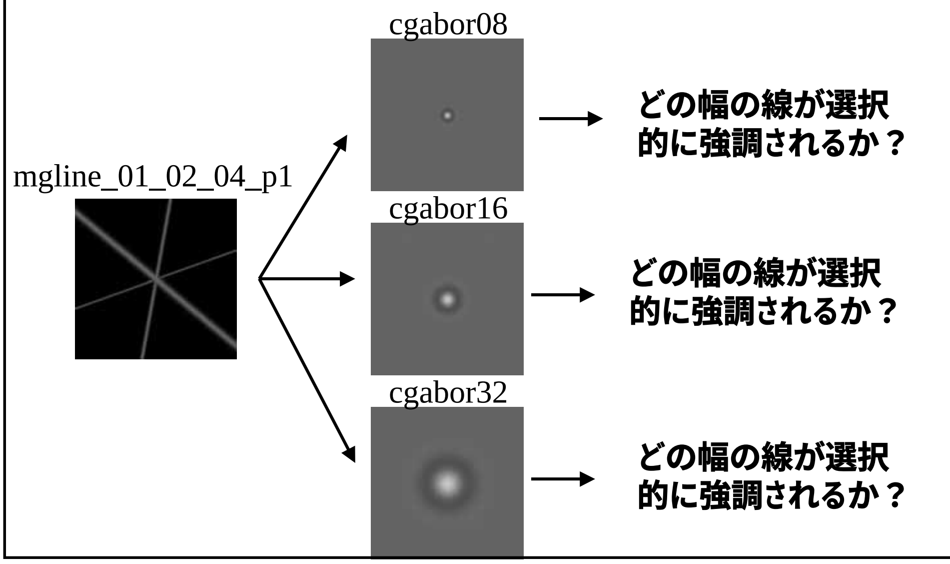
Mathematica 演習：特定幅の線の認識

- 3本の複数幅の線画像 “mgline_xx_yy_zz_pi.pgm” をダウンロードする。(xx,yy,zz は線の幅、iは線の配置パターン番号を表す。)



Mathematica 演習：特定幅の線の認識

- 3本の複数幅の線画像 “mgline_xx_yy_zz_pi.pgm” をダウンロードする。(xx,yy,zz は線の幅、iは線の配置パターン番号を表す。)



Mathematica 演習：特定幅の線の認識

様々な複数線の画像

mgline_01_02_04_p1.pgm
mgline_02_04_08_p1.pgm
⋮

のフーリエ変換と、04, 08, 16, 32 **の各幅（周期）の全方向加算2次元**Gabor**関数画像**

cgabor04.pgm
cgabor08.pgm
cgabor16.pgm
cgabor32.pgm

のフーリエ変換の（絶対値）の積をとる。

Mathematica 演習：特定幅の線の認識

複数線の画像

mgline_01_02_04_p1.pgm

を入力して、フーリエ変換する。

```
g = Import["/…./…./…./mgline_01_02_04_p1.pgm"];
line = g[[1,1]];
ListDensityPlot[line, Mesh->False,PlotRange->All];
fline = Fourier[line];
ListDensityPlot[Abs[fline], Mesh->False,PlotRange->All];
```

Mathematica 演習：特定幅の線の認識

全方向加算2次元Gabor関数画像
cgabor08.pgm
のフーリエ変換（の絶対値）との積をとり、逆フーリエ変換をする。

```
fcgabor08 = Fourier[cgabor08];
gfline = fline*Abs[fcgabor08]]/(8*8);
gline = InverseFourier[gfline];
ListDensityPlot[Abs[gfline], Mesh->False,PlotRange->All];
ListDensityPlot[Abs[gline], Mesh->False,
PlotRange->{40,50}];
```

Mathematica 演習：特定幅の線の認識

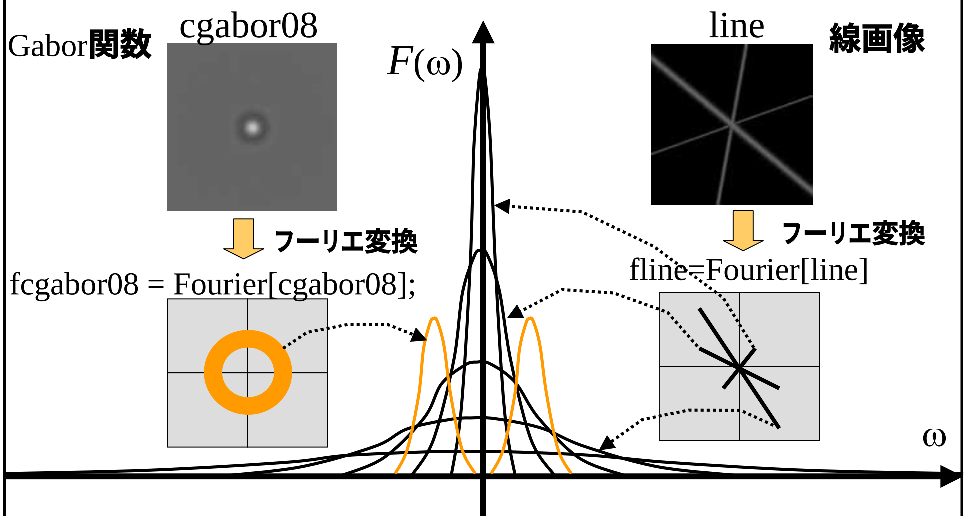
複数幅の線画像の入力・フーリエ変換・全方向加算Gaborフィルタリング

```
g = Import["/…./…./…./mgline_01_02_04_p1.pgm"];
line = g[[1,1]];
ListDensityPlot[line, Mesh->False,PlotRange->All];
fline = Fourier[line];
ListDensityPlot[Abs[fline], Mesh->False,PlotRange->All];

fcgabor08 = Fourier[cgabor08];
gfline = fline*Abs[fcgabor08]]/(8*8);
gline = InverseFourier[gfline];
ListDensityPlot[Abs[gfline], Mesh->False,PlotRange->All];
ListDensityPlot[Abs[gline], Mesh->False,
PlotRange->{40,50}];
```

ここを変えてやってみる

Mathematica 演習：特定幅の線の認識



線画像フーリエ変換とGabor関数フーリエ変換の積

gfline = fline*Abs[fcgabor08]]/(8*8) ← 出力レベル調整

逆フーリエ変換　Abs[InverseFourier[gfline]]

Mathematica 演習：特定幅の線の認識

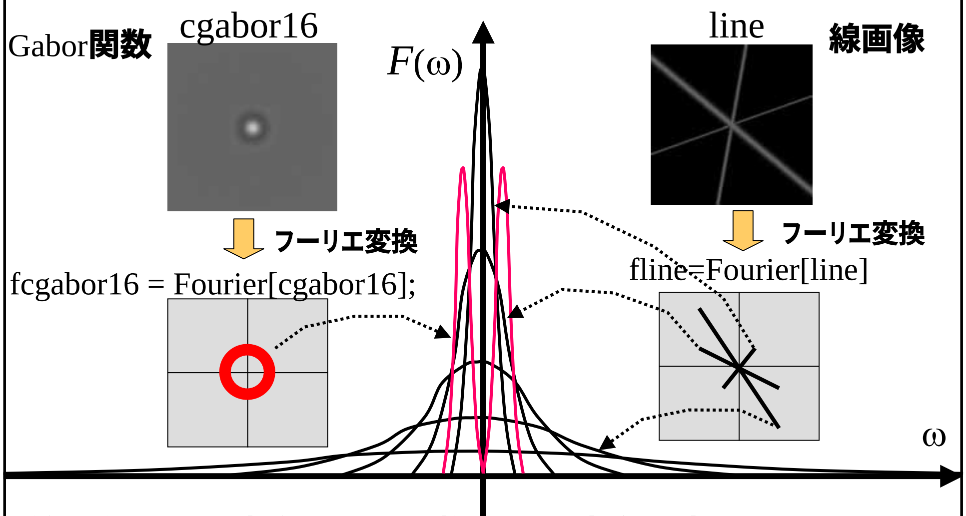
全方向加算2次元Gabor関数画像

cgabor16.pgm

のフーリエ変換（の絶対値）との積をとり、逆フーリエ変換をする。

```
fcgabor16 = Fourier[cgabor16];
gfline = fline*Abs[fcgabor16]]/(16*16);
gline = InverseFourier[gfline];
ListDensityPlot[Abs[gfline], Mesh->False,PlotRange->All];
ListDensityPlot[Abs[gline], Mesh->False,
PlotRange->{40,50}];
```

Mathematica 演習：特定幅の線の認識



線画像フーリエ変換とGabor関数フーリエ変換の積

gfline = fline*Abs[fcgabor16]/(16*16) ← 出力レベル調整

逆フーリエ変換　Abs[InverseFourier[gfline]]

Mathematica 演習：特定幅の線の認識

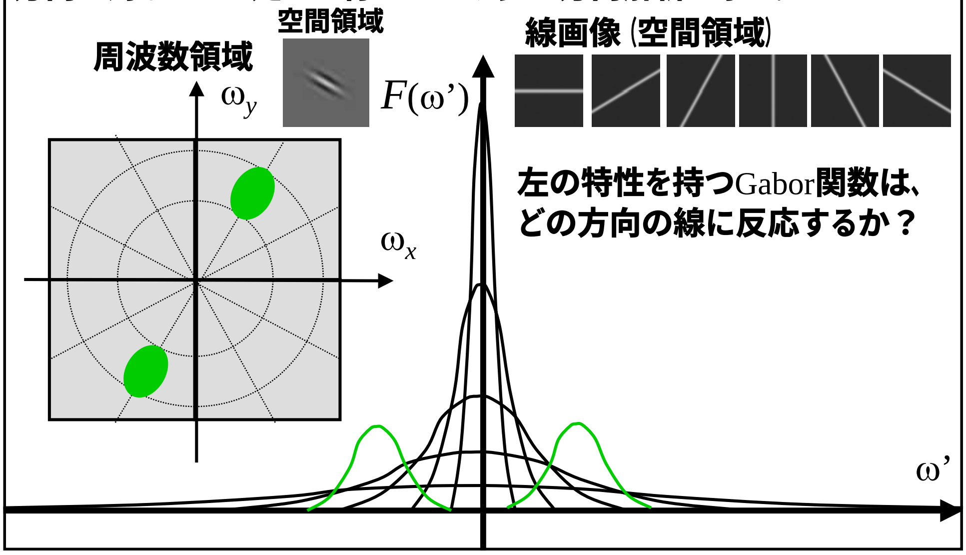
問題1：どの幅の線が、どの幅（周波数）の全方向加算2次元Gabor関数によって、強調されたか？ を整理せよ。

