Graph-Based Distributed Cooperative Navigation

V. Indelman, P. Gurfil

Distributed Space Systems Lab,

Aerospace Engineering, Technion

E. Rivlin

Computer Science, Technion

H. Rotstein

Electrical Engineering, Technion

May 2011

Contents

Introduction

- A group of cooperative platforms is considered
  - Required to autonomously perform different missions
  - Navigation is an essential capability
- Dead reckoning \ inertial navigation errors have to be compensated
- Several methods for cooperative navigation were proposed
  - Relative pose measurements between pairs of robots: e.g., “Distributed Multirobot Localization”, Roumeliotis S.I. and Bekey G.A., 2002
  - Two-view geometry and relative pose measurements between pairs of robots: “Multiple Relative Pose Graphs for Robust Cooperative Mapping”, Kim B. et al., 2010
  - Three-view geometry measurements between triplets\pairs of robots (camera only): “Distributed Vision-Aided Cooperative Navigation Based on Three-View Geometry”, Indelman V. et al., 2011

Introduction (Cont.)

- Common property – each measurement is constituted upon navigation information **obtained from different robots**
  - In the general case, these sources of information are **correlated**
  - Neglecting the correlations results in inconsistent information fusion
  - Therefore – appropriate correlations terms should be known

- Previous Work:
  - Augmented covariance matrix: e.g., “Distributed Multirobot Localization”, Roumeliotis S.I. and Bekey G.A., 2002
  - Consistent information fusion: “Consistent Cooperative Localization”, Bahr A. et al., 2009
  - Smoothing: “Multiple Relative Pose Graphs for Robust Cooperative Mapping”, Kim B. et al., 2010

- In this work: **Calculate explicitly correlation terms upon demand**

General Multi-Robot Measurement Model

- Assume a group of N robots
- A general number of robots, r, contribute navigation information and readings of its onboard sensors $\left(r \le N\right)$
  - Not necessarily from the same time (e.g., two-view measurements)

$$z\left(t\right) = \boldsymbol{h}\left(\left\{\boldsymbol{x}_i\left(t_i\right), \boldsymbol{y}_i\left(t_i\right)\right\}_{i=1}^{r}\right) \cong \sum_{i=1}^{r} H_i \boldsymbol{X}_i\left(t_i\right) + D_i\left(t_i\right)\boldsymbol{v}_i\left(t_i\right)$$

$\boldsymbol{x}_i\left(t_i\right)$: Navigation solution of robot *i* at time t_i $\left(t_i \le t\right)$

$\boldsymbol{y}_i\left(t_i\right)$: Readings of onboard sensors of robot *i* at time t_i

$z\left(t\right)$: Residual Multi-Robot measurement

$\boldsymbol{v}_i\left(t_i\right)$: Measurement errors of onboard sensors of robot *i* at time t_i

$\boldsymbol{X}_i\left(t_i\right)$: Navigation errors of robot *i* at time t_i

General Multi-Robot Measurement Model (Cont.)

- Navigation error development for the *i*-th robot: $X_i(t_b) = \Phi^i_{t_a \to t_b} X_i(t_a) + \omega^i_{t_a \to t_b}$

- Update step of the Kalman filter involves cross-covariance terms

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

  - $\tilde{X}_i(t_i)$: Estimation error of $X_i(t_i)$

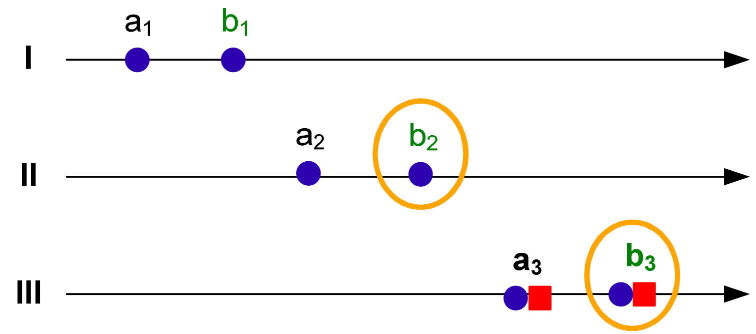
- Objective: Calculate $E\left[\tilde{X}_i(t_i)\tilde{X}_j^T(t_j)\right]$
  - Identities of the robots participating in the measurement are a priori unknown
  - The time instances t_i are also a priori unknown

