

画素単位での処理(2)

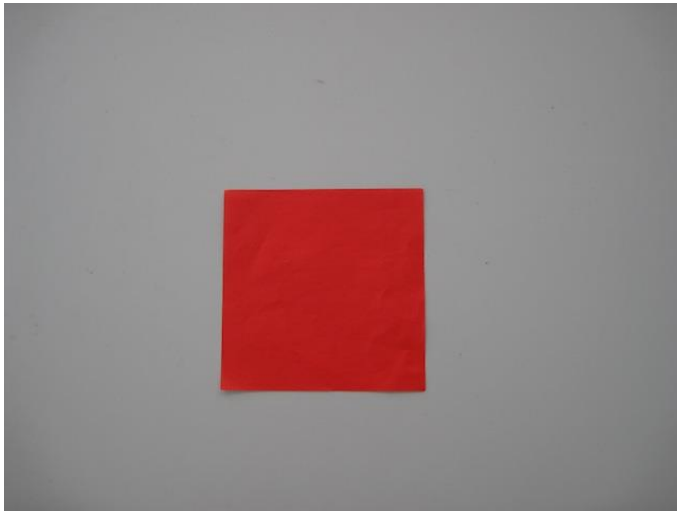
【本日の内容】

- | | |
|-------------------|------|
| ① ヒストグラムについて | 【講義】 |
| ② ヒストグラム生成 | 【演習】 |
| ③ コントラスト、コントラスト変換 | 【講義】 |
| ④ コントラスト最適化 | 【演習】 |

演習1 ヒストグラムの作成

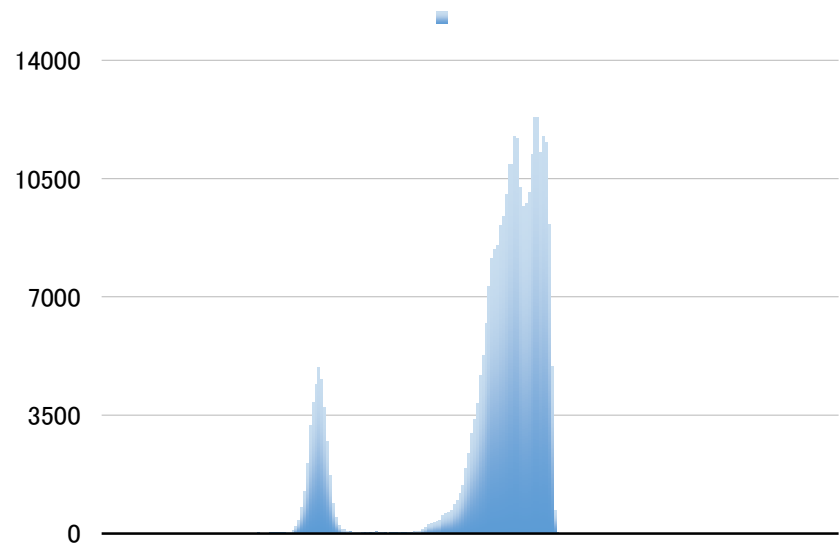
◆ プロジェクト名 : hist

課題：画像を読み込んで、ヒストグラムを作成する



red_rectangle.jpg

入力画像



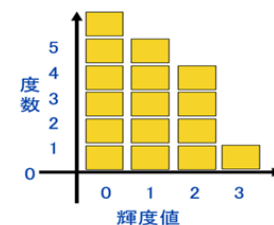
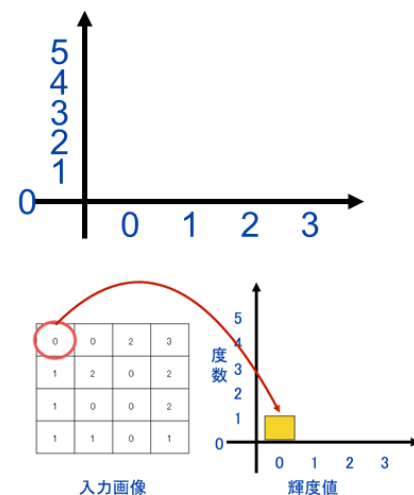
ヒストグラム

