国立大学法人 長岡技術科学大学 Nagaoka University of Technology

電気電子情報工学専攻　岩橋政宏
iwahashi@vos.nagaokaut.ac.jp

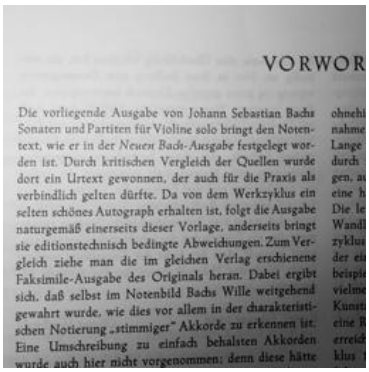
2．基本的な画像処理

2.1 二値化、モルフォロジー、エッジ検出

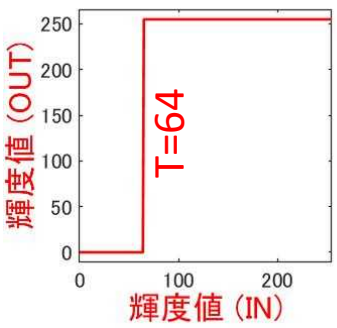
2.2 輝度補正、コントラスト調整、トーンマッピング

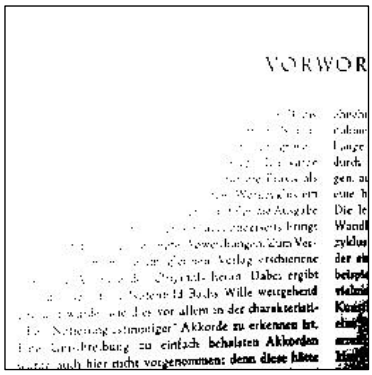
2.3 拡大、縮小、回転、歪み補正

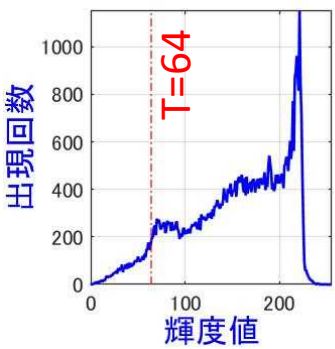
2.1 二値化、モルフォロジー、エッジ検出



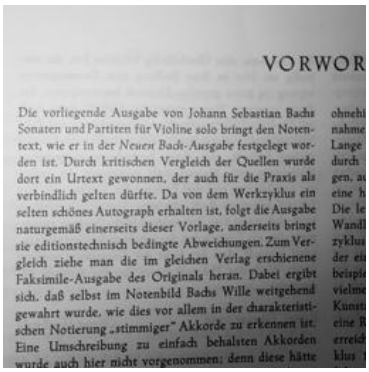
二値化（T=64）



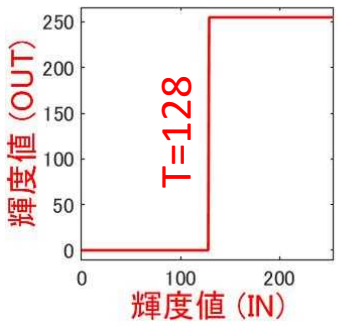


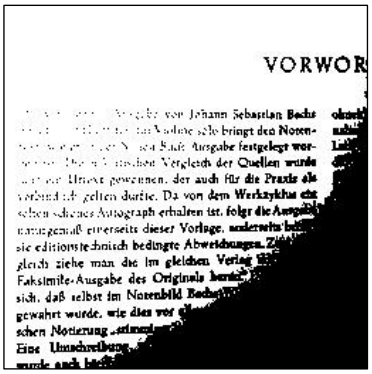


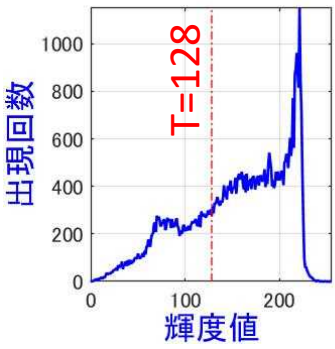
2.1 二値化、 モルフォロジー、エッジ検出



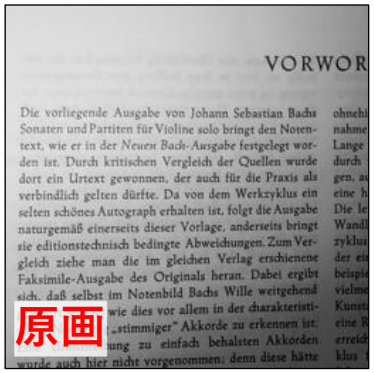
二値化（T=128）



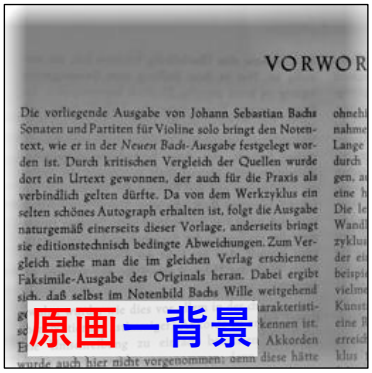




2.1 二値化、 モルフォロジー、エッジ検出

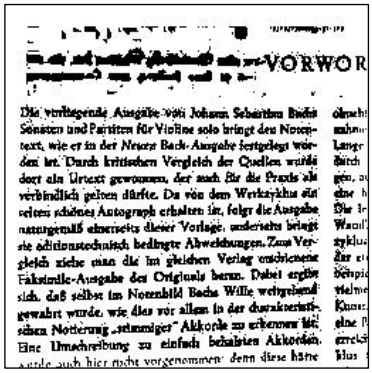


背景差分



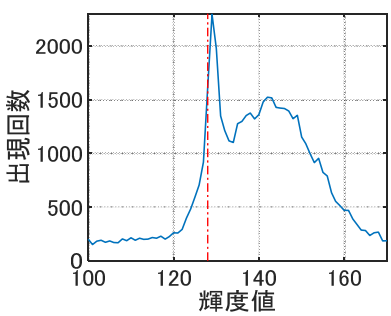
二値化

T=128



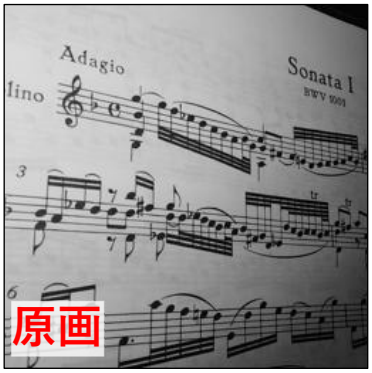
↑
画素＞T ならば 255
画素≦T ならば 0

平滑化

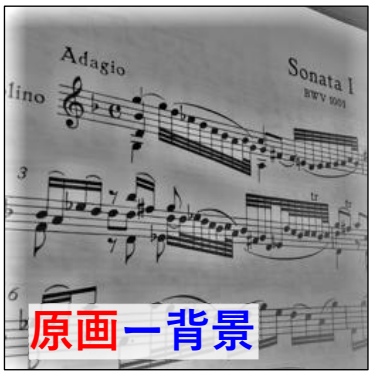


ヒストグラム

2.1 二値化、 モルフォロジー、エッジ検出

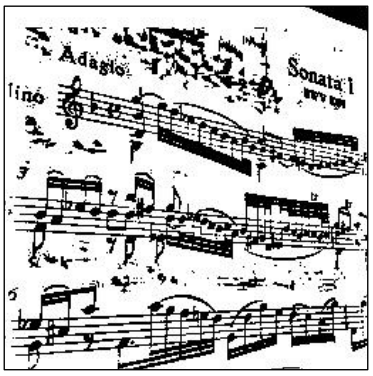


背景差分



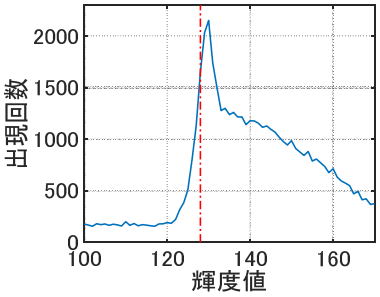
二値化

T=128



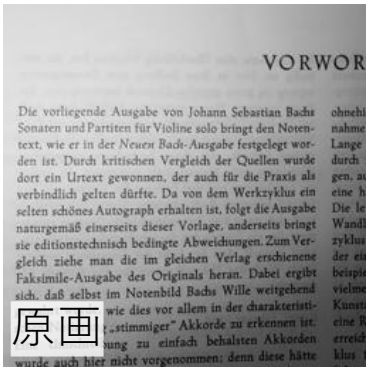
↑
画素＞T ならば 255
画素≦T ならば 0

平滑化

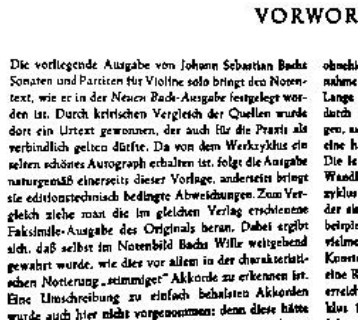


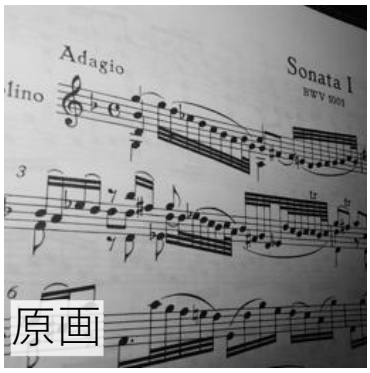
ヒストグラム

2.1 二値化、 モルフォロジー、エッジ検出

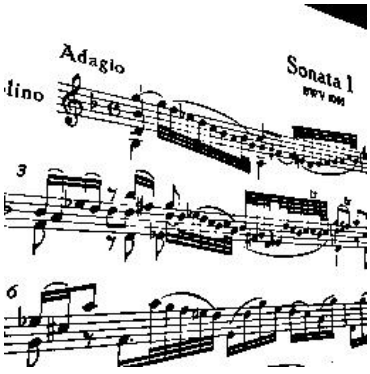


適応的二値化





適応的二値化



MATLAB: `I2=imbinarize(I0,'adaptive','ForegroundPolarity','dark','Sensitivity',0.5);`

2．基本的な画像処理

2.1 二値化、モルフォロジー、エッジ検出

2.2 輝度補正、コントラスト調整、トーンマッピング

2.3 拡大、縮小、回転、歪み補正

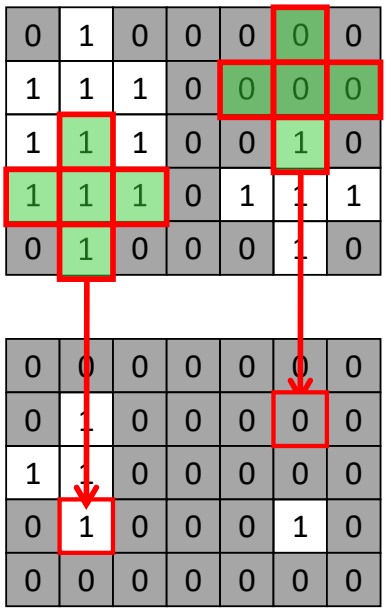
2.1 二値化、モルフォロジー、エッジ検出

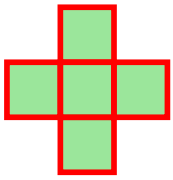
小さな**白い**点
が消滅した

白い線が
細すぎる

```
MATLAB
e = strel('square',3)
y = imerode(x,e)
```

「収縮」の演算





構造化要素
Structuring Element

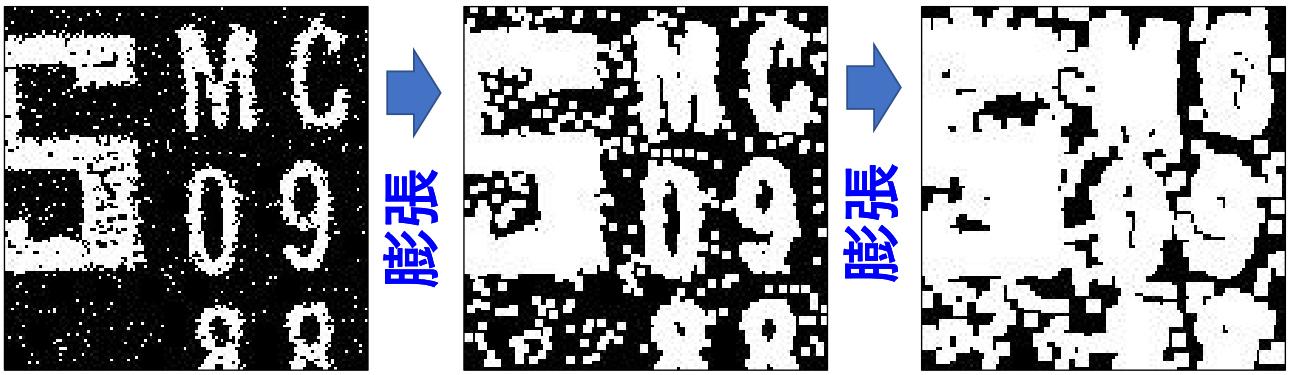
MATLAB
e = strel('diamond',1)

収縮 Erosion

・構造化要素の中に０が１つでもあれば，０を出力する

MATLAB
y = imerode(x,e)

2.1 二値化、モルフォロジー、エッジ検出

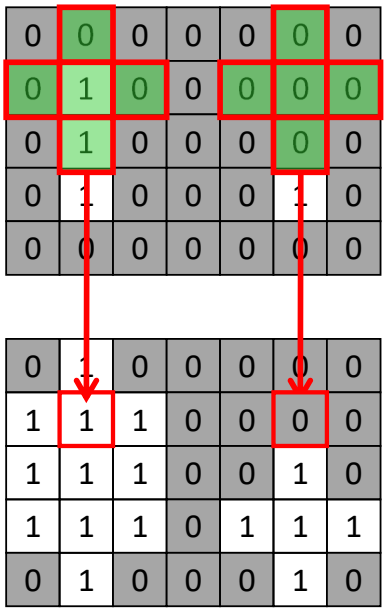


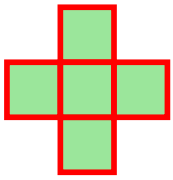
小さな**黒い**穴が塞がった

白い線が太すぎる

MATLAB
e = strel('square',3)
y = imdilate(x,e)

