Stable Vision-Aided Navigation for Large-Area Augmented Reality

Taragay Oskiper, Han-Pang Chiu, Zhiwei Zhu
Supun Samarasekera, Rakesh "Teddy" Kumar

Vision and Robotics Laboratory
SRI-International Sarnoff, Princeton, NJ, USA
Email: {han-pang.chiu,rakesh.kumar}@sri.com

Outline

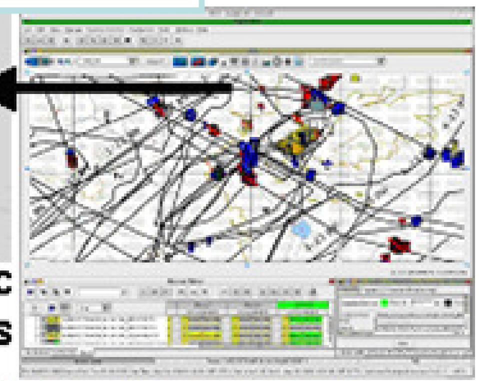
- Goal
  - Large-area augmented reality training/gaming systems using head mounted displays (HMDs)
- The real-time vision-aided navigation component
- An extended Kalman filter for 6 DOF pose estimation
  - Error-state, IMU-centric
  - Relative pose estimation through multi-camera visual odometry
  - Absolute correction from landmark matching with a pre-built database
  - Covariance modeling on landmark points for stabilization
  - Head prediction for real-time implementation
- Results
- Conclusions and future work

Large-Area Augmented Reality: *Making Live Training/Gaming Come to Life*

Insert Tracers, explosions, muzzle flashes, and "3D sound" in the real scene,

Place *intelligent synthetic* actors into the ***real*** 3D scene

Augmented outdoor scene



Synthetic Entities

Culturally realistic, reactive, dynamic, synthetic entities support non-kinetic and kinetic interactions

War-fighter Worn Video & HMD

Real scene

Seamless indoor/ outdoor tracking of trainee and weapon position and orientation

Realistic depth mapping and occlusion

- Closed loop full spectrum collective training,
- Repeatable and scriptable, with unlimited variation
- Rapidly deployable at home and deployed stations

- Exercise Review and Immersive After Action Analysis

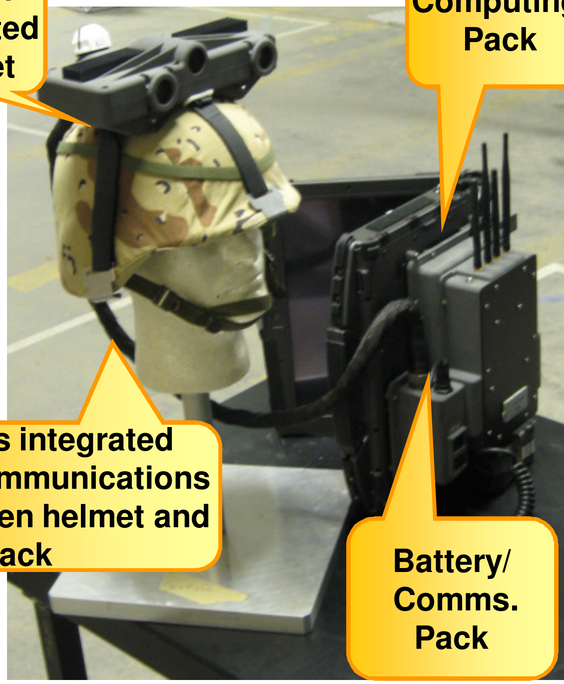
Demanding Requirements for Navigation

- It must estimate highly-accurate 6DOF pose estimation of the user's head, then the system knows where to insert virtual objects in the real scene viewed by users.

- The pose estimation needs to be consistent and stable. Jitter or drift on inserted objects disturbs the illusion of mixture between rendered and real world.

- It needs to operate seamlessly for large areas indoors and outdoors.

Helmet based Interface Subsystem for Navigation



Sensor Integrated Helmet

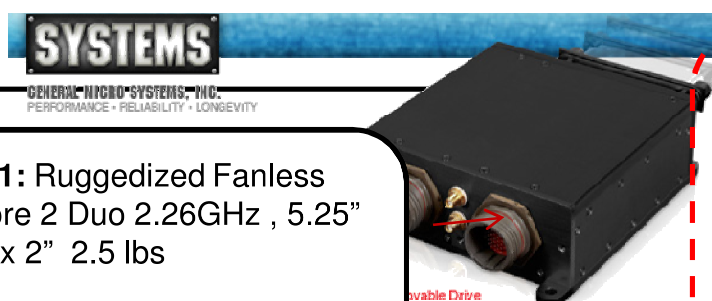
Computing Pack

Cables integrated for communications between helmet and backpack

Battery/ Comms. Pack

Portable Sensor Module. Front and Back stereo cameras and IMU. Attaches to sensor mount on helmet.

Processor package in backpack



Option1: Ruggedized Fanless Intel Core 2 Duo 2.26GHz , 5.25” x 5.25” x 2” 2.5 lbs

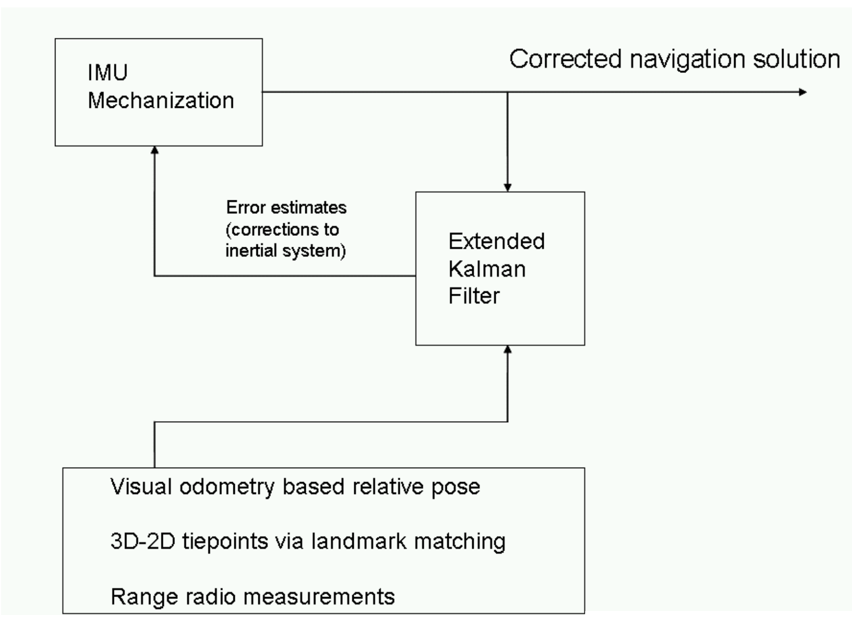
Option 2: Dell rugged laptop system, Intel Core 2 Dua 2.53GHz

