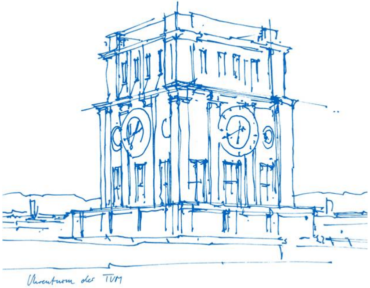
Computer Vision II: Multiple View Geometry (IN2228)

Chapter 14 SfM and SLAM

Dr. Haoang Li

19 April 2023 12:00-13:30



Announcement Before Class

- Updated Question Type of Exam

Originally, our exam consists of 22 multiple-choice and 3 calculation questions.

Based on the advice of some experienced lectures, **we adjusted the question form**.

1. We changed some multiple-choice questions into **short answer questions**.
2. We removed some numerical calculation. Accordingly, for some questions, students are no longer required to provide a specific scalar/vector as the solution. Instead, students are only required to **describe the algorithm pipeline** and **provide necessary formulas**.

Announcement Before Class

- Updated Question Type of Exam

Accordingly, our summer semester exam consists of **12** multiple-choice questions (1-4 correct answers for each question), as well as **5** calculation/short answer questions. Each calculation/short answer question may contain some sub-questions.

Note: knowledge review scope remain unchanged.

The questions of the winter semester exam have not been completed. I will update you in time. In theory, they should be aligned to the questions of the summer semester exam.

Announcement Before Class

- Knowledge Review Document for Chapter 14

Last week, I have uploaded a document for Chapters 11-13.
After today's class, I will update that document by adding the content about Chapter 14.

Today's Outline

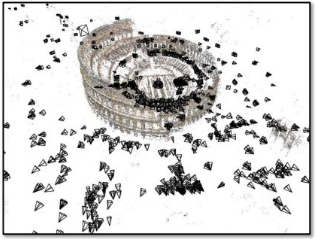
- Structure from Motion (SfM)
- Simultaneous Localization and Mapping (SLAM)

Structure from Motion (SfM)

➢ Definition

✓ Structure from motion (SfM) is the process of estimating the 3-D structure of a scene from a set of 2-D images.

✓ Intuitively, given many images, how can we 1) figure out where they were taken from? 2) build a 3D model of the scene?



Structure from Motion (SfM)

➢ Definition

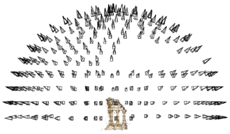
✓ Input

images with feature correspondences

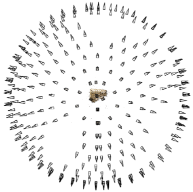
✓ Output

Structure: 3D positions of features

Motion: camera parameters including rotation and translation. Intrinsic parameters are optional.



Reconstruction (side)

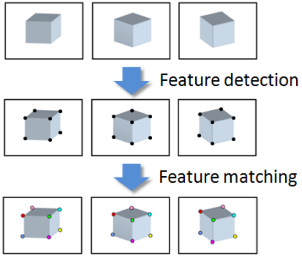


(top)

Structure from Motion (SfM)

➢ Basic Techniques

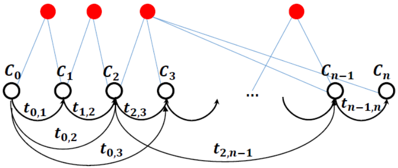
✓ Feature detection and feature matching
Feature correspondences are the basis



✓ Camera pose estimation
Epipolar geometry
Perspective-n-points
…

✓ 3D point reconstruction
Triangulation

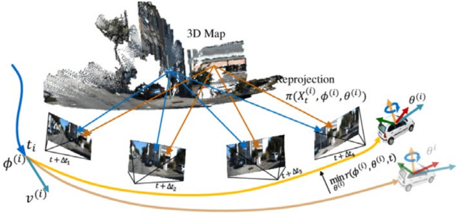
Joint optimization based on bundle adjustment



Structure from Motion (SfM)

➢ Two Types of SfM

✓ Hierarchical SfM [1]
- It takes a set of disordered images as input.
- It estimates the 3D structure and camera poses **in a bottom-up way**.



✓ Sequential SfM [2]
- It takes sequential images as input. It is equivalent to visual odometry (VO).
- It **incrementally** estimates the camera poses and 3D structure.

