

[ご購入はこちら](#)

# 色やノイズ付加

土井 伸洋

## 2-1, 色変換

色変換とは、画像の色空間を変換する処理のことです。色空間(=色を表現するモデル)としては通常RGBを用いますが、それ以外にもHSV, Lab, YUVなど、さまざまな色空間が定義されています。これらを使うことで、RGBとは別の観点で画像を表現し、また、計測/編集しやすくします。それ以外にも、特定の観点、例えば輝度のみを用いて画像を表現することもあります。ここでは、代表的な色変換処理について紹介します。

### ● グレー・スケール変換

RGB 3チャンネルの画像をグレー1チャンネル(=256階調)の画像へと変換する処理です。エッジ検出の前処理としても使います。プログラムをリスト1に、実行結果を図1に示します。

#### リスト1 画像のグレー・スケール変換

```
# グレー・スケール変換 | cv2.cvtColor
gray_image = cv2.cvtColor(src_image,
                           cv2.COLOR_RGB2GRAY)
plt.imshow(gray_image, cmap="gray")
# グレー・スケール画像をそのまま表示するときは cmap="gray" とする
```



図1 リスト1(画像のグレー・スケール化)の実行結果

### ● 2値化

しきい値を元に入力画像を2階調(0/255)の画像へと変換します。入力画像には、グレー・スケール変換後の画像を使用します。2値化に使用するアルゴリズムは大きく分けて2つあります。1つ目は1つのしきい値を全体に適用する単純2値化(Simple Thresholding), もう1つは画像内の領域ごとにしきい値を変更する適応的2値化(Adaptive Thresholding)です。画像内に暗い部分と明るい部分が混在するような場合は、後者の方が良い結果を生み出せる可能性があります。

ここでは、陰影の差が大きいレシートの写真を入力画像とし、単純2値化/適応的2値化の両方で画像処理を行ってみます。プログラムをリスト2に、実行結果を図2に示します。

単純2値化では、白とび/黒とびにより商品名がかすれたり、価格が黒でつぶれてたりしています(しき

#### リスト2 画像の2値化

```
# 2値化比較用画像の読み込み
receipt_image = cv2.imread("receipt_sample.png")
receipt_image = cv2.cvtColor(receipt_image,
                              cv2.COLOR_BGR2RGB)
gray_image = cv2.cvtColor(receipt_image,
                           cv2.COLOR_RGB2GRAY)

# 2値化(Simple) | cv2.threshold
# thresh: しきい値
# maxval: しきい値を超えた値をこの値で置き換える
# type: しきい値の種類
# THRESH_BINARY: しきい値を超えたらmaxvalに置き換え
# THRESH_BINARY_INV: しきい値を超えなかったら
# maxvalに置き換え
_, simple_image = cv2.threshold(gray_image,
                                thresh=128, maxval=255, type=cv2.THRESH_BINARY)
# 2値化(Adaptive) | cv2.adaptiveThreshold
# maxValue: しきい値を超えた値をこの値で置き換える
# adaptiveMethod: 使用する2値化アルゴリズム
# thresholdType: しきい値の種類(同上)
# blockSize: しきい値算出時に参照するローカル領域のサイズ(px)
# C: しきい値判定を行う際に、ローカル領域の平均から引かれる値。
# 大きいほど全体が明るくなる
adaptive_image = cv2.adaptiveThreshold(gray_image,
                                       maxValue=255,
                                       adaptiveMethod=cv2.ADAPTIVE_THRESH_MEAN_C,
                                       thresholdType=cv2.THRESH_BINARY,
                                       blockSize=11,
                                       C=7)
```