MICHIGAN ENGINEERING
UNIVERSITY OF MICHIGAN

iLoco: Real-Time Visual SLAM using an iPhone

Group 1: Adeep Das, Nikhil Sridhar, Velu Manohar, Muhammad Khan (adeepdas, niksrid, velu, mamankh)

Agenda

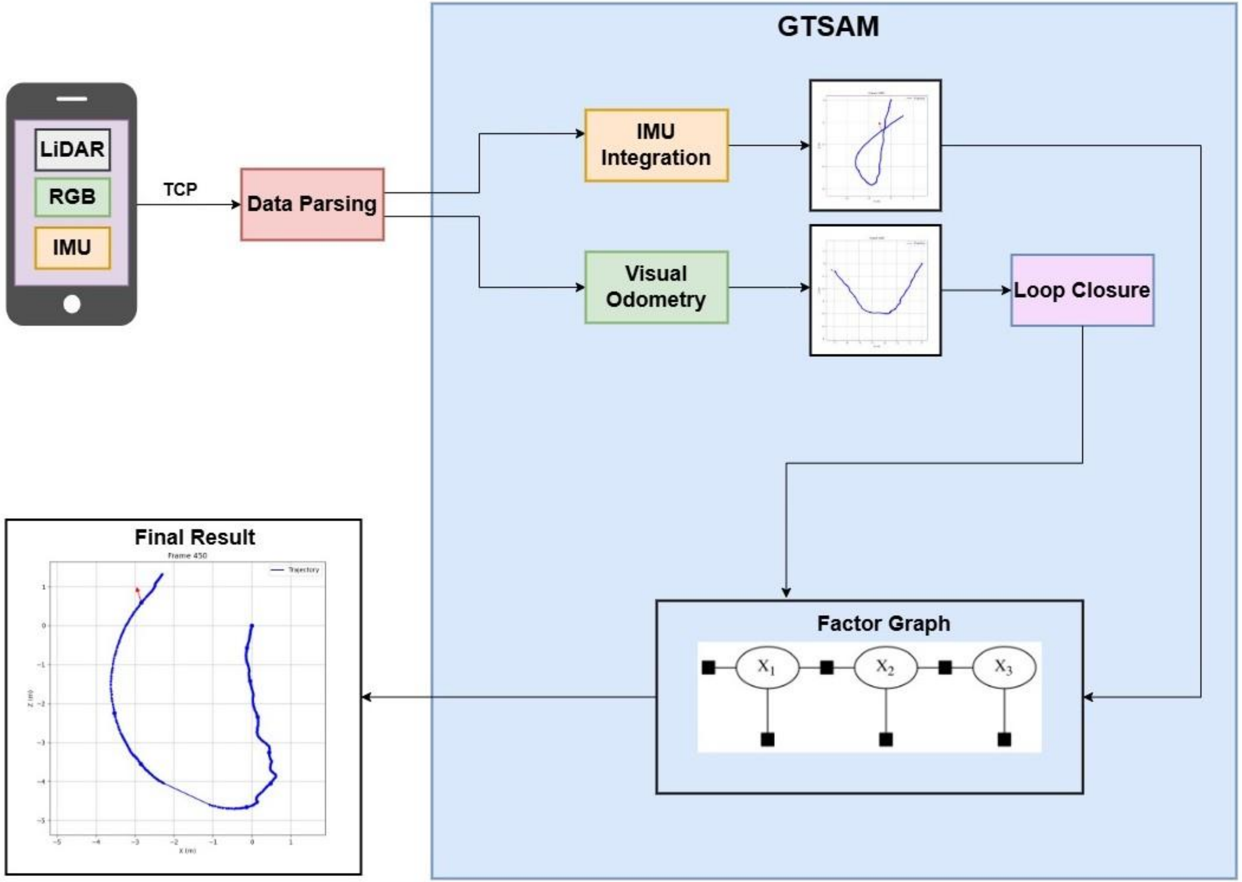
1. Overview
2. Our Approach
3. Data Collection
4. IMU Odometry
5. Visual Odometry
6. Loop Closure
7. GTSAM
8. Results
9. Conclusion

