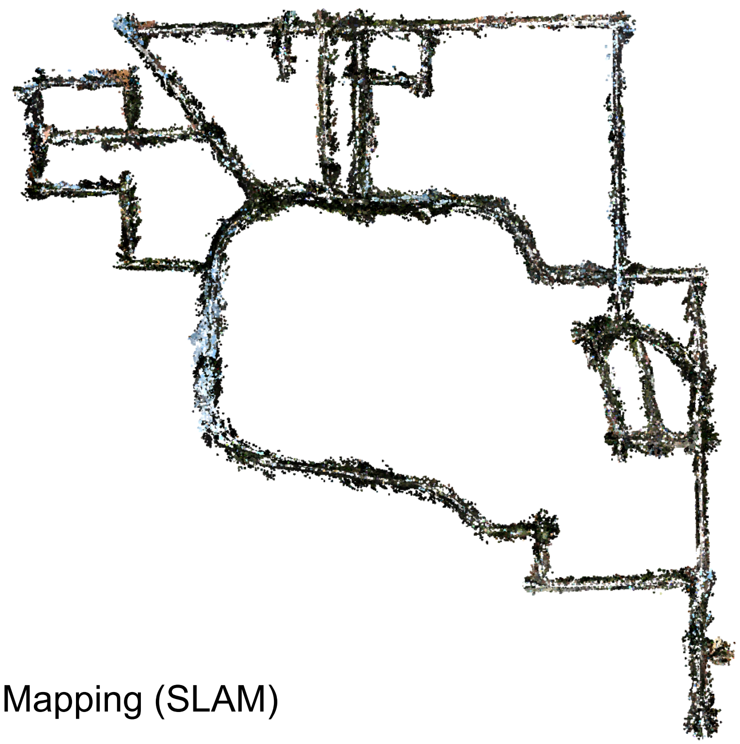
Incremental Light Bundle Adjustment for Robotics Navigation

Vadim Indelman, Andrew Melim, Frank Dellaert

Robotics and Intelligent Machines (RIM) Center
College of Computing
Georgia Institute of Technology

Introduction

- Robot Navigation: Recover the state of a moving robot over time through fusion of multiple sensors, including a monocular camera



Simultaneous Localization and Mapping (SLAM)

Left image courtesy of Georgia Tech Research Institute
Right image courtesy of Chris Beall

Vision-Aided Robot Navigation

- Fusion of monocular image measurements and IMU measurements
  - Full joint pdf:

$$p(X, L, B|Z) \propto \prod_{i}^{N} \left(p(x_i|x_{i-1}, b_i, z_{i,i-1}^{IMU}) \prod_{j \in M_i} p(z_{i,j}^{VIS}|x_i, l_j) \right)$$

N : number of robot states
M_i : set of observed 3D points at state i

l_j : j-th 3D point
x_i : Robot state at time i (pose and velocity)
$z_{i,j}^{VIS}$: Image observation
$z_{i,i-1}^{IMU}$: IMU measurement
b_i : IMU Bias at time i

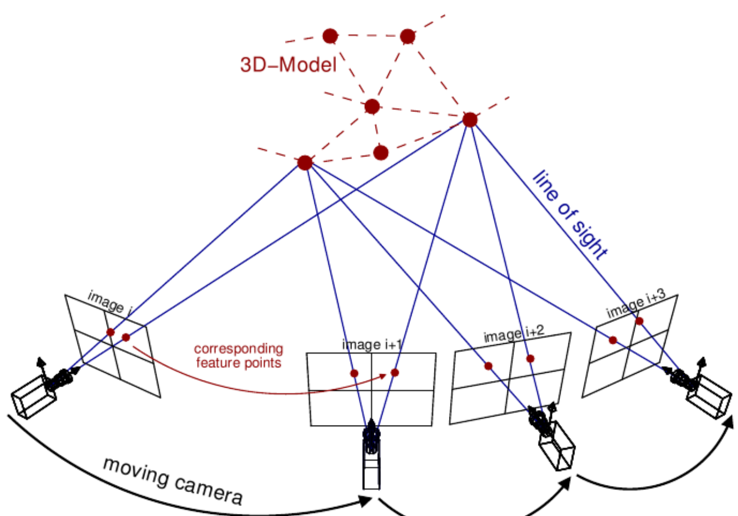


Image from: http://www.tnt.uni-hannover.de/project/motionestimation

Vision-Aided Robot Navigation

- Fusion of monocular image measurements and IMU measurements
  - Full joint pdf:

