CS133 Final Project
Optimization of Visual Odometry System

Ruiqi LIU & Haomin SHI & Zilin SI

CONTENTS

01 Introduction

02 Simple Visual Odometry System

03 Bundle Adjustment

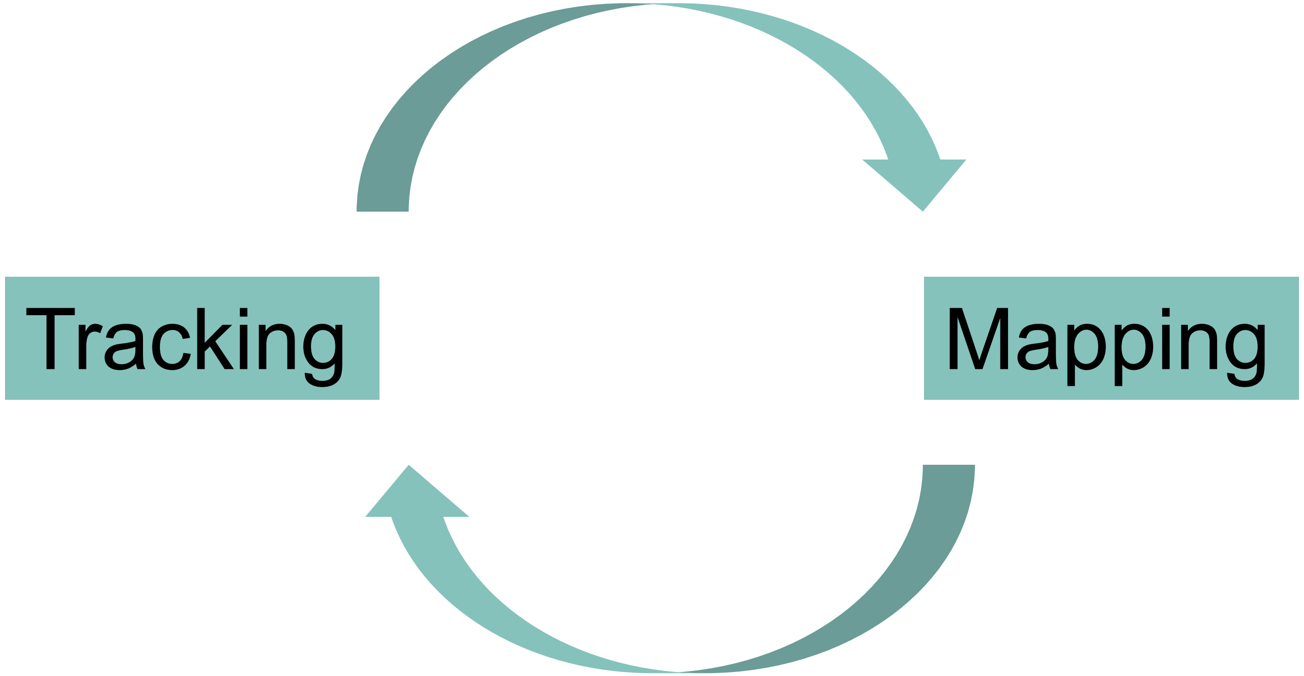
04 Loop Closure

05 Results & Future Works

Introduction

01

Background



Tracking

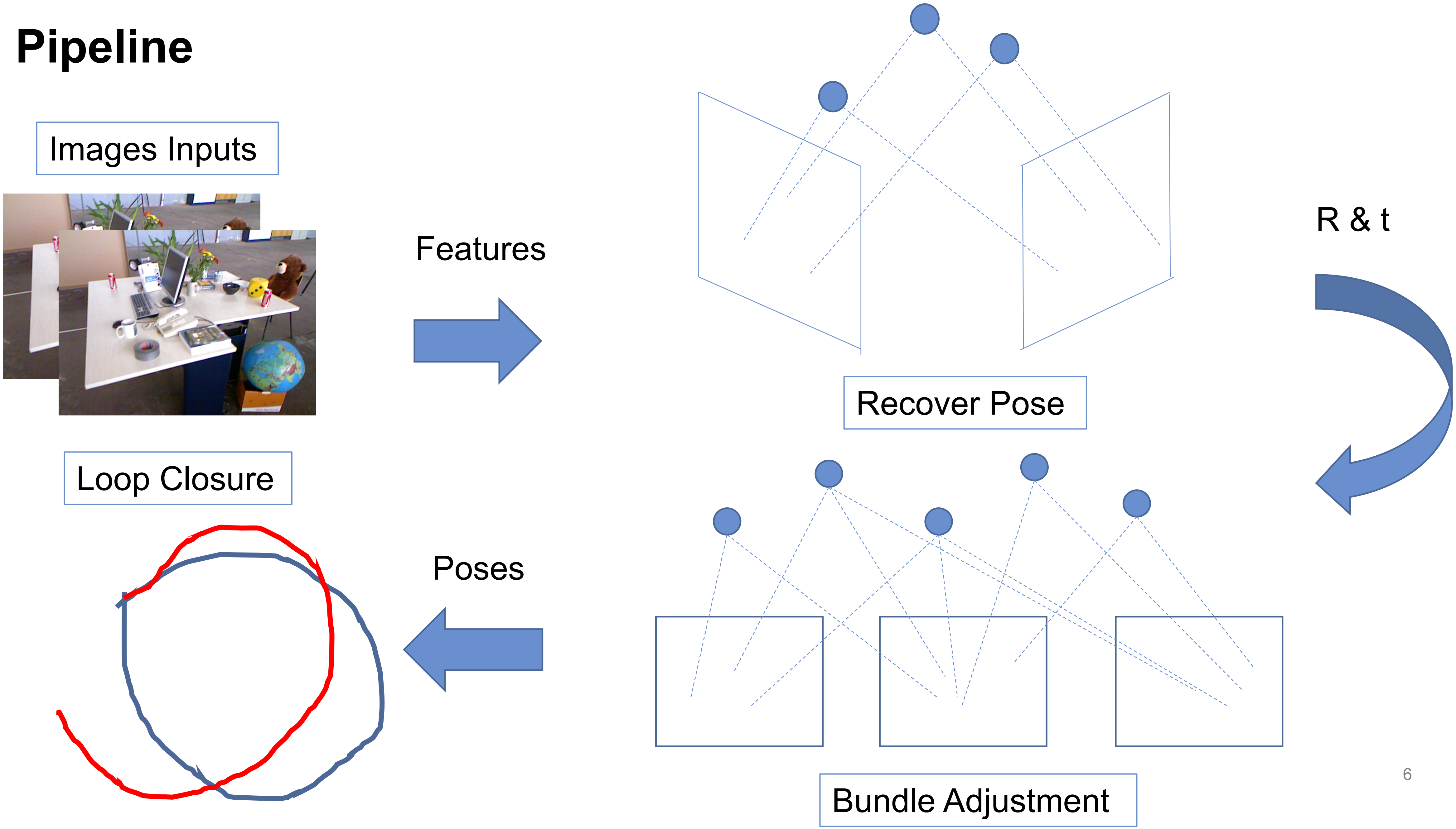
Mapping

Motivations & Goals

- Visual odometry is the basic part for tracking
- Clear pipeline, easier to collaborate in a group

- To implement the full visual odometry
- Improve the accuracy by doing local optimization
- Decrease the drift error by doing global optimization

