目次

★ 8 パターン情報処理の応用例 (1) — フィルタリング
– 変数が 1 次元の信号の時間領域フィルタリング／周波数領域フィルタリング
– 画像の空間領域でのフィルタリング／周波数領域でのフィルタリング
★ 9 パターン情報処理の応用例 (2) — 画像処理いろいろ
– グレイスケール画像の階調変換としきい値処理

★8 パターン情報処理の応用例 (1) — フィルタリング

今回と次回は，パターン情報処理の応用例として，音響信号や画像のようなパターン情報を加工する (例：音響信号から雑音を除去する，画像をぼかす，etc.) 処理について考える．このような処理の手法には様々なものがあるが，今回はそれらの中でも特に**フィルタリング**と呼ばれる処理を取り上げる．フィルタリングは，パターンを構成する値を直接いじる「時間領域／空間領域 (☆ 1) でのフィルタリング」と，離散フーリエ変換などで求めたパターンの周波数成分を変化させて逆変換する「周波数領域でのフィルタリング」に大別できる．

フィルタリング： filtering

☆ 1) 音響信号のように変数が時間の場合は「時間領域」，画像のように変数が画素の位置を表す場合は「空間領域」でのフィルタリングという．

★8.1 変数が 1 次元の信号の時間領域フィルタリング

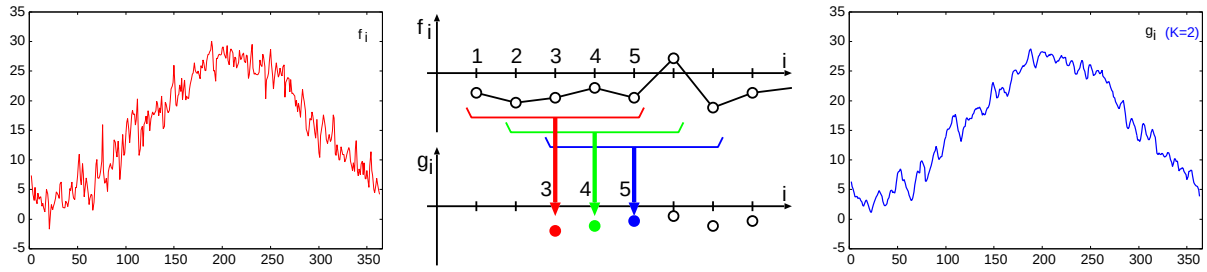
まずは，音響信号のように変数が 1 次元の離散信号の場合を考える．

★8.1.1 波形の平滑化

N 点から成る離散信号 $(f_1, f_2, \ldots, f_N)$ に対して，次のような計算を考えよう．

$$g_i = \frac{1}{5}(f_{i-2} + f_{i-1} + f_i + f_{i+1} + f_{i+2}) \tag{1}$$

g_i は，f_i の「端の方」，すなわち $i = 1, 2, N-1, N-2$ では求まらないが，それ以外の i に対しては計算することができる．以前にも登場した気温の信号を $\{f_i\}$ $(N = 366)$ として実際に $\{g_i\}$ を求めてみると，下図のような波形が得られる．



図からわかるように，式 (1) のような計算を行なうと，元の信号の細かい変動をならして大まかな変化をとらえやすくなる．このような処理を**平滑化**という．ある点とその前後の点（例では計 5 点）の値の平均を求める計算を，注目点を移動しながら繰り返すものとなっているので，**移動平均**をとるという言い方をすることもある．

平滑化: smoothing.
移動平均: moving average.

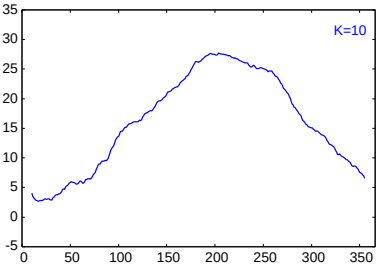
この平滑化の処理は，平均をとる点の数を一般化すると次のように表せる．

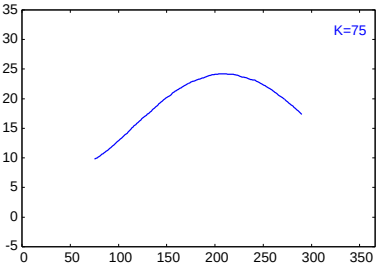
$$g_i = \frac{1}{2K+1} \sum_{j=-K}^{K} f_{i+j} \qquad (i = 1+K, 2+K, \ldots, N-K) \tag{2}$$

