

4駆動装置を利用した制御

作成者: 日下部

作成日: 2015/2/2

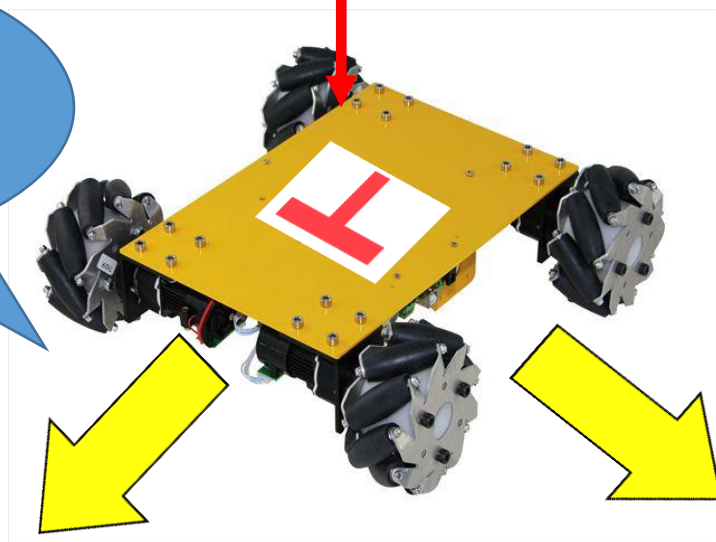
最終更新日: 2015/2/26

撮影用カメラを搭載して
人が近づけない場所の
撮影



画像処理で飛行体は
4駆動装置を追う

飛行体の代わりに
位置制御



＜ARDrone2.0+4駆動装置の制御方法＞

ARDrone2.0 を常に4駆動装置の上でホバリングさせる
⇒ 4駆動装置で水平方向の制御ができる

4駆動装置を使う利点

○水平方向の制御が容易

飛行体 : 常に6自由度を持つ制御

4駆動装置 : 平坦な道であれば3自由度
(X, Y+機体の向き)

○移動距離の計算が容易

飛行体 : GPSが利用できない場合、
移動経路の計算が困難

4駆動装置 : ロータリーエンコーダとコンパスで
移動経路が計算可能

利用する機器、ライブラリ



飛行体

機体: ARDrone 2.0 Elite edition

Linux 2.6.32搭載

通信で搭載しているセンサの値やカメラの映像を取得できる

4駆動装置の追跡

追跡カメラ: ARDrone搭載の対地即測定用QVGAカメラを利用

画像処理: OpenCV(画像処理用ライブラリ、C/C++/Python/Java対応)

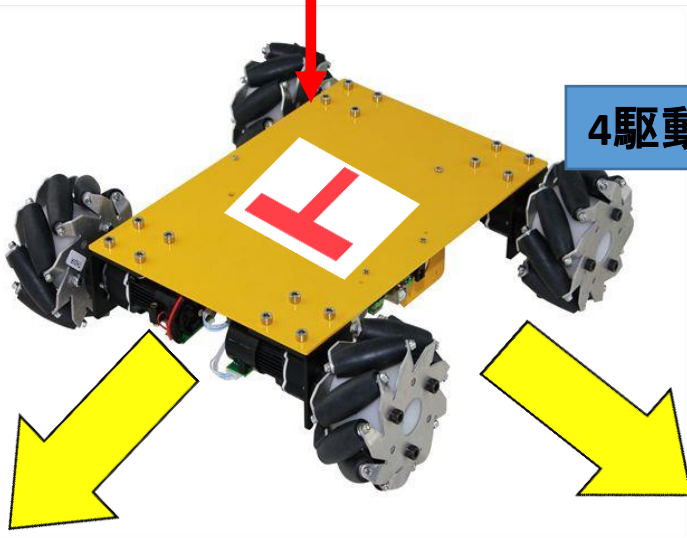
※ARDrone用に作られたCVDroneという非公式ライブラリを利用する構想

4駆動装置

4駆動装置: メカナムホイールロボット

メカナムホイールという形状のタイヤを利用し、各タイヤの回転方向を変えれば前後左右に移動が可能

通常の四輪駆動装置に比べてARDrone2.0に近い動きができる



OpenCVとは

正式名称: Open Source Computer Vision Library

WindowsやLinux,MacOSの他Android,iOSに対応した画像処理用公開ライブラリ(BSDライセンス)

