画素単位の処理(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2　コントラストの改善

◆ プロジェクト名 : contrast

課題 : 入力画像のコントラストを改善する

入力画像

出力画像

演習2　コントラストの改善

◆ 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 初期値の設定（Imin,Imax）
③ 画像（グレースケール）の読込み
④ 出力画像の領域確保
⑤ 入力画像輝度値の最小値、最大値の取得
⑥ 線形変換（入力画像→出力画像）
⑦ 表示

コントラストの改善　1/4

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

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

//ウィンドウ名
#define WINDOW_NAME_INPUT "input"
#define WINDOW_NAME_OUTPUT "output"

int main(int argc, const char * argv[]) {
    //1. 変数の宣言
    int x, y;

    //変換前の画素値の最小値・最大値，変換後の画素値の最小値・最大値
    unsigned char Imin, Imax, Dmin, Dmax;

    //2. 初期値の入力 (Imin, Imax)
    Imax = 0;
    Imin = 255;

    //変換後の画素値の入力
    Dmin = 0;
    Dmax = 255;
```

コントラストの改善　2/4

```
//3. 画像（グレースケール）の読み込み
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);
}

//4. 出力画像のメモリ確保（グレー）
cv::Mat dst_img = cv::Mat(src_img.size(), CV_8UC1);
```

コントラストの改善　3/4

```
//5. 最小値，最大値の取得
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);
        //最小値の取得
        if (Imin > s) {
            Imin = s;
        }
        //最大値取得
        if (Imax < s) {
            Imax = s;
        }
    }
}
fprintf(stderr, "Imin=\n, Imax=\n", Imin, Imax);
```

コントラストの改善　4/4

```
    //6. 線型変換 (入力画像, 出力画像)
    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);

            s = (Dmax - Dmin) / (Imax - Imin) * (s - Imin) + Dmin;

            dst_img.at<unsigned char>(y, x) = s;
        }
    }

    //7. 表示
    cv::imshow(WINDOW_NAME_INPUT, src_img);//画像の表示
    cv::imshow(WINDOW_NAME_OUTPUT, dst_img);//画像の表示
    cv::waitKey(); //キー入力待ち (止める)

    return 0;
}
```

演習2　コントラストの改善

◆ 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 初期値の設定（Imin,Imax）
③ 画像（グレースケール）の読込み
④ 出力画像の領域確保
⑤ 入力画像輝度値の最小値、最大値の取得
⑥ 線形変換（入力画像→出力画像）
⑦ 表示

コントラストの改善　1/4

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

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

//ウィンドウ名
#define WINDOW_NAME_INPUT "input"
#define WINDOW_NAME_OUTPUT "output"

int main(int argc, const char * argv[]) {
    //1. 変数の宣言
    int x, y;

    //変換前の画素値の最小値・最大値，変換後の画素値の最小値・最大値
    unsigned char Imin, Imax, Dmin, Dmax;

    //2. 初期値の入力 (Imin, Imax)
    Imax = 0;
    Imin = 255;

    //変換後の画素値の入力
    Dmin = 0;
    Dmax = 255;
```

コントラストの改善　3/4

```
//5. 最小値，最大値の取得
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);
        //最小値の取得
        if (Imin > s) {
            Imin = s;
        }
        //最大値取得
        if (Imax < s) {
            Imax = s;
        }
    }
}
fprintf(stderr, "Imin=\n, Imax=\n", Imin, Imax);
```

コントラストの改善　4/4

```
    //6. 線型変換（入力画像，出力画像）
    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);

            s = (Dmax - Dmin) / (Imax - Imin) * (s - Imin) + Dmin;

            dst_img.at<unsigned char>(y, x) = s;
        }
    }

    //7. 表示
    cv::imshow(WINDOW_NAME_INPUT, src_img);//画像の表示
    cv::imshow(WINDOW_NAME_OUTPUT, dst_img);//画像の表示
    cv::waitKey(); //キー入力待ち（止める）

    return 0;
}
```