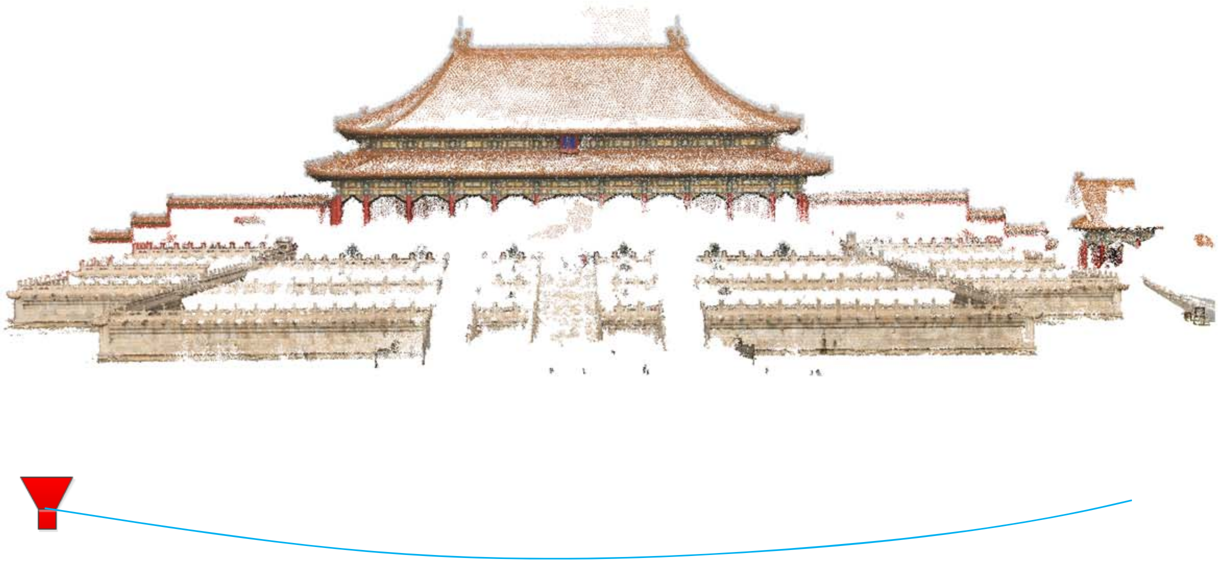
12. Simultaneous Localization And Mapping (SLAM)



Outline

- LiDAR SLAM
- Visual SLAM
- Robustness Techniques

SFU

ZHEJIANG UNIVERSITY

SLAM

- Simultaneous Localization and Mapping
  - Localization: estimating the sensor's pose (location and orientation)
  - Mapping: building a map
  - SLAM: building a map and locating the sensor at the same time
- A chicken-and-egg problem
  - A map is needed for localization
  - A pose estimate is needed for mapping

Visual SLAM Demos

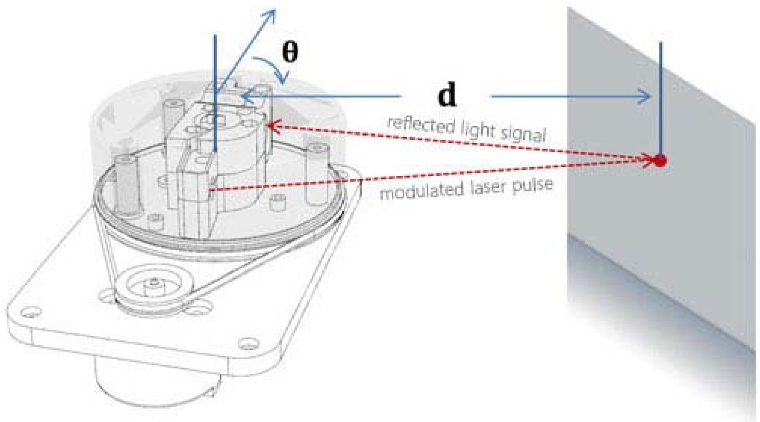
Visual SLAM Applications

Augmented Reality

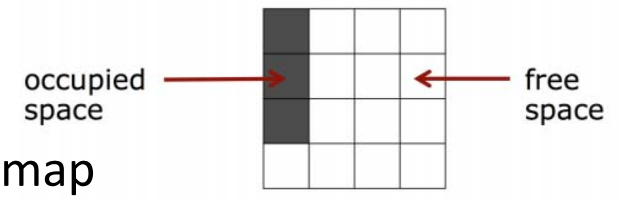
Microsoft Hololens

2D LiDAR SLAM

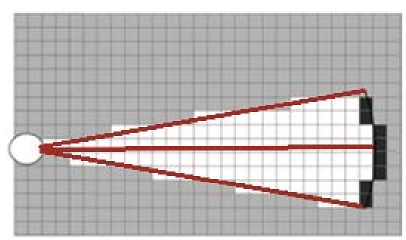
- Widely used for ground robots
- Use a 2D LiDAR sensor to track planar motion
- 2D LiDAR sensor
  - With a rotating laser beam
  - Return distances to obstacles in a plane
  - E.g. 10×360 measurements per second

