Navigation Performance Enhancement using Online Mosaicking

Vadim Indelman

Supervisors: Prof. Pini Gurfil, AE, Technion

Prof. Ehud Rivlin, CS, Technion

Advisor: Dr. Hector Rotstein, Rafael

Nov. 2010

Introduction

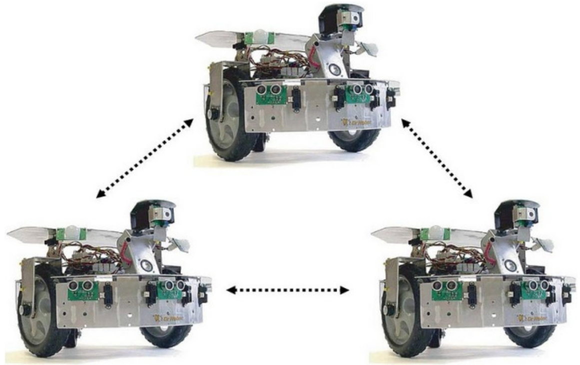
Navigation: calculation of position, velocity and attitude over time

Aerial, underwater, ground

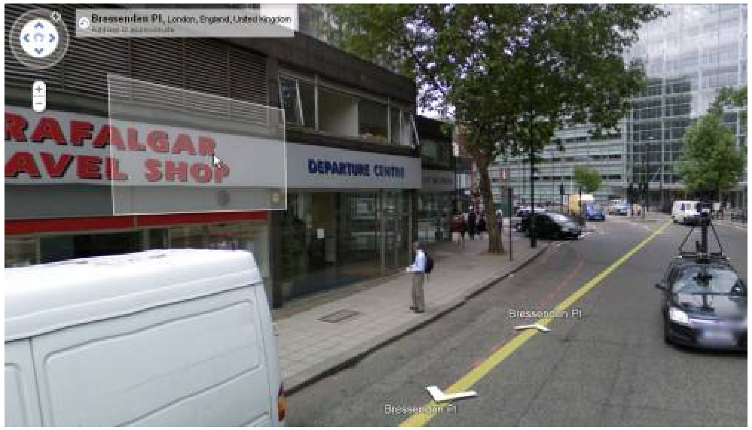
Space

Other planets

Multi robot



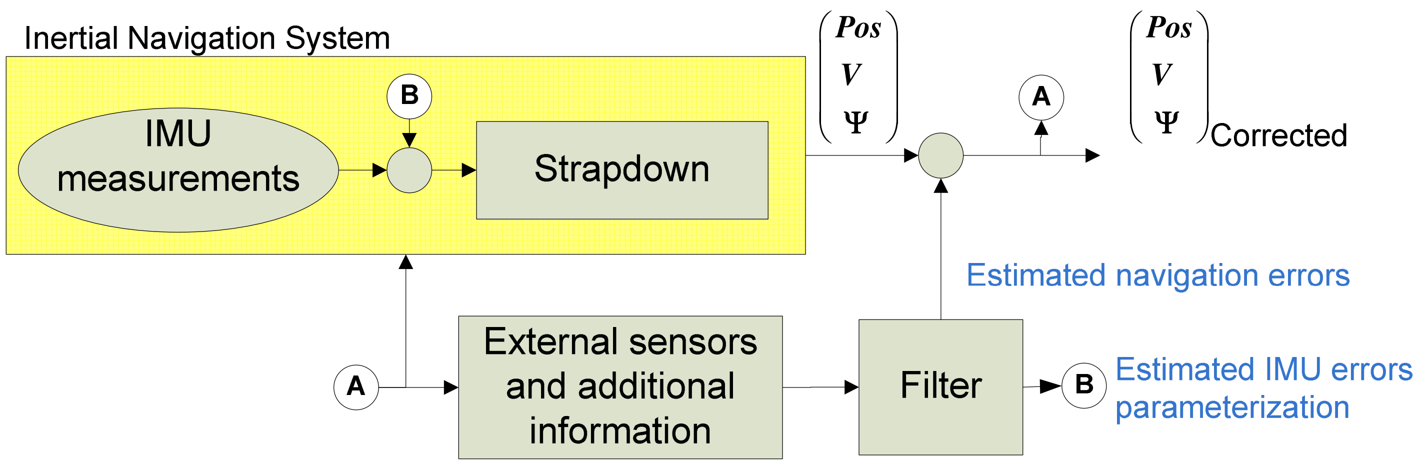
Consumer applications



Autonomous

Navigation Aiding Concept

- Inertial navigation solution is calculated based on measurements of inertial sensors (or other dead reckoning sensors)
- Imperfectness of these sensors leads to developing navigation errors
- Use external sensors to estimate navigation errors and correct navigation solution
  - Estimate also parameterization of inertial measurement unit (IMU) errors, and correct all subsequent IMU measurements



- Most common: GPS

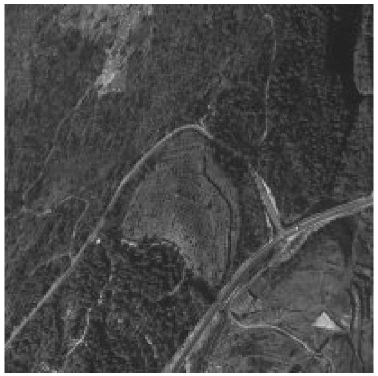
Research Concept

- The GPS is unavailable or unreliable when operating
  - Indoors, underwater, in urban environments, and over other planets
- This research – use images captured during motion for navigation aiding
- Research setup
  - Platform is equipped only with an inertial navigation system (INS) and a single (gimbaled) camera
    - No other sensors, no a-priori information (except for initial conditions, camera calibration parameters)
  - The images captured during motion are stored in a repository, or/and used for constructing mosaic images
- Objective:
  - Utilize mosaic construction and stored imagery for navigation aiding
  - Computational efficiency

Mosaic

- Constructed based on camera-captured images
- Represents the observed-so-far environment
- Encodes information about the platform's navigation history

Source images



Mosaic image

Navigation-Aiding Algorithms

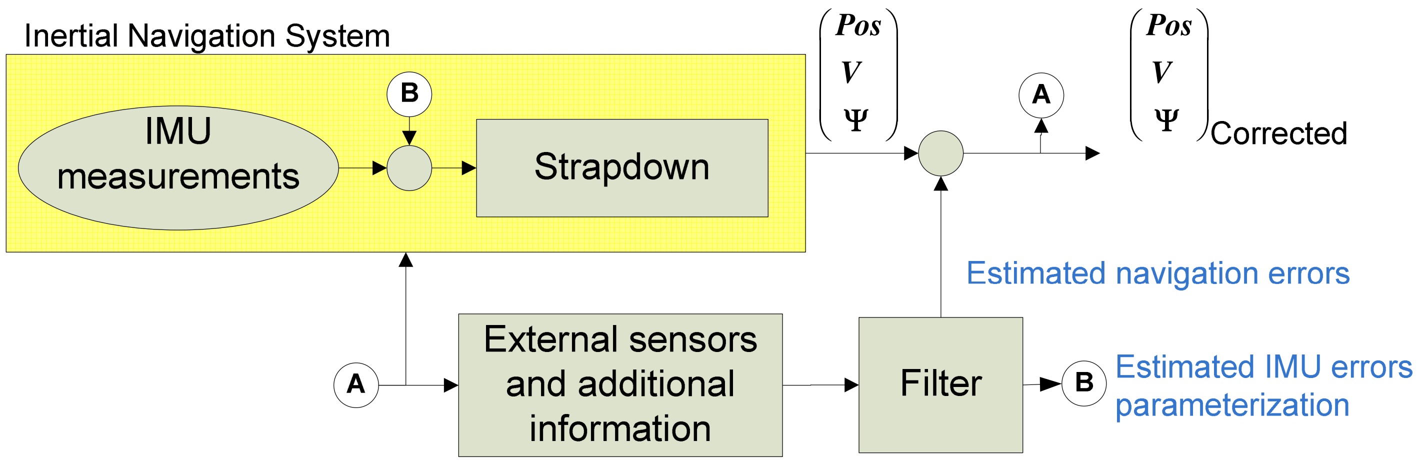
1 Navigation Aiding Based on **Coupled Online Mosaicking and Camera Scanning**

2 Vision-Aided Navigation Based on **Three-View Geometry**

3 **Distributed** Vision-Aided **Cooperative Navigation** Based on **Three-View Geometry**

3a **Calculating Cross-Covariance for a General Multi-Platform Measurement Model**

Inertial Navigation Errors Model



- State vector (navigation errors, IMU error parameterization)

$$\boldsymbol{X} \equiv \begin{bmatrix} \Delta \boldsymbol{P}^T & \Delta \boldsymbol{V}^T & \Delta \boldsymbol{\Psi}^T & \boldsymbol{d}^T & \boldsymbol{b}^T \end{bmatrix}^T$$

- Process model (Discrete)

$$\boldsymbol{X}(t_b) = \Phi_{t_a \to t_b} \boldsymbol{X}(t_a) + \boldsymbol{\omega}_{t_a \to t_b}$$

- How can $\mathbf{X}$ be estimated only based on INS and incoming imagery?

1

Navigation Aiding Based on Coupled Online Mosaicking and Camera Scanning

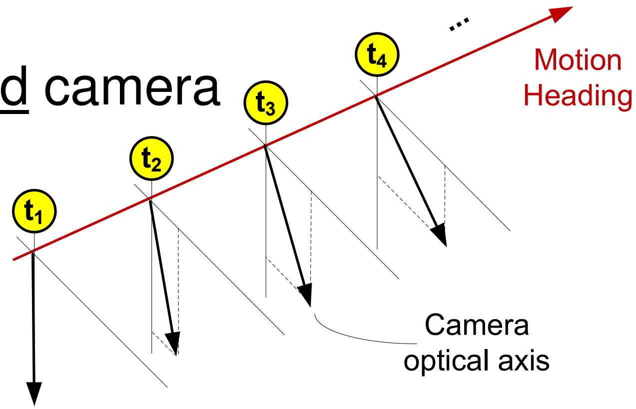
“Navigation Aiding Based on Coupled Online Mosaicking and Camera Scanning’’, AIAA *JGCD,* vol. 33, no. 6, 2010, p. 1866-1882
“Real-Time Mosaic-Aided Aerial Navigation: I. Motion Estimation’’, *AIAA GNC Conference,* 2009
“Real-Time Mosaic-Aided Aerial Navigation: II. Sensor Fusion’’, *AIAA GNC Conference,* 2009

Motivation and Related Work

- Objective
  - Vision-based navigation aiding in challenging scenarios
    - Narrow field-of-view camera
    - Low texture scenes
    - Poor weather conditions
  - Utilize online mosaicking and gimbaled camera scanning procedures
- Related Work
  - Motion estimation + online mosaicking: “Improving Vision-based Planar Motion Estimation for Unmanned Aerial Vehicles through online Mosaicking”, Caballero F. et. al., 2006
  - Epipolar constraints + INS: “Epipolar Constraints for Vision-Aided Inertial Navigation”, Diel D. et. al., 2005
  - SLAM: “6DoF SLAM aided GNSS/INS Navigation in GNSS Denied and Unknown Environments”, Kim J. and Sukkarieh S., 2005

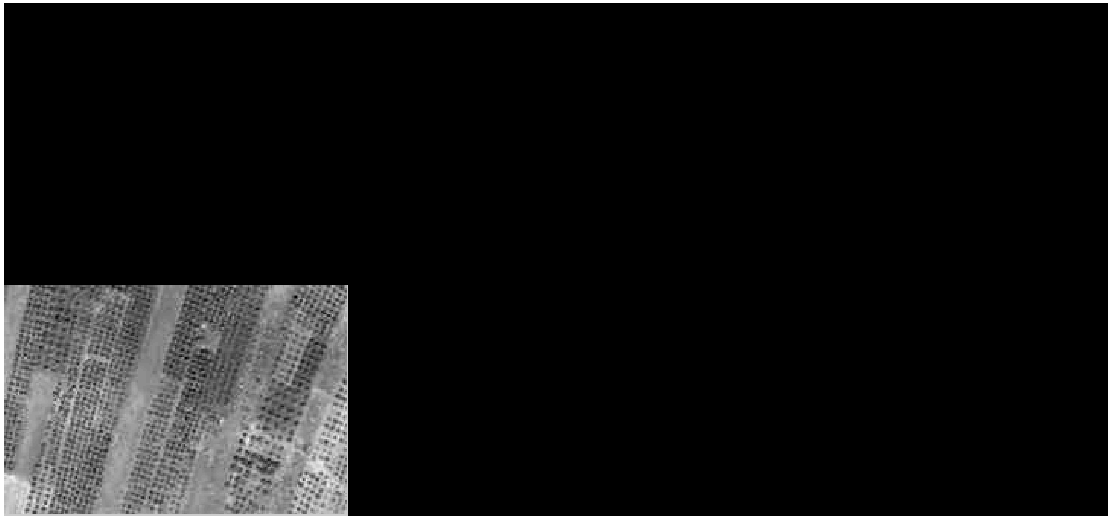
Main Idea

- Platform equipped only with an INS and a <u>gimbaled</u> camera
- Couple between
  - Camera scanning
  - Online mosaicking
- Construct mosaic based on camera-captured imagery
- Mosaic-based motion estimation
- Update INS



Camera scanning image sequence (from Google Earth)

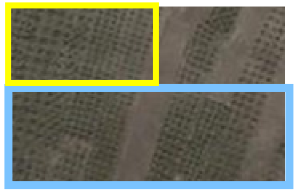
Mosaic image construction



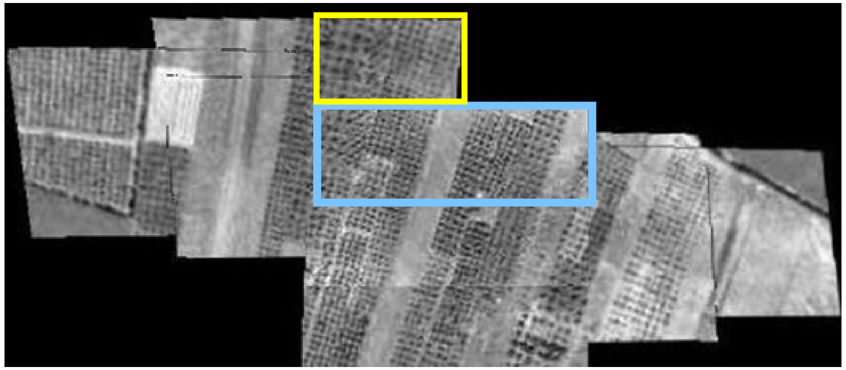
Main Idea (cont.)