問題2：ある幅（周波数）の全方向加算2次元Gabor関数によって、特定の幅の線のみが抽出できる原理を図を用いて、説明せよ。

多重方向解析

多重方向解析

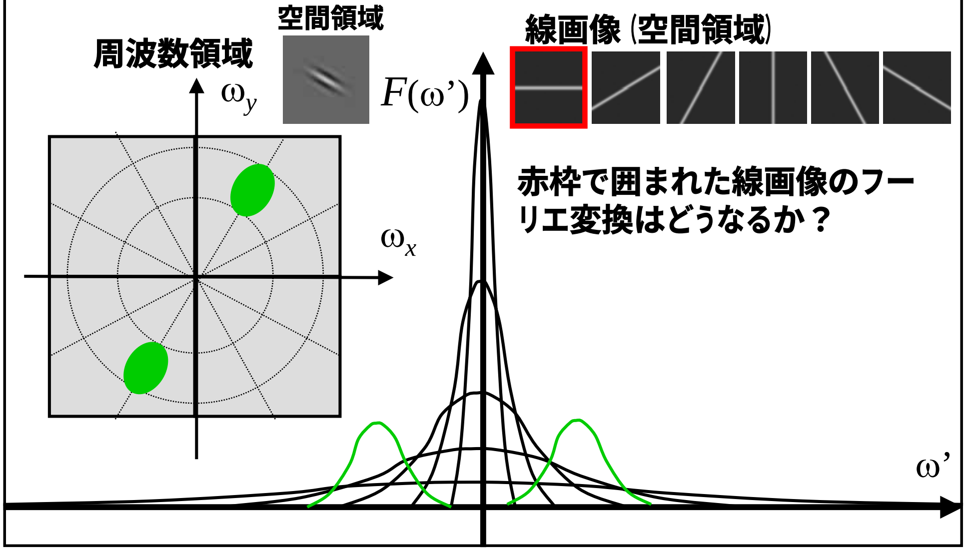
特定の方向（の範囲）のみに反応するような画像解析を行う。複数の方向に対してこの処理を行うことを、多重方向解析と呼ぶ。



左の特性を持つGabor関数は、どの方向の線に反応するか？

多重方向解析

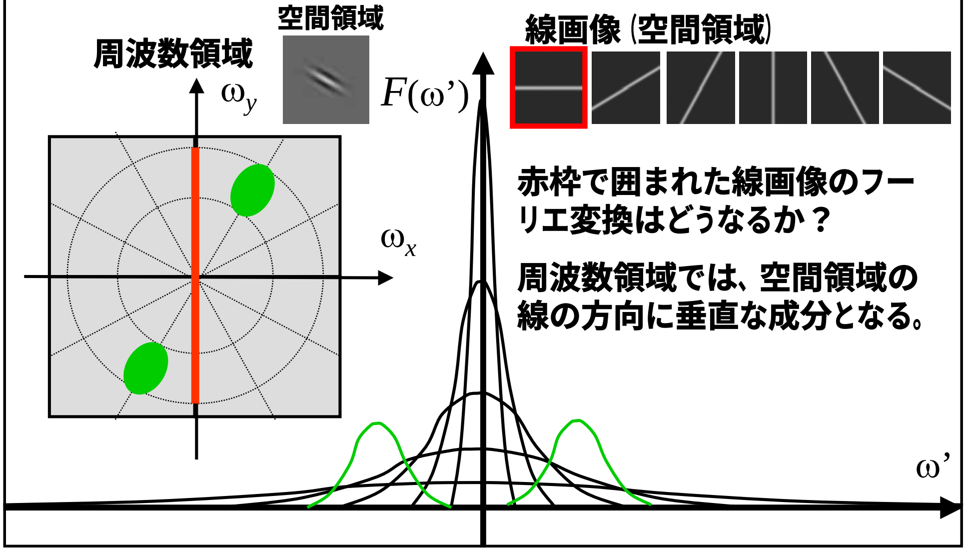
特定の方向（の範囲）のみに反応するような画像解析を行う。複数の方向に対してこの処理を行うことを、多重方向解析と呼ぶ。



赤枠で囲まれた線画像のフーリエ変換はどうなるか？

多重方向解析

特定の方向（の範囲）のみに反応するような画像解析を行う。複数の方向に対してこの処理を行うことを、多重方向解析と呼ぶ。

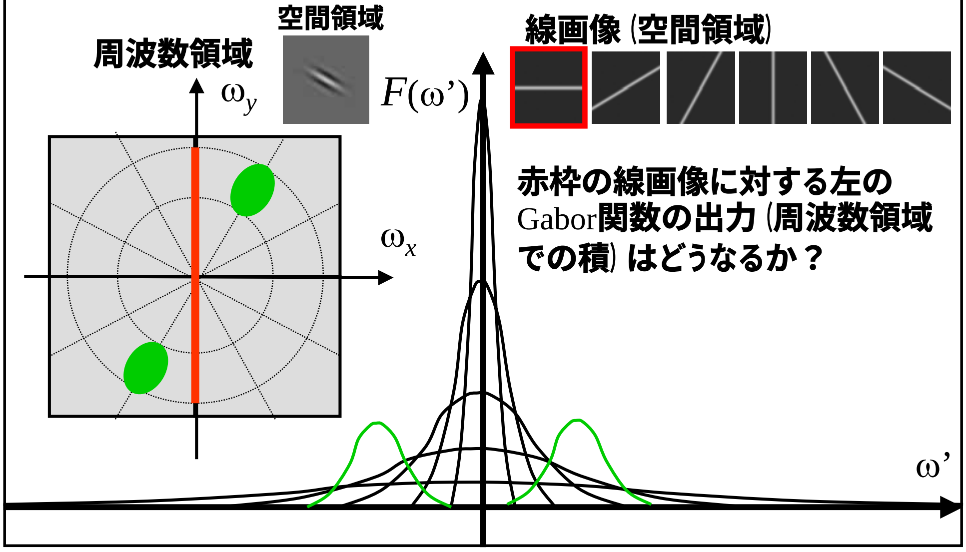


赤枠で囲まれた線画像のフーリエ変換はどうなるか？

周波数領域では、空間領域の線の方向に垂直な成分となる。

多重方向解析

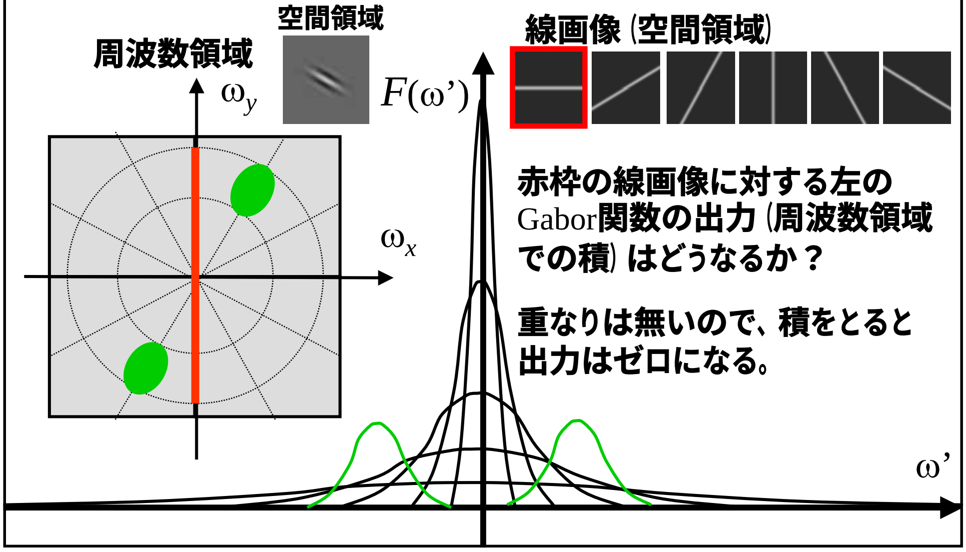
特定の方向（の範囲）のみに反応するような画像解析を行う。複数の方向に対してこの処理を行うことを、多重方向解析と呼ぶ。