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

Basic Example

- Three-robot measurement model:

$$z(t) \cong \sum_{i=1}^{3} H_i X_i(t_i) + D\boldsymbol{v}$$



- a priori information
- update event (only one robot is updated)

- Assume first update (**a$_3$**) was carried out
- Objective: Calculate $\boldsymbol{E}\left[\tilde{\boldsymbol{X}}_{III}(t_{b_3})\tilde{\boldsymbol{X}}_{II}^{T}(t_{b_2})\right] \equiv P_{b_3 b_2}$

$$\tilde{\boldsymbol{X}}_{III}^{-}(t_{b_3}) = \Phi_{a_3\to b_3}^{III}\tilde{\boldsymbol{X}}_{III}^{+}(t_{a_3}) + \omega_{a_3\to b_3}^{III} \quad , \quad \tilde{\boldsymbol{X}}_{II}^{-}(t_{b_2}) = \cdots$$

$$\tilde{\boldsymbol{X}}_{III}^{+}(t_{a_3}) = \left(I - K_{a_3}H_{a_3}\right)\tilde{\boldsymbol{X}}_{III}^{-}(t_{a_3}) - K_{a_3}H_{a_2}\tilde{\boldsymbol{X}}_{II}^{-}(t_{a_2}) - K_{a_3}H_{a_1}\tilde{\boldsymbol{X}}_{I}^{-}(t_{a_1}) - K_{a_3}D_a\boldsymbol{v}_a$$

$$\Rightarrow \quad P_{b_3 b_2} = \Phi_{a_3\to b_3}^{III}\left[\left(I - K_{a_3}H_{a_3}\right)P_{a_3 a_2}^{-} - K_{a_3}H_{a_2}P_{a_2 a_2}^{-} - K_{a_3}H_{a_1}P_{a_1 a_2}^{-}\right]\left(\Phi_{a_2\to b_2}^{II}\right)^{T}$$

