

マルチメディア情報処理

レポート 4

学籍番号 : 165719E

氏 名 : 伊波卓浩

提出日 : 平成 31 年 1 月 16 日

マンガ風の画像生成

作成したコード

作成したコードを以下に示す。

```
1 import cv2
2 import numpy as np
3
4 def manga_filter(src, screen, th1, th2):
5
6     # グレースケール変換
7     gray = cv2.cvtColor(src, cv2.COLOR_BGR2GRAY)
8     screen = cv2.cvtColor(screen, cv2.COLOR_BGR2GRAY)
9
10    # スクリーントーン画像を入力画像と同じ大きさにリサイズ
11    screen = cv2.resize(screen, (gray.shape[1], gray.shape[0]))
12
13    # Canny アルゴリズムで輪郭検出し、色反転
14    edge = 255 - cv2.Canny(gray, 50, 230)
15
16    # 三値化
17    gray[gray <= th1] = 0
18    gray[gray >= th2] = 255
19    gray[ np.where((gray > th1) & (gray < th2)) ] = screen[ np.where((gray > th1)&(gray < th2)) ]
20
21    # 三値化画像と輪郭画像を合成
22    return cv2.bitwise_and(gray, edge)
23
24 # 入力画像とスクリーントーン画像を取得
25 img = cv2.imread("input.png")
26 screen = cv2.imread("screen.png")
27
28 # 画像の漫画化
29 manga = manga_filter(img, screen, 80, 215)
30
31 # 結果を出力
32 cv2.imwrite("output.png", manga)
```

実行結果

元となる画像を以下の図 1 に示す。



図1 元となる画像

マンガ風にした画像を以下の図2に示す。



図2 出力結果

解説

今回入力画像をマンガ風の画像に編集するにあたって、よりマンガっぽく見えるようスクリーントーンを選択や数値の値にこだわった。まず、入力画像とスクリーントーン画像をグレースケール変換し、スクリーントーン画像を入力画像の大きさのサイズを直す。次に、輪郭検出をし、エッジ画像の色を反転させる。そして、三値化を行い引数で受け取った値の間に当てはまる部分をスクリーントーン画像と入れ替える。最後に、三値化画像とエッジ画像を合成する。

29行目の `manga_filter` 関数の第三、第四引数に三値化する際の値の範囲を持たせた。スクリーントーン画像が多く見える方がマンガ風に見えると考え、最大値の方を高くすることで白色の部分が減り、よりマンガ風な画像ができたと感じる。