当初、開発・公開はインテルが行っていたが現在ではWillow Garageが行っている

○実装されている機能

画像処理

勾配, エッジ, コーナー
サンプリング, 補間, 幾何変換
モルフォロジー演算
フィルタと色変換
ピラミッドとその応用
画像分割、領域結合、輪郭検出
画像と形状のモーメント
特殊な画像変換
ヒストグラム
マッチング

構造解析

輪郭処理
計算幾何
平面再分割

モーション解析と物体追跡

背景統計量の累積
モーションテンプレート
物体追跡
オプティカルフロー
推定器

パターン認識

物体検出

カメラキャリブレーションと3次元再構成

カメラキャリブレーション
姿勢推定
エピポーラ幾何

機械学習

単純ベイズ分類器
k近傍法
サポートベクターマシン
決定木
ブースティング
Random forest
EMアルゴリズム
ニューラルネットワーク

ユーザインタフェース

シンプルGUI
画像の読み込みと保存
ビデオ入出力

追跡処理の案

○テンプレートマッチング＋オプティカルフロー

テンプレートマッチング: あらかじめ用意したテンプレートとの一致度が高い場所を画像中から探し出す処理
オプティカルフロー : 2枚の画像を比較して、画像間で対応する場所を探し出す処理

- ・テンプレートマッチングで追跡対象がどこにあるかを推定
- ・オプティカルフローで追跡対象の移動量を推定

取得したオプティカルフローをテンプレートマッチングで絞込み → 移動方向と必要な速度の推定

・移動方向

対象の移動方向とカメラの中心からのずれを組み合わせて推測したい

・必要な速度

対象の速度に応じて高速移動と低速移動を切り替えたい

テンプレートマッチング

対象の画像とテンプレートを比較して、各地点の一致度(どれだけ一致しているかを示した値)を算出する
例)テンプレート画像の位置を画像中から探す

入力画像

1.対象となる画像



