

SLAM開発における課題と 対策の一例の紹介

3D勉強会 #6

井上

自己紹介



井上

株式会社Preferred Networks エンジニア



学生

SfMをフルスクラッチで書く

1社目

MVSをフルスクラッチで書く

2社目

SLAMをフルスクラッチで書く



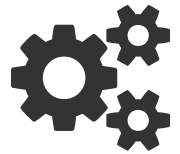
SLAMは楽しい

 SLAMの開発はつらい

どうしてつらいか？



真値が取れない



大量のハイパーパラメーター



テストしにくい

POINT 01

真値が取れない



画像出典：colmap.github.io

カメラポーズ・点群はどの程度正しい？

評価のための真値が必要

真値をとるためにはコストがかかる

真値あり公開データセットを使う
or
コストをかけて真値を取る

データセット例 1/2

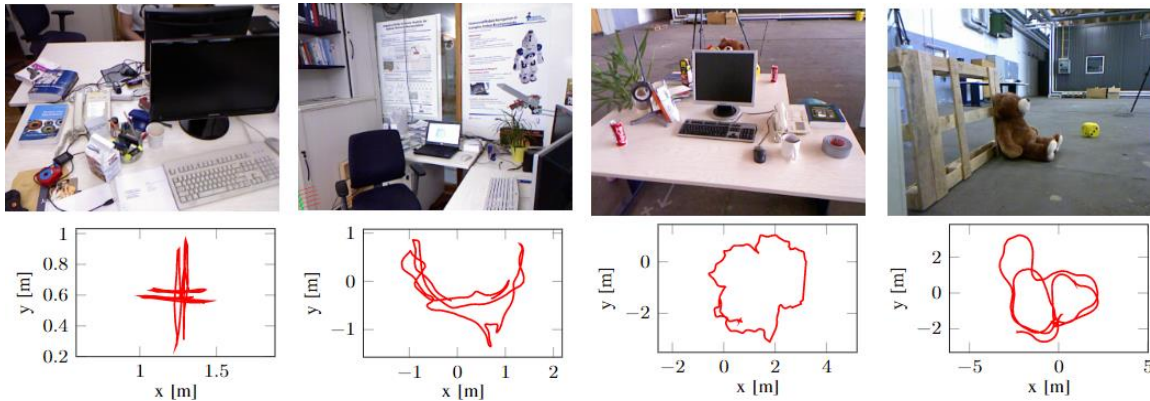


KITTI [IJRR 2013]
屋外、ステレオ、
LiDAR、GPS、IMU



EuRoC [IJRR 2016]
屋内、ステレオ、IMU、
モーションキャプチャ、
レーザースキャナ

データセット例 2/2



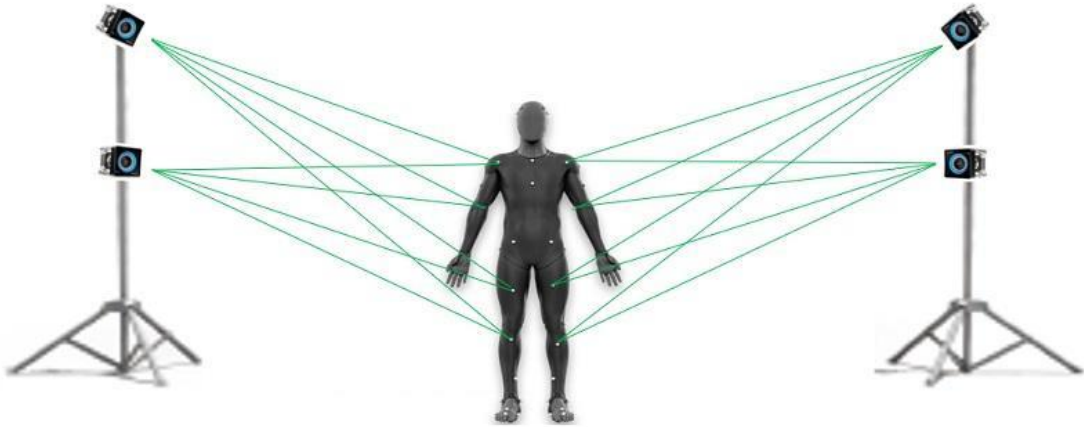
TUM RGB-D [IROS 2012] **ICL-NUIM [ICRA 2014]**
屋内、RGB-D、
モーションキャプチャ
屋内、CG