Pipeline



Images Inputs

Features

Recover Pose

R & t

Bundle Adjustment

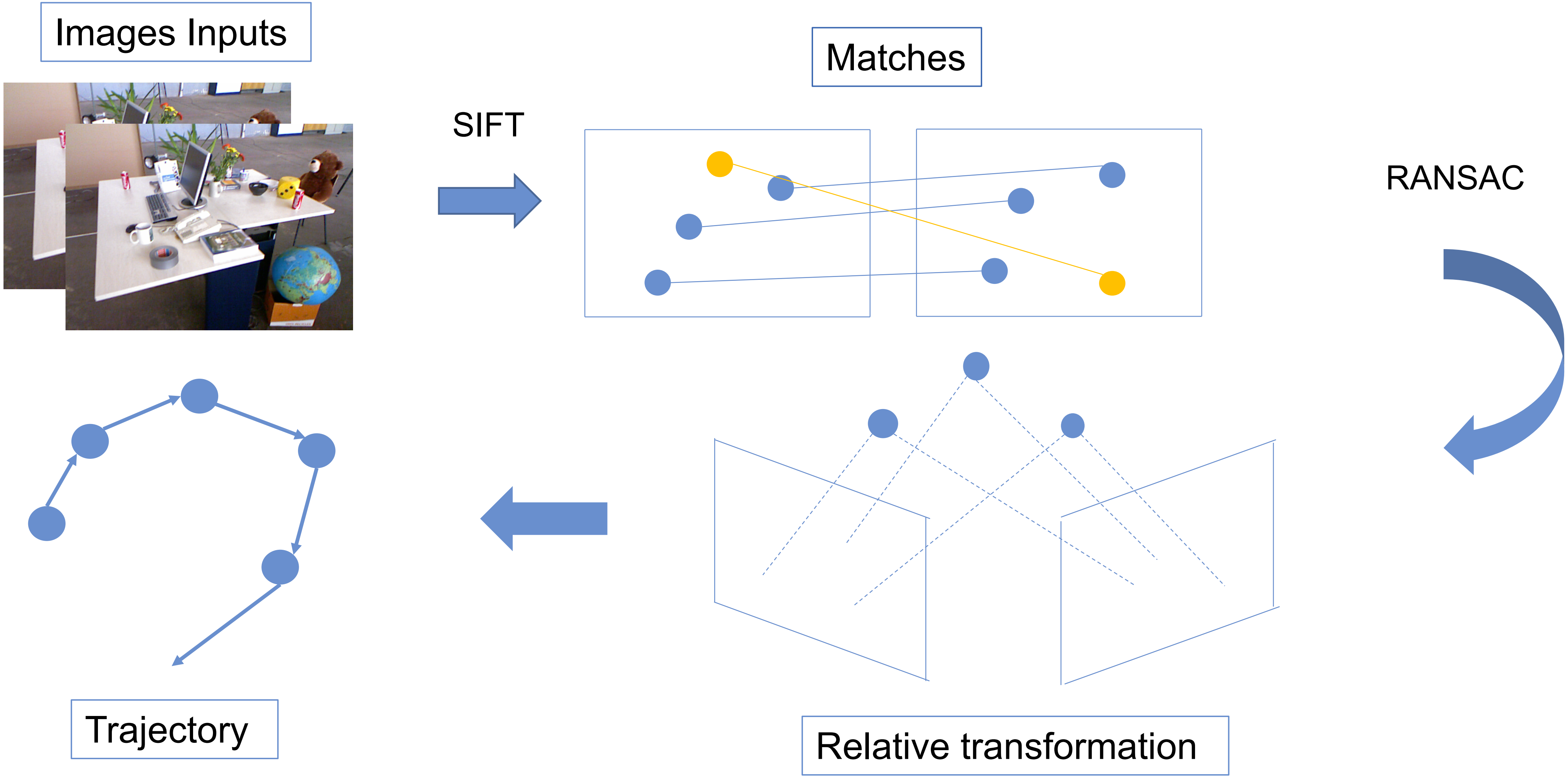
Poses

Loop Closure

Simple Visual Odometry System

02

Visual Odometry



Images Inputs

SIFT

Matches

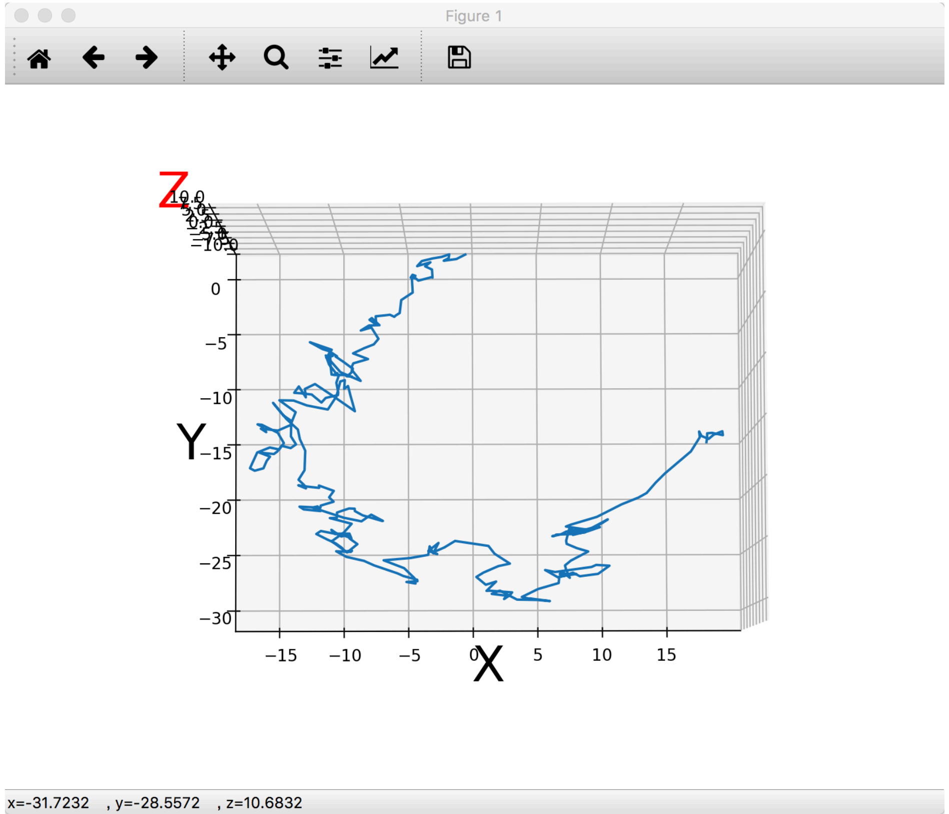
RANSAC

Relative transformation

Trajectory

Implementation

- Pick a key frame from each ten frames
- Extract SIFT features to do matching
- Based on Opencv
- Apply RANSAC with 90% certainty to find the correspondence matches & essential matrix

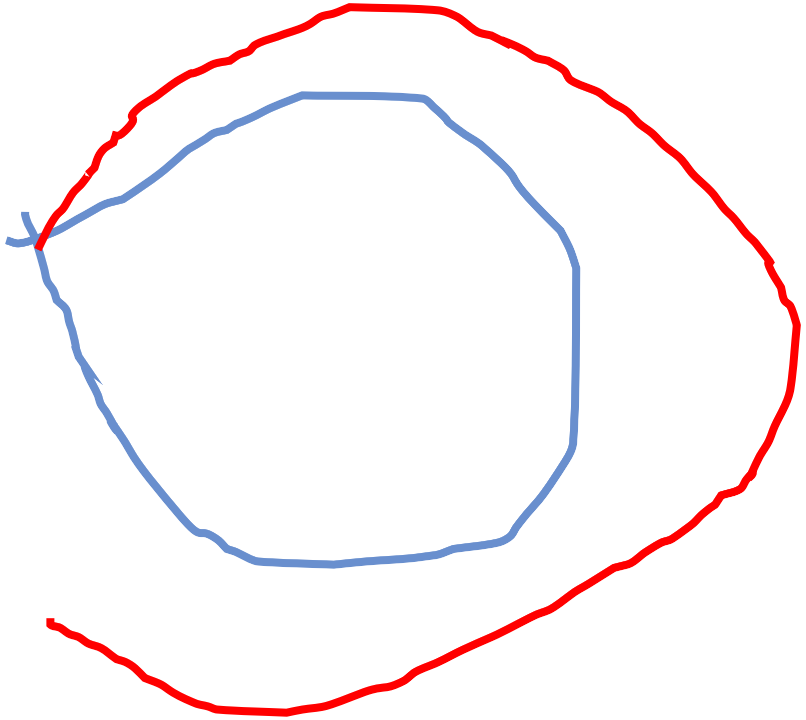


x=-31.7232 , y=-28.5572 , z=10.6832

Some problems about VO

- Scale ambiguity
- Error accumulation and drift

Need optimization!



