

## 数字認識について

- OpenCV というライブラリ（関数群）を使用している。
- あ

### □ 数字認識プログラムの構成

#### ○ ヘッダファイルの構成

num_threshold_value.h	img/	num_detect()	ターゲットの色の閾値の値の設定。
bmp.h	img/	数字認識全般	画像処理で統一できそうな値を宣言するためのヘッダファイル。 共通して使える表現などを記述しておくために用意した。
num_detect_extern.h	img/	num_detect()	数字認識プログラムのための extern 宣言集。
use_window.h	main/	num_detect()	画像を表示するかどうかを定義。
show_window.h	main/	number_test()	Window の名前の定義。use_window.h をインクルードしている。 <b>num_detect()または ball_detect()を使用するメイン関数ではインクルードする必要がある。</b>

#### ○ 認識シーケンス

カメラより画像の取り込み -> 画像をファイルに保存 -> 画像の2値化処理（白黒変換）、エッジ検出 -> 2 値処理した画像とエッジ検出処理をした画像を組み合わせ、指定した色の平面を検出する -> 検出した平面に対してラベリングを行う -> ラベリング領域内を2値化する -> その画像がターゲットかどうかを判別する -> ターゲットである場合、周辺分布法を使用して数字を識別する。

#### ○ ラベリングについて

この処理は単体で用いられることはまずなく、2 値化やエッジ検出における後処理として重要である。ラベリングとは画像上の画素の集合に番号をつけることで画素を分類する処理のこと。この処理を行うことによって画像上の画素を分類できるため、ラベリングされた領域ごとに区別して処理できる。

#### ○ その他備考

画像のピクセル数をVGAにすると、CPUの使用率が90%にも跳ね上がるらしい(ドキュメント参照)。QVGAでは60%ほどだが、使用率が非常に高い。

また、いくつもの処理を経由することで、認識に関する調整パラメータが多く、カメラを使用した認識では、背景画像がめまぐるしく変化することから、認識率を上げるにはとても苦勞すると思われる。

基本的なパラメータは

- ・色の閾値の変更(台紙の検出に関わる)
- ・FONT\_COLORの変更(紙に対しての文字の明るさ)
- ・エッジ検出時のパラメータの変更(平面の検出に関わるエッジ検出の厳しさ)
- ・可変閾値法のパラメータの変更(数字認識のパラメータ)

#### ○ このプログラムの制約

- ・カメラの視野中に複数のターゲットがないものとする。
- ・カメラの視野中にターゲットに良く似たものがないものとする。
- ・ターゲット全体が見えているものとする。
- ・ターゲットはカメラに対して正対し、回転していないものとする。
- ・ターゲットとカメラとの距離が1m程度であるものとする。

#### ○ 既知の問題

- ・複数のターゲットの同時認識は不可。
- ・今回用いた数字以外の形は認識不可。
- ・ターゲットの回転は考慮していない。
- ・背景の条件によっては認識できない場合もある。