Additional area
Original area

- Motion estimation between current image and previous mosaic image
  - Based on the homography model
- Increased overlapping region
  - Additional matching features
    - Same quality?
- Improved motion estimation in challenging scenarios
  - Narrow field-of-view camera
  - Low-texture scenes
- BUT:
  - Translation is estimated up to scale ⇨ **Allows reducing navigation errors <u>normal</u> to the motion heading**



Current image



Previous <u>**mosaic**</u> image

What happens when the same scene is observed by several (>2) views?

2

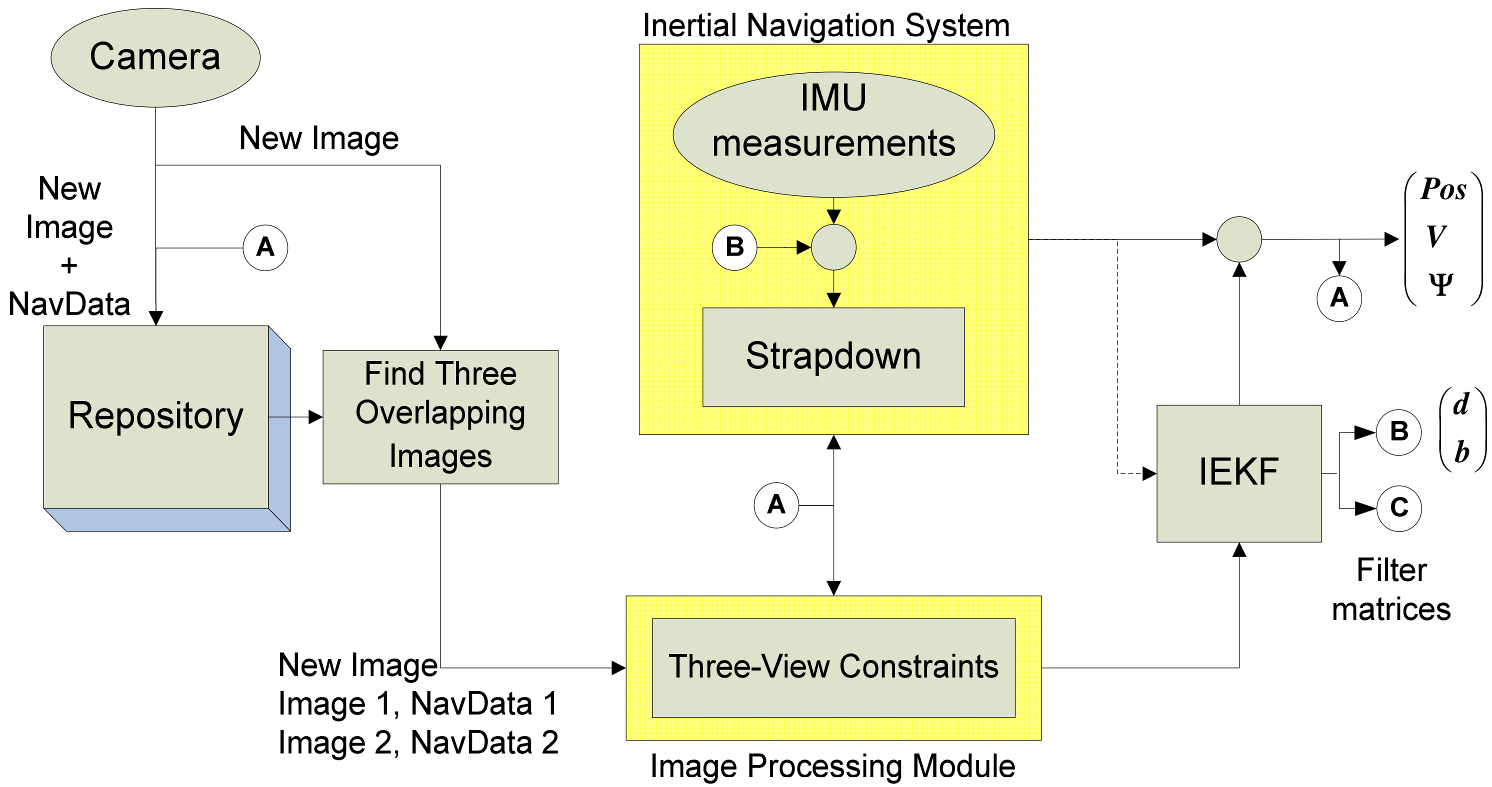
Vision-Aided Navigation Based on Three-View Geometry

“Real-Time Vision-Aided Localization and Navigation Based on Three-View Geometry’’, *IEEE TAES,* submitted
“Mosaic Aided Navigation: Tools, Methods and Results’’, *IEEE\ION PLANS Conference*, 2010

Motivation and Related Work

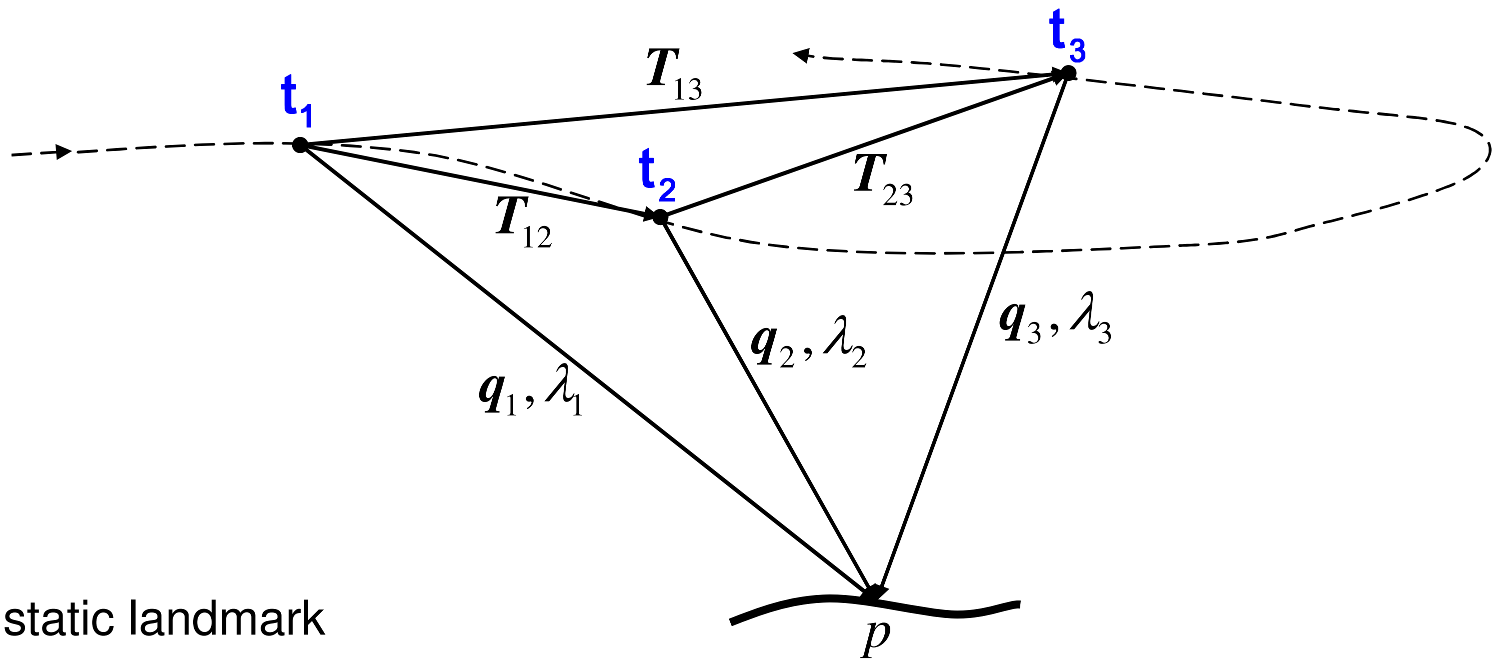
- Objective
  - Position and velocity update in all axes
    - Setup: INS, camera, online-constructed repository
  - Real time navigation aiding
    - Efficiently handle **loop scenarios**
- Related work
  - Smoothing + Range: "Improved Real-Time Video Mosaicking of the Ocean Floor", Fleischer D. et. al., 1995
  - SLAM: "6DoF SLAM aided GNSS/INS Navigation in GNSS Denied and Unknown Environments", Kim J. and Sukkarieh S., 2005
  - Pinhole projection + Multi-view + INS: "A Multi-State Constraint Kalman Filter for Vision-Aided Inertial Navigation", Mourikis A. and Roumeliotis S., 2007
  - Multiple-view + Bundle adjustment: "A Dual-Layer Estimator Architecture for Long-term Localization", Mourikis A.I. and Roumeliotis S.I., 2008
  - Trifocal tensor + Motion estimation: "Recursive Camera-Motion Estimation With the Trifocal Tensor", Yu Y.K., et. al., 2006

Concept



- Coordinate systems
  - L - Local Level Local North (LLLN)
  - C - Camera

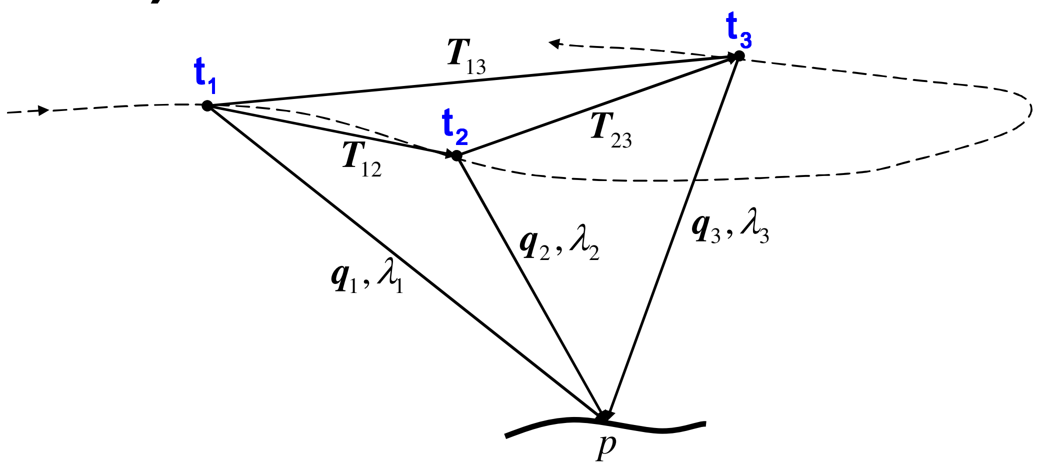
Three View Geometry



- p - static landmark
- $\boldsymbol{q}$ - line of sight (LOS)
- λ - scale parameter, s.t. $\|\lambda \boldsymbol{q}\|$ is the range to landmark
- $\boldsymbol{T}_{ij}$ - translation from i to j

Three View Geometry (cont.)

- Position of a static landmark p relative to camera position at t_1 :



$$\lambda_1 \boldsymbol{q}_1 = \boldsymbol{T}_{12} + \lambda_2 \boldsymbol{q}_2$$

$$\lambda_1 \boldsymbol{q}_1 = \boldsymbol{T}_{12} + \boldsymbol{T}_{23} + \lambda_3 \boldsymbol{q}_3$$

- Matrix formulation

$$\underbrace{\begin{bmatrix} \boldsymbol{q}_1 & -\boldsymbol{q}_2 & \boldsymbol{0}_{3\times1} & -\boldsymbol{T}_{12} \\ \boldsymbol{0}_{3\times1} & \boldsymbol{q}_2 & -\boldsymbol{q}_3 & -\boldsymbol{T}_{23} \end{bmatrix}_{6\times4}}_{A} \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \\ 1 \end{bmatrix}_{4\times1} = \boldsymbol{0}_{6\times1}$$

- Note:
  - Range parameters are unknown
  - LOS and translation vectors are expressed in LLLN of t_2

$$\rightarrow \quad rank(A) < 4$$

Three View Geometry (cont.)

- Theorem: *rank(A)<4* if and only if all of the following conditions are satisfied

$$q_1^T \left(T_{12} \times q_2\right) = 0$$

$$q_2^T \left(T_{23} \times q_3\right) = 0$$

$$\left(q_2 \times q_1\right)^T \left(q_3 \times T_{23}\right) = \left(q_1 \times T_{12}\right)^T \left(q_3 \times q_2\right)$$

- First two equations – epipolar constraints
- Third equation – relates between the magnitudes of T_{12} and T_{23}
- Sufficient and necessary conditions
- Reformulating:

$$\Longrightarrow \quad \begin{aligned} \left[g^T\right]_{1\times3} T_{12} &= 0 \\ \left[f^T\right]_{1\times3} T_{23} &= 0 \\ \left[u^T\right]_{1\times3} T_{23} &= \left[w^T\right]_{1\times3} T_{12} \end{aligned} \qquad \text{where} \qquad \begin{aligned} g &= g\left(q_1, q_2\right) \\ f &= f\left(q_2, q_3\right) \\ u &= u\left(q_1, q_2, q_3\right) \\ w &= w\left(q_1, q_2, q_3\right) \end{aligned}$$

Three View Geometry (cont.)