赤枠の線画像に対する左のGabor関数の出力（周波数領域での積）はどうなるか？

多重方向解析

特定の方向（の範囲）のみに反応するような画像解析を行う。複数の方向に対してこの処理を行うことを、多重方向解析と呼ぶ。

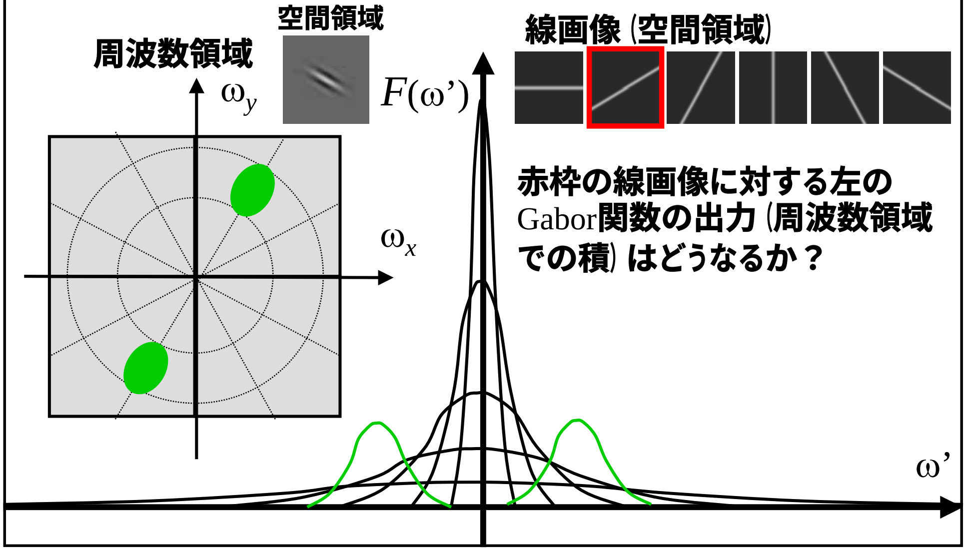


赤枠の線画像に対する左のGabor関数の出力（周波数領域での積）はどうなるか？

重なりは無いので、積をとると出力はゼロになる。

多重方向解析

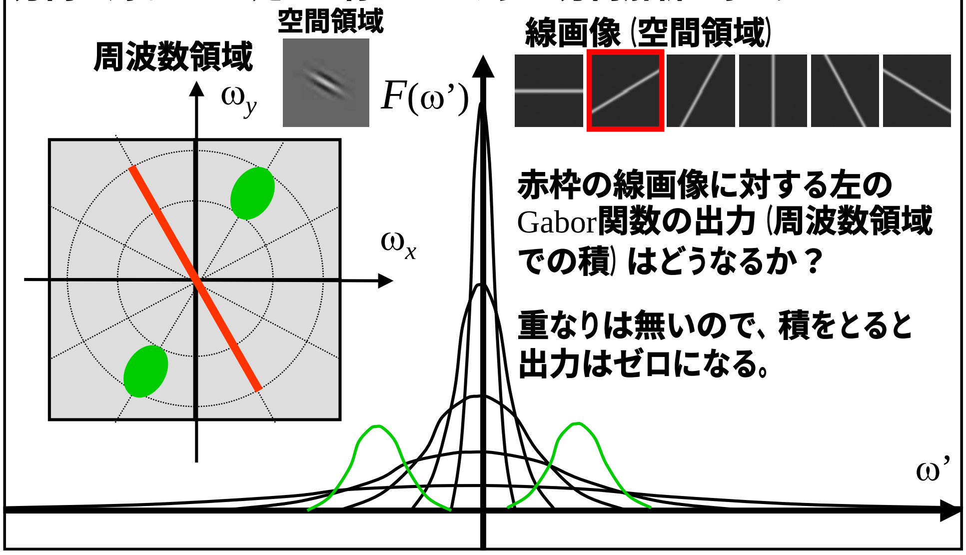
特定の方向 (の範囲) のみに反応するような画像解析を行う。複数の方向に対してこの処理を行うことを、多重方向解析と呼ぶ。



