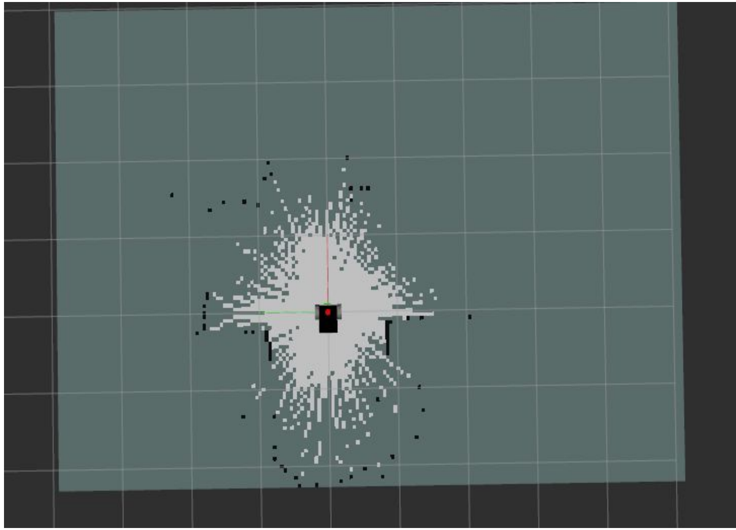
EECE5554 - ROBOT SENSING AND NAVIGATION

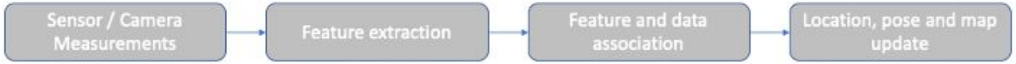
ORB SLAM -3

M.A. Razaq Khan | Tejasri K | M. Irfan Ali | Adnan Amir

What is SLAM?

- SLAM is a process where a mobile robot creates and utilizes a map to determine its location in an environment.
- Both the trajectory of the platform and the location of all landmarks are estimated without a priori knowledge of location.
- SLAM has two parts
    - Localization
    - Mapping





Simplified generic feature-based SLAM process

Lidar SLAM or Visual SLAM?

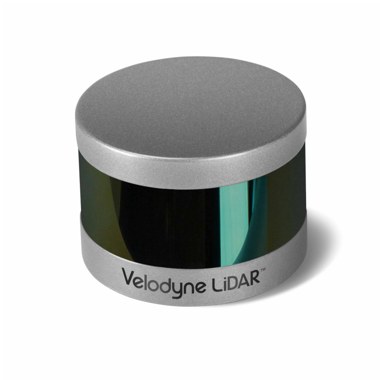
Lidar SLAM Vs Visual SLAM

Accuracy: LIDAR sensor can measure distances with high precision.

Reliability: LIDAR sensors are less affected by environmental factors.

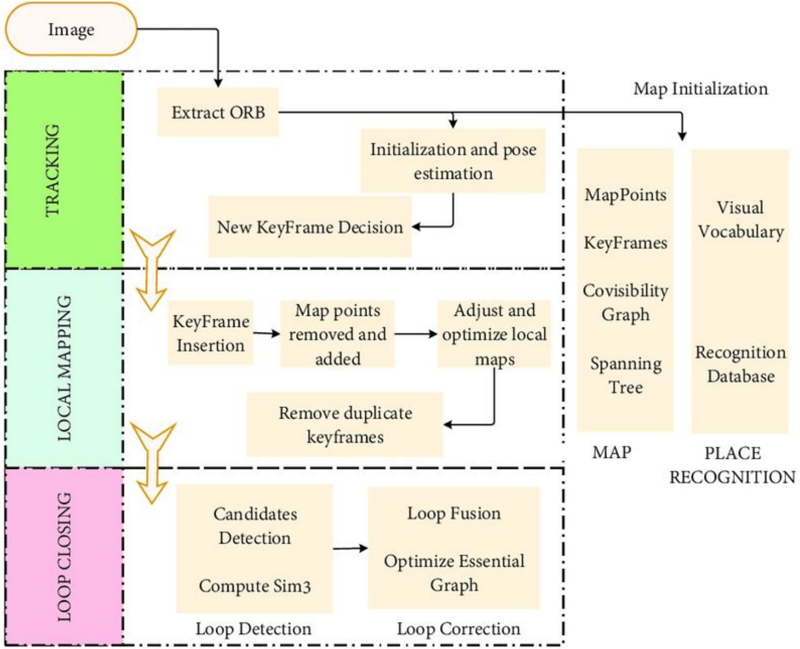
Speed: Visual SLAM is faster than LiDAR SLAM due to quicker image capture, influenced by system computational power and environmental complexity.