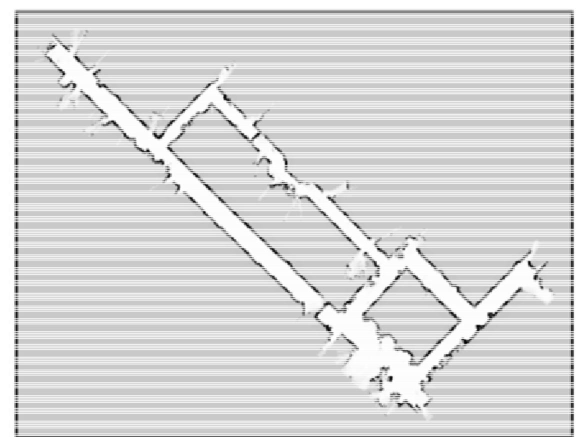


Map Representation (Occupancy Grid Map)



- Discretize the environment by a grid
  - E.g. 10 m × 20 m space, 5 cm resolution → 200 × 400 map
  - Large maps require substantial memory resources
- Each grid cell can be empty, occupied, or unknown
  - E.g. white is empty, black is occupied, and grey is unknown
- Each laser beam tells the occupancy of some cells
  - Mark empty for cells on its path
  - Mark occupied for the cell at its end





SFU

浙江大学 ZHEJIANG UNIVERSITY

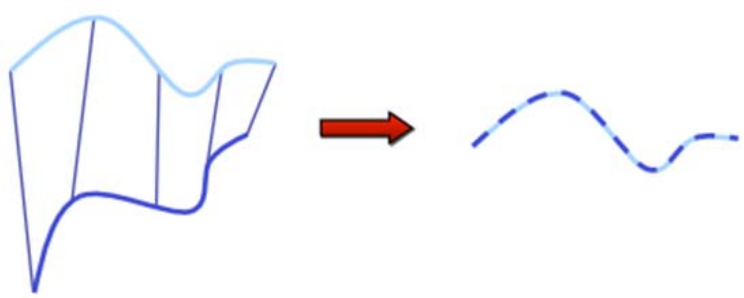
Mapping with Known Poses

- Suppose the robot pose is known at all time
  - E.g. by some SLAM algorithms
- Accumulate empty/occupied votes from the LiDAR sensor over time
  - A cell is occupied, if the number of occupied votes is larger (by a threshold)
  - A cell is empty, if the number of empty votes is larger (by a threshold)
  - Otherwise, a cell is unknown

A sample map built from known poses along the red trajectory

Pose Estimation

- Find the sensor pose according to its scan and a map
- It might be solved by the ICP (iterative closest point) algorithm
  - A widely used algorithm to register two sets of points
- ICP iterates the following two steps till converge
  - Find correspondence as nearest neighbors
  - Solve sensor motion from the found correspondences



Registration with Known Correspondence

- Given two sets of corresponding points

$$X = \{x_1, x_2, \ldots, x_n\} \text{ and } P = \{p_1, p_2, \ldots, p_n\}$$

- Want: translation t and rotation R that register these two sets
- Mathematically, that means to minimize the following error

$$E(R, t) = \sum_{i=1}^{n} |x_i - Rp_i - t|^2$$

- A closed-form solution can be derived easily (try it yourself)