「膨張」の演算





構造化要素
Structuring Element

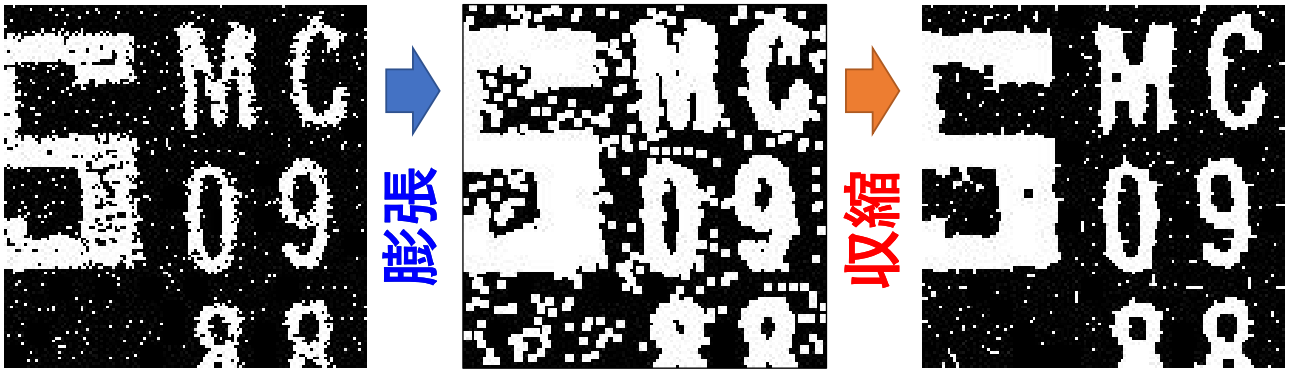
MATLAB
e = strel('diamond',1)

膨張 Dilation

・構造化要素の中に１が１つでもあれば，１を出力する

MATLAB
y = imdilate(x,e)

2.1 二値化、モルフォロジー、エッジ検出



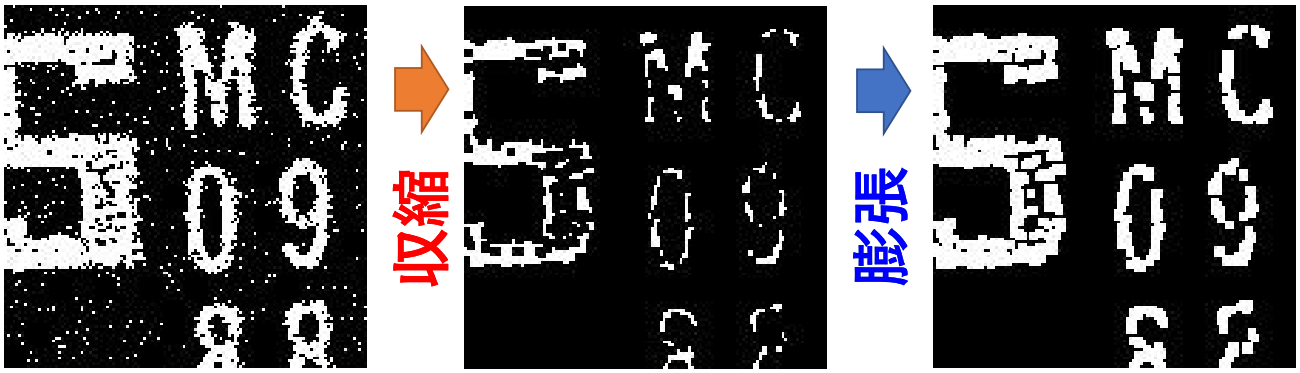
クロージング

小さな**黒い**穴が減少した

白い線の太さは同じ

MATLAB
e = strel('square',3)
y = imdilate(x,e)
y = imerode(y,e)

2.1 二値化、モルフォロジー、エッジ検出



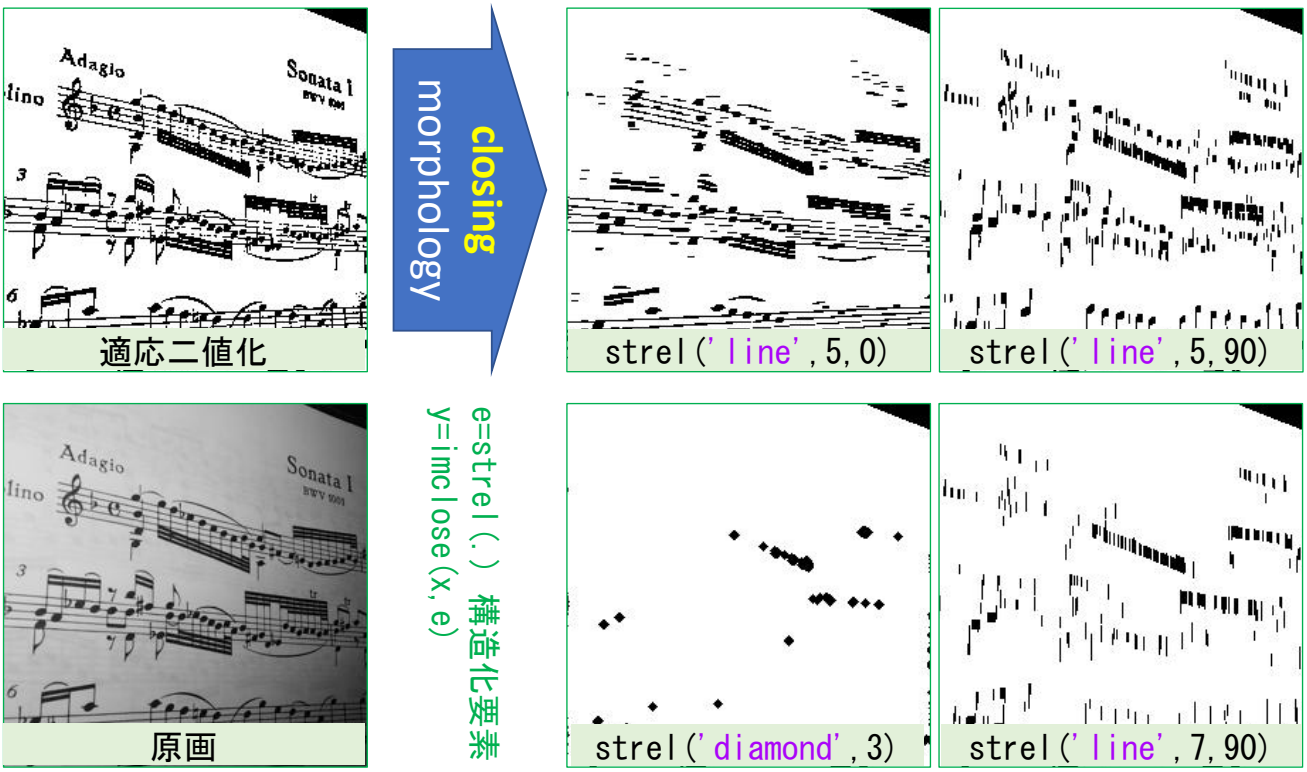
オープニング

小さな**白い**点が消滅した

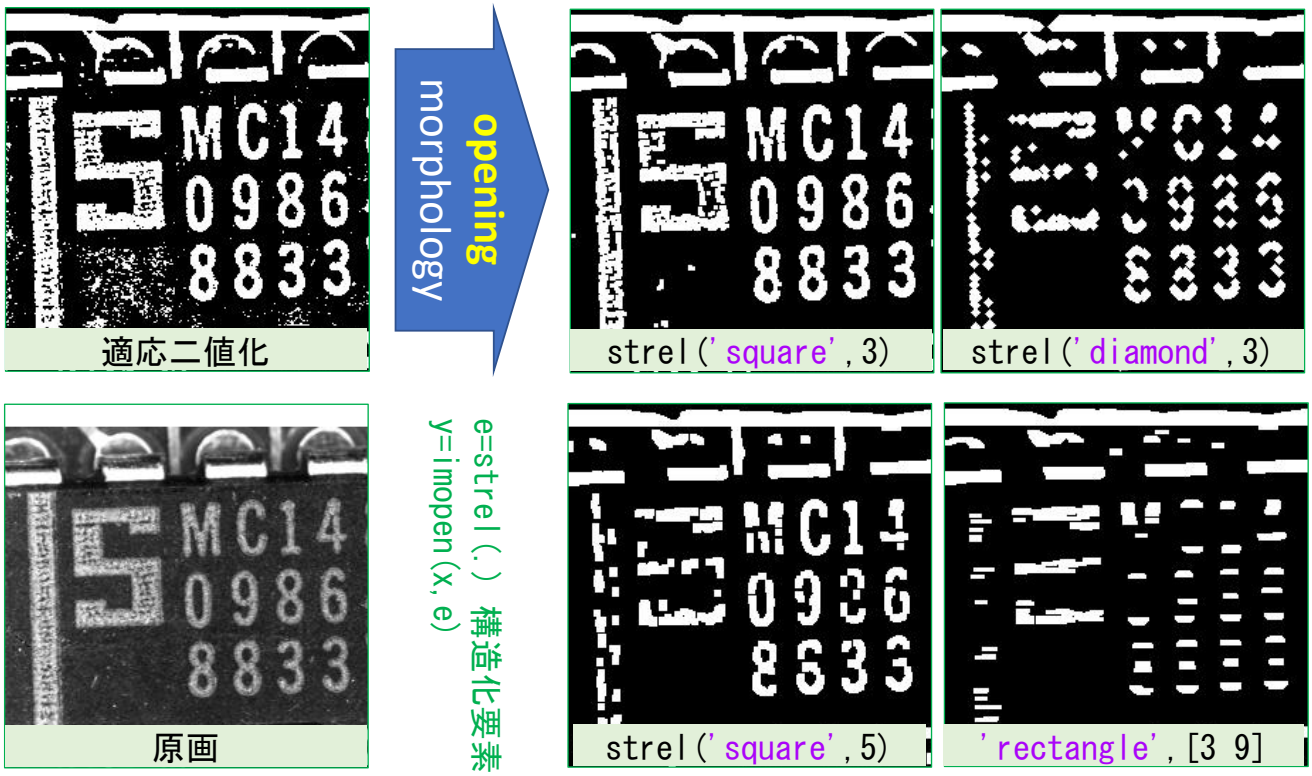
白い線の太さは同じ

```
MATLAB
e = strel('square',3)
y =  imerode(x,e)
y = imdilate(y,e)
```

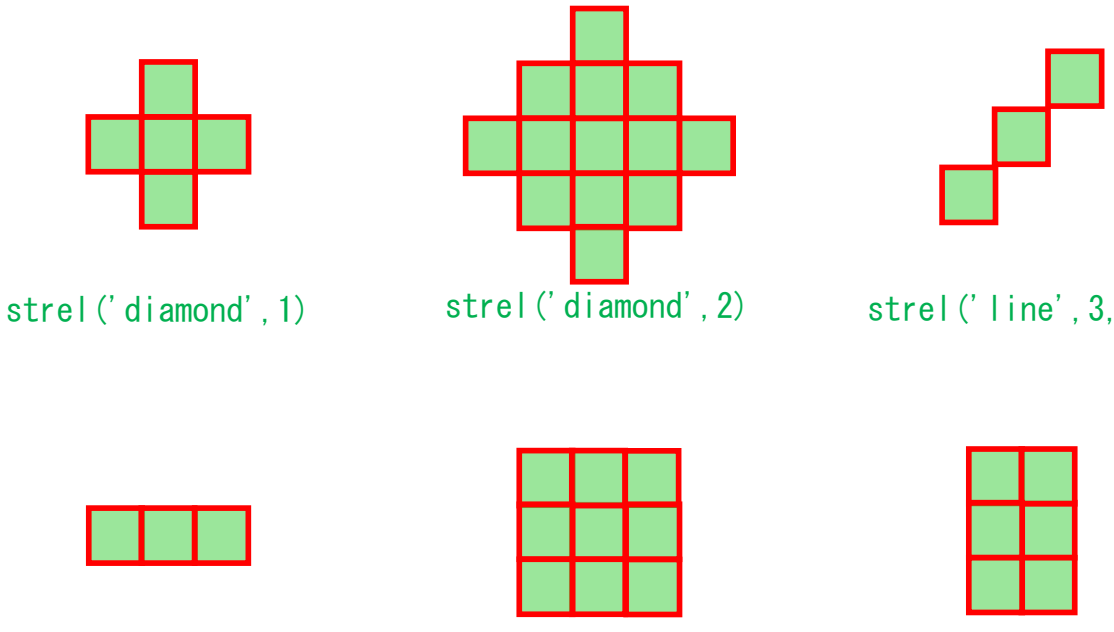
2.1 二値化、モルフォロジー、エッジ検出



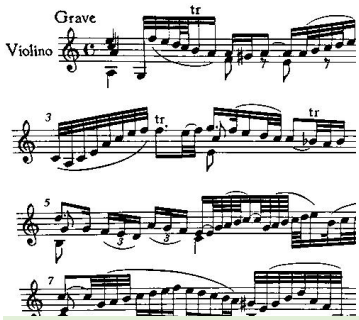
2.1 二値化、モルフォロジー、エッジ検出



構造化要素のバリエーション



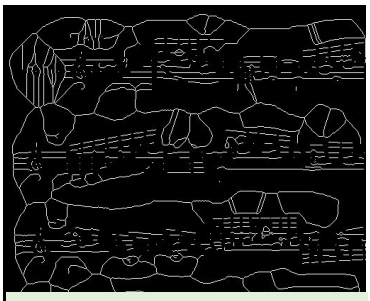
2.1 二値化、モルフォロジー、エッジ検出



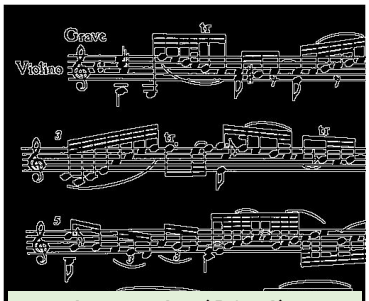
適応二値化

morphology

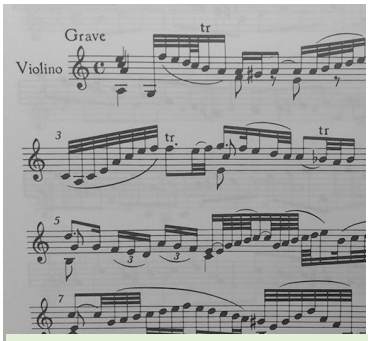
色々な



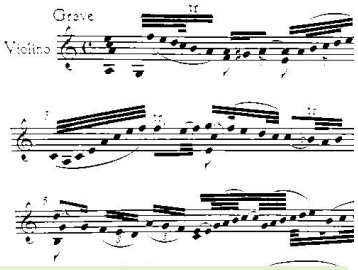
bwskel
骨格化



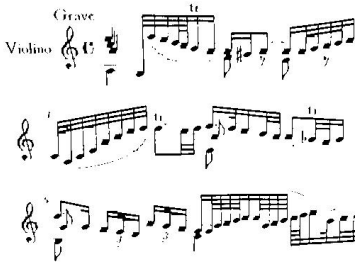
bwperim(I1,4)
周囲検出



原画

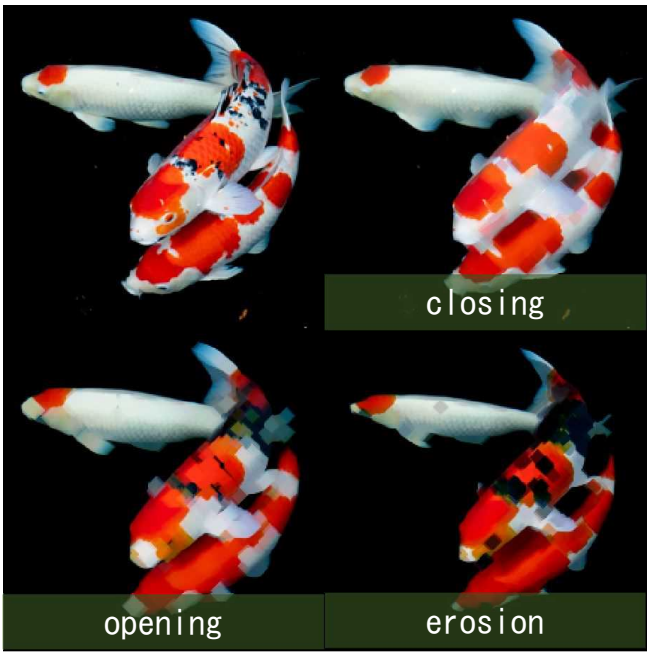


strel('line',3,00)
膨張



strel('line',3,90)
膨張

多値のモルフォロジー（二値化しない）



膨張(dilation)：構造化要素の中の**最大値**を出力する
収縮(erosion)：構造化要素の中の**最小値**を出力する

```
I0=imread('image.jpg');

SE = strel('diamond',5) ;
I1 = imclose(I0,SE);
I2 = imopen( I0,SE);
I3 = imerode(I0,SE);

montage({I0,I1,I2,I3});
```