Bundle Adjustment

03

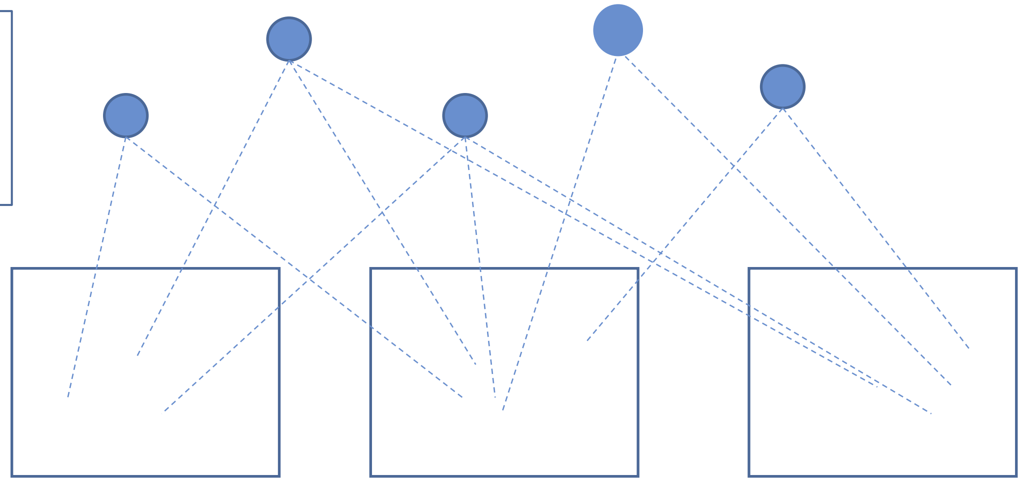
Local Bundle Adjustment

Inputs:

- 2D points in a set of images
- Corresponding 3D points
- Initial positions of cameras and 3D points
- Camera calibration
- Reprojection error function

Output:

- Optimized R(Rotation), t(translation), X(3D point's positions)



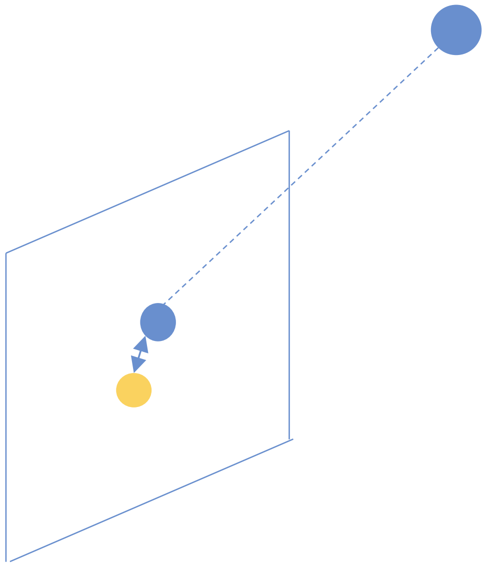
Local Bundle Adjustment

Optimization objective:

$$\{t_{i,opt},\ R_{i,opt},\ X_{j,opt}\ \} = min_{t_i,R_i,X_j} \sum_{t_i,R_i,X_j} d_{i,j}^2$$

Reprojection Error:

$$d_{i,j} = \left\| x_j^i - \frac{K_{2*3} * (R_i * X_j + t_i)}{K_{1*3} * (R_i * X_j + t_i)} \right\|$$

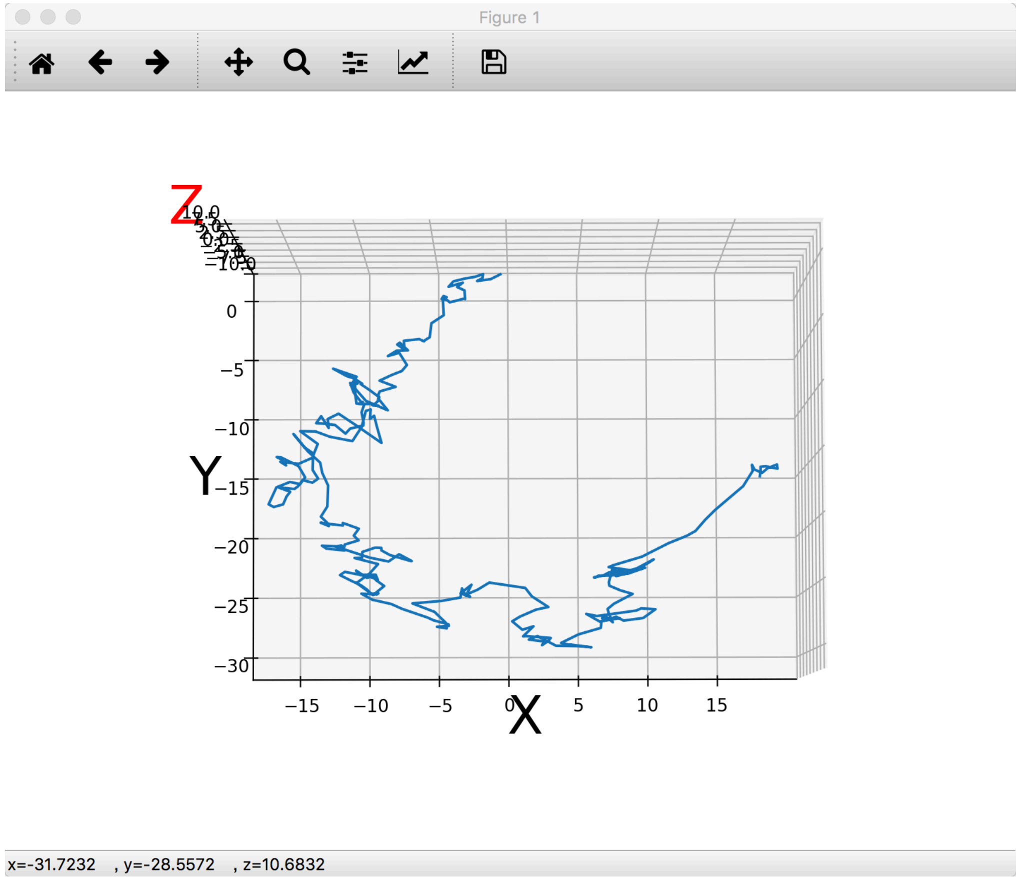


Implementation

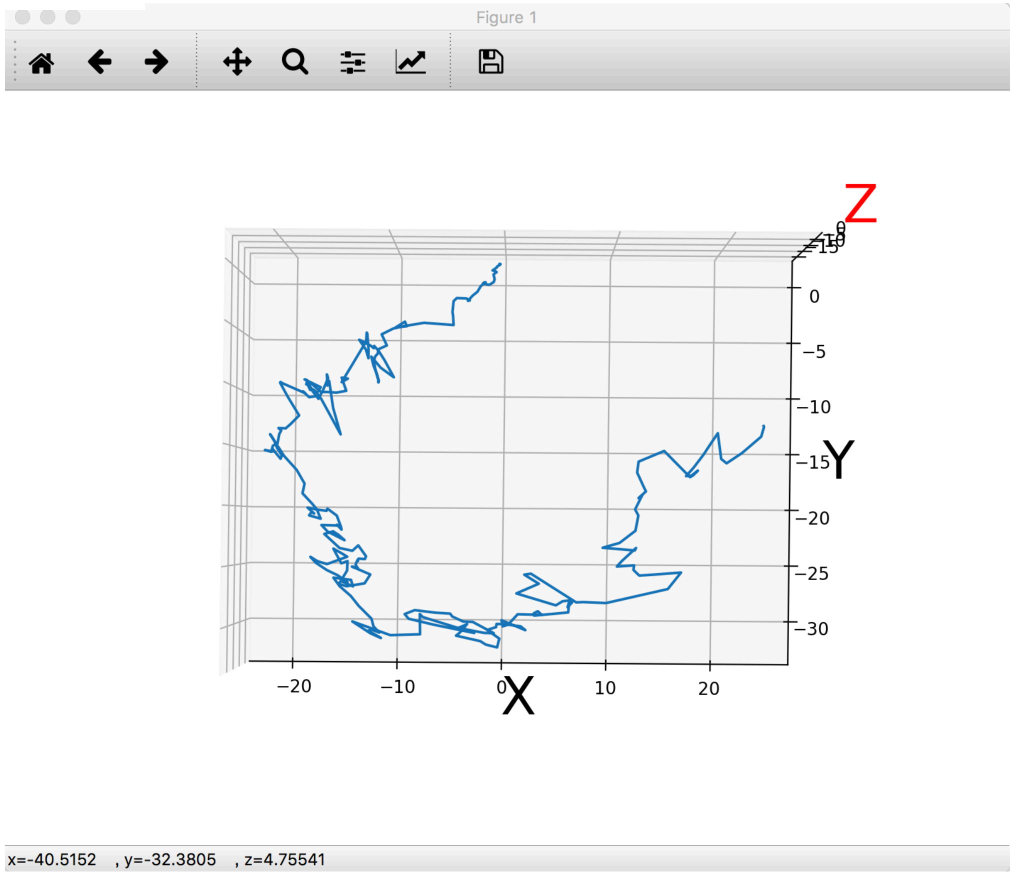
- Each ten key frames do once bundle adjustment
- Based on Ceres
- Solver = Levenberg Marquardt
- Maximum number of iterations = 200
- Use Sparse Schur
- Use Quaternion to represent Rotation