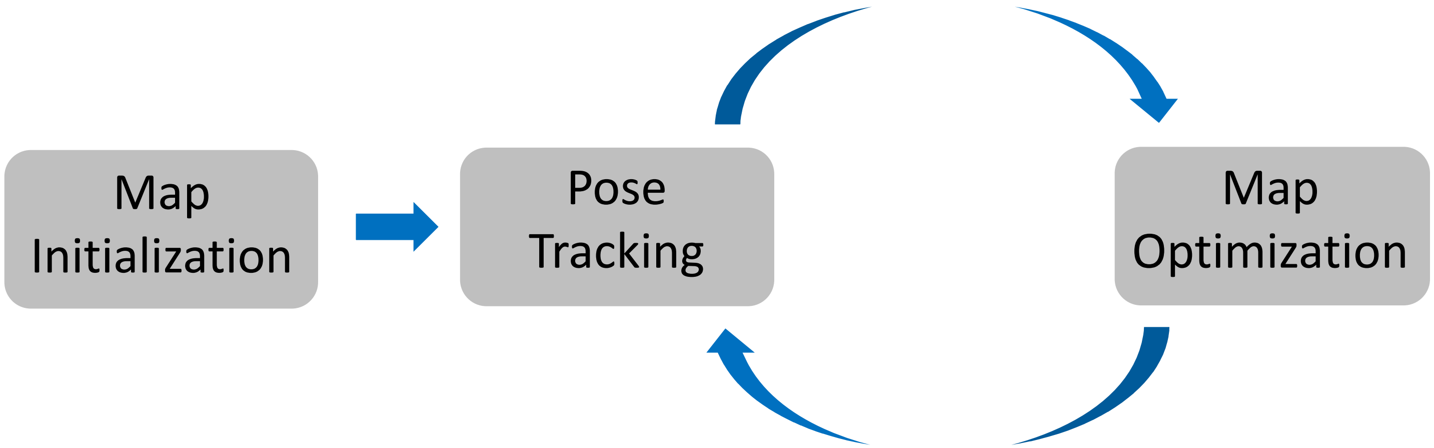
SFU

浙江大学 ZHEJIANG UNIVERSITY

2D LiDAR SLAM Summary

- Initialize at t=0
  - The raw LiDAR scan is the initial map
- Start from t=1, iterate the following to steps:
  - Solve sensor pose at time t by the ICP algorithm
  - Update map according to the new scan at time t

Typical SLAM Systems Architecture



- LiDAR SLAM
  - Initialization: the first scan
  - Pose Tracking: ICP
  - Map Optimization: occupancy grid map
- Visual SLAM
  - Initialization: (essential matrix, triangulation, etc)
  - Pose Tracking:
    1. Feature tracking (next-next week)
    2. Pose-only BA (today)
  - Map Optimization:
    1. Triangulation, BA
    2. Loop closure, pose-graph (today)

SFU

ZHEJIANG UNIVERSITY

Questions?

Outline

- LiDAR SLAM
- Visual SLAM
- Robustness Techniques

Visual SLAM by SfM

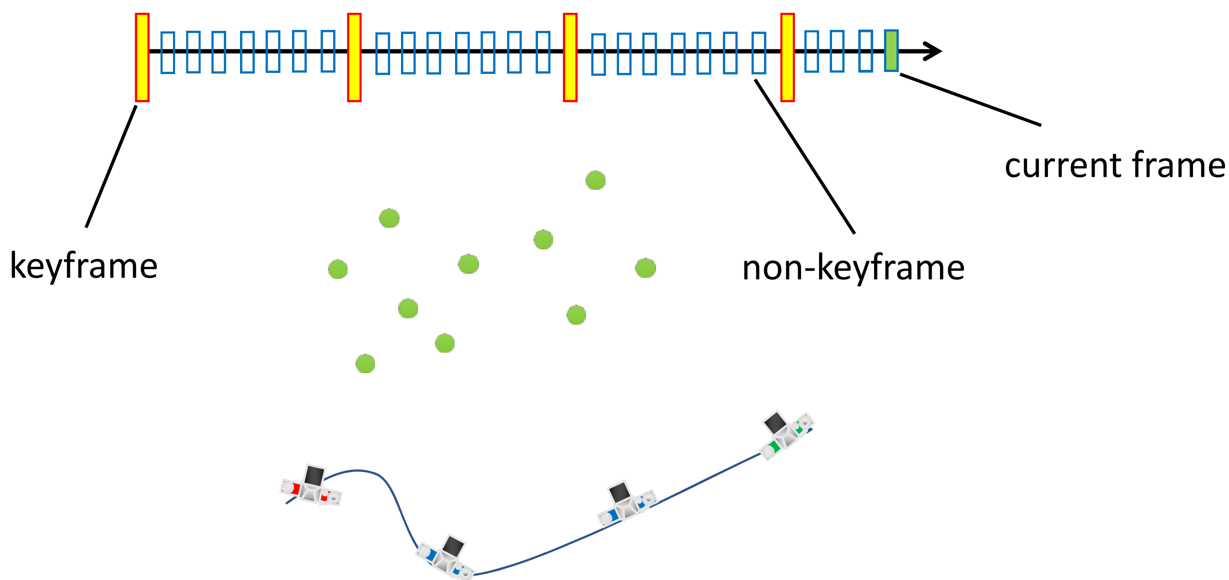
- Solve an incremental SfM at every new frame

- Realtime constraint (many trade-offs for better efficiency)
  - Keyframe based mapping (only a subset of frames are used for mapping)
  - Local BA (bundle adjustment with only nearby video frames)

