

画像処理入門 with Processing

慶應義塾大学SFC

メディア技術基礎(ネットワーク・画像処理)

笥 康明

第3章：画像処理プログラミングの基本

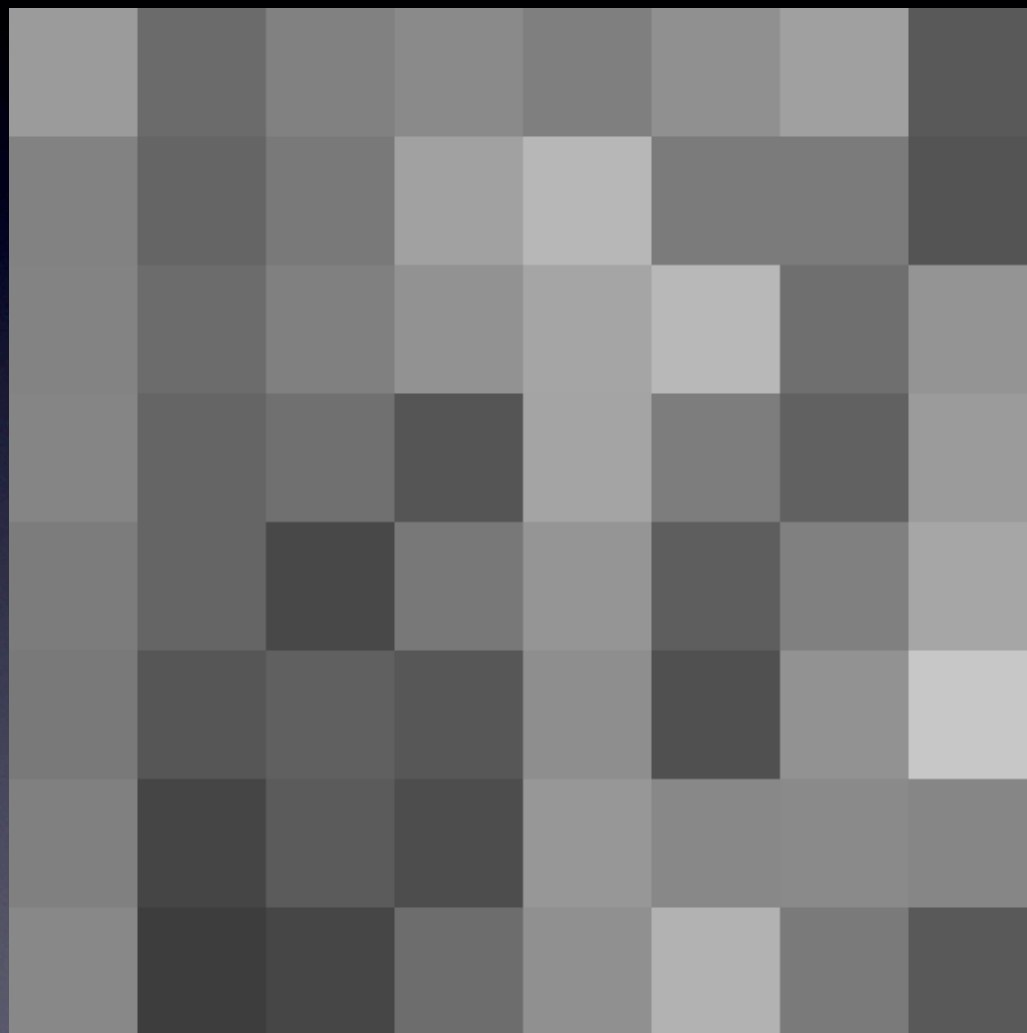
デジタル画像の生成

- まずは白黒画像を扱う
- アナログ白黒画像は連続的濃淡情報で表される

デジタル画像の構成



デジタル画像の構成



画像の座標系

原点

0

y軸

512



デジタル画像の構成

| | | | | | | | |
|-----|----|----|-----|-----|-----|-----|-----|
| 160 | 50 | 63 | 120 | 54 | 72 | 171 | 25 |
| 123 | 45 | 54 | 180 | 210 | 61 | 63 | 25 |
| 142 | 53 | 73 | 90 | 190 | 220 | 62 | 167 |
| 139 | 38 | 63 | 20 | 190 | 78 | 40 | 96 |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |

標本化と量子化

- 標本化・・・連続濃淡画像を離散点へ分割するプロセス (sampling)
- 量子化・・・分割された点の濃淡のデジタル値を決定するプロセス (quantization)

画像解像度

- 元の画像の構成要素を解像・分解できるデジタル画像の能力
 - 空間解像度
 - 輝度分解能（階調数）

forループのおさらい

```
int tmp=0;  
for(int i=0; i<5;i++){  
    tmp +=i;  
}
```

//tmpの値の表示

```
println(tmp);
```


とにかく画像を表示

```
size(512,512);
```

```
PImage img = loadImage("lenna_mono.jpg");
```

```
image(img, 0, 0);
```

とにかく画像を表示

```
// 画面サイズを指定  
size(512,512);  
// 画像データの読み込み  
PImage img = loadImage("lenna_mono.jpg");  
// 画像の描画  
image(img, 0, 0);
```


size(x, y)

- **描画のためのキャンバスサイズの指定**

x: 横軸のピクセル数

y: 縦軸のピクセル数

PImage

- 画像情報格納クラス

[http://processing.org/reference/
PImage.html](http://processing.org/reference/PImage.html)

- フィールド

width、height、pixels[]

- メソッド

get() / set() / copy() / mask() / blend() /
filter() / save()

クラスとインスタンス

- クラスにはフィールドとメソッドが一緒に定義される
- そのインスタンスによっては中のフィールドが異なる
- メソッドが呼ばれると、フィールドの値に応じて動く

loadImage("...")