Overview

iLoco: Real-Time SLAM on iPhone

- Traditional SLAM systems often require specialized hardware and complex calibration
- New Smartphones, especially iPhones, offer powerful sensors
- Leverages iPhone’s built-in RGB-D camera and IMU for accurate, real-time localization
- Uses ORB feature matching, Bag-of-Words, GTSAM for our pipeline
- “Slap on” design: minimal setup, no external calibration required
- Ideal for:
    - Rapid prototyping
    - Educational demos
    - Accessible SLAM research

Our Approach



Data Capture

Swift App on iPhone

- Captures RGB + Depth via AVCapture APIs
- Depth and RGB tightly synchronized using AVCaptureDataOutputSynchronizer

Inertial Data via CoreMotion

- Accelerometer + Gyroscope at **100 Hz**
- Timestamped and aligned with visual stream

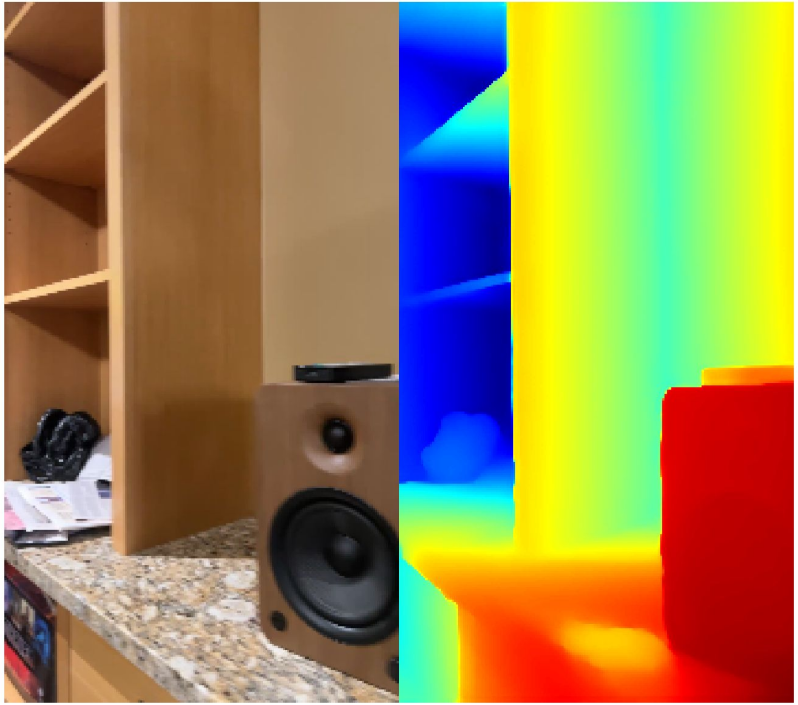
Dual TCP Streaming Architecture

- IMU streamed at 100 Hz
- RGB-D streamed at 30 FPS (H.264 encoded)
- Separate channels reduce packet delay and jitter

Latency-Optimized Design

- H.264 video compression done on iPhone
- Python server decodes stream in real time
- Supports efficient, live SLAM processing

RGB (Left) and Depth (Right) Frames



IMU Odometry

Timestamped IMU Data

- Collected accelerometer & gyroscope readings at 100 Hz
- Data aligned with RGB-D stream for fusion

