

# デジタルメディア処理2

担当: 井尻 敬

## デジタルメディア処理2、2017（前期）

4/13 デジタル画像とは : イントロダクション  
4/20 フィルタ処理1 : 画素ごとの濃淡変換、線形フィルタ、非線形フィルタ  
4/27 フィルタ処理2 : フーリエ変換、ローパスフィルタ、ハイパスフィルタ  
5/11 画像の幾何変換1 : アフィン変換  
5/18 画像の幾何変換2 : 画像の補間、イメージモザイク  
5/25 画像領域分割 : 領域拡張法、動的輪郭モデル、グラフカット法、  
6/01 前半のまとめ (約30分)と中間試験 (約70分)  
6/08 特徴検出1 : テンプレートマッチング、コーナーエッジ検出  
6/15 特徴検出2 : DoG、SIFT特徴量、Hough変換  
6/22 画像認識1 : パターン認識概論、サポートベクターマシン  
6/29 画像認識2 : ニューラルネットワーク、深層学習  
7/06 画像処理演習 : ImageJを用いた画像処理入門  
7/13 画像処理演習 : Pythonを用いた画像処理プログラミング入門  
7/20 後半のまとめ (約30分)と期末試験 (約70分)

## 目的

- 本講義で解説した画像処理手法を体験する

## 準備 : ImageJのインストール

- Fiji-win64.zipをダウンロード
- Zipを展開しImageJ-win64.exeをダブルクリック

## Image-J

- NIH(アメリカ国立衛生研究所)が開発した画像解析ソフトウェア
  - Java
  - Windows/Mac/Linux
  - Open source
  - <http://rsbweb.nih.gov/ij/>
- 医用・生物画像の解析に優れ多くの研究者が利用
  - 美顔フィルタなどのエンタメ目的というよりは、学術研究目的のツール
- 拡張性が高くプラグイン開発可能

# FIJI ( Fiji Is Just ImageJ)

- Web-page <http://fiji.sc/Fiji>
- Image-Jに基づいた画像処理ソフト (Image-Jの実装の1つ)
- 自然科学者が手軽に利用できるように…
  - インストールが容易
  - 自然科学研究用の画像処理に適したプラグインが充実
  - 各処理に関するドキュメントが充実

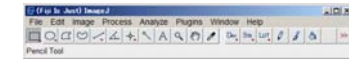
- 今回はこれを利用します

## Fijiを起動する

1. <http://fiji.sc/Fiji> にアクセス
2. 『Download Fiji now』をクリック
3. OSにあったzipをダウンロード
4. zipを展開し『imageJ-win\*.exe』をダブルクリック
5. 起動を確認する



3. OSにあったzipをダウンロード

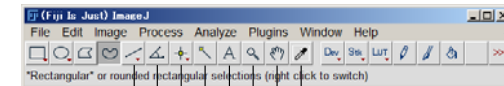


※必要なファイルはFijiappフォルダ内にあるので、アンインストールするときはFijiappフォルダを捨てればOK

## 画像データ

- 講義web pageに画像データを置いたので適宜利用してください  
[takashiijiri.com/classes/index.html](http://takashiijiri.com/classes/index.html)

## Image-J の 基本画面



- Area selection tools
- Color Picker
- Scrolling tool
- Zooming tool
- Text tool
- Wand
- Point tool
- Angle tool
- Line selection tool

- メニューバー
- ツールバー
- ステータスバー

<http://rsb.info.nih.gov/ij/docs/tools.html>



画像1



画像2

# 画像の読み込み と Format

## 1. 画像の読み込み

『File > Open』をクリックし画像を選択  
画像をImageJ上へドロップしてもOK

## 2. 画像のFormatを確認

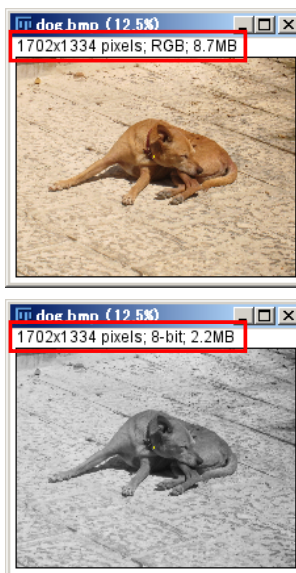
読み込んだ画像の上部にFormatが表示される

## 3. Format変換

『Image > type > \*』より変換先を選択  
(Morphologyはグレースケールのみなど、  
処理によって対応していないFormat有り)