$$p(X, L, B|Z) \propto \prod_{i}^{N} \left(p(x_i|x_{i-1}, b_i, z_{i,i-1}^{IMU}) \prod_{j \in M_i} p(z_{i,j}^{VIS}|x_i, l_j) \right)$$

N : number of robot states
M_i : set of observed 3D points at state i

l_j : j-th 3D point
x_i : Robot state at time i (pose and velocity)
$z_{i,j}^{VIS}$: Image observation
$z_{i,i-1}^{IMU}$: IMU measurement
b_i : IMU Bias at time i

- Assuming Gaussian distributions:
  - MAP estimate X^*, Y^*, B^* is obtained by

$$X^*, L^*, B^* = \underset{X, L, B}{\arg\max}\; p(X, L, B|Z)$$

$$J(X, L, B) \doteq \sum_{i=1}^{N} \left(||x_i - pred(x_{i-1}, b_i z_{i,i-1}^{IMU})||^2_{\sum^{IMU}} + \sum_{j \in M_1} ||z_{i,j}^{VIS} - \pi(x_i, l_j)||^2_{\sum^{VIS}} \right)$$

Projection of a 3D point into the image plane

$$\pi(x_i, l_j) \doteq K_i \begin{bmatrix} R_i & t_i \end{bmatrix} l_j$$

Mahalanobis squared distance

$$\|a\|^2_\Sigma \doteq a^T \Sigma^{-1} a$$

Image from: http://www.tnt.uni-hannover.de/project/motionestimation

Factor Graph Representation [Kschischang et al. 2001 ToIT]

- Factor graph: a graphical representation of the joint pdf factorization

$$p\left(\mathcal{X}|Z\right) \propto \prod_s f_s\left(\mathcal{X}_s\right)$$

- Full SLAM pdf:

$$p(X,L,B|Z) \propto \prod_{i}^{N} \left(p(x_i|x_{i-1}, b_i, z_{i,i-1}^{IMU}) \prod_{j \in M_i} p(z_{i,j}^{VIS}|x_i, l_j) \right)$$

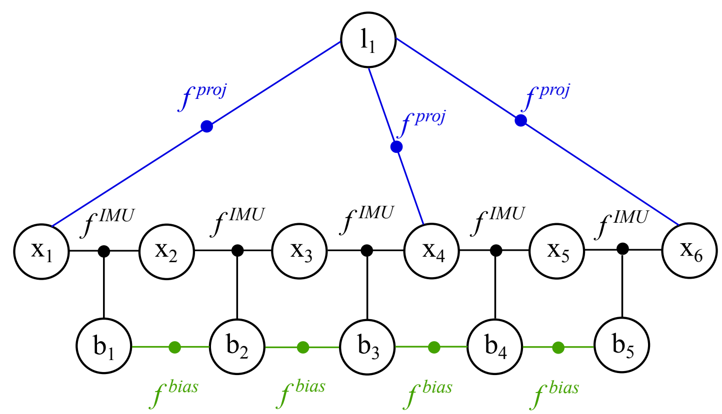
$$\chi \doteq \{X, L, B\}$$

$$p\left(z_{i,j}|x_i, l_j\right) \propto \exp\left(-\frac{1}{2}\left\|z_{i,j} - \pi\left(x_i, l_j\right)\right\|_{\Sigma}^2\right) \doteq f^{proj}\left(x_i, l_j\right)$$

