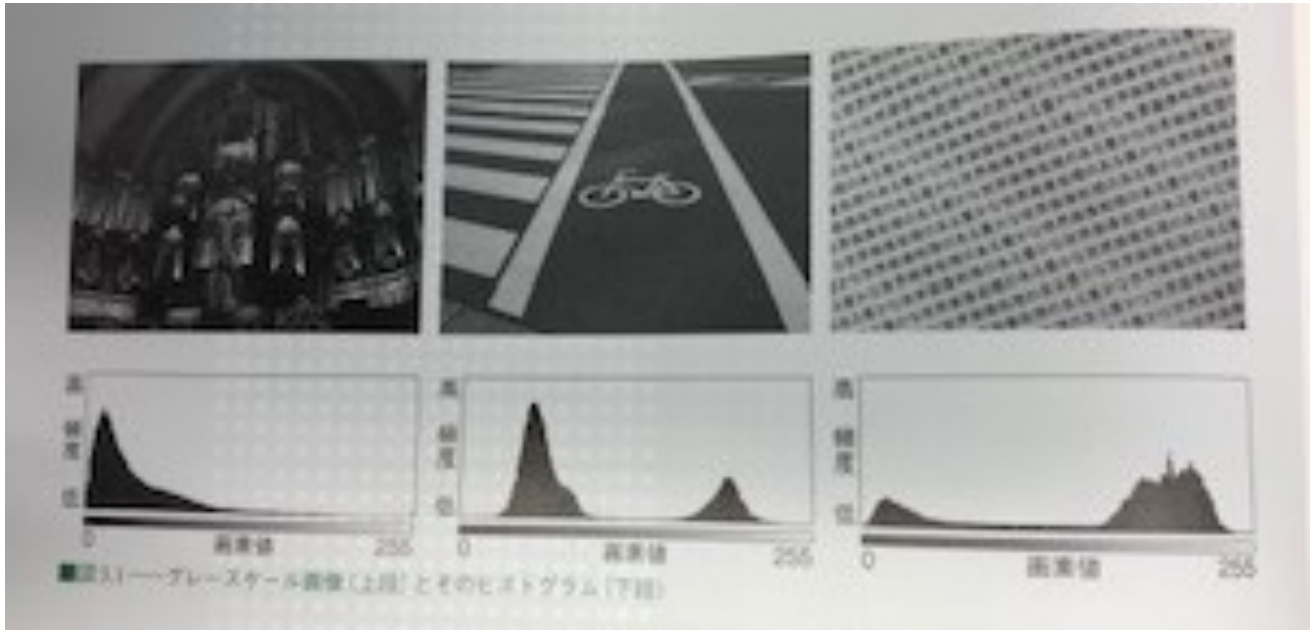
メディアコンピューティング2022

第3回　画像の統計量とヒストグラム

画像の性質を表す諸量

■ヒストグラムと統計量

- ➢ヒストグラム（または濃淡ヒストグラム）
  - ✓横軸：画素値
  - ✓縦軸：各画素値の頻度（その画素値をもつ画素の個数）
    - ◆画像中にどのような値の画素値がどれほど含まれているか



演習03-1

■画像のヒストグラムの作成

- ➢`cv::Mat make_histogram(const cv::Mat src, const cv::Size hsize)`
  - ✓ `src`: ヒストグラムを作りたい画像
  - ✓ `hsize`: ヒストグラム用画像のサイズ
  - ✓ 戻り値: ヒストグラム画像

■ヒストグラムの描画

- ➢`void show_histogram(const string winname, const cv::Mat hImage)`
  - ✓ `Winname`: ヒストグラムのウィンドウ名
  - ✓ `hImage`: ヒストグラム画像

■histogramという名前空間を作って

- ➢`cv::Mat histogram::make(const cv::Mat src, const cv::Size hsize);`
- ➢`void histogram::show(const string winname, const cv::Mat hImage)`