データセット例 まとめ

Dataset	ステレオ	Depth Map	IMU	カメラ姿勢	構造	屋内外
KITTI	○	-	○	3D (GPS)	点群	屋外
EuRoC	○	-	○	6D (mocap)	点群	屋内
TUM RGB-D	-	Dense	-	6D (mocap)	-	屋内
ICL-NUIM	-	Dense	-	6D (synthetic)	サーフェス	屋内

- センサ構成がマッチしている場合には積極的に使いたい
- 他手法との比較にも重宝する
- これだけで評価が十分ということにはならない

真値を取る方法（カメラポーズ）



画像出典：optitrack.jp

モーションキャプチャ
100Hz
0.1mm



画像出典：www.topcon.co.jp

トータルステーション
10Hz
1～5mm

※ 数値は大雑把な目安です。製品によって異なります。以降同様。

真値を取る方法（構造）



画像出典：velodynelidar.com

LiDAR

10～50Hz

30～100mm



画像出典：faro.com

レーザースキャナー

0.001Hz

1mm w/RGB

真値を取る方法 まとめ

カメラポーズセンサー		対象	精度	測定速度	測定範囲	価格
モーションキャプチャー		位置+回転	0.1mm	100Hz	10m	数百万円
トータルステーション		位置	1～5mm	10Hz	5000m	数百万円

構造センサー		色	動物体	精度	測定速度	測定範囲	価格
LiDAR		-	○	30～100mm	10～50Hz	200m	数百万円
レーザースキャナー		○	-	1mm	0.001Hz	1000m	数百万円

- 得意不得意があり、解きたい問題ごとに最適なセンサーを選ぶ必要がある
- センサー間のレジストレーション等泥臭い作業が必要

まとめ

最初は公開データセットで評価しつつ、
目的のシーンで真値を計測し評価をする

POINT 02

大量のハイパーパラメーター

SLAMはハイパーパラメーターだらけ

画像サイズ

ロバストコスト関数選択

LM法のlambda

イテレーション回数

VisionとIMUのバランス

外れ値しきい値

画像ピラミッドレイヤー数

RANSAC試行回数

収束判定条件

初期復元ペアの選び方

特徴点数

各種Priorの重み

etc.

自動で最適化する



Optuna: A hyperparameter optimization framework

[pypi](#) [v0.19.0](#) [license](#) [MIT](#) [PASSED](#) [docs](#) [passing](#) [codecov](#) [89%](#) [chat](#) [on gitter](#)

[Website](#) | [Docs](#) | [Install Guide](#) | [Tutorial](#)

Optuna is an automatic hyperparameter optimization software framework, particularly designed for machine learning. It features an imperative, *define-by-run* style user API. Thanks to our *define-by-run* API, the code written with Optuna enjoys high modularity, and the user of Optuna can dynamically construct the search spaces for the hyperparameters.

出典：<https://github.com/optuna/optuna>

```
import optuna
from subprocess import run
```

探索するパラメーターの
種類と範囲を指定

```
def objective(trial):
    threshold = trial.suggest_uniform('threshold', 0.1, 10.0)
    iter      = trial.suggest_int('iter', 50, 300)
```

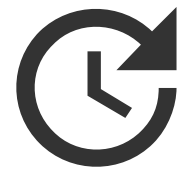
```
    score = run(
        ['your/slam/binary', f'{threshold}', f'{iter}'],
        capture_output=True, text=True).stdout
    return score
```

```
study = optuna.create_study()
study.optimize(objective, n_trials=100)
```

