

キーポイントパッチ抽出法を用いた 三次元点群レジストレーションに関する諸検討

16T2806E 齊藤 陽平

2020 年 2 月 2 日

1 研究背景

三次元点群はコンピュータビジョンの位置分野であり、物体を複数の離散的な点で表現したものである。近年では三次元点群の取得できる安価なセンサの普及によりモデリングやロボティクス、測量などの分野で三次元点群の技術に対する需要が高くなっている。特に三次元点群処理は構成する点の数が莫大になり計算コストが高くなるため、高効率な手法が求められている。

実空間上の物体をレーザなどで測定し点群化するとき、物体の影や裏側にはレーザが届かず一回で完全に物体の全面を一つの点群として構成することは計測が不可能である。そのため完全なモデルを作成するには複数の位置から取得した点群を一つにまとめる必要がある。このような場合に複数の点群の相対関係を決定する問題をレジストレーションという。本稿ではレジストレーションにおいて重ねる基準となる点群をターゲット点群 $\mathcal{T} = \{q_1, q_2 \dots q_{N_t}\}$ ターゲット点群に重なる点群をソース点群 $\mathcal{S} = \{p_1, p_2 \dots p_{N_s}\}$ とする。

2 レジストレーション手法と先行研究

代表的なレジストレーションの手法には ICP、特徴点のマッチング、進化計算などが挙げられる。中でも進化計算を使ったレジストレーション (ECR) では精度や成功確率が高い一方で計算時間が長いというデメリットが存在する。進化計算レジストレーションでは式 (1) の評価関数を最小化する事で、最適な重なりとなる剛体変換 f を求める。ここで、剛体変換 f を構成するのに必要な要素は $(\theta_x, \theta_y, \theta_z, t_x, t_y, t_z)$ の 6 次元であり、進化計算の遺伝子もこの 6 次元ベクトルである。

$$FS = \frac{1}{N_s} \sum_{p_i \in \mathcal{S}} \|f(p_i) - \arg \min_{q_j \in \mathcal{T}} \|f(p_i) - q_j\|^2 \quad (1)$$

計算時間が長い理由は評価関数で用いられる最近傍探索の回数が個体数 \times 世代数 $\times N_s$ と非常に多くなってしまうためである。そこで植西ら [1] はキーポイントパッチ (KPP) 抽出法による ECR(KPP-ECR) を提案した。この手法ではソース点群をキーポイントとその周辺の点に限定することで、ECR を 100 倍程度の高速化に成功した。しかし、KPP-ECR では点群同士重ならない場所に KPP を抽出するとその KPP がターゲット点群に惹きつけられてしまうためその KPP の位置に成功可能かどうかは深く関係する。しかし、KPP が重なるかどうかは最適な剛体変換を求めるまで不明である。そこで重なる場所の KPP だけを使う工夫が必要になる。しかし、従来法では KPP の位置に対しての最適な配置に関する考察は未だなされていない。本研究ではこの KPP の位置を自動的に

補正する方法について初歩的な提案とその実験・考察を記す。

3 提案手法

本提案手法では予めキーポイントパッチを大量に抽出し、その中から使用するパッチを適宜選択することで、キーポイントパッチの位置を最適化の過程で変化させる。各世代で評価関数が変動するため、進化計算手法には CMA-ES[3] を用いる。次のようなステップで実行する。

- Step1. キーポイント周囲の点をパッチ集合 \mathcal{A} として抽出する
- Step2. CMA-ES のパラメータを設定する
- Step3. CMA-ES の平均を用いて、パッチを評価する
- Step4. パッチの評価値からパッチ集団を更新する
- Step5. パッチ集団の中から評価に使うパッチ \mathcal{U} を選択する
- Step6. CMA-ES で生成した個体を \mathcal{U} を使って評価する
- Step7. CMA-ES のパラメータを更新する
- Step8. 収束条件ならば終了、そうでなければ Step3. に戻る