- Sequential video input
  - Sorted input images (match each image to its previous frame)
  - Regular time interval between frames (motion model to facilitate matching)

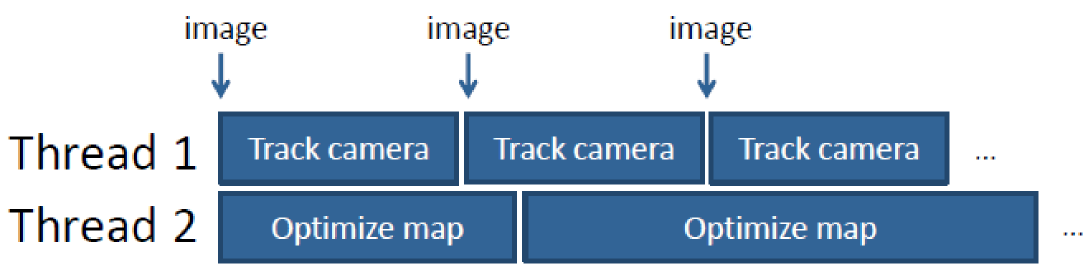
Key-frame based Visual SLAM

- Solving camera pose for every input frame
- Only use some “keyframes” to triangulate/optimize map points



Parallel Tracking and Mapping (PTAM)

- Parallel tracking and mapping
  - A real time tracking thread runs in real-time (30Hz)
  - An offline mapping thread for map maintenance



Parallel Tracking and Mapping for Small AR Workspaces

Georg Klein* David Murray†

Active Vision Laboratory
Department of Engineering Science
University of Oxford

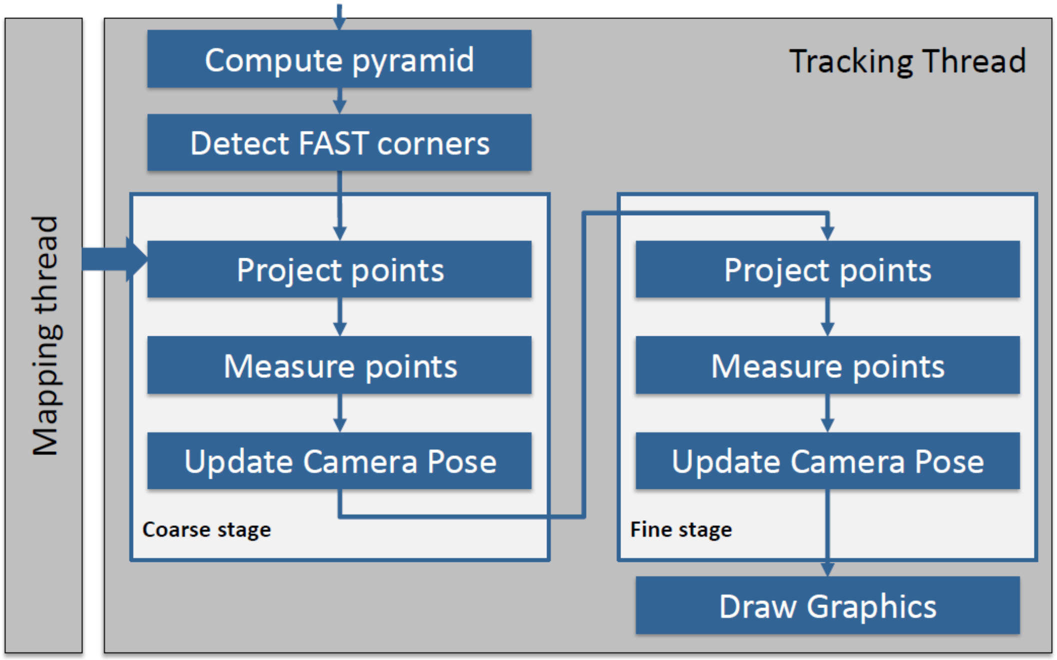
[ISMAR 2007]

SFU

ZHEJIANG UNIVERSITY

Tracking

- Step 1: feature correspondence
  - Option 1: KLT feature tracking (next week)
  - Option 2: feature detection & matching (within a nearby neighborhood)



[PTAM 2007]

SFU

浙江大學 ZHEJIANG UNIVERSITY

feature correspondence

- Generate 8x8 matching template (warped from keyframe)
- Search for correspondence in a fixed radius around projected position
  - Using SSD
  - Only search at pre-detected corner points (e.g. FAST points)