2． 基本的な画像処理

2.1 二値化、モルフォロジー、エッジ検出

2.2 輝度補正、コントラスト調整、トーンマッピング

2.3 拡大、縮小、回転、歪み補正

2.1 二値化、モルフォロジー、エッジ検出

横の1階微分 $y_{i,j} = x_{i,j} - x_{i,j-1}$

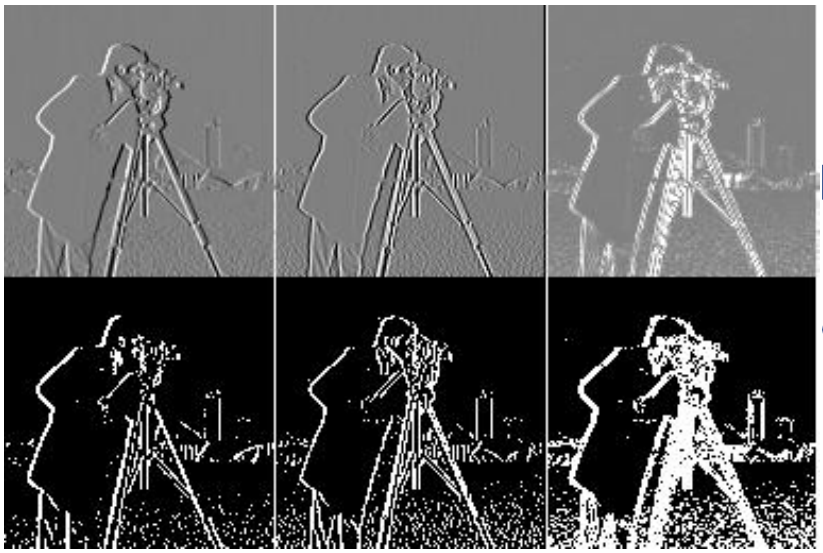
横の2階微分 $u_{i,j} = y_{i,j+1} - y_{i,j}$

2階微分の絶対値 $v_{i,j} = |u_{i,j}|$

↑ 原画像

$x_{i,j}$ は画素値$\in [0, 255]$

iは縦の位置、jは横の位置



二値化

上段の画像を → 二値化した

↑ 縦線と斜線が抽出された

2.1 二値化、モルフォロジー、エッジ検出

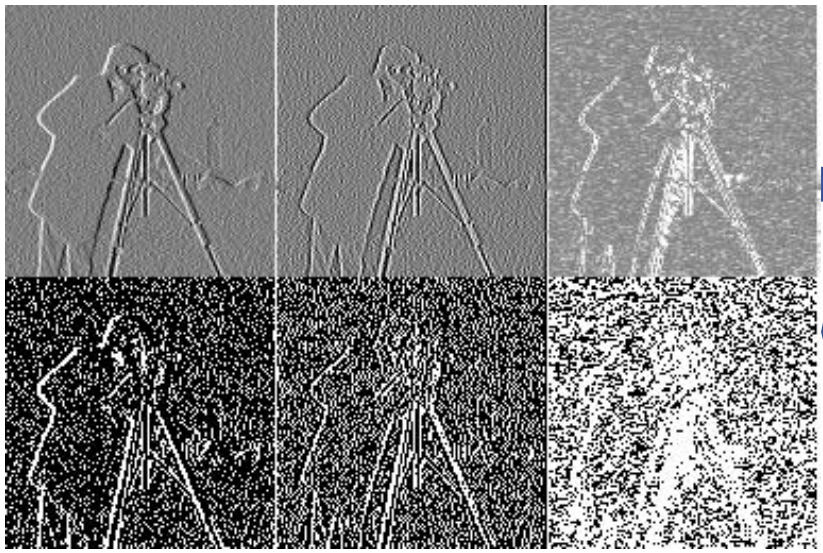
横の1階微分 $y_{i,j} = x_{i,j} - x_{i,j-1}$

横の2階微分 $u_{i,j} = y_{i,j+1} - y_{i,j}$

2階微分の絶対値 $v_{i,j} = |u_{i,j}|$

↑ 原画像

＋ノイズ



二値化

上段の画像を → 二値化した

↑ このままではノイズに弱い

2.1 二値化、モルフォロジー、エッジ検出

縦の平均値フィルタ

$$w_{i,j} = \frac{x_{i-1,j} + x_{i,j} + x_{i+1,j}}{3}$$

原画像＋ノイズ

横の 1 階微分

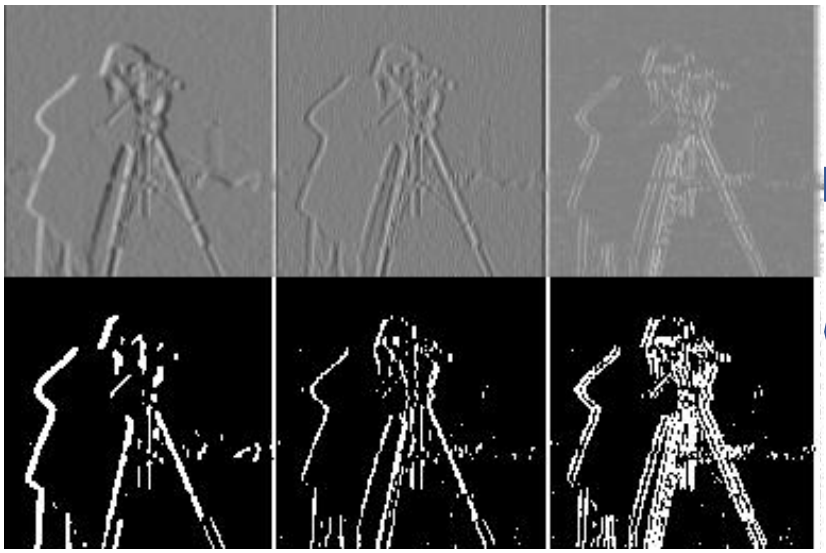
$$y_{i,j} = w_{i,j} - w_{i,j-1}$$

横の 2 階微分

$$u_{i,j} = y_{i,j+1} - y_{i,j}$$

2階微分の絶対値

$$v_{i,j} = |u_{i,j}|$$



二値化

↑ ノイズに強くなった

2.1 二値化、モルフォロジー、エッジ検出

横の 1 階微分

$$y_{i,j} = w_{i,j} - w_{i,j-1}$$

$$= \begin{bmatrix} 1 & -1 \end{bmatrix} \begin{bmatrix} w_{i,j} \\ w_{i,j-1} \end{bmatrix}$$

代入

横の 2 階微分

$$u_{i,j} = y_{i,j+1} - y_{i,j}$$

$$= \begin{bmatrix} 1 & -1 \end{bmatrix} \begin{bmatrix} y_{i,j+1} \\ y_{i,j} \end{bmatrix}$$

$$= w_{i,j+1} - w_{i,j} - w_{i,j} + w_{i,j-1}$$

$$= \begin{bmatrix} 1 & -2 & 1 \end{bmatrix} \begin{bmatrix} w_{i,j+1} \\ w_{i,j} \\ w_{i,j-1} \end{bmatrix}$$

横の 1 階微分

$$\Sigma \begin{bmatrix} 0 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \odot \begin{bmatrix} w_{i-1,j-1} & w_{i-1,j} & w_{i-1,j+1} \\ w_{i,j-1} & w_{i,j} & w_{i,j+1} \\ w_{i+1,j-1} & w_{i+1,j} & w_{i+1,j+1} \end{bmatrix}$$

横の 2 階微分

$$\Sigma \begin{bmatrix} 0 & 0 & 0 \\ 1 & -2 & 1 \\ 0 & 0 & 0 \end{bmatrix} \odot \begin{bmatrix} w_{i-1,j-1} & w_{i-1,j} & w_{i-1,j+1} \\ w_{i,j-1} & w_{i,j} & w_{i,j+1} \\ w_{i+1,j-1} & w_{i+1,j} & w_{i+1,j+1} \end{bmatrix}$$

↑ 要素の和

↑ アダマール積
（同じ位置の要素毎の積）

補足

横の 1 階微分

$$y_{i,j} = [1 \quad -1]\begin{bmatrix} w_{i,j} \\ w_{i,j-1} \end{bmatrix}$$

代入

$$= [-1 \quad 1 \quad 0]\begin{bmatrix} w_{i,j-1} \\ w_{i,j} \\ w_{i,j+1} \end{bmatrix}$$

←

重心が**左へ**
1/2画素分
だけ**ズレる**

横の 2 階微分

$$u_{i,j} = [1 \quad -1]\begin{bmatrix} y_{i,j+1} \\ y_{i,j} \end{bmatrix}$$

$$= [0 \quad -1 \quad 1]\begin{bmatrix} y_{i,j-1} \\ y_{i,j} \\ y_{i,j+1} \end{bmatrix}$$

→

重心が**右へ**
1/2画素分
だけ**ズレる**

$$= [1 \quad -2 \quad 1]\begin{bmatrix} w_{i,j+1} \\ w_{i,j} \\ w_{i,j-1} \end{bmatrix}$$

$$= [1 \quad -2 \quad 1]\begin{bmatrix} w_{i,j-1} \\ w_{i,j} \\ w_{i,j+1} \end{bmatrix}$$

↑

重心が
ズレない

フィルタのサイズと位置ズレ

$$y_i = x_i - x_{i-1} = [-1 \quad 1]\begin{bmatrix} x_{i-1} \\ x_i \end{bmatrix}$$

← 2 →
フィルタサイズ

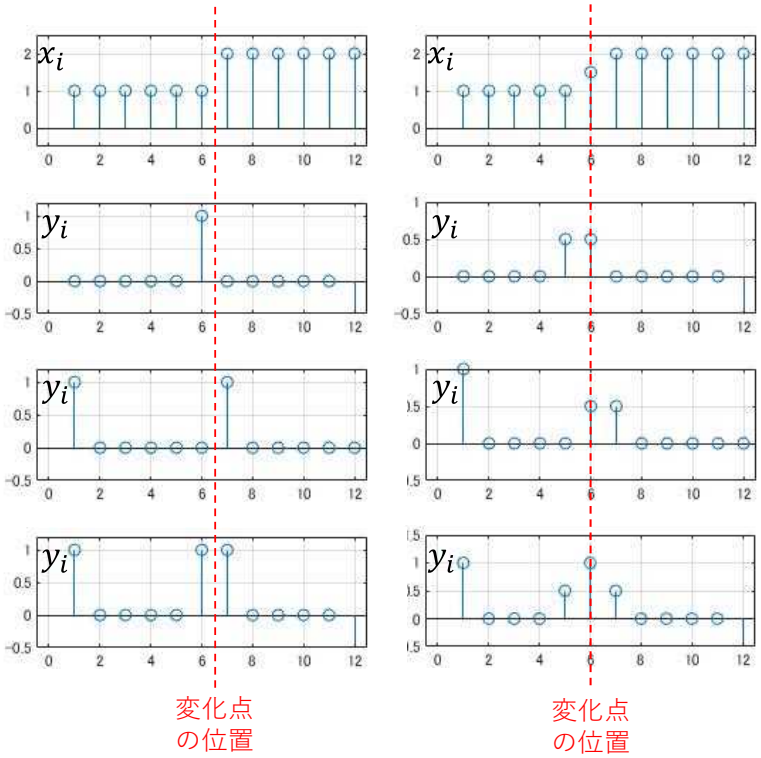
$$y_i = x_{i+1} - x_i = [-1 \quad 1]\begin{bmatrix} x_i \\ x_{i+1} \end{bmatrix}$$

← 2 →
フィルタサイズ

$$y_i = -x_{i-1} + x_{i+1}$$

$$= [-1 \quad 0 \quad 1]\begin{bmatrix} x_{i-1} \\ x_i \\ x_{i+1} \end{bmatrix}$$

← 3 →
フィルタサイズ



2.1 二値化、モルフォロジー、エッジ検出

縦の平均

$$w_{i,j} = (x_{i-1,j} + x_{i,j} + x_{i+1,j})/3$$

$$= \frac{1}{3}\begin{bmatrix} x_{i-1,j} & x_{i,j} & x_{i+1,j} \end{bmatrix}\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

横の 1 階微分

$$y_{i,j} = w_{i,j} - w_{i,j-1}$$

$$= \begin{bmatrix} 1 & -1 \end{bmatrix}\begin{bmatrix} w_{i,j} \\ w_{i,j-1} \end{bmatrix}$$

横の 2 階微分

$$u_{i,j} = y_{i,j+1} - y_{i,j}$$

$$= \begin{bmatrix} 1 & -1 \end{bmatrix}\begin{bmatrix} y_{i,j+1} \\ y_{i,j} \end{bmatrix}$$

縦の平均 & 横の 2 階微分

$$\frac{1}{3}\boldsymbol{\Sigma}\begin{bmatrix} 1 & -2 & 1 \\ 1 & -2 & 1 \\ 1 & -2 & 1 \end{bmatrix} \odot \begin{bmatrix} x_{i-1,j-1} & x_{i-1,j} & x_{i-1,j+1} \\ x_{i,j-1} & x_{i,j} & x_{i,j+1} \\ x_{i+1,j-1} & x_{i+1,j} & x_{i+1,j+1} \end{bmatrix}$$

補足

縦の平均

$$w_{i,j} = \frac{1}{3}\begin{bmatrix} x_{i-1,j} & x_{i,j} & x_{i+1,j} \end{bmatrix}\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

横の 2 階微分

$$u_{i,j} = \begin{bmatrix} 1 & -1 \end{bmatrix}\begin{bmatrix} y_{i,j+1} \\ y_{i,j} \end{bmatrix}$$