g_i は，i 番目の点とその前後 K 点の計 $2K+1$ 点の平均値である．
右図に，$K = 10, 75$ での平滑化の結果を示す．

平滑化の計算は，次式のように注目点以前の点の値のみを用いる場合もある．

$$g_i = \frac{1}{L}(f_i + f_{i-1} + \cdots + f_{i-L+1})$$





式 (2) の方法では，g_i を求めるために，より先の $f_{i+1}, \ldots, f_{i+K}$ の値が必要となるので，しばらく待たないと g_i を計算できない．しかし，上記のようにすると f_i が得られた時点ですぐに g_i を計算できるので，時々刻々やってくるデータを次々と処理するような場合に時間遅れを生じさせずに済むという利点がある．

★ 8.1.2 時間領域でのフィルタリング

式 (2) の平滑化の処理を一般化してみよう．次の式を考える．

$$g_i = \sum_{j=-K}^{K} w_j f_{i+j} \qquad (i = 1+K, 2+K, \ldots, N-K) \tag{3}$$

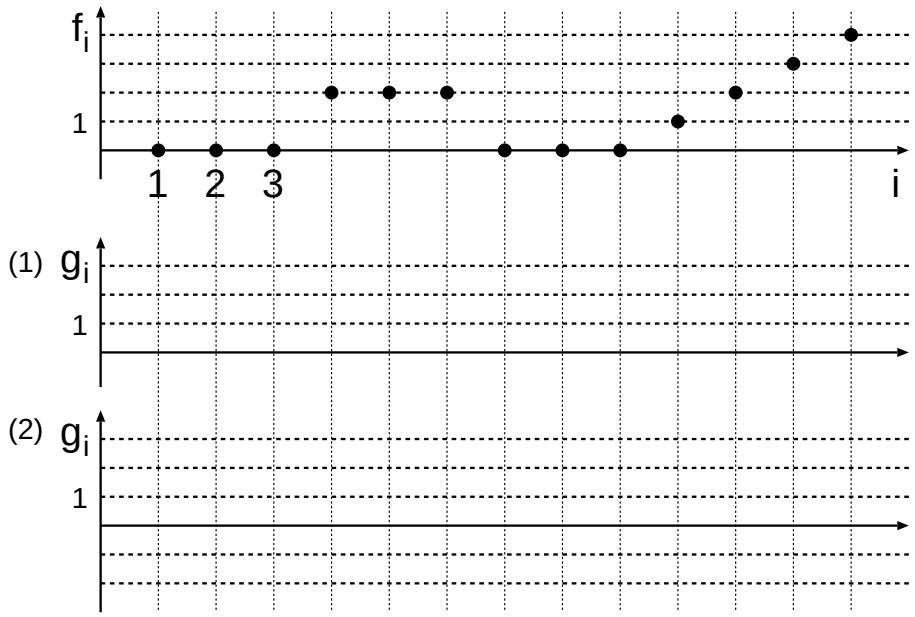
この式で $w_{-K} = \cdots = w_0 = \ldots w_K = \frac{1}{2K+1}$ とおけば，式 (2) と同じものになる．しかし，$\{w_j\}$ を様々に設定することで，平滑化とは異なる処理を実現することもできる．このような処理を**フィルタリング**といい，$\{w_j\}$ を**フィルタ**という (☆ 2)．また，式 (3) のような演算のことを**畳み込み**ということもある．

☆ 2) マスク，オペレータ，またはカーネルと呼ぶこともある．
畳み込み: convolution.

Q1. 下図に示す信号 $\{f_i\}$ に対して，式 (3) によるフィルタリングの結果 $\{g_i\}$ はどのようになるか計算し図示しなさい．ただし，$K = 1$ とし，$\{w_j\}$ の値は次のように定めるものとする．