Cost: LIDAR sensors are more expensive and require more processing power.



Understanding ORB-SLAM

- ORB-SLAM comes from Oriented FAST and Rotated BRIEF (ORB) features to detect and match keypoints in images.
- Combined techniques, such as loop closing and pose optimization, to achieve robust and accurate localization and mapping.
- The system is divided into three threads
  - Tracking
  - Local mapping
  - Loop closing



ORB-SLAM to ORB-SLAM 3

Key Features: ORB-SLAM

- Accurate Monocular SLAM solution
- ROS is integrated
- Map initialization and tracking using keyframe-based methods.

Key Features: ORB-SLAM 2

- Enhanced support for Monocular, Stereo and RGB-D cameras
- ROS is optional
- Improved tracking and mapping accuracy and optimization.
- Loop closing for global consistency.

Key Features: ORB-SLAM 3

- A monocular and stereo visual-inertial SLAM system
- Improved-recall place recognition
- ORB-SLAM Atlas
- An abstract camera representation

ORB-SLAM 3

- It is a feature-based tightly-integrated visual-inertial SLAM system.
- It is a multiple map system that relies on a new place recognition method with improved recall.
- Reuse in all the algorithm stages all previous information.
  - Short-term data association
  - Mid-term data association
  - Long-term data association
  - Multi-map data association

ORB-SLAM 3

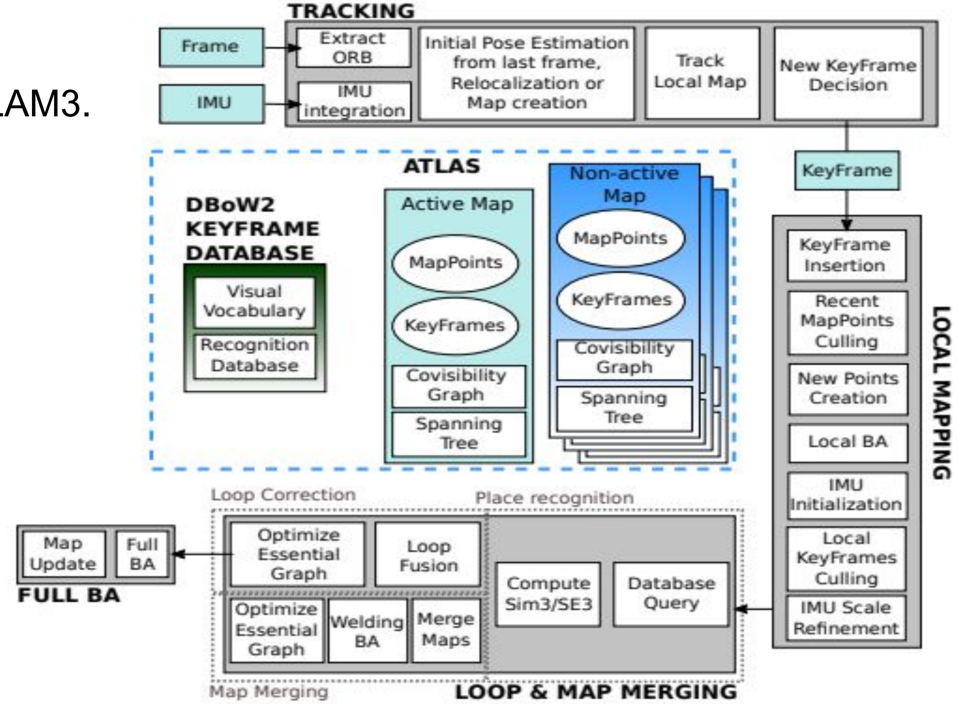
The main techniques used for estimation and data association:

- **Visual SLAM:**
- **Visual-Inertial SLAM:**
- **Multi-Map SLAM:**

SYSTEM FRAMEWORK

Main system components of ORB-SLAM3.