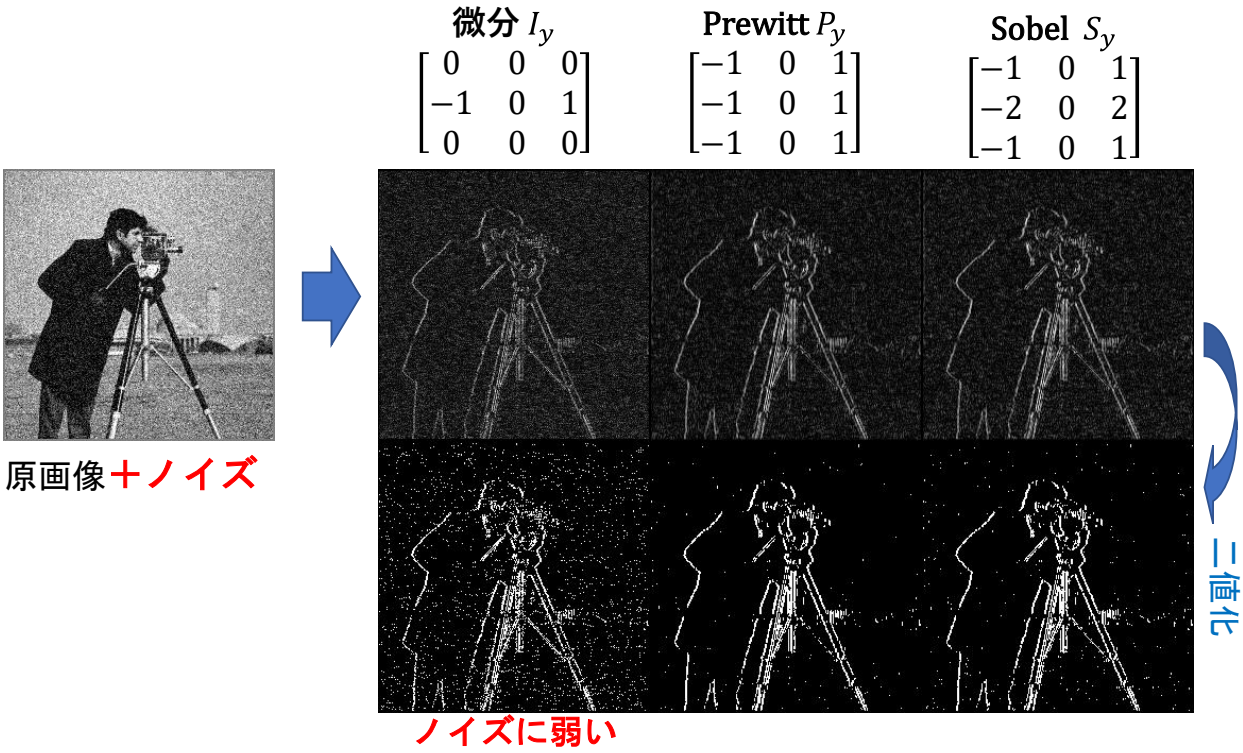
$$= \begin{bmatrix} 1 & -2 & 1 \end{bmatrix}\begin{bmatrix} w_{i,j-1} \\ w_{i,j} \\ w_{i,j+1} \end{bmatrix}$$

$$u_{i,j} = \begin{bmatrix} 1 & -2 & 1 \end{bmatrix}\begin{bmatrix} \frac{1}{3}\begin{bmatrix} x_{i-1,j-1} & x_{i,j-1} & x_{i+1,j-1} \end{bmatrix}\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \\ \frac{1}{3}\begin{bmatrix} x_{i-1,j} & x_{i,j} & x_{i+1,j} \end{bmatrix}\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \\ \frac{1}{3}\begin{bmatrix} x_{i-1,j+1} & x_{i,j+1} & x_{i+1,j+1} \end{bmatrix}\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \end{bmatrix} = \frac{1}{3}\left\{ \begin{bmatrix} x_{i-1,j-1} & x_{i,j-1} & x_{i+1,j-1} \end{bmatrix}\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} + \begin{bmatrix} x_{i-1,j} & x_{i,j} & x_{i+1,j} \end{bmatrix}\begin{bmatrix} -2 \\ -2 \\ -2 \end{bmatrix} + \begin{bmatrix} x_{i-1,j+1} & x_{i,j+1} & x_{i+1,j+1} \end{bmatrix}\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \right\} = \frac{1}{3}\left\{ \begin{matrix} 1\,x_{i-1,j-1} & +1\,x_{i,j-1} & +1\,x_{i+1,j-1} \\ -2\,x_{i-1,j} & -2\,x_{i,j} & -2\,x_{i+1,j} \\ +1\,x_{i-1,j+1} & +1\,x_{i,j+1} & +1\,x_{i+1,j+1} \end{matrix} \right\}$$

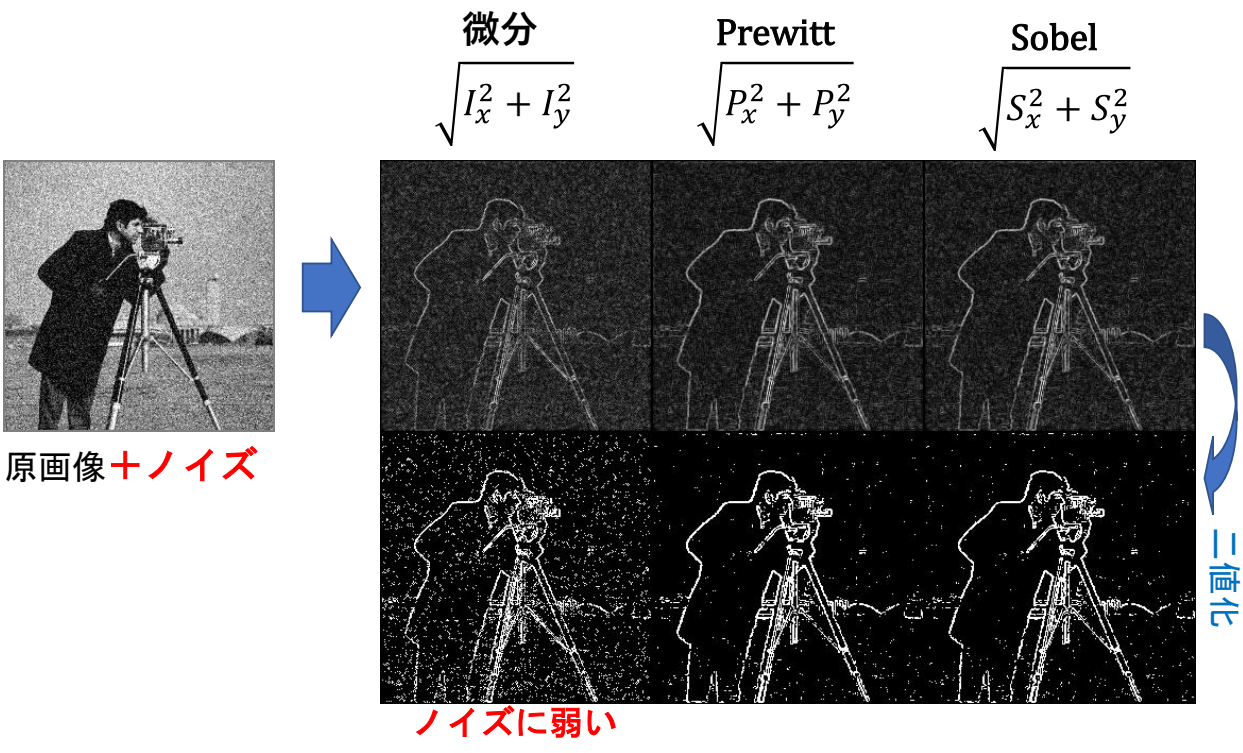
縦の平均 & 横の 2 階微分

$$u_{i,j} = \frac{1}{3}\left\{ \begin{matrix} 1\,x_{i-1,j-1} & -2\,x_{i-1,j} & +1\,x_{i-1,j+1} \\ +1\,x_{i,j-1} & -2\,x_{i,j} & +1\,x_{i,j+1} \\ +1\,x_{i+1,j-1} & -2\,x_{i+1,j} & +1\,x_{i+1,j+1} \end{matrix} \right\}$$

2.1 二値化、モルフォロジー、エッジ検出



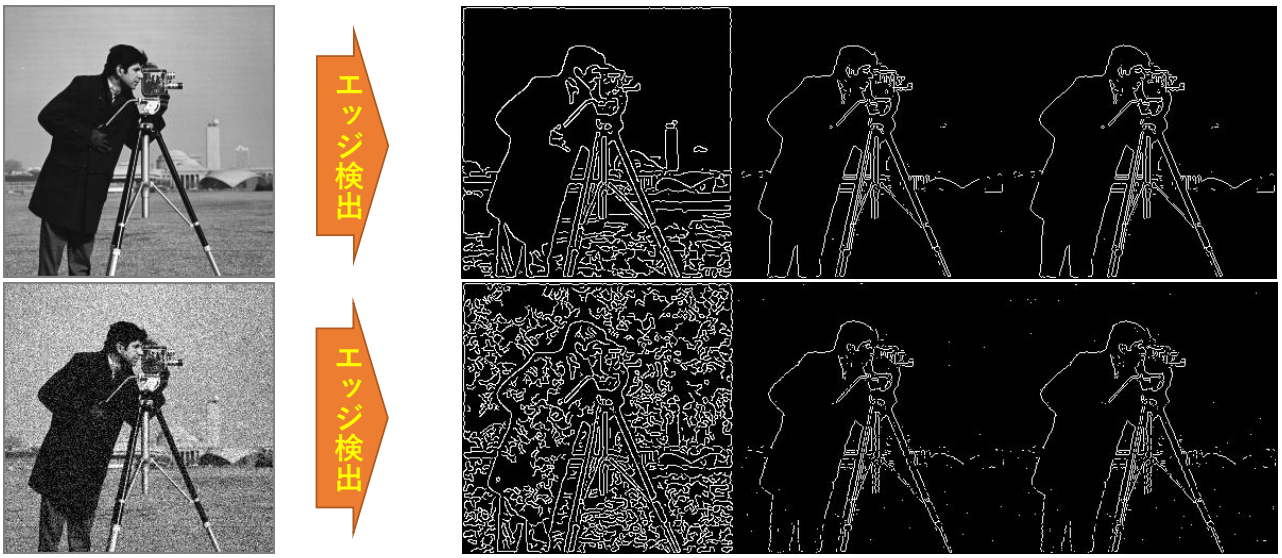
2.1 二値化、モルフォロジー、エッジ検出



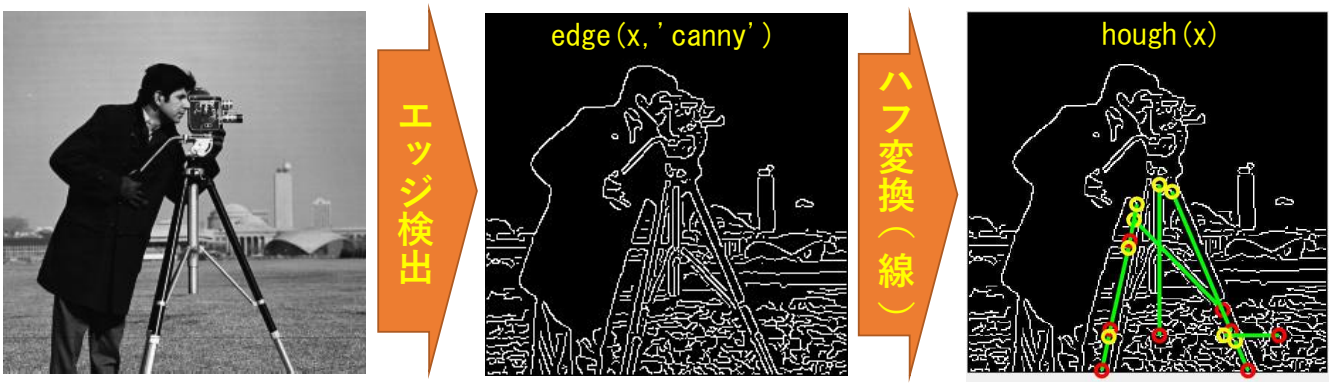
2.1 二値化、モルフォロジー、エッジ検出

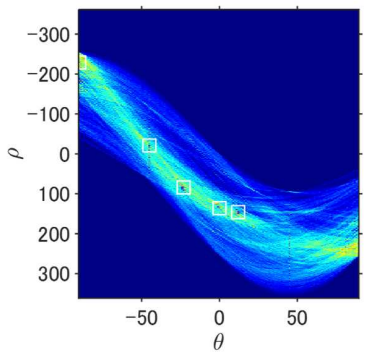
MATLAB の関数を使った場合

edge(x,'Canny')　edge(x,'Prewitt')　edge(x,'Sobel')



2.1 ハフ変換（線の検出）





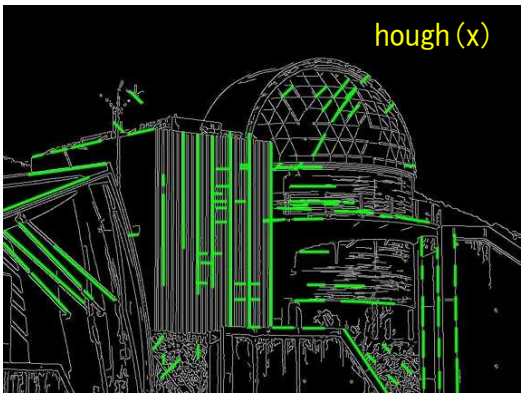
↑
緑：検出された線分
黄：線分の始点
赤：線分の終点

← パラメータの投票結果

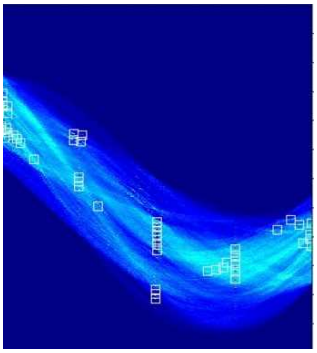
2.1 ハフ変換（線の検出）

エッジ検出

ハフ変換（線）



↑
緑：検出された線分



← パラメータの
投票結果

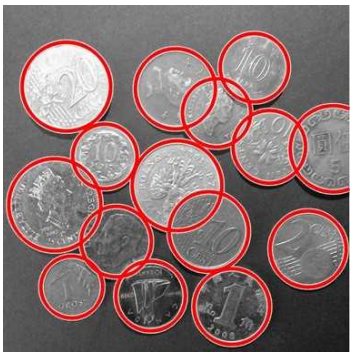
2.1 ハフ変換（円の検出）

↓赤：検出された円

二値化

imbinarize(x,'adaptive')

ハフ変換（円）



imfindcircles(x)

半径 と 中心座標

47	153	225
55	451	361
51	220	430
50	372	95
65	96	123
他、		10個

```
I=imread('貨幣重畳.jpg');  % 画像を読み込む
I=rgb2gray(I);             % カラーを白黒へ
imshow(I);                 % 画像を表示

[centers,radii,metric] = imfindcircles(I,[40 400],'Sensitivity',0.9);
viscircles(centers, radii);% 画像上に円を表示
[radii centers]            % 円の半径と中心座標
```

関連する MATLAB の関数　(1/3)

モルフォロジー

bwareaopen	バイナリ イメージからの小さなオブジェクトの削除
bwconncomp	バイナリ イメージ内の連結要素を検出
bwmorph3	バイナリ ボリュームのモルフォロジー演算
bwskel	すべてのオブジェクトを 2 次元のバイナリ イメージまたは
imbothat	ボトム ハット フィルター処理
imclose	イメージにモルフォロジー クロージングを行う
imdilate	イメージの膨張
imerode	イメージの収縮
imopen	イメージのモルフォロジー オープニング
imreconstruct	モルフォロジー再構成
imregionalmax	局所的な最大値
imregionalmin	局所的な最小値
imtophat	トップ ハット フィルター処理
offsetstrel	モルフォロジー オフセット構造化要素
strel	モルフォロジー構造化要素
watershed	Watershed 変換

