

モーション検出プログラムのレポート

T11702M 飯塚正樹

➤ 使用したパッケージ

- cv_bridge
- roscpp
- std_msgs
- image_transport
- opencv_apps

➤ プログラムの流れ

1. 前処理

1. グレイスケール変換

オブティカルフローの処理を単純化するために、入力のカラ画像をグレイスケール画像に変換する。

2. 画像のリサイズ

本プログラムにて導入したオブティカルフローのアルゴリズム「TV_L1」は処理に時間がかかるため、画像を 1/10 倍してできるかぎり処理コストを減らす。

2. オブティカルフロー

1. オブティカルフローの算出

現フレームの画像 (GrayImg) と一つ前の画像 (PreImg) の画像を入力として、オブティカルフローを算出する。なお、カメラで撮影を始めた最初のフレーム時のように、一つ前の画像が存在しないときは現フレームの画像を 2 枚入力する。

2. オブティカルフローの強度の算出

オブティカルフローの強度 (magnitude) を算出し、これを正規化する。

3. 強度画像 (モーション領域) の獲得

オブティカルフローにおいて、強度が 0.5 以上を持つ画素のみを取り出し、これをモーション領域とみなして出力する。

4. 強度画像の補正

モーション領域においてオープニング処理, クロージング処理をおこなうことによって, 領域内の穴を埋める.

3. モーション領域を囲うバウンディングボックスの検出

1. ラベリングの実行

Labeling.h というライブラリを利用して, 隣接するモーション領域を統合したセグメントごとにラベリングをおこなう. また, 検出されたセグメントがあまりに小さいときは, そのセグメントをノイズと判断してラベリングの対象外とする処理も加えている.

2. 最大サイズのセグメントのバウンディングボックスの位置 (bb_pos) の取得

Labeling.h の機能を利用して, 最大サイズのセグメントのバウンディングボックスの位置を取得する. 具体的には, バウンディングボックスの左上, 右下の 2 点の座標を取得する.

3. 物体検出結果の表示

OpenCV の `cv::rectangle` という関数を利用して, 画像中のモーション領域を囲うバウンディングボックスを描画する.

➤ 参考資料

“[opencv] オプティカルフローを可視化する | たかみんつ”,

http://bicycle.life.coocan.jp/takamints/index.php/doc/opencv/doc/opticalflow_farneback

“ラベリングクラス | Imura Laboratory”, <http://imura-lab.org/products/labeling/>