- Tracking
- Mapping
- ATLAS
- Loop Cloosing



Tracking

- The goal is to get Pose (T), Velocity(v) and biases(b)(assumed to be brownian)
- Done using VO, IO and VIO (IMU Initialization)
- In VO
  - Good initial estimate but the map scale is bad
  - Extracts features using ORB algorithm
  - Creates a keyframe
  - Performs visual only bundle adjustment to estimate the pose
- In IO
  - Scale converges fast when treated as an optimization variable in BA
  - Uses Pose estimated from VO and IMU measurements
  - Calculate a term called inertial residual
  - Solve the optimization problem to get pose

Tracking

- In VIO
  - Combines VO and IO into a single optimization problem
  - Uses IMU pre-integration to calculate Inertial Residual
  - Combines Visual and Inertial residuals to form an optimization problem
  - Solves problem using factor graphs to get the required pose
- If tracking is lost for 20s, new map created
- If new map tracking is lost 15s after IMU initialization, map is discarded

Re-Localization

- Required to find the position based on previous info when tracking fails
- Uses an algorithm called Max Likelihood Perspective n Points (MLPnP)
- MLPnP works on any type of camera

Place Recognition

- Find Candidate Key frames
- Look for nearby Keyframes with similar matching map points
- Compute Transformation using RANSAC to align map points and choose best transformation
- Refine the Transformation by solving a nonlinear optimization problem
- Verify false positives
- Verify gravity vector

If this is done, the algorithm goes into map merging

Loop closing/Map Merging

- Transformation is applied to 2 maps to be merged in a small region
- Merged maps transition into the new active map
- Common map points are removed to avoid duplicate points
- Local BA is performed in the welding window
- Once the maps are merged the poses are recalculated by optimizing the new essential graph

CAMERA MODEL and CALIBRATION

- ORB-SLAM assumed in all system components a pin-hole camera model.
- SLAM systems rectify either the whole image, or the feature coordinates, to work in an ideal planar retina.
- This forces to crop-out the outer parts of the image, losing the advantages of large Field of View
  - Relocalization:

    ORB-SLAM solves the relocalization problem by setting a Maximum Perspective-n-Point algorithm (MLPnP) that is completely decoupled from the camera model as it uses projective rays as input.

  - Non-rectified Stereo SLAM:

    Considering the stereo rig as two monocular cameras having:

    1) a constant relative SE transformation between them, and

    2) optionally, a common image region that observes the same portion of the scene.

VISUAL-INERTIAL SLAM

ORB-SLAM-VI provides a fast and accurate IMU initialization technique with pin-hole and fisheye cameras.

- **IMU Initialization:**
    - Vision-only MAP Estimation
    - Inertial-only MAP Estimation
    - Visual-Inertial MAP Estimation

VISUAL-INERTIAL SLAM

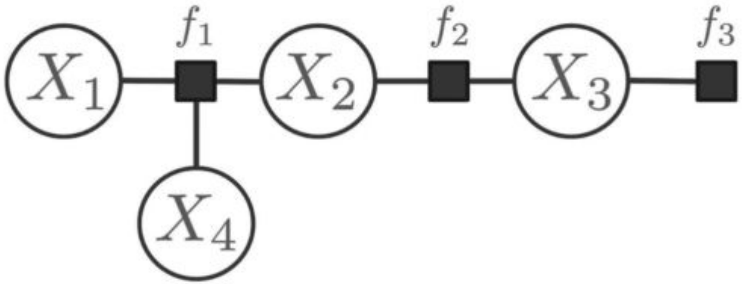
Tracking and Mapping:

- Tracking solves a simplified visual-inertial optimization where only the states of the last two frames are optimized, while map points remain fixed.