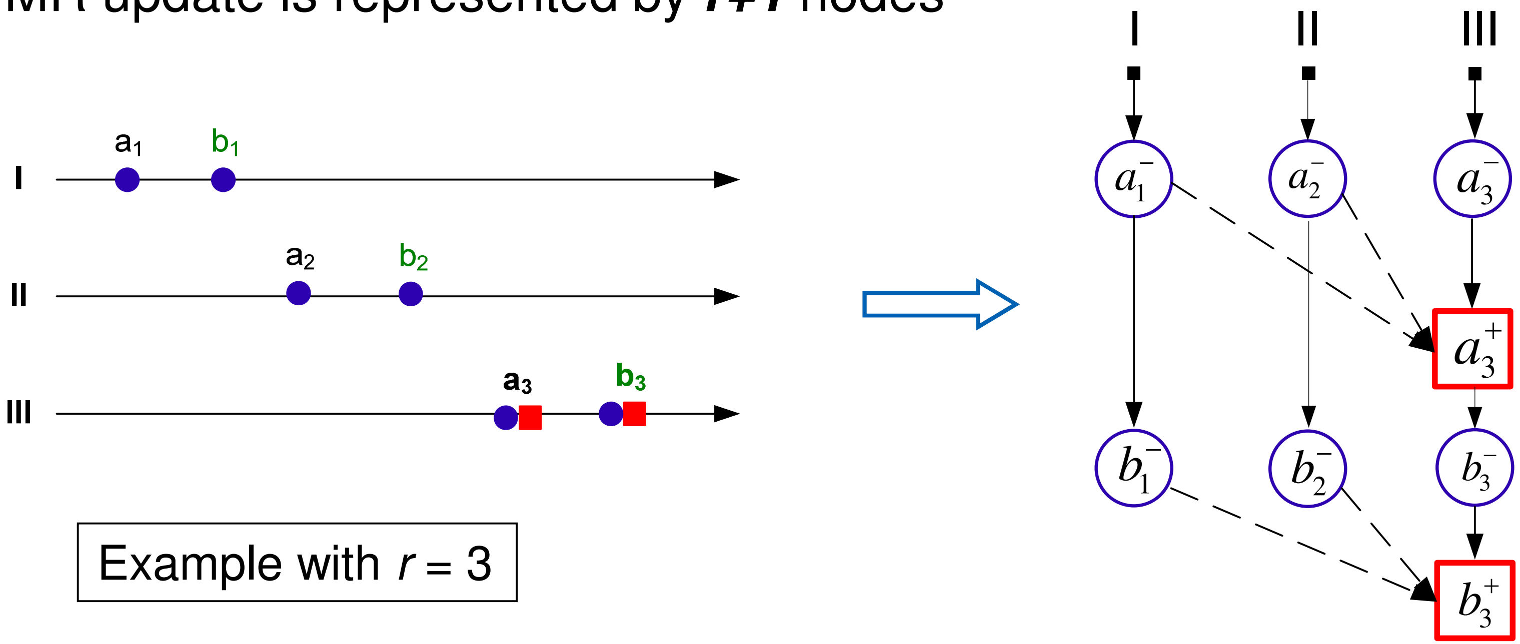
Concept

- A general MR measurement: $\boldsymbol{z} \cong \sum_{i=1}^{r} H_i \boldsymbol{X}_i(t_i) + D_i(t_i)\boldsymbol{v}_i(t_i)$
- Objective: Calculate $\boldsymbol{E}\left[\tilde{\boldsymbol{X}}_i(t_i)\tilde{\boldsymbol{X}}_j^T(t_j)\right]$

1. Represent all MR updates executed so far in a directed acyclic graph (DAG) G.
2. Express $\tilde{\boldsymbol{X}}_i(t_i)$ and $\tilde{\boldsymbol{X}}_j(t_j)$ according to the history of MR 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.

Concept (cont.)

- Each platform maintains its own DAG G
- A-priori and a-posteriori covariance and cross-covariance matrices are stored in G after <u>each</u> MR update
- Two node types in G: a-priori and a-posteriori nodes
  - Nodes representing (a-priori) information participating in an MR measurement
  - Update-event node, representing a-posteriori estimate of the updated robot
- Each MR update is represented by ***r+1*** nodes



Example with $r = 3$

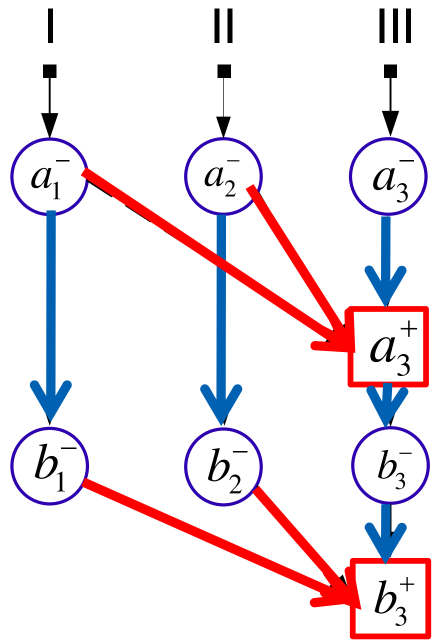
Graph Representation (cont.)

- Each node can be connected to another node by a
  - Transition relation $\tilde{X}_b = \Phi^i_{t_a \to t_b} \tilde{X}_a + \omega^i_{t_a \to t_b}$
  - MR update relation

