二値画像処理

森本，塚田，澤野

本日の目標

- 二値画像処理
  - 二値化画像とは
  - 二値化処理

➡ **固定閾値 (しきいち) 法**

➡ **pタイル法**

➡ **判別分析法**
（cv::threshold関数使用）

二値化 (p. 59)

- 白(255)もしくは黒(0)の画像に
  変換すること
- グレースケール画像を二値化する場合,
  ある値を基準に画素値の変換

  ➡ **基準の画素値: しきい値**

- 画像から**位置**・**形状**の情報に変換
  - 幾何学的に取り扱うことが可能

二値画像はどんなとき使うか

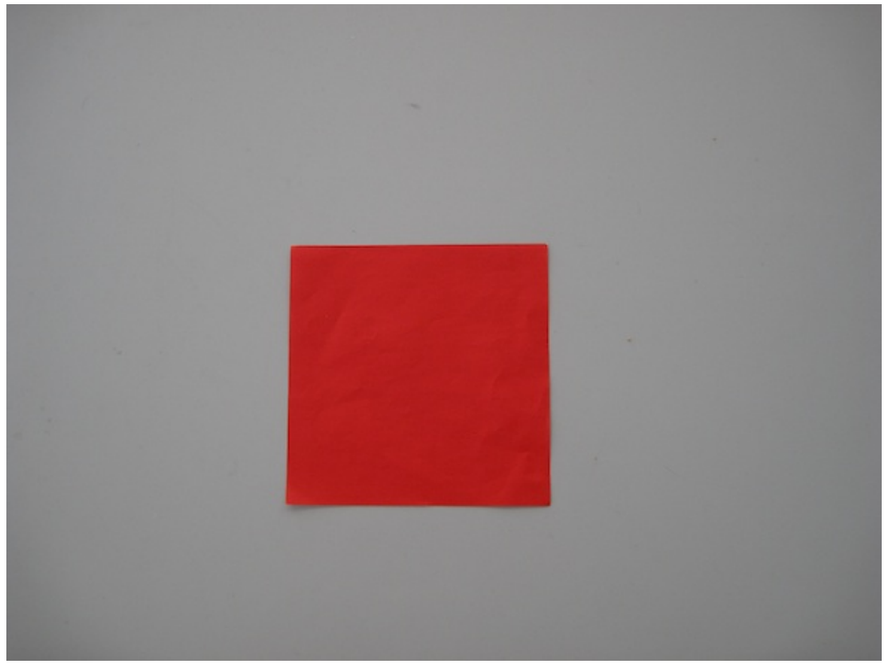
- 画像の明瞭化 (見た目)

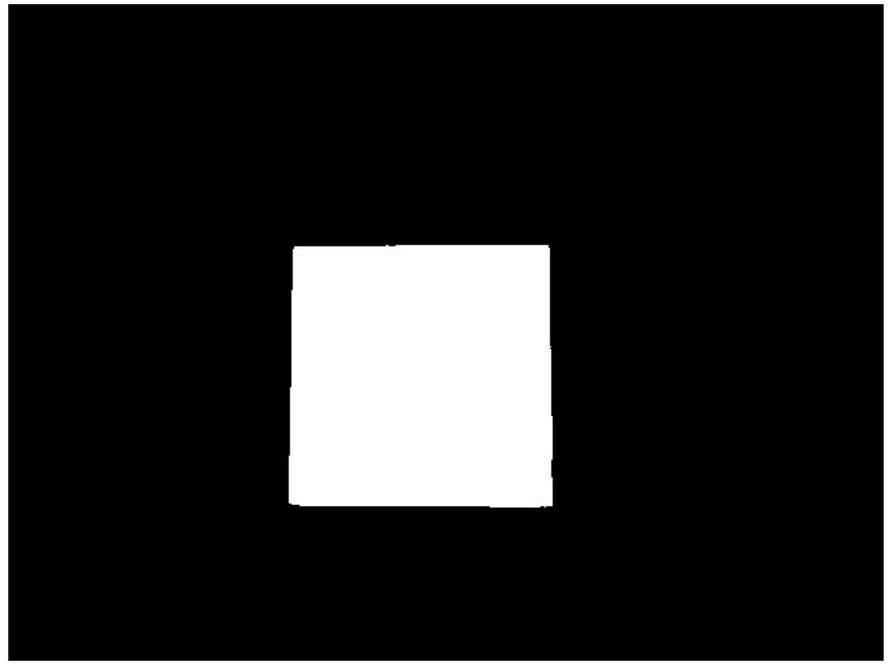
入力画像

二値画像

応用例 (形状)

- 部品の欠陥を見つける処理
- 領域分割，抽出
- マスク処理 (クロマキー処理)