演習1 ヒストグラムの作成

◆ 処理の流れ

(プログラム中にコメント文で書く)

- ① 変数の宣言
- ② 画像をグレースケールで入力
- ③ ヒストグラム用配列の初期化
- ④ ヒストグラムの生成
- ⑤ 出力
(Numbers または Excel用に印字)



ヒストグラムの作成（前半）

```
#include <iostream>
#include <opencv2/opencv.hpp>

#define COLOR_MAX (256)

#define FILE_NAME "../Debug/red_rectangle.jpg"

int main(int argc, const char * argv[]) {
    //1. 変数の宣言
    int i, x, y; //アクセス用の変数

    int hist[COLOR_MAX]; //ヒストグラム用配列

    //2. 画像をグレースケールで入力
    cv::Mat src_img = cv::imread(FILE_NAME, 0);
    if (src_img.empty()) { //入力失敗の場合
        fprintf(stderr, "File is not opened.\n");
        return (-1);
    }
}
```

ヒストグラムの作成（後半）

//3. ヒストグラム用配列の初期化

```
for (i=0; i<COLOR_MAX; i++) {  
    hist[i] = 0;  
}
```

//4. ヒストグラムの生成

```
for (y=0; y<src_img.rows; y++) {  
    for (x=0; x<src_img.cols; x++) {  
        //画素値の取得  
        unsigned char s = src_img.at<unsigned char>(y, x);  
  
        hist[(int)s]++; //画素数の計算  
    }  
}
```

//5. 出力 (Numbers or Excel用に印字)

```
for (i=0; i<COLOR_MAX; i++) {  
    printf("%d\n", hist[i]);  
}  
}
```

ヒストグラム用配列の作成

```
#include <iostream>
```

```
#include <opencv2/opencv.hpp>
```

```
#define COLOR_MAX (256)
```

```
#define FILE_NAME "../Debug/red_rectangle.jpg"
```

```
int main(int argc, const char * argv[]) {
```

```
    //1. 変数の宣言
```

```
    int x, y; //アクセス用の変数
```

```
    int hist[COLOR_MAX]; //ヒストグラム用配列
```

```
    //2. 画像をグレースケールで入力
```

```
    cv::Mat src_img = cv::imread(FILE_NAME, 0);
```

```
    if (src_img.empty()) { //入力失敗の場合
```

```
        fprintf(stderr, "File is not opened.\n");
```

```
        return (-1);
```

```
    }
```

輝度値範囲

0~255

(横軸範囲)

画像入力

```
#include <iostream>
#include <opencv2/opencv.hpp>

#define COLOR_MAX (256)

#define FILE_NAME "../Debug/red_rectangle.jpg"

int main(int argc, const char * argv[]) {
    //1. 変数の宣言
    int i, x, y; //アクセス用の変数

    int hist[COLOR_MAX]; //ヒストグラム用配列

    //2. 画像をグレースケールで入力
    cv::Mat src_img = cv::imread(FILE_NAME, 0);
    if (src_img.empty()) { //入力失敗の場合
        fprintf(stderr, "File is not opened.\n");
        return (-1);
    }
}
```

入力失敗時
への対応

ヒストグラムの初期化

//3. ヒストグラム用配列の初期化

```
for (i=0; i<COLOR_MAX; i++) {  
    hist[i] = 0;  
}
```

//4. ヒストグラムの生成

```
for (y=0; y<src_img.rows; y++) {  
    for (x=0; x<src_img.cols; x++) {  
        //画素値の取得  
        unsigned char s = src_img.at<unsigned char>(y, x);  
  
        hist[(int)s]++; //画素数の計算  
    }  
}
```

//5. 出力 (Numbers or Excel用に印字)

```
for (i=0; i<COLOR_MAX; i++) {  
    printf("%d\n", hist[i]);  
}
```

アクセスされ
なかった場合
不定値

度数の計算

//3. ヒストグラム用配列の初期化

```
for (i=0; i<COLOR_MAX; i++) {  
    hist[i] = 0;  
}
```

//4. ヒストグラムの生成

```
for (y=0; y<src_img.rows; y++) {  
    for (x=0; x<src_img.cols; x++) {  
        //画素値の取得  
        unsigned char s = src_img.at<unsigned char>(y, x);  
  
        hist[(int)s]++; //画素数の計算  
    }  
}
```

