第3章 画像処理プログラムの基本構成

吉田 大海

本章では，画像のグレー化処理プログラムを例として，画像処理プログラムの内容について説明します．画像処理プログラムの全体構造とグレー化処理のプログラムを**図1**，**リスト1**にそれぞれ示します．以降では，このプログラムと照らし合わせながら読んでいただけると，理解の助けになると思います．また，自分自身で作成した画像処理を試したいときも，最初のうちは筆者のプログラムに上書きしていく形が簡単なのでお勧めです．

メイン関数の中身

● 入力画像の読み込み

リスト1の9行目で入力画像を読み込んでいます．この1行で入力画像の読む込みができます．この文の意味をもう少し説明すると，画像フォルダに入った入力画像 .bmpを，cv::Mat型で定義した変数Imageに格納するとなります．cv::Mat型は，OpenCVでディジタル画像を扱う代表的な型の1つです．

● 画像処理をする

リスト1の11行目で画像処理（ここではグレー化）をしています．この文は，Image内に格納されたディジタル画像をグレー化処理として定義された関数CHで画像処理するという意味です．どのように定義しているのかは後述するので，今は定義をしたら簡単な画像処理は1行で済むという程度に思っていてください．

● 処理結果を出力画像として保存する

メイン関数内の最後（**リスト1**の14行目）で画像処理結果を保存しています．この文の意味をもう少し説明すると，Imageに格納されたディジタル画像を画像フォルダに出力画像 .bmpと名付けて保存するということになります．このとき，画像の名前だけでなく拡張子（.bmp）も含めて記載することを忘れないでおきましょう．

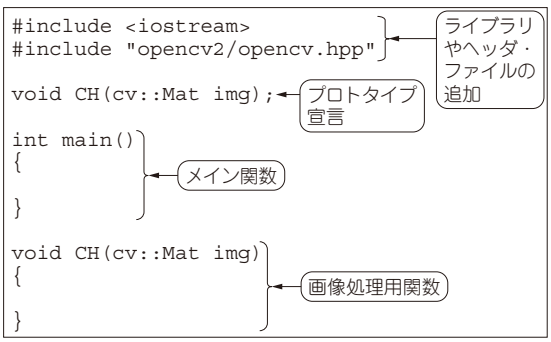

図1　画像処理プログラムの全体構造

リスト1　画像をグレー化処理するプログラムを例に基本構成を説明する

```
 1 #include <iostream>
 2 #include "opencv2/opencv.hpp"
 3
 4 void CH(cv::Mat img);//****< グレー化処理 >****
 5
 6 int main()
 7 {
 8 //入力画像としてImageを読み込みます
 9 cv::Mat Image = cv::imread("画像/入力画像.bmp", -1);
10
11  CH(Image);//画像をグレー化します
12
13  //出力画像としてImageを保存します
14  cv::imwrite("画像/出力画像.bmp", Image);
15  }
16
17  //****< グレー化処理 >****
18  void CH(cv::Mat img)
19  {
20  int x, y;
21  //画素値のx座標，y座標
22
23  int X = img.cols, Y = img.rows;
24  //画像の横画素数X，画像の縦画素数Y
25
26  int B, G, R, P;
27  //カラーチャネルBGR，計算結果P
28
29   for (y = 0; y < Y; y++) {
30     for (x = 0; x < X; x++) {
31
32     //----< 入力画像から画素値を読み込む >----
33     B = img.at<cv::Vec3b>(y, x)[0];
34     G = img.at<cv::Vec3b>(y, x)[1];
35     R = img.at<cv::Vec3b>(y, x)[2];
36
37     P = cvRound(0.114 * B + 0.587 * G +
                          0.299 * R);//BGRから輝度値を計算
38
39     //----< 出力画像へPを画素値として書き込む >----
40     img.at<cv::Vec3b>(y, x)[0] = P;
41     img.at<cv::Vec3b>(y, x)[1] = P;
42     img.at<cv::Vec3b>(y, x)[2] = P;
43    }
44   }
45  }
46  //****< グレー化処理 >****
```


プログラムは本誌ウェブ・ページから入手できます．
https://www.cqpub.co.jp/interface/download/contents.htm