赤枠の線画像に対する左のGabor関数の出力 (周波数領域での積) はどうなるか？

多重方向解析

特定の方向 (の範囲) のみに反応するような画像解析を行う。複数の方向に対してこの処理を行うことを、多重方向解析と呼ぶ。

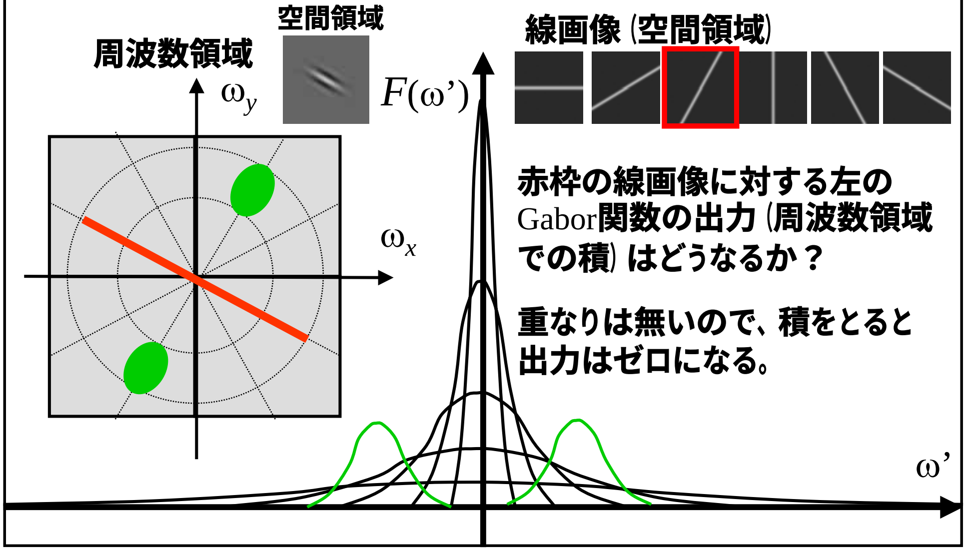


赤枠の線画像に対する左のGabor関数の出力 (周波数領域での積) はどうなるか？

重なりは無いので、積をとると出力はゼロになる。

多重方向解析

特定の方向 (の範囲) のみに反応するような画像解析を行う。複数の方向に対してこの処理を行うことを、多重方向解析と呼ぶ。

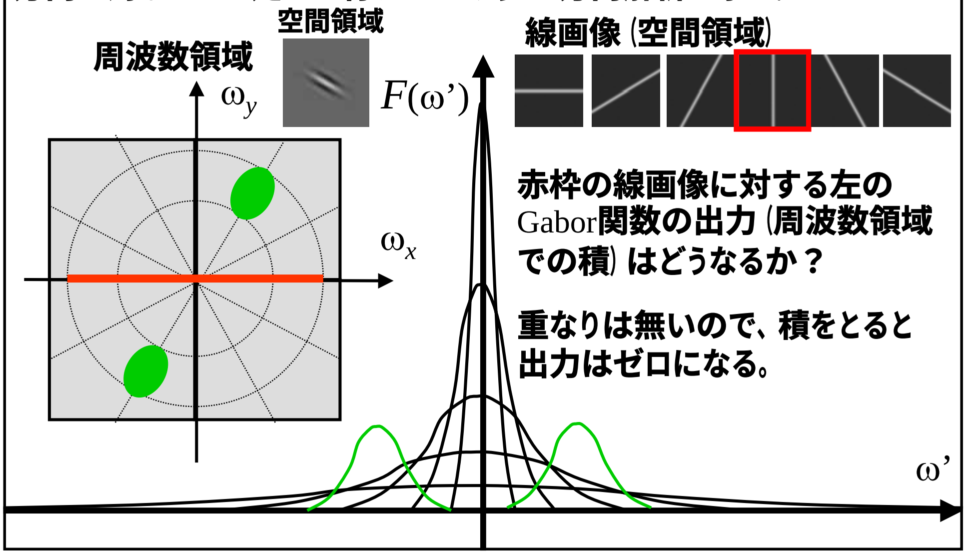


赤枠の線画像に対する左のGabor関数の出力 (周波数領域での積) はどうなるか？

重なりは無いので、積をとると出力はゼロになる。

多重方向解析

特定の方向 (の範囲) のみに反応するような画像解析を行う。複数の方向に対してこの処理を行うことを、多重方向解析と呼ぶ。

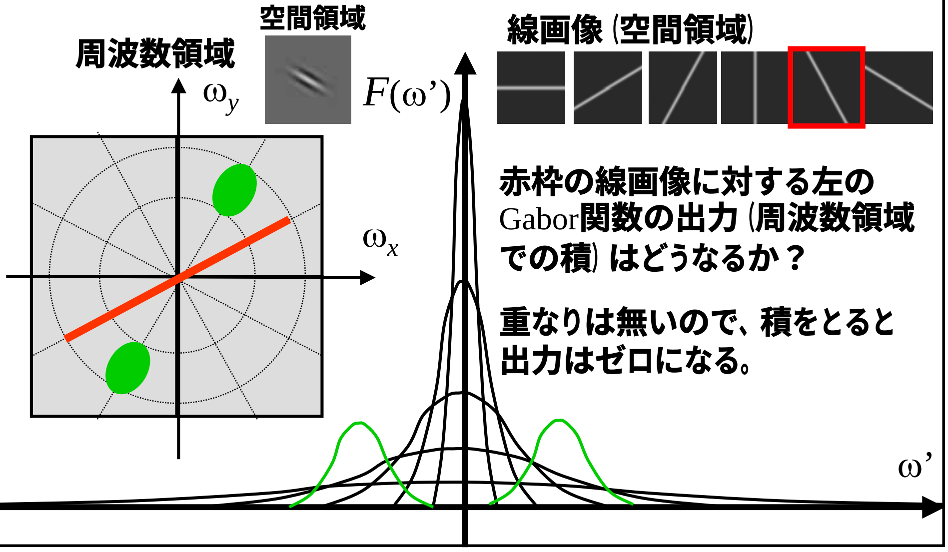


赤枠の線画像に対する左のGabor関数の出力 (周波数領域での積) はどうなるか？

重なりは無いので、積をとると出力はゼロになる。

多重方向解析

特定の方向（の範囲）のみに反応するような画像解析を行う。複数の方向に対してこの処理を行うことを、多重方向解析と呼ぶ。

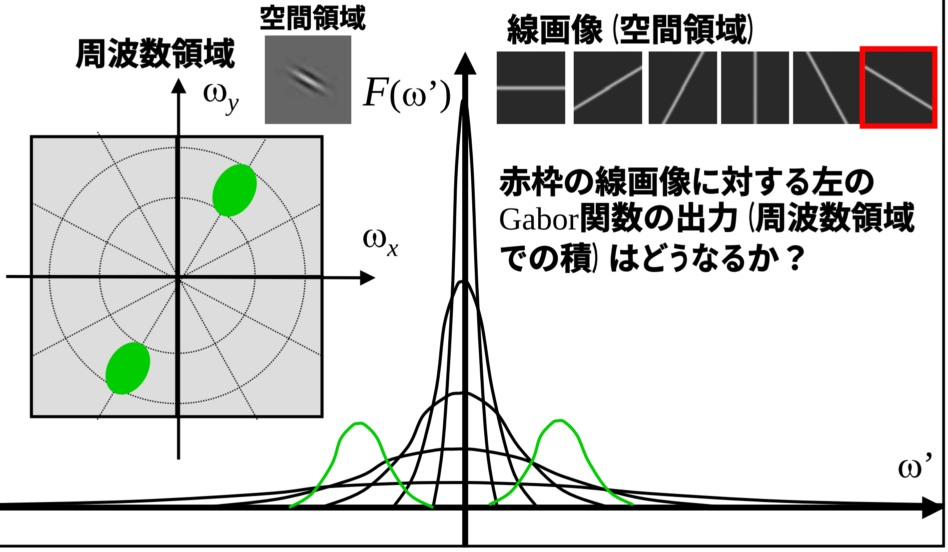


赤枠の線画像に対する左のGabor関数の出力（周波数領域での積）はどうなるか？

重なりは無いので、積をとると出力はゼロになる。

多重方向解析

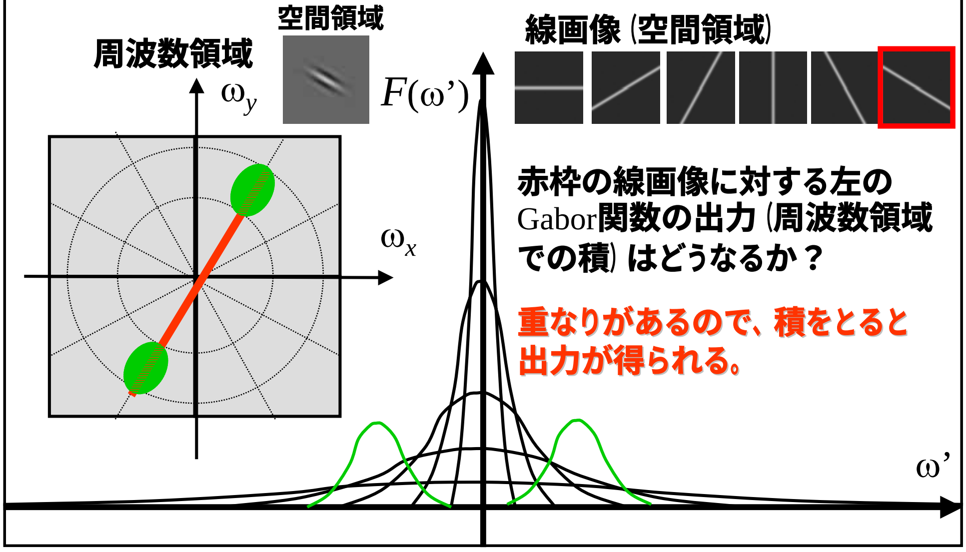
特定の方向（の範囲）のみに反応するような画像解析を行う。複数の方向に対してこの処理を行うことを、多重方向解析と呼ぶ。



赤枠の線画像に対する左のGabor関数の出力（周波数領域での積）はどうなるか？

多重方向解析

特定の方向（の範囲）のみに反応するような画像解析を行う。複数の方向に対してこの処理を行うことを、多重方向解析と呼ぶ。

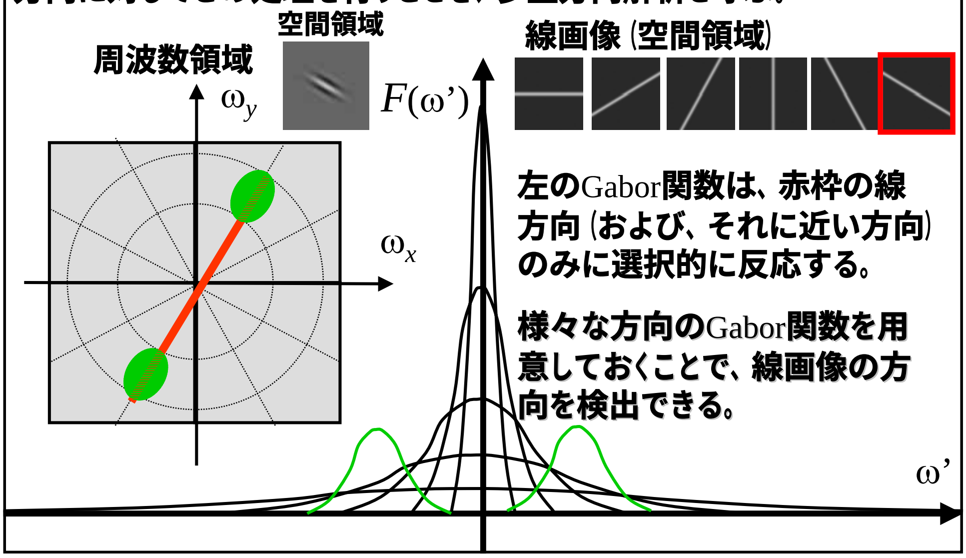


赤枠の線画像に対する左のGabor関数の出力（周波数領域での積）はどうなるか？

重なりがあるので、積をとると出力が得られる。

多重方向解析

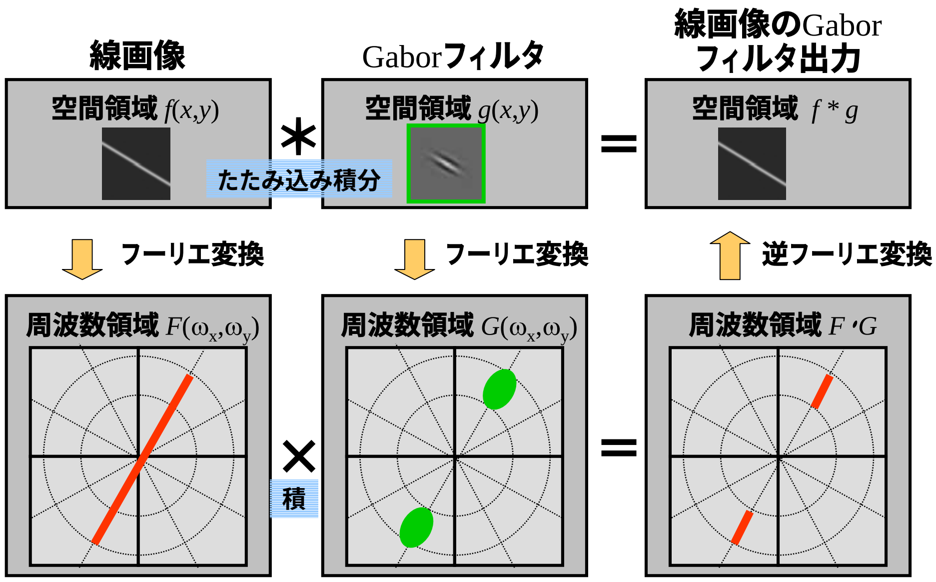
特定の方向（の範囲）のみに反応するような画像解析を行う。複数の方向に対してこの処理を行うことを、多重方向解析と呼ぶ。



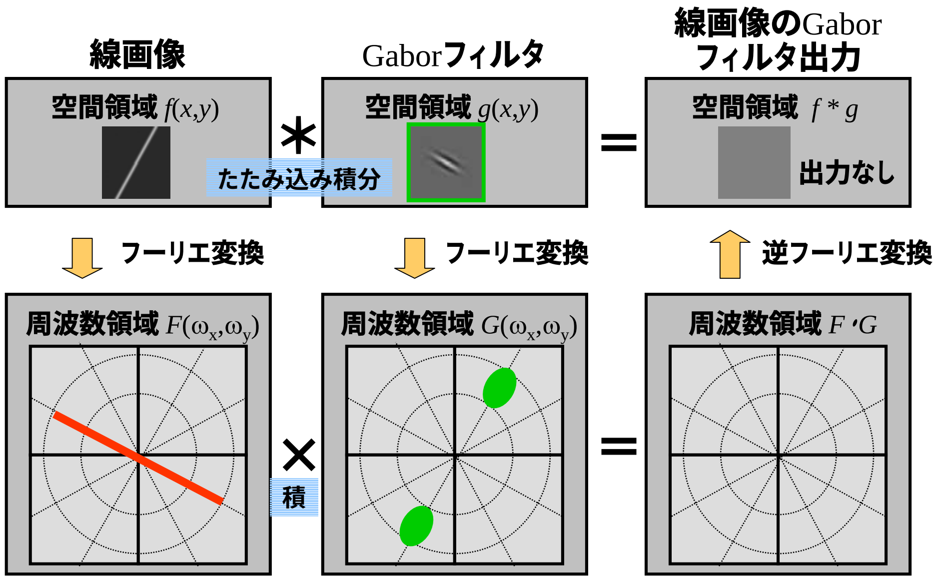
左のGabor関数は、赤枠の線方向（および、それに近い方向）のみに選択的に反応する。