関連する MATLAB の関数　(2/3)

モルフォロジー演算の実行

imerode	イメージの収縮
imdilate	イメージの膨張
imopen	イメージのモルフォロジー オープニング
imclose	イメージにモルフォロジー クロージングを行う
imtophat	トップ ハット フィルター処理
imbothat	ボトム ハット フィルター処理
imclearborder	イメージ境界と連結している明るい構造を非表示にする
imfill	イメージ領域と穴の塗りつぶし
bwhitmiss	バイナリ ヒットミス演算
bwmorph	バイナリ イメージのモルフォロジー演算
bwmorph3	バイナリ ボリュームのモルフォロジー演算
bwperim	バイナリ イメージのオブジェクトの周囲を検出
bwskel	すべてのオブジェクトを 2 次元のバイナリ イメージまたは
bwulterode	最終的な収縮

関連する MATLAB の関数　(3/3)

円の検出

imfindcircles	円のハフ変換を使用した円の検索
viscircles	円の作成

エッジと勾配の検出

edge	強度イメージ内のエッジの検出
edge3	3 次元強度ボリューム内のエッジの検出
imgradient	2 次元イメージの勾配の大きさと方向の検出
imgradientxy	2 次元イメージの方向勾配の検出
imgradient3	3 次元イメージの勾配の大きさと方向の検出
imgradientxyz	3 次元イメージの方向勾配の検出

ラインの検出

hough	ハフ変換
houghlines	ハフ変換に基づく線分の抽出
houghpeaks	ハフ変換のピークの特定
radon	ラドン変換
iradon	逆ラドン変換

