

OenCVの使い方 (第1回)

氏名 山村武史
クラス 学際科学科
学生証番号 08-142025

1.3節 例1

○プログラムリスト

```
1 #include <iostream>
2 #include <opencv2/core/core.hpp>
3 #include <opencv2/imgproc/imgproc.hpp>
4 #include <opencv2/highgui/highgui.hpp>
5
6 using namespace cv;
7 using namespace std;
8
9
10 int main(int argc, char* argv[]){
11     if (argc <= 1){
12         cerr << "usage: grayscale <file>" << endl;
13         return -1;
14     }
15     Mat image = imread(argv[1]);
16     if (!image.data){
17         cerr << "no such file: " << argv[1] << endl;
18         return -1;
19     }
20     Mat grayImage;
21     cvtColor(image, grayImage, CV_BGR2GRAY);
22     const string fileName = "gray.jpg";
23     imwrite(fileName, grayImage);
24
25     string windowName = "Original";
26     namedWindow(windowName, CV_WINDOW_AUTOSIZE);
27     imshow(windowName, image);
28     windowName = "Gray Image";
29     namedWindow(windowName, CV_WINDOW_AUTOSIZE);
30     imshow(windowName, grayImage);
31     waitKey(0);
32     image.release();
33     grayImage.release();
34 }
```

```
35 return 0;  
36 }
```

○実行結果

gray画像



○考察

opencvの基本的な使い方として画像を読み込み色を灰色に変化させ、gray.jpgという名前でファイルを保存した。

プログラムを順に追っていくと、まず最初にコマンド引数として画像ファイルの名前を送り、それを15行目で読み込んでいる。

MatとはMatrix行列型の変数であることを示し、画像ファイルを行列として保持していることがわかる。

11~13行目で画像ファイルが正しく読み込めていない場合は!image.dataがtrueになりエラー終了させている

色の変換は21行目のcvtColor関数でおこなっており第1引数の画像データを第三引数の色で変換し、第二引数に保存している。

26、27行目でウィンドウの生成及び画像の表示を行っている。26行目のCV_WINDOW_AUTOSIZEはウィンドウの幅が画像ファイルに合わせた大きさになることを示している。

1.4節 例1

○プログラムリスト

```
1 #include <iostream>  
2 #include <opencv2/core/core.hpp>  
3 #include <opencv2/imgproc/imgproc.hpp>
```

```
4 #include <opencv2/highgui/highgui.hpp>
5
6 using namespace cv;
7 using namespace std;
8
9 int main(int argc, char* argv[]){
10  if (argc <= 1){
11    cerr << "usage: imageInfo <file>" << endl;
12    return -1;
13  }
14  Mat image = imread(argv[1]);
15  if (!image.data){
16    cerr << "no such file: " << argv[1] << endl;
17    return -1;
18  }
19  if (image.isContinuous())
20    cout << "isContinuous() = true" << endl;
21  else
22    cout << "isContinuous() = false" << endl;
23  cout << "size of an element = " << image.elemSize() << " Bytes" << endl;
24  cout << "size of a channel = " << image.elemSize1() << " Bytes" << endl;
25  cout << "type = " << hex << image.type() << dec << endl;
26  cout << "depth = " << hex << image.depth() << dec << endl;
27  switch (image.depth()){
28    case CV_8U: cout << "CV_8U" << endl; break;
29    case CV_8S: cout << "CV_8S" << endl; break;
30    case CV_16U: cout << "CV_16U" << endl; break;
31    case CV_16S: cout << "CV_16S" << endl; break;
32    case CV_32S: cout << "CV_32S" << endl; break;
33    case CV_32F: cout << "CV_32F" << endl; break;
34    case CV_64F: cout << "CV_64F" << endl; break;
35    default: cout << " something else" << endl;
36  }
37  cout << "size of a step = " << image.step1() << endl;
38  cout << "no. of channels = " << image.channels() << endl;
39  cout << "no. of rows = " << image.rows << endl;
40  cout << "no. of columns = " << image.cols << endl;
41  namedWindow("Image", CV_WINDOW_AUTOSIZE);
42  imshow("Image", image);
43  waitKey(0);
44  image.release();
45
46  return 0;
47 }
```

○実行結果

isContinuous() = true
size of an element = 3 Bytes

size of a channel = 1 Bytes
type = 10
depth = 0
CV_8U
size of a step = 2247
no. of channels = 3
no. of rows = 410
no. of columns = 749