- Multiple features
  - Matching pairs between 1st and 2nd view $\left\{ \boldsymbol{q}_{1_i}^{C_1}, \boldsymbol{q}_{2_i}^{C_2} \right\}_{i=1}^{N_{12}}$
  - Matching pairs between 2nd and 3rd view $\left\{ \boldsymbol{q}_{2_i}^{C_2}, \boldsymbol{q}_{3_i}^{C_3} \right\}_{i=1}^{N_{23}}$
  - Matching triplets between the three views $\left\{ \boldsymbol{q}_{1_i}^{C_1}, \boldsymbol{q}_{2_i}^{C_2}, \boldsymbol{q}_{3_i}^{C_3} \right\}_{i=1}^{N_{123}}$

$$\left[\boldsymbol{u}_i^T\right]_{1\times3} \boldsymbol{T}_{23} = \left[\boldsymbol{w}_i^T\right]_{1\times3} \boldsymbol{T}_{12} \qquad i = 1, \ldots, N_{123}$$

$$\left[\boldsymbol{f}_j^T\right]_{1\times3} \boldsymbol{T}_{23} = 0 \qquad j = 1, \ldots, N_{23}$$

$$\left[\boldsymbol{g}_k^T\right]_{1\times3} \boldsymbol{T}_{12} = 0 \qquad k = 1, \ldots, N_{12}$$

$$\Rightarrow \begin{bmatrix} U \\ F \\ 0 \end{bmatrix}_{N\times3} \boldsymbol{T}_{23} = \begin{bmatrix} W \\ 0 \\ G \end{bmatrix}_{N\times3} \boldsymbol{T}_{12}$$

$$N = N_{123} + N_{12} + N_{23}$$

Fusion with Navigation using Implicit Extended Kalman Filter (IEKF)

- Residual Measurement

$$z \equiv \begin{bmatrix} U \\ F \\ 0 \end{bmatrix}_{N\times 3} \boldsymbol{T}_{23} - \begin{bmatrix} W \\ 0 \\ G \end{bmatrix}_{N\times 3} \boldsymbol{T}_{12}$$

- Recall
  - All original LOS vectors are expressed in camera system of the appropriate view
  - $\boldsymbol{T}_{23}, \boldsymbol{T}_{12}$ are functions of $\boldsymbol{Pos}(t_3), \boldsymbol{Pos}(t_2), \boldsymbol{Pos}(t_1)$

$$\Downarrow$$

$$z = \boldsymbol{h}\left(\boldsymbol{Pos}(t_3), \Psi(t_3), \boldsymbol{Pos}(t_2), \Psi(t_2), \boldsymbol{Pos}(t_1), \Psi(t_1), \left\{\boldsymbol{q}_{1_i}^{C_1}, \boldsymbol{q}_{2_i}^{C_2}, \boldsymbol{q}_{3_i}^{C_3}\right\}\right)$$

Fusion with Navigation using IEKF (cont.)

- Recall: $$\boldsymbol{X} = \begin{bmatrix} \Delta\boldsymbol{P}^T & \Delta\boldsymbol{V}^T & \Delta\boldsymbol{\Psi}^T & \boldsymbol{d}^T & \boldsymbol{b}^T \end{bmatrix}^T$$
- Linearization

$$z(t_3,t_2,t_1) = \boldsymbol{h}\left(\boldsymbol{Pos}(t_3),\Psi(t_3),\boldsymbol{Pos}(t_2),\Psi(t_2),\boldsymbol{Pos}(t_1),\Psi(t_1),\left\{\boldsymbol{q}_{1_i}^{C_1},\boldsymbol{q}_{2_i}^{C_2},\boldsymbol{q}_{3_i}^{C_3}\right\}\right)$$
$$\cong H_3\boldsymbol{X}_3 + H_2\boldsymbol{X}_2 + H_1\boldsymbol{X}_1 + D\boldsymbol{v} + H.O.T.$$

- t_3 is the **current** time $\boxed{\boldsymbol{X}_i \equiv \boldsymbol{X}(t_i)}$
  - $\boldsymbol{X}_3$ needs to be estimated
  - $\boldsymbol{X}_2,\boldsymbol{X}_1$ are represented by the covariance matrices attached to images

$$\boldsymbol{P}_2 = \boldsymbol{E}\left[\tilde{\boldsymbol{X}}_2\tilde{\boldsymbol{X}}_2^T\right] \quad , \quad \boldsymbol{P}_1 = \boldsymbol{E}\left[\tilde{\boldsymbol{X}}_1\tilde{\boldsymbol{X}}_1^T\right]$$

- In a general case, $\boldsymbol{X}_3,\boldsymbol{X}_2$ and $\boldsymbol{X}_1$ may be correlated

Fusion with Navigation using IEKF (cont.)

- Since only X_3 is estimated, the Kalman gain is given by:

$$K = P_{X(t_3)z(t_3,t_2,t_1)} P^{-1}_{z(t_3,t_2,t_1)}$$

with

$$P_{X(t_3)z(t_3,t_2,t_1)} = P_3 H_3^T + P_{32} H_2^T + P_{31} H_1^T$$

$$P_{z(t_3,t_2,t_1)} = H_3 P_3 H_3^T + \begin{bmatrix} H_2 & H_1 \end{bmatrix} \begin{bmatrix} P_2 & P_{21} \\ P_{21}^T & P_1 \end{bmatrix} \begin{bmatrix} H_2 & H_1 \end{bmatrix}^T + DRD^T$$

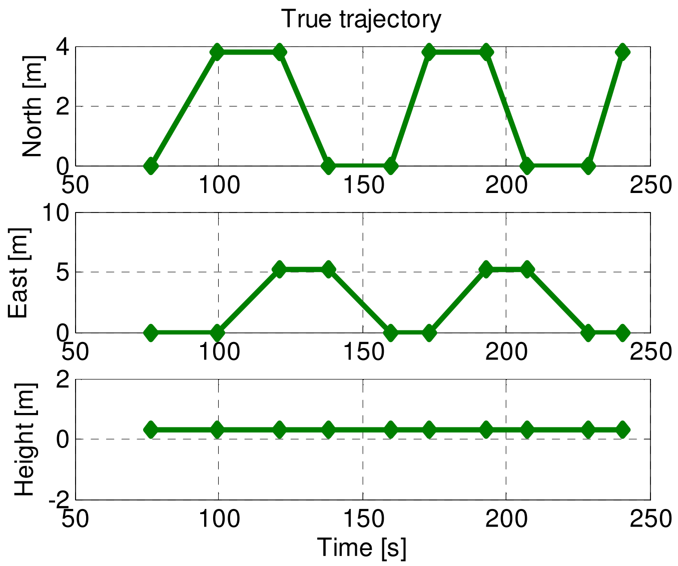
- Problem: how to calculate P_{31}, P_{32}, P_{21} when t_1, t_2 are unknown a-priori?
  - Inertial navigation is assumed between t_1 and t_2: $P_{21} = \Phi_{t_1 \to t_2} P_1$
  - P_{31}, P_{32} are neglected
    - Valid for $t_3 >> t_2, t_1$. e.g.: loop scenarios
- Otherwise, calculate as shown in the sequel

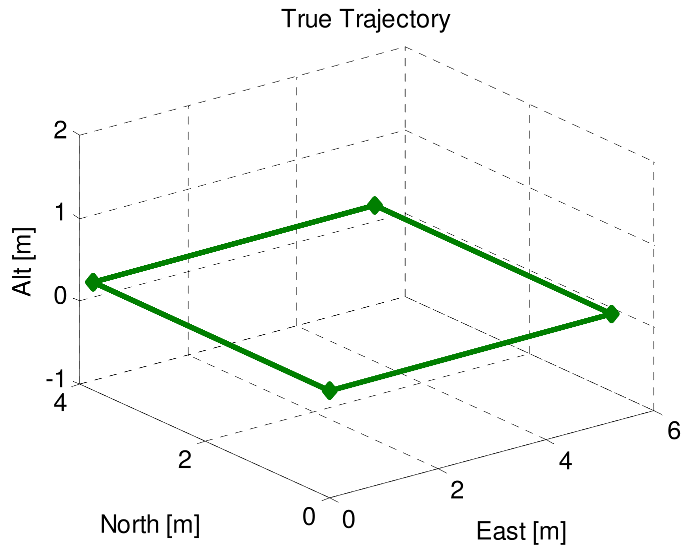
Results - Experiment

- Experiment Setup
  - An IMU and a camera were mounted on top of a ground vehicle
  - IMU\INS: Xsens MTi-G
  - Camera: Axis 207MW
- IMU data and captured images were stored and synchronized
  - IMU data @ 100Hz
  - Imagery data @ 15Hz
- The method was applied in two scenarios
  - Sequential updates
  - Loop updates

Results – Experiment (cont.)

True trajectory





Recorded imagery

Results – Experiment (cont.)

Example

Image 1

Image 2

Image 3

Matching Triplets

- Implementation details (simplified)
  -
  - s
  - Fundamental matrix model
    - The Fundamental matrix is not required elsewhere

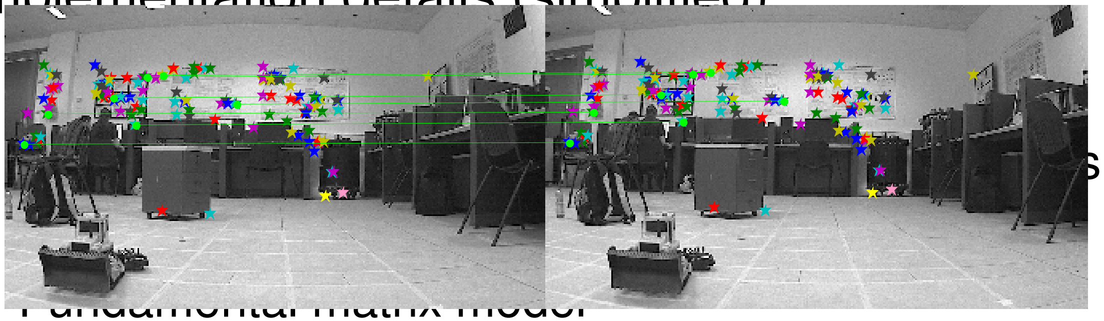


Image 1 Image 2

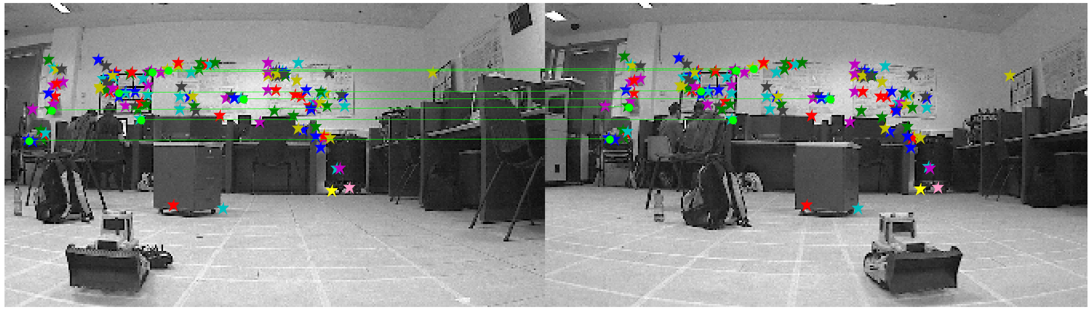
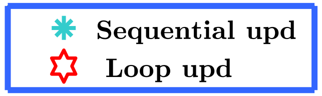


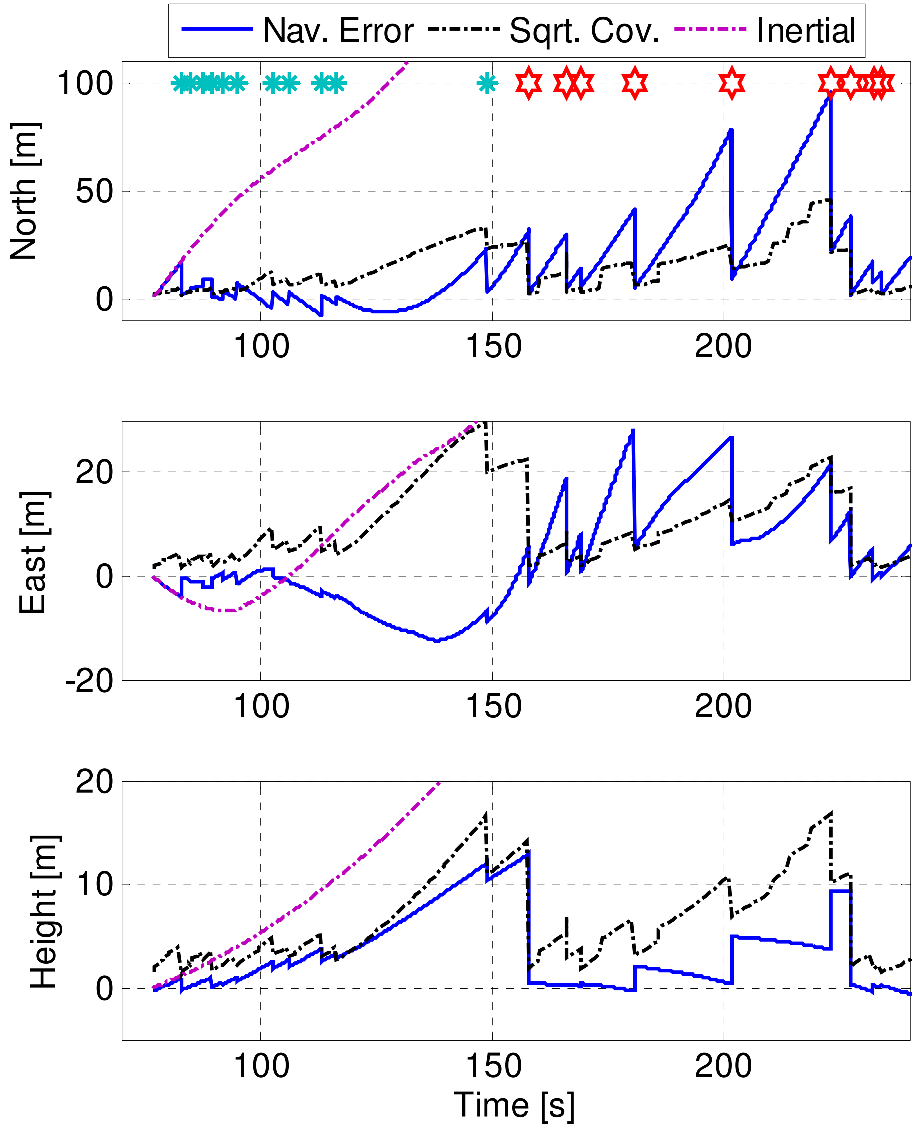
Image 2 Image 3

$$\left\{ \boldsymbol{q}_{1_i}^{C_1}, \boldsymbol{q}_{2_i}^{C_2}, \boldsymbol{q}_{3_i}^{C_3} \right\}_{i=1}^{N_{123}}, \left\{ \boldsymbol{q}_{1_i}^{C_1}, \boldsymbol{q}_{2_i}^{C_2} \right\}_{i=1}^{N_{12}}, \left\{ \boldsymbol{q}_{2_i}^{C_2}, \boldsymbol{q}_{3_i}^{C_3} \right\}_{i=1}^{N_{23}}$$

Results – Experiment (cont.)