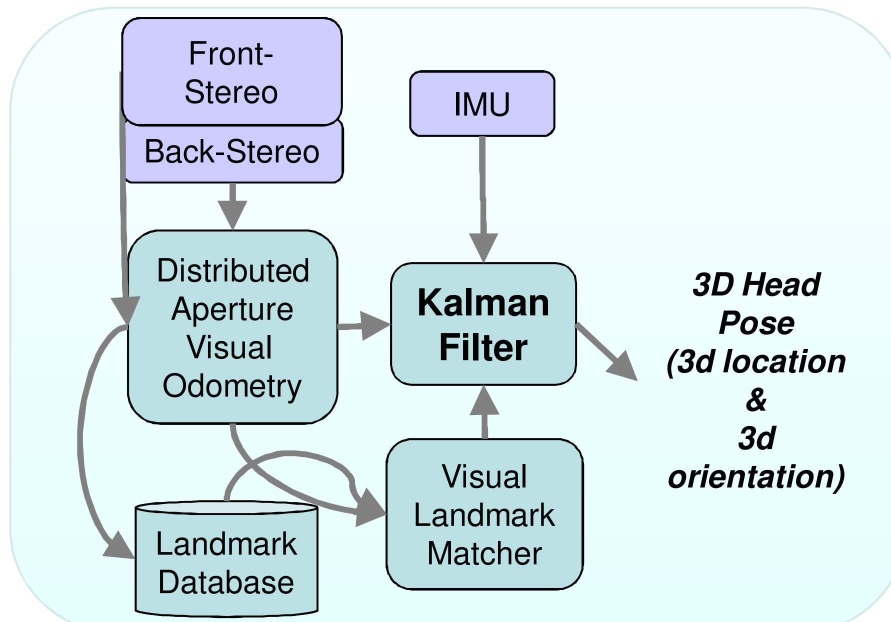
Extended Kalman Filter

- Our Kalman filter adopt the so called "error-state" formulation, so there is no need to specify an explicit dynamic motion model.

- The filter dynamics follow from the IMU error propagation equations
  - Which evolve slowly over time
  - And are more amenable to linearization

• The updating of the Kalman filter comes from two external source data
  • Relative pose information provided by visual odometry module
  • Global measurements provided by the visual landmark matching module





Prediction (IMU Propagation)

- The total states of our filter: camera location T_{CG}, the gyroscope bias vector b_g, velocity vector v in global coordinate frame, accelerometer bias vector b_a, and ground to camera orientation q_{GC}.

$$s = [q_{GC}^T \quad b_g^T \quad v^T \quad b_a^T \quad T_{CG}^T]^T$$

- IMU mechanization equations for the state estimate propagation with the gyroscope $\omega_m(t)$ and accelerometer $a_m(t)$ readings from the IMU between consecutive video frame time instants.

- The Kalman filter error state:

$$\delta s = [\delta\Theta^T \quad \delta b_g^T \quad \delta v^T \quad \delta b_a^T \quad \delta T_{CG}^T]^T$$

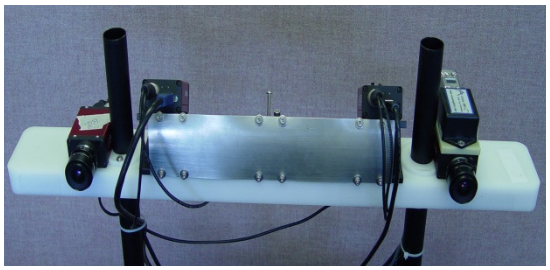
- The dynamic process model of the error-state:

$$\delta s = F\delta s + Gn$$

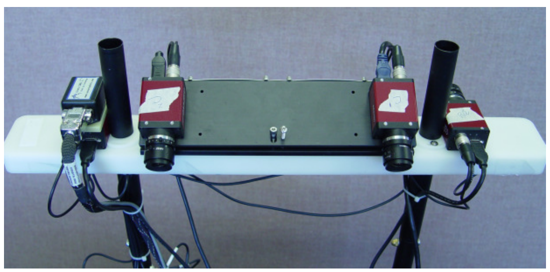
$$\mathbf{F} = \begin{bmatrix} -[\hat{\omega}]_\times & -\mathbf{I}_3 & \mathbf{0}_{3x3} & \mathbf{0}_{3x3} & \mathbf{0}_{3x3} \\ \mathbf{0}_{3x3} & \mathbf{0}_{3x3} & \mathbf{0}_{3x3} & \mathbf{0}_{3x3} & \mathbf{0}_{3x3} \\ -\hat{\mathbf{R}}_{GC}^T[\hat{\alpha}]_\times & \mathbf{0}_{3x3} & \mathbf{0}_{3x3} & -\hat{\mathbf{R}}_{GC}^T & \mathbf{0}_{3x3} \\ \mathbf{0}_{3x3} & \mathbf{0}_{3x3} & \mathbf{0}_{3x3} & \mathbf{0}_{3x3} & \mathbf{0}_{3x3} \\ \mathbf{0}_{3x3} & \mathbf{0}_{3x3} & \mathbf{I}_3 & \mathbf{0}_{3x3} & \mathbf{0}_{3x3} \end{bmatrix}, \mathbf{n} = \begin{bmatrix} \mathbf{n}_g \\ \mathbf{n}_{wg} \\ \mathbf{n}_a \\ \mathbf{n}_{wa} \end{bmatrix}, \text{ and } \mathbf{G} = \begin{bmatrix} -\mathbf{I}_3 & \mathbf{0}_{3x3} & \mathbf{0}_{3x3} & \mathbf{0}_{3x3} \\ \mathbf{0}_{3x3} & \mathbf{I}_3 & \mathbf{0}_{3x3} & \mathbf{0}_{3x3} \\ \mathbf{0}_{3x3} & \mathbf{0}_{3x3} & -\hat{\mathbf{R}}_{GC}^T & \mathbf{0}_{3x3} \\ \mathbf{0}_{3x3} & \mathbf{0}_{3x3} & \mathbf{0}_{3x3} & \mathbf{I}_3 \\ \mathbf{0}_{3x3} & \mathbf{0}_{3x3} & \mathbf{0}_{3x3} & \mathbf{0}_{3x3} \end{bmatrix}$$