//5. 出力 (Numbers or Excel用に印字)

```
for (i=0; i<COLOR_MAX; i++) {  
    printf("%d\n", hist[i]);  
}
```

画像を走査
画素値(s)読取り
輝度値配列に加算

ヒストグラムの出力方法

//3. ヒストグラム用配列の初期化

```
for (i=0; i<COLOR_MAX; i++) {  
    hist[i] = 0;  
}
```

//4. ヒストグラムの生成

```
for (y=0; y<src_img.rows; y++) {  
    for (x=0; x<src_img.cols; x++) {  
        //画素値の取得  
        unsigned char s = src_img.at<unsigned char>(y, x);  
  
        hist[(int)s]++; //画素数の計算  
    }  
}
```

//5. 出力 (Numbers or Excel用に印字)

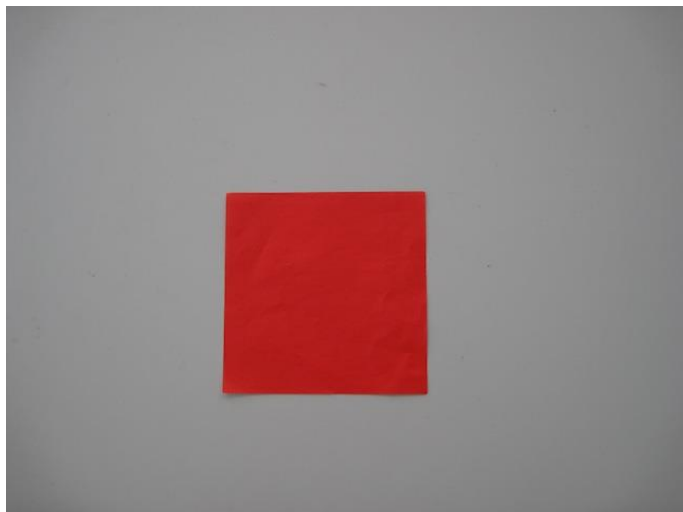
```
for (i=0; i<COLOR_MAX; i++) {  
    printf("%d\n", hist[i]);  
}
```

配列をprintfで出力し、
Numbers(Excel)でグラフ化

演習1 ヒストグラムの作成

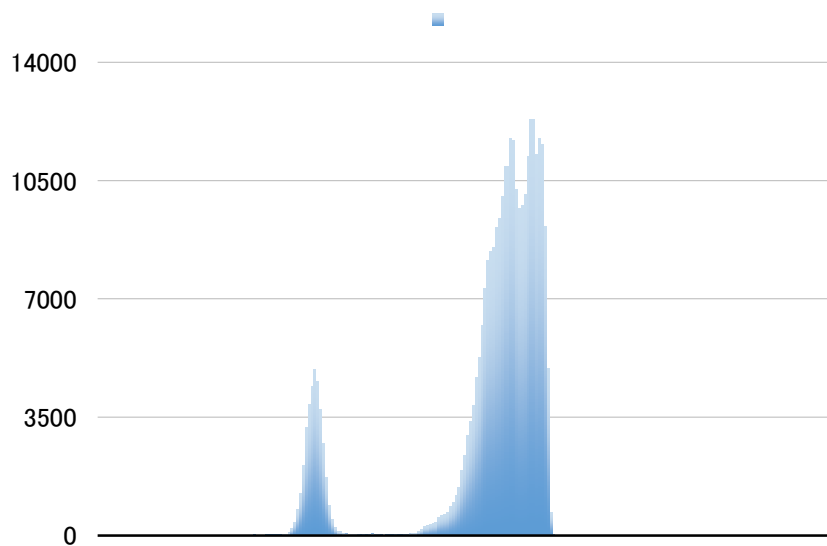
◆ プロジェクト名 : hist

課題：画像を読み込んで、ヒストグラムを作成する



red_rectangle.jpg

入力画像



ヒストグラム

ヒストグラム用配列の作成

```
#include <iostream>
#include <opencv2/opencv.hpp>

#define COLOR_MAX (256)

#define FILE_NAME "img/red_rectangle.jpg"

int main(int argc, const char* argv[]) {
    //1. 変数の宣言
    int i, x, y; //アクセス用の変数

    int hist[COLOR_MAX]; //ヒストグラム用配列
```