Position errors



Conclusions

- Application of three-view constraints for navigation aiding
- New formulation of three-view constraints for a general static scene
- The method allows
  - Reduction of position and velocity errors in all axes to the levels of errors present while the first two images were captured
  - Reduction of attitude errors, partial estimation of bias
- Efficiently handle loop scenarios
- Reduced computational requirements for vision-aided navigation phase
  - Environment representation construction (e.g. mosaic) may be executed in a background process
- Various potential applications
  - Cooperative navigation - next

3 Distributed Vision-Aided Cooperative Navigation Based on Three-View Geometry

"Distributed Vision-Aided Cooperative Localization and Navigation based on Three-View Geometry'', *IEEE Aerospace Conference,* submitted

Motivation and Related Work

- Objective
  - Navigation update whenever the same scene is observed by different platforms
    - Not necessarily at the same time
    - The camera is not required to be aimed towards other platforms (in contrast to relative pose measurements)
  - Properly handle correlation terms involved in the fusion process

- Related work
  - Use some robots as landmarks: "Cooperative Positioning with Multiple Robots", Kurazume R. et. al., 1994
  - Relative pose measurements between pairs of robots: "Distributed Multirobot Localization", Roumeliotis S.I. and Bekey G.A., 2002
  - Direct & indirect encounters between pairs of robots, nonlinear optimization: "Multiple Relative Pose Graphs for Robust Cooperative Mapping", Kim B. et. al., 2010
  - Consistent information fusion: "Consistent Cooperative Localization", Bahr A. et. al., 2009

Setup

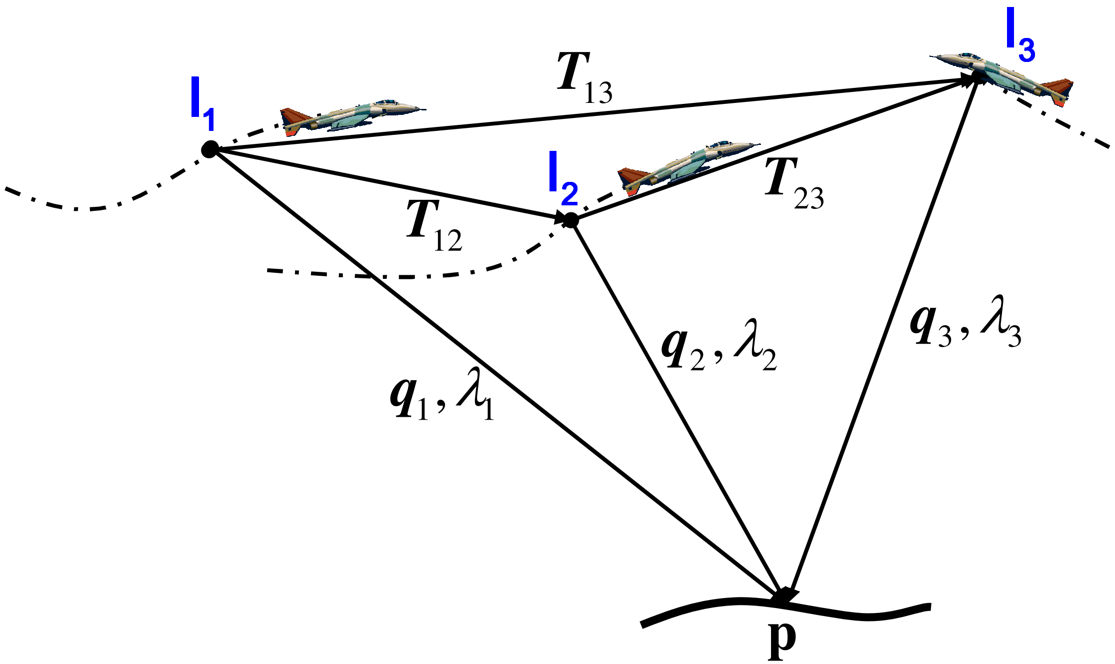
- Consider a group of cooperative platforms
- Each platform is equipped with its own
  - INS
  - Camera
  - Perhaps, additional sensors or a-priori information
- All\some platforms maintain a repository of stored images associated with navigation information
- The platforms are able to exchange navigation and imagery data
- Inertial navigation error of the i-th platform

$$\boldsymbol{X}_i\left(t_b\right)=\Phi^i_{t_a\to t_b}\boldsymbol{X}_i\left(t_a\right)+\boldsymbol{\omega}^i_{t_a\to t_b}$$

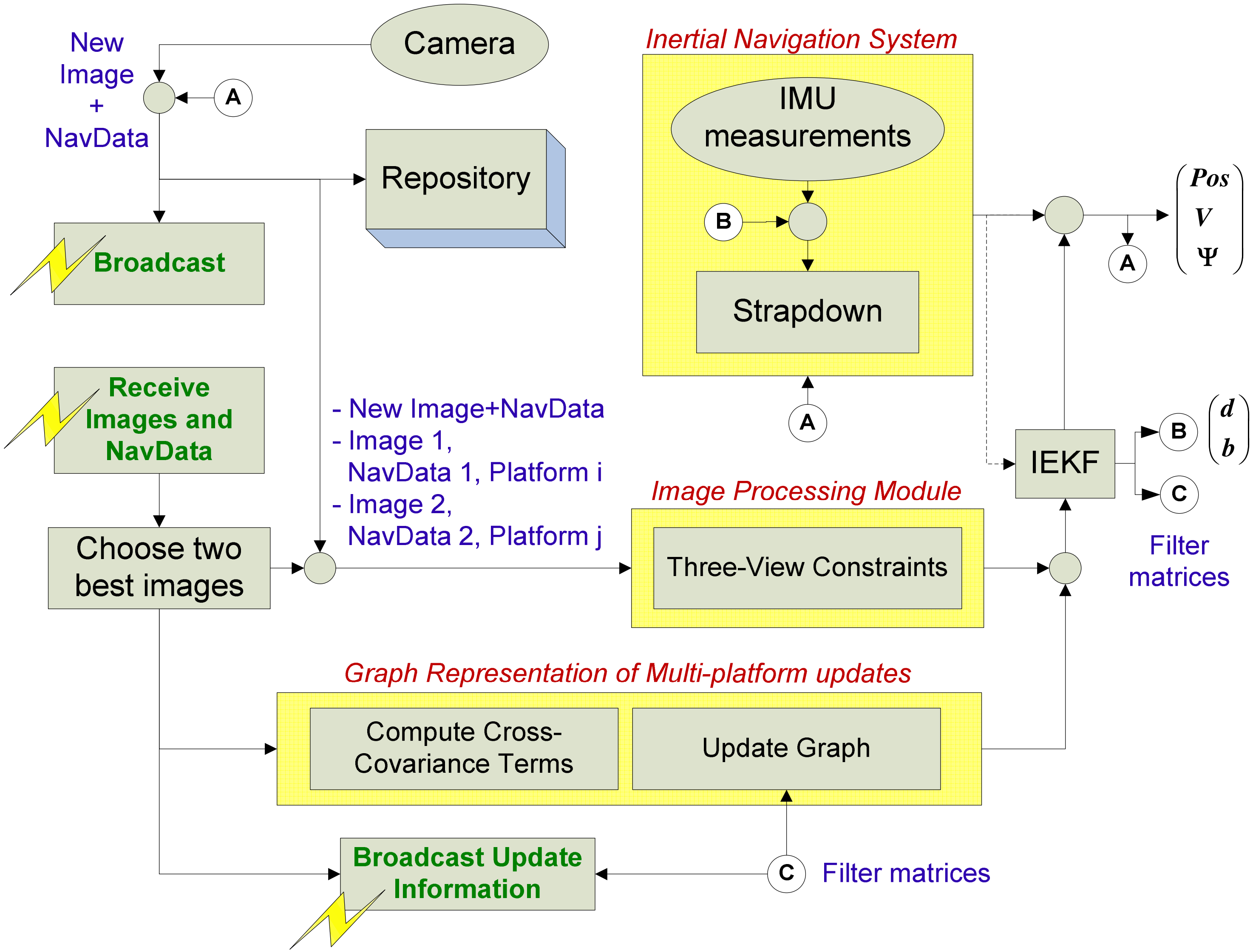
- Each platform maintains a local graph, required for correlation calculation

Three-view Geometry – Several Platforms

- This time, each image may be captured by a different platform
- The images are not necessarily captured at the same time
  - Some images may be stored in repositories and retrieved upon demand



Overview


New Image + NavData
Camera
A
Repository
Broadcast
Inertial Navigation System
IMU measurements
B
Strapdown
A
Pos
V
Ψ
A
Receive Images and NavData
- New Image+NavData
- Image 1, NavData 1, Platform i
- Image 2, NavData 2, Platform j
IEKF
d
b
B
C
Image Processing Module
Three-View Constraints
Filter matrices
Choose two best images
Graph Representation of Multi-platform updates
Compute Cross-Covariance Terms
Update Graph
Broadcast Update Information
C
Filter matrices

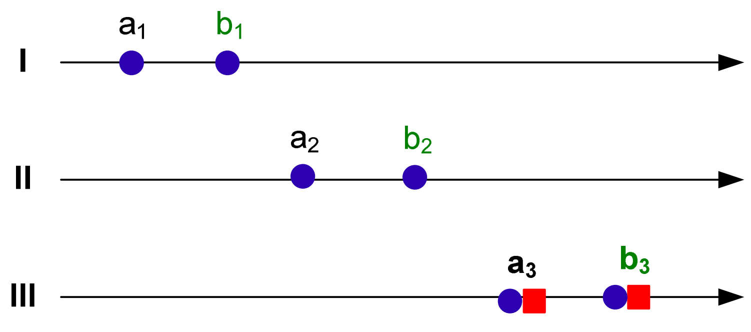
Fusion with Navigation

- Back to the residual measurement model

$$z \cong H_3 \boldsymbol{X}_{\mathbf{III}}(t_3) + H_2 \boldsymbol{X}_{\mathbf{II}}(t_2) + H_1 \boldsymbol{X}_{\mathbf{I}}(t_1) + D\boldsymbol{v} + H.O.T.$$

  - $\boldsymbol{X}_{\mathbf{III}}(t_3), \boldsymbol{X}_{\mathbf{II}}(t_2), \boldsymbol{X}_{\mathbf{I}}(t_1)$ represent navigation errors of <u>different</u> platforms at <u>different</u> time instances.
    - **None of these are known a-priori**
  - Theoretically, all the participating platforms can be updated

- Example:
  - Only platform III is updated
  - 2 three-view measurements



Fusion with Navigation (cont.)

- The measurement update step involves cross-covariance terms
  - E.g., if only platform III is updated: $K = P_{X_{III}(t_3)z(t_3,t_2,t_1)} P^{-1}_{z(t_3,t_2,t_1)}$
    - with

$$P_{X(t_3)z(t_3,t_2,t_1)} = P_3 H_3^T + P_{32} H_2^T + P_{31} H_1^T$$

$$P_{z(t_3,t_2,t_1)} = H_3 P_3 H_3^T + \begin{bmatrix} H_2 & H_1 \end{bmatrix} \begin{bmatrix} P_2 & P_{21} \\ P_{21}^T & P_1 \end{bmatrix} \begin{bmatrix} H_2 & H_1 \end{bmatrix}^T + DRD^T$$

    - where

$$P_{ij} \equiv E\left[\tilde{X}_i(t_i) \tilde{X}_j^T(t_j) \right]$$

- Maintaining all the possible cross-covariance terms – impractical
  - In contrast to relative pose measurements
- Therefore: either neglect, or calculate upon-demand

3 Distributed Vision-Aided Cooperative Navigation Based on Three-View Geometry

3a **Calculating Cross-Covariance for a General Multi-Platform Measurement Model**

“Graph-based Distributed Cooperative Navigation”, *IEEE ICRA,* submitted

General Multi-Platform Measurement Model

- Assume a general Multi-Platform (MP) measurement model that involves information from ***r*** platforms

$$\boldsymbol{z} \cong \sum_{i=1}^{r} H_i \boldsymbol{X}_i(t_i) + D_i(t_i)\boldsymbol{v}_i(t_i)$$

- Objective: Calculate $\boldsymbol{E}\left[\tilde{\boldsymbol{X}}_i(t_i)\tilde{\boldsymbol{X}}_j^T(t_j)\right]$

Concept

1. Store covariance and cross-covariance terms from all the past Multi-Platform (MP) measurement updates
2. Express $\tilde{\boldsymbol{X}}_i(t_i)$ and $\tilde{\boldsymbol{X}}_j(t_j)$ according to the history of MP measurement updates
3. Calculate $\boldsymbol{E}\left[\tilde{\boldsymbol{X}}_i(t_i)\tilde{\boldsymbol{X}}_j^T(t_j)\right]$ based on expressions from step 2.