Multi-Camera Visual Odometry: Front/Back Stereo Pairs

Backpack system with two stereo pairs



Frontal view

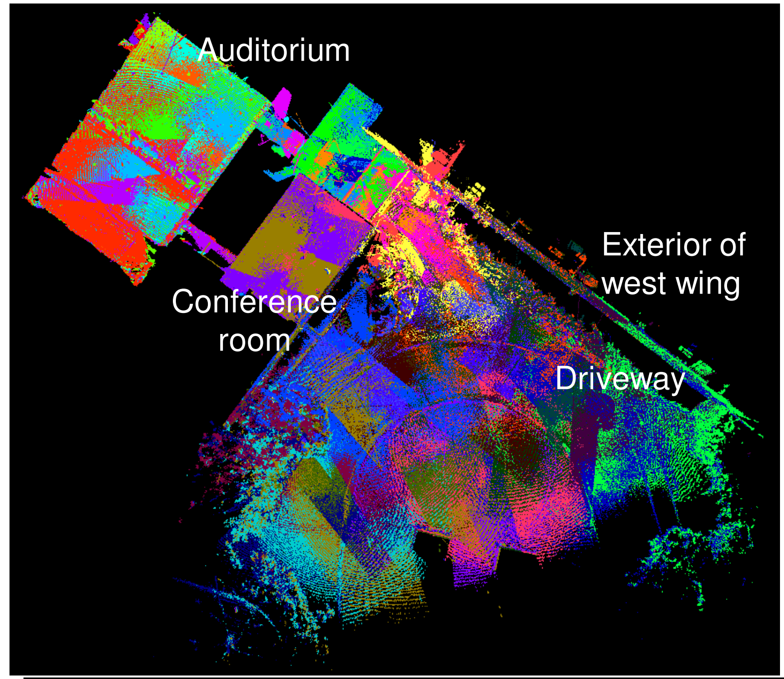


Back view

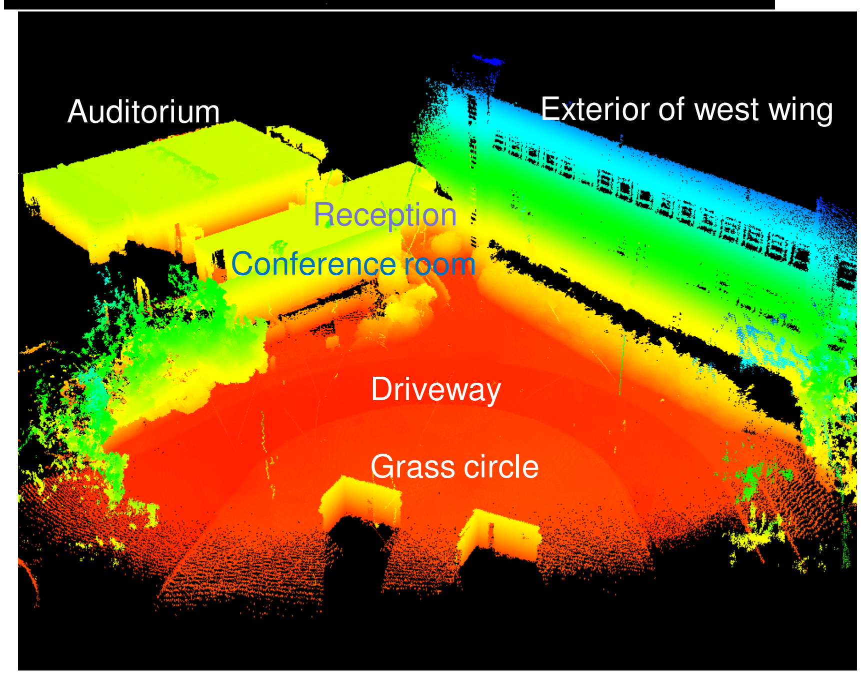
Wide total FOV greatly enhances accuracy/robustness

- Ambiguity of estimation between rotation and translation is mitigated
- Moving objects unlikely to dominate total FOV
- Harris-corner features are tracked across frames to estimate relative pose
- Improved precision by using multiple cameras and tracking features across large FOV

Building a Landmark database from Lidar and Video

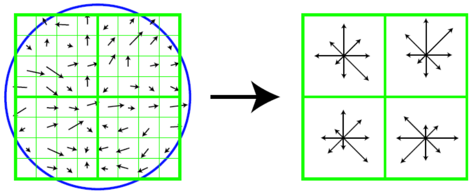


- 40 scans and video taken outdoor and indoor (every 5 m)
- Automatic alignment of Lidar scans using coarse-to-fine algorithms
- Different colors show overlapping of different aligned scans



Accumulated Point Cloud

Global Landmark Matching



Histogram of Gradient (HOG) features

Matched landmarks under different viewpoints

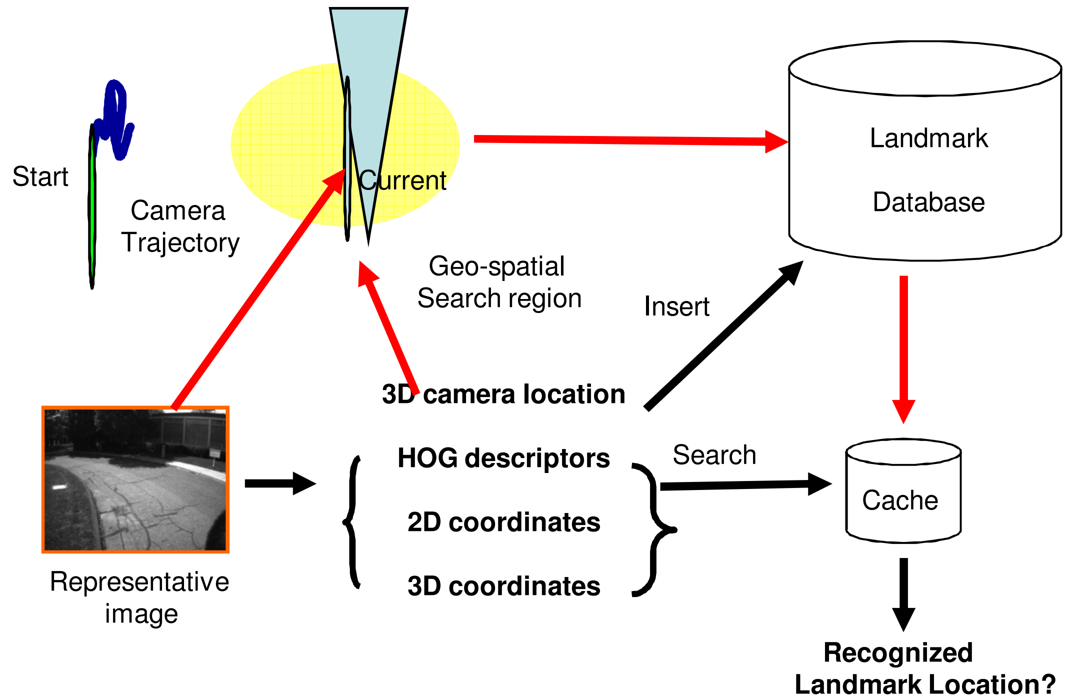
Technical Approach

• HOG descriptor-based feature representation of landmark scene points

• Geometry-constrained landmark matching with outlier removal

•Detection of distinctive natural landmarks under various viewpoint illumination, and distance changes

• Robust, real-time matching of landmarks from a large database



Correction (Vision Measurements)

- We use the stochastic cloning approach to handle relative pose measurements from visual odometry module.