Rotation Updates via Angular Velocity

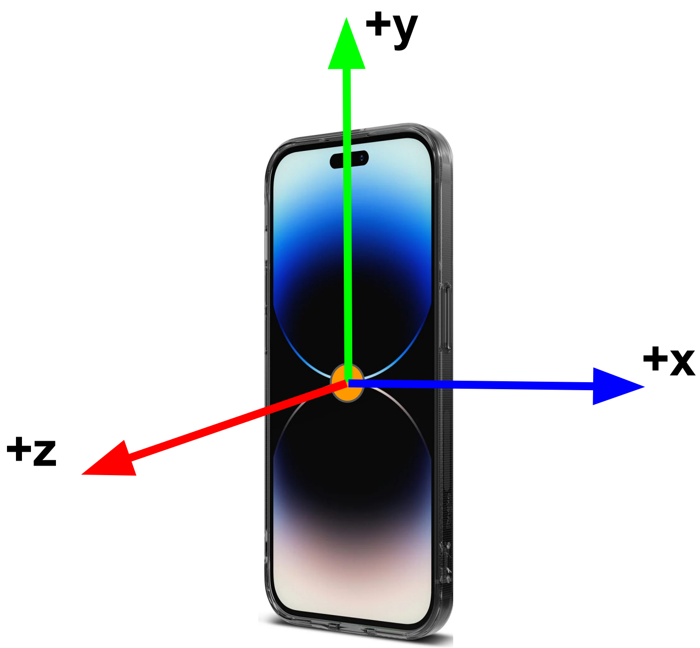
- Skew-symmetric matrix + Rodrigues' formula
- Small-angle approximation with Taylor expansion
- Exponential map applied to angular velocity for rotation matrix

State Propagation

- Discrete-time integration with Zero-Order Hold assumption
- Constant acceleration & angular velocity assumed between frames
- Velocity and position updated from integrated accelerations

Bias-Corrected Integration

- Gyro and accelerometer biases compensated
- Results in smooth, real-time pose estimation



Visual Odometry

Feature Detection

- RGB frames + depth maps captured via iPhone's camera + LiDAR
- ORB keypoints + binary descriptors computed per frame

Feature Matching

- Matched with FLANN (LSH indexing)
- Lowe's ratio test filters unreliable matches

3D Projection from Depth

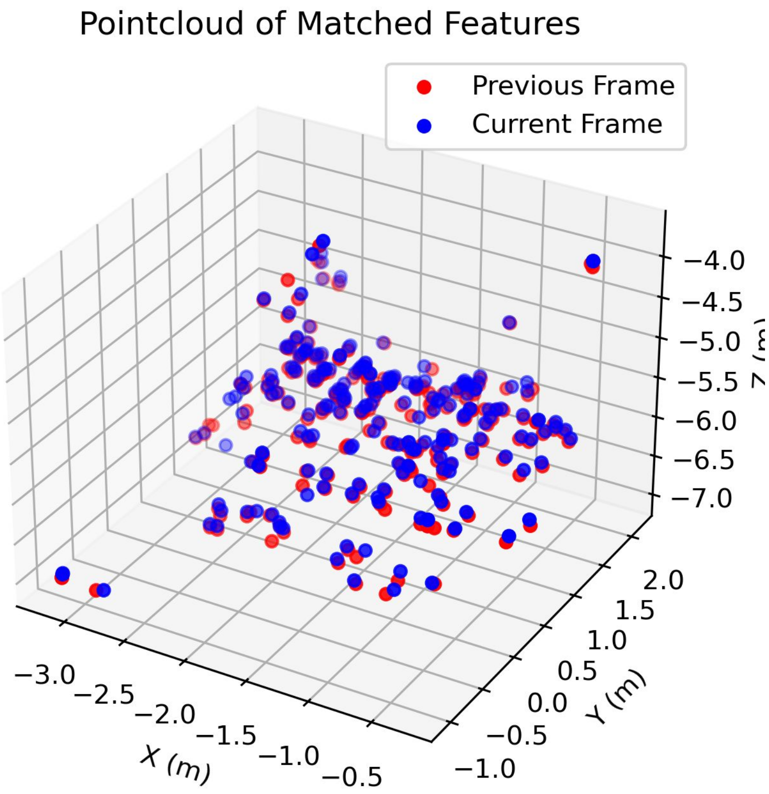
- Depth + camera intrinsics → 3D point cloud
- Only valid depth pixels retained