- Algorithm objective: Automation of the above for general scenarios using graph representation

General Multi-Platform Measurement Model

- Assume a general Multi-Platform (MP) measurement model that involves information from ***r*** platforms

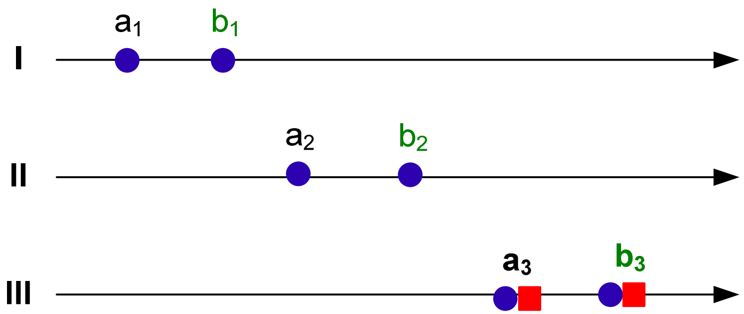
$$\boldsymbol{z} \cong \sum_{i=1}^{r} H_i \boldsymbol{X}_i(t_i) + D_i(t_i)\boldsymbol{v}_i(t_i)$$

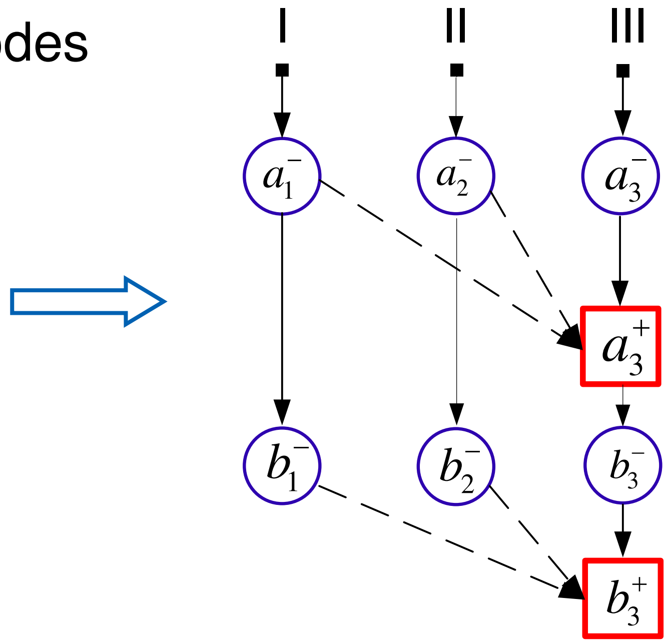
- Objective: Calculate $\boldsymbol{E}\left[\tilde{\boldsymbol{X}}_i(t_i)\tilde{\boldsymbol{X}}_j^T(t_j)\right]$
- Represent all MP updates executed so far in a directed acyclic graph (DAG)
  - Acyclic graph is assumed ⟹ In a general scenario, allows updating only platforms that contribute their current (and not past) information
  - For simplicity, we consider updating only *one* such platform
  - Specific scenarios exist in which all the involved platforms can be updated
- A-posteriori estimation error of the updated platform (denoted by ***q***)

$$\tilde{\boldsymbol{X}}_q^+ = \left(I - K_q H_q\right)\tilde{\boldsymbol{X}}_q^- - K_q \sum_{i=1, i\neq q}^{r} H_i \tilde{\boldsymbol{X}}_i^- - K_q \sum_{i=1}^{r} D_i \boldsymbol{v}_i$$

Graph Representation

- Each platform maintains its own DAG G=(V,E)
- A-priori and a-posteriori covariance and cross-covariance matrices are stored in G after each MP update
- Two node types in G:
  - Nodes representing (a-priori) information participating in an MP measurement
    - E.g., an image and navigation data obtained from some platform
  - Update-event node, representing a-posteriori estimate of the updated platform
- Each MP update is represented by ***r+1*** nodes





Graph Representation (cont.)

Each node can be connected to another node by a

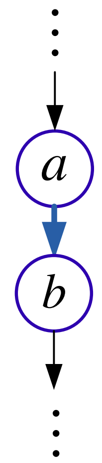
- **Transition relation** between node a and b

$$\tilde{\boldsymbol{X}}_b = \Phi^i_{t_a \to t_b} \tilde{\boldsymbol{X}}_a + \boldsymbol{\omega}^i_{t_a \to t_b}$$

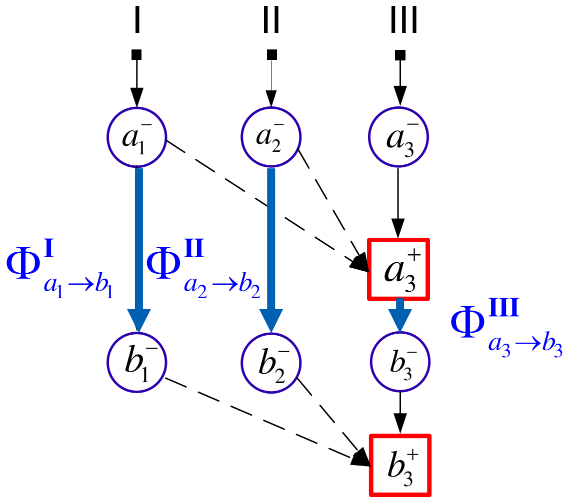
  – Arc weight: $w(a,b) \equiv \Phi^i_{t_a \to t_b}$

  – The process noise covariance is also stored:

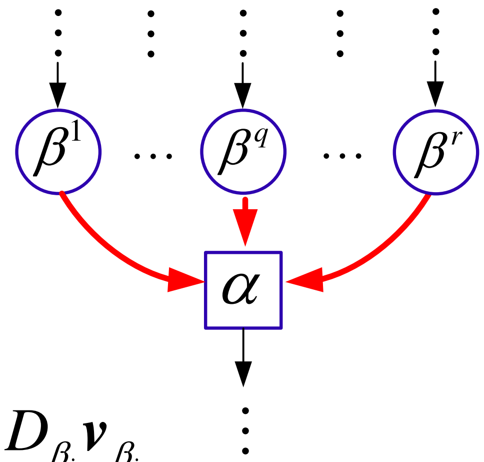
$$Q^i_{t_a \to t_b} = E\left[\boldsymbol{\omega}^i_{t_a \to t_b} \left(\boldsymbol{\omega}^i_{t_a \to t_b}\right)^T\right]$$



- Example:



Graph Representation (cont.)

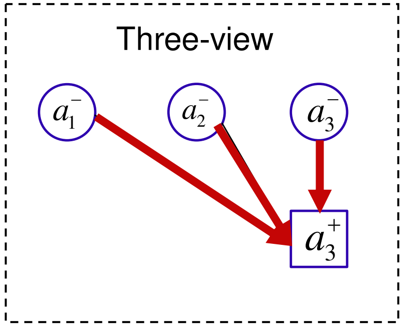


- **MP update relation**
  - β^i : a-priori information of the r participating platforms
  - α : a-posteriori estimation of the updated platform q

$$\tilde{\boldsymbol{X}}_{\alpha} = \left(I - K_{\beta_q} H_{\beta_q}\right)\tilde{\boldsymbol{X}}_{\beta_q} - K_{\beta_q} \sum_{\substack{i=1,\\ i\neq q}}^{r} H_{\beta_i}\tilde{\boldsymbol{X}}_{\beta_i} - K_{\beta_q}\sum_{i=1}^{r} D_{\beta_i}\boldsymbol{v}_{\beta_i}$$

  - Arc weight

$$w\left(\beta^i,\alpha\right) = \begin{cases} I - K_{\beta_q} H_{\beta_q} & i = q \\ -K_{\beta_q} H_{\beta_i} & i \neq q \end{cases}$$

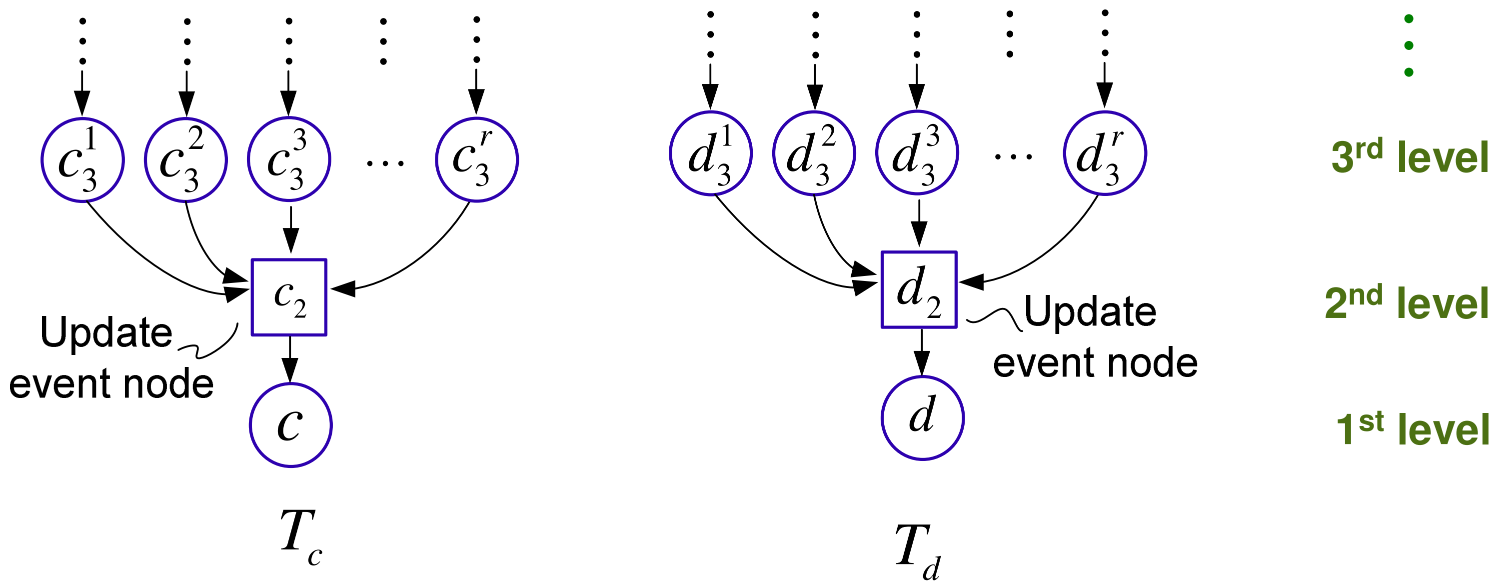


  - The measurement noise covariance is also stored:

$$\forall i \qquad K_{\beta_q} D_{\beta_i} R_{\beta_i}\left(K_{\beta_q} D_{\beta_i}\right)^T \quad , \quad R_{\beta_i} = E\left[\boldsymbol{v}_{\beta_i}\boldsymbol{v}_{\beta_i}^T\right]$$

Algorithm Concept

- Assume we need to calculate $P_{cd} \equiv E\left[\tilde{X}_c \tilde{X}_d^T\right]$
  - **First**: construct two inverse-trees containing all the routes in **G** to the nodes **c** and **d.** Denote the trees as T_c , T_d
  - Notation: a_{i+1} is the parent of the node a_i . If several parents exist: $a_{i+1}^1, a_{i+1}^2, \dots$



  - **Next**: Express P_{cd} using information stored in nodes in T_c , T_d
    - Start with 1st level and proceed upwards

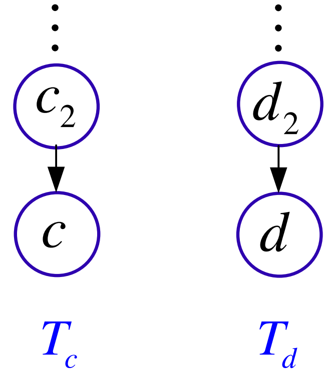
Algorithm Concept (cont.)