としてもよい

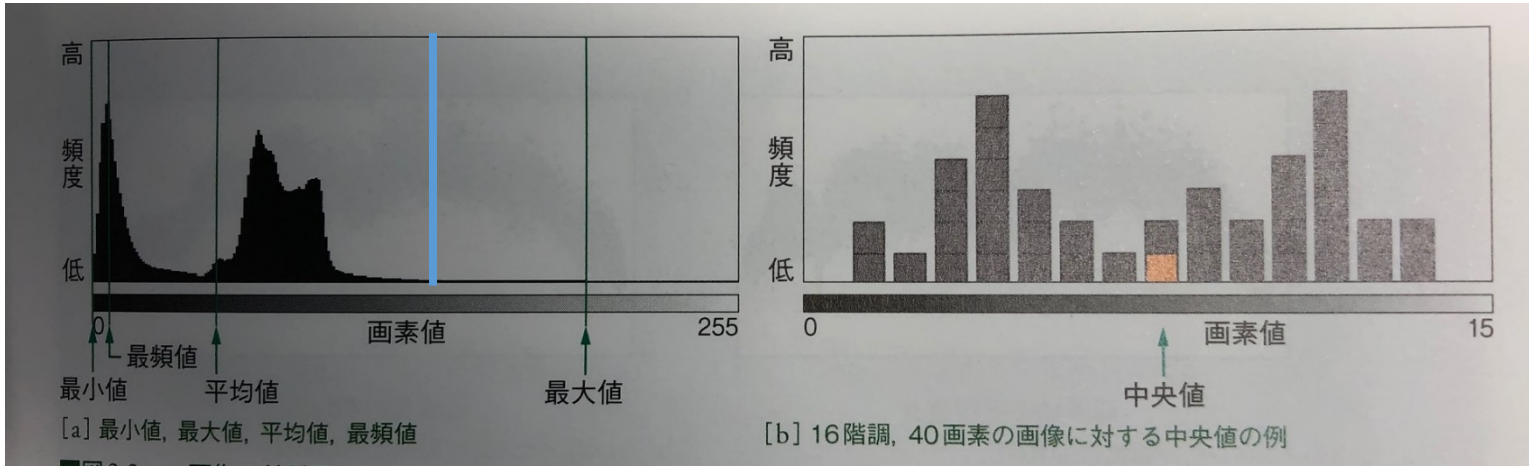
（こっちの方がオブジェクト指向チック）

画像の性質を表す諸量

■ヒストグラムと統計量

➢統計量

- ✓最小値，最大値
- ✓平均値
- ✓中央値
- ✓最頻値
- ✓分散（標準偏差）



◆ヒストグラムからも求めることができる

- ✓が，あまりやらない
- ✓一度ヒストグラムを作れば，複数の統計量を別途求めるときはヒストグラムの方が速い

演習03–2　統計量を計算する関数を作る

Stats.hを作り，そこに記述

- 最小値，最大値
  - int min(cv::Mat src)
  - int max(cv::Mat src)
- 平均値
  - double average(cv::Mat src)
- 中央値
  - int median(cv::Mat src)
- 最頻値
  - int mode(cv::Mat src)
- 分散（標準偏差）
  - double variance(cv::Mat src)
  - double stdev(cv::Mat src)

できる人はSTLで表現（Matの型に依らない関数定義）
https://docs.opencv.org/3.4/d3/d63/classcv_1_1Mat.html

演習03–2　統計量を計算する関数を作る

■局所統計量

- ➢実際の画像処理では，画像全体の統計量に加えて，局所領域の統計量が欲しい場合が多い
  - ◆実装方法
    - ✓局所用に新たに関数を作る
      - 関数名の頭に"local"を付けるなど
    - ✓同じ関数名で局所領域の原点と矩形サイズを引数に追加（関数のオーバーロード）
    - ✓追加した引数にデフォルト値を設定（関数は増やさない）

課題03

- 演習03-2を完成させる
- ヒストグラムから各種統計量を計算する関数の作成
  - 関数のオーバーロード
  - 局所統計量はヒストグラムを作るオーバーヘッドが大きいので作成しなくても良い
- `Stats.h`を「課題03提出箱」に提出