Results

```
Bundle Adjustment statistics (approximated RMSE):
 #residuals: 7772
 Initial RMSE: 3.95806
 Final RMSE: 1.41796
 Time (s): 60.7058
```



x=-31.7232 , y=-28.5572 , z=10.6832

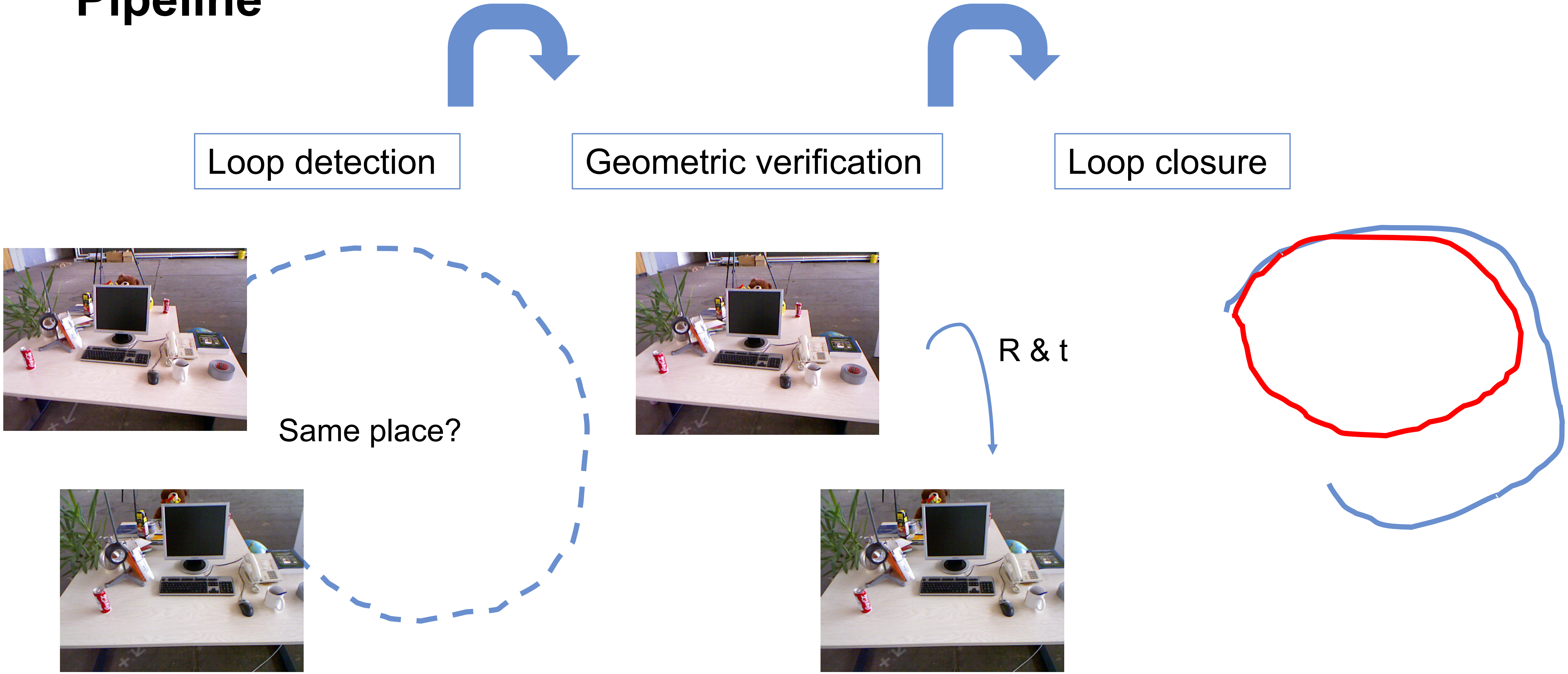


x=-40.5152 , y=-32.3805 , z=4.75541

Loop Closure

04

Pipeline



Loop detection

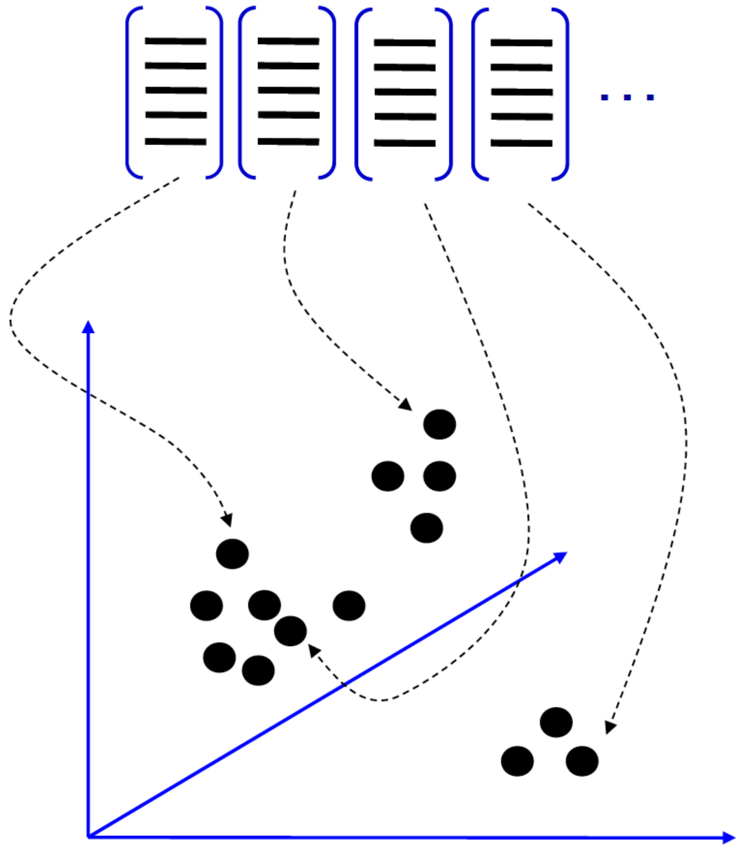
Geometric verification

Loop closure