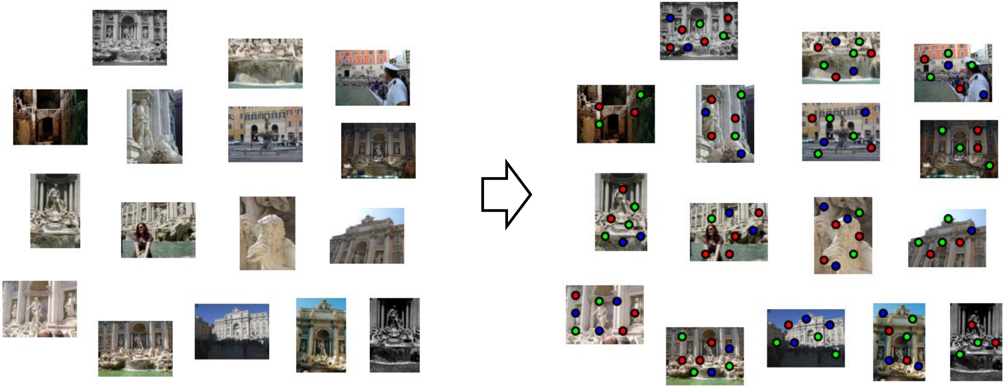
[1] S. Agarwal, N. Snavely, I. Simon, S. M. Seitz and R. Szeliski, "Building Rome in a day," IEEE International Conference on Computer Vision, 2009
[2] Nister, D; Naroditsky, O.; Bergen, J., “Visual Odometry,” IEEE Conference on Computer Vision and Pattern Recognition, 2004.

Structure from Motion (SfM)

- ➢ Hierarchical SfM

✓ Detect features of disordered images

Detection in each image is independent.

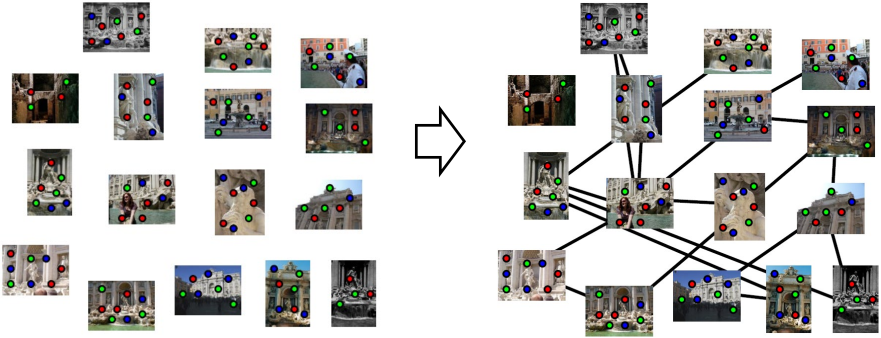


Structure from Motion (SfM)

➢ Hierarchical SfM

✓ Match features between disordered images

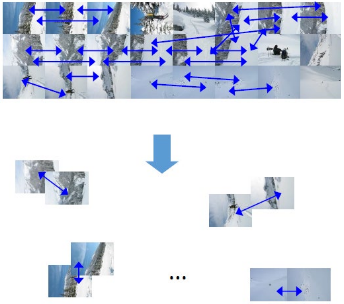
A straightforward strategy is to try all the potential pairs. There are some acceleration strategies (not introduced in our lecture).

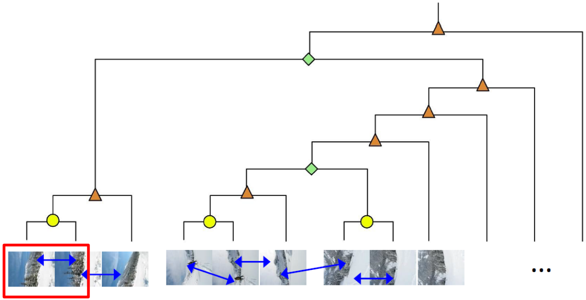


Structure from Motion (SfM)

➤ Hierarchical SfM

✓ Build numerous clusters consisting of **close frames**
✓ Generate a topological tree based on the number of matches. We skip the details.
✓ Start from the terminal nodes to perform two-view SfM



