

# オブジェクト位置・姿勢推定アルゴリズム

作成日:2015/3/11

## 1. 概要

kinect 等で得られたポイントクラウドから, CVFH(Clusterd Viewpoint Feature Histogram)特徴量[1]をベースとした特徴量を用いたオブジェクトの位置と姿勢を推定する方法を説明する. ここで説明する方法は[2][3]を参考にした.

## 2. 方法

CVFH 特徴量はある視点から点群を見たときの特徴量であり, この特徴量が一致するものは視点が同じ, つまり姿勢が同一であることを示している. この性質を利用することによりオブジェクト位置姿勢を推定する. ただし CVFH 特徴量だけでは, カメラロール角が一意に定まらないので CRH(Camera Roll Histogram)特徴量を利用し最終的な位置姿勢を推定している.

### 2.1. フロー

オブジェクト位置・姿勢認識はオフラインで行う CVFH 特徴量データ作成と, オンラインでの位置・姿勢推定処理に分けられる.

#### [オフライン処理]

事前準備として, 把持対象オブジェクトの CAD モデルから特徴量等のデータセットを作成する. 特徴量は, CAD モデルをある視点から計測した点群(パーシャルビュー)を仮想的に生成し, その点群に対して算出する. 特徴量を複数の視点から計測した場合について算出し, それらの集合をデータセットとする.

#### [オンライン処理]

位置姿勢推定処理の流れを下図に示す.

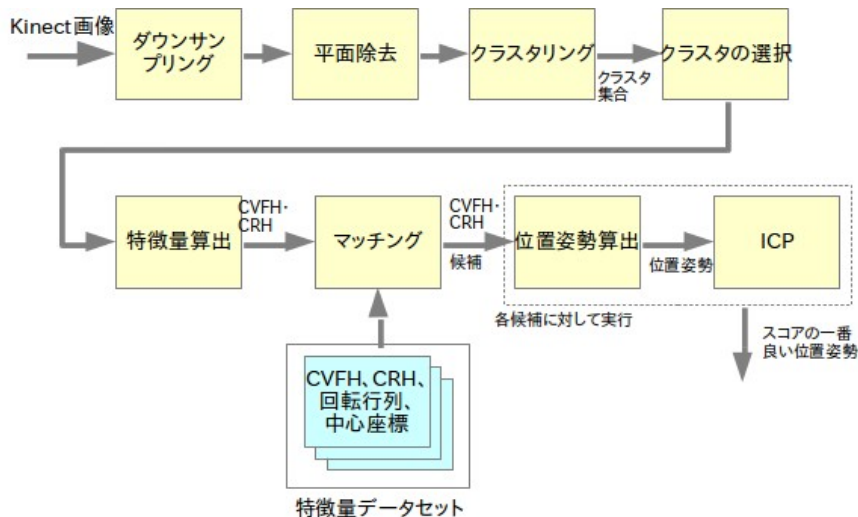


図 1: 処理の流れ(オンライン処理)

以下の手順で Kinect で得た点群から位置姿勢を推定する.

1. Kinect で得た点群をダウンサンプリングを行う.
2. 点群から平面を検出し除去する.
3. 点群に対し, セグメンテーションを行い, クラスタに分割する.
4. 得られたクラスタに対しマッチング候補となるクラスタを選択する. 各クラスタに対し 5.~8. を実行する.
5. クラスタの点群に対し特徴量を算出する.
6. 事前処理で作成した CVFH 特徴量データセットから 5. で算出した CVFH 特徴量と近いものを k 個探索

し候補姿勢とする。

7.対象候補に対して以下を実行する。

7-1.CVFH,CRH 特徴量からオブジェクトの位置姿勢を推定する。

7-2.7-1 で推定した位置姿勢を初期値として ICP(Iterative Closest Point)を実行する。

8.候補の中でスコアが良い位置姿勢を出力する。

## 2.2. 特徴量データセット作成

対象オブジェクトの CAD モデルデータから複数のパーシャルビューを作成し、各パーシャルビューに対し、以下のデータを作成する。実装では1つのモデルにつき 42 視点の特徴量を作成した。