Loop detection

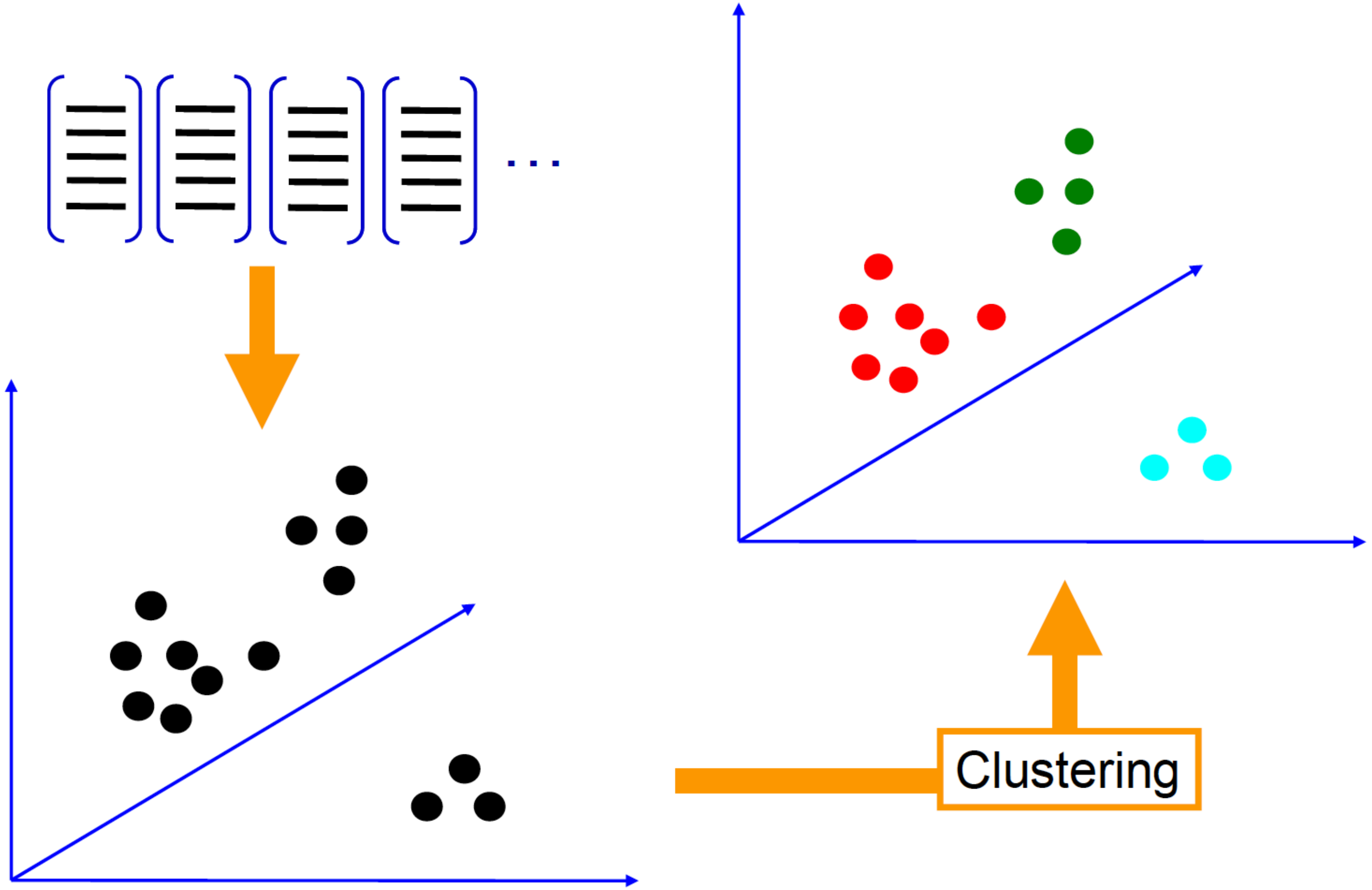
- Build bag of words dictionary

Extract feature’s descriptors



Loop detection

- Build bag of words dictionary

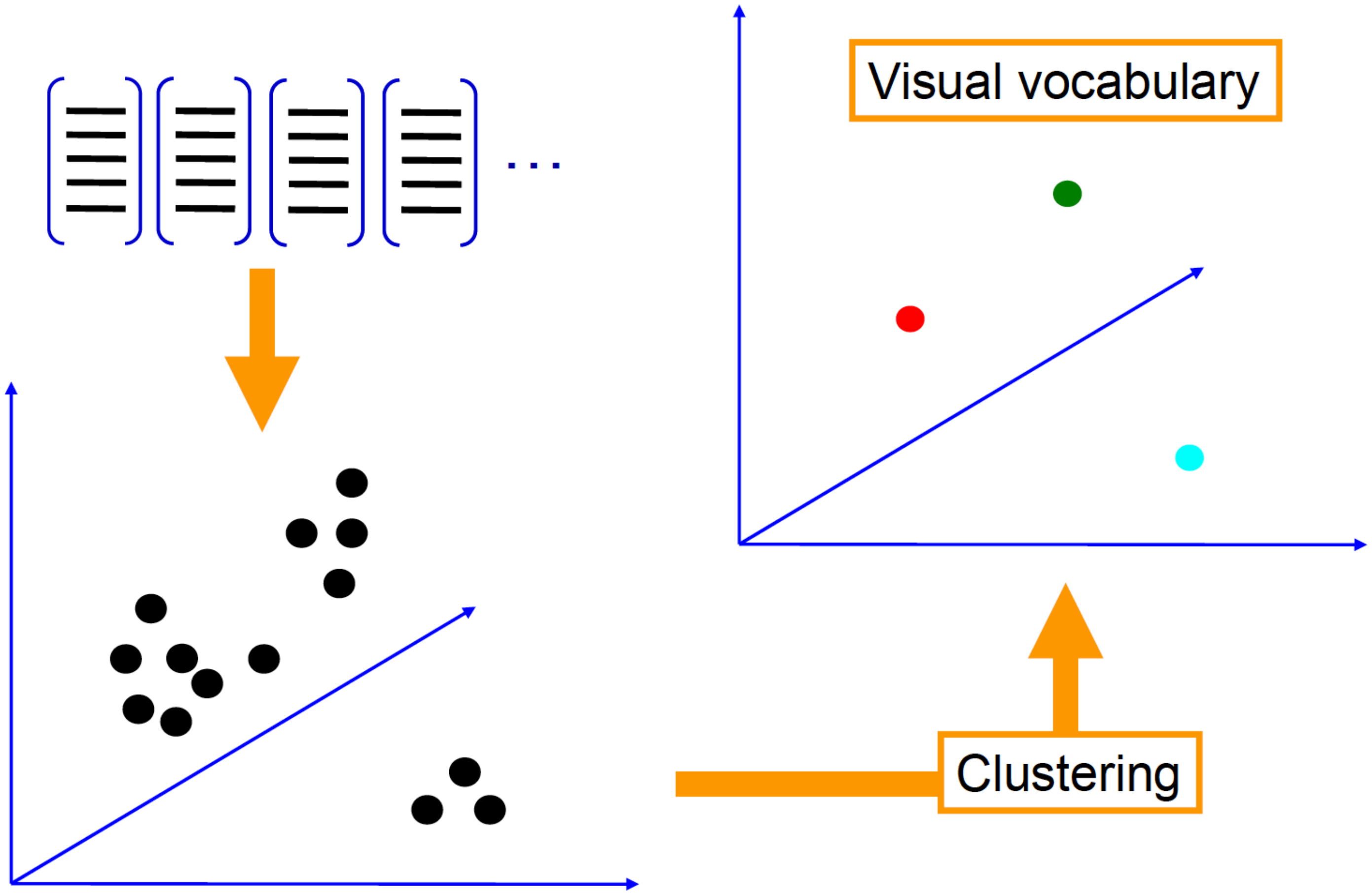


Clustering

Cluster by K-means

Loop detection

- Build bag of words dictionary



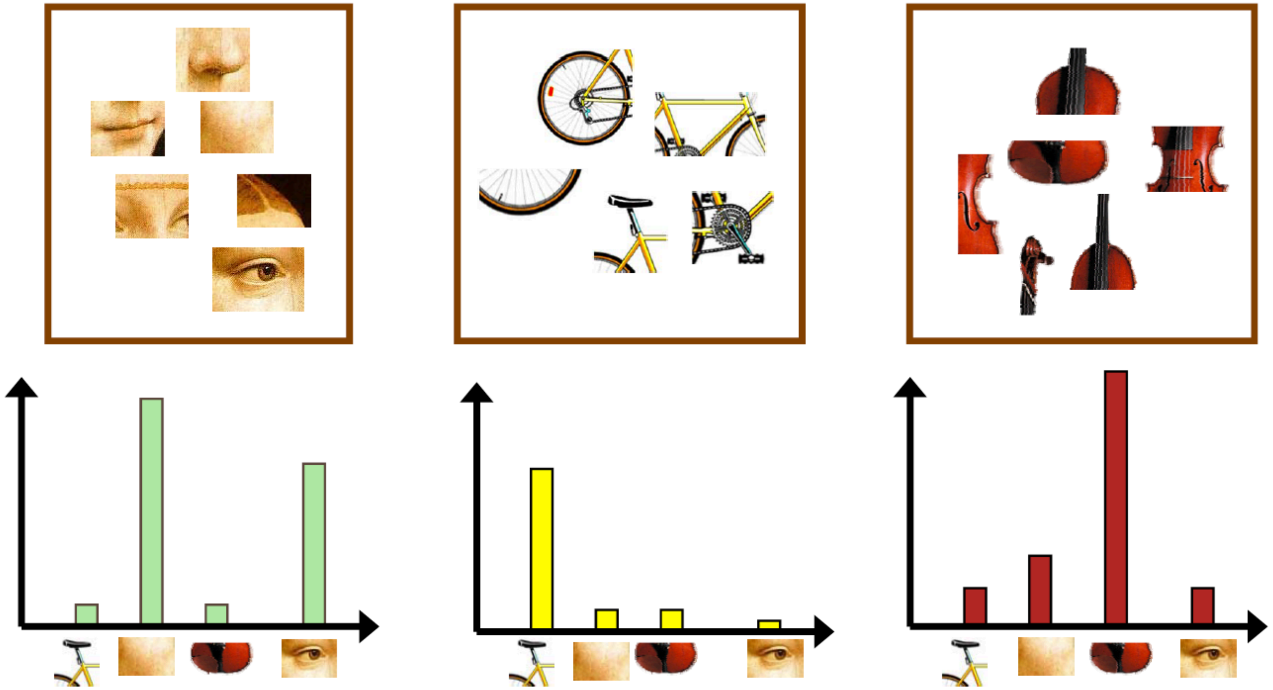
Visual vocabulary

Clustering