a priori nodes

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

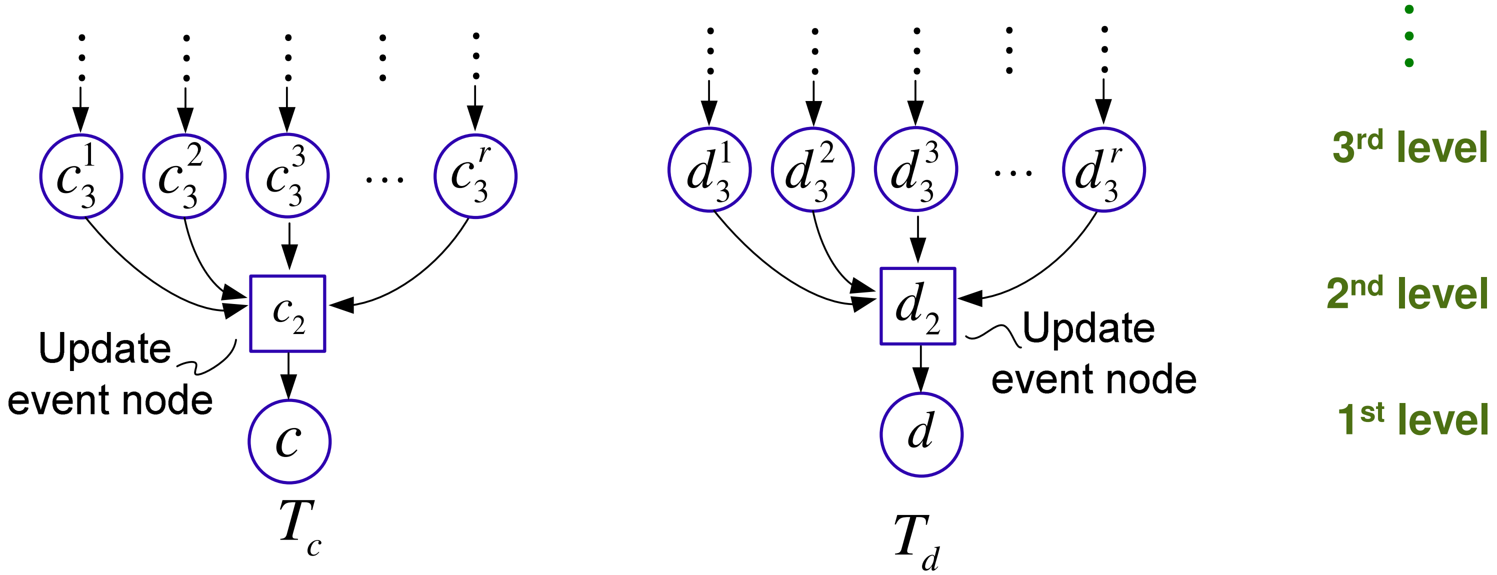
a posteriori node



- The process and measurement noise covariance matrices are also stored

- Assume we need to calculate $P_{cd} \equiv E\left[\tilde{X}_c \tilde{X}_d^T\right]$
  - **First**: Construct two inverse-trees T_c , T_d
    - Containing all the routes in **G** to the nodes **c** and **d**
  - **Next**: Express P_{cd} using information stored in nodes in T_c , T_d

Algorithm Concept



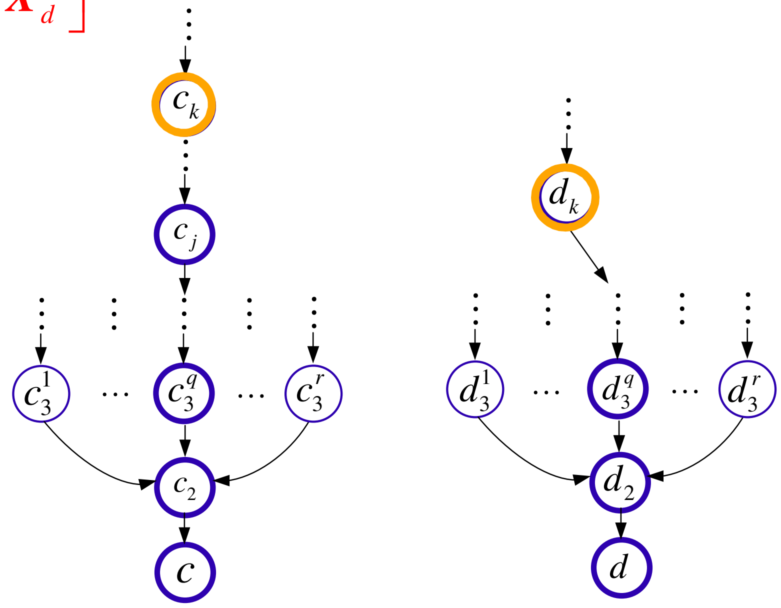
- Start with first-level nodes of T_c , T_d : **c** and **d**
- Since $E\left[\tilde{X}_c \tilde{X}_d^T\right]$ is unknown, proceed to next level in the trees
  - According to relation types represented by arc weights
- <u>For example</u>, assume transition relation in both cases