Tracking

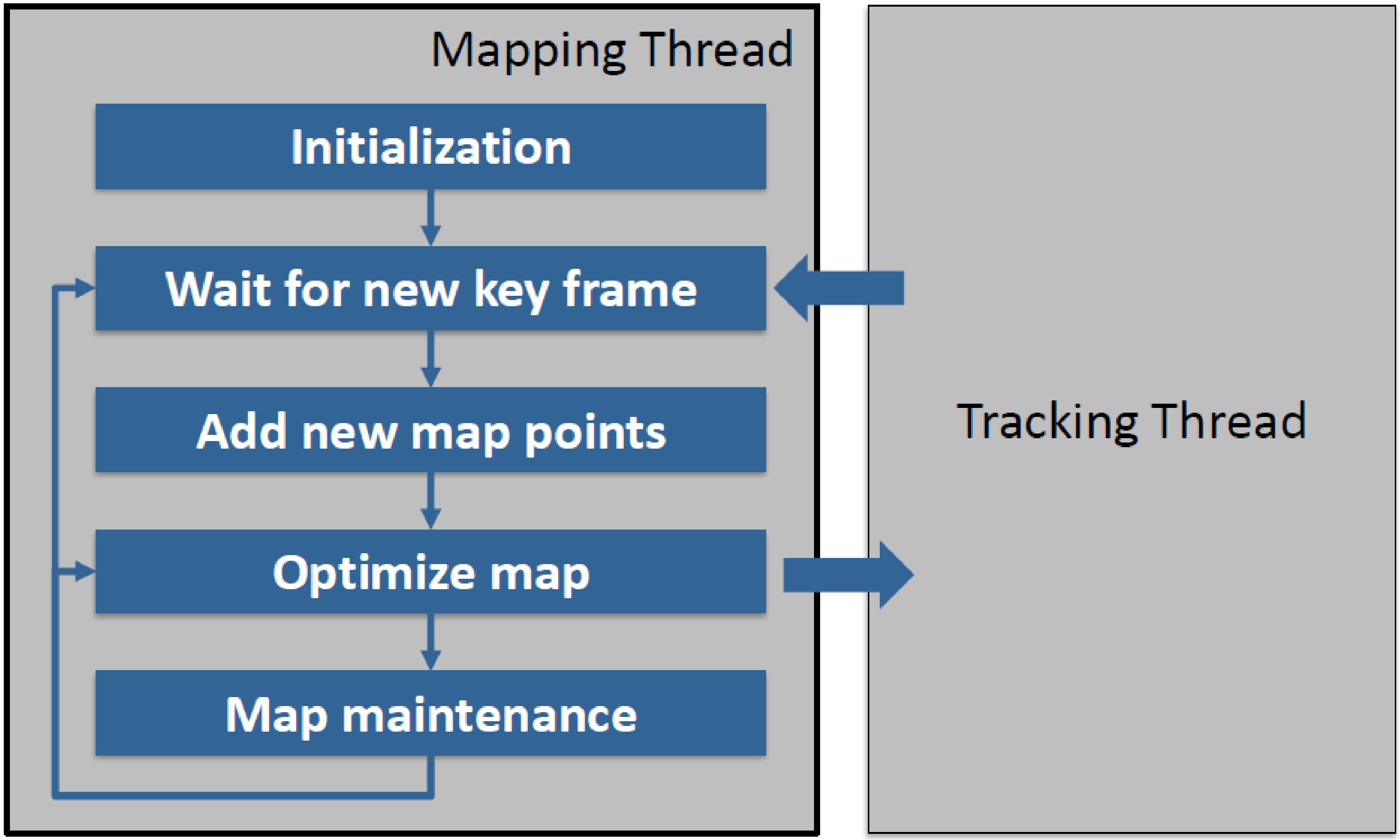
- Step 2: solve camera motion
  - With camera pose initialized by extrapolation, e.g. constant velocity motion
  - With 3D map points fixed
  - Perform local camera only BA (BA with map points fixed)

  - Typically, use a robust cost function ρ on the re-projection error
  - Camera might also be initialized by PnP

Mapping

- Triggered by the insertion of a new keyframe
- Triangulate any tracked image corners that are not reconstructed
- Run corner detection (e.g. Harris) to generate more points for tracking
- Call local BA to optimize both points and poses
  - BA is slow, may be called after several keyframe insertions