Robustness to tracking loss

- visual-inertial system enters into visually lost state when less than 15 point maps are tracked, and achieves robustness.
- If the system gets lost within 15 seconds after IMU initialization, the map is discarded.
- This prevents to accumulate inaccurate and meaningless maps.

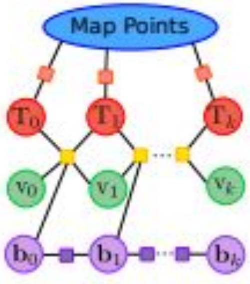
FACTOR GRAPHS



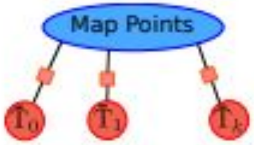
$$g(X_1, \ldots, X_4) = f_1(X_1, X_2, X_4)\, f_2(X_2, X_3)\, f_3(X_3)$$

- Bi-partite graph.
- Simplification of Probability Distributions.
- Used to visualise independence relations between variables.
- Efficient execution of inference tasks.
- Used to solve least-square problems.

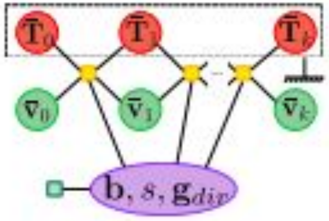
FACTOR GRAPHS



(a) Visual-Inertial



(b) Visual-Only



(c) Inertial-Only

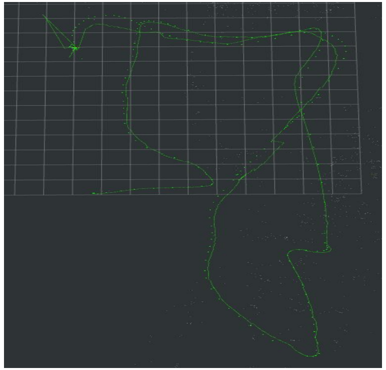
$$\mathcal{Y}_k^* = \arg\min_{\mathcal{Y}_k} \left(\|\mathbf{b}\|^2_{\Sigma_b^{-1}} + \sum_{i=1}^{k} \|\mathbf{r}_{\mathcal{I}_{i-1,i}}\|^2_{\Sigma^{-1}_{\mathcal{I}_{i-1,i}}} \right)$$