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

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

Algorithm Concept (cont.)

- If unknown, proceed to higher levels in T_c and T_d
  - Until all the terms required for calculating $E\left[\tilde{X}_c \tilde{X}_d^T\right]$ are known
  - Or, reaching top level in both trees
- Consider reaching the <u>k-th level</u> and analyzing some pair (c_k, d_k) $\quad c_k \in T_c$, $d_k \in T_d$
- Look for the pair (c_j, d_k) so that $E\left[\tilde{X}_{c_j} \tilde{X}_{d_k}^T\right]$ is known (i.e. stored in G), with smallest j
  - Or (c_k, d_j) with a known $E\left[\tilde{X}_{c_k} \tilde{X}_{d_j}^T\right]$



- The contribution of a known term $E\left[\tilde{X}_{c_j} \tilde{X}_{d_k}^T\right]$ to $E\left[\tilde{X}_c \tilde{X}_d^T\right]$ is calculated as

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

Overall weight of the route $c_j \to \cdots \to c$ in T_c

Overall weight of the route $d_k \to \cdots \to d$ in T_d

Contribution of process and measurement noise terms. See paper…

Results

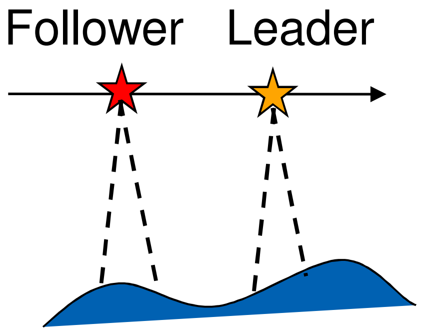
Three-view measurement model: $\boldsymbol{z}(t) \cong \sum_{i=1}^{3} H_i \boldsymbol{X}_i(t_i) + D\boldsymbol{v}$

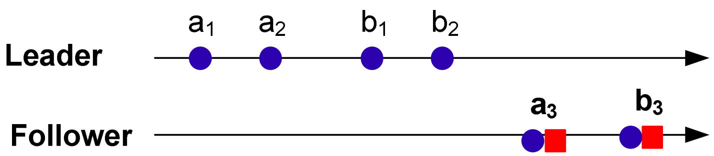
Details: "Distributed Vision-Aided Cooperative Navigation Based on Three-View Geometry", Indelman V. et al., 2011

