

色変換関数による変換

```
//グレースケール化 (カラー => グレー)
```

```
cv::cvtColor(src_img, dst_img, cv::COLOR_BGR2GRAY);
```

- 引数: 入力画像, 出力画像, コード
- コード (変換命令)
 - `cv::COLOR_BGR2GRAY`: BGR画像からグレースケール
(教科書p.28 : HSV変換の例 (後日))

ノート

OpenCVの色変換関数

```
//HSV変換 (RGB => HSV)
```

```
cv::cvtColor(src_img, dst_img, cv::COLOR_BGR2HSV);
```

- 引数: 入力画像, 出力画像, コード
- コード (変換命令)
 - **cv::COLOR_BGR2HSV**
 - ◎ HSV変換 RGB→HSV
 - **cv::COLOR_HSV2BGR**
 - ◎ 逆HSV変換 HSV→RGB

(教科書p.28)

ノート

HSV色空間処理の流れ

1. カラー画像 (BGR画像) の入力
2. 色変換 (cv::cvtColor関数で**BGR** → **HSV**)
3. 何らかの画像処理
4. 逆色変換 (**HSV** → **BGR**)
5. 表示

画像表示はBGR画像で行う必要がある

HSV色空間での画素アクセス例

```
cv::Vec3b p1, p2; //色値
```

```
//BGR -> HSV
```

```
cv::cvtColor(src_img, hsv_img, cv::COLOR_BGR2HSV);
```

```
for (y=0; y<hsv_img.rows; y++) {  
    for (x=0; x<hsv_img.cols; x++) {  
        p1 = hsv_img.at<cv::Vec3b>(y, x);  
        p2[0] = p1[0]; //H: 色相  
        p2[1] = p1[1]; //S: 彩度  
        p2[2] = p1[2]; //V: 明度  
        hsv_img.at<cv::Vec3b>(y, x) = p2;  
    }  
}
```

ここで何か処理

```
//HSV -> BGR
```

```
cv::cvtColor(hsv_img, dst_img, cv::COLOR_HSV2BGR);
```

ノート

OpenCVでのHSV値域

- 8bitで収まるように調整している **注意**
 - 色相 (H) : [0:360]度→[0:180] (半分になっている)
 - 彩度 (S) : [0:100]%→[0:255]
 - 明度 (V) : [0:100]%→[0:255]