$$p(x_i|x_{i-1}, b_i z_{i,i-1}^{IMU}) \propto \exp(-\frac{1}{2}||x_i - pred(x_{i-1}, b_i, z_{i,i-1}^{IMU}||_{\Sigma^{IMU}}^2) \doteq f^{IMU}(x_i, x_{i-1}, b_i)$$

$$f^{bias}\left(b_{k+1}, b_k\right) \doteq \exp\left(-\frac{1}{2}\left\|b_{k+1} - h^b\left(b_k\right)\right\|_{\Sigma_b}^2\right)$$

The naïve IMU factor
can add a significant number of
unnecessary variables!



Incremental Light Bundle Adjustment (iLBA) for Robot Navigation

- Problems!
  - **3D structure is expensive to compute (and not necessary for navigation)**:
    - Algebraically eliminate 3D points using multi-view geometry constraints
    - Significantly reduce the number of variables for optimization
    - 3D points can always be reconstructed (if required) based on optimized camera poses
  - **High rate sensors introduce large number of variables:**
    - Utilize pre-integration of IMU to reduce the number of variables **[Lupton et al., TRO 2012]**
    - Incremental inference requires only partial re-calculation
      - Update factorization rather than compute from scratch

Three-View Constraints

[Indelman et al., TAES 2012]

- **Theorem**: Algebraic elimination of a 3D point that is observed by 3 views *k,l* and *m* leads to:

$$q_i \doteq R_i^T K_i^{-1} z_i$$

$$g_{2v}\left(x_k, x_l, z_k, z_l\right) \doteq q_k \cdot \left(t_{k\to l} \times q_l\right) = 0$$

$$g_{2v}\left(x_l, x_m, z_l, z_m\right) \doteq q_l \cdot \left(t_{l\to m} \times q_m\right) = 0$$

Epipolar constraints

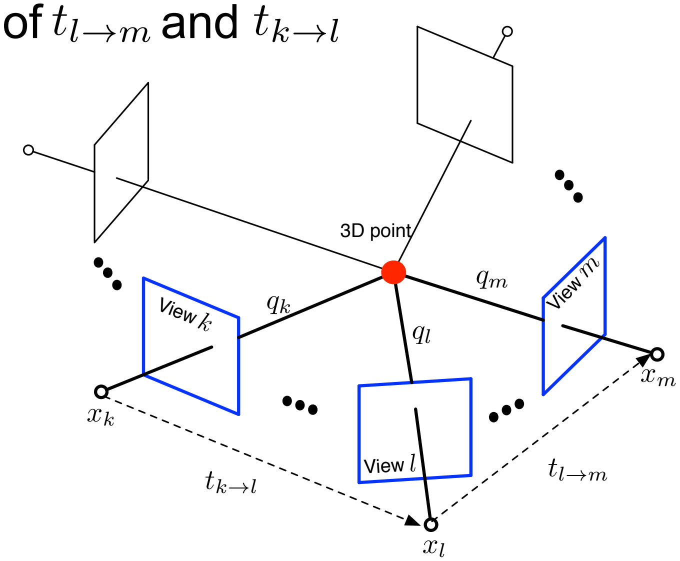
$t_{i\to j}$: translation from view *i* to view *j*

R_i : rotation from global frame to view *i*

$$g_{3v}\left(x_k, x_l, x_m, z_k, z_l, z_m\right) \doteq \left(q_l \times q_k\right) \cdot \left(q_m \times t_{l\to m}\right) - \left(q_k \times t_{k\to l}\right) \cdot \left(q_m \times q_l\right) = 0$$

Scale consistency

- Third equation – relates between the magnitudes of $t_{l\to m}$ and $t_{k\to l}$
- Necessary and sufficient conditions