Step3. パッチの評価では 1 世代に一度パッチ \mathcal{A}_i 毎に式 (2) によって評価値を求める。ここで cl は p_i の最近傍の点番号である。 $p_{(i,n)}$ は点 p_i の法線を示す。この評価関数では法線による値と距離による値の積で評価を行っている。理想的に重なる点群かどうかは形状の情報のみを利用することが望ましい。形状だけの情報で評価するとターゲット点群の穴の中のパッチが稀に残ってしまうので今回は距離情報も加味した。

$$FS_p(\mathcal{A}_i) = \sqrt{1 - \frac{\sum_{p_i \in \mathcal{A}_i} |p_{(i,n)} \cdot q_{(cl,n)}|}{\text{count}(\mathcal{A}_i)}} \sqrt{\frac{\sum_{p_i \in \mathcal{A}_i} \|p_i - q_{cl}\|^2}{\text{count}(\mathcal{A}_i)}} \quad (2)$$

Step4. パッチ集団の更新では世代 g でのパッチ集団 \mathcal{A}^g を式 (3,4) で更新していく。更新によって重ならないパッチを選択肢から除外する。ここで G は終了世代数、 $\bar{\mathcal{A}}^g, \sigma_{\mathcal{A}^g}$ はそれぞれ世代のパッチ評価値の平均、分散である。

$$\mathcal{A}^{g+1} = \begin{cases} \text{remove}(\mathcal{A}^g) & N_u < \text{count}(\text{remove}(\mathcal{A}^g)) \\ \mathcal{A}^g & \text{otherwise} \end{cases} \quad (3)$$

$$\text{remove}(\mathcal{A}^g) = \{\mathcal{A}_i^g \in \mathcal{A}^g \mid \bar{\mathcal{A}}^g + k\sigma_{\mathcal{A}^g} < FS_p(\mathcal{A}_i^g)\} \quad (4)$$

$$k_g = \alpha + \beta \frac{G - g}{G + g} \quad (5)$$

Step5. パッチ選択は評価関数に使用するためのパッチを N_u 個、選択肢から選択する。評価が良いパッチも悪いパッチも選ぶために、式 (6,7) を利用して評価値全体でまんべんなく評価を選択する。ここで $\mathcal{A}_i^{(g+1)'}$ は評価値でソートされたパッチ集団 $\mathcal{A}_i^{(g+1)}$ であり、 $L_g = \lfloor \frac{\text{count}(\mathcal{A}_i^{(g+1)})}{N_u} \rfloor$ である。このよう

に選択する事で、一部のパッチのみが収束してしまうことを防ぐ。

$$\mathcal{U}^{g+1} = \bigcup_{i \in I^g} \mathcal{A}_i^{(g+1)'} \quad (6)$$

$$I^g = \{L_g j + \text{rand}(0, L_g) | j \in \mathbb{Z} \cap [1, N_u]\} \quad (7)$$

4 実験

本手法と従来の KPP-ECR で実際にレジストレーションを行った。点群に Stanford レポジトリ [2] を使用し、実装には Point Cloud Library[4] を使用した。実験では時間短縮のため、各点群に各辺 2mm のボクセルサンプリングを行った。実験は各データセットで 30 回づつ行い、結果はその平均を表示する。結果を 2 に示す。ここで結果の単位はメッシュ解像度であり、成功はレジストレーション結果と真の変換を行った点群との平均距離がターゲット点群のメッシュ解像度以下、すなわち RMSE が 1 以下の時とした。

表 1. 実験に使用した点群

実験	ソース点群	ターゲット点群	オーバーラップ率
exp1	Bun315	Bun000	74.5 %
exp2	Bun315	Bun045	57.3 %
exp3	Armadillo150	Armadillo180	68.4 %
exp4	Armadillo150	Armadillo210	46.6 %
exp5	Dragon336	Dragon0	81.1 %
exp6	Dragon336	Dragon24	65.0 %
exp7	happyStand.336	happyStand.0	79.7 %
exp8	happyStand.336	happyStand.24	56.1 %

