

3次元回転不変位相限定相関法を用いた 3次元点群位置合わせ手法の高速化

Acceleration for method 3-D point cloud registration
using 3-D rotation invariant phase only correlation

河野 隆太[†]
Ryuta Kono

藤田 茂[‡]
Shigeru Fujita

1. はじめに

複数の点群を重ね合わせる処理は位置合わせと呼ばれており、3次元モデリングやSLAM(Simultaneous Localization and Mapping)[1]といった分野で利用される。

代表的な位置合わせ手法として、ICP(Iterative Closest Point) アルゴリズム [2] が挙げられる。ICP アルゴリズムは、点群間の各点の対応付けと、各対応点間距離の最小化を繰り返すことで位置合わせを行う。しかし、ICP アルゴリズムは、点群の点数が増加すると計算コストが膨大になる問題がある。

ICP アルゴリズムと比較して計算時間が短い手法として、位相限定相関法 (POC:Phase Only Correlation) と回転不変位相限定相関法 (RIPOC:Rotation Invariant Phase Only Correlation) を組み合わせた手法 [3] が提案されている。この手法では、まず点群を3次元配列へ格納する。次に、画像マッチング手法である POC と RIPOC を3次元へと拡張し、適用することで位置合わせを行う。POC と RIPOC を組み合わせた手法は、最適化計算や特徴点ベースの位置合わせ手法で用いられる3次元形状の評価を行う必要がないため、点群の点数による計算量の変化が少ない。ここで、環境の形状を表した点群を格納した配列は非常に疎な配列となる。また、POC と RIPOC ではフーリエ変換を多用する。そのため、疎な配列に対してフーリエ変換を適用することとなり無駄な計算が多くなるという課題がある。

そこで、本研究では点群が持つ座標情報を用いることで計算量を減らし、フーリエ変換処理の高速化を図る。

2. POC と RIPOC を組み合わせた手法

2.1 画像マッチング

画像 f, g が入力されたとき、以下の手順で画像の平行移動と回転のずれの量を求めることができる。

1. f, g に RIPOC を適用し回転量 θ を得る
2. g を θ だけ回転させたものを g' とする
3. f, g' に POC を適用し平行移動量 (x, y) を得る

2.2 点群の位置合わせ

点群を3次元配列へと格納し、3次元へ拡張した POC と RIPOC を画像マッチングと同じように適用することで、2つの点群の回転と平行移動についてのずれの量を求めることができる。

2.3 既存手法の課題点

以下に、3次元フーリエ変換の処理手順を述べ、無駄な計算となる部分を示す。

3次元フーリエ変換は、複数回の1次元フーリエ変換で表される。3次元配列に対して、行方向、列方向、奥行き方向の3方向について1次元フーリエ変換を行うことで、3次元フーリエ変換となる。サイズ N の配列 $Data$ が入力されたときの1次元フーリエ変換の処理手順を次に示す。また、1次元フーリエ変換の処理イメージを図1に示す。なお、ここではフーリエ変換を行う際に一般的に用いられる高速フーリエ変換 (Fast Fourier Trans) を対象としている。

1. $Data$ の中央から見て、添字の小さい方のデータを S 、大きい方のデータを R とする
2. サイズを $N/2$ とした配列 $Even, Odd$ を用意
3. $Even[i] \leftarrow S[i] + R[i]$, $Odd[i] \leftarrow (S[i] - R[i]) * W^i$ ($W^i = e^{-j2\pi i/N}$)
4. $Even, Odd$ のサイズが1になるまで、それぞれを入力配列 $Data$ として1. から3. の処理を再帰的に繰り返す
5. $Data[i * 2] \leftarrow Even[i]$, $Data[i * 2 + 1] \leftarrow Odd[i]$ となるようにデータを格納

このとき、1次元フーリエ変換の処理3. について、 $S[i]$, $R[i]$ のどちらにもデータが格納されておらず値が0のとき無駄な計算となる。

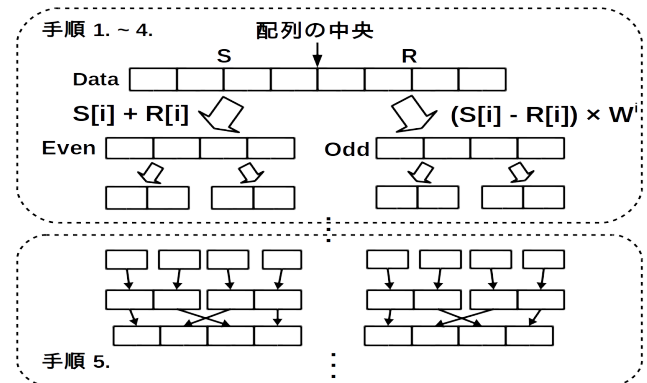


図 1: フーリエ変換の処理イメージ

また、Kinect を用いて取得した点群を図2に、点群を取得した環境を図3に示す。

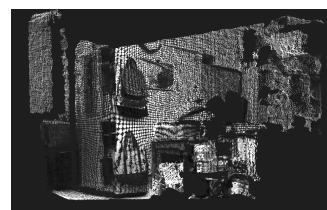


図 2: 取得点群

図 3: 点群取得環境

[†]千葉工業大学大学院 情報科学研究科 情報科学専攻
[‡]千葉工業大学 情報科学部 情報工学科

図 2 の点群を格納したデータサイズ 256^3 の 3 次元配列は、全体の要素数に対してデータが格納された要素数の割合が 0.25[%] と非常に疎な配列となった。点群は環境の形状の表面のみを表すため、点群を格納した配列は疎な配列となる。