(1) $w_{-1} = w_0 = w_1 = \frac{1}{2K+1}$

(2) $w_{-1} = -1, w_0 = 1, w_1 = 0$



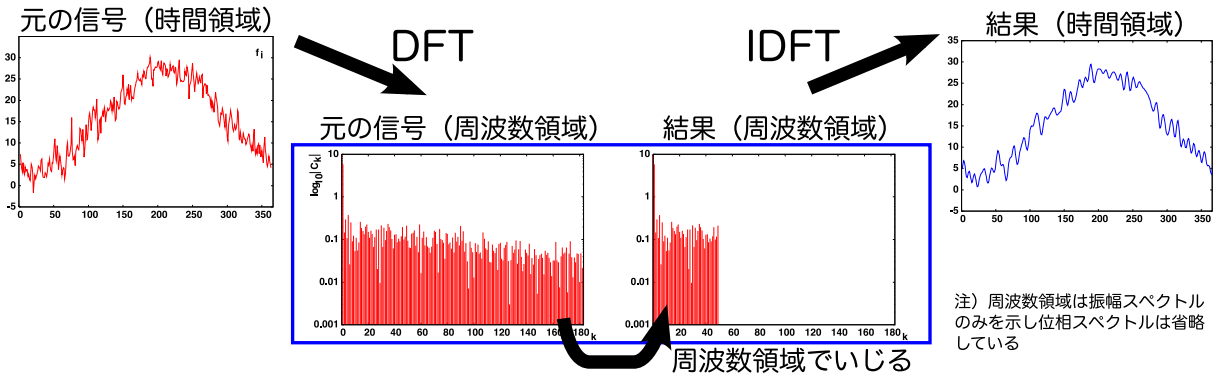
上記の問の答からも分かるように，$\{w_j\}$ の設定次第で，平滑化のように信号の大まかな変動（低い周波数の成分）のみを抽出したり，逆に激しい変動（高い周波数の成分）のみを抽出したりと，様々に信号／パターンを加工することができる (☆ 3)．また，この授業では説明しないが，フィルタリングとしては，式 (3) のような単純な線形和だけでなく，もっと複雑な演算を用いることもある．

☆ 3) 「ディジタル信号処理」等を勉強すると，$\{w_j\}$ の設定法やその数学的背景について学ぶことができるだろう．

★ 8.2　変数が 1 次元の信号の周波数領域フィルタリング

上述のフィルタリングの処理では信号（パターン） の値を直接いじって加工していたが，信号をいったん離散フーリエ変換（あるいは類似の変換を）してスペクトルを求め，そのスペクトルをいじってから逆変換することでも，同様のことができる（下図参照）．

DFT は，信号を，時間や空間を変数とする世界（時間領域や空間領域）から周波数を変数とする世界（周波数領域）へと変換する操作である．これまでの授業でみてきたように，信号は周波数成分に分解して表現した方がその性質をとらえやすい場合が多いので，周波数領域で加工（周波数領域でのフィルタリング）する方が，時間領域で直接加工（時間領域でのフィルタリング）するよりも便利なことが多い．



注）周波数領域は振幅スペクトルのみを示し位相スペクトルは省略している

上記の図は，時間領域での平滑化と同様の処理を，周波数領域で行なう過程を示したものである．左端に示す信号を DFT すると，フーリエ係数が得られる（2 つ目の図にスペクトルを示す (☆ 4)）．フーリエ係数は，対応する波の成分の周波数が低いものから順にならんでいるので，適当な範囲の係数の値を全て 0 にしてやることで，ある周波数より高い周波数の成分を取り除いてしまうことができる (☆ 5)．このように加工したフーリエ係数を IDFT すると，図右端に示すように，元の信号から高い周波数の成分を取り除いた信号が得られる．この図の例では，基本周期 366 日 ($k = 1$) の信号で $k = 50$ より大きい成分を取り除いているので，結果として得られる信号には，周期が $366/50 = 7.32$ 日以上の成分のみが含まれている．

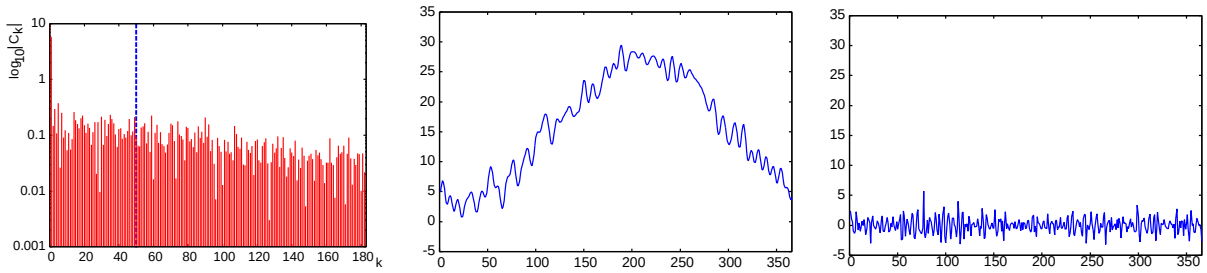
☆ 4) 図の縦軸はフーリエ係数の絶対値そのものではなく，その対数をとった値を示している．