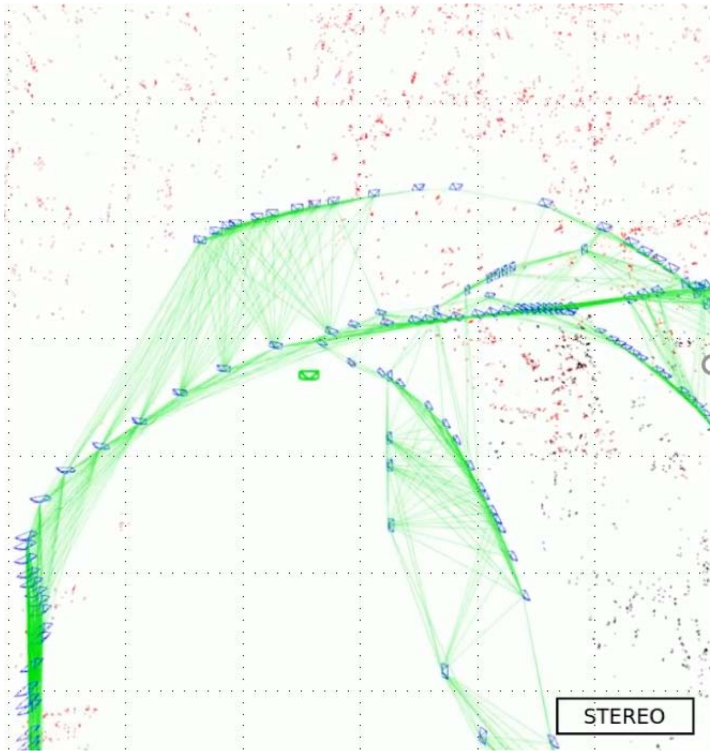
Mapping



[PTAM 2007]

Keyframes

- Defined heuristically, various from system to system
- A keyframe is typically inserted when:
  - Camera tracking is robust
  - There are insufficient corner points to track
  - There is a large camera motion (rotation or translation
  - There is no keyframe inserted for a while (e.g. more than 30 frames)
  - Etc..



PTAM – Example Timings

- Tracking thread

Total	19.2 ms
Key frame preparation	2.2 ms
Feature Projection	3.5 ms
Patch search	9.8 ms
Iterative pose update	3.7 ms

- Mapping thread

Key frames	2-49	50-99	100-149
Local Bundle Adjustment	170 ms	270 ms	440 ms
Global Bundle Adjustment	380 ms	1.7 s	6.9 s

Questions?

SFU

浙江大學 ZHEJIANG UNIVERSITY

Outline

- LiDAR SLAM
- Visual SLAM
- Robustness Techniques

SFU

浙江大學 ZHEJIANG UNIVERSITY

Re-localization

- Tracking can lose due to various reasons
  - Motion blurs
  - Moving objects
  - Large occlusion
  - Sudden fast motion
  - Sudden illumination change

- Re-localization is to recover from such a sudden tracking failure

Re-localization

- Re-localization typically includes the following steps
  - Image search: search the current frame among the pre-indexed keyframes
    - E.g. by bag-of-words models (next next week)
    - It returns a keyframe with sufficient view overlap with the current frame
  - Feature matching between these two frames
  - PnP and local BA to register the current frame to the map
  - Continue the original SLAM

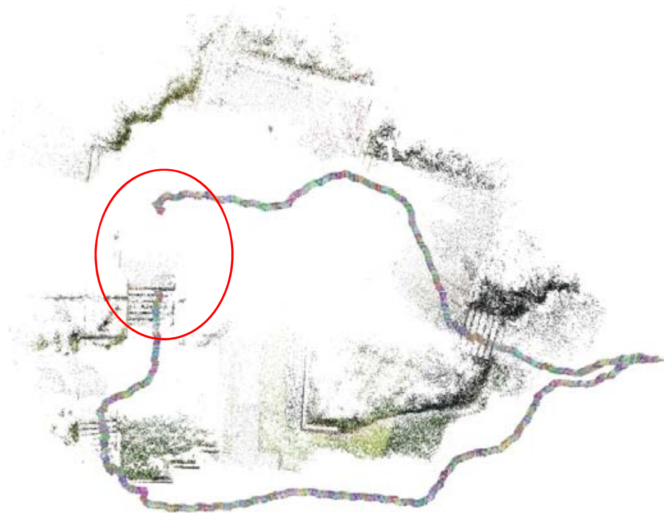
Localization

- The re-localization technique can be used in a different scenario:
  - Solve the mapping beforehand (offline)
  - Solve only the tracking online
    - E.g. solve every frame by re-localization
    - Frame-to-frame constraint might also be included