- Start with first-level nodes of T_c , T_d : **c** and **d**
- Since $E\left[\tilde{X}_c \tilde{X}_d^T\right]$ is unknown, proceed to next level in the trees
  - According to relation types represented by arc weights
- Assume, e.g., transition relation in both cases
  - $E\left[\tilde{X}_c \tilde{X}_d^T\right]$ may now be expressed as

$$E\left[\tilde{X}_c \left(\Phi_{d_2 \to d} \tilde{X}_{d_2} + \omega_{d_2 \to d}\right)^T\right] =$$

$$E\left[\left(\Phi_{c_2 \to c} \tilde{X}_{c_2} + \omega_{c_2 \to c}\right) \tilde{X}_d^T\right] =$$

$$E\left[\left(\Phi_{c_2 \to c} \tilde{X}_{c_2} + \omega_{c_2 \to c}\right)\left(\Phi_{d_2 \to d} \tilde{X}_{d_2} + \omega_{d_2 \to d}\right)^T\right]$$



  - Are any of the following expressions known (i.e. was stored in the graph)?

$$E\left[\tilde{X}_c \tilde{X}_{d_2}^T\right], E\left[\tilde{X}_{c_2} \tilde{X}_d^T\right], E\left[\tilde{X}_{c_2} \tilde{X}_{d_2}^T\right]$$

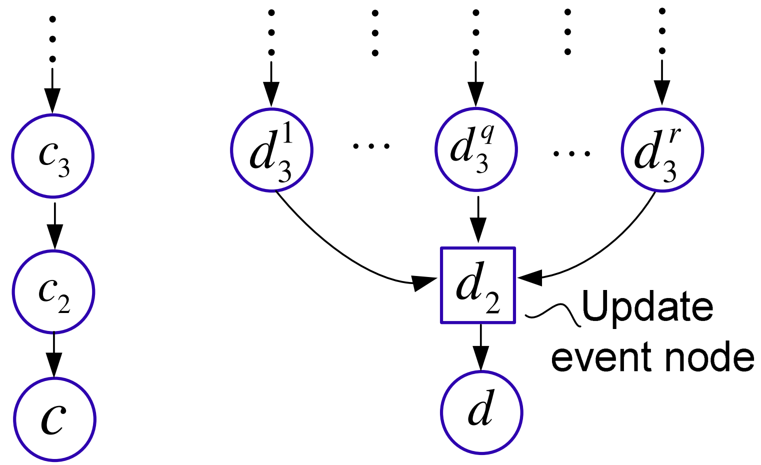
  - Noise terms?

Algorithm Concept (cont.)

- If unknown, proceed to next level in trees – the third level
  - E.g., assume a transition relation in T_c and an MP update relation in T_d
    - Let d_3^q, d_2 be the nodes representing the a-priori and a-posteriori estimations of the updated platform

$$\tilde{X}_{c_2} = \Phi_{c_3 \to c_2} \tilde{X}_{c_3} + \omega_{c_3 \to c_2}$$

$$\tilde{X}_{d_2} = \left(I - K_{d_3^q} H_{d_3^q} \right) \tilde{X}_{d_3^q} - \\ - K_{d_3^q} \sum_{i=1, i \neq q}^{r} H_{d_3^i} \tilde{X}_{d_3^i} - K_{d_3^q} \sum_{i=1}^{r} D_{d_3^i} \boldsymbol{v}_{d_3^i}$$



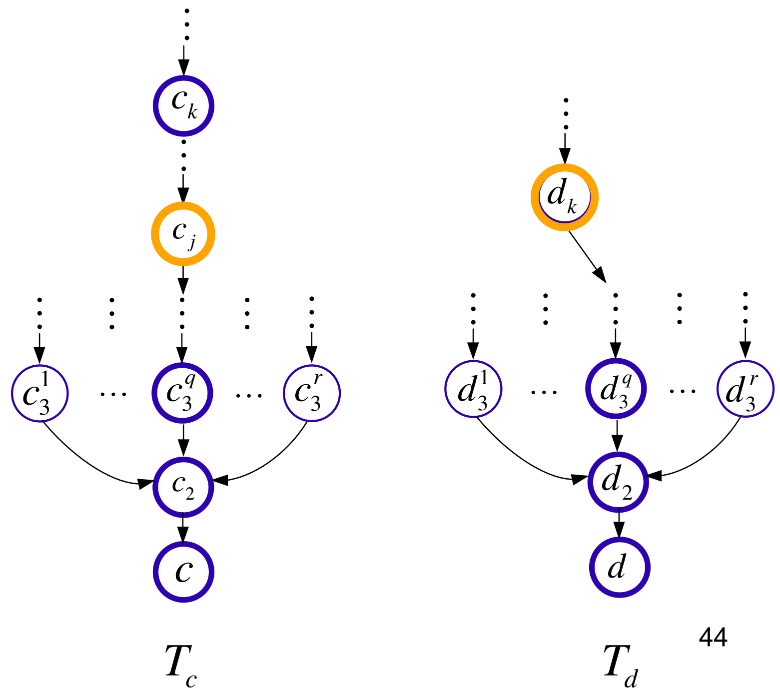
- Now, $E\left[\tilde{X}_c \tilde{X}_d^T \right]$ may be expressed using nodes from the third level and lower levels. For example:

$$E\left[\left(\Phi_{c_2 \to c} \underline{\tilde{X}_{c_2}} + \omega_{c_2 \to c} \right) \left(\Phi_{d_2 \to d} \left(\left(I - K_{d_3^q} H_{d_3^q} \right) \underline{\tilde{X}_{d_3^q}} - K_{d_3^q} \sum_{i=1, i \neq q}^{r} H_{d_3^i} \underline{\tilde{X}_{d_3^i}} - K_{d_3^q} \sum_{i=1}^{r} D_{d_3^i} \boldsymbol{v}_{d_3^i} \right) + \omega_{d_2 \to d} \right)^T \right]$$

Algorithm Concept (cont.)

- The algorithm proceeds to higher levels in T_c and T_d until all the terms required for calculating $E\left[\tilde{\boldsymbol{X}}_c \tilde{\boldsymbol{X}}_d^T\right]$ are known
  - Or, reaching top level in both trees
- Consider reaching the k-th level and analyzing some pair (c_k, d_k) with c_k from T_c and d_k from T_d
  - Look for the pair (c_j, d_k) or (c_k, d_j), so that $E\left[\tilde{\boldsymbol{X}}_{c_j} \tilde{\boldsymbol{X}}_{d_k}^T\right]$ or $E\left[\tilde{\boldsymbol{X}}_{c_k} \tilde{\boldsymbol{X}}_{d_j}^T\right]$ is known (i.e. stored in G), with smallest j
- Assume $E\left[\tilde{\boldsymbol{X}}_{c_j} \tilde{\boldsymbol{X}}_{d_k}^T\right]$ is known
  - The two nodes (c_j in T_c and d_k in T_d) participated in the same MP update in the past
  - Or, the two nodes are the same

$$c_j \equiv d_k$$



Algorithm Concept (cont.)

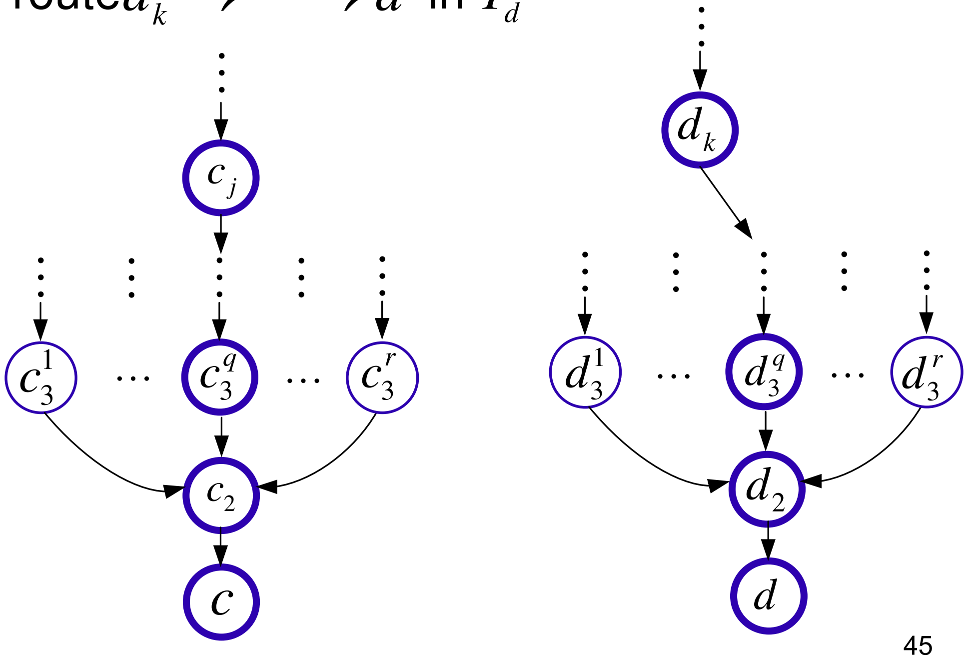
- Consider that some term $E\left[\tilde{X}_{c_j}\tilde{X}_{d_k}^T\right]$ is known
  - No need to proceed to nodes from higher levels, which are related to $E\left[\tilde{X}_{c_j}\tilde{X}_{d_k}^T\right]$
  - The contribution of $\left(c_j, d_k\right)$ to $E\left[\tilde{X}_c\tilde{X}_d^T\right]$ is calculated as

$$W_c\left(c_j\right)E\left[\tilde{X}_{c_j}\tilde{X}_{d_k}^T\right]W_d^T\left(d_k\right)+\bar{Q}_{c_jd_k}$$

    - $W_c\left(c_j\right)$ is the overall weight of the route $c_j \rightarrow \cdots \rightarrow c$ in T_c
    - $W_d\left(d_k\right)$ is the overall weight of the route $d_k \rightarrow \cdots \rightarrow d$ in T_d

$$W_c\left(c_j\right)=\prod_{i=2}^{j} w\left(c_i, c_{i-1}\right)$$

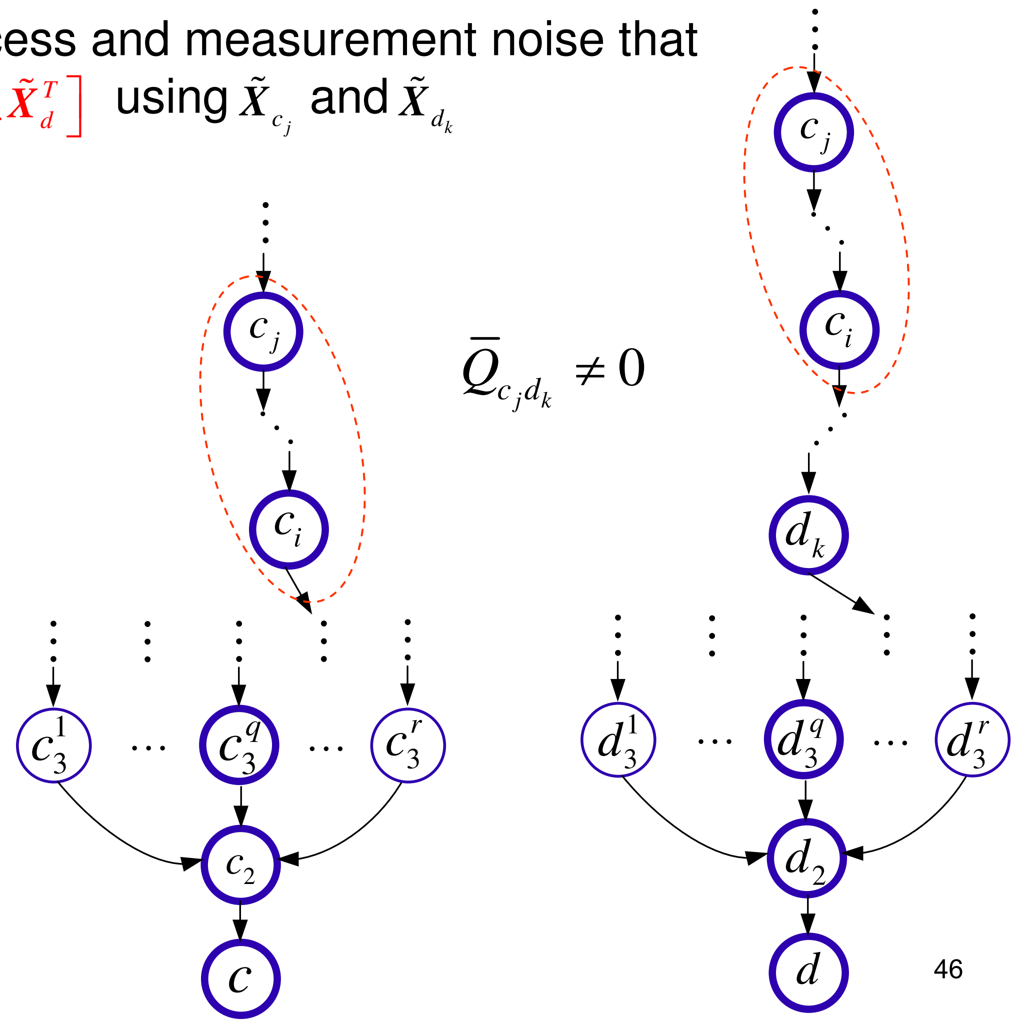
$$W_d\left(d_k\right)=\prod_{i=2}^{k} w\left(d_i, d_{i-1}\right)$$



Algorithm Concept (cont.)

$$W_c\left(c_j\right)E\left[\tilde{\boldsymbol{X}}_{c_j}\tilde{\boldsymbol{X}}_{d_k}^T\right]W_d^T\left(d_k\right)+\bar{Q}_{c_jd_k}$$

- The term $\bar{Q}_{c_jd_k}$
  - Represents the contribution of process and measurement noise that are involved when expressing $E\left[\tilde{\boldsymbol{X}}_c\tilde{\boldsymbol{X}}_d^T\right]$ using $\tilde{X}_{c_j}$ and $\tilde{X}_{d_k}$
  - Lemma: $\bar{Q}_{c_jd_k}=0$ only if
    - c_j in T_c does not have any descendants that are ancestors of d_k in T_d, AND
    - d_k in T_d does not have any descendants that are ancestors of c_j in T_c
  - Otherwise: $\bar{Q}_{c_jd_k}\neq 0$
    - Should be calculated



…Back to the Three-View Measurement Model

3 **Distributed Vision-Aided Cooperative Navigation Based on Three-View Geometry**

…Back to the Three-View Measurement Model

- Measurement model:

$$\boldsymbol{z} \cong H_3 \boldsymbol{X}_{\mathbf{III}}\left(t_3\right) + H_2 \boldsymbol{X}_{\mathbf{II}}\left(t_2\right) + H_1 \boldsymbol{X}_{\mathbf{I}}\left(t_1\right) + D\boldsymbol{v} + H.O.T.$$