Use cluster centers as visual vocabularies to build the dictionary

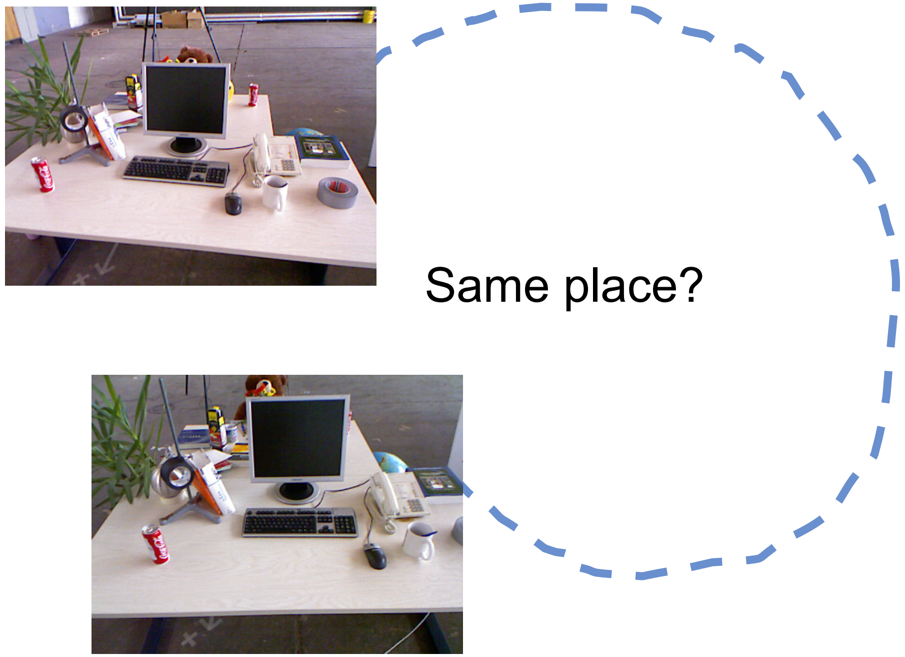
Loop detection

- Build bag of words dictionary
- Represent images by the frequencies of visual words



Loop detection

- Build bag of words dictionary
- Represent images by the frequencies of visual words
- Detect the same place by comparing the bow descriptors

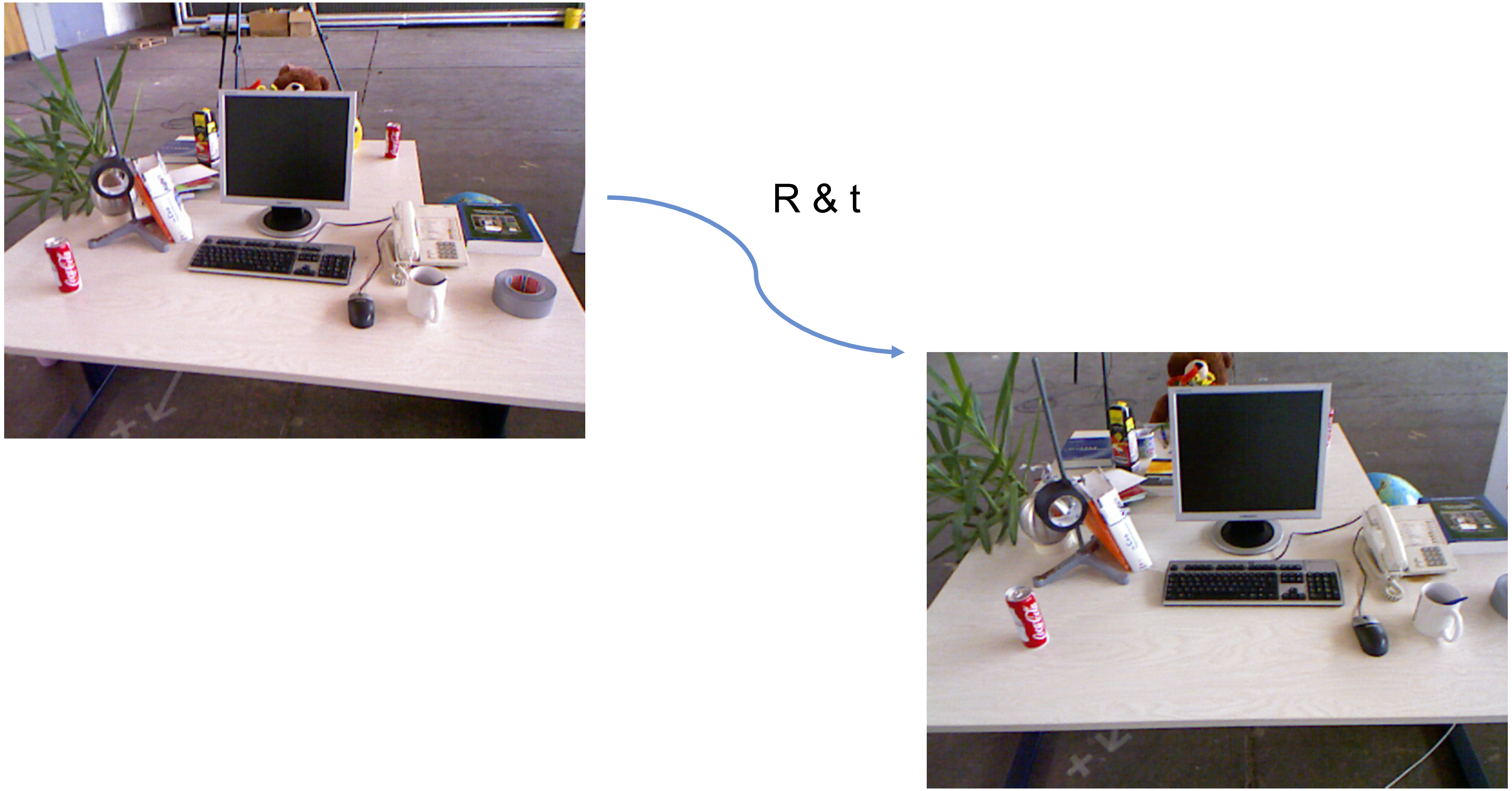


Same place?