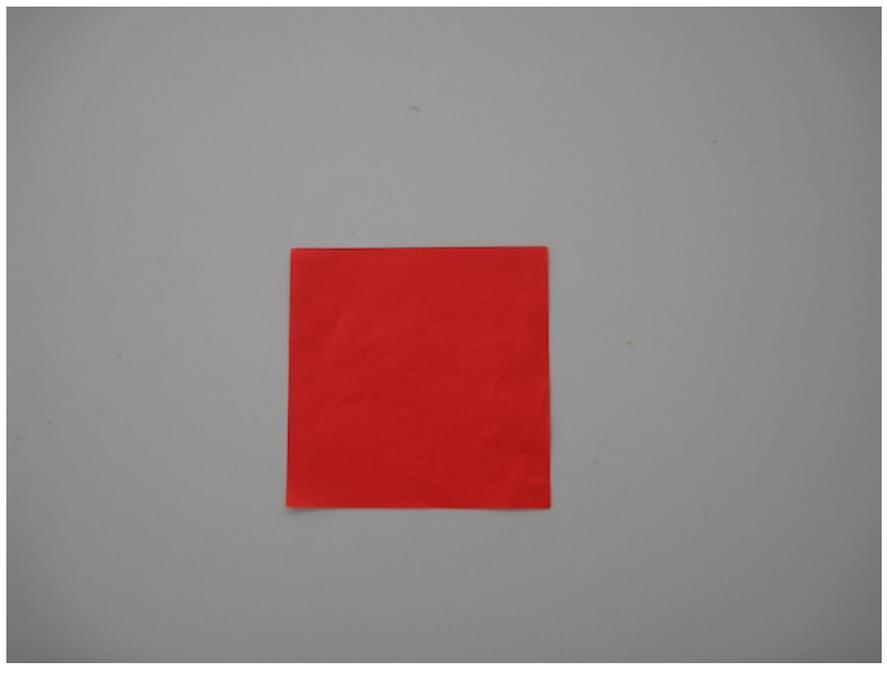


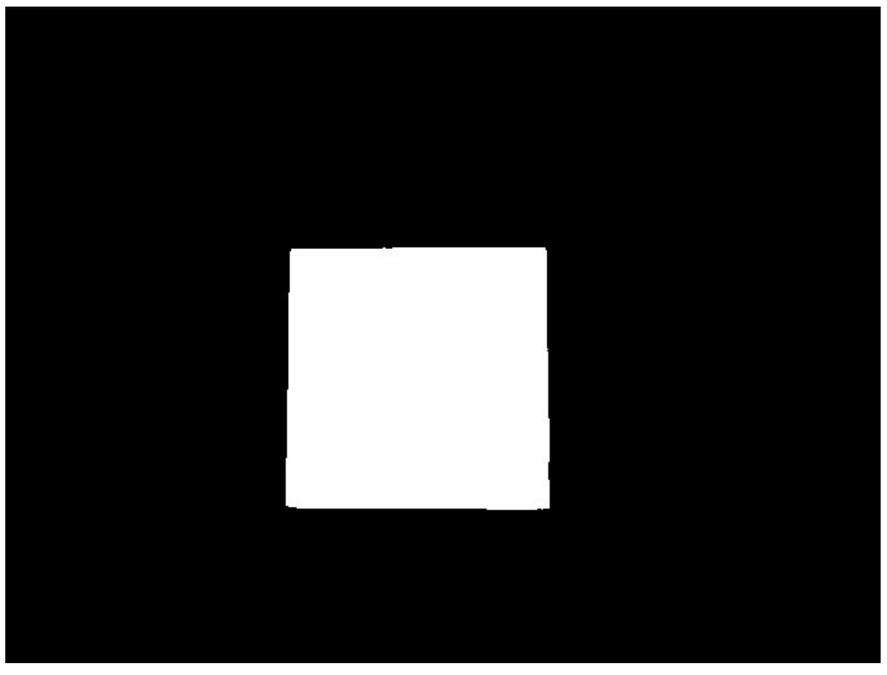
固定しきい値法

- 固定しきい値で二値化
- 環境が変わらないことが前提

演習

- 固定しきい値(t=100)を使用して二値化
- プロジェクト名: fixTh
- 参考: 教科書p. 62, プログラム5.1
- 画像: red_rectangle.jpg





プログラム (一部)

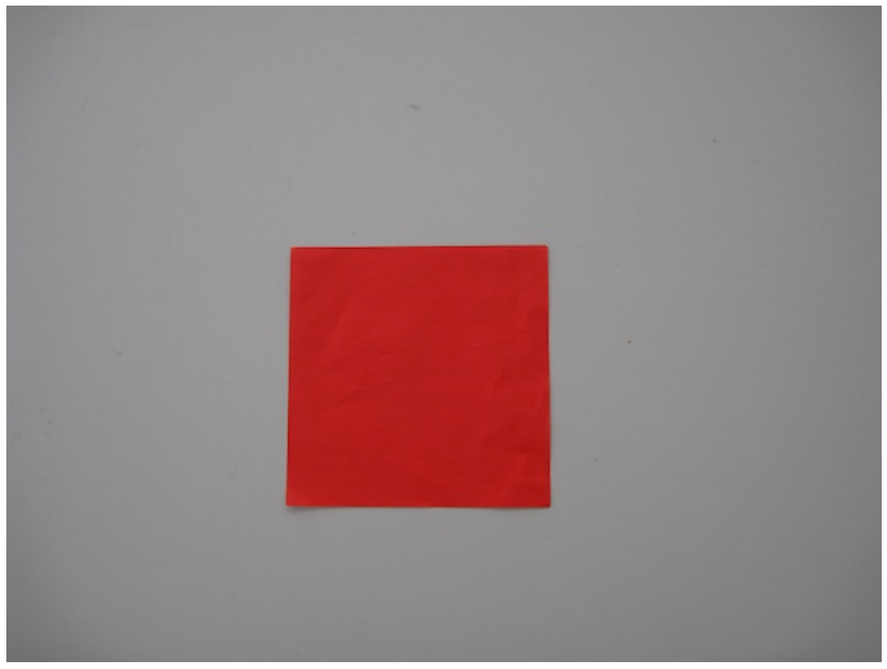
```
//二値画像のメモリ確保
bin_img = cv::Mat(gray_img.size(), CV_8UC1);

//二値画像に変換
for (int y=0; y<gray_img.rows; y++) {
    for (int x=0; x<gray_img.cols; x++) {
        if (gray_img.at<uchar>(y, x) < TH) {
            bin_img.at<uchar>(y, x) = HIGHVAL;
        }else{
            bin_img.at<uchar>(y, x) = LOWVAL;
        }
    }
}
```

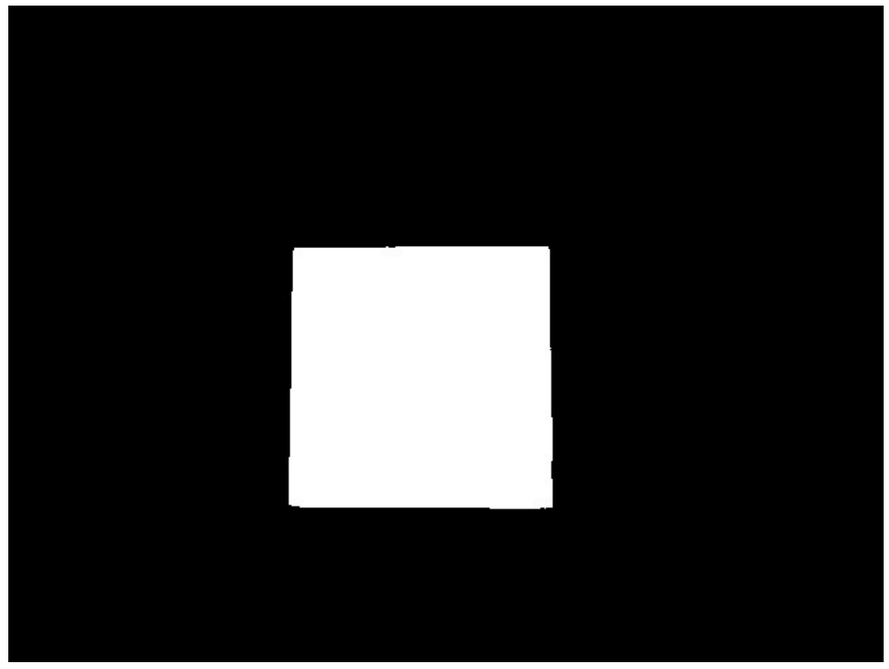
- 255と0を入れ替えてみよう!