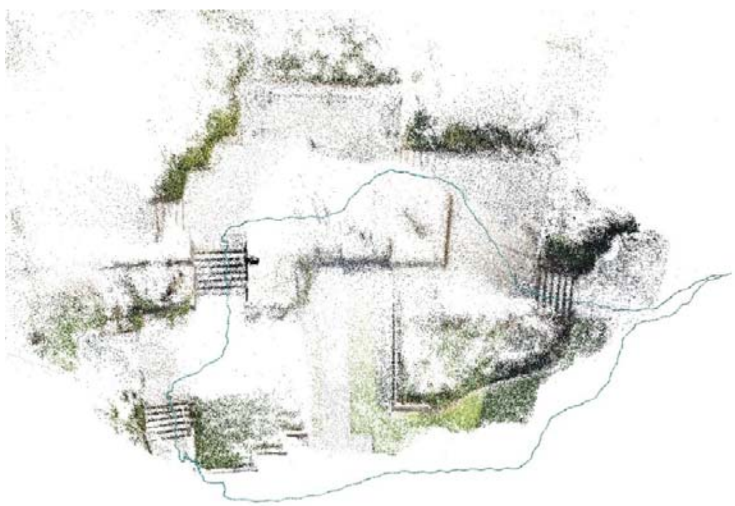
- The advantages of separate mapping and tracking
  - A high quality map is guaranteed
  - The most time consuming step (i.e. global BA) is moved to offline

Drifting

- Small error accumulates to large map distortions
- The technical to reduce drifting is called loop closure



result with drifting error



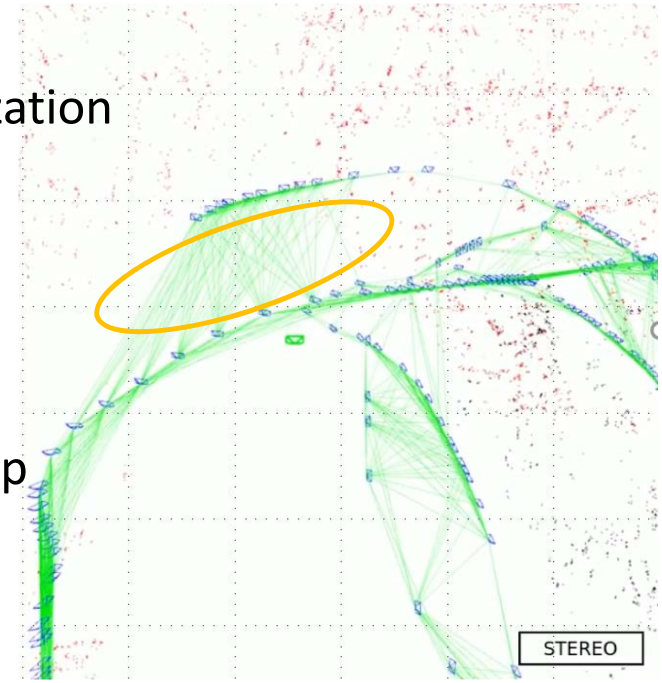
result after loop closure

SFU

浙江大學 ZHEJIANG UNIVERSITY

Loop Closure

- Loop detection
  - Identify if a loop exist
  - Again, by the same image search technical in re-localization
  - Typically, applied to every newly inserted keyframe
- Construct a pose-graph for the next step
  - Where a keyframe is a vertex
  - Two keyframes are connected if they have view overlap



connection due to loop detection

SFU

ZHEJIANG UNIVERSITY

Loop Closure

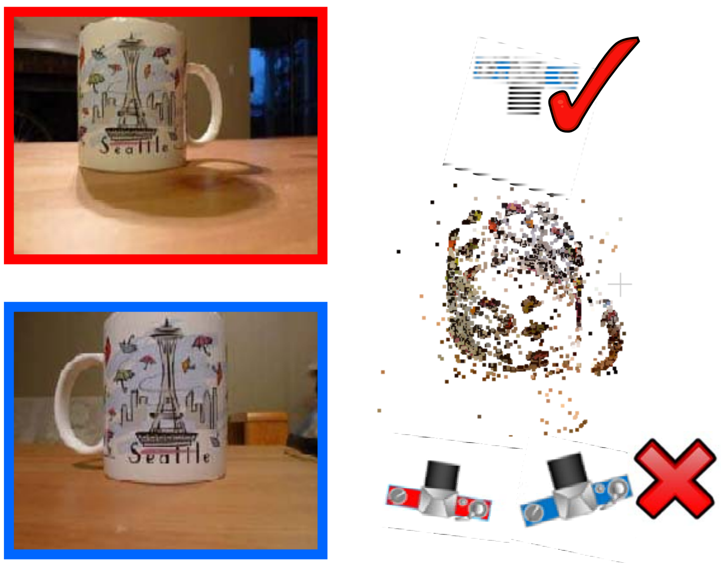
- Loop optimization
  - In the simplest case, a direct BA can generate good result
    - In many cases, the drifting error is too large to be corrected by BA
- Most of the time, a global SfM is desirable
  - Solving all keyframe poses from input pairwise relative motion constraints
  - Update map points afterwards
  - Referred as ‘pose-graph optimization’ in robotics
    - the keyframe graph is a graph with camera poses (no 3D points)
  - The most challenging problem is to deal with wrong loops (due to repetitive structures)
    - Studied in both computer vision and robotics community