Pose Estimation

- RANSAC filters outliers
- Rigid transform computed via SVD

Trajectory Accumulation

- Transformations chained over time
- Produces full 6-DoF trajectory in global frame



Loop Closure

Bag-of-Words with ORB

- Visual vocabulary trained from ORB descriptors
- New frames converted into compact BoW vectors

Efficient Candidate Search

- BoW vectors stored in a KD-Tree
- Compared using cosine similarity
- Thresholded to find loop candidates

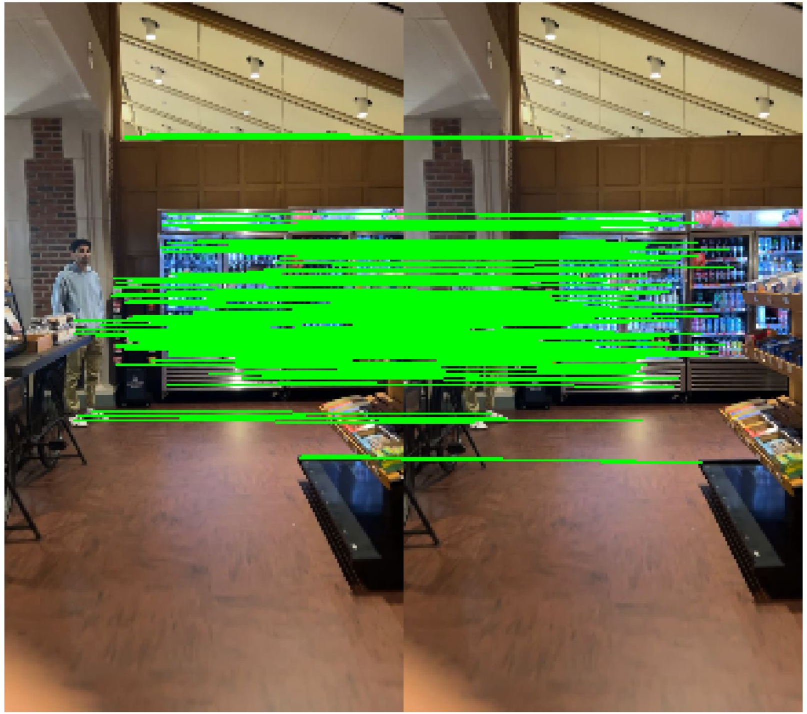
Verification via ORB Matching

- Feature-level validation of candidates
- Loop accepted if enough reliable matches found

