

Distributed Vision-Aided Cooperative Localization and Navigation Based on Three-View Geometry

1

Vadim Indelman, Pini Gurfil, Ehud Rivlin and Hector Rotstein

Abstract

This paper presents a new method for distributed vision-aided cooperative localization and navigation for multiple inter-communicating autonomous vehicles based on three-view geometry constraints. Each vehicle is equipped with a standard inertial navigation system and an on-board camera only. In contrast to the traditional approach for cooperative localization, which is based on relative pose measurements, the proposed method formulates a measurement whenever the same scene is observed by different vehicles. Each such measurement is comprising of three images, which are not necessarily captured at the same time. The captured images, to which some navigation parameters are attached, are stored in repositories by some of the vehicles in the group. A graph-based approach is applied for calculating the correlation terms between the navigation parameters associated to images participating in the same measurement. The proposed method is examined using a statistical simulation and is further validated in an experiment that involved two vehicles in a holding pattern scenario. The experiments show that the cooperative three-view-based vision-aided navigation may considerably improve the performance of an inferior INS.

I. INTRODUCTION

Cooperative localization and navigation has been an active research field for over a decade. Research efforts have been devoted to developing methods enabling a group of vehicles to autonomously perform various missions. These missions include cooperative mapping and localization [1], [2], [3], formation flying [4], autonomous multi-vehicle transport [5], and other applications.

Precise navigation is a key requirement for carrying out any autonomous mission by a group of cooperative vehicles. If the global positioning system (GPS) is available, then each of the vehicles is capable of computing the navigation solution on its own, usually by fusing the GPS data with an inertial navigation system (INS). However, whenever the GPS signal is absent, such as when operating indoors, underwater, in space, in urban environments, or on other planets, alternative methods should be devised for updating the navigation system due to the evolving dead reckoning errors.

While various methods exist for navigation-aiding of a single vehicle, collaboration among several, possibly heterogeneous, vehicles, each equipped with its own set of sensors, is expected to improve performance even further [6]. Indeed, different methods have been developed for effectively localizing a group of vehicles with respect to some static coordinate system or with respect to themselves. Typically, these methods assume that each vehicle is capable of measuring the relative range and bearing to other vehicles.

V. Indelman is with the School of Interactive Computing, Georgia Institute of Technology, Atlanta, GA 30332, USA (e-mail: indelman@cc.gatech.edu).

P. Gurfil is with the Faculty of Aerospace Engineering, Technion - Israel Institute of Technology, Haifa 32000, Israel (e-mail: pgurfil@technion.ac.il).

E. Rivlin is with the Department of Computer Science, Technion - Israel Institute of Technology, Haifa 32000, Israel (e-mail: ehudr@cs.technion.ac.il).

H. Rotstein is with RAFAEL - Advanced Defense Systems Limited, Israel (e-mail: hector@rafael.co.il).

One of the pioneering works on cooperative localization proposed to restrain the development of navigation errors by using some of the robots as static landmarks for updating the other robots in the group, and then switching between the roles after a certain amount of time [7]. While the method was further improved by others, all the derived methods share the same drawback: some of the robots must stop for updating the others.

Another important work is by Roumeliotis and Bekey [6], wherein a centralized approach for sensor fusion was applied based on the available relative pose measurements between the vehicles in the group. This architecture was then de-centralized and distributed among the vehicles. In [8], an extension was proposed to handle more general relative observation models. A similar setup has also been studied in other works, including [5], [9], [10], [11] and [12].

A different body of works [13], [14], [15] suggests to maintain in each vehicle an estimation of parameters for all the vehicles in the group. For example, in [13] and [14] each vehicle estimates the pose of *every* other vehicle relative to itself, while in [15] each vehicle estimates the navigation state (position, velocity and attitude) of all the other vehicles by exchanging information from inertial measurement units (IMU) and relative pose measurements.