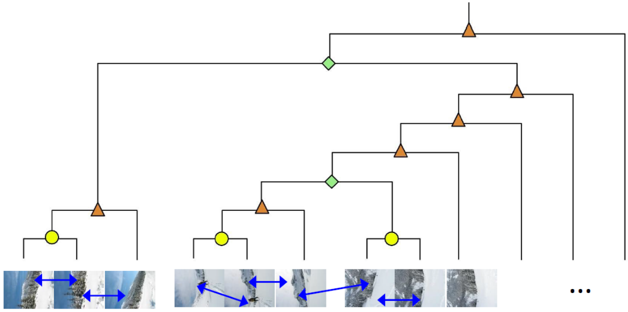


Structure from Motion (SfM)

➢ Hierarchical SfM

✓ Generate a local model based on 2D-2D geometry and 3D-2D geometry
✓ Merge different models based on 3D-3D geometry

The circle ○ corresponds to the creation of a stereo-model, the triangle △ corresponds to applying PNP, the diamond ◇ corresponds to a fusion of two partial independent models.

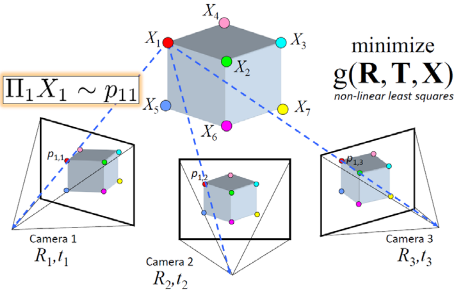


Structure from Motion (SfM)

➢ Hierarchical SfM

✓ Re-projection minimization

Jointly optimize camera poses and 3D points



Number of 3D points

Number of cameras

$$g(\mathbf{X},\mathbf{R},\mathbf{T}) = \sum_{i=1}^{m}\sum_{j=1}^{n} \underbrace{\boxed{w_{ij}}}_{\substack{\textit{indicator variable}\text{:}\\ \text{is point } i \text{ visible in image } j\text{ ?}}} \cdot \left\| \underbrace{\mathbf{P}(\mathbf{x}_i,\mathbf{R}_j,\mathbf{t}_j)}_{\substack{\textit{predicted}\\ \text{image location}}} - \underbrace{\begin{bmatrix} u_{i,j} \\ v_{i,j} \end{bmatrix}}_{\substack{\textit{observed}\\ \text{image location}}} \right\|^2$$

Joint estimation using non-linear least-squares optimization

Structure from Motion (SfM)

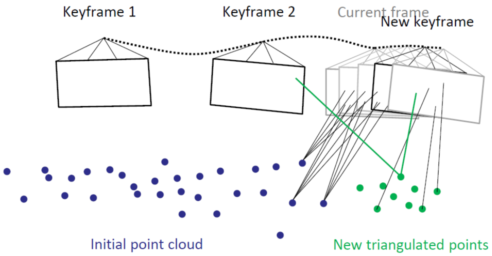
➢ Hierarchical SfM

✓ Some representative results

S. Agarwal, N. Snavely, I. Simon, S. M. Seitz and R. Szeliski, "Building Rome in a day," IEEE International Conference on Computer Vision, 2009

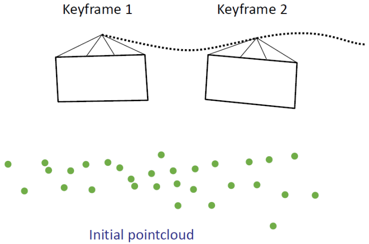
Structure from Motion (SfM)

➢ Sequential SfM



It is equivalent to visual odometry (introduced in Chapter 10)

✓ Step 1: Initialize the structure and motion from 2 views



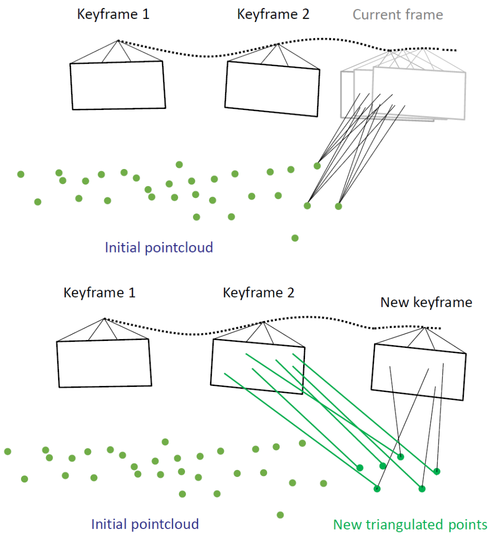
Structure from Motion (SfM)

➢ Sequential SfM

✓ Step 2: Absolute pose estimation from 3D-2D point correspondences.