Challenges

- Noisy VO/IMU data
- Short runtime limited loop closure effectiveness

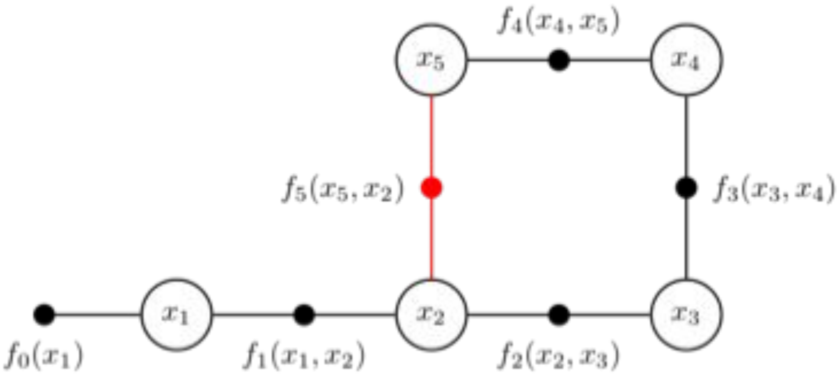
Feature Matches



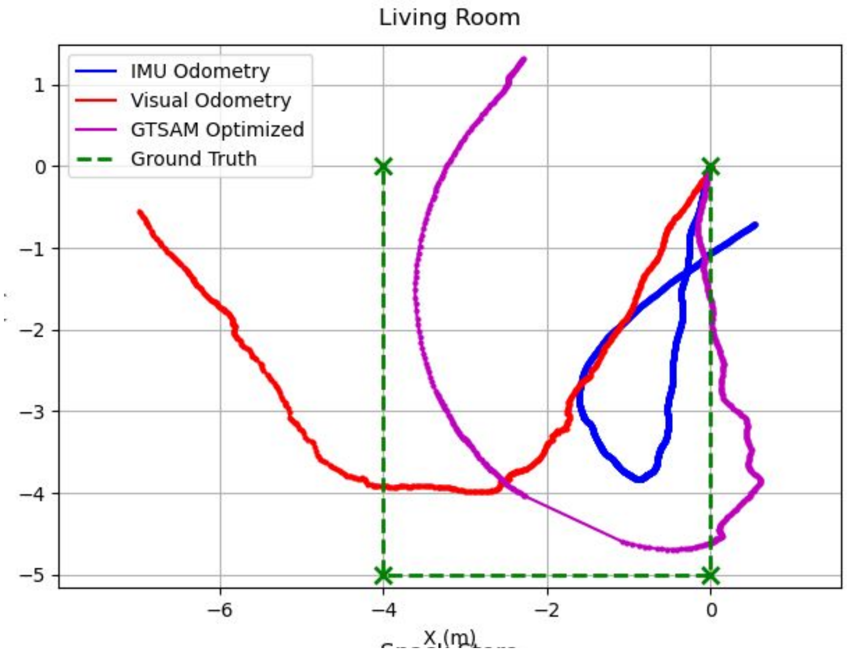
Loop Closure between Frame 150 (left) and frame 12 (right)

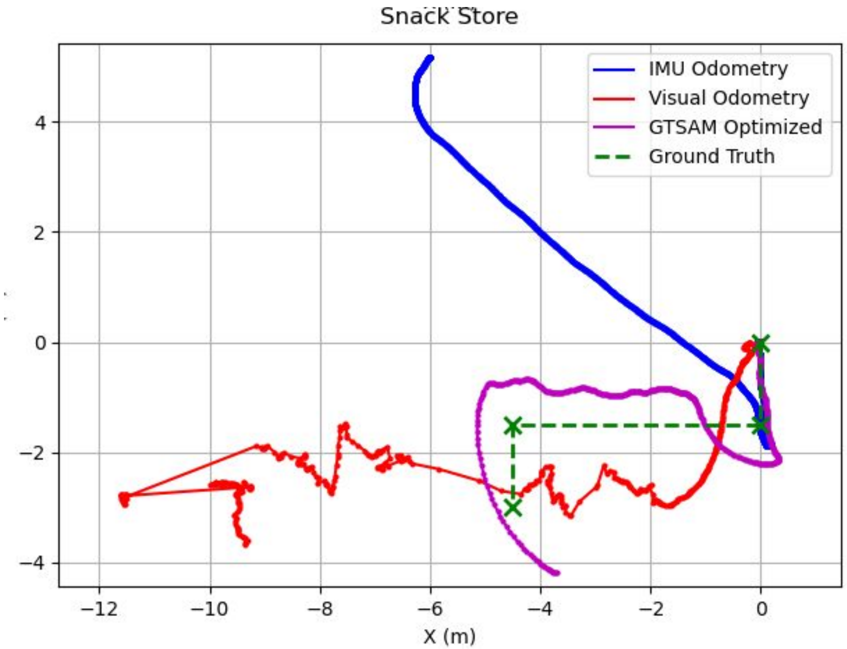
GTSAM

- Performs batch SLAM by combining IMU preintegration with visual odometry
- Builds a factor graph:
  - IMU provides high-frequency motion constraints between keyframes
  - Visual odometry supplies relative pose constraints
- Uses ISAM2 for efficient incremental optimization
- Fuses multi-rate, multi-modal data into a globally consistent trajectory
- Incorporates motion priors and sensor noise models for robustness
- Produces smooth, drift-reduced pose estimates
- Backbone for evaluating downstream video alignment and trajectory prediction