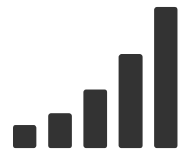
SLAMを実行して
評価結果を返す

公式ドキュメントや講演資料を参照

- <https://optuna.readthedocs.io/>
- <https://www.slideshare.net/pfi/pydatatokyo-meetup-21-optuna>



1. 復元から評価まで自動化する



2. 最大(小)化したい評価指標を定義



OPTUNA

実績：手動調整より**10%向上**

まとめ

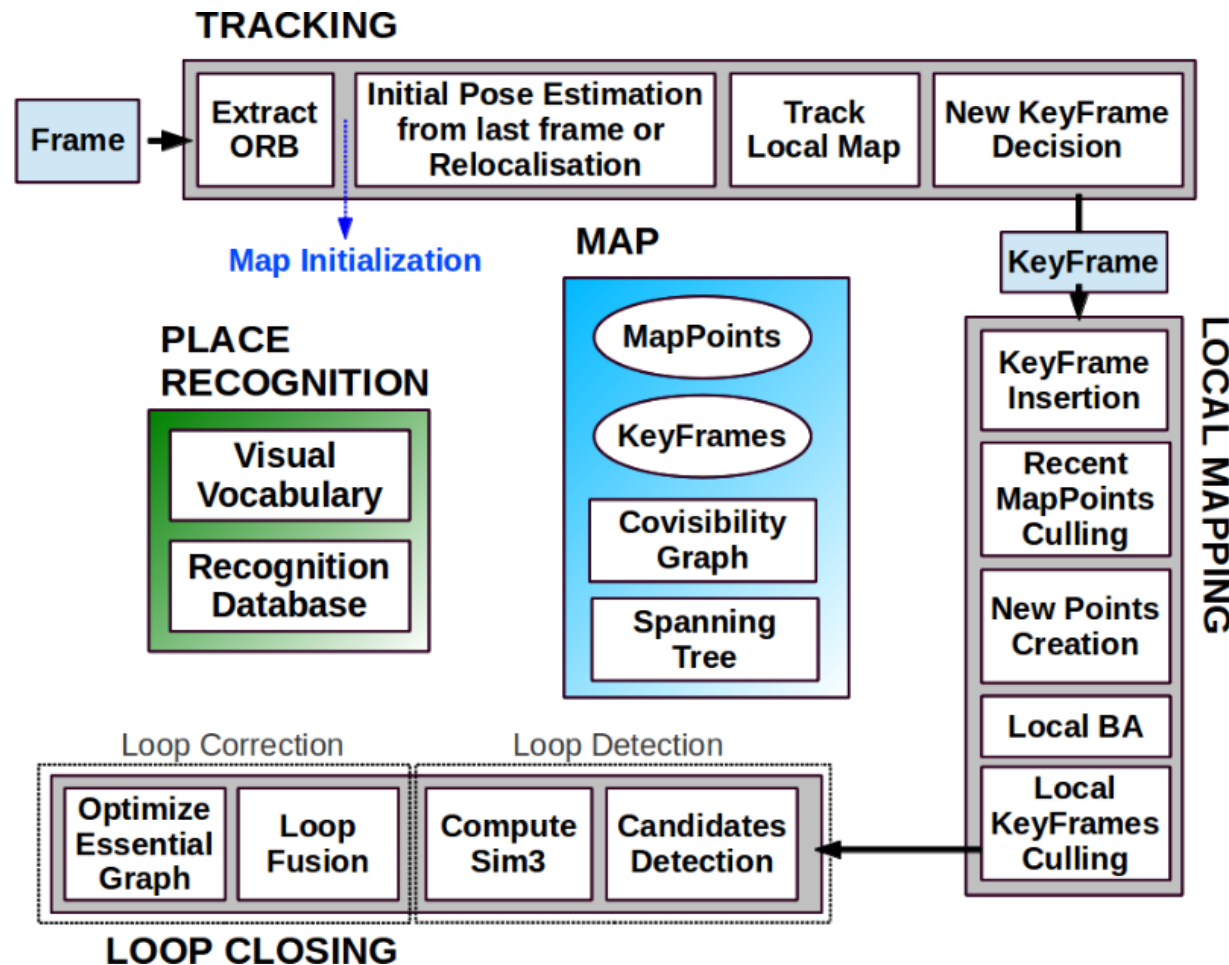
ハイパーパラメーターの調整は自動化できる

課題

- 評価指標をどう定義するか？

POINT 03

テストしにくい




モジュールの結合度が高くなりがち

モジュールの結合度が高くなりがちで
ユニットテストがしにくい
= バグを発見できない

バグを減らすテクニックを併用する

バグを減らすテクニック

 静的解析ツール・Sanitizer（省略）

 代数計算ツール

 継続的な性能のトラッキング

代数計算ツール

$$r(x) = y - f(x)$$

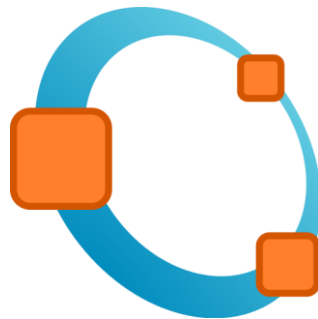
$$J = \frac{\partial r(x)}{\partial x}$$

$$x \leftarrow x - (J^T J)^{-1} J^T r(x)$$

SLAMではGauss-Newton法をよく使う
= ヤコビ行列を書く必要がある



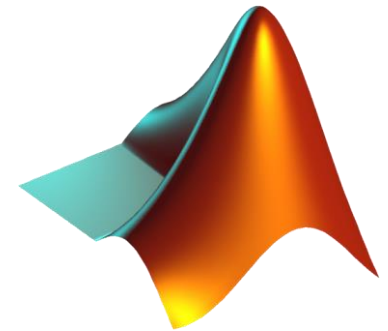
SymPy



Octave



Mathematica



MATLAB

微分計算を自動化

```
[3] # 透視投影
sympy.Matrix([f_x * x / z + c_x, f_y * y / z + c_y])
```



$$\begin{bmatrix} c_x + \frac{f_x x}{z} \\ c_y + \frac{f_y y}{z} \end{bmatrix}$$

```
[4] # ヤコビ行列
_.jacobian([x, y, z, f_x, f_y, c_x, c_y])
```



$$\begin{bmatrix} \frac{f_x}{z} & 0 & -\frac{f_x x}{z^2} & \frac{x}{z} & 0 & 1 & 0 \\ 0 & \frac{f_y}{z} & -\frac{f_y y}{z^2} & 0 & \frac{y}{z} & 0 & 1 \end{bmatrix}$$