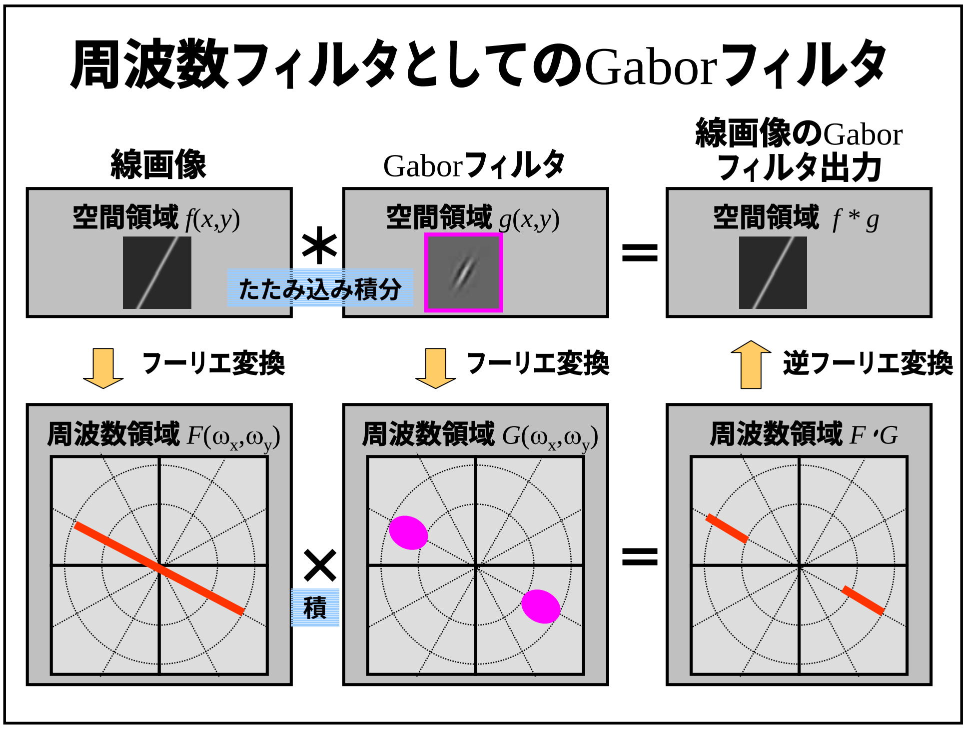
様々な方向のGabor関数を用意しておくことで、線画像の方向を検出できる。

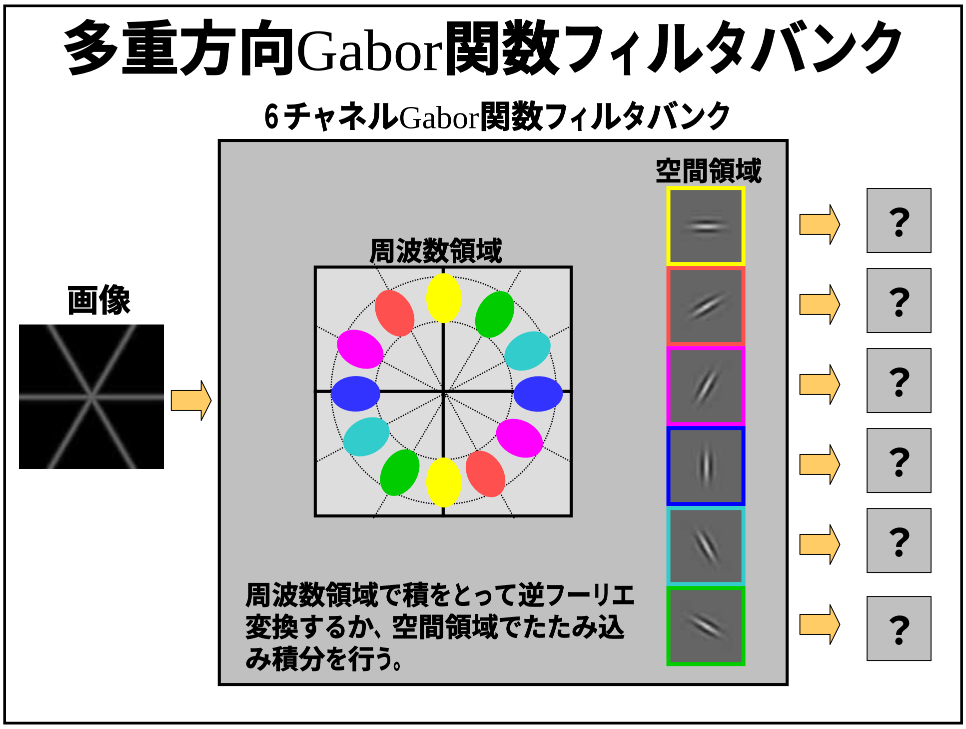
周波数フィルタとしてのGaborフィルタ



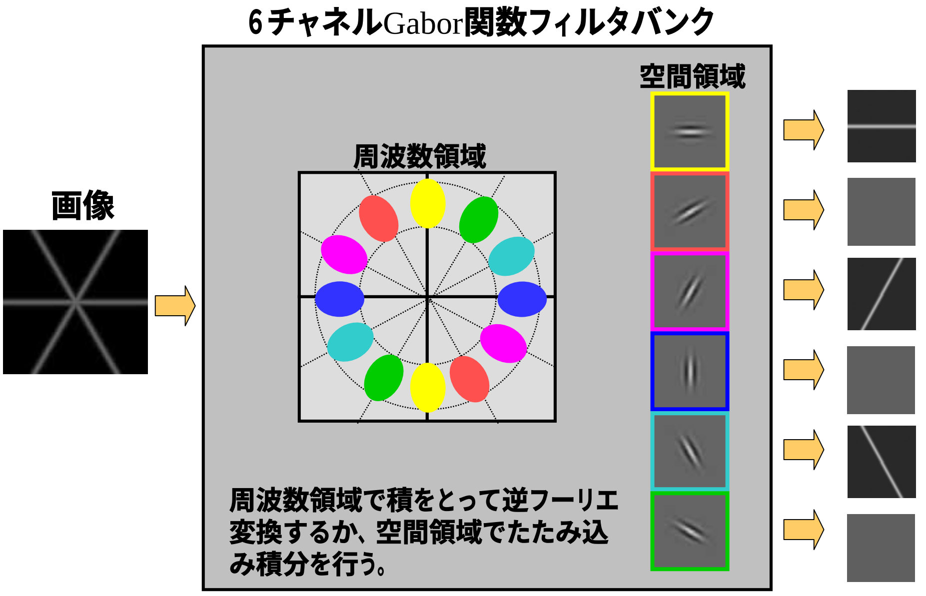
周波数フィルタとしてのGaborフィルタ




周波数フィルタとしてのGaborフィルタ
線画像
Gaborフィルタ
線画像のGabor
フィルタ出力
空間領域 f(x,y)
空間領域 g(x,y)
空間領域 f * g
たたみ込み積分
フーリエ変換
フーリエ変換
逆フーリエ変換
周波数領域 F(ωx,ωy)
周波数領域 G(ωx,ωy)
周波数領域 F ·G
積


多重方向Gabor関数フィルタバンク
6チャネルGabor関数フィルタバンク
画像
空間領域
周波数領域
周波数領域で積をとって逆フーリエ
変換するか、空間領域でたたみ込
み積分を行う。
?
?
?
?
?
?

多重方向Gabor関数フィルタバンク



Mathematica 演習：特定方向の線の認識

- 15度間隔の12チャネルGaborフィルタバンクを構成せよ。
  - 0, 15, 30, 45, ….., 135, 150, 165 度
- 単一方向線画像を入力して、出力画像を確認せよ。
  - Gabor フィルタと同一方向の線画像の出力は？
  - 少し異なる方向の線画像の出力画像は？
  - 全く異なる方向の線画像の出力画像は？
- 複数方向線画像を入力して、出力画像を確認せよ。
- 各Gaborフィルタに対する出力のノルム（各画素の二乗和の平方根）を求めて、角度を横軸、出力を縦軸にとってプロットせよ。
- 単一方向線画像を入力して、各Gaborフィルタに対する出力のノルムを利用して、線の方向をできるだけ正確に推定する方法を考案せよ。

Mathematica 演習：特定方向の線の認識

- １５度間隔のGaborフィルタバンクを構成せよ。
  - 0, 15, 30, 45, ….., 135, 150, 165 度

例：１２０度のGabor関数（幅8）の入力とそのフーリエ変換

```
g = Import["d: :/…./…./…. /gabor_08_120.pgm"];
gabor08120 = g[[1,1]]-100;
fgabor08120 = Fourier[gabor08120];
```

その表示（画像表示と3次元プロット表示）

```
ListDensityPlot[gabor08120, Mesh->False, PlotRange->All];
ListPlot3D[gabor08120, PlotRange->All];
ListDensityPlot[Abs[fgabor08120], Mesh->False, PlotRange->All];
ListPlot3D[Abs[fgabor08120], PlotRange->All];
```

すべての角度について行う（以上の入力はすでに多重解像度解析で行っている。）

Mathematica 演習：特定方向の線の認識

- 単一方向線画像を入力して、出力画像を確認せよ。
  - Gabor フィルタと同一方向の線画像の出力は？
  - 少し異なる方向の線画像の出力画像は？
  - 全く異なる方向の線画像の出力画像は？

線画像入力、フーリエ変換、Gaborフィルタ適用

```
g = Import["d:/…./…./…./gline_02_120.pgm"];
line = g[[1,1]];
fline = Fourier[line];
igfline = InverseFourier[fline*Abs[fgabor08120]]/(8*8);
```

表示（画像表示、3次元プロット表示）

```
ListDensityPlot[line, Mesh->False,PlotRange->All];
ListDensityPlot[Abs[igfline], Mesh->False,PlotRange->All];
```

gline_02_110 gline_02_100 gline_02_090 gline_02_080 などを試してみよ。

Mathematica 演習：特定方向の線の認識

- 複数方向線画像を入力して、出力画像を確認せよ。

線画像入力、フーリエ変換、Gaborフィルタ適用

```
g = Import["d:/…./…./…./mgline_02_p2.pgm"];
line = g[[1,1]];
fline = Fourier[line];
igfline = InverseFourier[fline*Abs[fgabor08120]]/(8*8);
```

表示 (画像表示、3次元プロット表示)

```
ListDensityPlot[line, Mesh->False,PlotRange->{0,20}];
ListDensityPlot[Abs[igfline], Mesh->False,PlotRange->{0.20}];
```

fgabor08000, fgabor08015, fgabor08030, fgabor08045, ………, fgabor08150, gfabor08165の全方向を試してみよ。

mgline_02_p1　mgline_02_p3 も試してみよ。

Mathematica 演習：特定方向の線の認識

- 各Gaborフィルタに対する出力画像のノルム (各画素の二乗和の平方根) を求めよ。

線画像入力、フーリエ変換、Gaborフィルタ適用

```
g = Import["d:/…./…./…./gline_02_110.pgm"];
line = g[[1,1]];
fline = Fourier[line];
gfline = fline*Abs[fgabor08120] /(8*8);
a120 = Norm[Flatten[gfline]];
```