Results


Living Room
IMU Odometry
Visual Odometry
GTSAM Optimized
Ground Truth
1
0
−1
−2
−3
−4
−5
−6
−4
−2
0
X (m)


Snack Store
IMU Odometry
Visual Odometry
GTSAM Optimized
Ground Truth
4
2
0
−2
−4
−12
−10
−8
−6
−4
−2
0
X (m)

More Results

- Measure error as MSE in Lie Group over time

Time (s)	IMU Odometry (Avg ± SD) [m^2]	Visual Odometry (Avg ± SD) [m^2]	Our Method (Avg ± SD) [m^2]
10	0.405 ± 0.075	0.281 ± 0.018	0.292 ± 0.732
20	1.632 ± 0.075	0.541 ± 0.243	0.56 ± 0.332
30	4.542 ± 0.195	1.27 ± 2.549	1.168 ± 1.352
40	8.988 ± 0.465	3.485 ± 1.162	2.14 ± 1.192
50	13.824 ± 1.485	6.13 ± 13.503	3.296 ± 2.088
60	18.6 ± 2.985	7.854 ± 6.778	4.024 ± 2.136

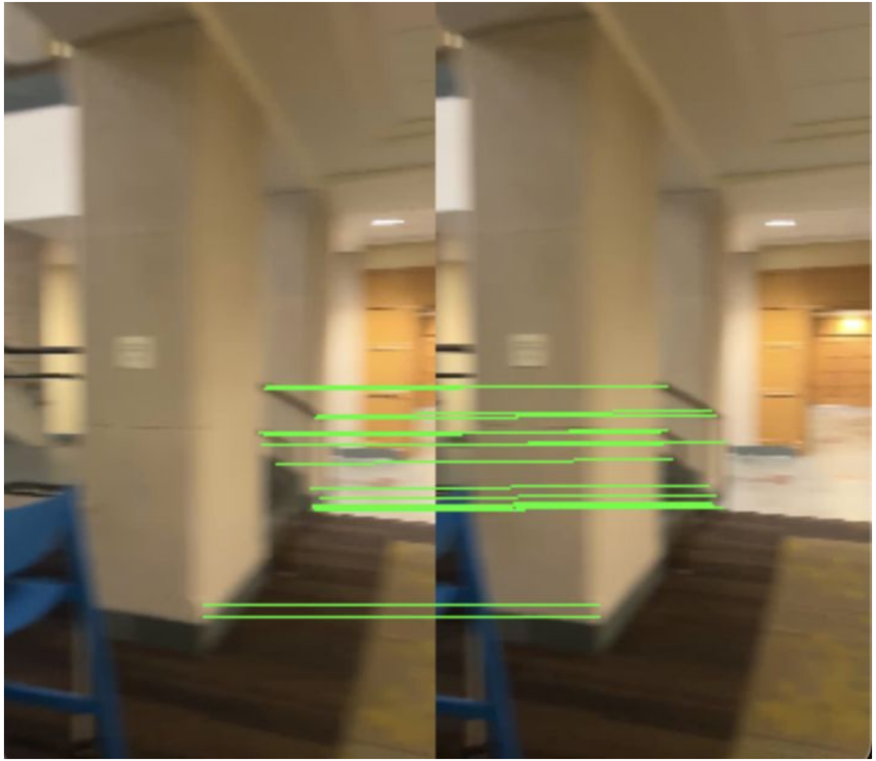
Conclusion

- **Visual Odometry:** Performs poorly during turns, especially in low-texture environments like plain walls.

- **GTSAM Optimization:** Mini-batch updates improve stability and convergence without full batch costs.

- **iLoco vs Baselines:** Our fused method outperforms standalone IMU/VO but lags behind ARKit in rotation accuracy.

- **Future Work:** Add mapping for loop closure and integrate visual and depth data (e.g., LiDAR) separately for better robustness.



Thank You!