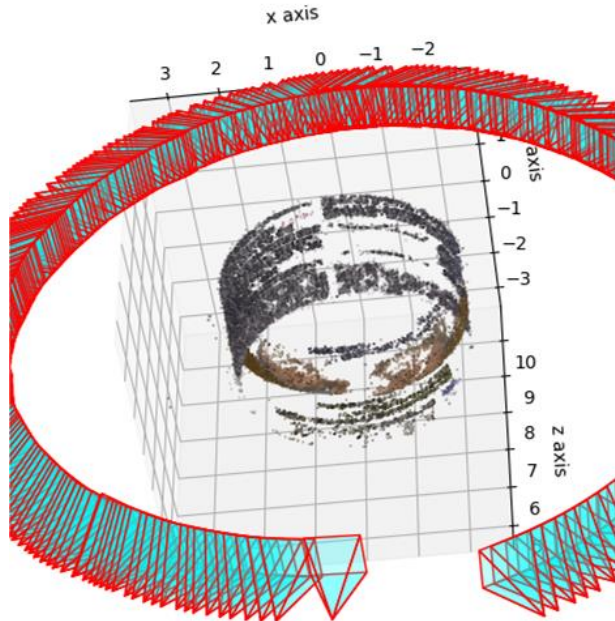


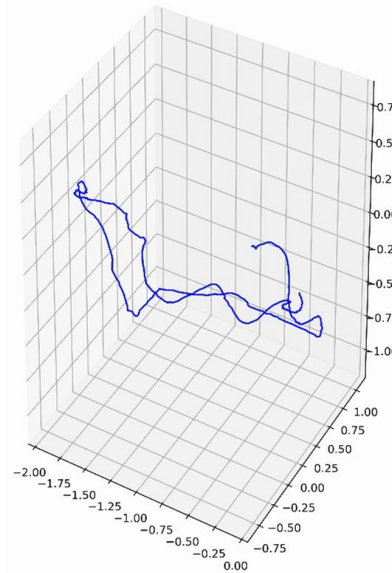
Visual SLAMフレームワークの開発

ー簡単に手法を切り替えられるフレームワークー

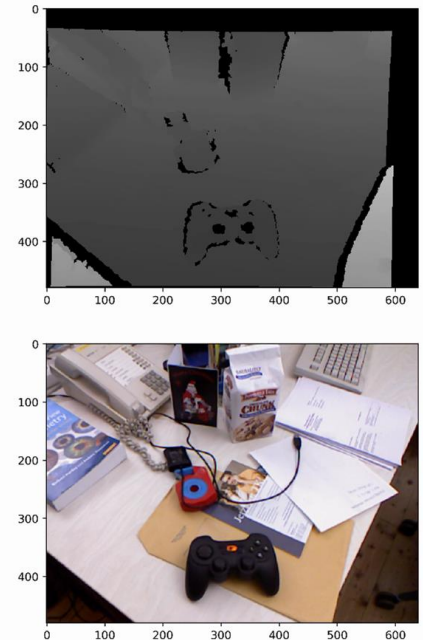
石田 岳志



Feature Based VOによる缶の復元



DVOによるカメラの移動経路の推定



【Visual SLAMとは】

画像列から3次元地図を構築する技術である。また、地図内でカメラがどこにいるかを推定することができる。自動運転における車の位置推定、ドローンの制御、建築や都市設計、ARなどに応用できる。

【Visual SLAMフレームワーク】

Visual SLAMにはさまざまな手法が提案されており、アプリケーションに応じて手法を使い分ける必要がある。しかしながら既存の実装は手法ごとに独立しており、自分のアプリケーションに適した手法を見つけることが難しかった。本プロジェクトでは、Visual SLAMの手法を簡単に切り替えられるフレームワークを作成することによって、この問題点を解決した。アプリケーションの開発者がさまざまなVisual SLAMの手法を簡単に試せるようになったことで、Visual SLAMの研究開発と産業応用の促進が期待できる。

```

1 from tadataka.dataset import TumRgbdDataset
2 from tadataka.vo import DVO
3
4
5 dataset = TumRgbdDataset(
6     "datasets/rgbd_dataset_freiburg1_desk",
7     which_freiburg=1
8 )
9
10 vo = DVO()
11
12 for frame in dataset:
13     pose = vo.estimate(frame)
14

```

リスト1：DVOを使う場合

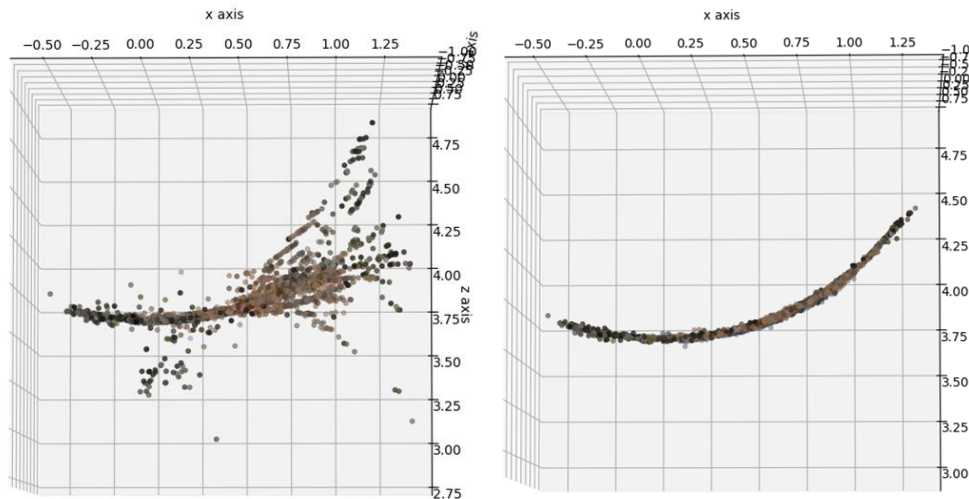


図1：左はBundle Adjustmentなしでの復元結果。右はBundle Adjustmentありでの復元結果。右は地図の破綻を防いでいる。

```

1 from tadataka.dataset import TumRgbdDataset
2 from tadataka.vo import FeatureBasedVO
3
4
5 dataset = TumRgbdDataset(
6     "datasets/rgbd_dataset_freiburg1_desk",
7     which_freiburg=1
8 )
9
10 vo = FeatureBasedVO()
11
12 for frame in dataset:
13     pose = vo.estimate(frame)
14

```

リスト2：FeatureBasedVOを使う場合



図2：作成したデータセット。形状が単純なため復元結果がわかりやすい。

本プロジェクトで開発したフレームワークにはTadatakaという名前をつけた。Tadatakaを使うと、たった2行書き換えるだけで別々の手法を切り替えることができる（リスト1，2）。

地図の復元過程では誤差が蓄積してしまう。Bundle Adjustmentを用いると、復元過程で地図が破綻することを防ぐことができる。本プロジェクトではBundle Adjustmentをパッケージとして切り離れた（図1）。

本プロジェクトではデータセットの作成を行った。Visual SLAMの既存のデータセットは部屋や建造物などを撮影したものが多く、被写体の形状が複雑である。このため、手法の開発段階で復元が正しく行えているかを確認することが難しかった。本プロジェクトでは缶詰という単純な形状の物体を撮影することで、Visual SLAMの開発段階でも扱いやすいデータセットを構築した（図2）。