演習 (OpenCVの関数利用)

- 固定しきい値(t=100)を使用して二値化
- プロジェクト名: fixThOpenCV

★cv:thresholdの利用 (説明します)





```
#include <iostream>
#include <opencv2/opencv.hpp>
#define FILE_NAME "red_rectangle.jpg"

#define HIGHVAL (255) //白画素の値

#define TH (100) //閾値

int main(int argc, const char * argv[]) {
    //画像をグレースケールで入力

    cv::Mat gray_img, bin_img;
    gray_img = cv::imread(FILE_NAME, 0);
    if (gray_img.empty()) { //入力失敗の場合

        fprintf(stderr, "File is not opened.\n");
        return (-1);
    }

    //二値化

    cv::threshold(gray_img, bin_img, TH, HIGHVAL, cv::THRESH_BINARY);
    //出力

    cv::imshow(“grayscale image", gray_img);//画像の表示

    cv::imshow(“binary image”, bin_img);//画像の表示

    cv::waitKey(); //キー入力待ち (止める)

    return 0;
}
```

しきい値処理関数の説明

- 関数紹介

```
cv::threshold(入力画像, 出力画像, しきい値, max_value, オプション);
```

  - オプション

    ➡ cv::THRESH_BINARY: しきい値以上をmax_valueに

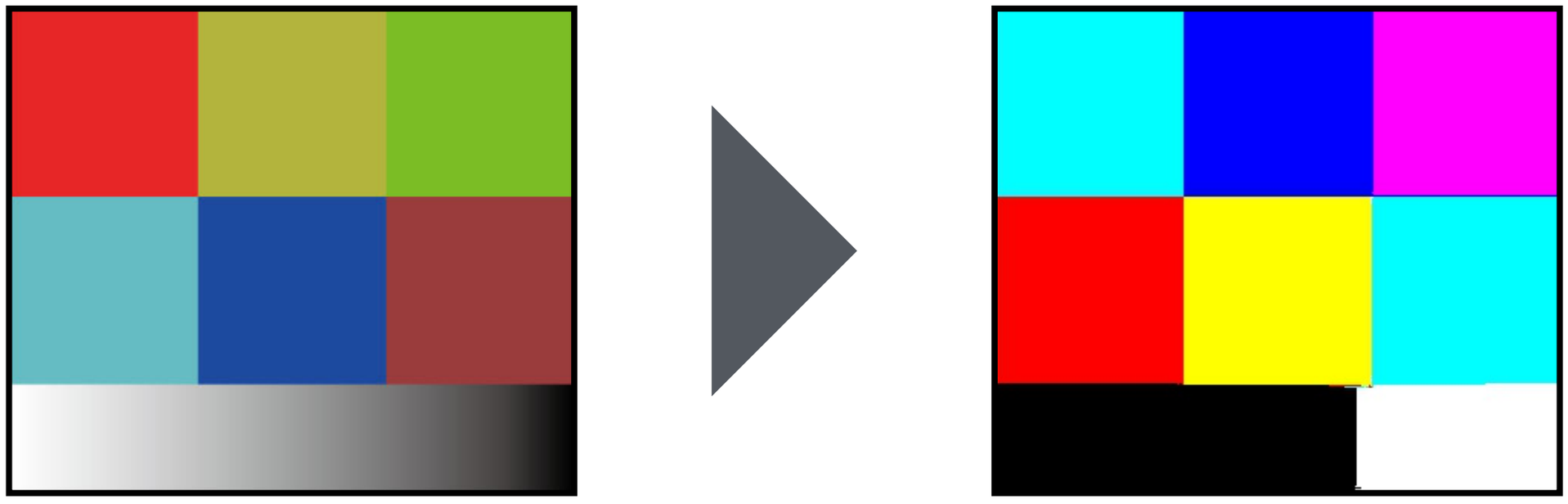
    ➡ cv::THRESH_BINARY_INV: しきい値以下をmax_valueに

    など

```
//しきい値処理. しきい値以上を255にする
cv::threshold(src_img, dst_img, THRESHOLD, 255, cv::THRESH_BINARY);
```

練習

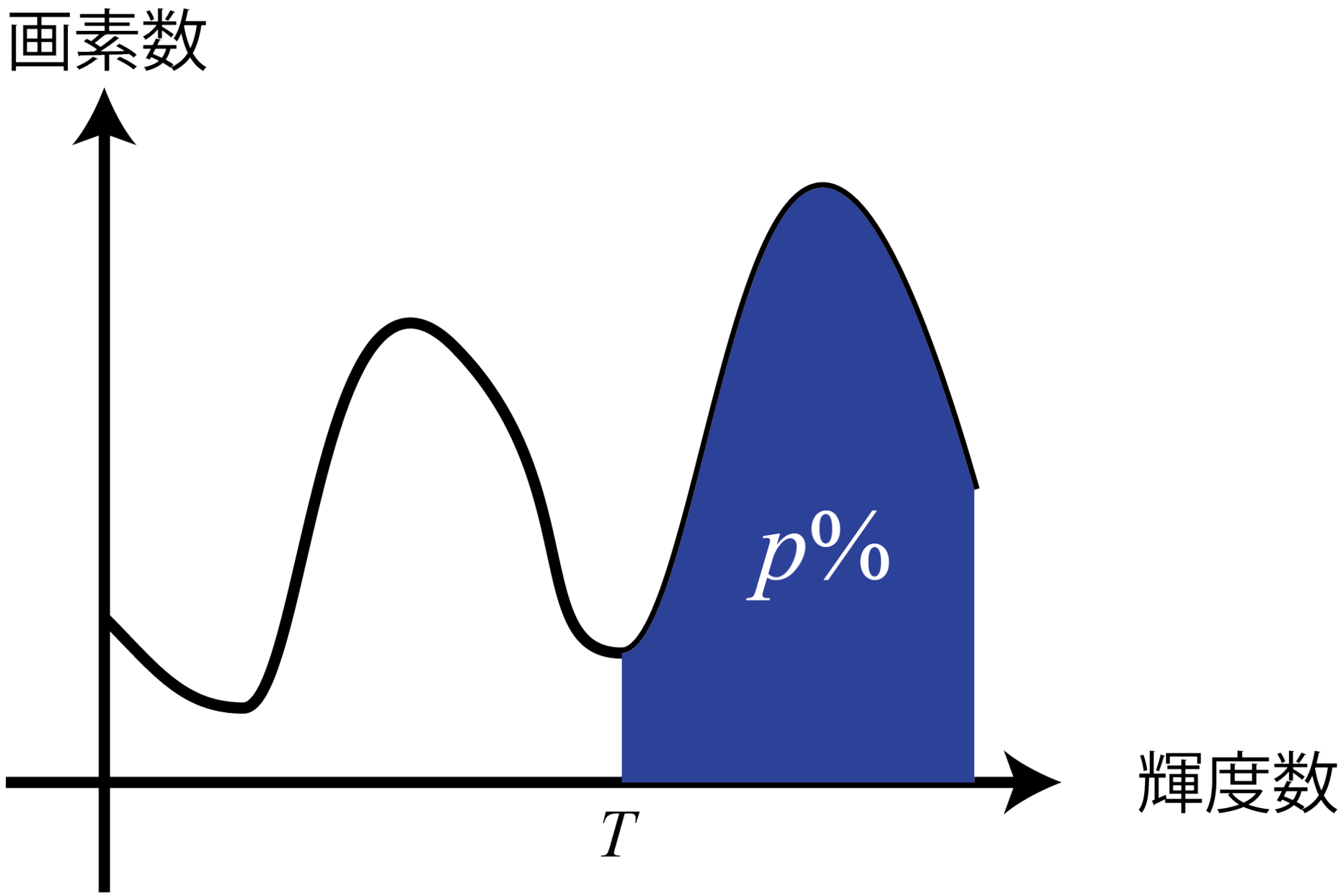
- cv::threshold (固定閾値) をカラー画像に適用
- fixThOpenCVの入力をカラーに変更
- 画像: canvas.jpg
- 閾値: 127に変更. cv::THRESH_BINARY_INV