2．基本的な画像処理

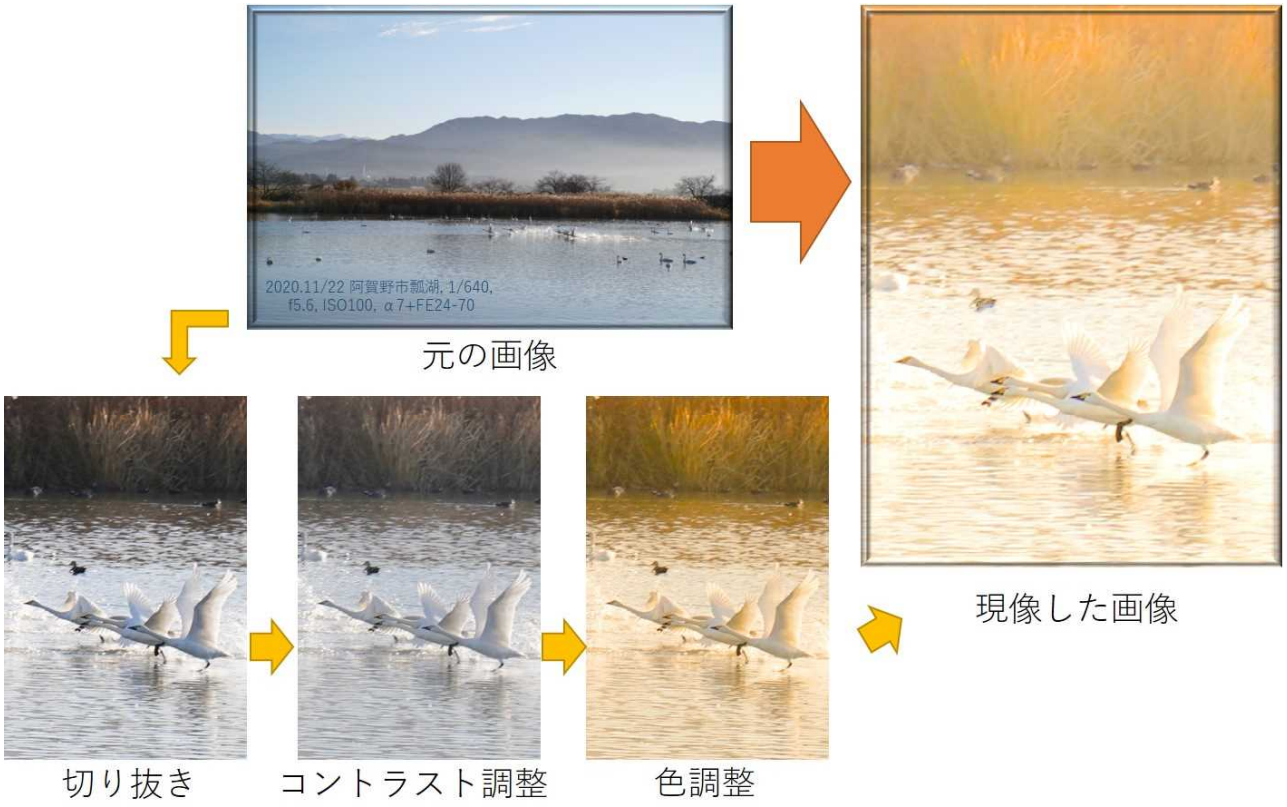
2.1 二値化、モルフォロジー、エッジ検出

2.2 輝度補正、コントラスト調整、トーンマッピング

2.3 拡大、縮小、回転、歪み補正

Photoshop
による例

2.2 Photoshop による現像（例）



2.2 Photoshop による現像（例）

コントラスト調整

色調整（色かぶり、彩度）

2.2 Photoshop による現像（例）

元の画像

コントラスト調整

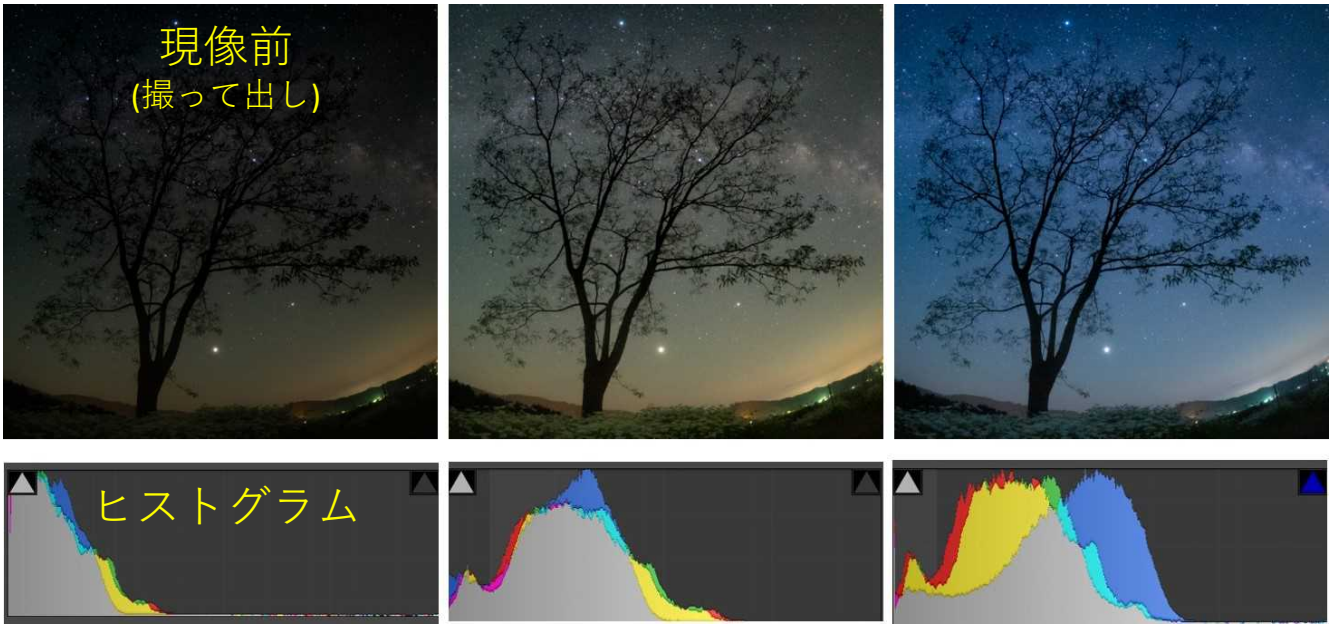
色調整

元の画像

コントラスト調整

色調整

現像の手順 1/2（lightroom）

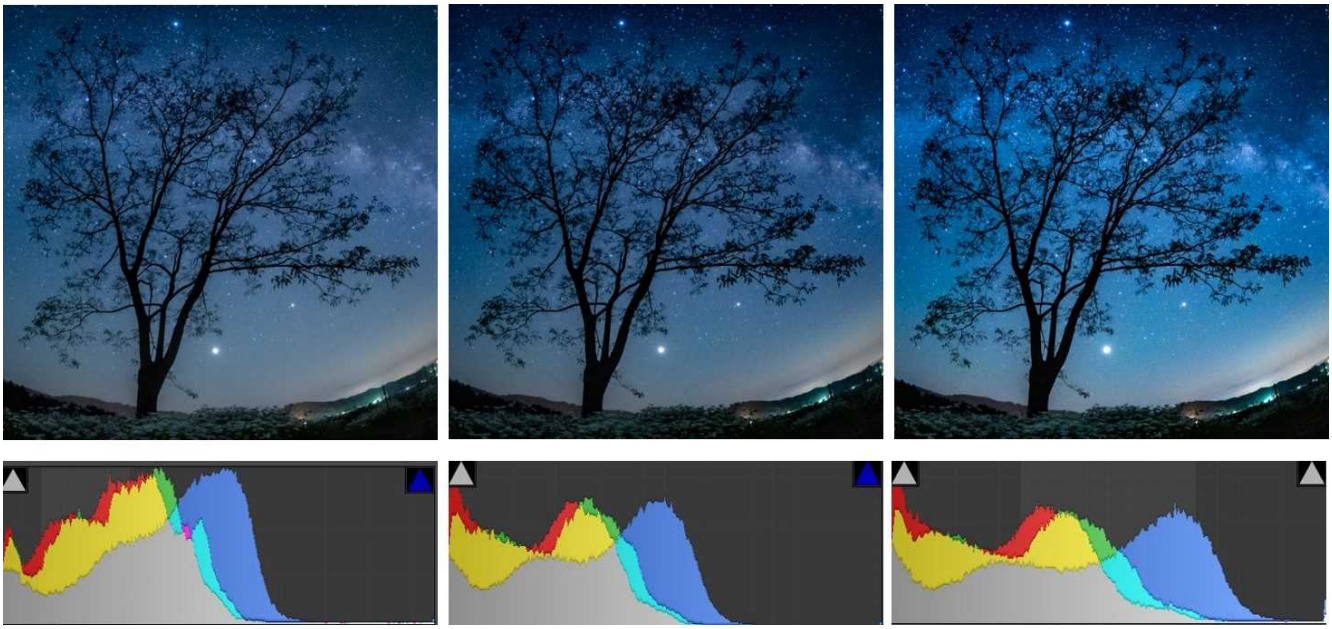


色温度　5100
色かぶり　+20

露光量
+1.36

色温度　3600
色かぶり　+30

現像の例 2/2（lightroom）



明瞭度
+60

カスミ除去
+41

白レベル
+45

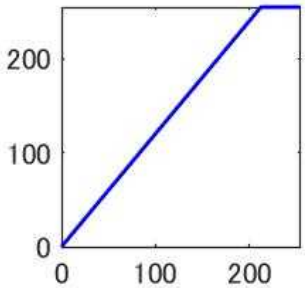
2． 基本的な画像処理

2.1 ニ値化、モルフォロジー、エッジ検出

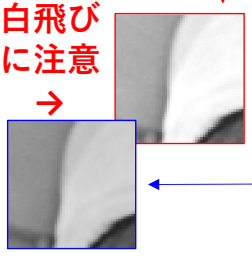
2.2 輝度補正、コントラスト調整、トーンマッピング

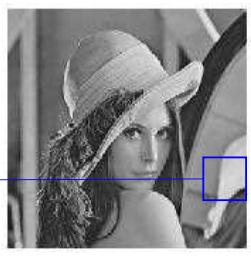
2.3 拡大、縮小、回転、歪み補正

明るくする（定数倍で）



白飛びに注意
→

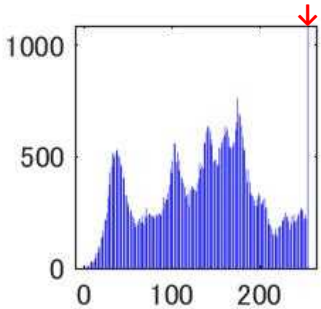




元の画像

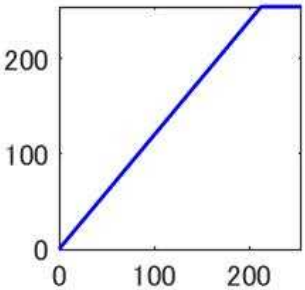
定数倍
$y = a\,x$
$a = 1.2$

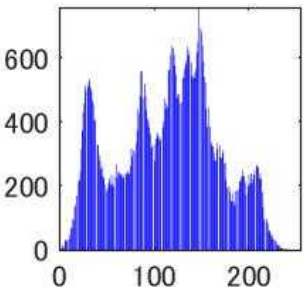
白飛び



ヒストグラム
（補正後）

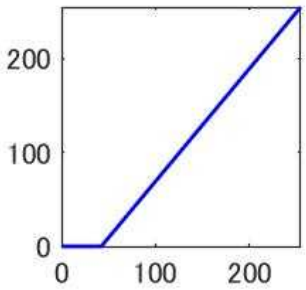
トーンマッピング
$\min(1.2x, 255)$



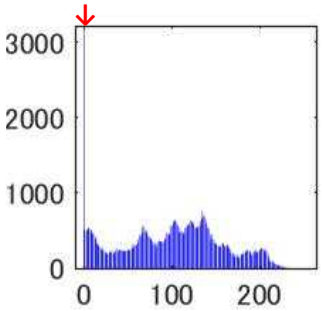


ヒストグラム
（補正前）

暗くする（定数倍で）

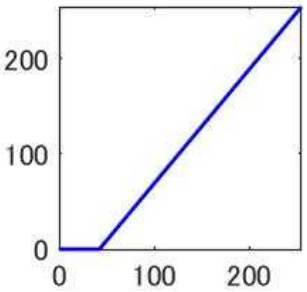


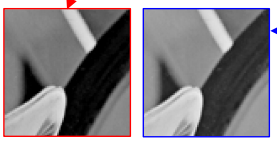
黒潰れ



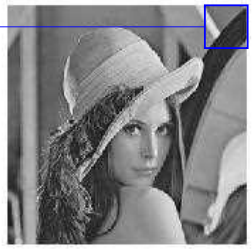
ヒストグラム
（補正後）

トーンマッピング
$\max(1.2x-51, 0)$

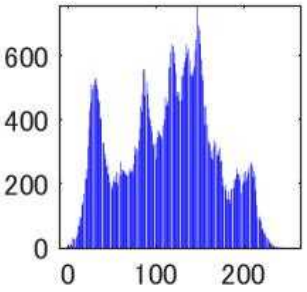




黒潰れ
に注意



元の画像



ヒストグラム
（補正前）

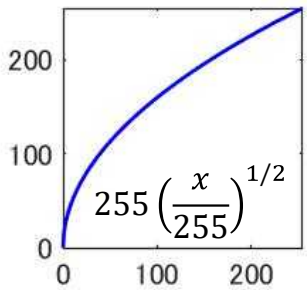
定数倍
$y = a\,x + b$

$a = 1.2, \quad b = -51$

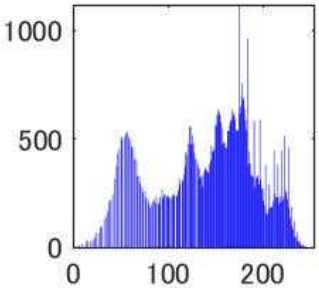
明るくする（ガンマ補正で）

明るくした
（補正後）

トーンマッピング

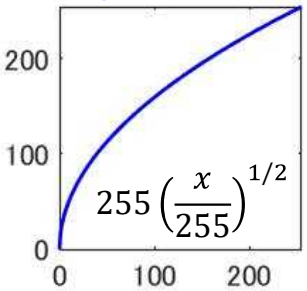


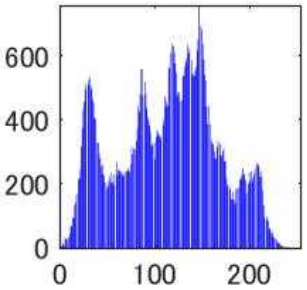
元の画像



ヒストグラム
（補正後）

トーンマッピング





ヒストグラム
（補正前）

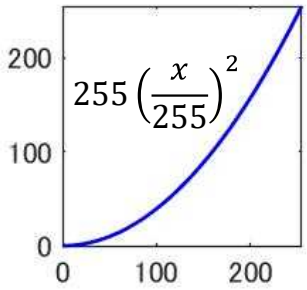
ガンマ補正
$q = p^{1/\gamma}$

$\gamma = 2$

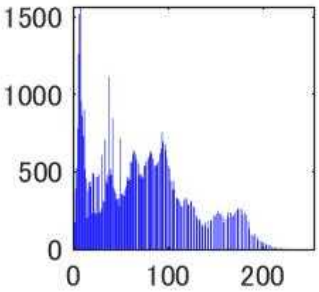
暗くする（ガンマ補正で）

暗くした
（補正後）

トーンマッピング



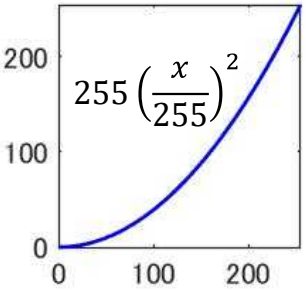
元の画像

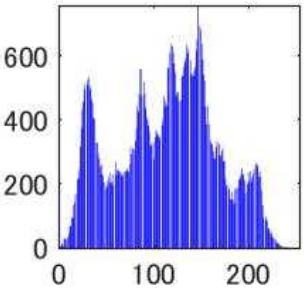


ヒストグラム
（補正後）

ガンマ補正
$q = p^{1/\gamma}$
$\gamma = 1/2$

トーンマッピング

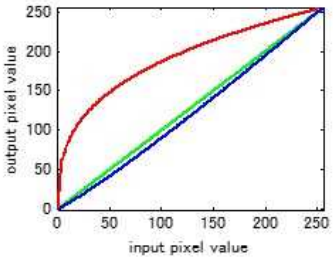




ヒストグラム
（補正前）

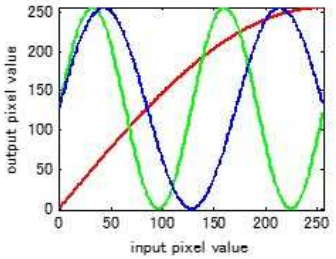
トーンマッピング（R,G,B 別々に）

赤くなった



元の画像

ｻｲｹﾃﾞﾘｯｸ?!