輝度値範囲
0~255
(横軸範囲)

ヒストグラム用配列の作成

```
#include <iostream>
#include <opencv2/opencv.hpp>

#define COLOR_MAX (256)

#define FILE_NAME "img/red_rectangle.jpg"

int main(int argc, const char * argv[]) {
    //1. 変数の宣言
    int i, x, y; //アクセス用の変数

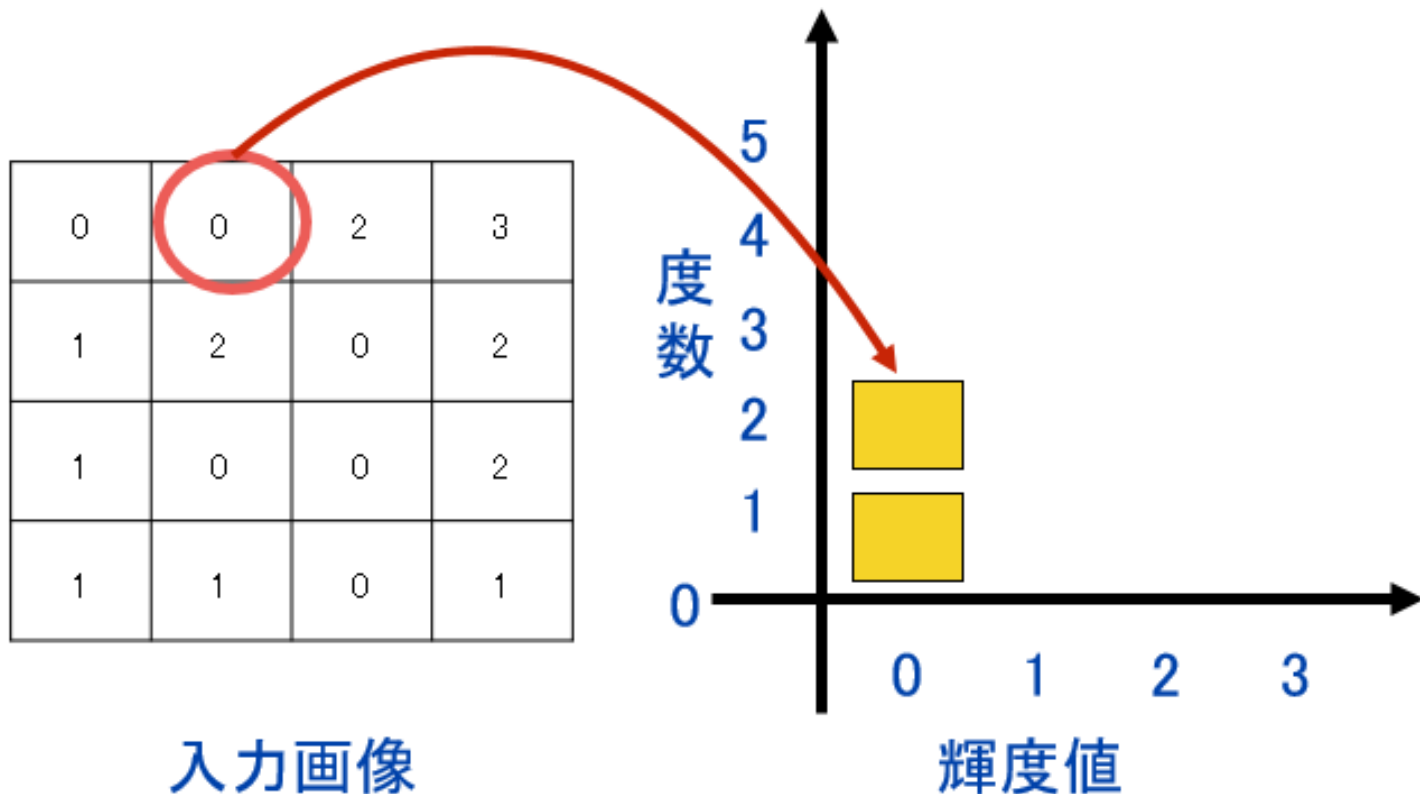
    int hist[COLOR_MAX]; //ヒストグラム用配列

    //2. 画像をグレースケールで入力
    cv::Mat src_img = cv::imread(FILE_NAME, 0);
    if (src_img.empty()) { //入力失敗の場合
        fprintf(stderr, "File is not opened.\n");
        return (-1);
    }
}
```