Implementation

- Randomly choose images from dataset to train the bow
- Dictionary size = 100
- Dissimilarity = $\sum_i ||x_i - y_i||$
- Do not consider the nearest 50 key frames
- The similar threshold is 0.31
- Each time search 10 frames uniformly, if not find the similar frame, then constrict the search range until find the similar frame or cannot search more.

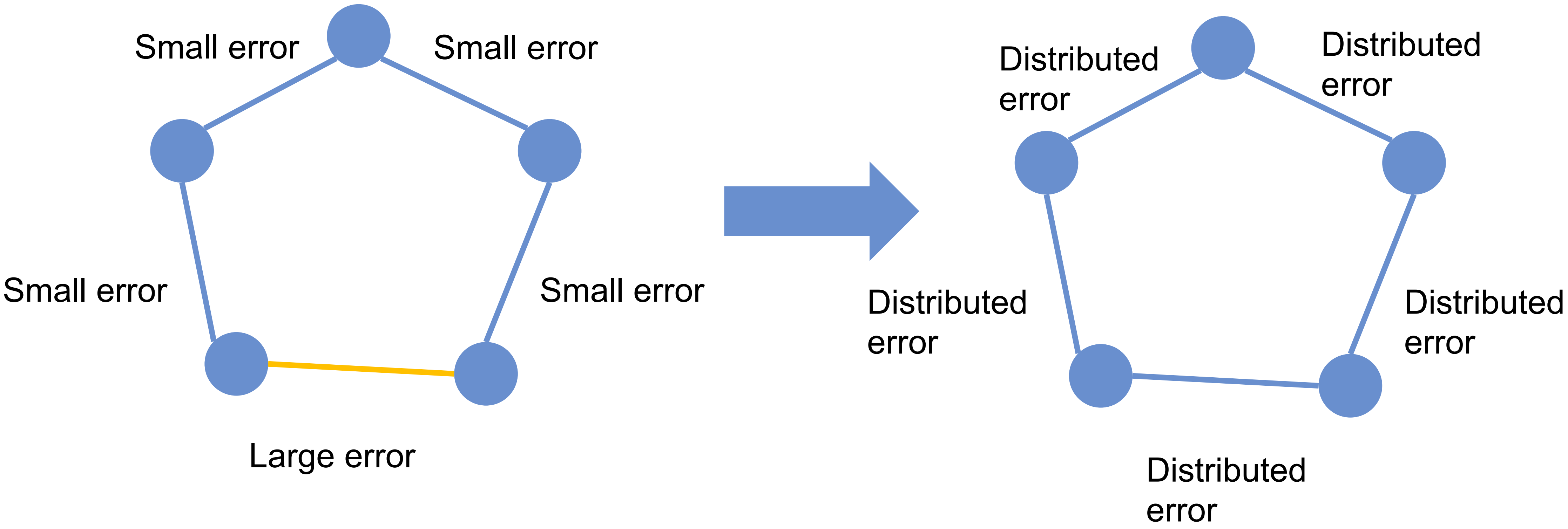
Geometric verification



R & t

Loop closure

- Use pose graph optimization



Loop closure

- Use pose graph optimization

Optimization objective:

x^{*} = $argmin_{x} \sum_{\{i,j\}} e_{ij}^{T}$(x)*$\Omega_{ij}$*$e_{ij}$ (x)

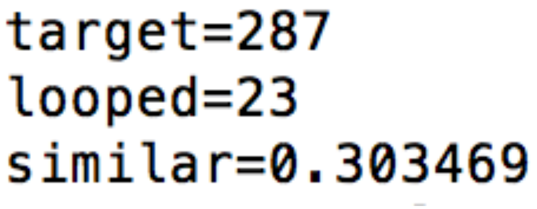
Error function:

e_{ij} (x_{i} , x_{j}) = t2v(Z_{ij}^{-1}*(T_{i}^{-1}*T_{j}))

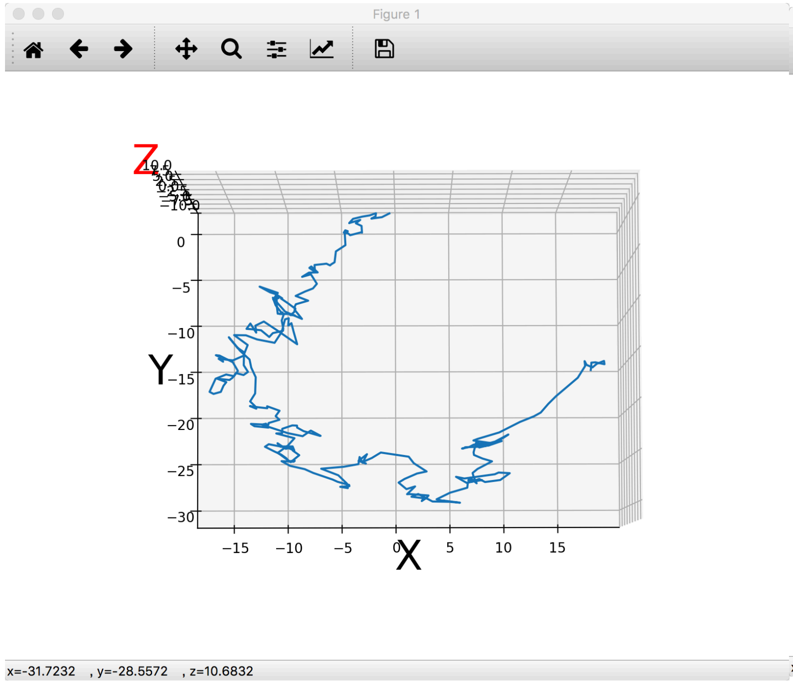
Information matrix Ω_{ij} represent the uncertainty

Implementation

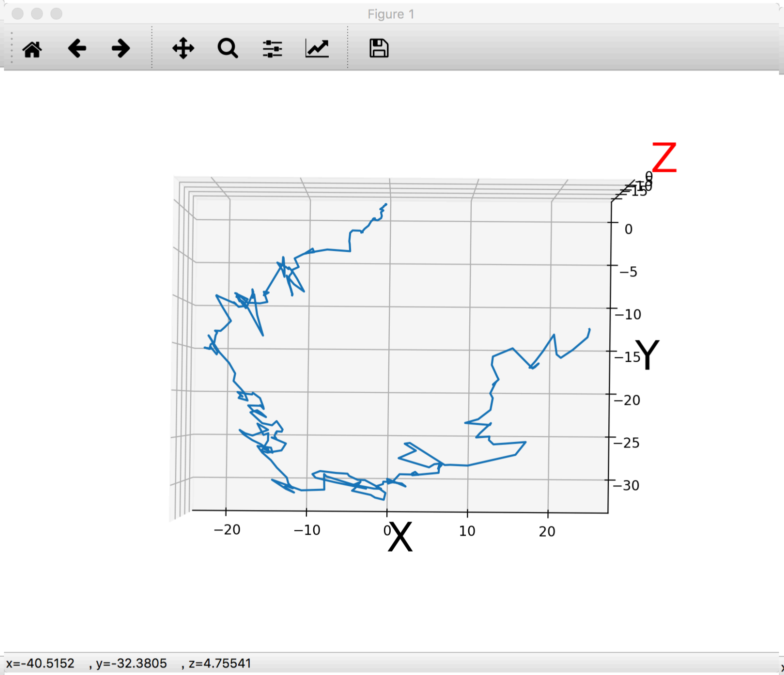
- Based on GTSAM factor graph
- Use GaussNewton Optimizer
- When meet a loop, do once optimization