- LBA cost function: $J_{LBA}(X) \doteq \sum_i^{N_h} \left\| h_i(X, Z) \right\|_{\Sigma_i}^2$

$$h_i \in \{g_{2v}, g_{3v}\}$$

V. Indelman, P. Gurfil, E. Rivlin, H. Rotstein, “Real-Time Vision-Aided Localization and Navigation Based on Three-View Geometry”, *IEEE Transactions on Aerospace and Electronic Systems*, 2012

Vision Only : Light Bundle Adjustment (LBA)

- LBA cost function: $$J_{LBA}(X) \doteq \sum_{i}^{N_h} \|h_i(X,Z)\|_{\Sigma_i}^2$$
  - h_i : i-th multi-view constraint
    - Involves several views and the corresponding image observations
  - Σ_i : An equivalent covariance $\Sigma_i = A_i \Sigma A_i^T$
  - A_i : Jacobian with respect to image observations

Number of optimized variables: $6N + 3M \Rightarrow 6N$

- Multi-view constraints - Different formulations in literature
  - Epipolar geometry, trifocal tensors, quadrifocal tensors etc.
  - Independent relations exist only between up to three cameras [Ma et al., 2004]
  - Here, three-view constraints formulation is used
    - Originally developed for navigation aiding **[Indelman et al., TAES 2012]**

Pre-Integrated IMU Factors

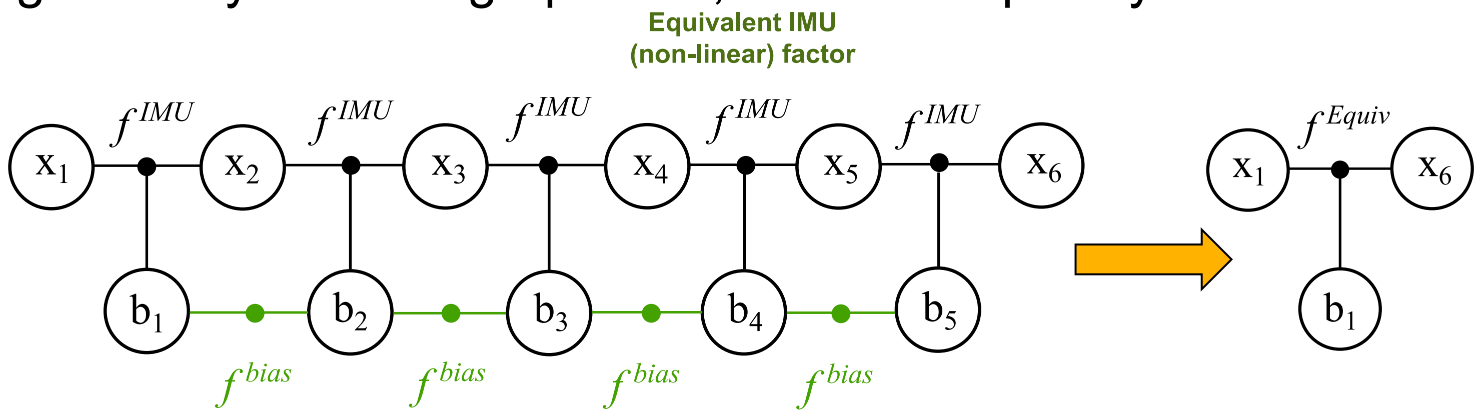
[Lupton et al., TRO 2012]

- Pre-integrate IMU measurements and insert equivalent factors only when inserting new LBA factors into the graph

$$\Delta x_{i\rightarrow j} \doteq \left\{\Delta p_{i\rightarrow j}, \Delta v_{i\rightarrow j}, R^i_j\right\} = \eta\left(Z^{IMU}_{i\rightarrow j}, b_i\right),$$

$$f^{Equiv}\left(x_j, x_i, b_i\right) \doteq \exp\left(-\frac{1}{2}\left\|x_j - h^{Equiv}\left(x_i, b_i, \Delta x_{i\rightarrow j}\right)\right\|^2_{\Sigma}\right)$$