入力画像　　適用後

pタイル法 (p. 61下)

- 物体と背景の割合が予測できる場合に使用

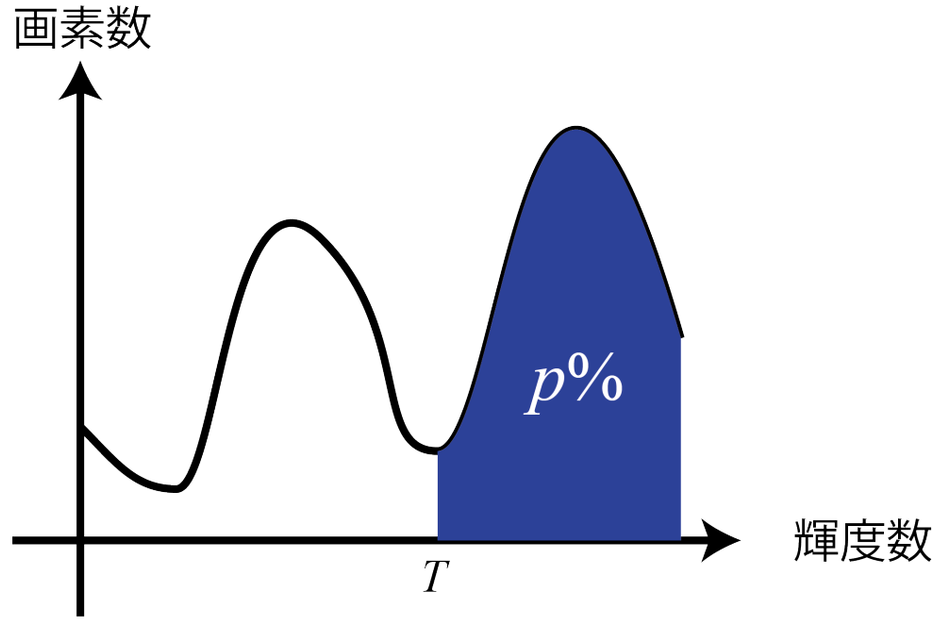


pタイル法概要

- 面積の比率pによってしきい値を決定する方法
- ヒストグラムで以下を考える
  - 全体の面積 (画像サイズ) S
  - 対象の面積 S_0

$$p = \frac{S_0}{S}$$

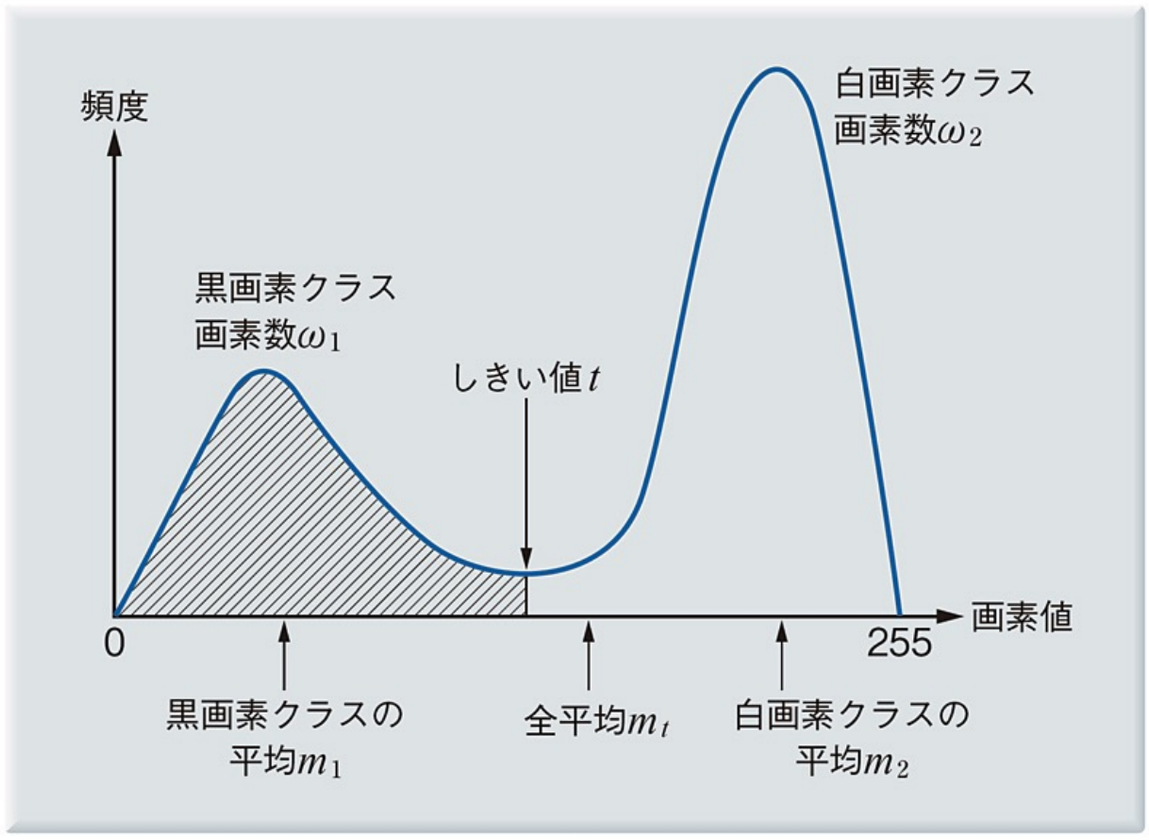
- 全体の面積に対して対象の割合が不変の場合



判別分析法

- 黒画素と白画素の分布を見つけ，その分布の分離度が最大となるしきい値を見つける方法

■図9.4——判別分析法によるしきい値 t の決め方



「ディジタル画像処理　改訂新版」2015年／公益財団法人 画像情報教育振興協会（CG-ARTS協会）

