

2-1 グレー・スケール化

プログラム名：Gray.py (GPU版はGray_gpu.py)
CPU版<https://interface.cqpub.co.jp/2-1gray-py/>, GPU版/2-1gray_gpu-py/



CPU版



GPU版

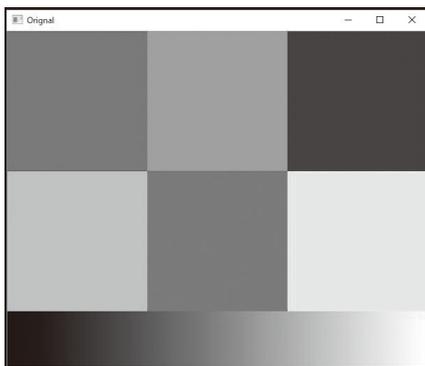


図1 グレー・スケール変換プログラムに使用するカラー画像

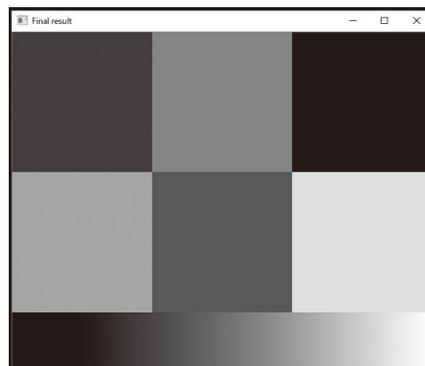


図2 グレー・スケール変換プログラムにより変換されたモノクロ画像

● 使用頻度の高いグレー・スケール変換

画像解析では、カラー画像の状態でも何らかの処理することは少なく、グレー・スケール化や2値化を行ってから解析をすることが多くあります。

● CPU版…プログラム名：Gray.py

グレー・スケール変換とは、図1の状態から図2の状態へ画像を変換させる処理です。各ビット・データを計算して変換することも可能ですが、カラー変換関数を使って処理します。

▶ リスト1：017行…グレー・スケールに変換する

```
img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
```

cvtColorには引数に応じてさまざまな色変換を行う機能があります。第1引数imgに処理前のカラー・データを、第2引数にはcv2.COLOR_BGR2GRAYを与えることで、MatデータのBGR画像からグレー・スケールに変換しています。必要な処理はこの1行だけで、図1から図2へのカラー変換が行われます。

リスト1 グレー・スケール変換プログラムCPU版(Gray.py)

```
000: import cv2
001: import numpy as np
002:
003: def __main__():
004:     img = createImage()
005:     cv2.imshow('Original', img)
006:
007:     # ここで処理を実行する
008:     img = getGray(img)
009:
010:     cv2.imshow('Final result', img)
011:     cv2.waitKey(0)
012:
013:     cv2.destroyAllWindows()
014:
015: def getGray(img):
016:     """CPUを使用"""
017:     img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
018:     # グレースケールに変換する
019:
020:     return img
021:
022: def createImage():
023:     width = 600
024:     height = 480
025:     size = (height, width, 3)
026:     # 縦480ピクセル 横640ピクセル 3チャンネル
027:
028:     img = np.zeros(size, dtype=np.uint8)
029:     rete = 255 / width
030:     for h in range(0, height):
031:         for w in range(0, width):
032:             data = int(w * rete)
033:             img[h, w] = [data, data, data]
034:
035:     red = np.array([0., 0., 255.])
036:     green = np.array([0., 255., 0.])
037:     blue = np.array([255., 0., 0.])
038:
039:
040:     cv2.rectangle(img=img, pt1=(0, 0),
041:                  pt2=(200, 200), color=red, thickness=-1)
042:     cv2.rectangle(img=img, pt1=(200, 0),
043:                  pt2=(400, 200), color=green, thickness=-1)
044:     cv2.rectangle(img=img, pt1=(400, 0),
045:                  pt2=(600, 200), color=blue, thickness=-1)
046:
047:     cv2.rectangle(img=img, pt1=(0, 200),
048:                  pt2=(200, 400), color=cyan, thickness=-1)
049:     cv2.rectangle(img=img, pt1=(200, 200),
050:                  pt2=(400, 400), color=magenta, thickness=-1)
051:     cv2.rectangle(img=img, pt1=(400, 200),
052:                  pt2=(600, 400), color=yellow, thickness=-1)
```