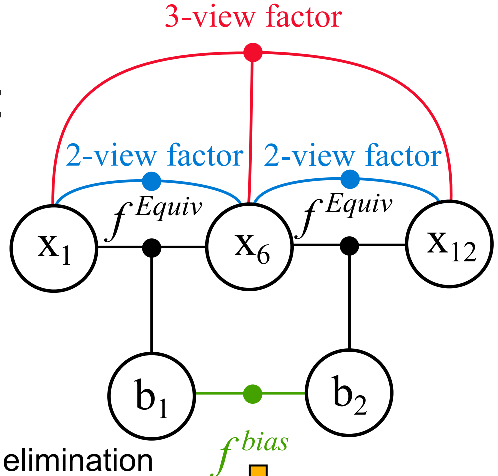
- Components of $\Delta x_{i\rightarrow j}$ are expressed in body-frame, not navigation frame, which allows relinearization of the factor without repeated computation
- Significantly reduces graph size, and subsequently time for elimination.



Todd Lupton and Salah Sukkarieh, “Visual-Inertial-Aided Navigation for High-Dynamic Motion in Built Environments Without Initial Conditions ”, *IEEE Transactions on Robotics,* 2012

Second Component - Incremental Inference [Kaess et al., 2012]

- When adding new variables\factors, calculations can be **reused**
  - Factorization can be updated (and not re-calculated from scratch)
- Example:



Linearization

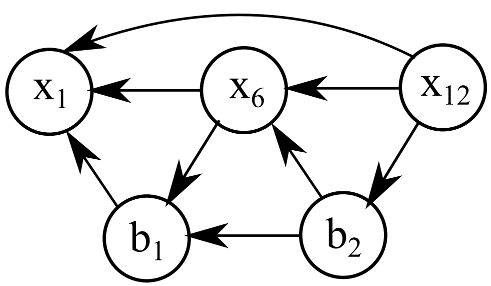
Jacobian matrix

$$A = \begin{array}{c} \begin{bmatrix} \times & \times & & & \\ & \times & \times & & \\ \times & \times & \times & & \\ \times & \times & & \times & \\ & \times & \times & & \times \\ & & & \times & \times \end{bmatrix} \\ \begin{array}{ccccc} x_1 & x_6 & x_{12} & b_1 & b_2 \end{array} \end{array}$$

Factorization

Linearization and elimination

Elimination order
$x_1, b_1, x_6, b_2, x_{12}$

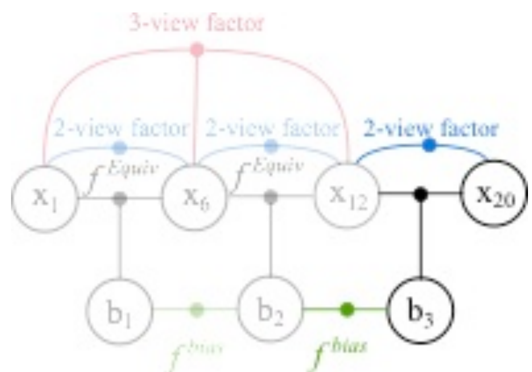


Factorized Jacobian matrix

$$R = \begin{array}{c} \begin{bmatrix} \times & \times & \times & & \times \\ & \times & \times & \times & \\ & & \times & \times & \times \\ & & & \times & \times \\ & & & & \times \end{bmatrix} \\ \begin{array}{ccccc} x_1 & b_1 & x_6 & b_2 & x_{12} \end{array} \end{array}$$

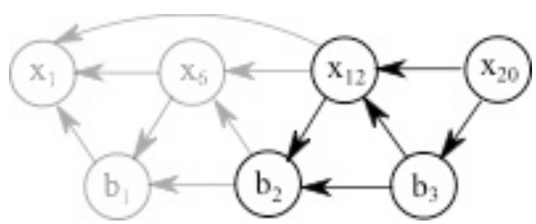
Incremental Inference in iLBA (Cont.)