判別分析法の概要

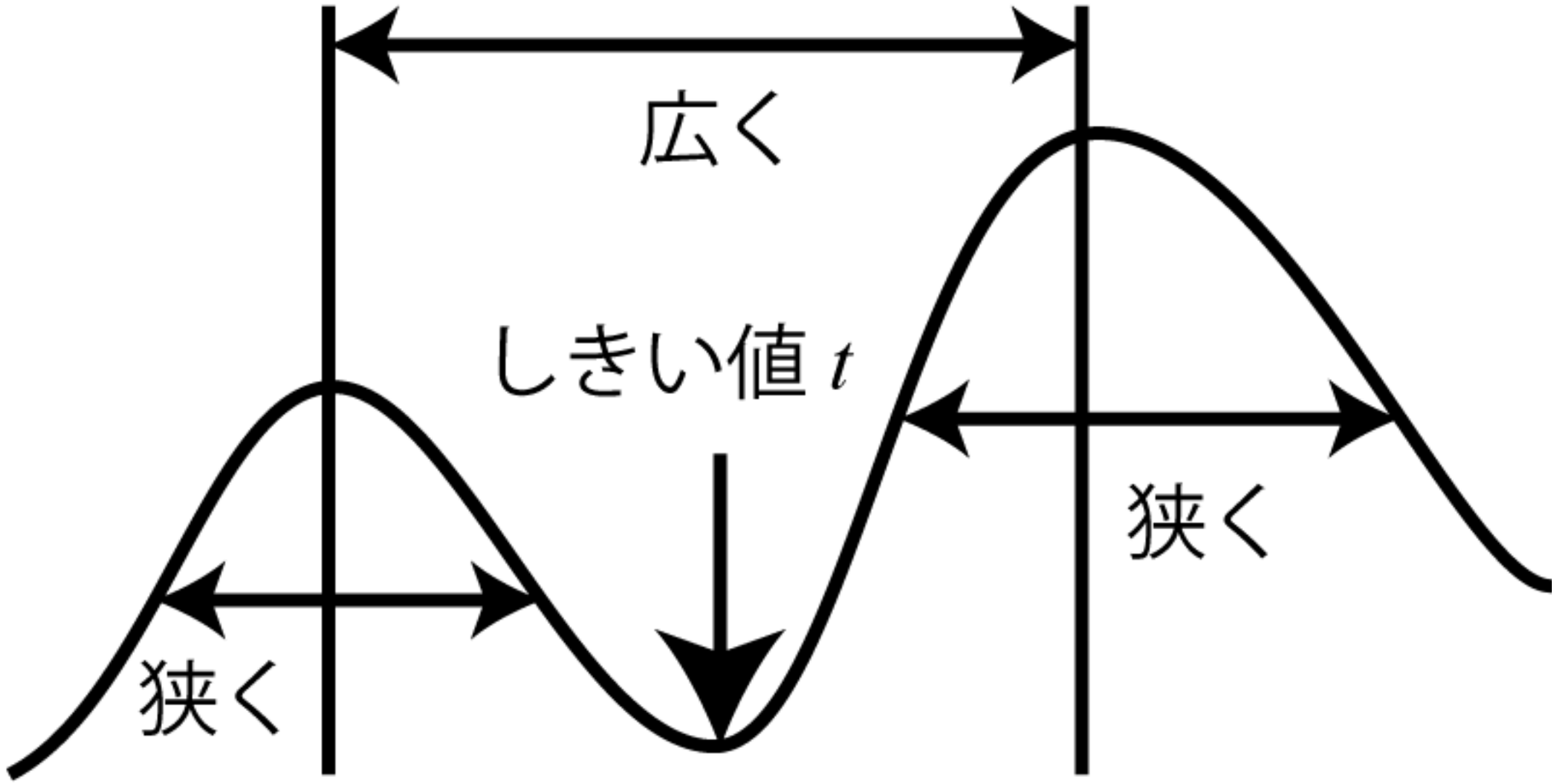
- クラス内 (山) の画素の値はほぼ同じ
- クラス間の画素値は離れる

$$\frac{\text{クラス間分散}}{\text{クラス内分散}} = \text{最大}$$

- 大津の判別分析法として有名

大津, "判別および最小２乗基準に基づく自動しきい値選定法", 電子通信学会論文誌, Vol.J63-D, No.4, pp.349-356, 1980.

判別分析法の概念



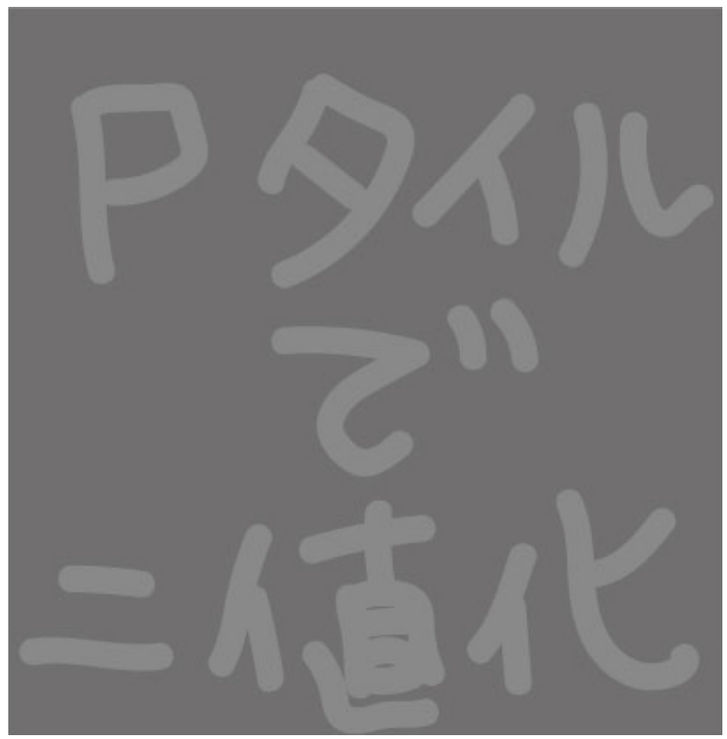
演習

- 判別分析法による二値化
- プロジェクト名: otsu
- プロジェクトfixThOpenCVのソースをコピーすること
- しきい値を以下のように設定 (1行のみ変更)