☆ 5) フーリエ係数の性質から，実際には番号の大きい方も残すことになる．詳細は信号処理等の参考書を参照．

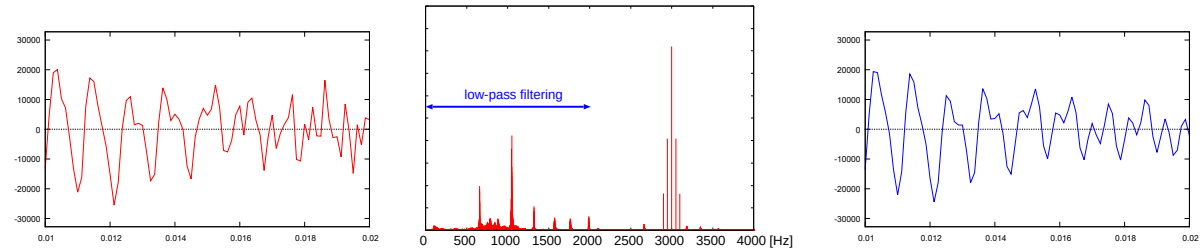
このように，ある周波数以下の成分を残してそれより高い周波数の成分を取り除くことを，**ローパスフィルタリング**という．逆に，ある周波数以上の成分を残してそれより低い周波数の成分を取り除くことを**ハイパスフィルタリング**，一定範囲の周波数成分のみを残すことを**バンドパスフィルタリング**という．

ローパスフィルタリング: low-pass filtering. ローパスフィルタ（低域通過フィルタ）をかけるとも．ハイパスフィルタリング: high-pass filtering. ハイパスフィルタ（高域通過フィルタ）をかけるとも． バンドパスフィルタリング: band-pass filtering. バンドパスフィルタ（帯域通過フィルタ）をかけるとも．

下図に，気温の信号をローパスフィルタリングした結果（真ん中）とハイパスフィルタリングした結果（右）の例を示す．フィルタリングの基準は図左のスペクトルに破線で示されている．上述のように，真ん中の信号は，周期がおよそ 7 日より長い，ゆっくりした気温変化のみをとらえており，逆に左の信号は短い周期での気温変化のみを表している．



このようなフィルタリングの処理は，信号から雑音（ノイズ）を取り除くためによく用いられる．下図に示したのは，音響信号にローパスフィルタリングを行なって高周波数の雑音成分を除去する過程の例である．



ここでは，周波数領域で特定の成分を 0 にするというフィルタリング処理のみを取り上げたが，スペクトルにより複雑な加工をほどこすことで，様々な信号処理が可能となることがわかるだろう．

★ 8.3　画像の空間領域でのフィルタリング

次に，変数が多次元の場合の代表例として，画像の空間領域フィルタリングについて説明する．

一般的なグレイスケール画像では，図 1 に示すように画素が正方格子上に縦横にならんでいる (☆ 6)．この場合，画像 $\{f_{x,y}\}$ に対する空間領域でのフィルタリング処理（のうち単純なもの）は，次式のように表すことができる．

$$g_{x,y} = \sum_i \sum_j w_{i,j} f_{x+i,y+j} \tag{4}$$

1 次元信号の場合と同様に，$\{w_{i,j}\}$ の選び方次第で多様な処理が可能である．$\{w_{i,j}\}$ のことを**フィルタ**と呼ぶのは 1 次元信号の場合と同じである．