元の画像

2．基本的な画像処理

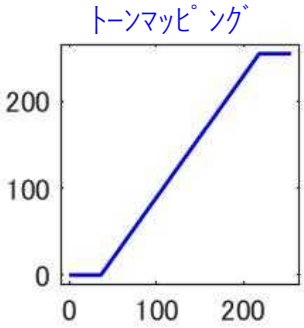
2.1 ニ値化、モルフォロジー、エッジ検出

2.2 輝度補正、コントラスト調整、トーンマッピング

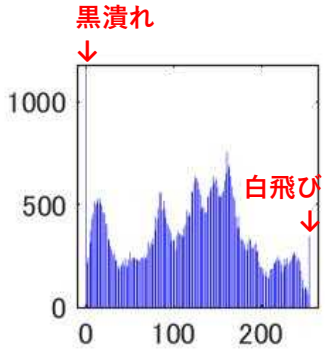
2.3 拡大、縮小、回転、歪み補正

ダイナミックレンジを拡大（定数倍で）

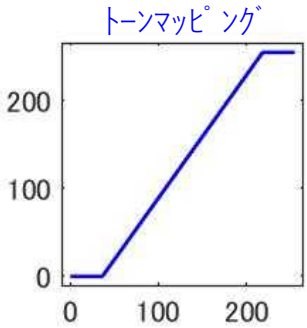
調整後

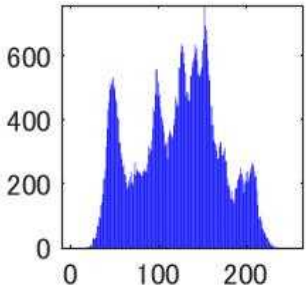


調整前



ヒストグラム
（補正後）





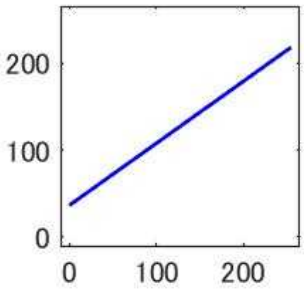
ヒストグラム
（補正前）

白飛び
と
黒潰れ
に注意

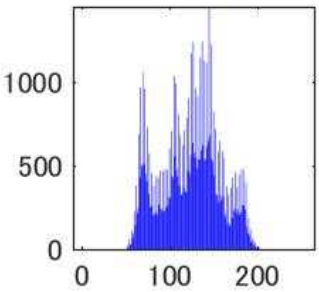
ダイナミックレンジを縮小（定数倍で）

調整後

トーンマッピング

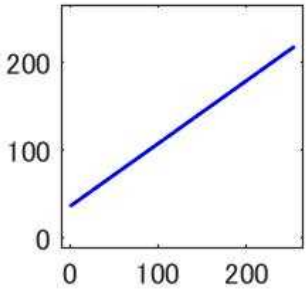


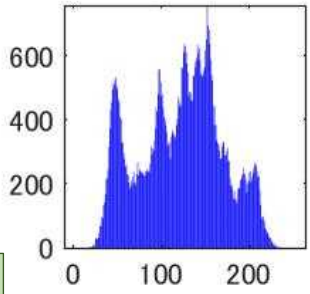
調整前



ヒストグラム
（補正後）

トーンマッピング





ヒストグラム
（補正前）

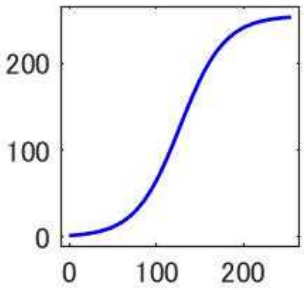
$a = 0.7, b = 1/2$

$$255\left(a\left(\frac{x}{255} - b\right) + b\right)$$

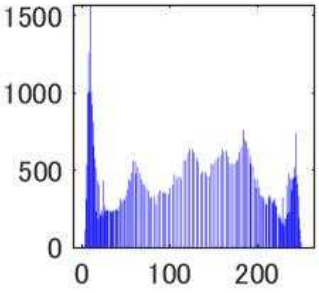
ダイナミックレンジを拡大（シグモイド関数で）

調整後

トーンマッピング

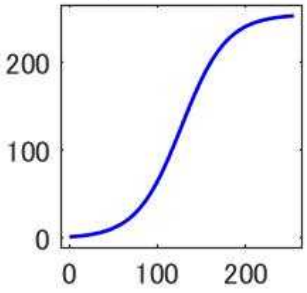


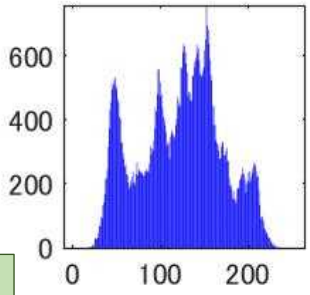
調整前



ヒストグラム
（補正後）

トーンマッピング





ヒストグラム
（補正前）

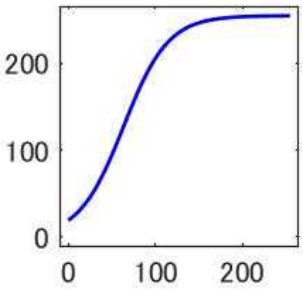
$a = 10, b = 1/2$

$$\frac{255}{1 + \exp\left(-a\left(\frac{x}{255} - b\right)\right)}$$

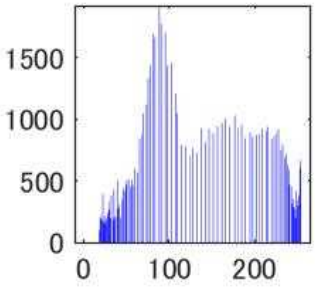
ﾀﾞｲﾅﾐｯｸﾚﾝｼﾞを拡大（シグモイド関数で）

調整後

トーンマッピング

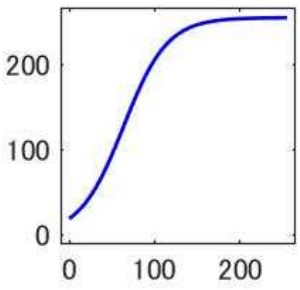


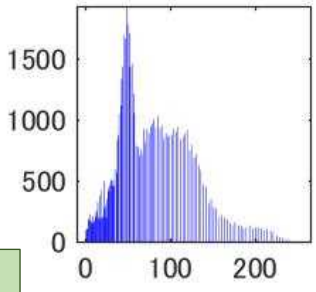
調整前



ヒストグラム
（補正後）

トーンマッピング





ヒストグラム
（補正前）

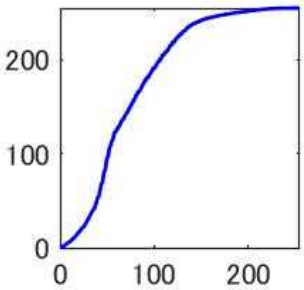
$a = 10, b = 1/4$

$$\frac{255}{1+\exp\left(-a\left(\frac{x}{255}-b\right)\right)}$$

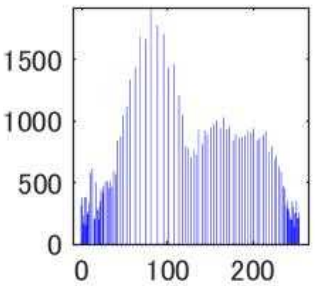
ｺﾝﾄﾗｽﾄ調整（ヒストグラム均等化）

調整後

トーンマッピング

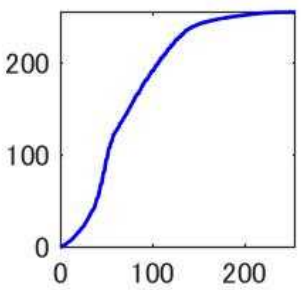


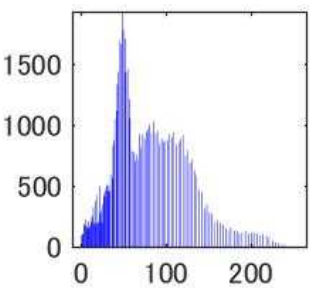
調整前



ヒストグラム
（補正後）

トーンマッピング



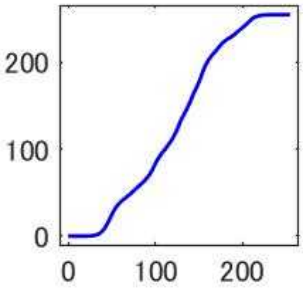


ヒストグラム
（補正前）

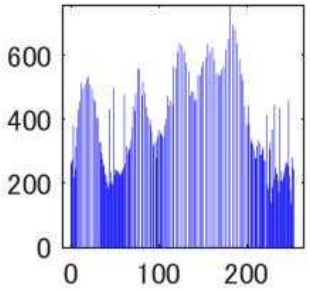
ｺﾝﾄﾗｽﾄ調整（ヒストグラム均等化）

調整後

トーンマッピング

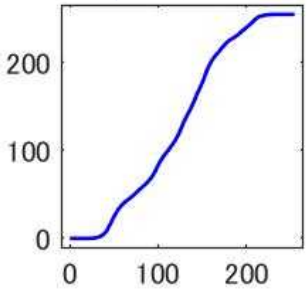


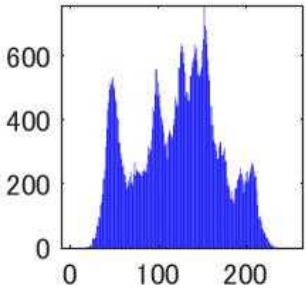
調整前



ヒストグラム
（補正後）

トーンマッピング



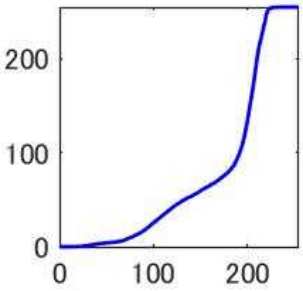


ヒストグラム
（補正前）

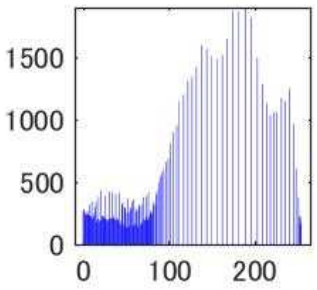
ｺﾝﾄﾗｽﾄ調整（ヒストグラム均等化）

調整後

トーンマッピング

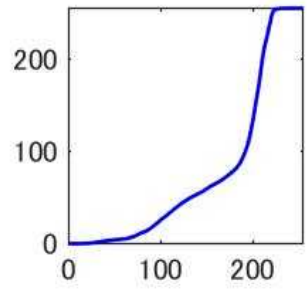


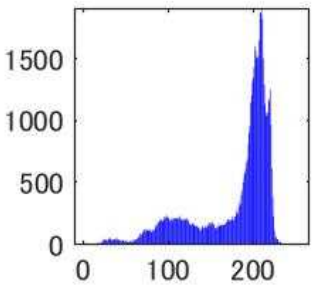
調整前



ヒストグラム
（補正後）

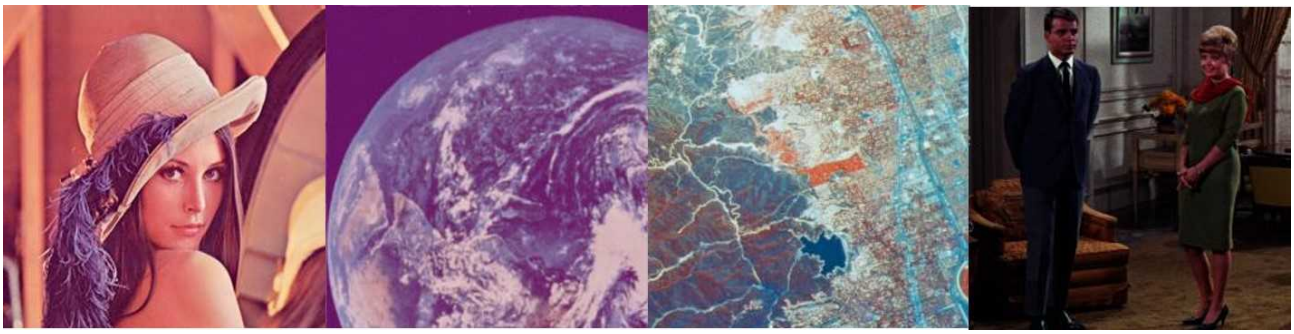
トーンマッピング





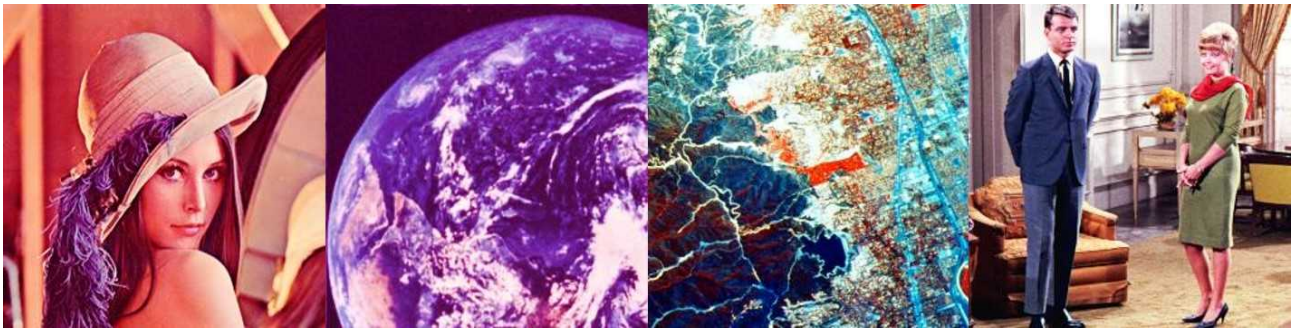
ヒストグラム
（補正前）

ヒストグラム均等化（MATLABによる）



↑　元の画像

↓　ヒストグラム均等化の適用後



MATLAB: histeq(X)

2．基本的な画像処理

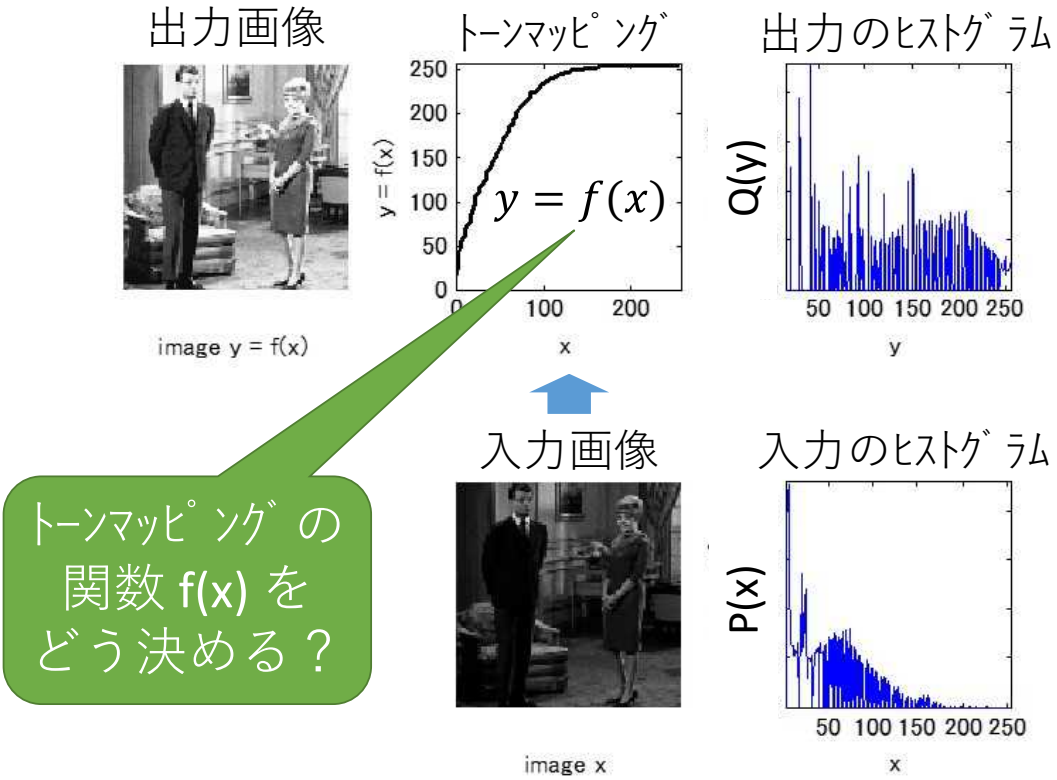
2.1 二値化、モルフォロジー、エッジ検出

2.2 輝度補正、コントラスト調整、トーンマッピング

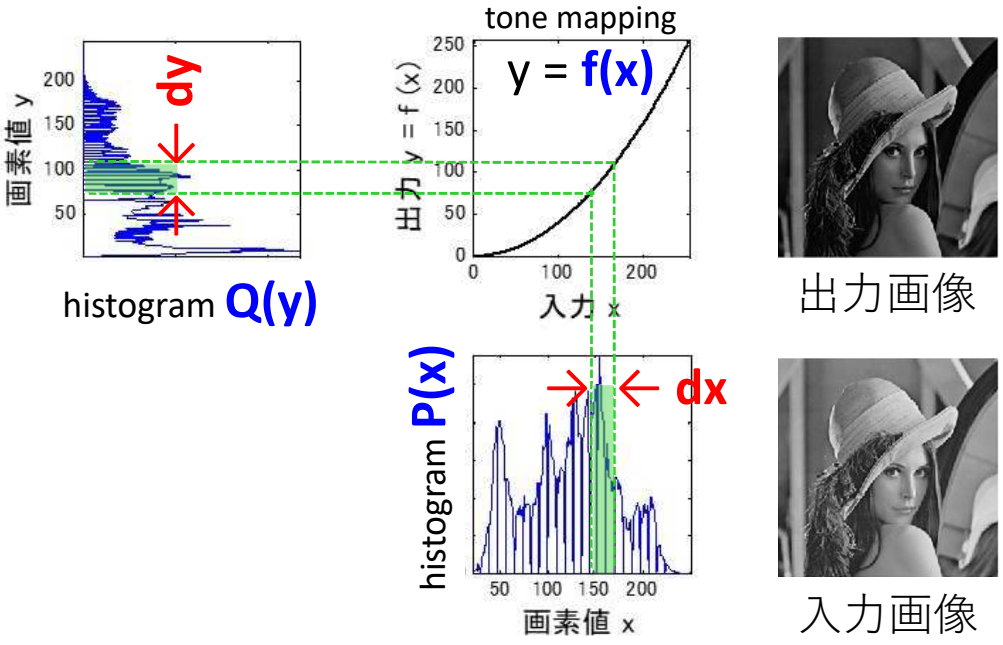
2.3 拡大、縮小、回転、歪み補正

ヒストグラム均等化の数理

ヒストグラム均等化の数理



問１　Q(y) を f(x) と P(x) で表したい



ヒント　Q(y) dy = P(x) dx　(図中の ■ の面積)

解

$$Q(y)dy = P(x)dx$$

より

$$Q(y) = P(x)\frac{dx}{dy} = \frac{P(x)}{\frac{dy}{dx}}$$

ここで

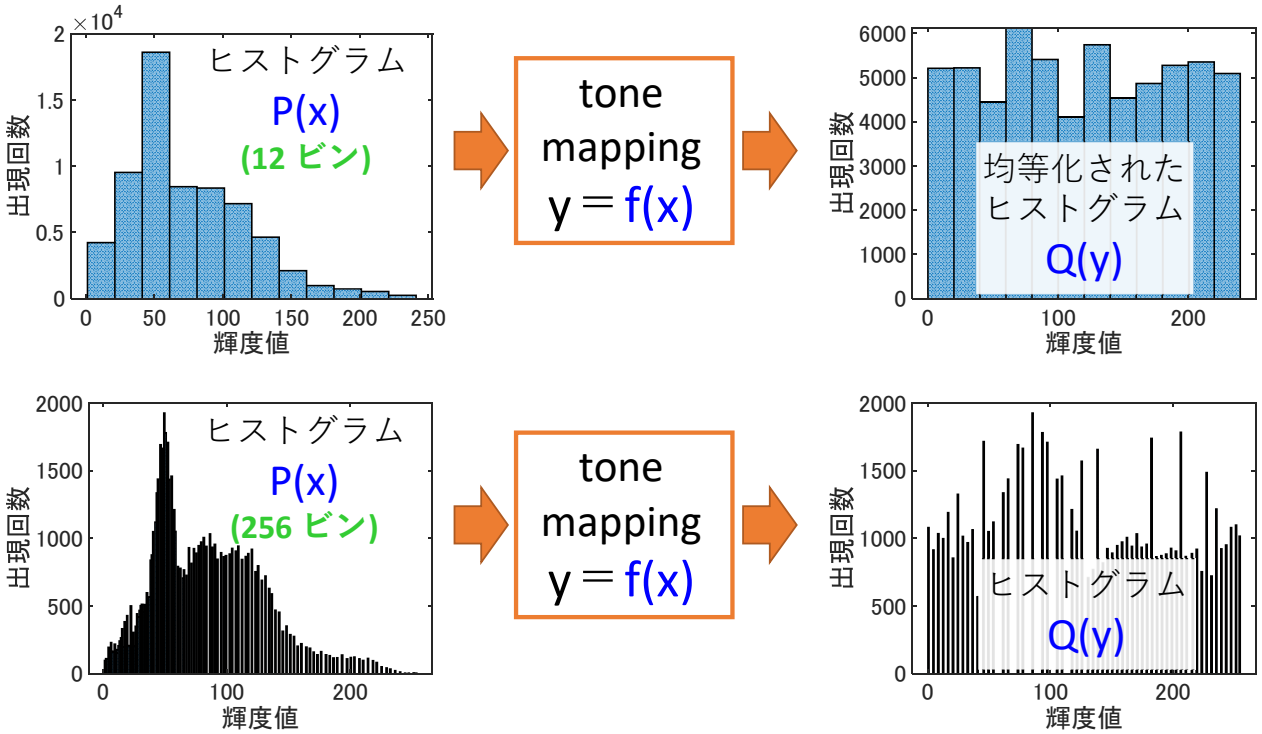
$$y = f(x), \quad \frac{dy}{dx} = f'(x)$$

なので

$$Q(y) = \frac{P(x)}{f'(x)}$$

トーンマッピング後の
ヒストグラム Q(y) を
f(x) と P(x) で表せた

問２　Q(y) を均等にする f(x) は？



解　f(x) を P(x) から決める

$$Q(y) = \frac{P(x)}{f'(x)} = Const.$$

← $Q(y)dy = P(x)dx$

$dy/dx = f'(x)$ より

なので、

$$f'(x) = \frac{P(x)}{Const.}$$

従って、

$$f(x) = \int_{\min}^{x} \frac{P(u)}{Const.} du$$

← 結果は**累積度数**

ﾄｰﾝﾏｯﾋﾟﾝｸﾞ後のﾋｽﾄｸﾞﾗﾑ Q(y) が均等となる様に
トーンマッピングの関数 f(x) が決定された

2．基本的な画像処理

2.1 二値化、モルフォロジー、エッジ検出

2.2 輝度補正、コントラスト調整、トーンマッピング

2.3 拡大、縮小、回転、歪み補正

ヒストグラム
均等化の欠点

トーンマッピングによるノイズの増減

増幅／低減の度合いは

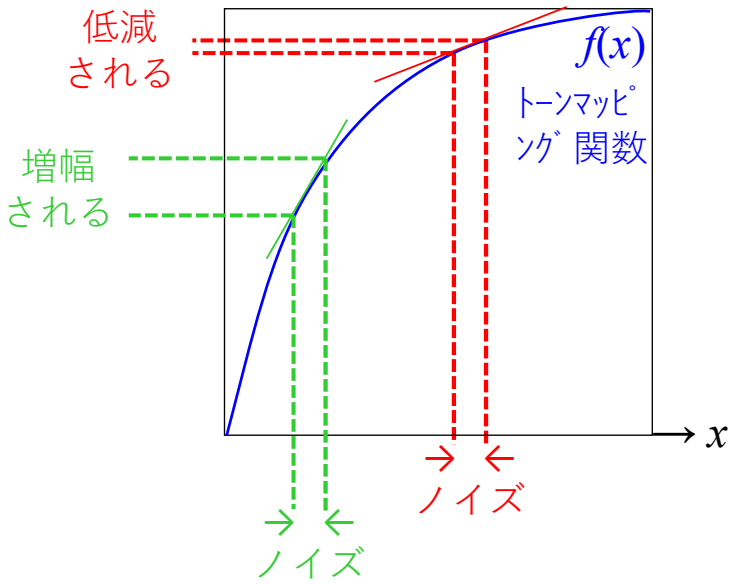
関数 $f(x)$ の微係数

$\frac{df(x)}{dx}$ で決まる

増幅もせず

低減もしないのは

$f(x) = x$

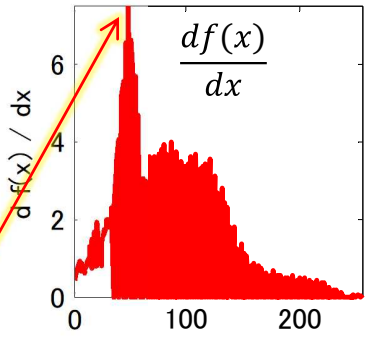


トーンマッピング前の
ノイズ（大きさは同じ）

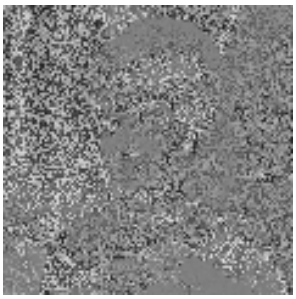
出力

image y = f(x + e)

トーンマッピング関数
の微係数



出力の誤差



noise in y

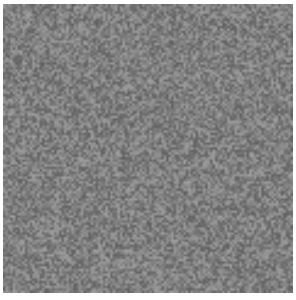
微係数が最も大きい

ノイズが最も大きく
増幅される

入力

image x + e

入力の誤差

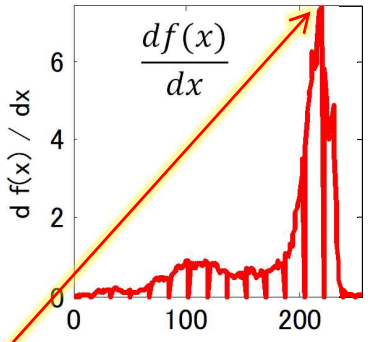


noise in x

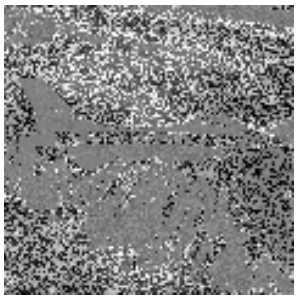
出力

image y = f(x + e)