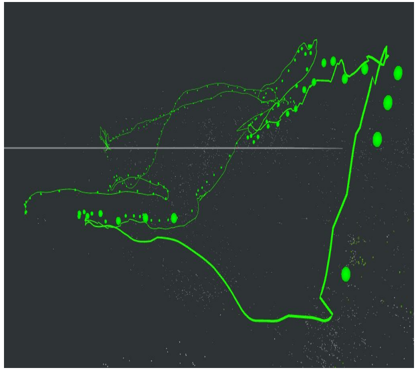
T: Pose

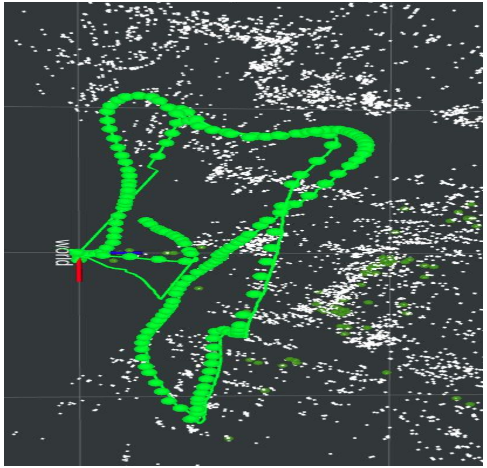
v: Velocities between keyframes

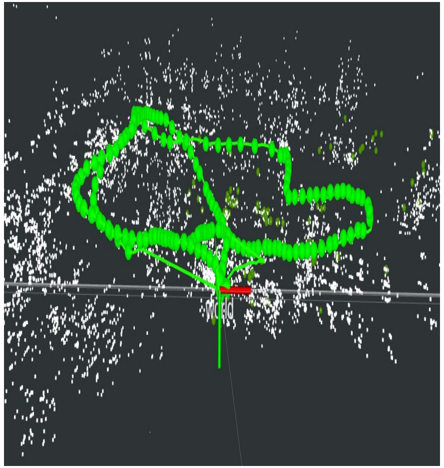
b: Accelerometer and gyroscope biases

EUROC DATASET



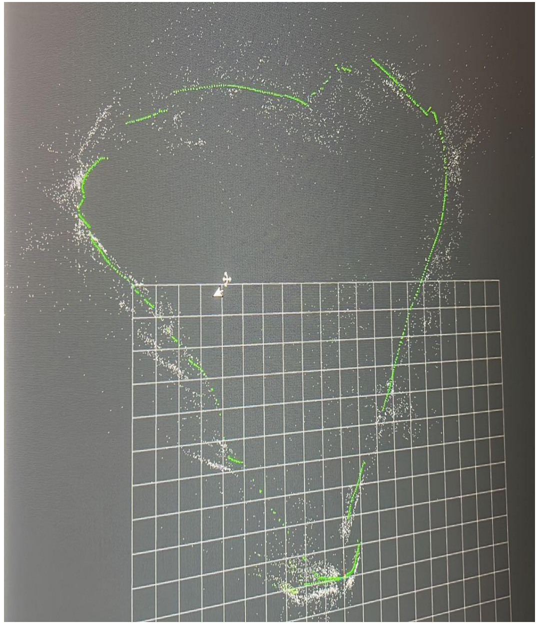


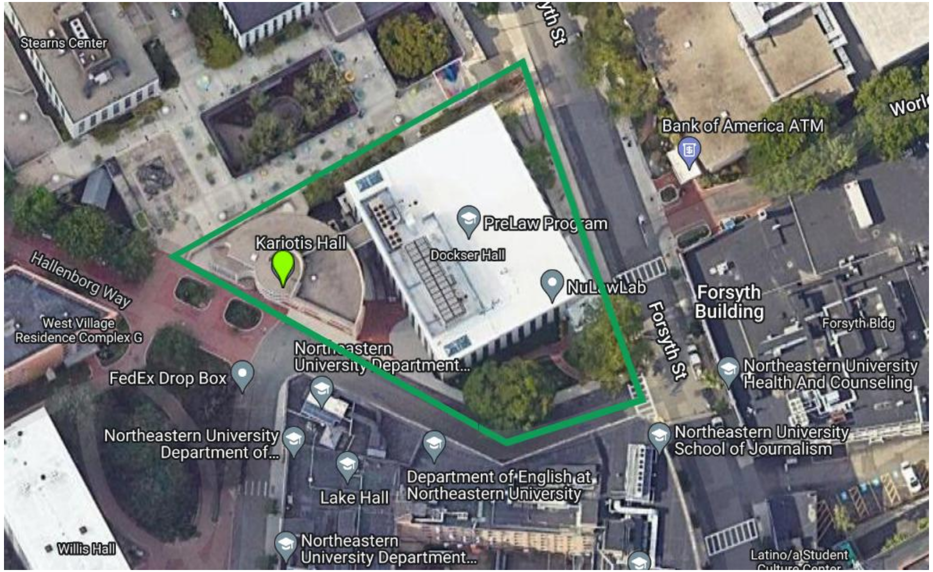




Indoor Snell Dataset

Outdoor Dataset





CHALLENGES

- Pangolin is not supported by every system and makes everything crash if its not disabled (it is required to build)
- Monocular mode is REALLY inconsistent
- Faces challenges in handling dynamic environments and large-scale mapping.
- Sudden Camera movement in corners loosing mapping points.
- Some failure cases under challenging conditions such as weak and repetitive texture, and bad illumination.

FUTURE IMPLEMENTATION

- Integration of an object detection module for semantic segmentation
- Test out a monocular depth detection algorithm to see if it performs any better
- Fuse Lidar odometry with Visual Inertial Odometry
- See if any of the above can give us the full path taken by NUANCE dataset


Q&A