- ・ダウンサンプリングしたパーシャルビュー(ポイントクラウド)
- ・CVFH 特徴量(変更 Ver.)
- ・CRH 特徴量
- ・CRH 特徴量(変更 Ver.)
- ・バウンディングボックスの大きさ
- ・姿勢行列
- ・中心座標

パーシャルビューの作成、姿勢行列、中心座標、CRH 特徴量の算出は PCL の関数を用いた。CVFH 特徴量(変更 Ver.)、CRH 特徴量(変更 Ver.)は PCL の関数をベースに後述する変更を適用したものをを用いた。

バウンディングボックスの大きさは以下のように算出した。

パーシャルビューの各点を xy 平面に投影し、xy 平面上でバウンディングボックスを求める。このバウンディングボックスの大きさと、元のパーシャルビューの z 方向の最小値と最大値の差をバウンディングボックスの大きさとする。

### 2.2.1. CVFH 特徴量(変更)

本方法では変更を加えた CVFH 特徴量を使用した。

[CVFH(オリジナル)]

オリジナルの CVFH 特徴量は以下のように求められる。

与えられた点群の中心座標を  $P_c$  , 平均法線ベクトルを  $n_c$  とし、

$$u_i = n_c$$

$$v_i = \frac{p_i - p_c}{\|p_i - p_c\|} \times u_i$$

$$w_i = u_i \times v_i$$

$$\cos(\alpha_i) = v_i \cdot n_i$$

$$\cos(\beta_i) = n_i \cdot \frac{p_c}{\|p_c\|}$$

$$\cos(\varphi_i) = u_i \cdot \frac{p_i - p_c}{\|p_i - p_c\|}$$

$$\theta_i = \text{atan2}(w_i \cdot n_i, u_i \cdot n_i)$$

$$SDC_i = \frac{(p_c - p_i)^2}{\max((p_c - p_i)^2)}$$

とする。すべての点  $p_i \in P$  に対し  $\alpha, \beta, \varphi, \theta, SDC$  を求めヒストグラムを作成し、連結したもの

$(\alpha, \varphi, \theta, SDC, \beta)$  が CVFH 特徴量である。ヒストグラムの各 bin 数は(45,45,45,45,128)である。

[変更点]

(1)SDC の計算方法の変更

SDC は中心座標からの距離であり、点群の中での最大距離で割り正規化を行っている。マッチングで複数種類のモデルを対象とした場合、モデルごとに最大距離が異なるため、点群の中での最大距離を分母にしていると考えられる。本方法はモデルが一種類のみの場合を想定しており、このことを利用すると SDC を下記のように変更することで、精度が向上すると考えられる。

$$SDC_i = \frac{(p_c - p_i)^2}{max\_distance}$$

ここで max\_distance は CAD モデルのバウンディングボックスの対角線長とする。

(2)点群分布ヒストグラムの追加

CVFH 特徴量では SDC が点群分布を表してしている。SDC は、中心点からの距離分布であるので、同一モデルの場合、少し回転した姿勢ともとの姿勢間での SDC の差はそれほどない。そこでカメラの視線方向の点群分布ヒストグラムを導入する。点群分布ヒストグラム ZDC を以下の式で計算する。

$$z_i = \frac{p_c}{\|p_c\|} \cdot p_i$$

$$ZDC_i = \frac{z_i - \min(z)}{\max(z) - \min(z)}$$

ここで ZDC の bin 数は 32 とした。

変更を加えた CVFH 特徴量は  $(\alpha, \varphi, \theta, SDC, \beta, ZDC)$  となり、合計 340 bin である。

### 2.2.2. CRH 特徴量(変更)

まず、CRH は以下のように算出される。

- 1)カメラの視線方向と Z 軸が重なるよう点群を回転させる。
- 2)各点の法線ベクトルを xy 平面へ投影し、xy 平面での角度を算出する。算出された角度に対応する bin のカウントをベクトルの大きさ分増やす。ここでヒストグラムは 4 度刻みの 90bin である。