よって、点群を格納した配列に対してフーリエ変換を行うと、疎な配列に対してフーリエ変換を行うこととなり、無駄な計算が多くなる。

3. 提案手法

そこで、提案手法では次に示す 2 つの処理を加えることで、無駄な計算を減らしフーリエ変換処理の高速化を図る。

3.1 点群を配列へ格納する際の処理

点群の各点は、点がある座標と点の色の 2 つの情報を持つ。ここで、座標情報を用いると、2.3 のフーリエ変換の処理 3. において、色の値に加えられる処理と *Even*, *Odd* のどこに格納されるかを求めることができる。よって、各点について座標情報から適当な処理を加えて適当な位置に格納することで、データが格納される要素についてのみ計算が行われ、無駄な計算を減らすことができる。以下に、処理手順を示す。

1. サイズを $N/2$ とした配列 *Even*, *Odd* を用意
2. 点群から点 (座標 x , 値 v) をひとつ選択
3. $x < N/2$ ならば,

$$\text{Even}[x] \leftarrow \text{Even}[x] + v,$$

$$\text{Odd}[x] \leftarrow \text{Odd}[x] + v * W^x$$
4. $x \geq N/2$ ならば,

$$\text{Even}[x - N/2] \leftarrow \text{Even}[x - N/2] + v,$$

$$\text{Odd}[x - N/2] \leftarrow \text{Odd}[x - N/2] - v * W^{x-N/2}$$
5. 全ての点に対して 2., から 4. を適用
6. *Even*, *Odd* を入力配列として、従来のフーリエ変換と同様の処理を行う

3.2 値が格納されている範囲の記憶

3.1 で示した処理 3., 4. のときに、値を格納した配列の添字 i について、($i < N/4$) の場合の最小値 i_{Smin} , 最大値 i_{Smax} と ($i \geq N/4$) の場合の最小値 i_{Rmin} と最大値 i_{Rmax} を記憶しておく。ここで、記憶した最小値と最大値は値が格納された最も端の位置を表している。求めた最小値と最大値を処理 5. を行う際に用いることで、計算を行う範囲を絞ることができ、無駄な計算を減らすことができる。図 4 にその様子を示す。

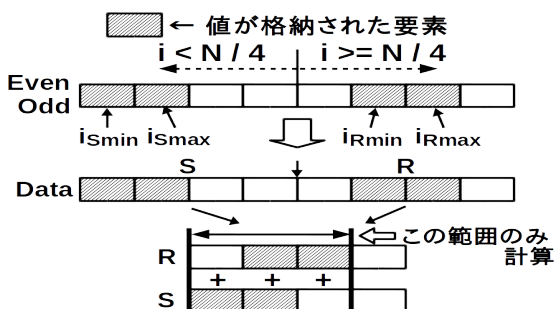


図 4: 計算を行う範囲を絞る

4. 評価実験

評価実験では、点群を配列へ格納しフーリエ変換をした場合の処理時間と、提案手法の処理時間とを比較する。ここでは、行方向のフーリエ変換のみ行う。3 次元配列サイズを 256^3 とし、配列要素数に対する点群の点数の割合を変化させたときの処理時間を図 5 に示す。

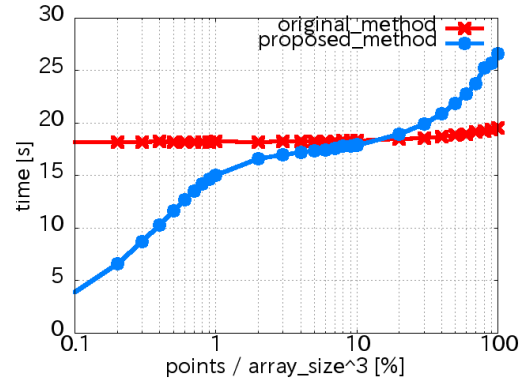


図 5: 測定結果

5. 考察

図 5 から提案手法は既存手法と比較して、点数の割合が 1[%] 以下のときに大きく高速化されていることが分かる。これは、点の密度が低くなるにつれデータが格納されていない行が増加し、処理 3.2 によって計算を行う範囲が大きく絞られたからだと考える。また、点数の割合が 10[%] を越えたあたりから提案手法の処理時間が既存手法の処理時間を上回っている。これは、提案手法の 1 点に対する処理時間が、既存手法の配列の 1 要素に対する処理時間を大きく上回っているためだと考える。

ここで、2 章で示したように点群を格納した配列は要素数に対する点数の割合が 1[%] 以下の疎な配列である。よって、POC と RIPOC を組み合わせた手法で位置合わせを行う際に、提案手法を用いることで処理の高速化を行うことができる。

6. おわりに

提案手法を用いることで、POC と RIPOC による位置合わせ手法の高速化が行えることを示した。今後は、実際に位置合わせ手法を実装し、全体の処理時間の比較を行う。

参考文献

- [1] Nicola Fioraio and Kurt Konolige. Realtime visual and point cloud slam. In *Proc. of the RGB-D workshop on advanced reasoning with depth cameras at robotics: Science and Systems Conf.(RSS)*, Vol. 27, 2011.
- [2] 増田健. ICP アルゴリズム. 研究報告コンピュータビジョンとイメージメディア (CVIM), Vol. 2009, No. 23, pp. 1–8, aug 2009.
- [3] 羽田遼, 寺田賢治. 3 次元回転不変位相限定相関法を用いた 3 次元モデリング. 電気学会論文誌 C (電子・情報・システム部門誌), Vol. 136, No. 8, pp. 1128–1134, 2016.