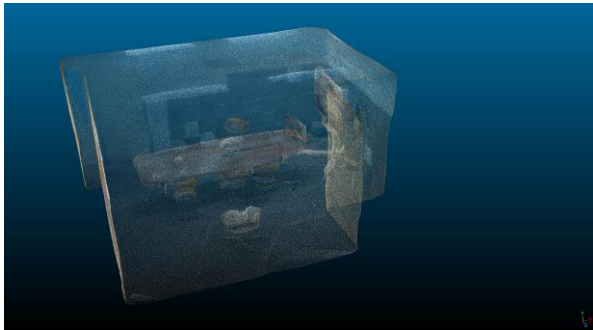
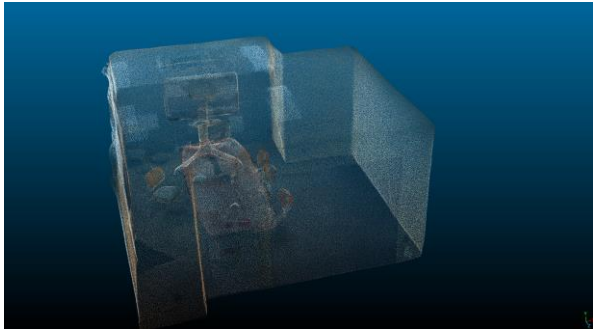


インターン課題

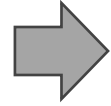
提出期限：5/6

細木 温登

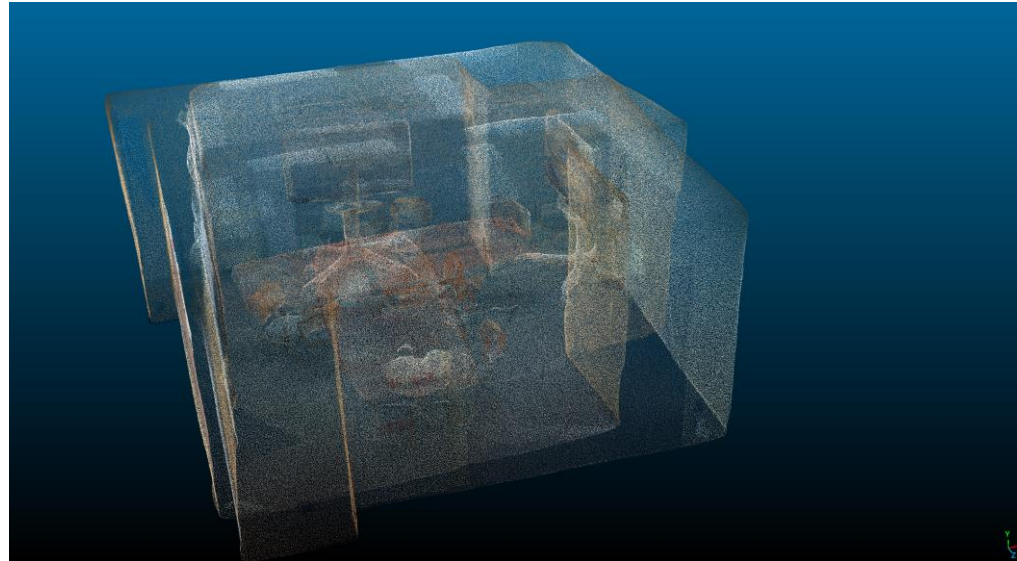
課題の要約



異なる座標系で撮影した
会議室の点群



課題テーマ：点群のレジストレーション



2つの点群を同じ座標系に表示したときの様子

これらの座標系を一致させる操作（レジストレーション）が今回の課題

技術の説明

使用したアルゴリズム：Iterative Closest Point(ICP)アルゴリズム

位置合わせ元となるソース点群と位置合わせ先となるターゲット点群を入力とし、ソース点群とターゲット点群に位置合わせするために必要な剛体変換（回転と並進）を推定する。

<アルゴリズムの流れ>

1. ソース点群とターゲット点群の対応付け
2. 剛体変換の推定
3. 物体姿勢のアップデート
4. 収束判定（収束しない場合は1.に戻る）

技術の説明

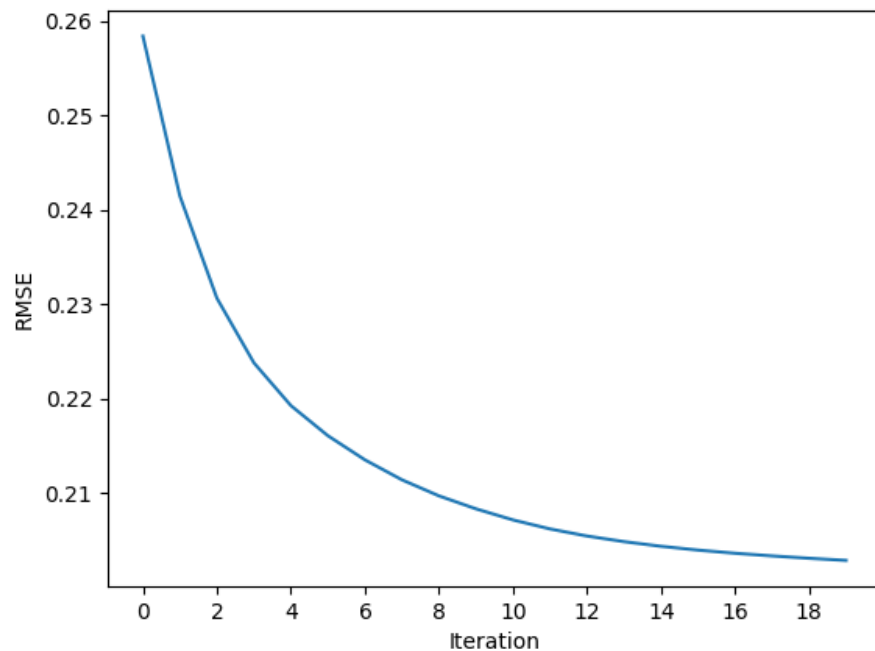
ソース点群とターゲット点群の対応付け

ソース点群とターゲット点群の対応付けには、kd-treeと呼ばれる最近傍探索法を用いた。すべてのソース点群に対して対応点を探すと処理が遅くなってしまうため、プログラムではソース点群から60000個ほどランダムに抽出して対応点を探索した。

収束判定

ソース点群とターゲット点群のレジストレーションを評価するために、RMSEを用いた。物体の姿勢をアップデートするたびにRMSEを計算し、値がどのように変化するかを可視化した。プログラムでは20回繰り返した。

結果



プログラムでは、前述した1~4の流れを20回繰り返し、レジストレーションの精度をソース点群とターゲット点群のRMSEで評価した。

グラフでは、回数を重ねるごとに、RMSEが減少し、誤差が小さくなっていることが確認できた。

考察

レジストレーションの処理を重ねるごとに、RMSEの値が減少したことから、精度が向上していったと考えられる。

最初はRMSEの値が大幅に減少したが、処理の回数を重ねるごとに減少幅が小さくなっていった。プログラムでは20回で処理を繰り返したが、これ以上回数を重ねてもRMSEの値はそれほど変化せず、頭打ちになることが考えられる。

プログラムの実行では処理時間がかかなり必要であった（PCのスペックも関係していると思う）ため、高速な処理のためにプログラムの改善も行う必要があると考えた。