✓ Step 3: Triangulation to increment 3D map.

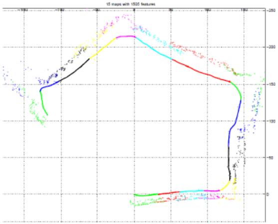
✓ Step 4: Refine structure and motion through bundle adjustment.



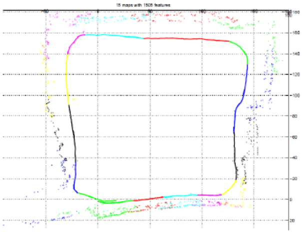
Simultaneous Localization and Mapping (SLAM)

➢ Definition

✓ SLAM can be treated as the combination of visual odometry (VO) and loop closure.
VO is prone to be affected by noise, and thus leads to drift error over time.
SLAM guarantees global consistency based on loop closure.



Visual odometry



Visual SLAM

Simultaneous Localization and Mapping (SLAM)

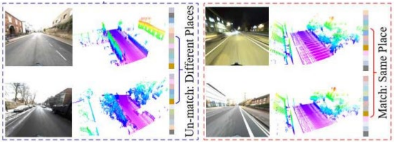
➢ Loop Closure

✓ Loop closure consists of two steps

Loop detection

Loop correction

Image-based

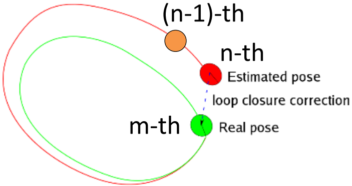


Point cloud-based

Loop detection

$$\mathrm{Sim}(3) \qquad \mathbf{T}_S = \begin{bmatrix} s\mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix}$$

7 degrees of freedom (DOF)

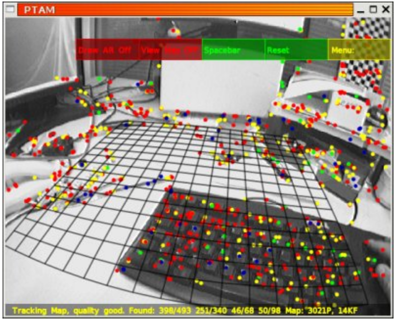


Loop correction: we use the estimated m-th pose and computed Sim(3) to update the n-th pose. Then we use the estimated **relative** pose to update (n-1)-th pose.

Simultaneous Localization and Mapping (SLAM)

➢ Representative SLAM Methods

✓ PTAM: Parallel Tracking and Mapping



Monocular only

Feature-based

- FAST corner detection (introduced later)
- Minimizes re-projection error
- Jointly optimizes poses & structure (sliding window-based bundle adjustment)

First method to introduce the concept of **keyframe**

Klein, Murray, Parallel Tracking and Mapping for Small AR Workspaces, International Symposium on Mixed and Augmented Reality (ISMAR),

Simultaneous Localization and Mapping (SLAM)

➢ Representative SLAM Methods

✓ PTAM: Parallel Tracking and Mapping

First method to run two modules, i.e., localization and mapping in two **independent threads**.

Real-time (30Hz). However, the global optimization is not done in real time but asynchronously.

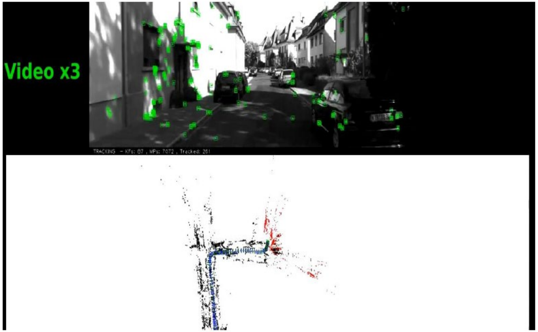
Limitation:

- Relocalization (a technique to solve the failure of camera tracking) only in a small neighborhood
- No global optimization

Simultaneous Localization and Mapping (SLAM)

➢ Representative SLAM Methods

✓ ORB-SLAM



Supports both monocular and stereo cameras

Feature-based

- ORB feature that is very fast to compute and match (introduced later)
- Minimizes re-projection error
- Jointly optimizes poses and structure (sliding-window bundle adjustment)

MurArtal, Montiel, Tardos, ORB SLAM: Large scale Feature based SLAM, IEEE Transactions on Robotics (T-RO), 2015.