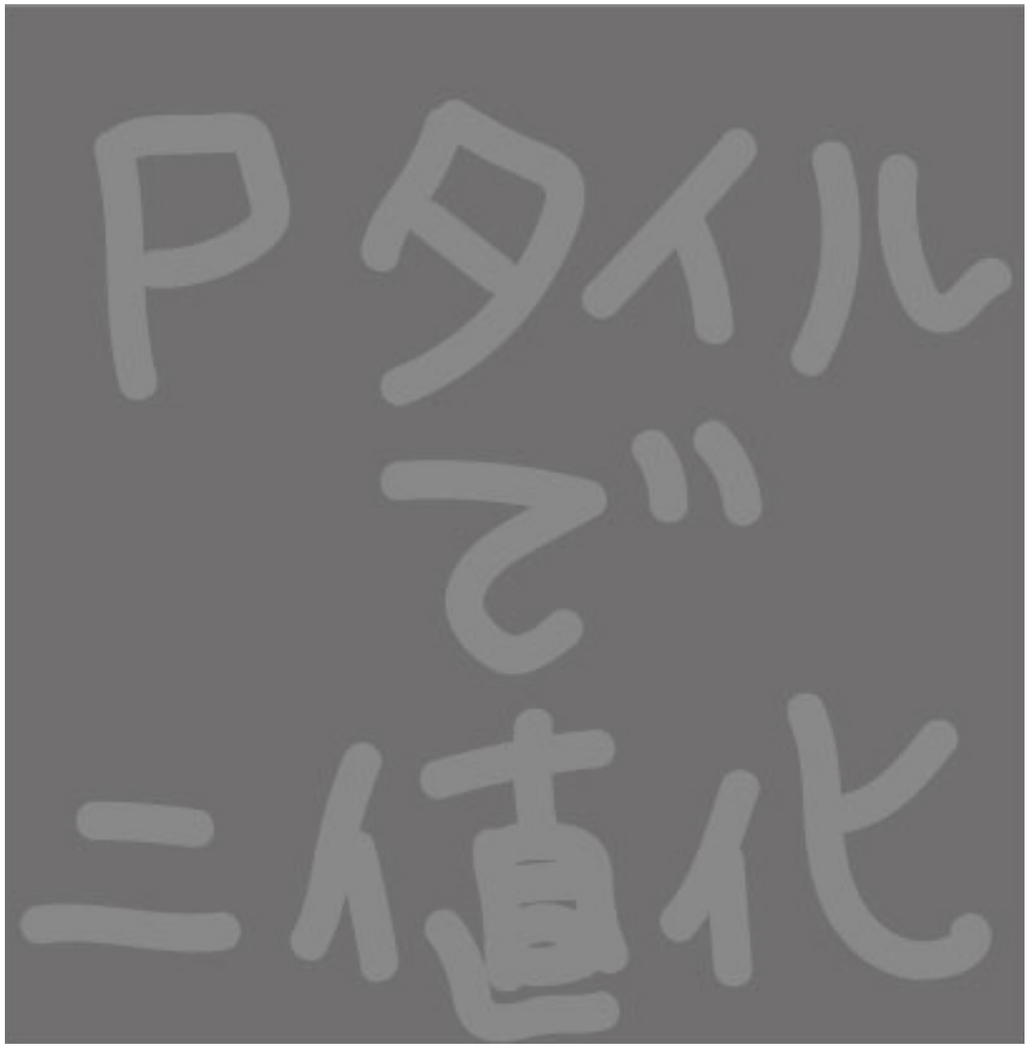
```
cv::threshold(src_img, dst_img, 0, 255,
        cv::THRESH_BINARY | cv::THRESH_OTSU);
```

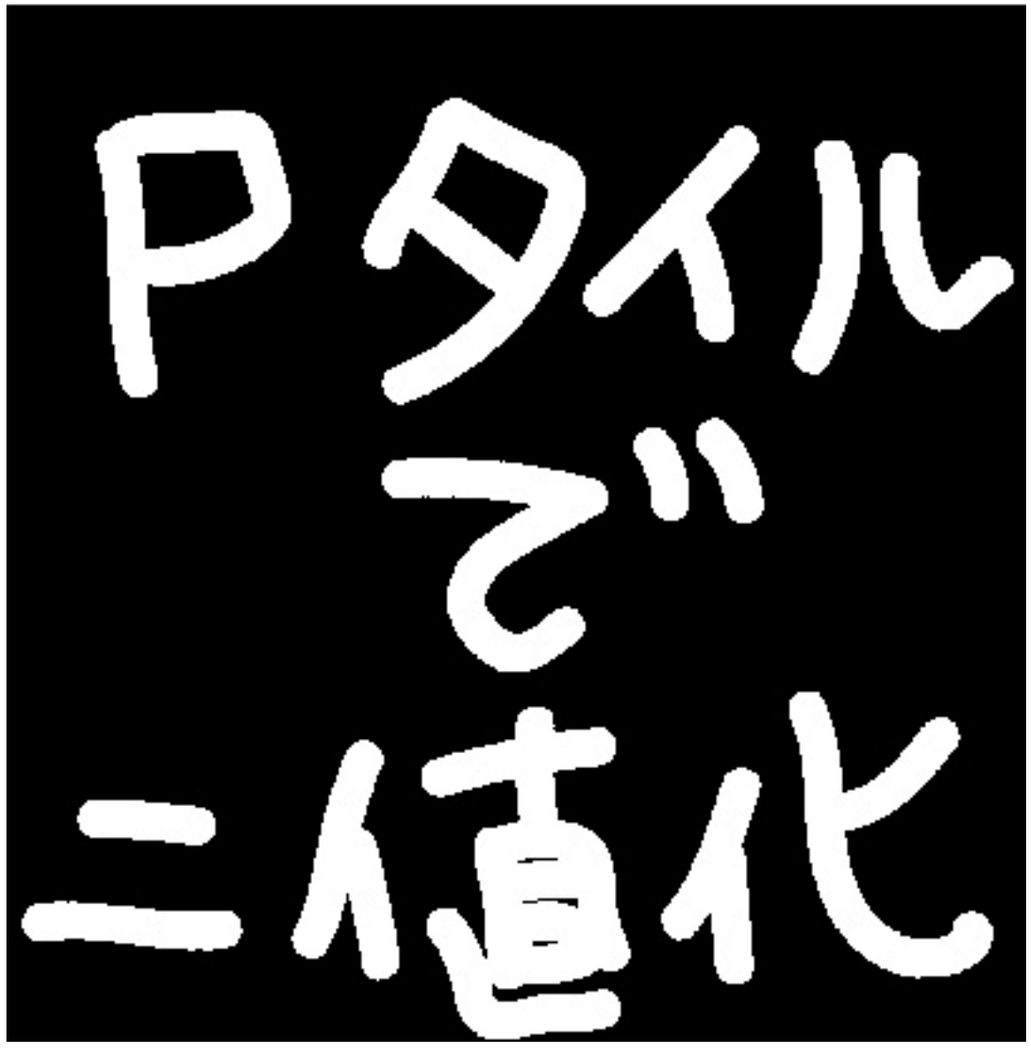
課題1

- 背景と文字の輝度値比率が4:1である画像に対して、pタイル法を用いて二値化せよ
- 入力画像: ptile.jpg
- 提出ファイル
  - プログラムソース:
    08_01_ptile_学籍番号.cpp
  - 出力画像:
    08_01_output_学籍番号.jpg (jpg以外も可能)
  - テキストファイル (pタイルで求められた閾値):
    08_01_info_学籍番号.txt