○考察

画像データが具体的にどのようなバイト数であるかなどを補油時したのが今回のプログラムである。

今回はデータの連続性がギャップなしであることがわかる。また一画素のバイト数が3バイト、1つのチャンネルが1バイトであることから

3つのチャンネルで1ピクセルが構成されていることがわかる。今回の画像のチャンネルデータはCV_8U 8ビット符号なし整数のフォーマットであることが示されている。

今回はギャップがないため一行のバイト数2247が一画素のバイト数3と行数の積749と確かに一致する。

ちなみにギャップとはbyte数を都合よく管理するためのパディングのようなものである。

1.5節 例1

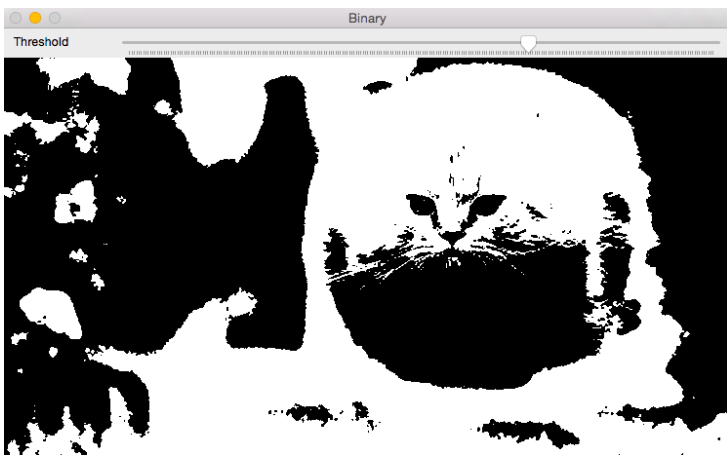
○プログラムリスト

```
1 #include <iostream>
2 #include <opencv2/core/core.hpp>
3 #include <opencv2/imgproc/imgproc.hpp>
4 #include <opencv2/highgui/highgui.hpp>
5
6 using namespace cv;
7 using namespace std;
8
9 #define TRACKBAR_MAX_VALUE 255
10 #define THRESHOLD_MAX_VALUE 255
11
12 const string sindowName = "Binary";
13 Mat grayImage;
14 void onChange(int value, void* data);
15
16 int main(int argc, char* argv[]){
```

```
17  if (argc <= 1){
18      cerr << "usage: binary <file>" << endl;
19      return -1;
20  }
21  Mat image = imread(argv[1]);
22  if (!image.data){
23      cerr << "no such file: " << argv[1] << endl;
24      return -1;
25  }
26  cvtColor(image, grayImage, CV_BGR2GRAY);
27  image.release();
28  int value = 128;
29  namedWindow(windowName, CV_WINDOW_AUTOSIZE);
30  const string trackbarName = "Threshold";
31  createTrackbar(trackbarName, windowName, &value, TRACKBAR_MAX_VALUE, onChange, NULL);
32  imshow(windowName, grayImage);
33  waitKey(0);
34  grayImage.release();
35  return 0;
36 }
37
38 void onChange(int value, void* data){
39     Mat binaryImage;
40     threshold(grayImage, binaryImage, value, THRESHOLD_MAX_VALUE, THRESHOLD_BINARY);
41     imshow(windowName, binaryImage);
42     binaryImage.release();
43 }
```

○実行結果

gray画像



○考察

トラックバーを動かすことで画像の明度を変化させるプログラムである。

38~43行目のコールバック関数がトラックバーを動かすたびに動作し、画像をある値より大きなものは全て黒、ある値より小さなものは全て白などのように画像ファイルをバイナリに直している。

またプログラムではトラックバーを動かすまではcallback関数が動かず、最初はグレースケールの画像。そこから少しでもバーを動かすと、バイナリ表現の画像となる。

せっくなので、はじめからバイナリ表現で画像が表示されるように変更したプログラムが以下のプログラムリストである。

○プログラムリスト(改訂版)

```
1 #include <iostream>
2 #include <opencv2/core/core.hpp>
3 #include <opencv2/imgproc/imgproc.hpp>
4 #include <opencv2/highgui/highgui.hpp>
5
6 using namespace cv;
7 using namespace std;
8
9 #define TRACKBAR_MAX_VALUE 255
10 #define THRESHOLD_MAX_VALUE 255
11
12 const string windowName = "Binary";
13 Mat grayImage;
14 void onChange(int value, void* data);
15
16 int main(int argc, char* argv[]){
17     if (argc <= 1){
18         cerr << "usage: binary <file>" << endl;
19         return -1;
20     }
21     Mat image = imread(argv[1]);
22     if (!image.data){
23         cerr << "no such file: " << argv[1] << endl;
24         return -1;
25     }
26     cvtColor(image, grayImage, CV_BGR2GRAY);
27     image.release();
28     int value = 128;
29     namedWindow(windowName, CV_WINDOW_AUTOSIZE);
30     threshold(grayImage, grayImage, value, THRESHOLD_MAX_VALUE, THRESH_BINARY);
31     const string trackbarName = "Threshold";
32     createTrackbar(trackbarName, windowName, &value, TRACKBAR_MAX_VALUE, onChange, NULL);
33     imshow(windowName, grayImage);
34     waitKey(0);
35     grayImage.release();
```

```
36 return 0;
37 }
38
39 void onChange(int value, void* data){
40     Mat binaryImage;
41     threshold(grayImage, binaryImage, value, THRESHOLD_MAX_VALUE, THRESH_BINARY);
42     imshow(windowName, binaryImage);
43     binaryImage.release();
44 }
```

□課題や授業に関して

○レポート作成に要した時間

2時間くらい

○特に苦勞した点

一番の難所は自分のパソコンにopencvの環境を構築する点であった。

g++ コマンドの引数 -Iや -lなどがなにをあらわしているか余り理解していなかったため自分の環境でどのようにコマンドをうてば、コンパイルできるか苦勞した。

最終的にg++ test.cc -o a.out `pkg-config --cflags --libs opencv`
ような形でコンパイルに成功したときは感動した。

○授業についての感想や希望

特に無し