  - measurements are a function of the propagated error-state of the current time instance and the cloned error-state from previous time instance.

- We transfer each 3d local landmark point to global coordinate as point measurements from landmark matching.

$$\mathbf{Y} = \mathbf{R}_{LG}\mathbf{X} + \mathbf{T}_{LG}$$

$$\delta\mathbf{Y} \simeq \hat{\mathbf{R}}_{LG}\delta\mathbf{X} + \left[\hat{\mathbf{R}}_{LG}\hat{\mathbf{X}}\right]_{\times}\rho + \delta\mathbf{T}_{LG}$$

$$\Sigma_Y \simeq \hat{\mathbf{R}}_{LG}\Sigma_X\hat{\mathbf{R}}_{LG}^T + [\tilde{\mathbf{X}}]_{\times}\Sigma_{\mathbf{R}_{LG}}[\tilde{\mathbf{X}}]_{\times}^T + \Sigma_{\mathbf{T}_{LG}}$$

$$\mathbf{z} = f(\mathbf{Z}) + \nu \quad \text{with} \quad f(\mathbf{Z}) = [Z_1/Z_3 \;\; Z_2/Z_3]^T$$

$$\mathbf{Z} = \mathbf{R}_{GC}\mathbf{Y} + \mathbf{T}_{GC} = \mathbf{R}_{GC}(\mathbf{Y} - \mathbf{T}_{CG}) \, .$$

$$\delta\mathbf{Z} \simeq \left[\hat{\mathbf{R}}_{GC}(\hat{\mathbf{Y}} - \hat{\mathbf{T}}_{CG})\right]_{\times}\delta\Theta + \hat{\mathbf{R}}_{GC}(\delta\mathbf{Y} - \delta\mathbf{T}_{CG}) + \nu \, .$$

$$\delta\mathbf{z}_L \simeq \mathbf{H}_L\delta\mathbf{s} + \eta$$

$$\mathbf{H}_L = \mathbf{J}_f[\mathbf{J}_\Theta \quad \mathbf{0}_{3\times3} \quad \mathbf{0}_{3\times3} \quad \mathbf{0}_{3\times3} \quad \mathbf{J}_{\delta\mathbf{T}_{CG}}]$$

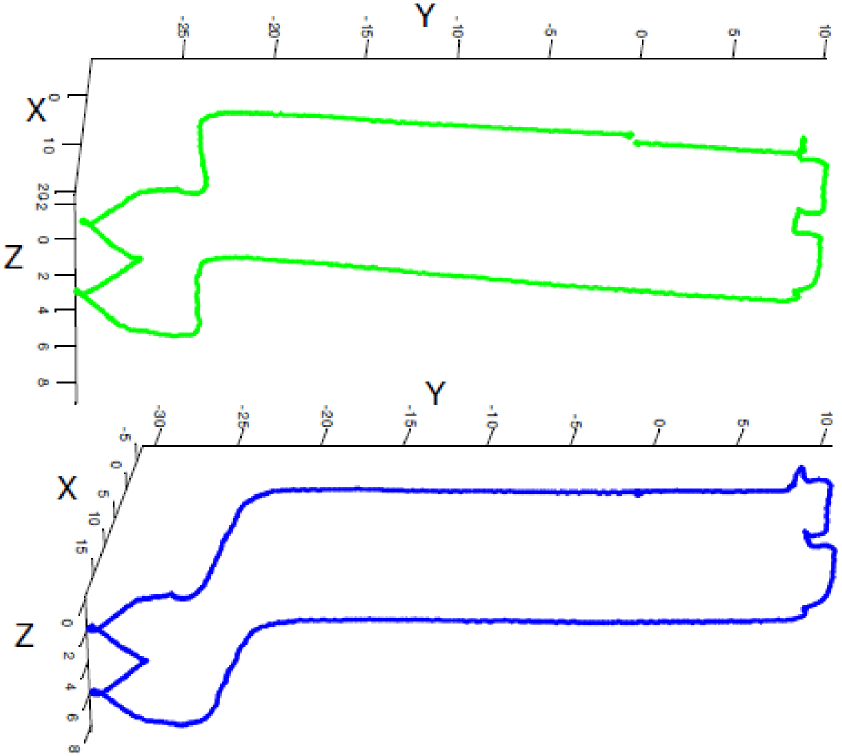
$$\mathbf{J}_f = \begin{bmatrix} 1/\hat{Z}_3 & 0 & -\hat{Z}_1/\hat{Z}_3^2 \\ 0 & 1/\hat{Z}_3 & -\hat{Z}_2/\hat{Z}_3^2 \end{bmatrix}$$

$$\mathbf{J}_\Theta = \left[\hat{\mathbf{R}}_{GC}(\hat{\mathbf{Y}} - \hat{\mathbf{T}}_{CG})\right]_{\times}, \quad \text{and} \quad \mathbf{J}_{\delta\mathbf{T}_{CG}} = -\hat{\mathbf{R}}_{GC}$$

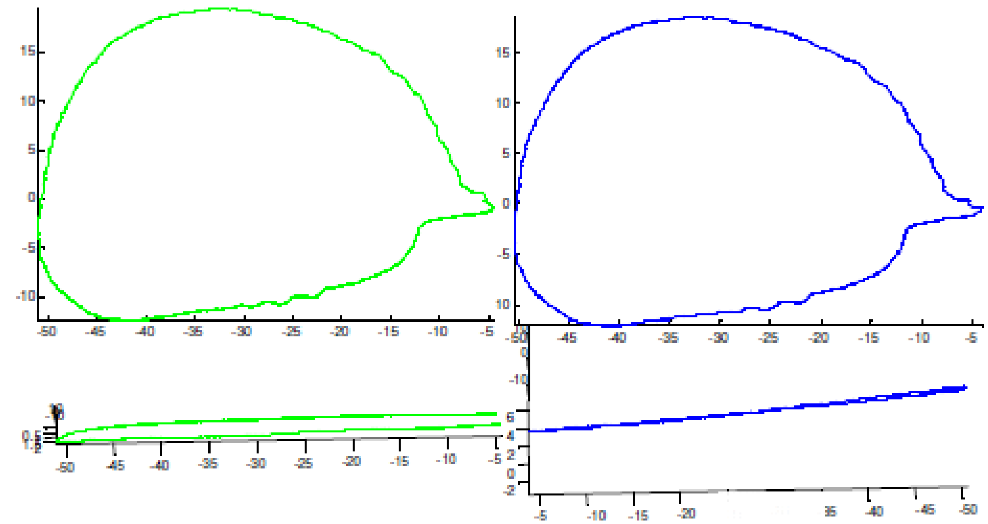
$$\Sigma_\eta = \mathbf{J}_f[\hat{\mathbf{R}}_{GC}\Sigma_Y\hat{\mathbf{R}}_{GC}^T]\mathbf{J}_f^T + \Sigma_\nu$$

Results on Fusing Visual Odometry and Inertial Data

- We compare our new error-state filter to our previous old filter
  - Previous old filter used a constant motion model assumption.
  - New filter based on error state model does not need to make any assumption.