Simultaneous Localization and Mapping (SLAM)

➢ Representative SLAM Methods

✓ ORB-SLAM

Same workflow as PTAM (keyframe-based, parallel localization and mapping threads).

Includes:

- Re-localization in larger scale
- Final global optimization

Efficiency: Real-time (30Hz). However, the global optimization is not done in real time but asynchronously.

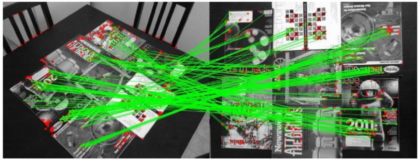
Simultaneous Localization and Mapping (SLAM)

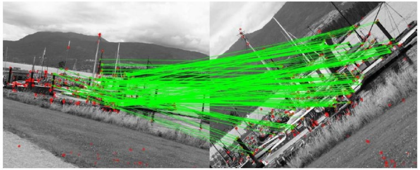
➢ Supplementary Knowledge

✓ ORB feature: Oriented FAST and Rotated BRIEF

Keypoint detector is based on the variant of FAST algorithm.
Binary descriptor is based on the variant of BRIEF descriptor.

In our lecture, we only briefly introduce the original FAST and BRIEF.





Simultaneous Localization and Mapping (SLAM)

➢ Supplementary Knowledge

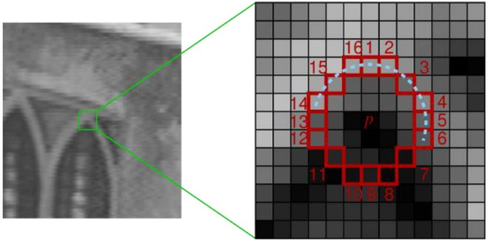
✓ FAST corner detector: **F**eatures from **A**ccelerated **S**egment **T**est

Analyse intensities along a ring of 16 pixels centered on the pixel of interest $\boldsymbol{p}$

$\boldsymbol{p}$ is a FAST corner if a set of N contiguous pixels on the ring are :

- all brighter than the pixel intensity $\boldsymbol{I}(\boldsymbol{p})+\boldsymbol{threshold}$,
- or all darker than $\boldsymbol{I}(\boldsymbol{p})-\boldsymbol{threshold}$

Common value of N: 12



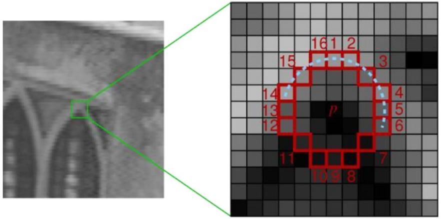
Simultaneous Localization and Mapping (SLAM)

➢ Supplementary Knowledge

✓ FAST corner detector: **F**eatures from **A**ccelerated **S**egment **T**est

FAST can be treated as a simple classifier to check the quality of corners and reject the weak ones.

FAST is the fastest corner detector ever made: can process 100 million pixels per second.

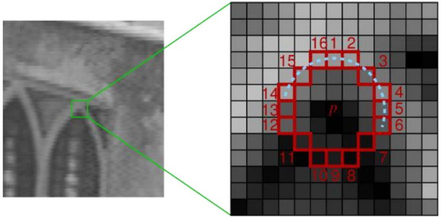


Simultaneous Localization and Mapping (SLAM)

➢ Supplementary Knowledge

✓ FAST corner detector: **F**eatures from **A**ccelerated **S**egment **T**est

Issue: it is very sensitive to image noise (large noise in weak illumination). This is why Harris is still more common despite a bit slower.



Simultaneous Localization and Mapping (SLAM)

➢ Supplementary Knowledge

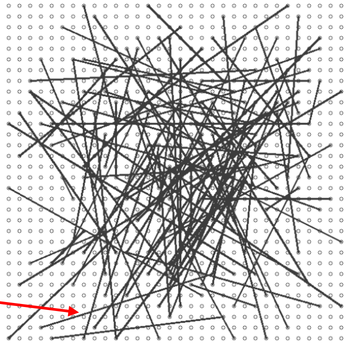
✓ BRIEF descriptor: Binary Robust Independent Elementary Features

Goal: high-speed descriptor computation and matching

Binary descriptor formation:

- Smooth image
- **for each** detected keypoint (e.g. FAST),
- ***sample*** 128 intensity pairs $(p_1{}^i, p_2{}^i)$ $(i = 1, \dots, 128)$ within a squared patch around the keypoint
- Create an empty 128-element descriptor
- **for each** i^{th}**pair**
  - **if** $I_{p_1{}^i} < I_{p_2{}^i}$ **then *set*** i^{th} bit of descriptor to **1**
  - **else** to **0**

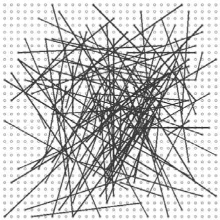
A pair



Pattern for intensity pair samples – generated randomly

Simultaneous Localization and Mapping (SLAM)

➢ Supplementary Knowledge



Pattern for intensity pair samples – generated randomly

✓ BRIEF descriptor: Binary Robust Independent Elementary Features

The pattern is generated randomly only once. Then, the same pattern is used for all the patches.

Pros: Binary descriptor: allows very fast Hamming distance matching (count of the number of bits that are different in the descriptors matched)

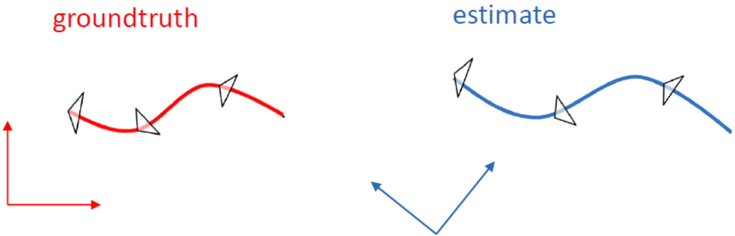
Cons: Not scale/rotation invariant

Simultaneous Localization and Mapping (SLAM)

➢ Evaluation Metrics

- ✓ Main idea: compare the **estimated** trajectory with the **ground truth** trajectory (obtained by GPS or motion tracking systems).
- ✓ The key question is what error metric we should use.

- ✓ Challenges
  - Different coordinate systems
  - Different scales



Simultaneous Localization and Mapping (SLAM)

➢ Evaluation Metrics

Naive but ineffective strategy is to align the first poses and measure the error of the final pose.

✓ Not repeatable:
Most SLAM methods are non-deterministic (e.g., RANSAC and multi-threading). Every time you run them on the same dataset, you get different results.

✓ Not very meaningful:
The error of the final pose is sensitive to the trajectory shape.
We can hardly obtain the statistical information of error.

Simultaneous Localization and Mapping (SLAM)

➢ Evaluation Metrics

✓ A Representative metric: absolute trajectory error (ATE)

Step 1: align the estimated trajectory to the ground truth from the start to the end using a **similarity transformation** (i.e., R, T, s) by minimizing the sum of square position errors.

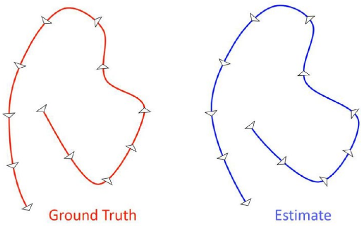
groundtruth positions

$$R, T, s = \underset{R,T,s}{\operatorname{argmin}} \sum_{k=0}^{n} \left\| \hat{C}_k - sRC_k - T \right\|^2$$

estimated positions

Parameters of the similarity transformation that we want to find

This can be solved based on **Horn's method or Umeyama's method** (we mentioned them in the Chapter of 3D-3D geometry)



3D-3D point correspondences are obtained by **timestamp alignment**

Simultaneous Localization and Mapping (SLAM)

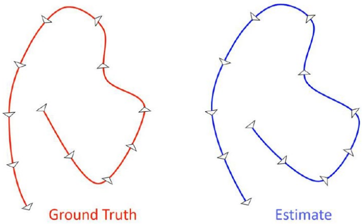
➢ Evaluation Metrics

✓ A representative metric: absolute trajectory error (ATE)

Step 2: compute the root mean square error (RMSE) after alignment:

$$RMSE = \sqrt{\frac{\sum_{k=1}^{n} \left\| \hat{C}_k - sRC_k - T \right\|^2}{n}}$$

3D-3D point correspondences are obtained by **timestamp alignment**



Pros and cons

- Single number metric
- Captures the global error
- Does not encode the relative error

Summary

- Structure from Motion (SfM)
- Simultaneous Localization and Mapping (SLAM)

Thank you for your listening!
If you have any questions, please come to me :-)