- Example: New camera and factors are added



Linearization

$$A = \begin{array}{c} \left[\begin{array}{ccccccc}
\times & \times & & & & & \\
& \times & \times & & & & \\
\times & \times & \times & & & & \\
\times & \times & & \times & & & \\
& \times & \times & & \times & & \\
& & & \times & \times & & \\
& & \times & & & \times & \times \\
& & \times & & & \times & \\
& & & \times & & & \times
\end{array}\right] \\
\begin{array}{ccccccc} x_1 & x_6 & x_{12} & b_1 & b_2 & x_{20} & b_3 \end{array}
\end{array}$$

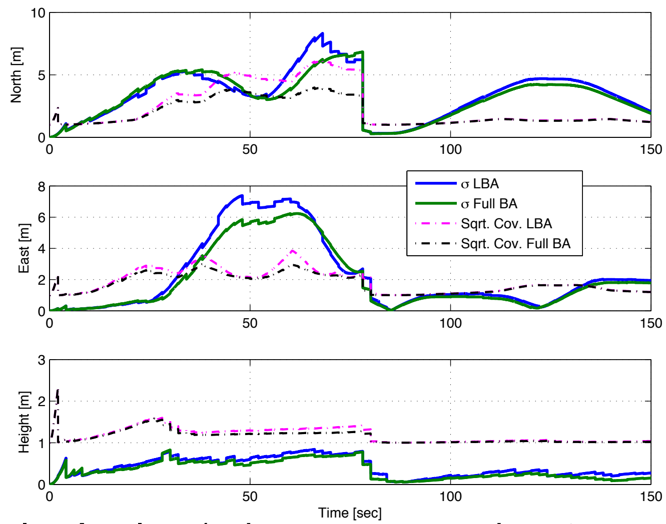


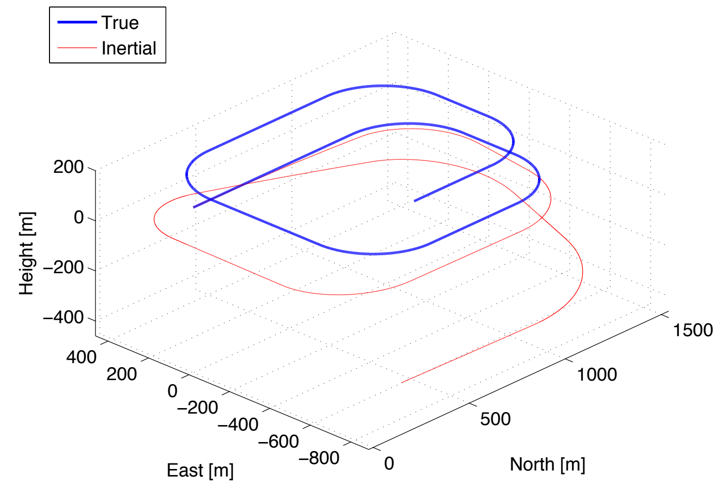
$$R = \begin{array}{c} \left[\begin{array}{ccccccc}
\times & \times & \times & & \times & & \\
& \times & \times & \times & & & \\
& & \times & \times & \times & & \\
& & & \times & \times & \times & \\
& & & & \times & \times & \times \\
& & & & & \times & \times \\
& & & & & & \times
\end{array}\right] \\
\begin{array}{ccccccc} x_1 & b_1 & x_6 & b_2 & x_{12} & b_3 & x_{20} \end{array}
\end{array}$$

- What should be re-calculated?
  - Nodes in all paths that lead from the last-eliminated node to nodes involved in new factors
  - Efficiently calculated using Bayes tree [Kaess et al., 2012]