課題1の出力例





課題1の進め方

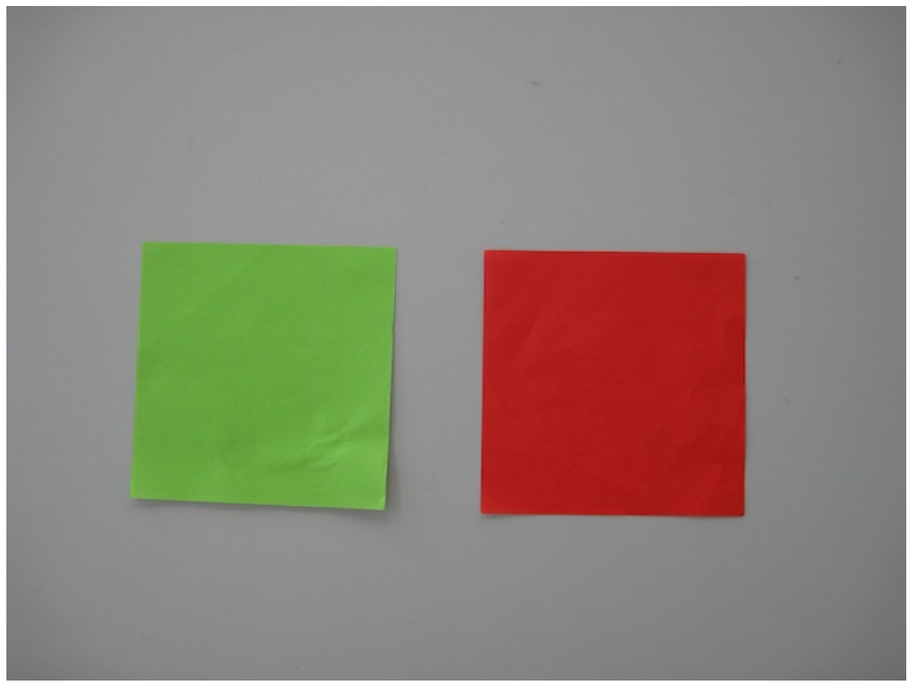
- ヒストグラムの生成 (配列)
- 全体面積のp%から，該当の面積の取得
  - 該当の面積 = 全体の面積 x p
- ヒストグラム用の配列を使用して，
  度数0(背景)から走査して，
  該当の面積に到達する度数(閾値)を求める
- 二値化

課題1のヒント

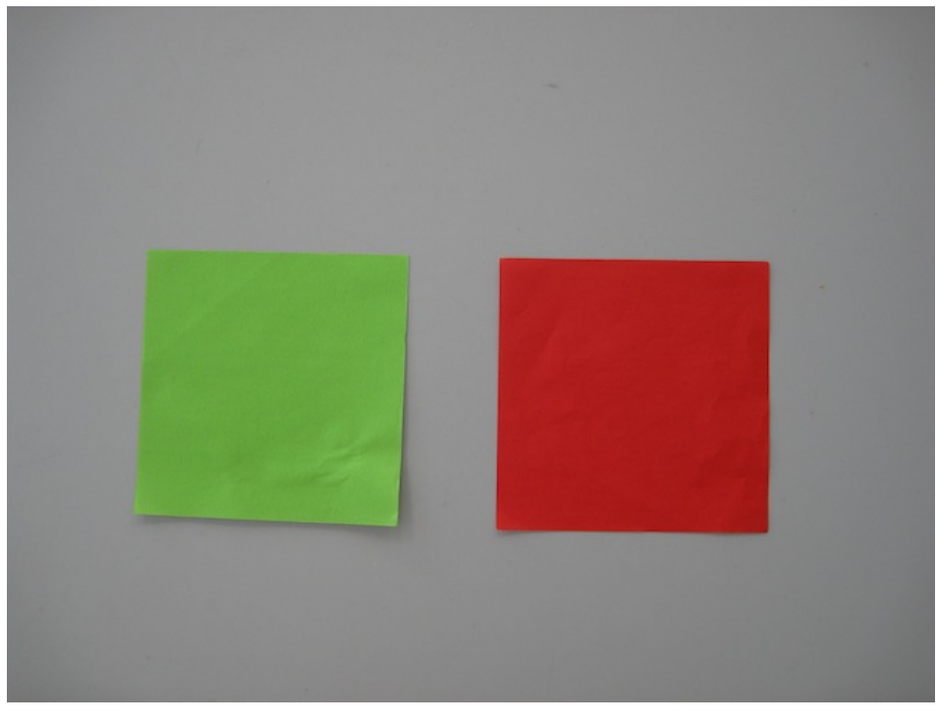
- 背景と文字の輝度値比率が4:1が提示されているので，プログラム中に4, 1が記載されていることが望ましい．以下に例を示す．
  - double p = 4.0 / (4.0 + 1.0);
- ヒストグラムのループから度数の和の求め方
  - sum += hist[i]; //度数の和
- 度数を記載したテキストの提出を忘れない

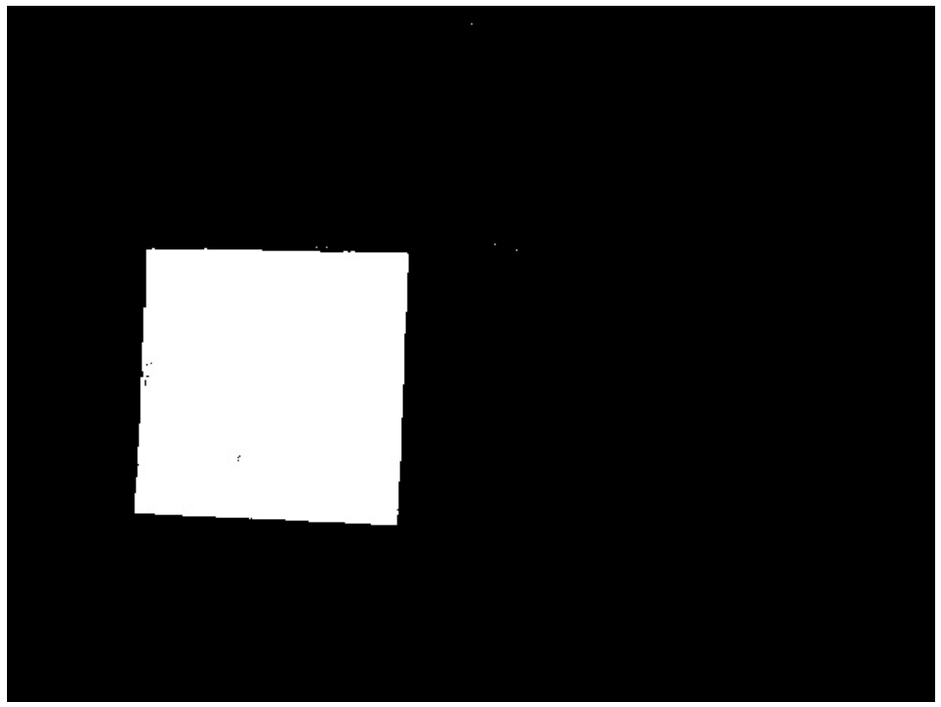
課題2 (チャレンジ課題, 提出自由)