**実習:** 適当な画像を用意し、ImageJで読み込み、formatを変換してみてください。



# LUT (Look Up table)

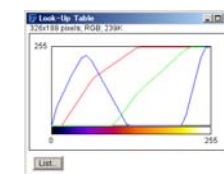
LUT: グレースケール画像に疑似カラーをつける機能

## 0. グレースケール画像の読み込み

1. 『menu > image > lookup tables > \*(疑似カラーセット名)』
  2. 『menu > image > color > show LUT』でLUTの中身を表示
- ※ LUTは疑似カラーで表示されるだけで、画像自体がカラーになるわけでない  
※ 『menu > image > type > 8 bit (※元の画像タイプ)』でLUTの効果が消える

## 実習:

1. 適当な画像をImageJで読み込み、
2. formatをグレースケールに変換し、
3. 色々な疑似カラーを適用してみてください

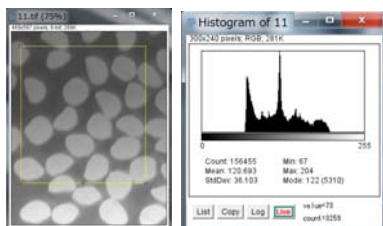


# 簡単な解析 - ヒストグラム

## 実習

以下の手順で画像のヒストグラムを可視化してください。

1. 選択ツールで画像の一部を選択
2. 『menu > analysis > Histogram』もしくは『h』キー
3. Histogram dialogの『live』をonにする  
→ 選択領域を変更しながらProfileを確認できる

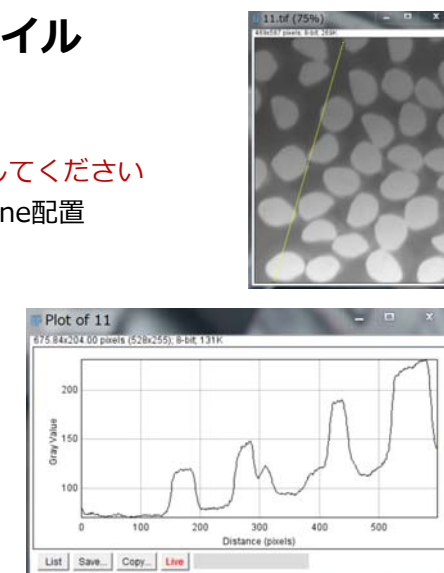


# 簡単な解析 - 輝度値プロファイル

## 実習

以下の手順で画像のヒストグラムを可視化してください

1. Line tool を選択し読み込んだ画像上にLine配置
2. 『menu > Analysis > Plot Profile』
3. Profile dialogの『live』をonにする  
→ lineを変更しながらProfileを確認できる



## 線形フィルタ

### 実習

以下の手順で画像に線形フィルタを適用してください  
フィルタ係数を変化させその効果を確認してください

0. 画像を読み込む

1. menu > Process > Filters > Convolve
2. Dialogで係数を編集する



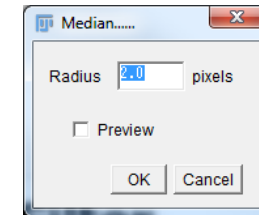
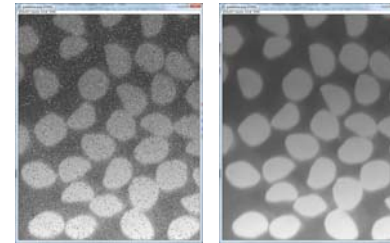
## Median Filter

### 実習

以下の手順でMedian Filterを適用し、効果を確認して下さい

0. 画像を読み込みグレースケールに

1. menu > Process > Noise > Salt and Papper
2. menu > Process > Filters > Median
2. Dialogから窓サイズを指定

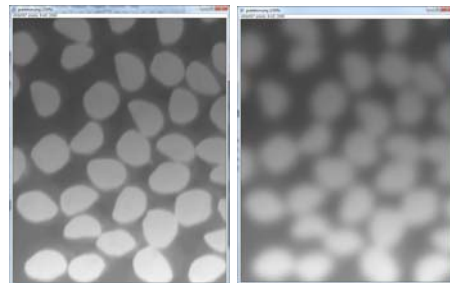


## Gaussian Filter

### 実習

以下の手順でGaussian Filterを適用し、効果を確認して下さい

1. 画像を読み込みグレースケールに
2. Menu > Process > Filters > Gaussian Blur
3. Dialogから窓サイズを指定