値の出力と全方向出力のプロット

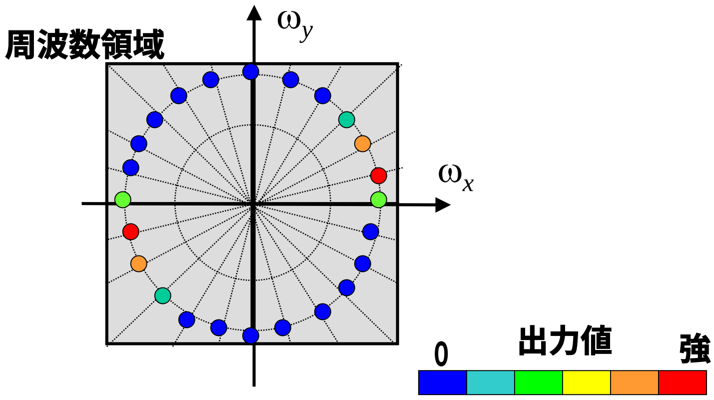
```
Print[a120];
ListPlot[{a000,a015,a030,….,a165},PlotJoined->True}];
```

fgabor08000, fgabor08015, fgabor08030, fgabor08045, ………, fgabor08150, gfabor08165の全方向についてのノルム a000, a015, a030, a045, ……., a150, a165 を計算し、値を出力せよ。

Mathematica 演習：特定方向の線の認識

- 単一方向線画像を入力して、各Gaborフィルタに対する出力のノルムを利用して、線の方向をできるだけ正確に推定する方法を考案せよ。

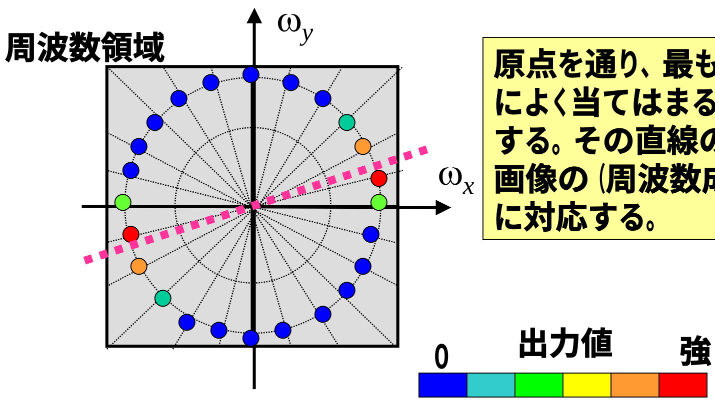
110度の線画像に対するGaborフィルタバンクの出力ノルム



Mathematica 演習：特定方向の線の認識

- 単一方向線画像を入力して、各Gaborフィルタに対する出力のノルムを利用して、線の方向をできるだけ正確に推定する方法を考案せよ。

110度の線画像に対するGaborフィルタバンクの出力ノルム



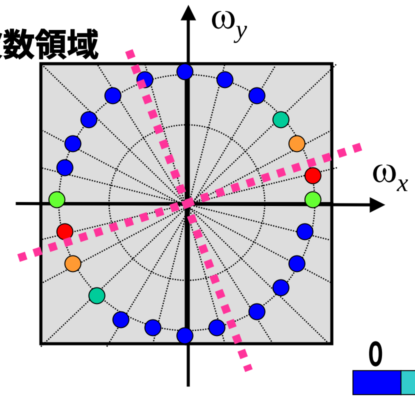
原点を通り、最もこの出力値によく当てはまる直線を推定する。その直線の方向が、線画像の（周波数成分の）方向に対応する。

Mathematica 演習：特定方向の線の認識

- 単一方向線画像を入力して、各Gaborフィルタに対する出力のノルムを利用して、線の方向をできるだけ正確に推定する方法を考案せよ。

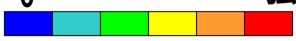
110度の線画像に対するGaborフィルタバンクの出力ノルム

周波数領域



最大モーメント軸
出力値を質量とみなした場合、それをコマの軸にして回せば、最も安定する。

最小モーメント軸
それをコマの軸にして回せば、最も不安定である（最大モーメント軸に直交）。

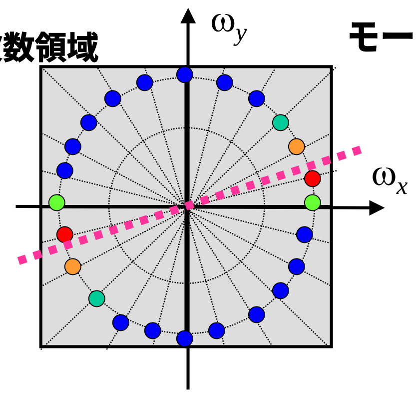


Mathematica 演習：特定方向の線の認識

- 原点を通り、最もこの出力値によく当てはまる直線を推定する。

モーメント行列の固有ベクトルを求める。モーメント行列の最小固有値に対応する固有ベクトル方向が「最小モーメント軸」となる直線方向である。

周波数領域



モーメント行列 $\begin{pmatrix} M_{xx} & M_{xy} \\ M_{xy} & M_{yy} \end{pmatrix}$

$$M_{xx} = \sum_{i=1}^{n} w_i x_i^2$$

$$M_{xy} = \sum_{i=1}^{n} w_i x_i y_i$$

$$M_{yy} = \sum_{i=1}^{n} w_i y_i^2$$

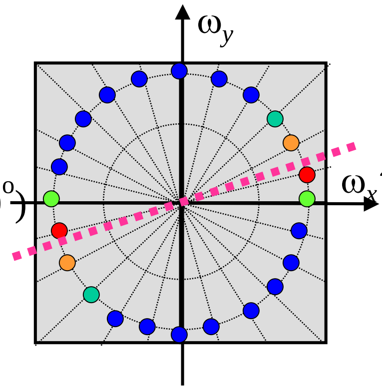
Mathematica 演習：特定方向の線の認識

- 原点を通り、最もこの出力値によく当てはまる直線を推定する。

モーメント行列の固有ベクトルを求める。モーメント行列の最小固有値に対応する固有ベクトル方向が「最小モーメント軸」となる直線方向である。

0度の出力値
a000

*x*座標値 *y*座標値
Cos(180°) Sin(180°)



*x*座標値 *y*座標値
Cos(0°) Sin(0°)

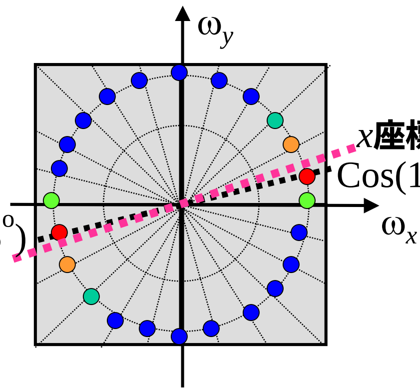
Mathematica 演習：特定方向の線の認識

- 原点を通り、最もこの出力値によく当てはまる直線を推定する。

モーメント行列の固有ベクトルを求める。モーメント行列の最小固有値に対応する固有ベクトル方向が「最小モーメント軸」となる直線方向である。

15度の出力値
a015

*x*座標値 *y*座標値
Cos(15°) Sin(195°)



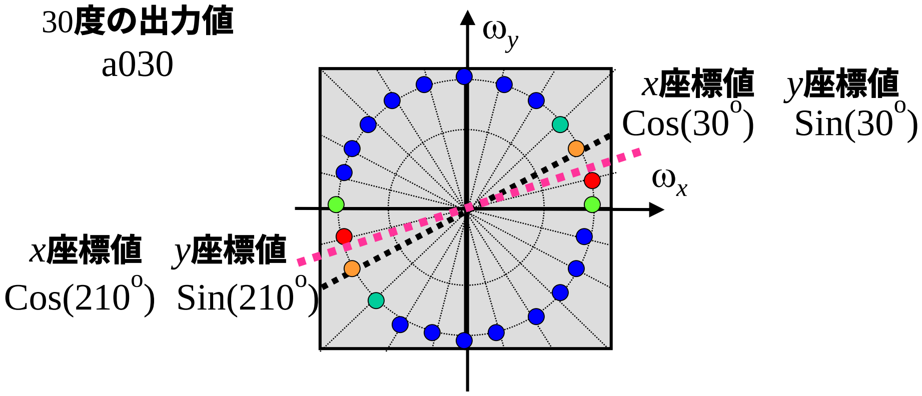
*x*座標値 *y*座標値
Cos(15°) Sin(15°)