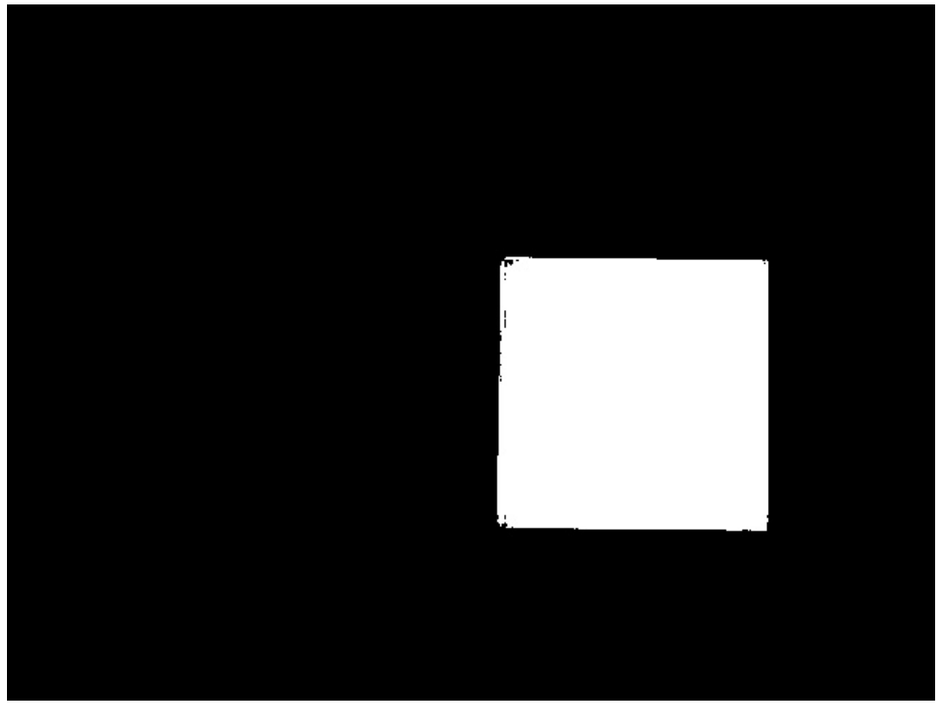
- 赤矩形と緑矩形を分離した二値画像を表示するプログラムを書け. (多少の精度の厳密性は問わない)
- 画像: red_green_rectangle.jpg
- 提出ファイル
  - プログラムソース: 08_02_bin_学籍番号.cpp
  - 出力画像 (キャプチャ可): 08_02_output_学籍番号.jpg (jpgでなくてもよい)
  - 説明ファイル: 08_02_method_学籍番号.txt



課題2の出力例







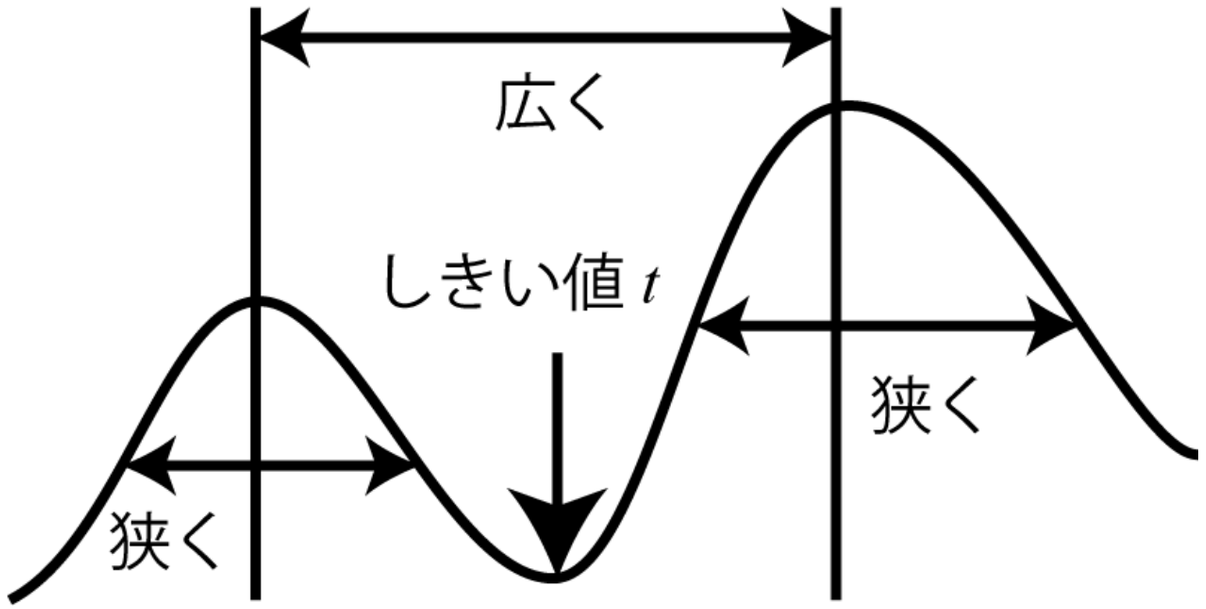
課題2の説明

- 今回の課題はいくつか方法があります。手法のアルゴリズム (手順) を説明用ファイル (08_02_method.txt) に記載して、対応する箇所をプログラムにコメント文で記載してください。
- 手法の例
  - 色値のヒストグラムを作って，色値の指定を確認後，範囲指定により二値化

付録

判別分析法の参考資料1

- クラス0 (黒): 平均 m_0，分散 σ_0^2，画素数 w_0
- クラス1 (白): 平均 m_1，分散 σ_1^2，画素数 w_1



クラス0　　クラス1

クラス内分散とクラス間分散の求め方

- クラス内分散

$$\sigma_w^2 = \frac{w_0\sigma_0^2 + w_1\sigma_1^2}{w_0 + w_1}$$

- クラス間分散

$$\sigma_b^2 = \frac{w_0 w_1 (m_0 - m_1)^2}{(w_0 + w_1)^2}$$