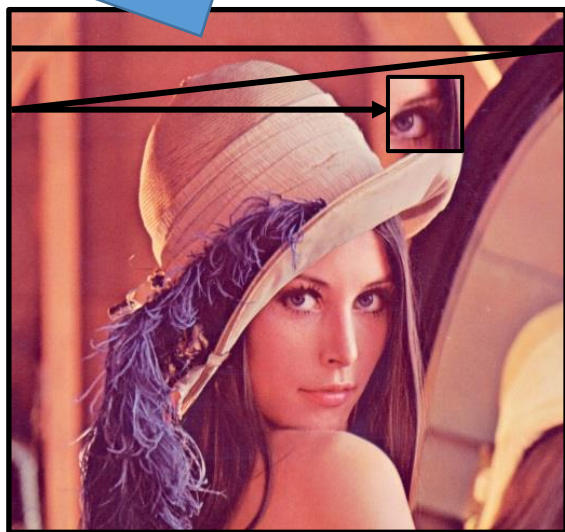
2.テンプレート画像



マッチング

テンプレート画像と対象画像を比較

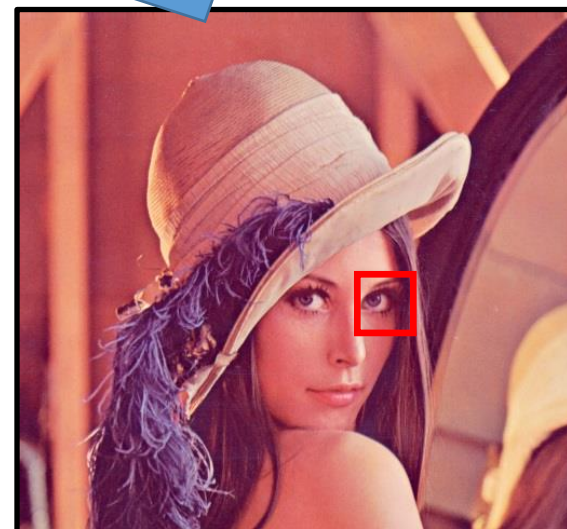
テンプレートの位置を少しずつずらしながら、各地点でどれほど一致するか調べる



出力画像

得た結果から処理を行う

各地点の値から最も一致している場所を赤枠で囲う



OpenCVにある一致度の算出方法：輝度差の二乗和、相互相関、相関係数

オプティカルフロー

画像を比較して対応する箇所を推定し、どれがどこに移動したかを算出するもの。

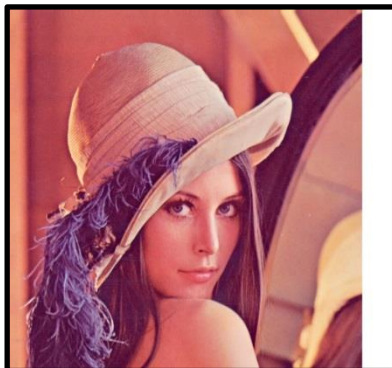
例) 特徴点をベースに元画像とそれを水平に移動させた画像のオプティカルフローの算出

入力画像

1. 移動前の画像

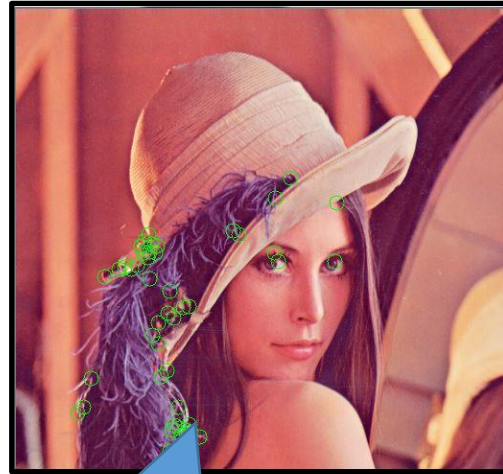


2. 移動後の画像



特徴点検出

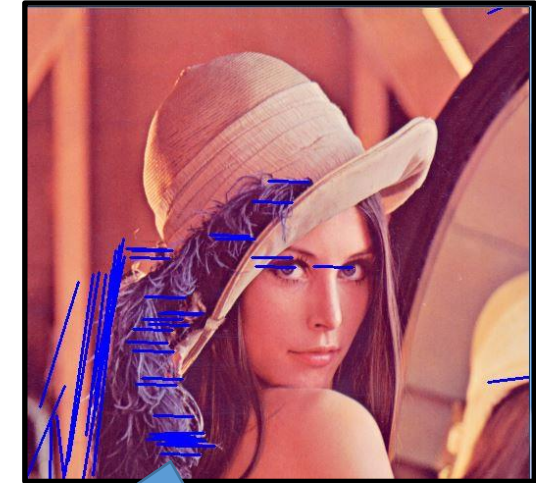
各画像の中から特徴点を検出



画像の中から角(=特徴点)と認識できる部分を検出(図中の緑の○の場所)

オプティカルフロー計算

それぞれ検出した特徴点で対応するものを調べる



移動前と移動後で対応する点を結ぶ

追跡処理の流れ

画像処理

1. 画像中の各特徴点がどこに移動したかを検出(=オプティカルフロー)
2. テンプレートマッチングで対象領域(=4駆動装置が写っている位置)を特定
3. 2.の結果からオプティカルフローを絞り込み
4. 絞り込んだ結果の平均から4駆動装置がどちらに動いたかを推測
5. 事前に用意している移動方法のうち、どれが最も近いかを判別

移動

6. 移動モードに移行(計算処理を停止させる)
7. 移動モード終了、ホバリング状態に移行
8. 1.に戻る