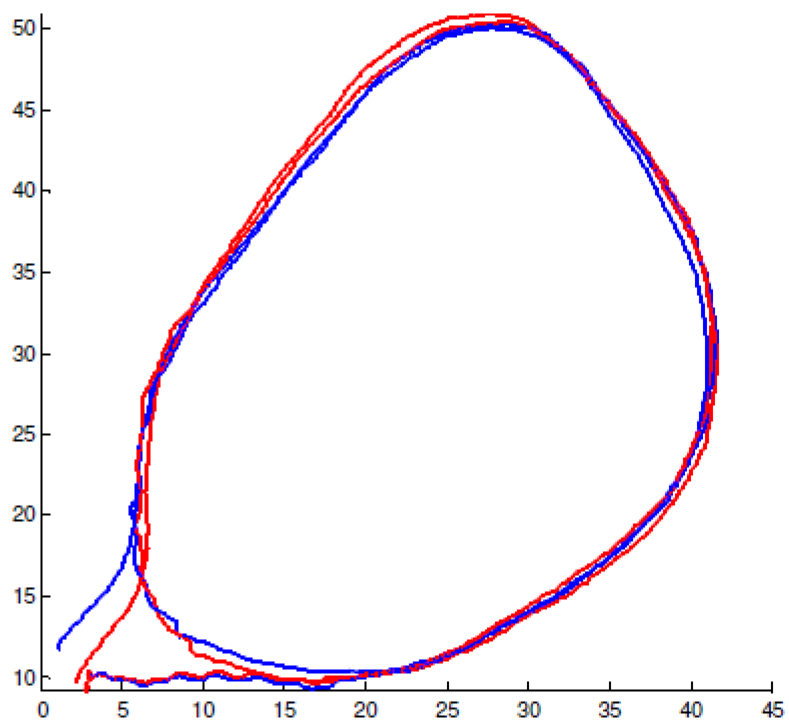
Indoor: Total distance – 157.5 meters
Closure error: Old filter: 0.6760 meter,
New filter: 0.4639 meter

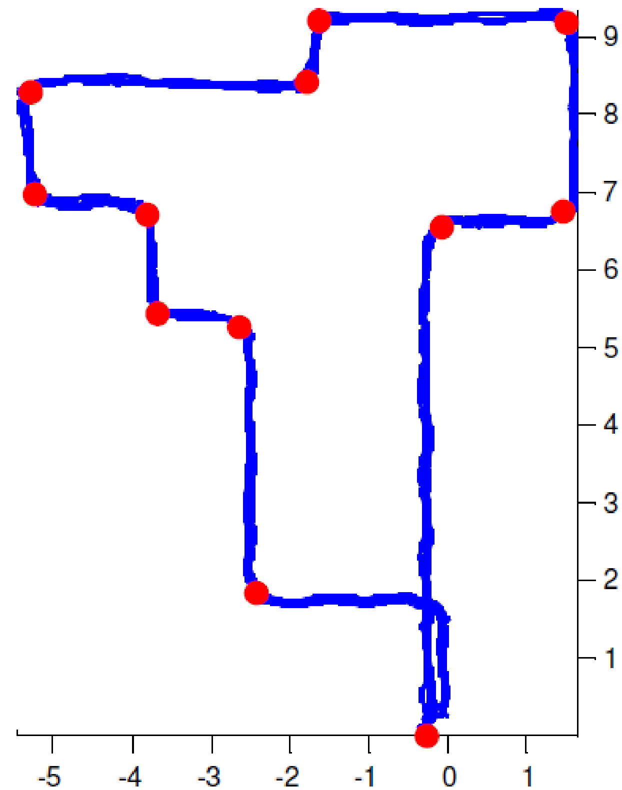


Outdoor: Total distance – 129 meters
Closure error: Old filter: 1.2020 meter,
New filter: 0.3916 meter

Results on Fusing Local and Global Measurements

- Blue: local measurements, Red: local and global measurements.