```
[5] # C++コード出力
codegen(_)
```



```
const double k0 = 1.0/z;
const double k1 = std::pow(z, -2);
Eigen::Matrix<double, 2, 7> jacobian;
jacobian << f_x*k0, 0, -f_x*k1*x, k0*x, 0, 1, 0, 0, f_y*k0, -f_y*k1*y, 0, k0*y, 0, 1;
```

<https://colab.research.google.com/drive/1wflhGRVzdlosxHsC63HX2WvXrCG-b8p0>

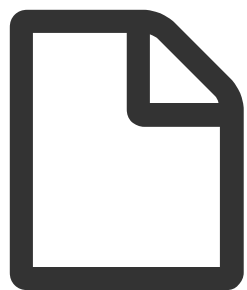
代数計算ツール まとめ

考慮できないものもある

- リー群の群作用の偏微分
- 浮動小数点数の計算誤差
- SIMD

最終的には、手で書いた微分と数値微分の計算結果が一致するかテストしている

継続的な性能のトラッキング



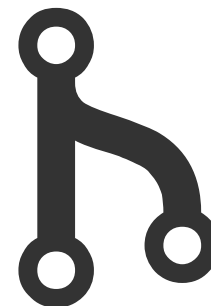
コーディング



単体テスト



レビュー



マージ



統合テスト
毎時
毎週

テスト結果をどう管理する？



MLflow: A Machine Learning Lifecycle Platform

MLflow is a platform to streamline machine learning development, including tracking experiments, packaging code into reproducible runs, and sharing and deploying models. MLflow offers a set of lightweight APIs that can be used with any existing machine learning application or library (TensorFlow, PyTorch, XGBoost, etc), wherever you currently run ML code (e.g. in notebooks, standalone applications or the cloud). MLflow's current components are:

- **MLflow Tracking:** An API to log parameters, code, and results in machine learning experiments and compare them using an interactive UI.
- **MLflow Projects:** A code packaging format for reproducible runs using Conda and Docker, so you can share your ML code with others.
- **MLflow Models:** A model packaging format and tools that let you easily deploy the same model (from any ML library) to batch and real-time scoring on platforms such as Docker, Apache Spark, Azure ML and AWS SageMaker.

docs latest build passing pypi v1.4.0 conda-forge v1.4.0 cran v1.4.0 maven-central v1.4.0 license Apache 2

出典：<https://github.com/mlflow/mlflow>

Experiments

Default
scene_01
scene_02
scene_03
scene_04



scene_04

Experiment ID: 4

Artifact Location: file:///mlruns/4

▼ Description: [\[link\]](#)Search Runs:

State: Active

Search

Filter Params: Filter Metrics:

Showing 11 matching runs

Compare

Delete

Download CSV



ハイパラ



評価結果

試行



<input type="checkbox"/>	Date	User	Run Name	Source	Version	Tags	Parameters	rotation_error	translation_error
<input type="checkbox"/>	2019-12-12 17:48:42	root		report...	1acb06		ba_iter: 19 image_size: 600 pyramid_level: 4 threshold: 3	0.161	0.51
<input type="checkbox"/>	2019-12-12 17:48:42	root		report...	1acb06		ba_iter: 18 image_size: 600 pyramid_level: 4 threshold: 3	0.201	0.62
<input type="checkbox"/>	2019-12-12 17:48:42	root		report...	1acb06		ba_iter: 17 image_size: 600 pyramid_level: 4 threshold: 3	0.252	0.73
<input type="checkbox"/>	2019-12-12 17:48:42	root		report...	1acb06		ba_iter: 16 image_size: 600 pyramid_level: 4 threshold: 3	0.315	0.84
<input type="checkbox"/>	2019-12-12 17:48:41	root		report...	1acb06		ba_iter: 15 image_size: 600 pyramid_level: 4 threshold: 3	0.393	0.95
<input type="checkbox"/>	2019-12-12 17:48:41	root		report...	1acb06		ba_iter: 14 image_size: 600 pyramid_level: 4 threshold: 3	0.492	1.06
<input type="checkbox"/>	2019-12-12 17:48:41	root		report...	1acb06		ba_iter: 13 image_size: 600 pyramid_level: 4 threshold: 3	0.614	1.17

▼ Metrics

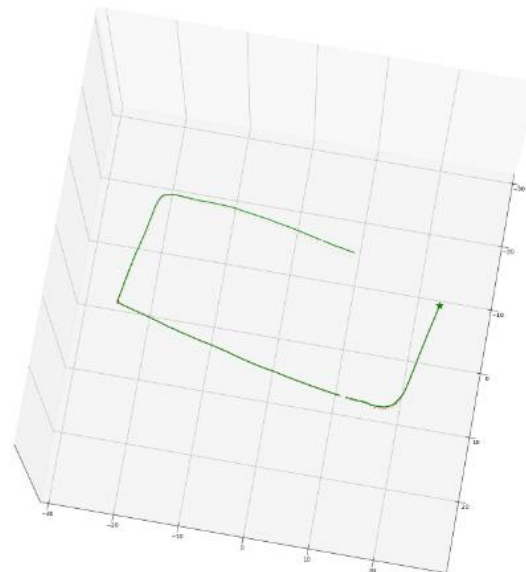
Name	Value
rotation_error	0.161
translation_error	0.51