Mathematica 演習：特定方向の線の認識

- 原点を通り、最もこの出力値によく当てはまる直線を推定する。

モーメント行列の固有ベクトルを求める。モーメント行列の最小固有値に対応する固有ベクトル方向が「最小モーメント軸」となる直線方向である。

30度の出力値
a030

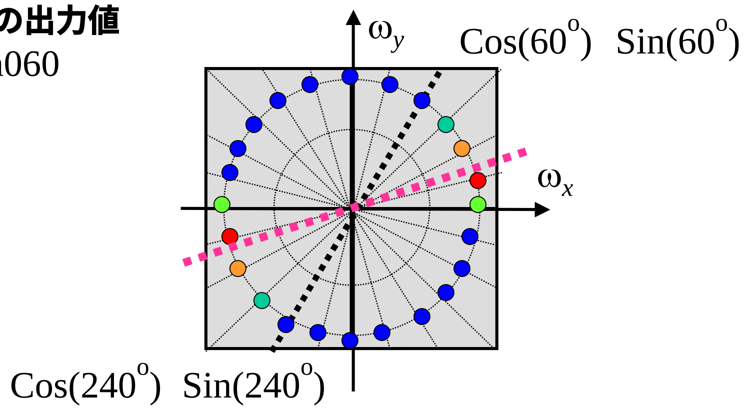


Mathematica 演習：特定方向の線の認識

- 原点を通り、最もこの出力値によく当てはまる直線を推定する。

モーメント行列の固有ベクトルを求める。モーメント行列の最小固有値に対応する固有ベクトル方向が「最小モーメント軸」となる直線方向である。

60度の出力値
a060

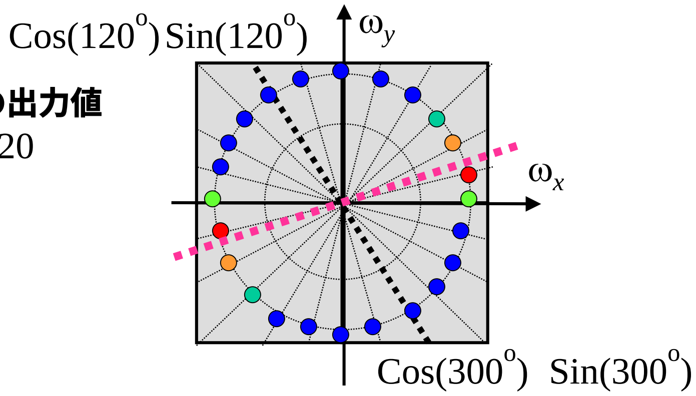


Mathematica 演習：特定方向の線の認識

- 原点を通り、最もこの出力値によく当てはまる直線を推定する。

モーメント行列の固有ベクトルを求める。モーメント行列の最小固有値に対応する固有ベクトル方向が「最小モーメント軸」となる直線方向である。

120度の出力値
a120

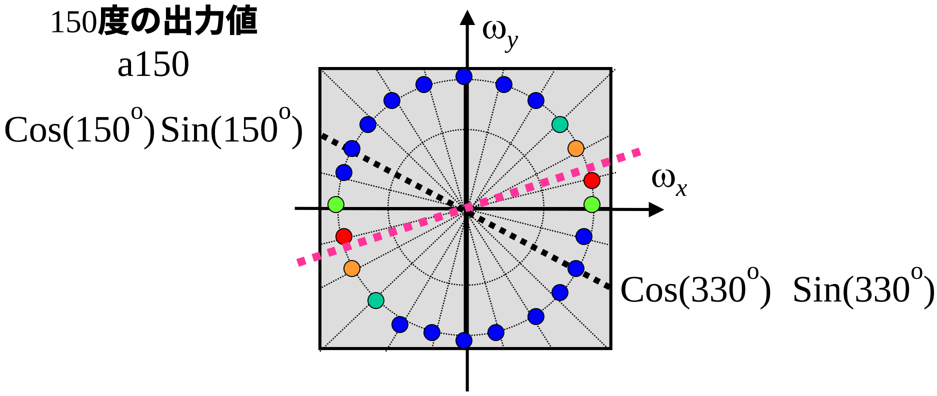


Mathematica 演習：特定方向の線の認識

- 原点を通り、最もこの出力値によく当てはまる直線を推定する。

モーメント行列の固有ベクトルを求める。モーメント行列の最小固有値に対応する固有ベクトル方向が「最小モーメント軸」となる直線方向である。

150度の出力値
a150



Mathematica 演習：特定方向の線の認識

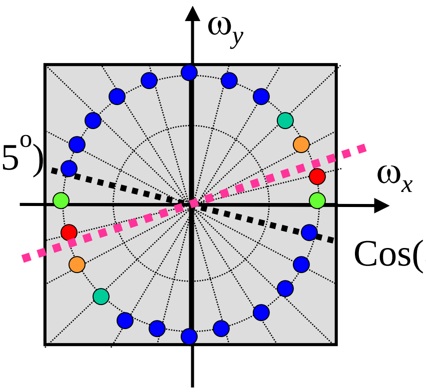
- 原点を通り、最もこの出力値によく当てはまる直線を推定する。

モーメント行列の固有ベクトルを求める。モーメント行列の最小固有値に対応する固有ベクトル方向が「最小モーメント軸」となる直線方向である。

165度の出力値

a165

Cos(165^{o}) Sin(165^{o})

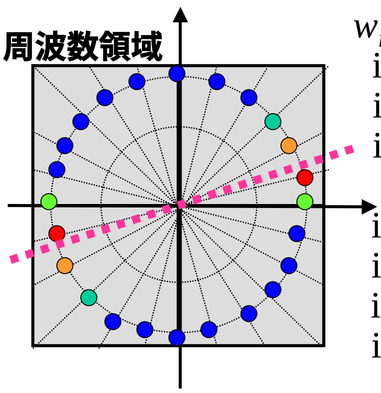


Cos(345^{o}) Sin(345^{o})

Mathematica 演習：特定方向の線の認識

モーメント行列の固有ベクトルを求める。モーメント行列の最小固有値に対応する固有ベクトル方向が「最小モーメント軸」となる直線方向である。

周波数領域

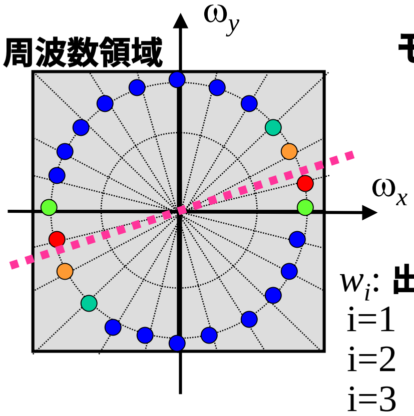


w_i: 出力ノルム値, x_i: x座標値, y_i: y座標値

i=1	a000	Cos(0^{o})	Sin(0^{o})
i=2	a015	Cos(15^{o})	Sin(15^{o})
i=3	a030	Cos(30^{o})	Sin(30^{o})
:	:	:	:
i=11	a150	Cos(150^{o})	Sin(150^{o})
i=12	a165	Cos(165^{o})	Sin(165^{o})
i=13	a000	Cos(180^{o})	Sin(180^{o})
i=14	a015	Cos(195^{o})	Sin(195^{o})
:	:	:	:
i=23	a150	Cos(330^{o})	Sin(330^{o})
i=24	a165	Cos(345^{o})	Sin(345^{o})

Mathematica 演習：特定方向の線の認識

モーメント行列の固有ベクトルを求める。モーメント行列の最小固有値に対応する固有ベクトル方向が「最小モーメント軸」となる直線方向である。



モーメント行列 $\begin{pmatrix} M_{xx} & M_{xy} \\ M_{xy} & M_{yy} \end{pmatrix}$

$$M_{xx} = \sum_{i=1}^{n} w_i x_i^2 \qquad M_{yy} = \sum_{i=1}^{n} w_i y_i^2$$

$$M_{xy} = \sum_{i=1}^{n} w_i x_i y_i$$

w_i: **出力ノルム値**, x_i: x**座標値**, y_i: y**座標値**

i=1	a000	Cos(0^o)	Sin(0^o)
i=2	a015	Cos(15^o)	Sin(15^o)
i=3	a030	Cos(30^o)	Sin(30^o)
	:		
i=24	a165	Cos(345^o)	Sin(345^o)

Mathematica 演習：特定方向の線の認識

モーメント行列の固有ベクトルを求める。モーメント行列の最小固有値に対応する固有ベクトル方向が直線方向である。

モーメント行列 $\begin{pmatrix} M_{xx} & M_{xy} \\ M_{xy} & M_{yy} \end{pmatrix}$ $M_{xx} = \sum_{i=1}^{n} w_i x_i^2 \quad M_{yy} = \sum_{i=1}^{n} w_i y_i^2$

$$M_{xy} = \sum_{i=1}^{n} w_i x_i y_i$$

mxx = N[a000*Cos[0^o] *Cos[0^o] + a015*Cos[15^o] *Cos[15^o] + …….
+ a165* Cos[165^o] *Cos[165^o] + a000* Cos[180^o] *Cos[180^o] +
a015* Cos[195^o] *Cos[195^o] + ……. + a165* Cos[345^o] *Cos[345^o]] ;

w_i: **出力ノルム値**,		x_i: x**座標値**,	y_i: y**座標値**
i=1	a000	Cos(0^o)	Sin(0^o)
i=2	a015	Cos(15^o)	Sin(15^o)
i=3	a030	Cos(30^o)	Sin(30^o)
	:		
i=24	a165	Cos(345^o)	Sin(345^o)

Mathematica 演習：特定方向の線の認識

モーメント行列の固有ベクトルを求める。モーメント行列の最小固有値に対応する固有ベクトル方向が直線方向である。

モーメント行列 $\begin{pmatrix} M_{xx} & M_{xy} \\ M_{xy} & M_{yy} \end{pmatrix}$ $M_{xx} = \sum_{i=1}^{n} w_i x_i^2$ $M_{yy} = \sum_{i=1}^{n} w_i y_i^2$ $M_{xy} = \sum_{i=1}^{n} w_i x_i y_i$

mxy = N[a000*Cos[0^o] *Sin[0^o] + a015*Cos[15^o] *Sin[15^o] + …….
+ a165* Cos[165^o] *Sin[165^o] + a000* Cos[180^o] *Sin[180^o] +
a015* Cos[195^o] *Sin[195^o] + ……. + a165* Cos[345^o] *Sin[345^o]];

	w_i: 出力ノルム値,	x_i: x座標値,	y_i: y座標値
i=1	a000	Cos(0^o)	Sin(0^o)
i=2	a015	Cos(15^o)	Sin(15^o)
i=3	a030	Cos(30^o)	Sin(30^o)
	:		
i=24	a165	Cos(345^o)	Sin(345^o)

Mathematica 演習：特定方向の線の認識

モーメント行列の固有ベクトルを求める。モーメント行列の最小固有値に対応する固有ベクトル方向が直線方向である。

モーメント行列 $\begin{pmatrix} M_{xx} & M_{xy} \\ M_{xy} & M_{yy} \end{pmatrix}$ $M_{xx} = \sum_{i=1}^{n} w_i x_i^2$ $M_{yy} = \sum_{i=1}^{n} w_i y_i^2$ $M_{xy} = \sum_{i=1}^{n} w_i x_i y_i$

myy = N[a000*Sin[0^o] *Sin[0^o] + a015*Sin[15^o] *Sin[15^o] + …….
+ a165* Sin[165^o] *Sin[165^o] + a000* Sin[180^o] *Sin[180^o] +
a015* Sin[195^o] *Sin[195^o] + ……. + a165*Sin[345^o] *Sin[345^o]];

	w_i: 出力ノルム値,	x_i: x座標値,	y_i: y座標値
i=1	a000	Cos(0^o)	Sin(0^o)
i=2	a015	Cos(15^o)	Sin(15^o)
i=3	a030	Cos(30^o)	Sin(30^o)
	:		
i=24	a165	Cos(345^o)	Sin(345^o)