☆ 6) これとは異なる画素配列（六方格子など）をとる画像もあるし，画素の間隔が縦横で異なる場合もある（CCD センサから得た生の画像データなど）．

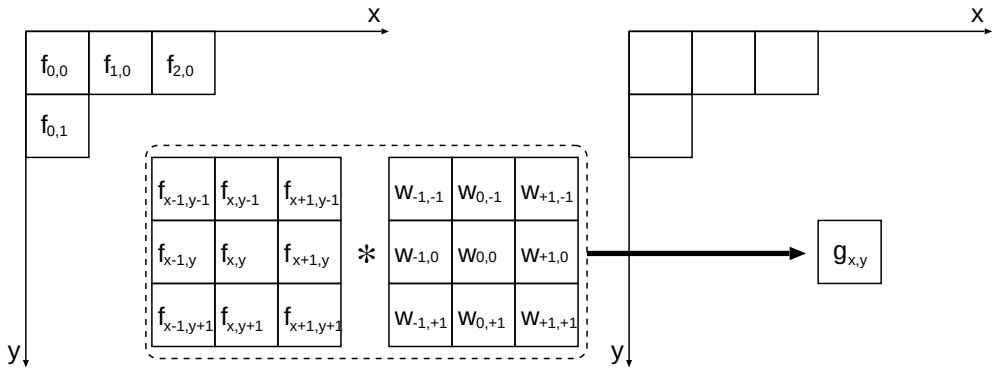


図 1: 画像のフィルタリング．$f_{x,y}$ と $g_{x,y}$ は入力と出力の位置 (x, y) の画素値を表し，$w_{i,j}$ はフィルタの値を表す．この例では，フィルタのサイズは 3×3 である．記号 $*$ は，式 (4) の畳み込み演算を表す．

★ 8.3.1 画像の平滑化

1 次元信号の場合と同様に，空間領域でのフィルタリングによって画像を平滑化することができる．右に示したのは，5×5 の平滑化フィルタの一例である．図 2 に示すように画像をぼかすのに使える．

$$\frac{1}{25}\begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

このフィルタリングは 5×5 の領域内の画素値の平均を求める演算に等しいが，平均値のかわりにメディアン（中央値）を求めるようにすると，明暗の境界をあまりぼかさずに画像を平滑化することができる（当然，このフィルタリング処理は式 (4) では表せない）．

このような平滑化処理は，画像からノイズを除去するためによく用いられる．

図 2: 画像の平滑化の例．左: 入力（下段の画像にはノイズがのっている），中: 5×5 平均による平滑化の結果，右: 5×5 メディアンフィルタによる平滑化の結果．

★ 8.3.2　エッジの検出

画像の中で明暗がステップ状に急激に変化する所を**エッジ**という．物体と背景や物体と物体の境界などはエッジになっていることが多いと考えられるので，画像の中からエッジ部分を見つける処理は重要である．

エッジ: edge. へり，ふち.

エッジは「画素値の変化の大きい所」＝「位置を変数とした画像の微分の値（絶対値）が大きい所」であるから，そのような微分演算に相当するフィルタリング処理によって浮かび上がらせることができる．図 3 は，エッジ検出に用いられる代表的なフィルタである Sobel フィルタ（右図）によるフィルタリングの例である．フィルタの方向に応じた輪郭部分が検出されていることがわかる．