- 画像を読み込む関数
(processingで読み込み可能なフォーマットはjpg, gif, png, tga)
- パス（画像ファイルの置き場所）に注意
通常はDataディレクトリに格納

**image(PImage img,
int offset_x, int offset_y)**

- **image(画像クラス, x方向のオフセット,
y方向のオフセット)**
- **image(画像クラス, x方向のオフセット,
y方向のオフセット, x方向の描画サイ
ズ, y方向の描画サイズ)**

キャンバスサイズを変える

```
size(1024, 1024);
```

```
PImage img = loadImage("lenna_mono.jpg");
```

```
image(img, 0, 0);
```


画像を表示する位置

```
size(1024,1024);
```

```
PImage img = loadImage("lenna_mono.jpg");
```

```
image(img,30, 20);
```

画像の大きさを変える

```
size(1024,1024);
```

```
PImage img = loadImage("lenna_mono.jpg");
```

```
image(img, 0, 0, 50, 50);
```


画像サイズを取得する

```
size(512,512);  
PImage img = loadImage("lenna_mono.jpg");  
image(img, 0, 0);  
// 画像サイズの表示  
println("width= " + img.width + " height= " +  
img.height);
```

カラー画像

- カラー画像に対しても、標本化、量子化、空間解像度、輝度分解能の同じ概念が適用可能
- 単一輝度の代わり3つの色成分を使って量子化される

加法混色性

- あらゆる色は赤(R)、緑(G)、青(B)の主色の光を混合することにより生成できる
- 放出光に基づく
- RGBカラー空間

表色系

- RGB
- HSB (HSL, HSV, HSI)
 - H: Hue (色相)、S: Saturation (彩
度)、B: Brightness (輝度)

RGB ⇔ HSB変換

- RSB 表色系とHSB表色系は相互に変換可能

カラー画像を読み込

// 画像データの読み込み

```
PImage img = loadImage("color.jpg");
```

// 画面サイズを指定

```
size(img.width, img.height);
```

// 画像の描画

```
image(img, 0, 0);
```


色合いを変える

// 画像データの読み込み

```
PImage img = loadImage("color.jpg");
```

// 画面サイズを指定

```
size(img.width, img.height);
```

```
tint(255, 0, 0);
```

// 画像の描画

```
image(img, 0, 0);
```

tint

- **tint(gray)**
- **tint(gray, alpha)**
- **tint(value1, value2, value3)**
- **tint(value1, value2, value3, alpha)**
- **tint(color)**
- **tint(color, alpha)**
- **tint(hex)**
- **tint(hex, alpha)**

ピクセル単位で画像を扱う

- カラー・モノクロ変換
- ポジ・ネガ変換
- 2値化

ポジ・ネガ変換



カラー・モノクロ変換



2値化



色抽出

- 赤い部分だけを2値化で抽出する

輝度情報の置き換え

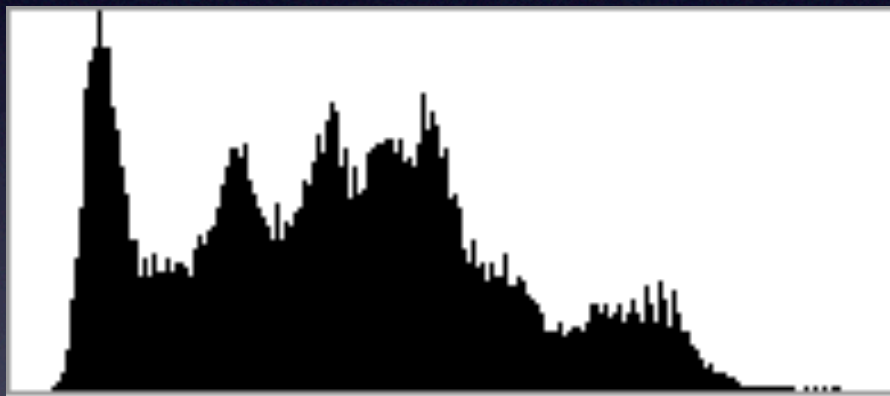
- 画像の反転



ヒストグラムを作ろう

- ヒストグラムとは・・・
 - 度数分布図、柱状グラフ、Histogram
 - 輝度やカラー情報に関するピクセル数の分布

ヒストグラムの例



ヒストグラムの例(2)



ヒストグラムの作成

- 輝度ヒストグラムとは・・・

画像内に含まれる輝度分布を棒グラフ
で表現したもの（横軸：輝度値、縦
軸：ピクセル数（の割合））

輝度の定義

- HSBのB (Brightness)
- $B = \text{Max}(R, G, B)$

拡大・縮小

- color.jpgの解像度を縦横2倍にして表示する

単一画像・局所処理

- 平均化（ノイズ除去）
- ノイズ強調

内挿処理の必要性

- 再近隣内挿法
 - 内挿したい点に最も近い位置の階調値を求める階調値とする
- 共一次内挿法
 - 内挿したい点の周囲4点の階調値の重み付き平均を用いる