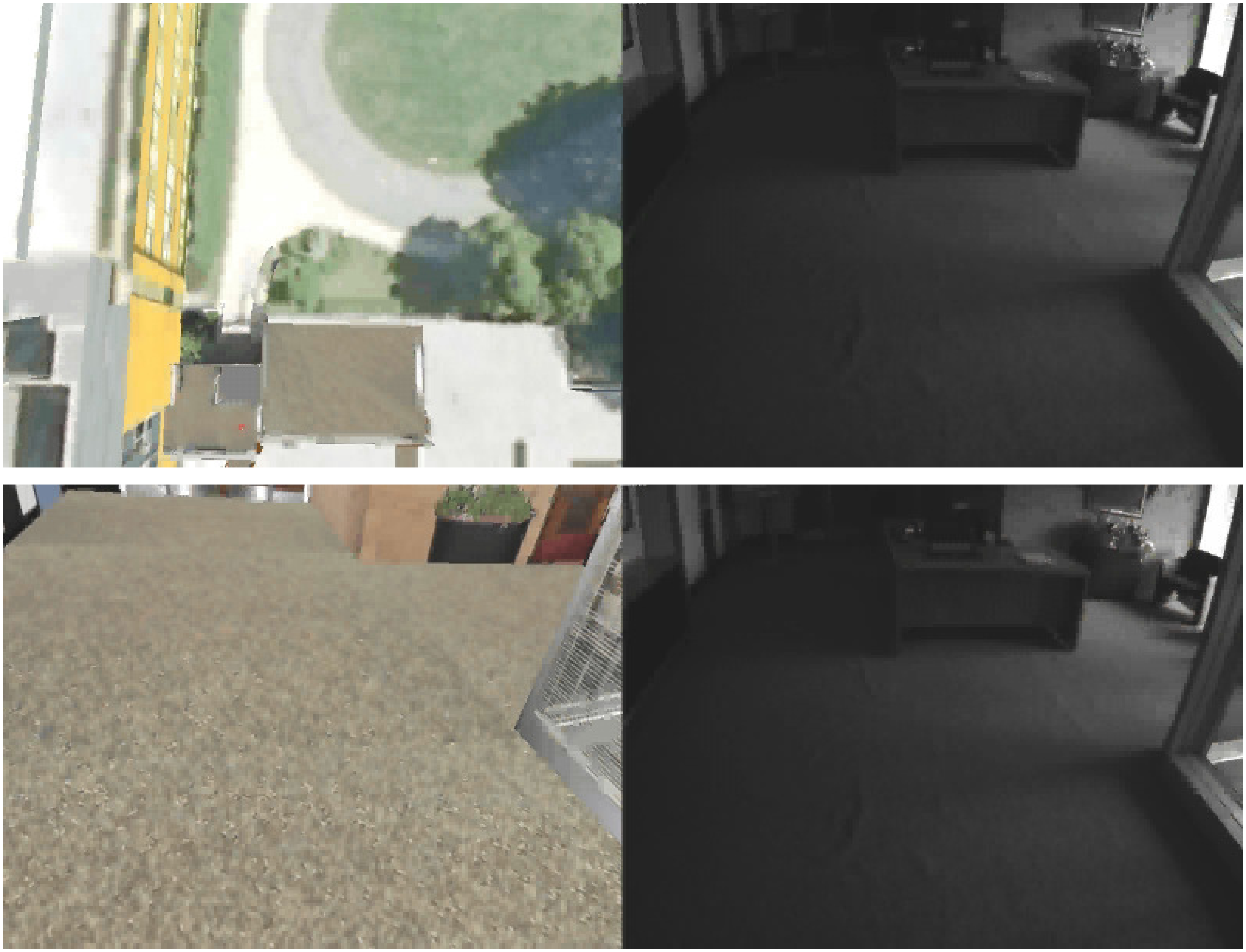




Outdoor: Total distance – 256 meters
Closure error: Local Measurements: 2.49 m,
Global Measurements: 0.57 m

Indoor: 5 repetitions, 165 meters
Average error on 12 marked positions (60 repetitions): 0.085 meters
Global Measurements

Indoor/ Outdoor Tracking Long Sequence Results



Top View

Camera View

Reducing Jitter in Insertion
Covariance-Based Filtering of Pose

- Inconsistent pose estimation causes jumps/jitters during insertion.
- The accuracy of pose estimation decreases if there are fewer landmark point matches closer to the camera where the "depth information" is more accurate.
- We model the 3d reconstruction uncertainty of landmarks P = [Px,Py,Pz] and implicitly rely more on closer landmark point matches in Kalman filter.

$$\Sigma_X = \mathbf{J} \begin{bmatrix} \mathbf{I}_2 & 0_{2x2} \\ 0_{2x2} & \mathbf{I}_2 \end{bmatrix} \mathbf{J}^T$$

$$pl = [pl_x \quad pl_y]^T + n = [\mathbf{P}_x/\mathbf{P}_z \quad \mathbf{P}_x/\mathbf{P}_z]^T + n$$

$$pr = [pr_x \quad pr_y]^T + n$$

$$pr_x = \frac{R_1\mathbf{P}_x + R_2\mathbf{P}_y + R_3\mathbf{P}_z + T_1}{R_7\mathbf{P}_x + R_8\mathbf{P}_y + R_9\mathbf{P}_z + T_3}$$

$$pr_y = \frac{R_4\mathbf{P}_x + R_5\mathbf{P}_y + R_6\mathbf{P}_z + T_2}{R_7\mathbf{P}_x + R_8\mathbf{P}_y + R_9\mathbf{P}_z + T_3}$$

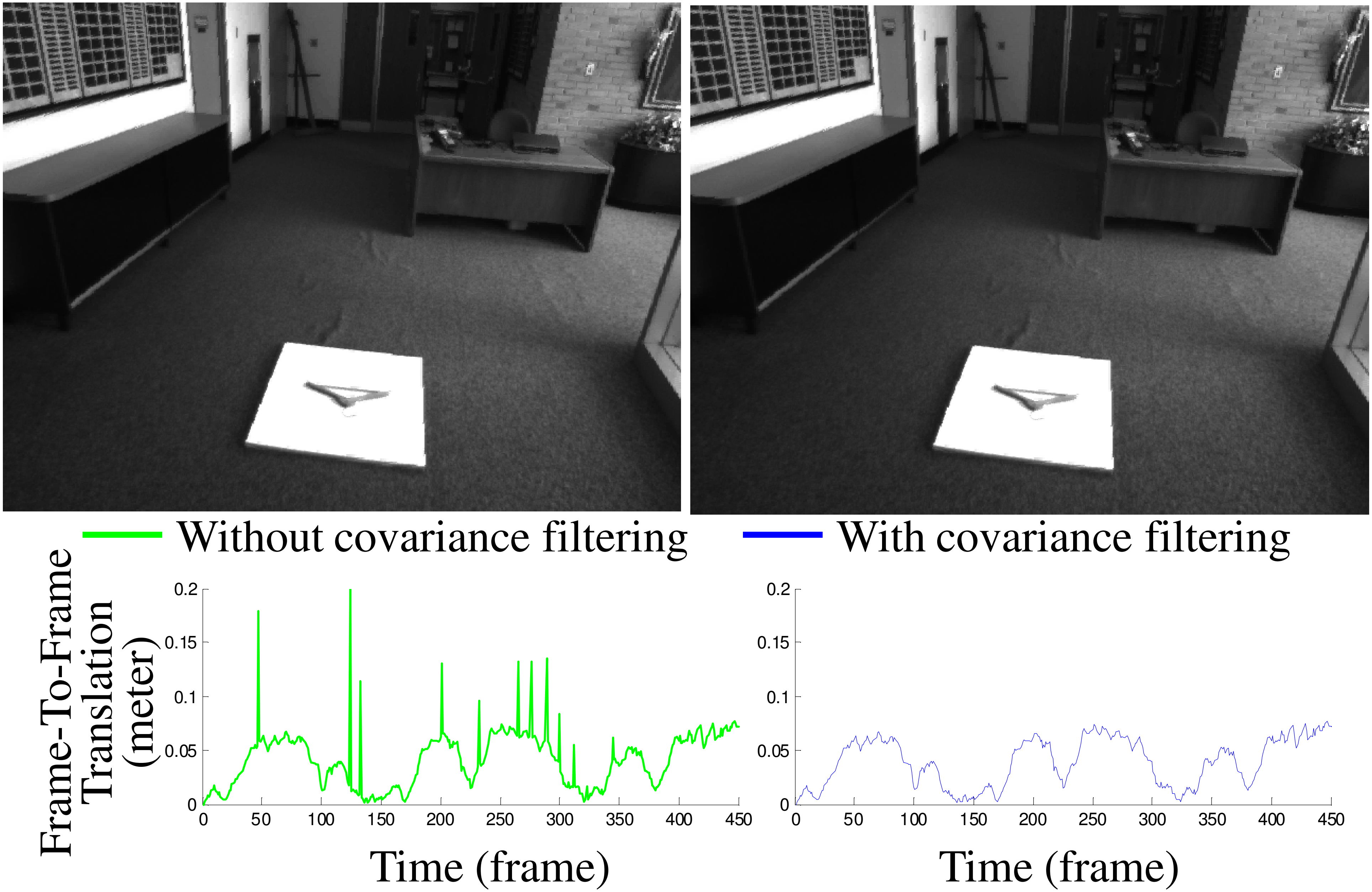
$$R = \begin{bmatrix} R_1 & R_2 & R_3 \\ R_4 & R_5 & R_6 \\ R_7 & R_8 & R_9 \end{bmatrix}, \quad T = [T_1 \quad T_2 \quad T_3]^T$$