-1	0	1
-2	0	2
-1	0	1

横方向の微分に相当する Sobel フィルタ

-1	-2	-1
0	0	0
1	2	1

縦方向の微分に相当する Sobel フィルタ

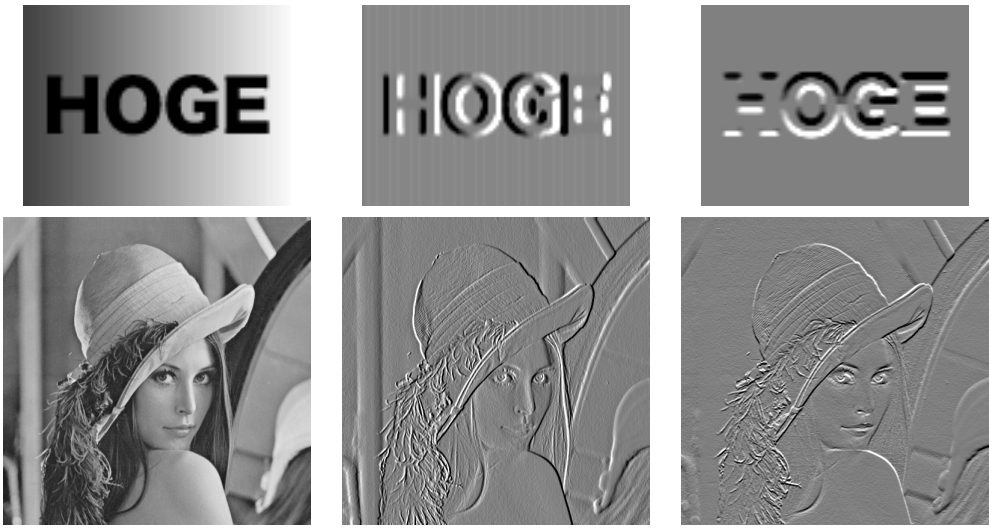


図 3: Sobel フィルタによるエッジ検出．左: 入力，中: 横方向の微分，右: 縦方向の微分．フィルタリング結果は，0 をグレーに対応させ，負ならそれより黒く，正ならより白く表している．

★ 8.4　画像の周波数領域でのフィルタリング

画像のような多次元の信号もフーリエ変換を利用して周波数領域で扱うことができる．詳しい説明は省略し，図のみを示す．

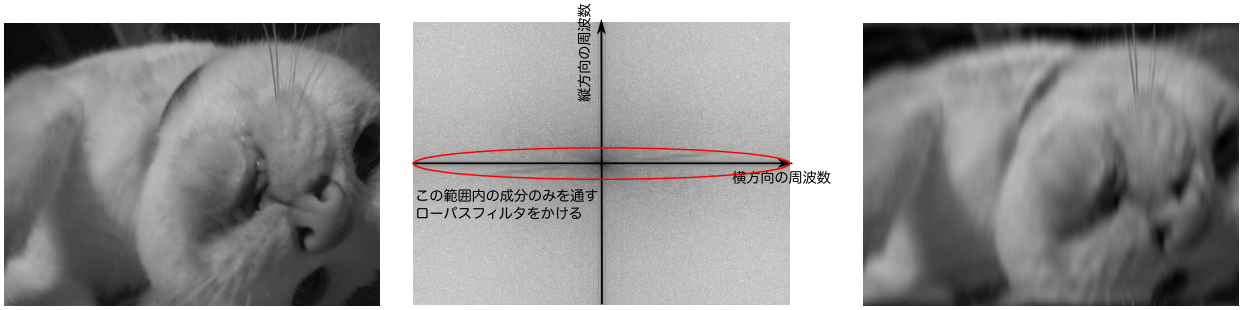


図 4: 周波数領域でのローパスフィルタリングの例．左から順に，元画像，その対数振幅スペクトル（黒いほどその位置に対応した周波数成分の振幅が大きい），フィルタリング結果．

★9　パターン情報処理の応用例 (2) — 画像処理いろいろ

画像を対象としてパターンを加工する処理を画像処理と呼ぶ．前回説明したフィルタリングも画像処理の手法の一つである．今回は，グレイスケール画像の画素値を変化させる方法 (**階調変換**) と，拡大縮小や回転のように画像の形を変える方法 (**幾何変換**) の一部 (画素値の補間と拡大縮小，アフィン変換) について述べる (☆ 7)．

☆ 7) この授業では，カラー画像，動画像，三次元画像等に関する話題は全て省略する．また，多岐にわたる画像処理とその応用のほんの一部しか紹介することができない．興味のある人は，この資料の末尾に記した参考文献等を参照してほしい．

★9.1　グレイスケール画像の階調変換としきい値処理

★9.1.1　画素値のヒストグラムと階調変換

画像の明るさを変化させる方法を考えよう．明るさの情報は，画素値の**ヒストグラム**を作ってみるとつかみやすい．例えば，図 5 左の暗い画像では小さな画素値が多い（この例では，画素値の最小値は 0（黒），最大値は 255（白）である）．この画像の画素値に一定数を加えると，図の真ん中に示すように明るい画像を作り出すことができる（この例では全ての画素値に 128 を加えている）．

ヒストグラム: histogram, 度数分布.