輝度値範囲
(横軸範囲)

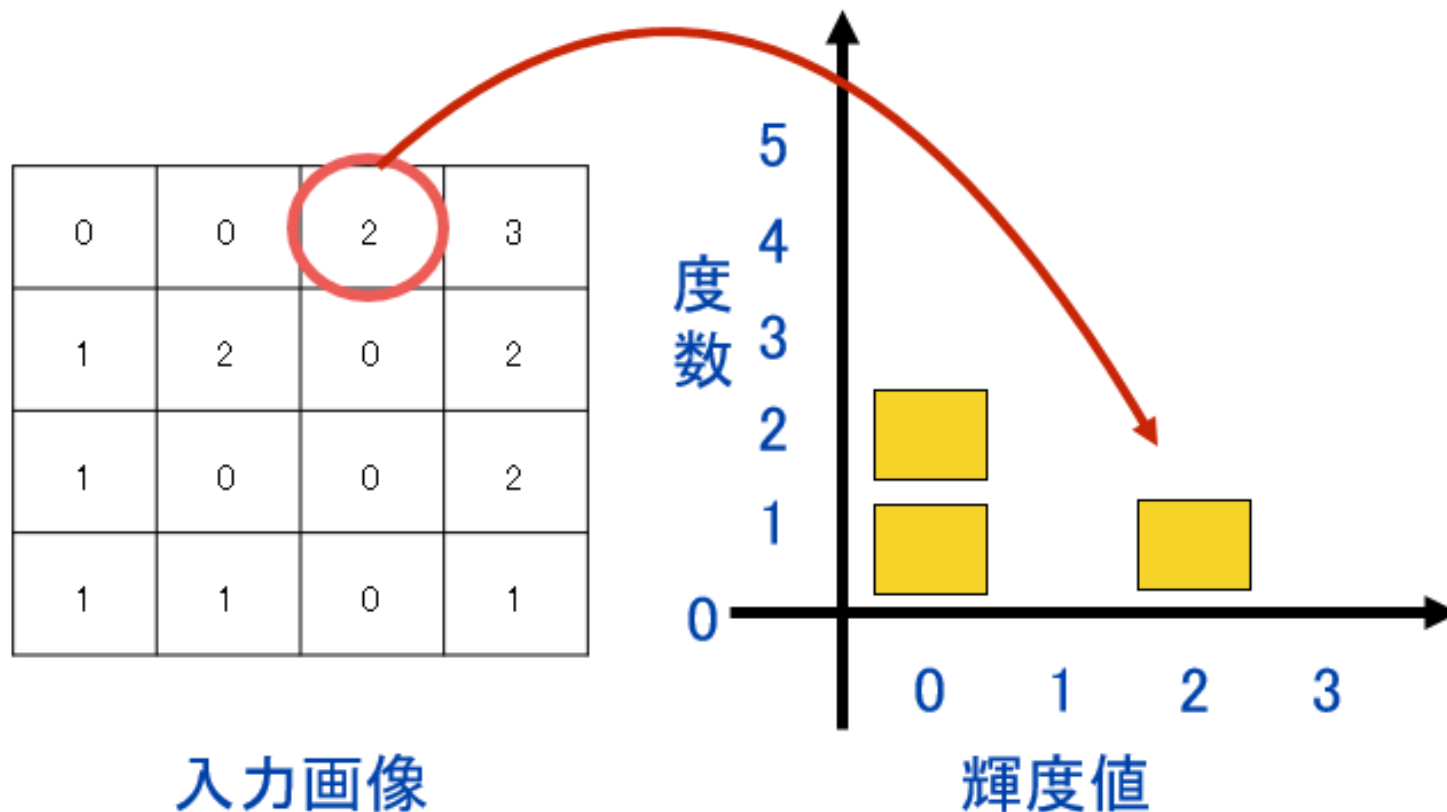
ヒストグラムの作成

◆ 画像の輝度値の度数グラフ



ヒストグラムの作成

◆ 画像の輝度値の度数グラフ

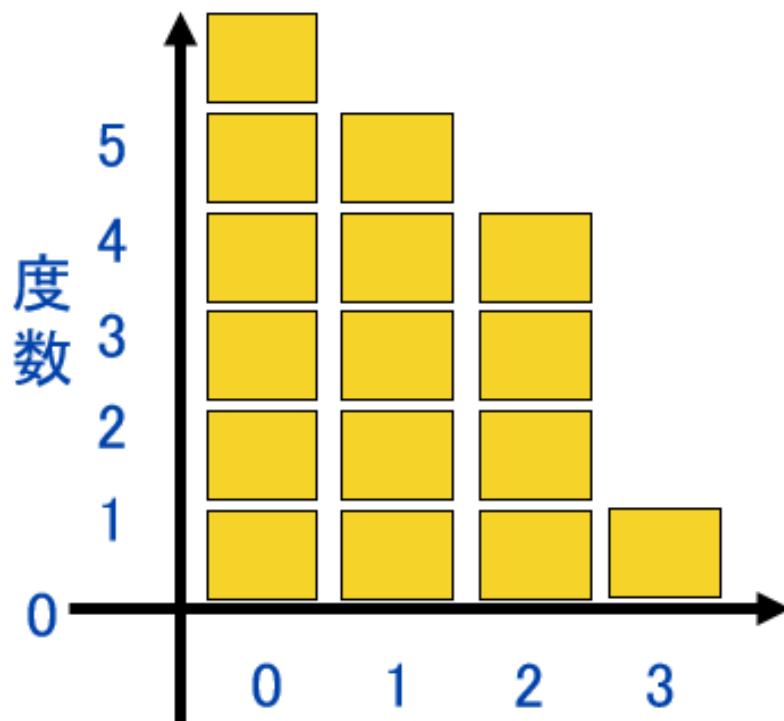


ヒストグラムの作成

◆ 画像の輝度値の度数グラフ

0	0	2	3
1	2	0	2