▶ Tags

▼ Artifacts

config.json
log.txt
points.xyz
trajectories.png
trajectories.xyz

Full Path: file:///mlruns/4/b356c2ab362a44edbe1338cf95740993/artifacts/trajectories.png
Size: 188.74KB



👉 各試行の結果

Parameters

ba_iter	19	18	17	16	15	14	13	12	11
image_size	600	600	600	600	600	600	600	600	600
pyramid_level	4	4	4	4	4	4	4	4	4
threshold	3	3	3	3	3	3	3	3	3

Metrics

rotation_error 🔗	0.161	0.201	0.252	0.315	0.393	0.492	0.614	0.768	0.96
translation_error 🔗	0.51	0.62	0.73	0.84	0.95	1.06	1.17	1.28	1.39

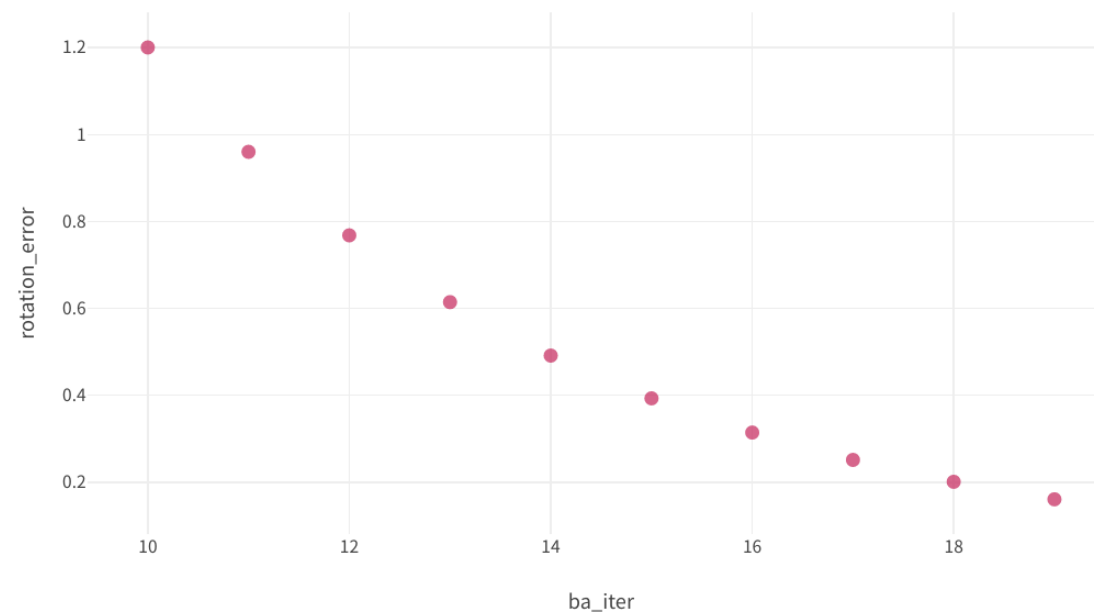


X-axis:

ba_iter

Y-axis:

rotation_error



👉 ハイパラ vs. 評価結果

まとめ

積極的に自動化してバグが混入する余地を減らす

混入してもすぐに検出できるシステムを構築する

SLAMの開発はつらいが
軽減する方法はある

 楽しくSLAM開発！

参考文献

- The KITTI Vision Benchmark Suite
<http://www.cvlibs.net/datasets/kitti/>
- The EuRoC MAV Dataset
<https://projects.asl.ethz.ch/datasets/doku.php?id=kmaavvisualinertialdatasets>
- RGB-D SLAM Dataset and Benchmark
<https://vision.in.tum.de/data/datasets/rgbd-dataset>
- The ICL-NUIM dataset
<https://www.doc.ic.ac.uk/~ahanda/VaFRIC/iclnuim.html>
- LiDAR Comparison Chart
<https://autonomoustuff.com/lidar-chart/>