Mathematica 演習：特定方向の線の認識

モーメント行列の固有ベクトルを求める。モーメント行列の最小固有値に対応する固有ベクトル方向が直線方向である。

モーメント行列 $\begin{pmatrix} M_{xx} & M_{xy} \\ M_{xy} & M_{yy} \end{pmatrix}$ $M_{xx} = \sum_{i=1}^{n} w_i x_i^2$ $M_{yy} = \sum_{i=1}^{n} w_i y_i^2$

$$M_{xy} = \sum_{i=1}^{n} w_i x_i y_i$$

モーメント値の出力	Print[mxx]; Print[mxy]; Print[myy];
モーメント行列	mmtx ={{mxx,mxy},{mxy,myy}};
固有ベクトル	evec = Eigenvectors[mmtx];
角度の計算	Print[180.0*ArcTan[evec[[1,1]],evec[[1,2]]]/π];

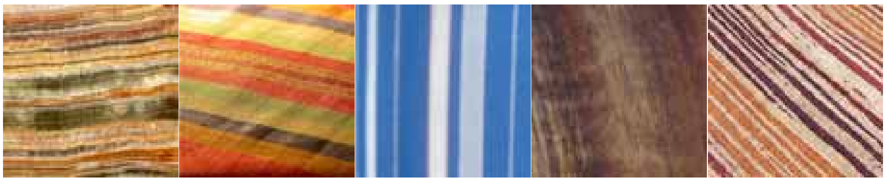
入力画像の線の方向をいろいろ変えて確かめよ。

テクスチャ解析

縞模様と水玉模様の識別

テクスチャ画像

- 縞模様と水玉模様のテクスチャ画像
  - 縞模様の画像例



stripe1.jpg stripe2.jpg stripe3.jpg stripe4.jpg stripe5.jpg

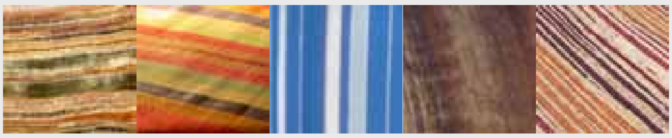
  - 水玉模様の画像例

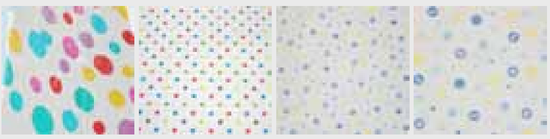
dot1.jpg dot2.jpg dot3.jpg dot4.jpg

画像認識ホームページよりダウンロードせよ。

テクスチャ画像認識：演習

- 縞模様と水玉模様のテクスチャ画像を識別せよ。
  - これらの画像を、縞模様か水玉模様かを認識するのは、どのようにすればよいか？

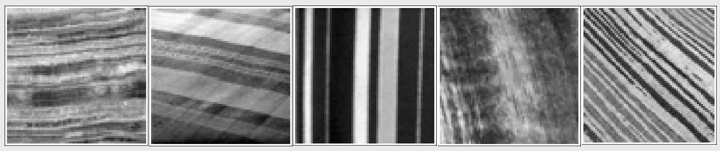




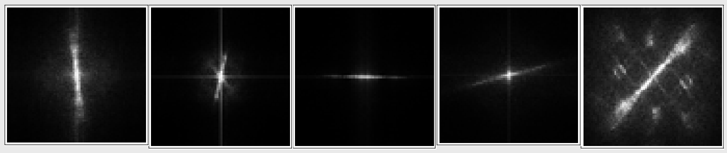
テクスチャ画像認識：演習

- 縞模様テクスチャ画像とそのフーリエ変換
  – フーリエ変換したときの特徴は？

入力画像



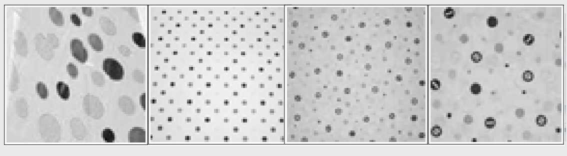
フーリエ変換



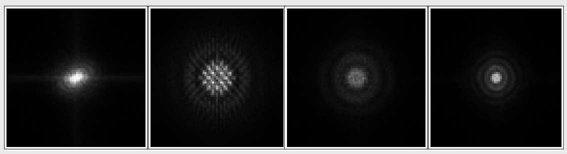
テクスチャ画像認識：演習

- 水玉模様テクスチャ画像とそのフーリエ変換
  – フーリエ変換したときの特徴は？

入力画像

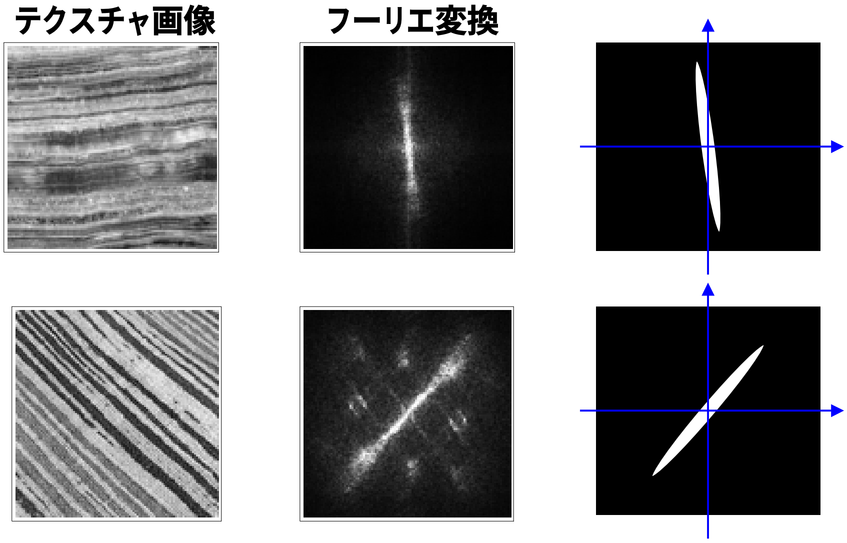


フーリエ変換



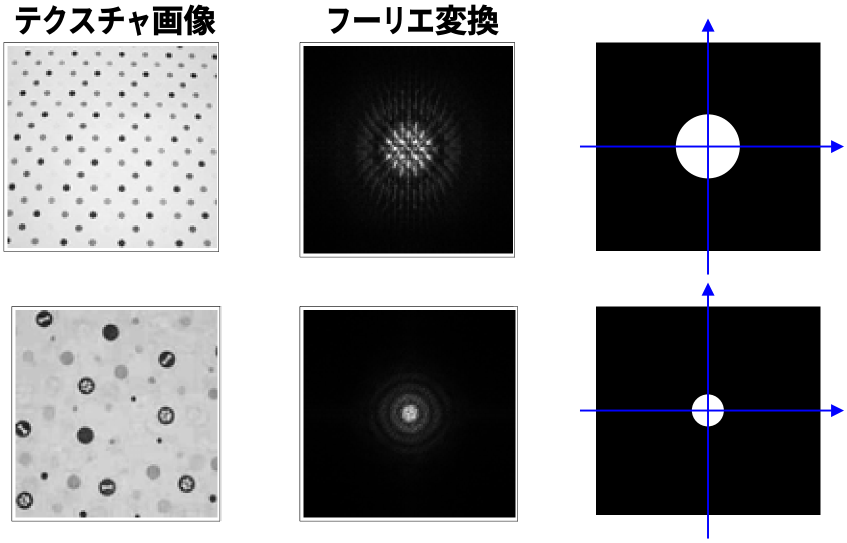
縞模様画像のフーリエ変換

テクスチャ画像　フーリエ変換



水玉模様画像のフーリエ変換

テクスチャ画像　フーリエ変換



テクスチャ画像のフーリエ変換と表示

テクスチャ画像(ＪＰＧカラー画像) の入力と表示 (カラー画像Gチャネルを白黒表示)

```
g = Import["/…./…./stripe1.jpg"];
im = g[[1,1]];
ListDensityPlot[im[[All,All,2]], Mesh->False,PlotRange->All];
```

テクスチャ画像のフーリエ変換と表示 (Gチャネルのフーリエ変換)

```
fim = Fourier[im[[All,All,2]];
ListDensityPlot[Abs[fmi], Mesh->False,PlotRange->{0,200}];
```

フーリエ変換画像の配置を変える

```
tfim1 = Join[Take[fim,-128,128],Take[fim,128,128]];
tfim2 = Join[Take[fim,-128,-128],Take[fim,128,-128]];
tfim = Transpose[Join[Transpose[tfim2],Transpose[tfim1]]];
ListDensityPlot[Abs[tfmi], Mesh->False,PlotRange->{0,200}];
```

テクスチャ認識：レポート課題

1. 縞模様、水玉模様の９枚のテクスチャ画像の12チャネル (15度間隔) Gaborフィルタ出力を計算し、プロットして表示せよ。
2. 12チャネルの出力値から、縞模様と水玉模様を識別する方法を考えて、その方法を試した結果を報告せよ。
3. 縞模様に関しては、縞模様の方向を求めよ。

- 提出方法
  - レポートは、ワードファイルやＰＤＦファイルで、メールの添付ファイルとして提出せよ。(Mathematicaのファイルも添付してよいが、その内容を説明する文書ファイルを付加すること。)
  - 提出先：yoshi@image.med.osaka-u.ac.jp
  - Subject (件名) は、「画像認識レポート (学籍番号)」とすること。
  - 提出期限：１月２７日 (金) 午後５時