- Assume only one platform is updated each time
  - e.g. update equations for platform III:
    - Calculate gain: $K = P_{\boldsymbol{X}_{III}(t_3)\boldsymbol{z}(t_3,t_2,t_1)} P^{-1}_{\boldsymbol{z}(t_3,t_2,t_1)}$

      with

$$P_{\boldsymbol{X}(t_3)\boldsymbol{z}(t_3,t_2,t_1)} = P_3 H_3^T + P_{32} H_2^T + P_{31} H_1^T$$

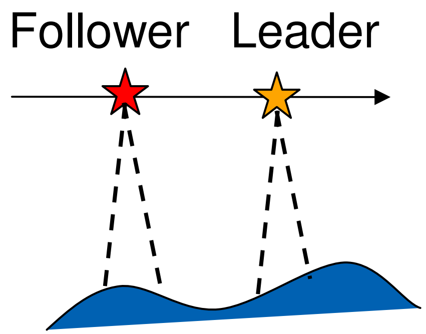
$$P^{-1}_{\boldsymbol{z}(t_3,t_2,t_1)} = H_3 P_3 H_3^T + \begin{bmatrix} H_2 & H_1 \end{bmatrix} \begin{bmatrix} P_2 & P_{21} \\ P_{21}^T & P_1 \end{bmatrix} \begin{bmatrix} H_2 & H_1 \end{bmatrix}^T + DRD^T$$

    - The cross-covariance terms are calculated based on the developed approach
    - Standard state and covariance update IEKF equations

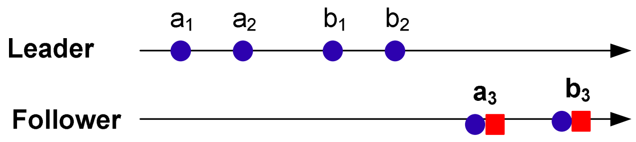
Simulation Results – Leader-Follower Scenario

- 2 platforms: Leader, Follower
  - Leader is equipped with a better IMU
  - Initial navigation errors and IMU errors:

Parameter	Description	Leader	Follower	Units
$\Delta\mathbf{P}$	Initial position error (1σ)	$(10, 10, 10)^T$	$(100, 100, 100)^T$	m
$\Delta\mathbf{V}$	Initial velocity error (1σ)	$(0.1, 0.1, 0.1)^T$	$(0.3, 0.3, 0.3)^T$	m/s
$\Delta\mathbf{\Psi}$	Initial attitude error (1σ)	$(0.1, 0.1, 0.1)^T$	$(0.1, 0.1, 0.1)^T$	deg
$\mathbf{d}$	IMU drift (1σ)	$(1, 1, 1)^T$	$(10, 10, 10)^T$	deg/hr
$\mathbf{b}$	IMU bias (1σ)	$(1, 1, 1)^T$	$(10, 10, 10)^T$	mg

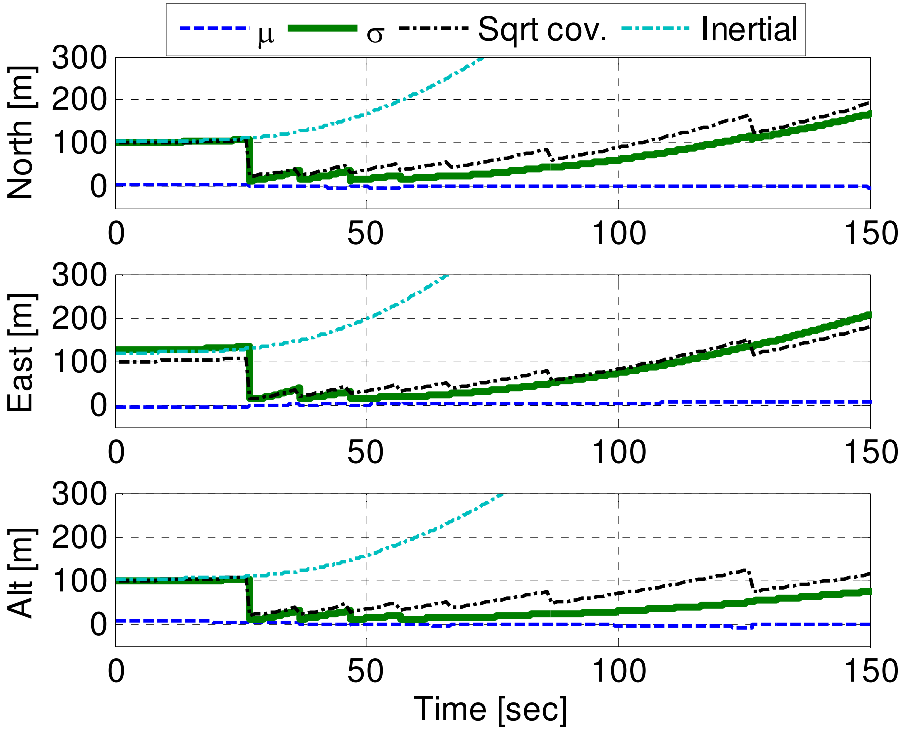


- Trajectory: Straight and level, north heading flight
  - Velocity: 100 m/s
  - Leader is 2000 m ahead (2 second delay)
  - Height above ground level: 2000±200m
- Follower is updated every 10 seconds
- Leader is not updated (inertial navigation)
- Synthetic imagery

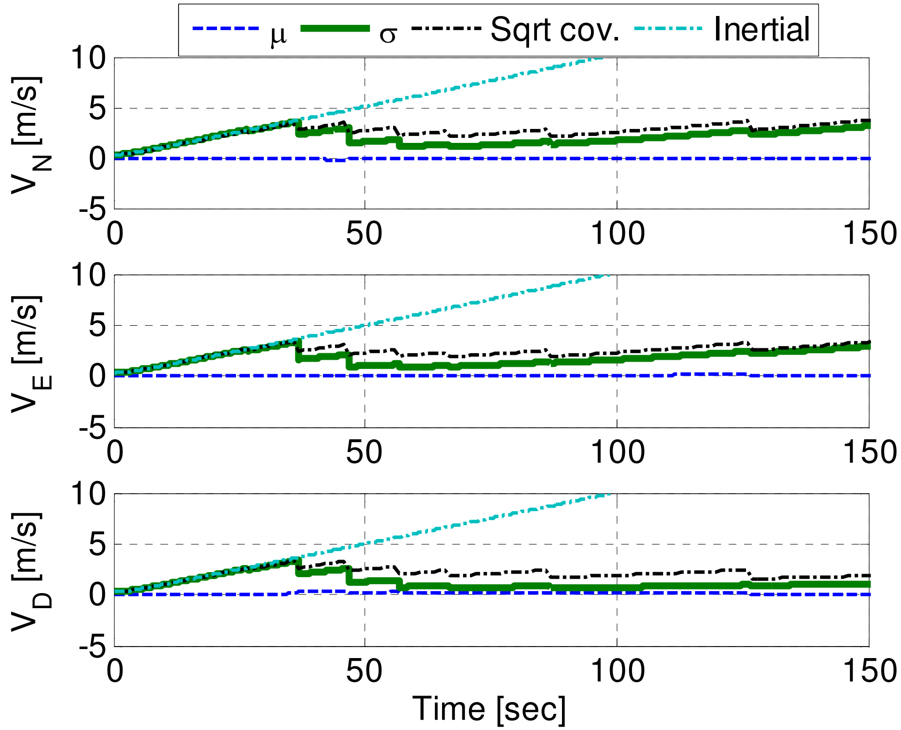


Simulation Results – Leader-Follower Scenario (cont.)

Monte Carlo results (1000 runs): Follower's navigation errors



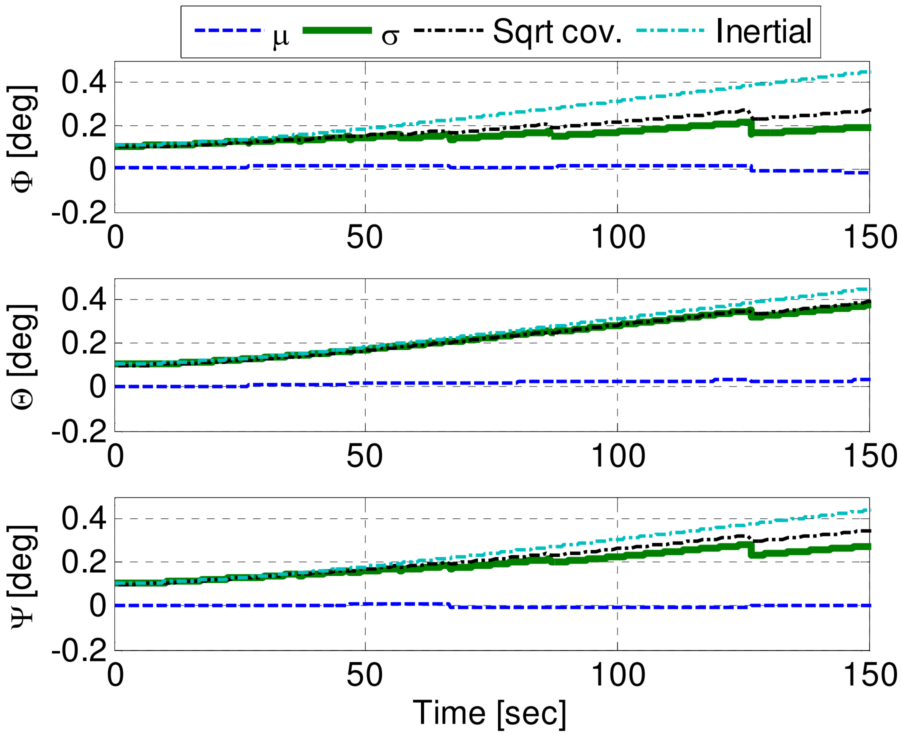
Position errors



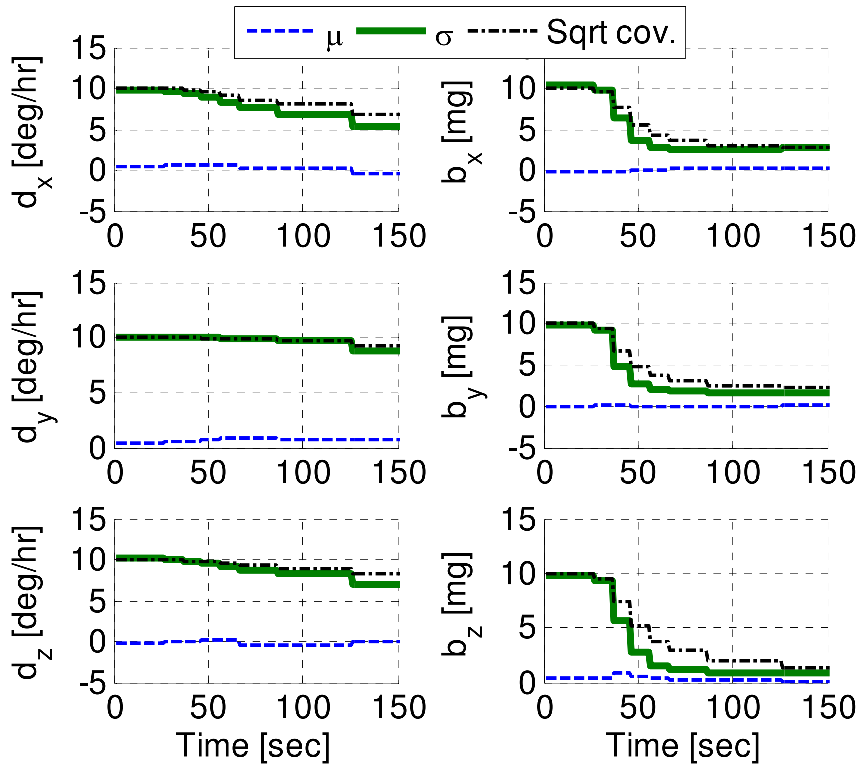
Velocity errors

Simulation Results – Leader-Follower Scenario (cont.)