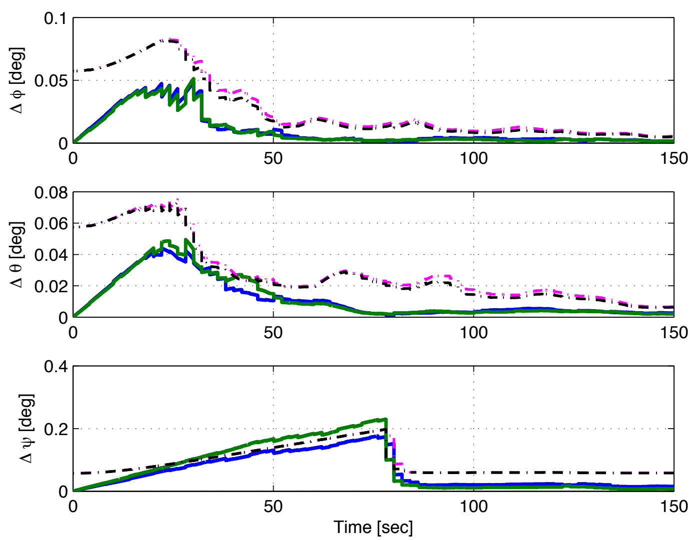
iLBA for Robotics – Monte-Carlo Study

Position 1-sigma errors and sqrt. covariance

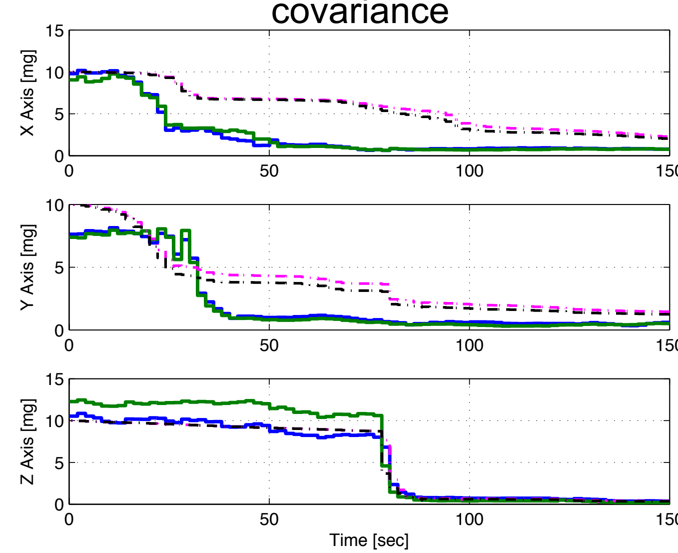




Euler Angles 1-sigma errors and sqrt. covariance



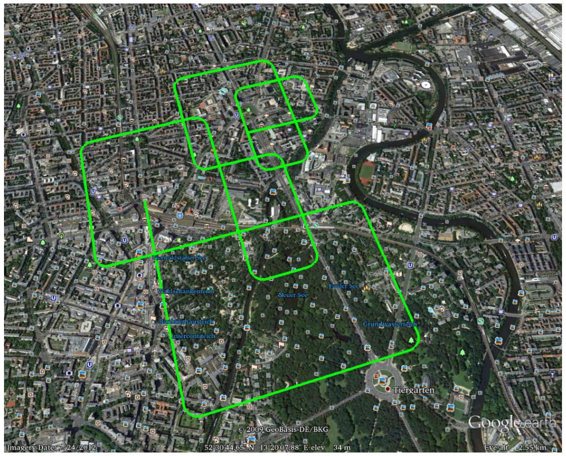
Acceleration bias 1-sigma errors and sqrt. covariance