$$\hat{\mathbf{P}}_x = pl_x \hat{\mathbf{P}}_z$$

$$\hat{\mathbf{P}}_y = pl_y \hat{\mathbf{P}}_z$$

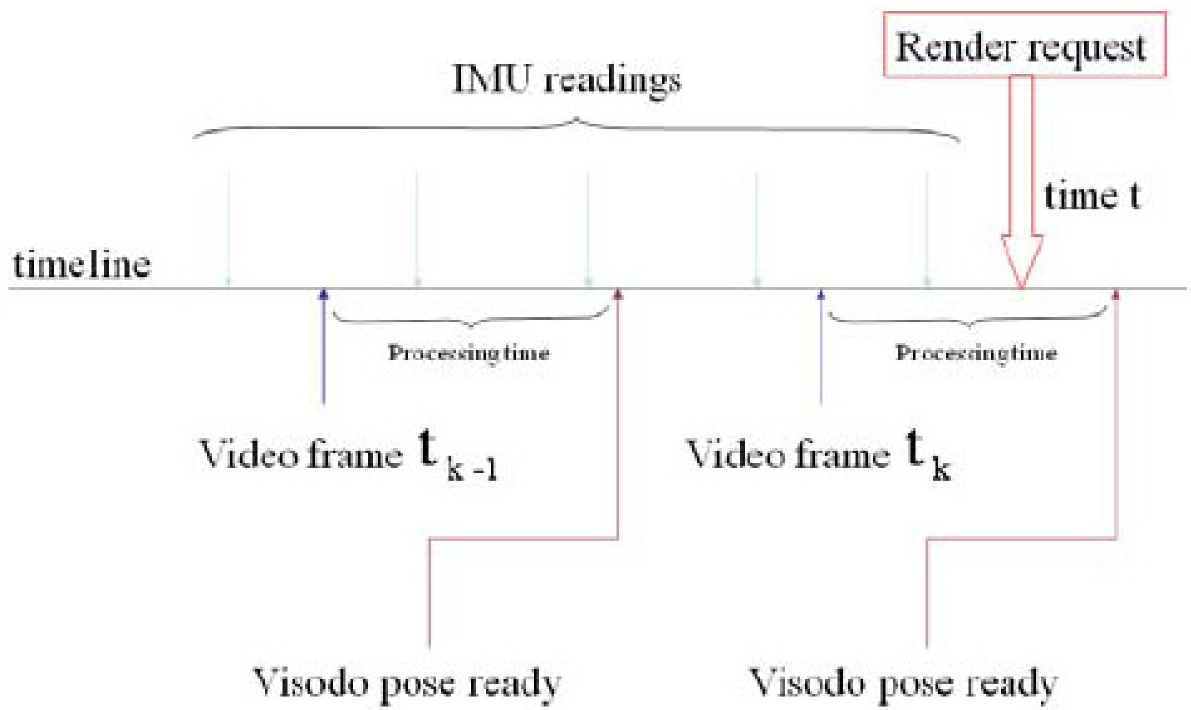
$$\hat{\mathbf{P}}_z = \frac{T_1 - T_3 pr_x}{pr_x(R_7 pl_x + R_8 pl_y + R_9) - (R_1 pl_x + R_2 pl_y + R_3)}$$

Results: Accurate Pose and Jitter-Free



Pose Prediction for Real-Time Implementation

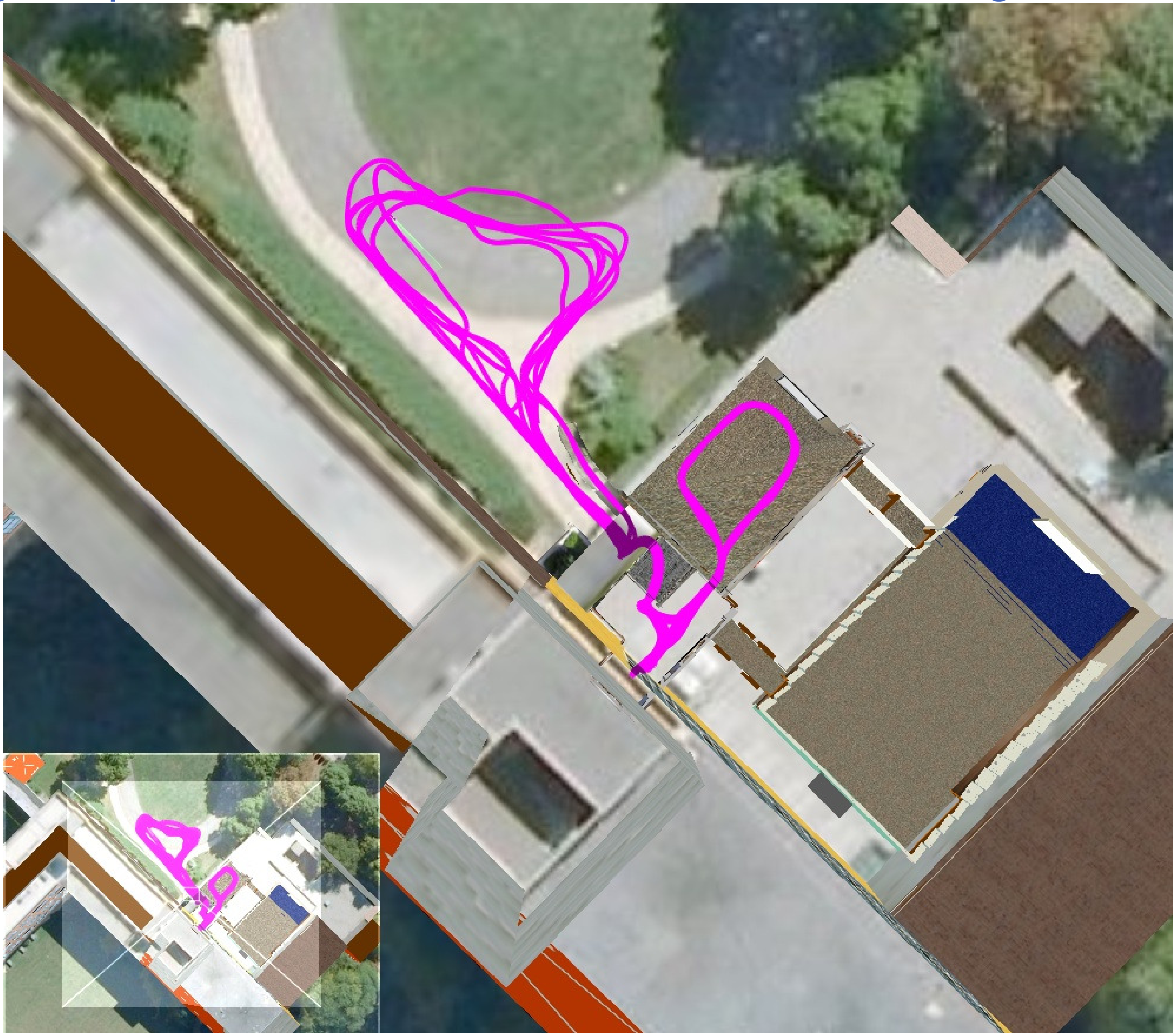
- We use the buffered high-frequency (100Hz) IMU data between the latest frame time and the current render time (15Hz frame rate) to predict the pose.
- This solution is effective when the camera poses are lagged within a single frame period (66 milliseconds at 15Hz frame rate).



