メモ

OpenCVでのHSV値域

注意

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

(教科書p.27)

HSV色空間での画素アクセス例

```
cv::Vec3b p; //色値(HSV)

for (h=0; h<HUE_MAX; h++) { //Hの値を1つずつ変化させる（0→179）
    p[0] = h; //色相
    //S, Vの値は最大値にする
    p[1] = SAT_MAX; //彩度
    p[2] = VAL_MAX; //明度
    for (y=0; y<IMG_HEIGHT; y++) {  //色値をセット　縦方向は同じ
        hsv_img.at<cv::Vec3b>(y, h) = p;
    }
}
```

ここで処理（今回は代入）

ここで格納

復習 色変換関数による濃淡変換

```
//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

メモ

OpenCVの色変換関数

```
// 色変換 (HSV => BGR)
cv::cvtColor(src_img, dst_img, cv::COLOR_HSV2BGR);
```

- 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　BGR→HSV**
  - **cv::COLOR_HSV2BGR**
    - **◉ 逆HSV変換　HSV→BGR**

(教科書p.28)

参考：Mat演算による実現

```
// 参考：Mat演算による実現
hsv_img = hsv_img - cv::Scalar(0,0,VAL_DIFF);
```

- Mat構造体は行列演算を提供している
- 上記はhsv_img全体からスカラー値を減算
- 二重ループ不要

メモ

参考：しきい値処理関数

```
// しきい値関数による実施
cv::inRange(src_img, cv::Scalar(B_MIN,G_MIN,R_MIN),
 cv::Scalar(B_MAX,G_MAX,R_MAX),dst_img);
```

- 引数:

  入力画像，しきい値の下限，上限，出力画像

- 上記の設定による判定条件
  - (B_MIN,G_MIN,R_MIN)≦(B,G,R)≦(B_MAX,G_MAX,R_MAX)

※今回R_MAXは定義していないが255

- 二重ループ不要

参考：チャンネル分割関数

```
cv::Mat planes[3];
cv::split(hsv_img, planes);
cv::imshow("Hue", planes[0]);
cv::imshow("Saturation", planes[1]);
cv::imshow("Value", planes[2]);
```

- split（入力画像, 出力画像配列）
  - チャンネル毎の画像に分割する
- 反対はmerge