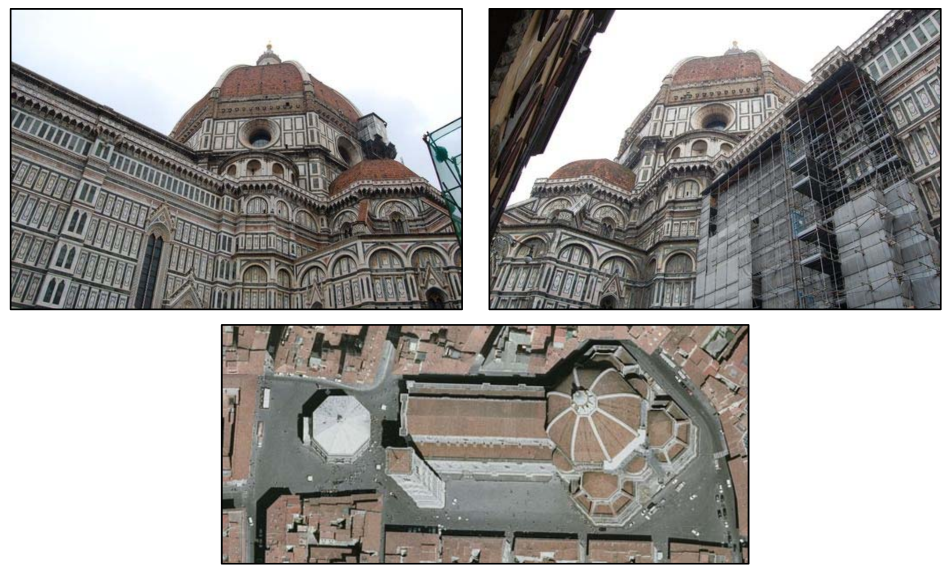
SFU

浙江大學 ZHEJIANG UNIVERSITY

Repetitive Structures

- Repetitive structures cause wrong essential matrices
- This is a problem for both global & incremental SfM
- Essentially, some pose-graph edges should be removed





SFU

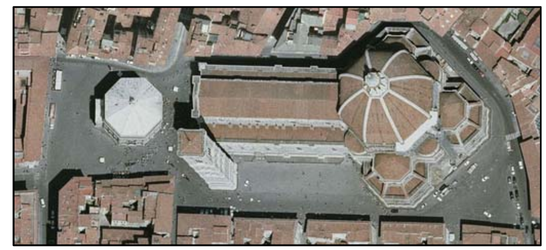
浙江大學 ZHEJIANG UNIVERSITY

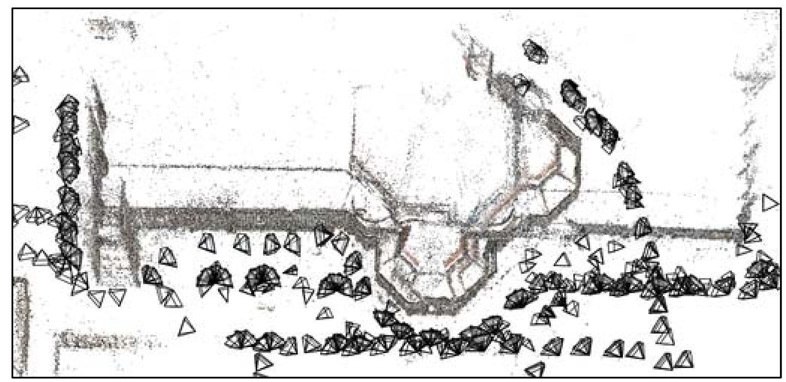
Repetitive Structures

- Repetitive structures could generate catastrophic fails to SfM (or pose-graph optimization)

True reconstruction

False reconstruction





False reconstruction

Questions?

A Brief History of Visual SLAM

- MonoSLAM, Andrew Davison [ICCV 2003] [PAMI 2007]
  - The first work of visual SLAM with a single camera
- Visual Odometry, David Nister [CVPR 2004]
  - Visual slam by SfM
- PTAM, Klein & Murray [ISMAR 2007] (open source)
  - Separating tracking and mapping
- LSD-SLAM, Engel et al. [ECCV 2014] (open source)
  - Direct method
- ORB-SLAM, Mur-Artal et al. [PAMI 2015] (open source)
  - A stronger version than PTAM, with re-localization, pose-graph, etc
- DSO, Engel et al. [PAMI 2017] (open source)
  - A stronger version of LSD-SLAM, with photometric auto-calibration, etc