Simulation Results – Leader-Follower Scenario

- 2 robots: Leader, Follower
  - Leader is equipped with a better IMU

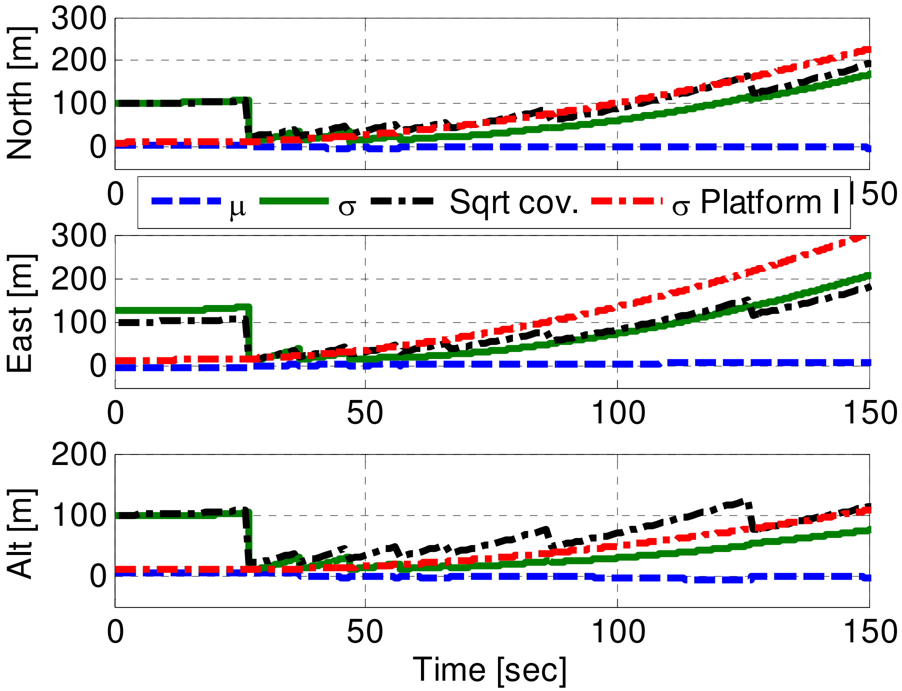
- Scenario:
  - Trajectory: Straight and level, north heading flight
  - Leader is 20 sec ahead
  - Follower is updated every 10 seconds
  - Leader <u>is not</u> updated (<u>inertial</u> navigation)
  - Synthetic imagery



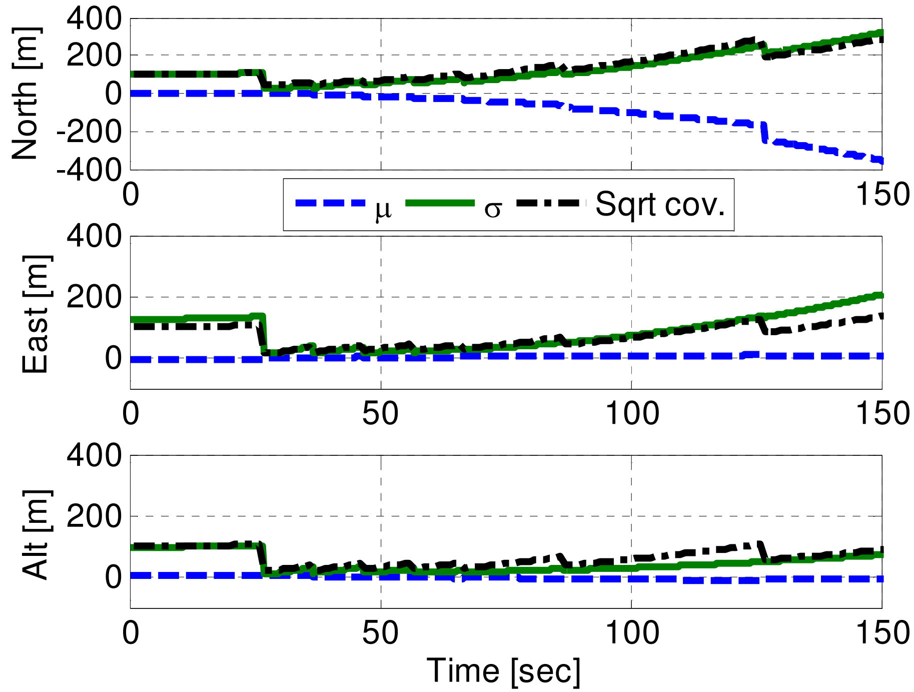


Simulation Results – Leader-Follower Scenario (cont.)

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



Position errors – compared to **Leader's** errors



Position errors – **Correlation is neglected**

Conclusions

- A method for on-demand explicit cross-covariance calculation was presented
  - Multi-Robot general measurement model. A measurement is composed from information obtained from
    - Any number of robots
    - Not necessarily at the same time
  - Graph-based approach was applied
    - Allows properly handling noise covariance terms
  - The method was demonstrated for a three-view measurement model