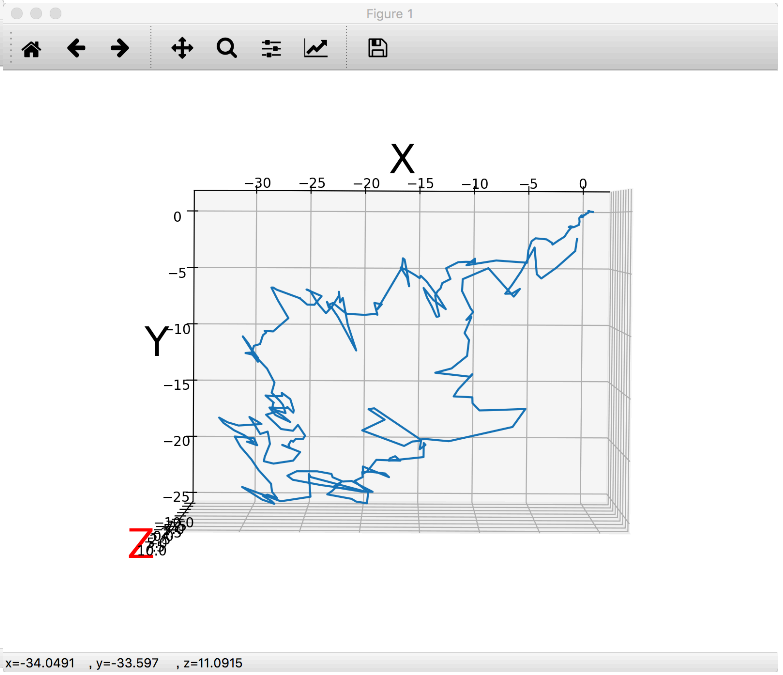
```
target=287
looped=23
similar=0.303469
```



x=-31.7232 , y=-28.5572 , z=10.6832



x=-40.5152 , y=-32.3805 , z=4.75541



x=-34.0491 , y=-33.597 , z=11.0915

Results & Future Works

05

Experiment

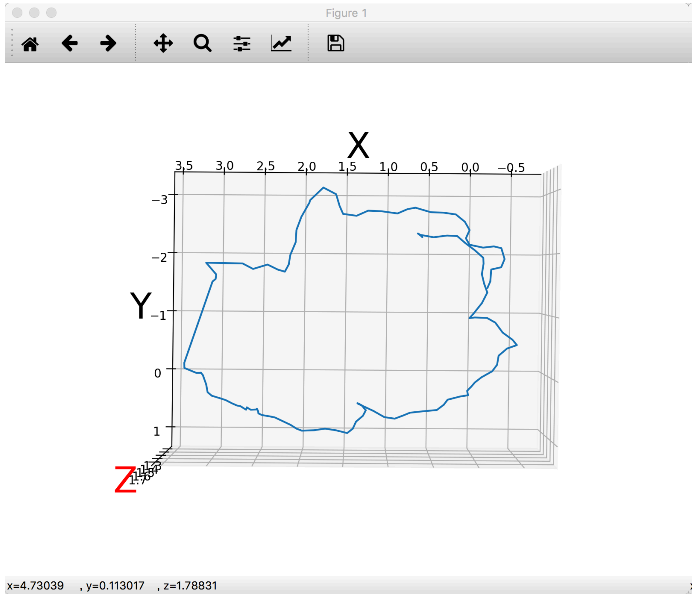
- Test the system on the tum **Sequence 'freiburg2_desk’** dataset
- Dependency: Eigen, Opencv, Boost, Ceres, GTSAM

Results

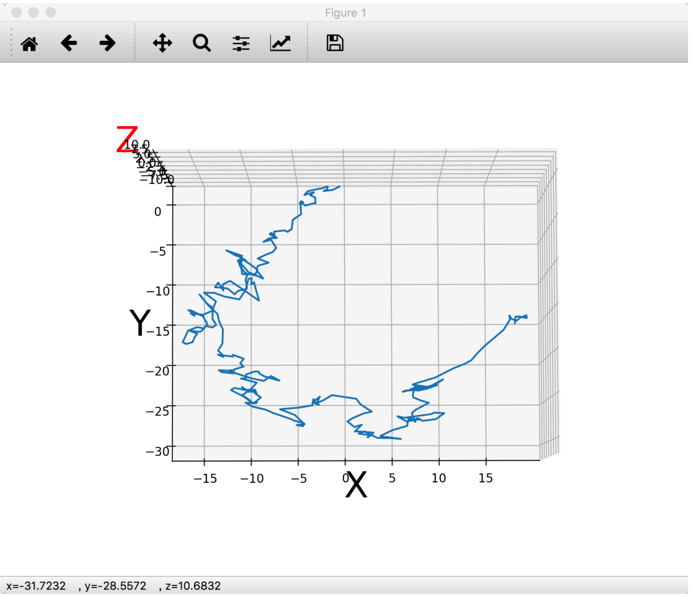
- Ground Truth
- After VO

  -RMSE = 0.14339
- After bundle adjustment

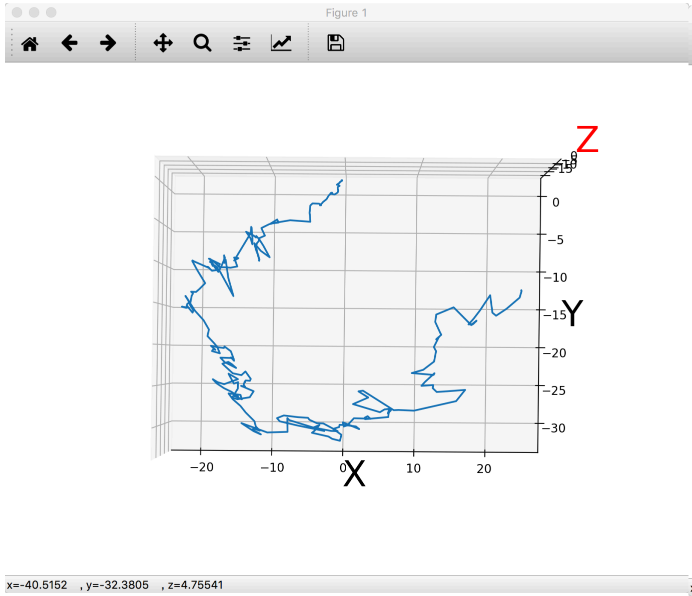
  -RMSE = 0.06041
- After loop closure



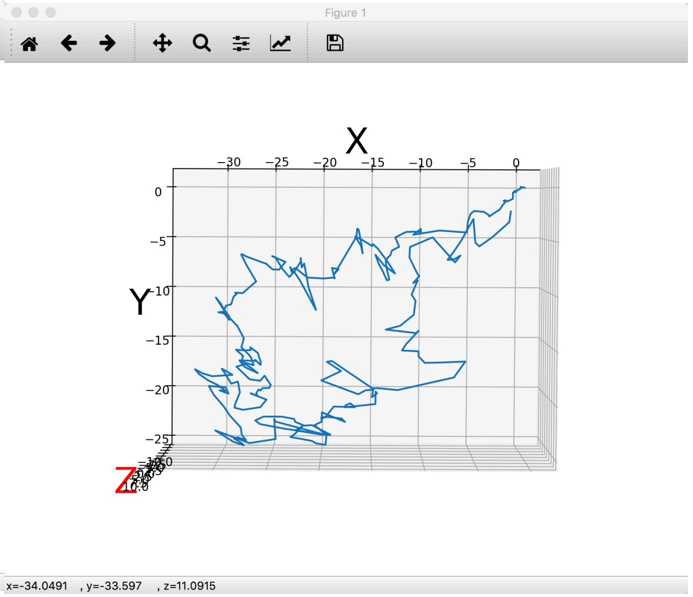
x=4.73039 , y=0.113017 , z=1.78831



x=-31.7232 , y=-28.5572 , z=10.6832



x=-40.5152 , y=-32.3805 , z=4.75541



x=-34.0491 , y=-33.597 , z=11.0915

Results

Doxygen:
file:///Users/ZilinSi/Desktop/CPP/CPP-Final-Project/Doxygen%20File/html/index.html

Future Work

- Implement parallel computing of front-end and back-end to realize real time calculation
- Improve the accuracy by extending the system to sensor fusion or stereo camera.

Thanks!

And Q & A