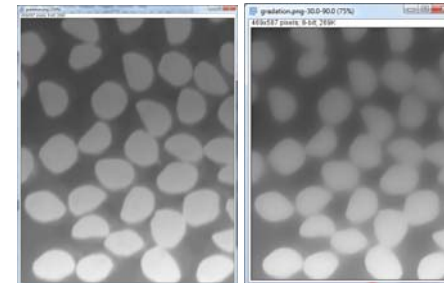


## Bilateral Filter

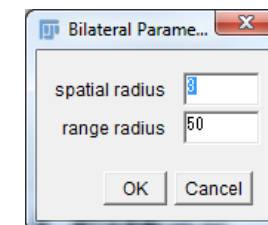
### 実習

以下の手順でBilateral Filterを適用し、効果を確認して下さい

1. 画像を読み込みグレースケールに
2. menu > Plug in > Process > Bilateral Filter
3. Dialogから窓サイズを指定



※顔画像などに適用すると効果が分かりやすいです。  
※ダイアログより spatial & range kernelの半径を指定できます。

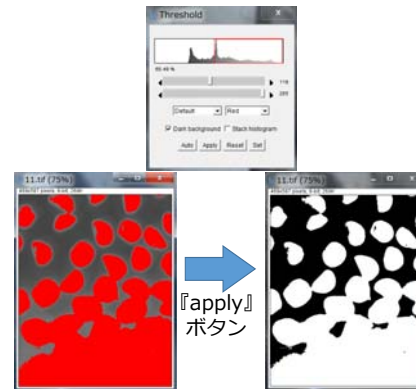


## 二値化 – gray scale

### 実習

以下の手順で画像を二値化してください

0. 画像を読み込みグレースケール化
1. menu > image > adjust > threshold
2. ダイアログから閾値（最大最小）をセット
2. ダイアログで『auto』ボタンをクリック  
※この時点で前景領域に赤色がつく  
(画像は変化せず前景領域が登録される)
3. 『apply』ボタンをクリックすると前景→白、背景→黒と二値化される  
(設定によって、LTU-invertが適用され、前景→黒・背景→白となることも)

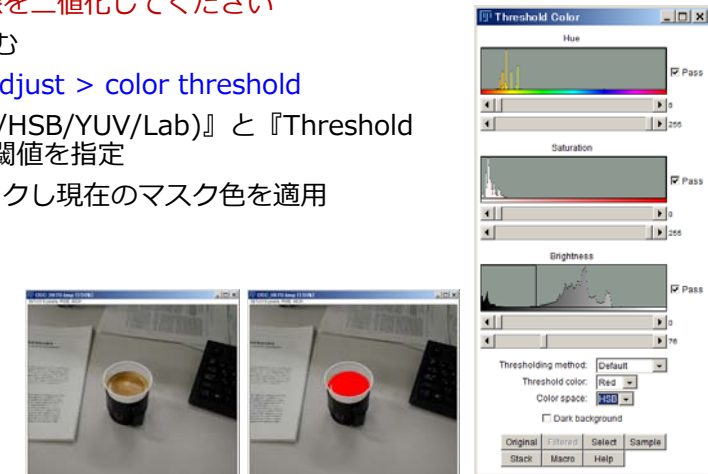


## 二値化 - color

実習: 以下の手順で画像を二値化してください

0. カラー画像を読み込む
1. menu > image > adjust > color threshold
2. 『Color space(RGB/HSB/YUV/Lab)』と『Threshold color(マスクの色)』, 閾値を指定
3. 『filtered』をクリックし現在のマスク色を適用

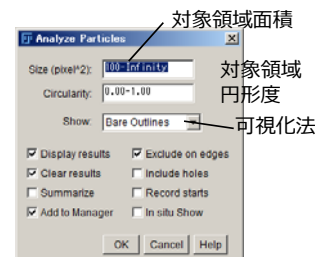
『Pass』の意味は...  
チェックすると、閾値内が前景に  
チェックを外すと閾値内が背景に