Monte Carlo results (1000 runs): Follower's navigation errors



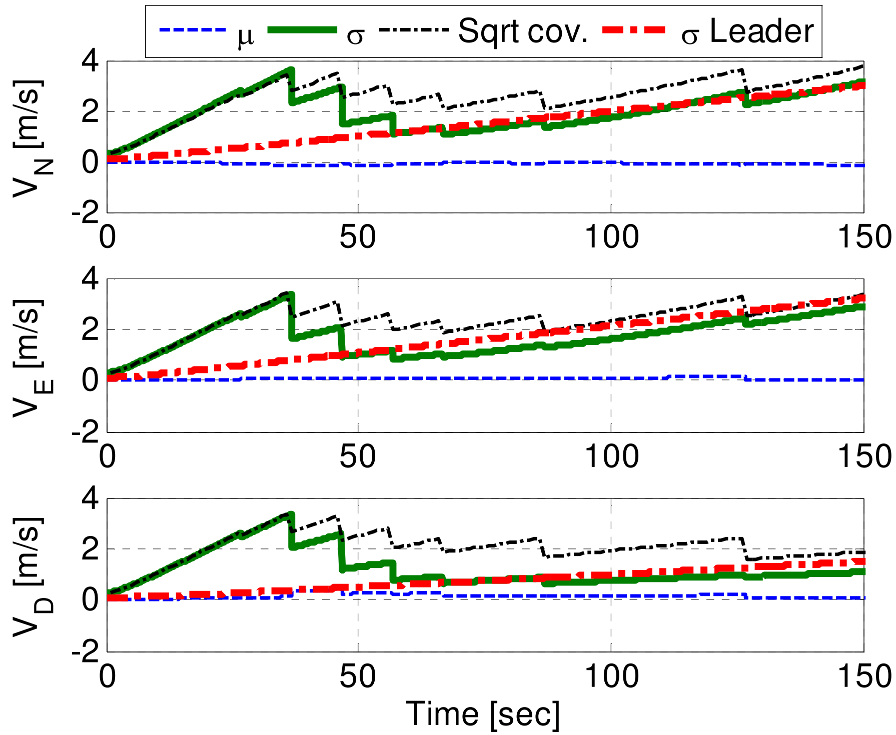
Euler angle errors



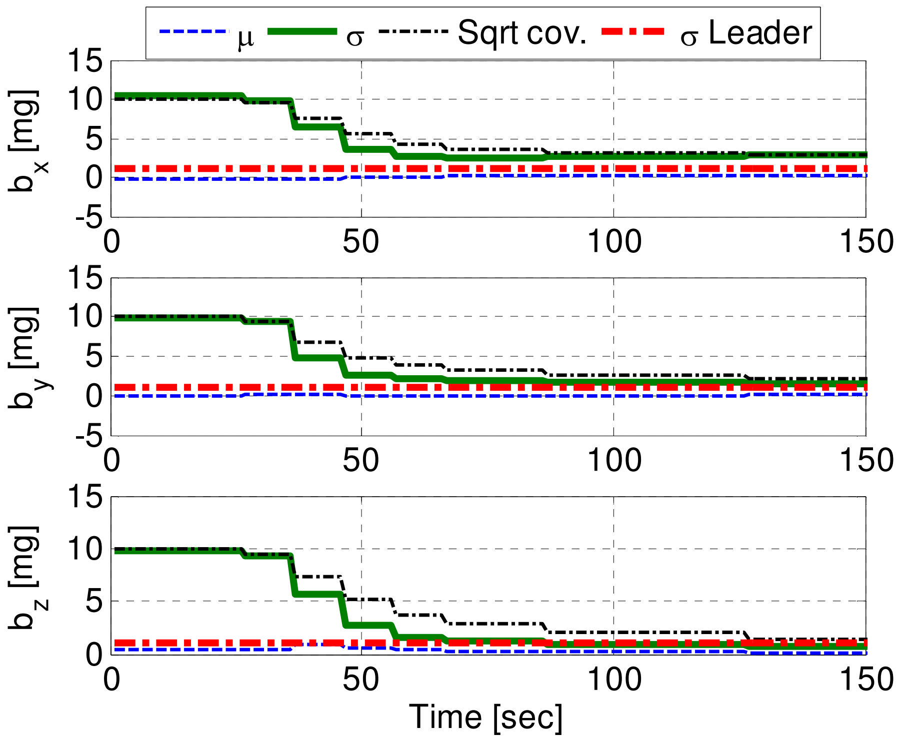
Drift and bias estimation errors

Simulation Results – Leader-Follower Scenario

Monte Carlo results (1000 runs): Follower's vs. Leader's navigation errors



Velocity errors

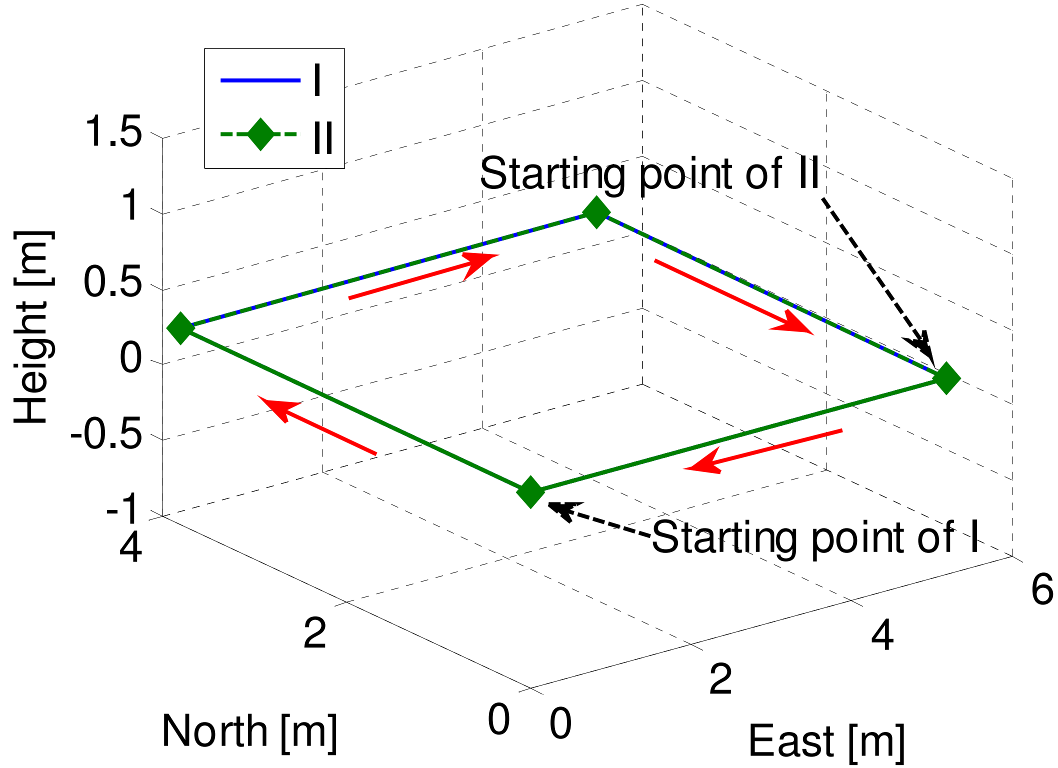


Bias estimation errors

Experiment Results – Pattern Holding Scenario

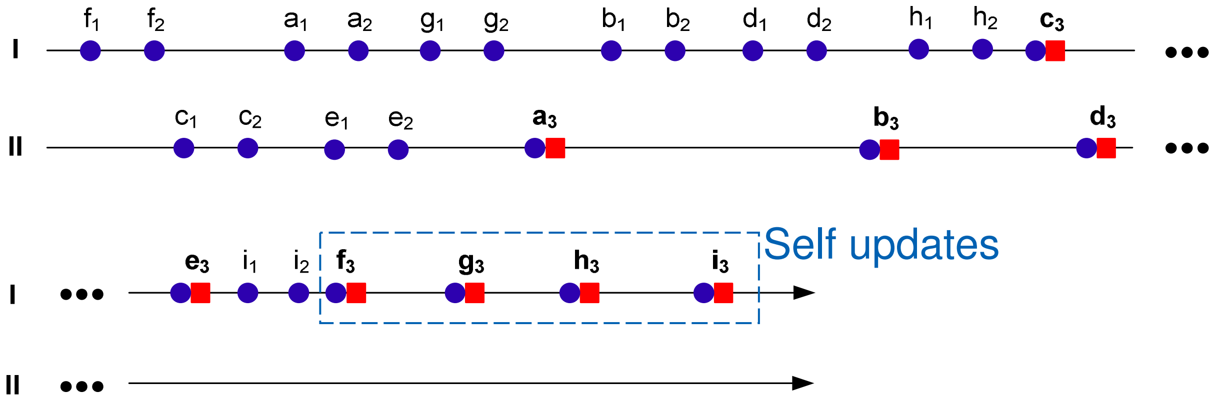
- The same experiment setup
  - Two different trajectories
  - IMU and camera were turned off in between

Two platforms with identical hardware (camera + IMU)



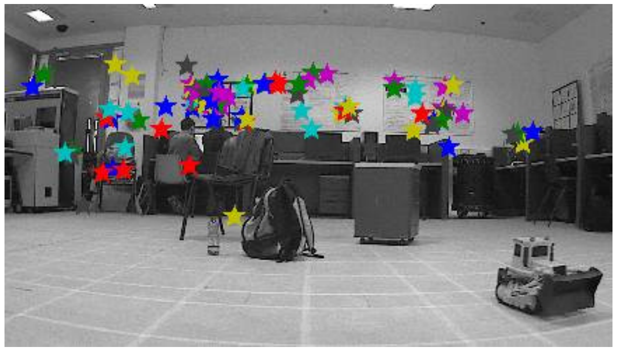
Experiment Results – Pattern Holding Scenario (cont.)

- Two modes:
  - ◇ Multi-platform update
  - × Self update (all images from the same platform)



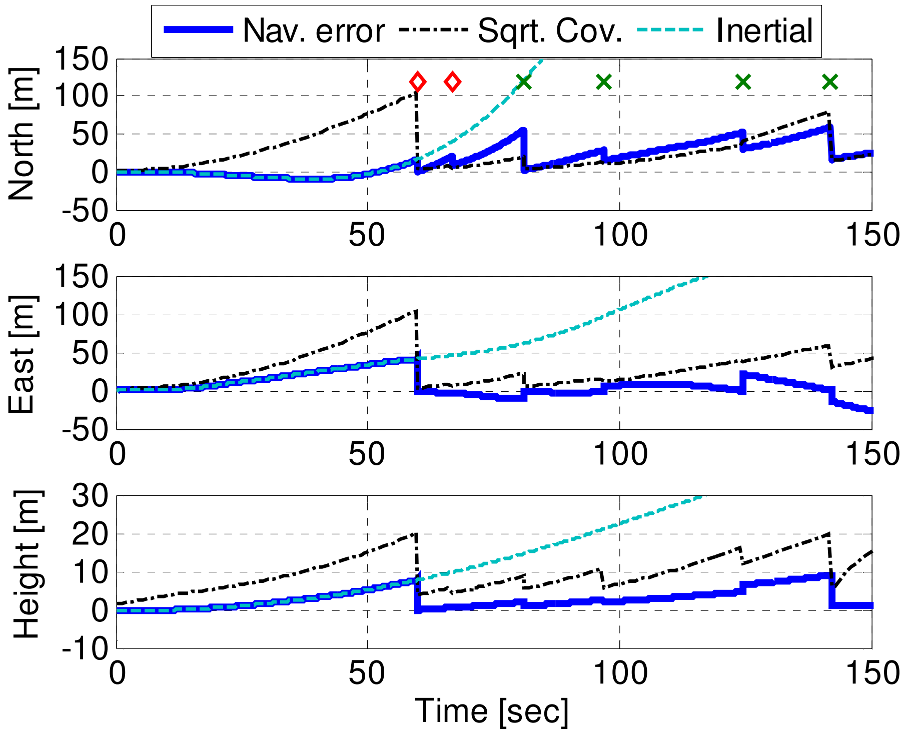
Images used in the first update:

a_1 a_2 a_3

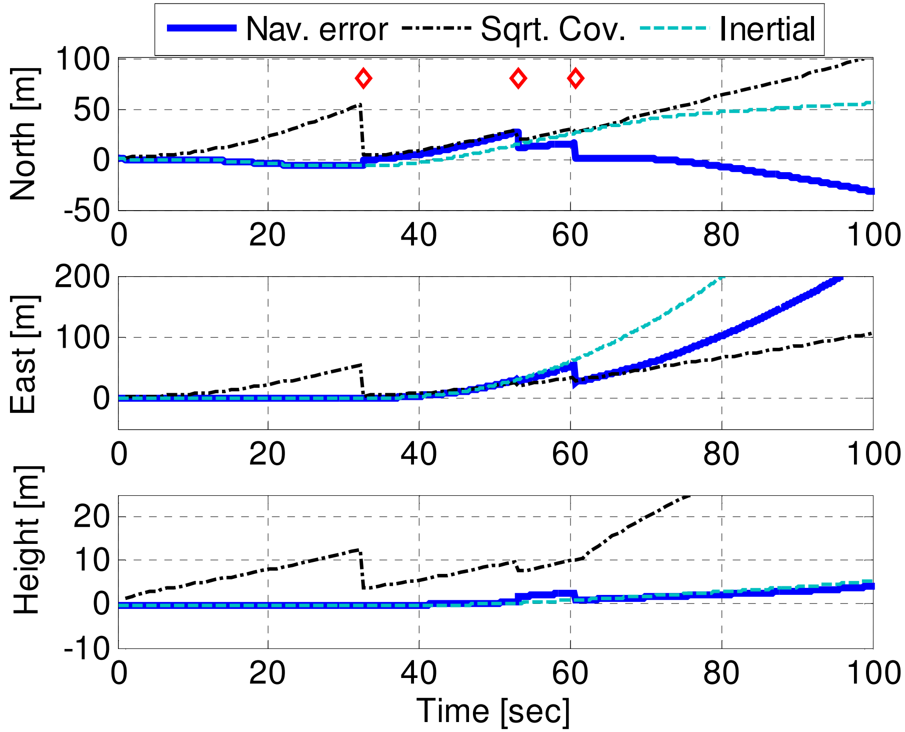


Experiment Results – Pattern Holding Scenario (cont.)

Position errors



Platform I



Platform II

Conclusions

- Distributed cooperative navigation aiding
  - Allows reduction of navigation errors in some platforms based on other platforms in the group
  - Three-view constraints are formulated whenever the same scene is observed by several platforms
    - The camera is no more required to be aimed towards other platforms (as in relative pose measurements)
    - Range sensor is not required
    - The views are not necessarily captured at the same time
- Graph-based approach for on-demand calculation of cross-covariance terms
  - General multi-platform measurement model
  - EKF framework

Summary

- Vision-aided navigation
  - INS, camera
  - Utilize incoming imagery, and constructed mosaics, for navigation aiding
- Algorithms
  - Coupled mosaicking and camera scanning
    - Improved navigation performance in challenging scenarios
    - Reduced computational requirements
  - Three-view geometry constraints
    - Reduction of position and velocity errors in all axes
    - Efficient handling of loop scenarios
  - Distributed cooperative navigation based on three-view constraints
    - Reduction of navigation errors of some platforms based on navigation and imagery information obtained from other platforms
    - Graph based approach for on-demand calculation of cross-covariance terms for a general multi-platform measurement model

Thank you …