1	0	0	2
1	1	0	1

入力画像

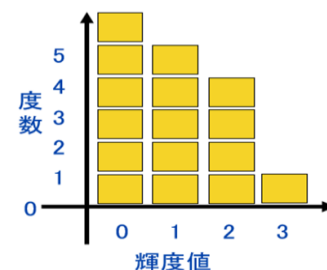
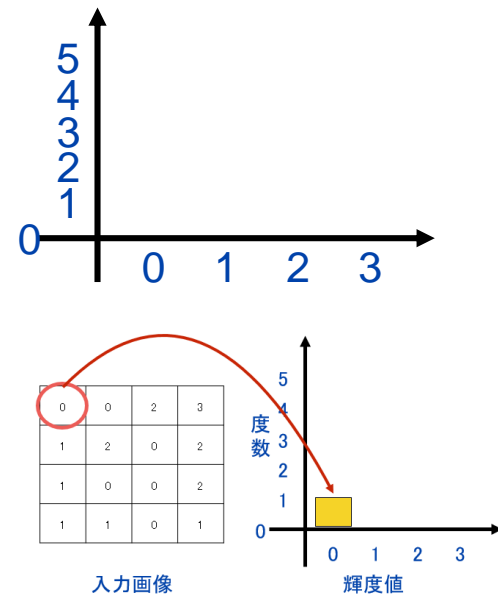


輝度値

ヒストグラムの作成

◆ 作成手順の確認

- ① ヒストグラム準備
横/縦軸範囲
- ② 画像走査
画素値読取り
画素値プロット
- ③ 表示



この次は手作業をプログラムで実現