表 2. 提案手法と従来法のレジストレーション比較

実験	手法	成功率	RMSE	RMSE-s	RMSE-f
exp1	従来	0.87	2.66	0.87	14.34
	提案	0.8	2.29	0.56	9.23
exp2	従来	0	9.39	-	9.39
	提案	0.63	4.34	0.58	10.83
exp3	従来	0	2.05	-	2.05
	提案	0.93	1.59	0.25	20.31
exp4	従来	0	3.57	-	3.57
	提案	0.23	15.13	0.35	19.63
exp5	従来	1	0.418	0.42	-
	提案	1	0.312	0.31	-
exp6	従来	1	0.74	0.74	-
	提案	0.9	1.62	0.39	12.69
exp7	従来	1	0.17	0.17	-
	提案	0.77	1.22	0.23	4.50
exp8	従来	0	6.47	-	6.47
	提案	0	8.18	-	8.18

5 考察・結言

exp3 では、RMSE は小さいものの、従来法の成功率が 0 となった。これはキーポイントパッチを従来法をパラメータ通りに抽出した結果、一部オーバーラップしていない点が抽出されてしまったためである。パラメータの調整を行ってから再実験を行うと、成功率が 1.0 となった。この結果からも、従来法のパラメータ調整法による位置指定の難しさがわかる。exp2, exp4 ではオーバーラップ率が低いいため、従来法でのレジストレーション

が不可能であった実験である。オーバーラップ率が小さいレジストレーションであっても、提案手法によって可能となるケースが現れた。これは、初歩的検討でありながら提案手法の妥当性の表れであると考えられる。一方で、exp1, 5, 7 では従来法よりも成功率が落ちた。また、exp8 では提案手法を用いても成功することができなかった。これはパッチの評価、更新、選択がいまだ洗礼されていないためであると考えられる。

パッチの評価では距離の影響が大きく出てしまう。exp7, exp8 で用いた happyStand は円柱に近い形状をしているため、点群の腹部や台座の距離は回転していても近くなってしまう。そのため、有効な収束を行うためのパッチが残ることができなかった。これを解決するためには距離を標準化することや形状による影響を大きくするなどして、距離がパッチ評価にかかる影響を小さくする必要がある。選択法ではよりパッチの相対位置の距離を離して選択する手法を考案する必要がある。これは、部分的な局所解に陥ることを防ぎ、パッチの更新によって残されるパッチが一部に集中することを防ぐのに有効であると思われる。また、exp1, 5, 7 では収束が遅い試行のとき、収束がなされていない段階でオーバーラップするほとんどのパッチが消されてしまう失敗が観察によって見受けられた。そこでパッチの更新では世代によって係数 k を決定するのではなく、CMA-ES のステップサイズ σ を利用して決定する必要がある。

参考文献

- [1] 植西一馬, サンドバル ハイメ, 岩切宗利, 田中清: "キーポイントパッチ抽出法を用いた高効率な進化計算による 3 次元点群レジストレーション", 画像電子学会誌 The journal of the Institute of Image Electronics Engineers of Japan : visual computing, devices & communications vol.47, no.2, pp.154-166, (2018)
- [2] "The Stanford 3D Scanning Repository." <http://graphics.stanford.edu/data/3Dscanrep/>
- [3] Hansen, N. : "The CMA evolution strategy: a comparing review", Towards a new evolutionary computation. Advances on estimation of distribution algorithms, Springer, pp. 17691776, CiteSeerX 10.1.1.139.7369 (2006)
- [4] R.B. Rusu, S. Cousins: "3D is here: Point Cloud Library (PCL)", Proc. of IEEE International Conference on Robotics and Automation, pp.14, IEEE(2011).