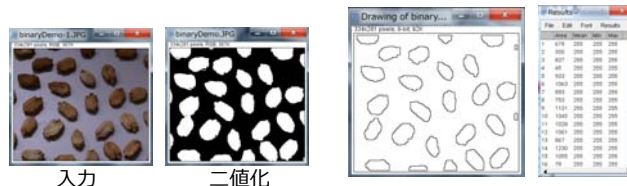
## 非連結領域解析

実習: 以下の手順で、非連結領域を解析してください

0. 画像を読み込み、グレースケール化
1. menu > Image > Adjust > thresholdで二値化しておく
1. menu > Analyze > Analyze particleを選択
2. Dialogから「対象領域サイズ/円形度/その他」を指定  
Display results / Clear resultsはチェックする  
Exclude on edgeをチェックすると 画像の端のparticleは無視される
3. 対象領域数と各領域の面積・輝度値が表示される



※円形度 =  $4\pi(\text{面積}/\text{円周}^2)$



## Morphological operation (二値画像)

- 二値画像からノイズを取り除くために良く用いられる手法
- Erosion → 領域を侵食させる
- Dilation → 領域を膨張させる





## Morphological operation (二値画像)

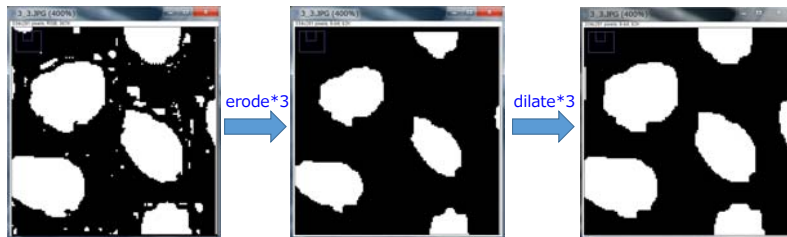
**実習：** Morphological operationの効果を確認してください

0. 画像を読み込み二値化する (menu > image > adjust > threshold)

1. erosion を適用 menu > Process > Binary > erode

2. dilation を適用 menu > Process > Binary > dilate

※erosion してから dilationすると元に戻りますか？



## 背景グラデーションの除去

• Image-Jには 背景グラデーション除去手法が実装されている

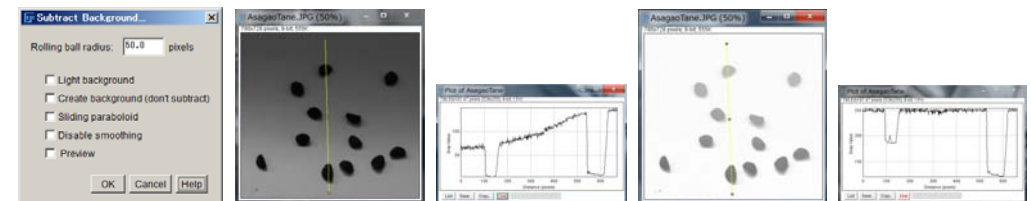
参考URL: <http://imagej.nih.gov/ij/docs/menus/process.html#background>