Long Sequence Results with Landmark Matching

Total Travelled Distance:
810.6 meters

Total Travelled Time:
16.46 minutes

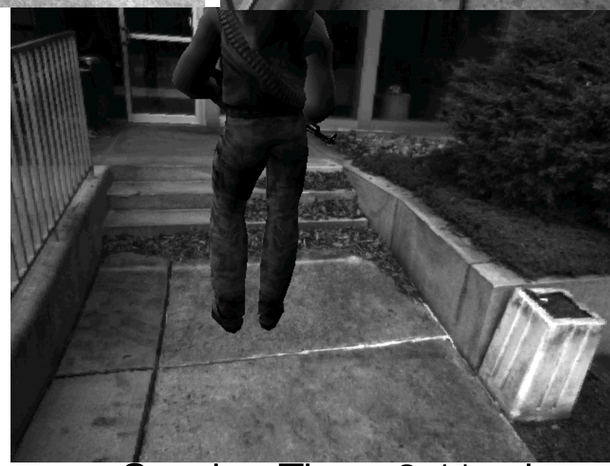


Repeat-Visit Consistency of Insertion

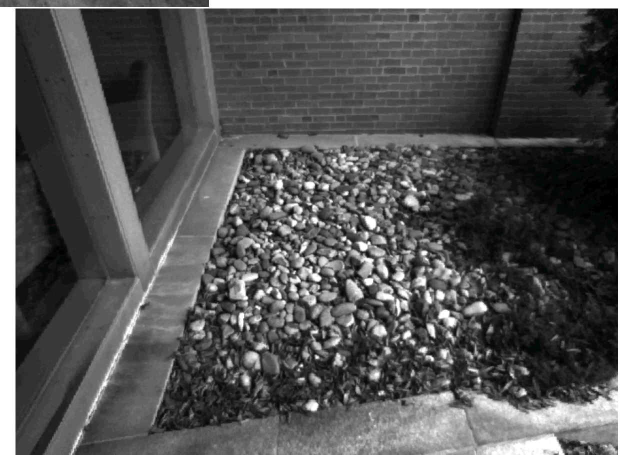
Starting Time: 10.37 min

Starting Time: 11.48 min

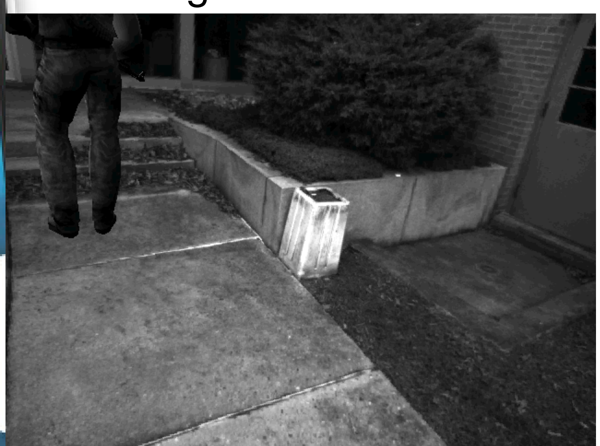
Starting Time: 1.26 min



Starting Time: 3.11 min

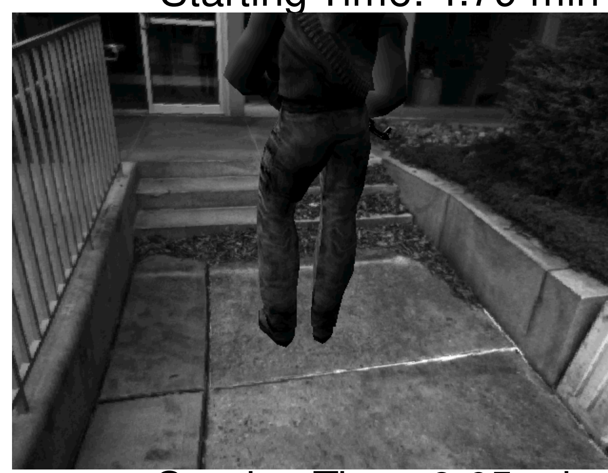


Starting Time: 4.79 min



Starting Time: 5.96 min

Starting Time: 7.51 min



Starting Time: 8.65 min

Insertion of Avatars (Outdoors)

Insertion is done using:

- Estimated 3D Head Pose and Location

Insertion of Avatars (Indoors)

Insertion is done using:

•Estimated 3D Head Pose and Location

•Depth map from stereo for occlusion culling

Video

Conclusions and Future Work

- We proposed a unified Kalman filter framework using local and global sensor data fusion for vision-aided navigation related to augmented reality applications.
- We use landmark matching from a pre-built landmark database to prevent long term drift.
- We capture the 3D reconstruction uncertainty of landmark points to improve the stability of pose estimation.
- Future work:
  - Reduce the number of cameras while maintaining accuracy
  - Update the landmark database automatically

Thank you!