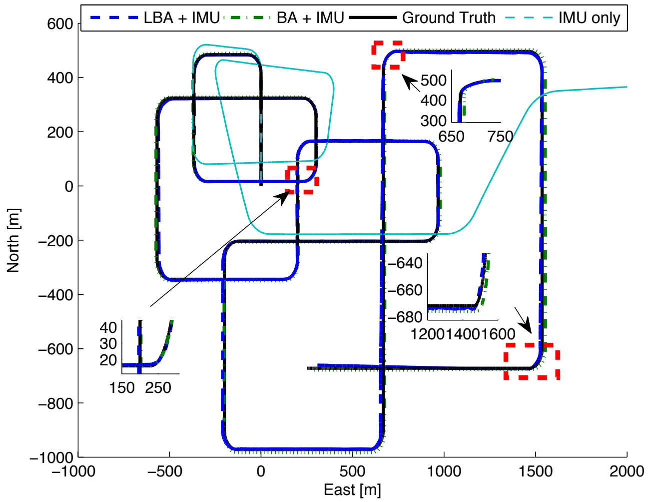
iLBA for Robotics – Simulation Results

- Scenario:
  - Single camera
  - IMU

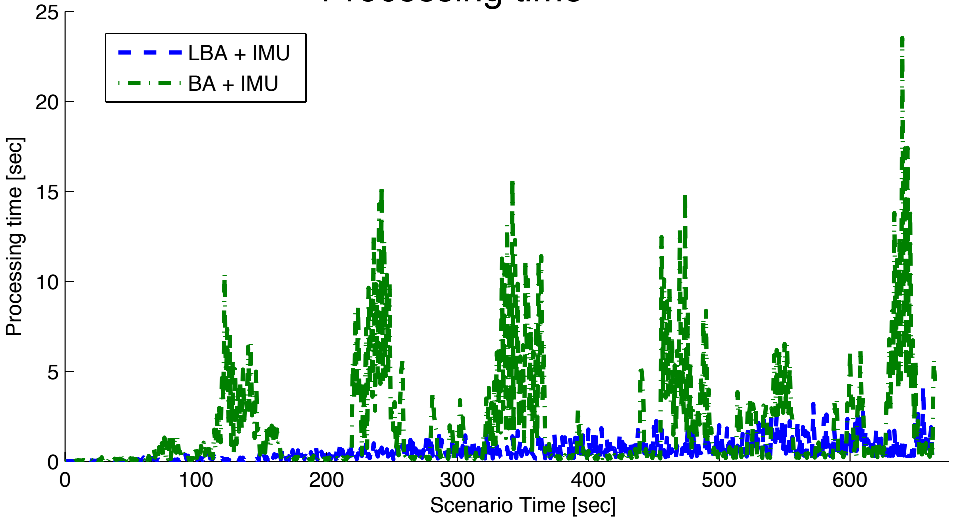
Scenario (GE for illustration)



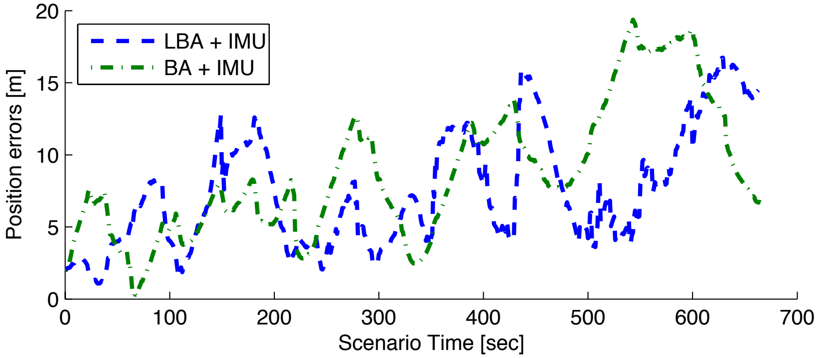
Estimated trajectory



Processing time



Position estimation errors



Georgia Institute of Technology

Robotics@GT & Intelligent Machines

Conclusions

- Algebraic elimination of 3D points significantly reduces the size of the optimization problem and provides speed up to online robot navigation
- Use of pre-integration methods for high-frequency inertial measurements also reduces the size of the problem
- Accuracy is similar to full SLAM
- At least 2-3.5x speed up in computation time

- Code and datasets are available from the author's website
  - http://www.cc.gatech.edu/~vindelma