トーンマッピング関数の微係数



出力の誤差

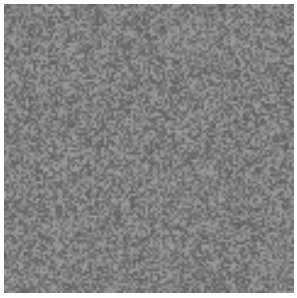


noise in y

入力

image x + e

入力の誤差



noise in x

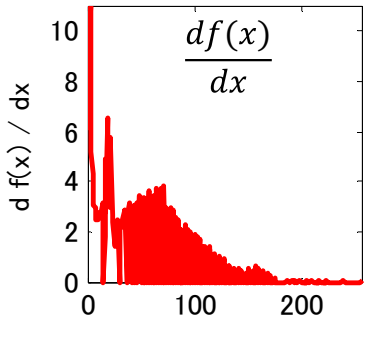
微係数が最も大きい

ノイズが最も大きく
増幅される

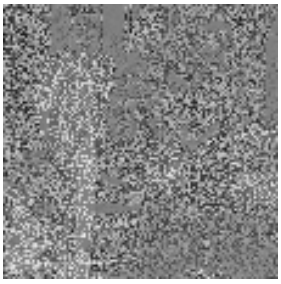
出力

image y = f(x + e)

トーンマッピング関数の微係数



出力の誤差

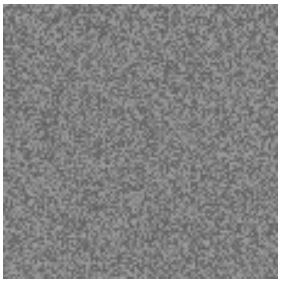


noise in y

入力

image x + e

入力の誤差



noise in x

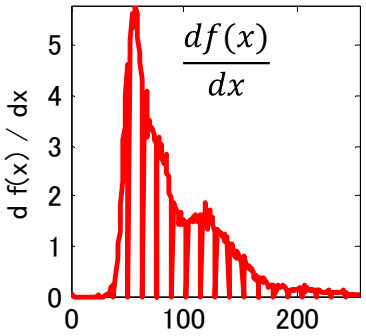
ヒストグラム均等化
の場合

トーンマッピング関数の
微係数は
入力画像のヒストグラム
に等しい

出力

image y = f(x + e)

トーンマッピング関数の微係数



出力の誤差

noise in y

入力

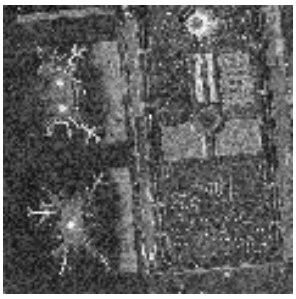
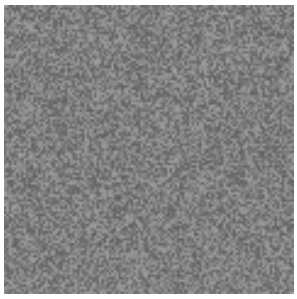


image x + e

入力の誤差



noise in x

コントラストの調整

imadjust	イメージの強度値またはカラーマップの調整
imadjustn	N 次元ボリューム イメージの強度値の調整
imcontrast	コントラスト調整ツール
imsharpen	アンシャープなマスキングを使用したイメージのシャープ化
imflatfield	2 次元イメージのフラットフィールド補正
imlocalbrighten	低光量イメージを明るくする
imreducehaze	大気によるかすみの低減
locallapfilt	イメージの高速な局所ラプラシアン フィルター処理
localcontrast	イメージのエッジ保存型の局所的コントラスト操作
localtonemap	局所的なコントラストを強調しながら HDR イメージを表示用にレンダリング
histeq	ヒストグラム均等化を使用したコントラストの強調
adapthisteq	コントラストに制限を付けた適応ヒストグラム均等化を実行 (CLAHE)
imhistmatch	参照イメージのヒストグラムに一致させるために 2 次元イメージのヒストグラムを調整
imhistmatchn	参照イメージのヒストグラムと一致させるために N 次元イメージのヒストグラムを調整
decorrstretch	無相関ストレッチをマルチチャネル イメージに適用
stretchlim	コントラスト ストレッチ イメージの範囲の確認
intlut	ルックアップ テーブルを使った整数値の変換
imnoise	イメージにノイズを付加

2．基本的な画像処理

2.1 ニ値化、モルフォロジー、エッジ検出

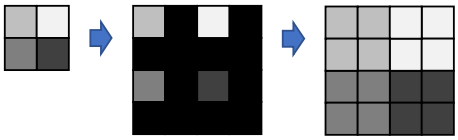
2.2 輝度補正、コントラスト調整、トーンマッピング

2.3 拡大、縮小、回転、歪み補正

2.3 拡大、縮小、回転、歪み補正

```
imresize(x,s,'box');
```

1つの値を3カ所にコピー

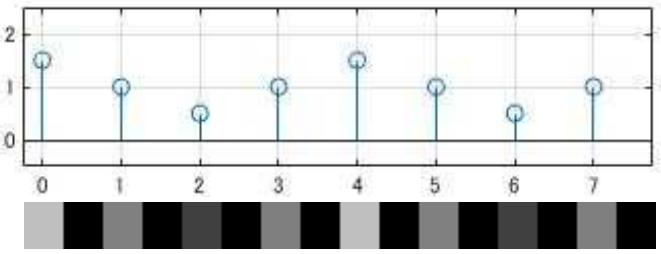


2.3 拡大、縮小、回転、歪み補正

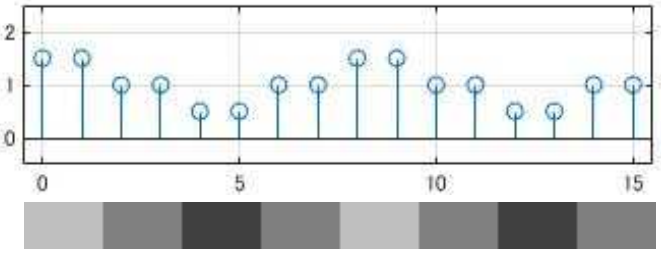
```
imresize(x,s,'cubic');

% 'box'        箱型のカーネル
% 'triangle'   三角カーネル ('bilinear')
% 'cubic'      3 次カーネル ('bicubic')
% 'lanczos2'   Lanczos-2 カーネル
% 'lanczos3'   Lanczos-3 カーネル
```

→ 超解像
super resolution

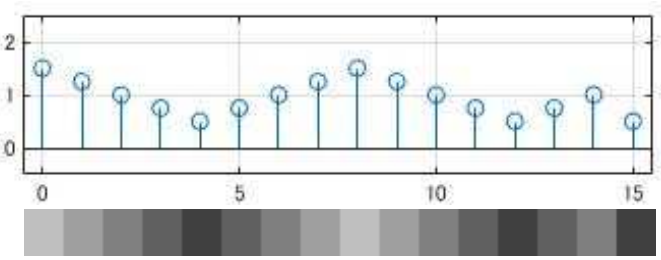
補間による拡大



元の信号



左隣と同じ値で
補間して
解像度を２倍にした

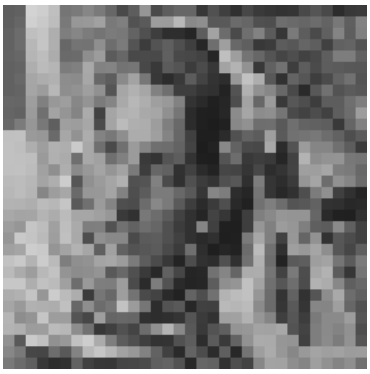
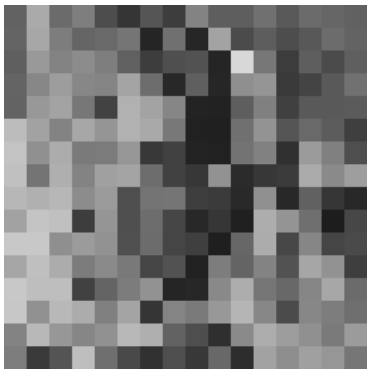


両側の値の平均値で
補間して
解像度を２倍にした

2.3 拡大、縮小、回転、歪み補正



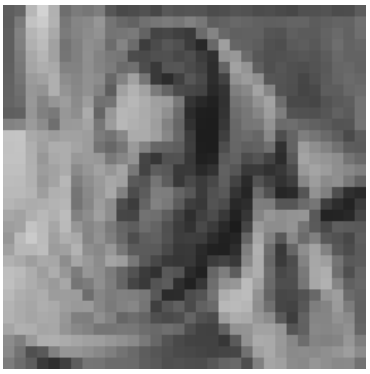
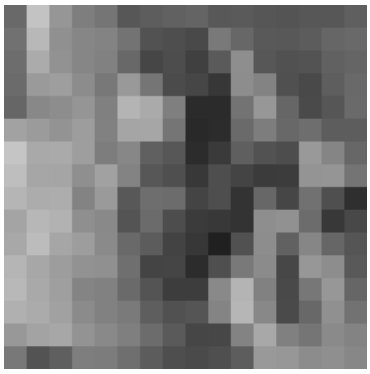
エイリアスを
除去しないと…

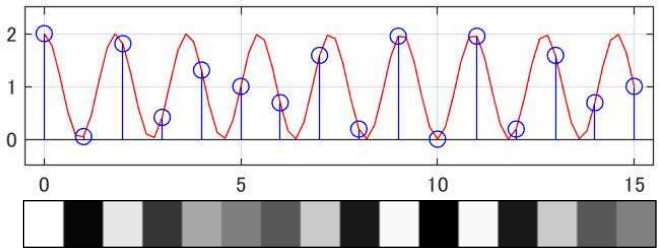

2.3 拡大、縮小、回転、歪み補正



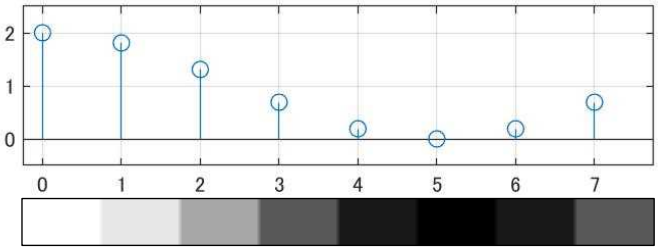
エイリアスを
除去した

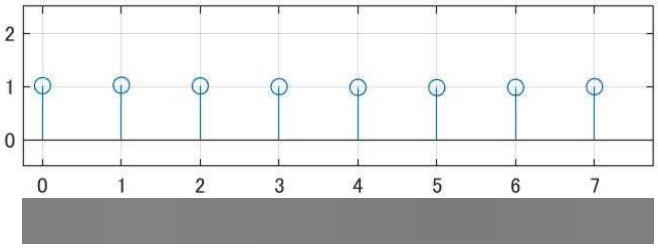

エリアス除去とは？



元の信号



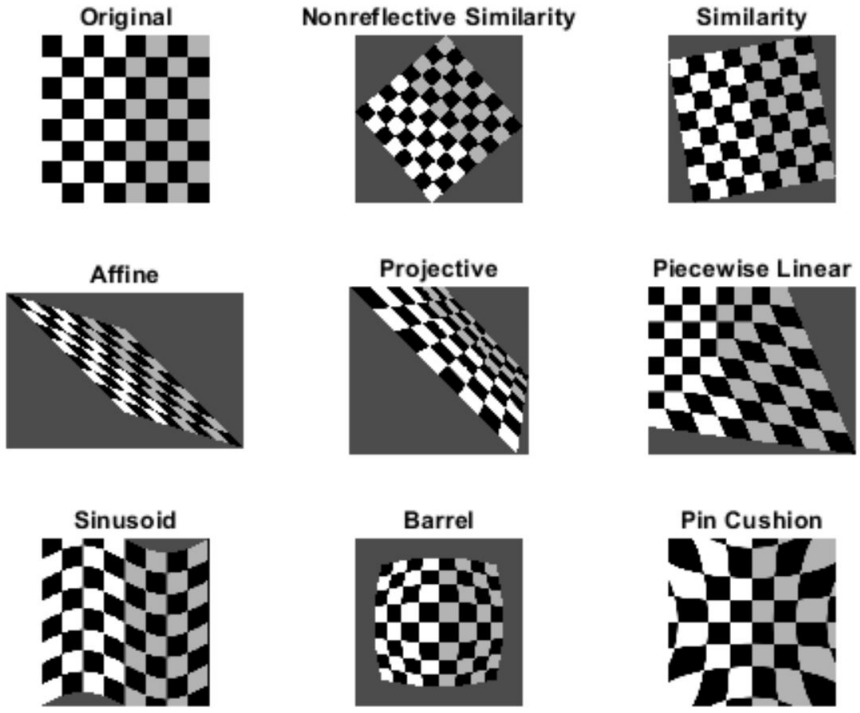
エリアスを除去せずに
解像度を1/2倍にした



フィルタ [1 2 1]/4 をかけて
エリアスを除去してから
解像度を1/2倍にした

2.3 拡大、縮小、回転、歪み補正

2.3 拡大、縮小、回転、歪み補正



出典：　MATLAB　変換イメージのギャラリーの作成

関連する MATLAB 関数

一般的な幾何学的変換

imcrop	イメージのトリミング
imcrop3	3 次元イメージのトリミング
imresize	イメージのサイズ変更
imresize3	3 次元ボリューム強度イメージのサイズ変更
imrotate	イメージの回転
imrotate3	3 次元ボリューム グレースケール イメージの回転
imtranslate	イメージの平行移動
impyramid	イメージ ピラミッドの縮小と拡張

汎用的な幾何学的変換

imwarp	イメージへの幾何学的変換の適用
affineOutputView	ワーピング イメージの出力表示の作成
fitgeotrans	幾何学的変換のコントロール ポイントのペアへの近似
findbounds	空間変換の出力境界の検出
fliptform	空間変換構造体の入力と出力の役割の反転
makeresampler	リサンプリング構造体の作成
maketform	空間変換構造体 (TFORM) を作成
tformarray	N 次元配列への空間変換の適用
tformfwd	フォワード空間変換の適用
tforminv	逆空間変換の適用

その他～　比較明合成

LightroomとPhotoshopの比較明合成で
印象的なホタル・星景写真作成

https://life-with-photo.com/lightsynthesis

比較明合成で花火を仕上げよう

https://silkypix.isl.co.jp/how-to/method/lighten-composite_fire-works/

星の軌跡をきれいに撮影できる
比較明合成のやり方とは？

https://capa.getnavi.jp/special/259316/

ライブコンポジット
（比較明合成）による撮影
OM-D E-M1, ISO 16000, F2.8, 10sec
2021.6.13 長岡市