この算出方法では、入力が平面に近い場合、ほとんどの点の法線ベクトルの向きがカメラ方向であるため、ヒストグラムは平坦なものになる。このため、平面の形状が長方形のとき、それが 90 度回転した場合のヒストグラムと元の状態のものは似たものになり、回転を認識することが困難である。

これを問題を解決するために、法線方向の代わりに、点の位置ベクトルの向きを使用した CRH 特徴量も利用する。位置ベクトル方向を使用した CRH 特徴量はオリジナルの CRH 特徴量と同様の手順で算出する。

マッチング時には、オリジナル CRH で推定したロール角に加え、位置ベクトル方向を使用した CRH で推定したロール角の両方を候補とする。

### 2.3. 位置・姿勢推定処理

位置・姿勢推定処理の各処理を説明する。

[ダウンサンプリング]

Kinect からの点群を指定した対象領域のみにし, さらにダウンサンプリングを行う。  
(使用 PCL 関数: PassThroughFilter, VoxelGrid)

[平面除去]

ダウンサンプリングした点群に対して, 平面除去を行う。  
(使用 PCL 関数: SACSegmentation)

[クラスタリング]

平面除去した点群に対し, セグメンテーションを実行し, 複数のクラスタに分割する。  
(使用 PCL 関数: EuclideanClusterExtraction)

[クラスタ選択]

クラスタリングで生成したクラスタに対し, マッチング候補とするクラスタを絞り込む。  
各クラスタに対しバウンディングボックスの大きさから優先度を計算し, 優先度の高いものから Cn 個のクラスタを候補とする。

優先度 priority はを利用して以下の式で計算する。

$$priority_i = 1 / \left( \left| 1 - \frac{edge\_long_i}{edge\_long_{model}} \right| + \left| 1 - \frac{edge\_mid_i}{edge\_mid_{model}} \right| \right)$$

ここで edge\_long はバウンディングボックスの一番長い辺の長さ, edge\_mid は 2 番めに長い辺の長さを表しており, 添字に model が付いているものは CAD モデルから算出したものである。

候補数 Cn は実行時に指定する。

[特徴量算出]

候補クラスタにそれぞれに対し, CVFH (変更), CRH, CRH (変更), 中心座標を算出する。

[マッチング]

特徴量算出で算出した CVFH 特徴量と, 特徴量データベースの CVFH 特徴量をマッチングを行う。  
k 近傍探索により特徴量算出で算出した CVFH 特徴量に近い k 個の CVFH 特徴量 (候補姿勢) を選択する。特徴量間の距離はカイ二乗距離を使用した。  
k は実行時に指定する。

[位置姿勢推定]

マッチングにより選択された CVFH 特徴量に対応する CRH, CRH (変更), 中心座標, 姿勢行列と特徴量算出で導出した特徴量により位置姿勢を算出する。

[ICP]

位置姿勢推定で算出した姿勢を初期解として, ICP を実行する。

[解の選択]

ICP で得られた解の中で一番スコアが小さいものを出力する。  
選択には以下のスコアを使用した。スコアは ICP でのスコアに outlier 率を加算したものである。

$$score = ICPscore + \alpha \cdot \frac{n_{outlier}}{n_{point}}$$

ここで ICPscore は対応点間の距離の 2 乗和,  $n_{point}$  は点群中の点の数  $n_{outlier}$  は対応点がない点の数,  $\alpha$  は係数である。

## 参考文献

- [1] A. Aldoma et al., "CAD-Model Recognition and 6DOF Pose Estimation Using 3D Cues," in Proc. ICCV 2011, pp.585-592.
- [2] Object Recognition with Global Features,  
<http://www.pointclouds.org/assets/iros2011/recognition.pdf>
- [3] Point Cloud Library Three-Dimensional Object Recognition and 6 DoF Pose Estimation,  
<http://robotics.usc.edu/~potthast/06299166.pdf>