このように，画素値を変化させれば，画像の明るさを変化させることができる．このような操作を**階調変換**という．上記のように定数を加減するだけでなく，画素値に定数を掛けたり，非線形な関数を用いたりと様々な演算が利用される．図右の画像は，画素値の分布がなるべく一様分布に近づくようにヒストグラムを均等化（均一化ともいう）した結果である．明暗がはっきりして見やすくなっていることがわかる．

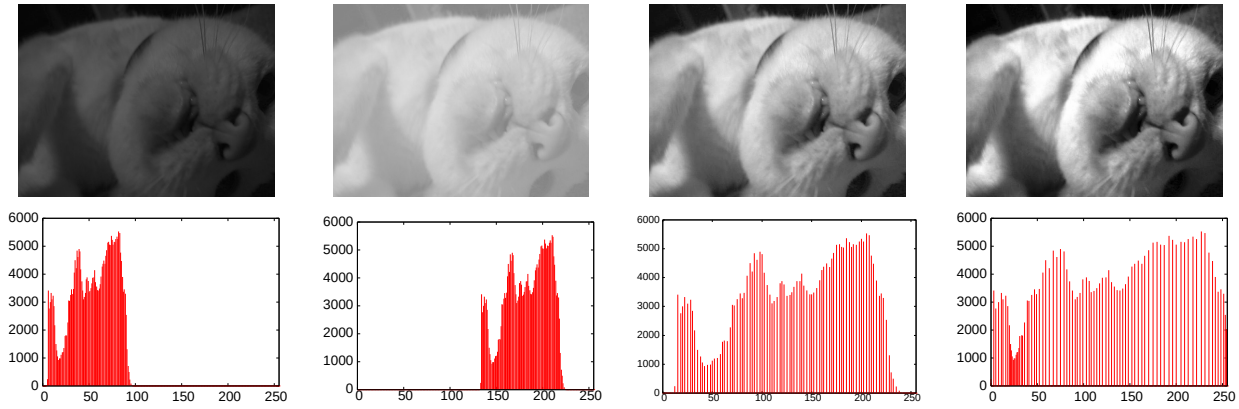


図 5: 画像と画素値のヒストグラム．上の画像の画素値のヒストグラムを下に示す．左から順に，元の画像，その画素値に定数を加えて作った画像，元の画素値に定数をかけて作った画像，ヒストグラムを均等化した画像.

★ 9.1.2　しきい値処理

階調変換の極端な場合として，画素値の基準（しきい値）を定め，画素値がしきい値以下ならその画素は真っ黒に，さもなくば真っ白にする，といった処理を考えることができる．このような処理を**しきい値処理**という．しきい値がひとつであれば画素値を 2 種類の値に分類することになる（この場合を特に**二値化**という）が，N 個あれば $N+1$ 値に分類することもできる．

しきい値: threshold.
二値化: binarization.

図 6 に，しきい値処理によって画像中の図と地を分離した例を示す．この例ではしきい値 (128 としている) を手動で設定しているが，実用的には，何らかの方法で自動的に決定する必要がある．そのため，画素値の分布などから適切なしきい値を求める手法や，位置毎にしきい値を適応的に変化させる手法など様々な手法が研究されている．

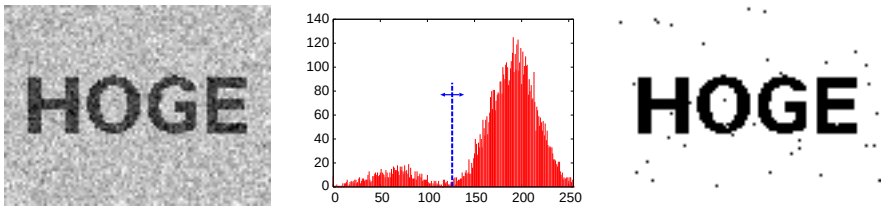


図 6: しきい値処理（二値化）の例．左から順に，元画像，そのヒストグラム，二値化した画像．

しきい値処理は，フィルタリングなどと組み合わせて画像の特徴をとらえるためにもよく用いられる．例えば，エッジ検出の結果をしきい値処理すれば，物体の輪郭線を抽出することができる．図 7 は，画像に

0	1	0
1	−4	1
0	1	0

というマスクを用いたフィルタリング (☆ 8) を行ってその値の絶対値をとり，その結果にしきい値処理を行ったものである．元の画像の中で画素値の変化が激しいところが黒く抜き出されている．

☆ 8) これは，画素値のラプラシアンに相当し，横方向の二階微分と縦方向の二階微分の和になっている．

図 7: フィルタリングとしきい値処理の組み合わせの例．左から順に，元画像，フィルタリングして絶対値をとったもの (見やすくするため白黒を反転してある)，それを二値化したもの．