**実習：** 以下の手順で画像のグラデーションを除去してください

0. 画像を読み込みグレースケール化

1. menu > Process > Subtract Backgroundを選択

2. DialogからBall radius (前景領域の半径) を指定し『ok』



## 画像の加算・減算

• ImageJには画像の足し算・引き算を行なう機能が実装されている

**実習：** 以下の手順でDoGを計算せよ

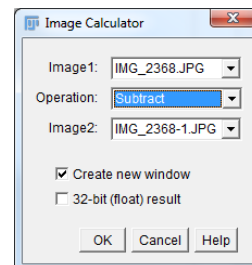
0. 画像を読み込みグレースケール化

1. Menu > image > duplicate, okを押して画像を複製

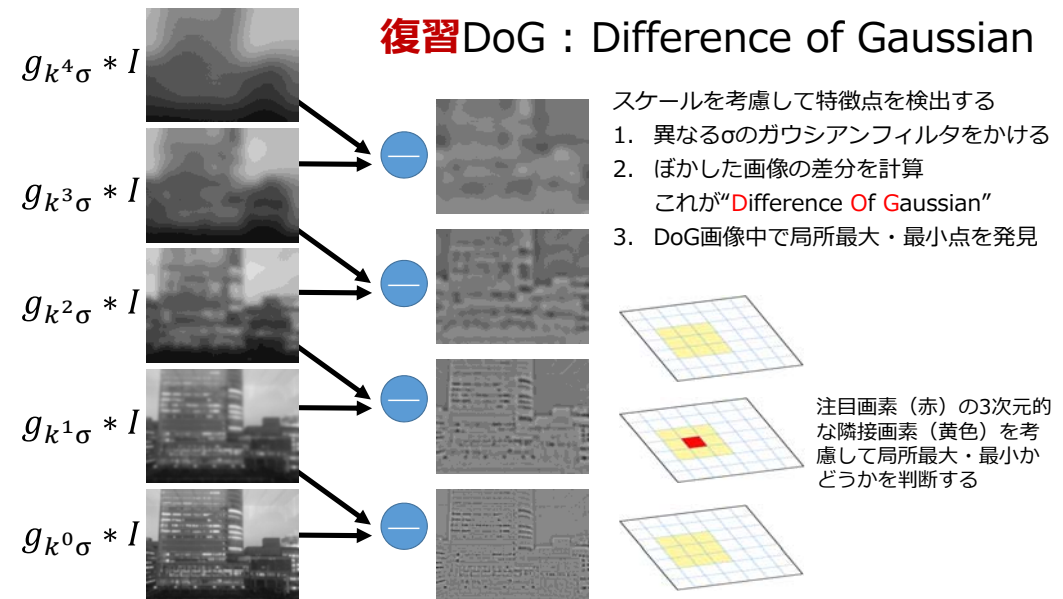
2. 二枚の画像に半径の異なるガウシアンフィルタをかける

3. Menu > image > Image Calculatorをクリック

4. ダイアログから画像と演算を選択し実行する



## 復習DoG : Difference of Gaussian



## 画像の加算・減算

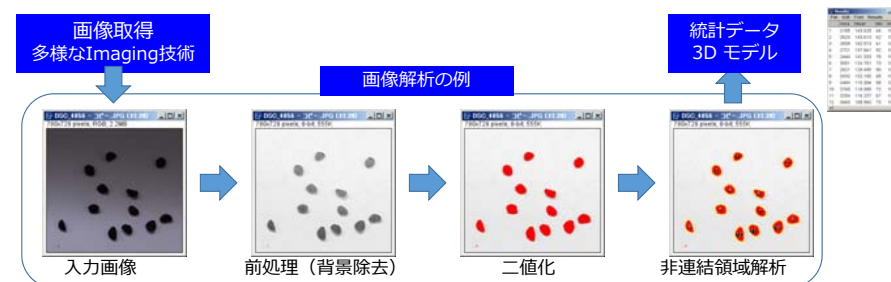
Skip?

0. 2枚の同じサイズのグレースケール画像を読み込む
  - 例えば...
  - 画像を読み込み『menu > Image > Duplicate』で複製
  - 片方に『erosion』,もう一方に『dilation』をかける
1. 『menu > Process > Calculator Plus』を起動し Operationを指定し『ok』



## まとめ：ImageJ を用いた画像処理

- ImageJを用いて講義内で紹介した手法の幾つかを実際に体験した
- 講義で紹介した手法は『比較的簡単に利用できる』ことや『知らないうちに使っている』こと、を体験を通して知ってほしい
- 画像処理を取り扱う研究室では、新しい画像処理ツールの開発や、画像処理手法の新しい応用法について研究開発している



## 課題

- 時間のある人はやってみてください
- 講義中に回答を紹介します

## 課題1：以下の線形フィルタを設計せよ

次の機能を持つ線形フィルタを設計しそれが動く理由を簡潔（1行程度）に述べよ

- + カーネル係数と理由を回答
- + 例題を参考に



入力画像  
1orig.jpg



例)横方向エッジ検出



1\_1) 斜めエッジ検出



1\_2) 先鋭化

## 課題2: ノイズ除去をせよ

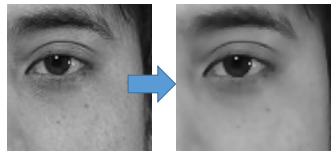


2\_1noise.bmp

**課題2-1.** ノイズ画像に何らかのFilterをかけ、元の画像に近づけよ。



2\_2trgt.jpg



**課題2-2.** 人物画像 に何らかのFilterを掛け、あらを消せ。  
(自分の顔画像でもやってみてください)

## 課題3: 種の数进行数えよ

- 以下の三枚の画像にある種の数进行Image-Jを利用して数え  
その数と利用した処理の流れを回答せよ



3\_1.jpg



3\_2.jpg



3\_3.jpg

※うまくパラメータを調整すると正解しますが  
数が確実にあっている必要はありません  
※数え間違いが生じる場合にはその原因を考察してください