The current work proposes to utilize vision-based navigation-aiding techniques for cooperative navigation. Each vehicle has a standard INS and a camera. No further sensors or a priori information is used. Several works [16], [17] on cooperative navigation have been published relying on this setup. In [16], a leader-follower formation of mobile ground vehicles is considered, in which each vehicle is equipped with a camera that provides bearing to the other vehicles. It is shown that it is possible to estimate the relative pose between the followers and the leader, except for configurations that are not observable. The authors of [17] propose to estimate the relative location of two moving ground robots by fusing camera bearing measurements with the robots' odometry.

The method proposed herein relies on a recently-developed technique [18], [19] for vision-aided navigation of a single vehicle based on a three-view geometry of a general scene. Cooperative navigation is a natural extension of this technique. As opposed to all the methods for cooperative localization and navigation mentioned above, the vehicle's camera is *not* required to be aimed at other vehicles for obtaining measurements of relative parameters (bearing, range). Instead, a measurement is formulated whenever the same scene is observed by different vehicles.

A similar concept has been proposed in [3] and [20], considering measurements that combine pairs of vehicles. In [20], a homography matrix is estimated whenever two airborne vehicles capture images of the same scene, which is assumed to be planar. Next, the relative motion between the two vehicles is extracted from the homography matrix. Assuming the position of the first vehicle and its height above ground level are known, the estimated relative motion is then used for updating the position of the second vehicle. Kim et al. [3] perform nonlinear optimization, involving the pose history of all the vehicles in the group, each time a new measurement arrives, considering relative pose and two-view measurements between pairs of vehicles.

In this work, three views of a common general scene are required for each measurement. These views are acquired by at least two different vehicles, i. e. either each view is captured by a different vehicle, or two of the three views are captured by the same vehicle. The scenario in which all the views were captured by the same vehicle is handled in [18], [19]. The constraints stemming from a general scene, observed from three different views reduce the navigation errors of the updated vehicle without any additional a priori information or sensors.

Another key aspect of the proposed method is that the three images of the same region are *not* necessarily captured at the same time. All, or some, of the vehicles maintain a local

repository of captured images that are associated with some navigation parameters [18], [19].³ These repositories are accessed on demand to check if a region, currently observed by one of the vehicles – the *querying vehicle* – has been observed in the past by other vehicles in the group. Images containing the same region are transmitted, with the attached navigation data, to the querying vehicle. The received information from other vehicles, in addition to the navigation and imagery data of the querying vehicle, allow updating the querying vehicle navigation system.

The navigation information participating in the update process can be statistically dependent. Therefore, in order to obtain a consistent navigation solution, the involved correlation terms should be either maintained or calculated upon demand. However, maintaining the correlation terms is impractical since the time instances of the images participating in the measurements are a priori unknown, in addition to the a priori unknown identity of the vehicles that captured these images. Thus, one has to neglect the involved correlation terms or calculate them upon request. This paper follows the latter approach, adopting a graph-based technique for calculating correlation for general multi-vehicle measurement models.

Consequently, the main contribution of this paper is a new method for vision-aided cooperative navigation that reduces navigation errors, including position and velocity errors in all axes, using only a single camera and an INS, without requiring any range measurements.

II. METHOD OVERVIEW

Figure 1 shows the concept of the proposed method for multi-vehicle vision-aided navigation. The proposed method assumes a group of cooperative vehicles capable of inter-communication. Each vehicle is equipped with a standard INS and an onboard camera, which may be gimballed. Some, or all, of the vehicles maintain a local repository comprised of images captured along the mission. These images are attached with navigation data when they are captured. The INS uses IMU measurements for calculating a navigation solution, which is accompanied by errors that develop over time.

In a typical scenario, a vehicle captures an image and broadcasts it, along with its current navigation solution, to other vehicles in the group, inquiring if they have previously captured images containing the same region. Upon receiving such a query, each vehicle performs a check in its repository looking for appropriate images. Among these images, only images with a smaller navigation uncertainty compared to the uncertainty in the navigation data of the query image, are transmitted back. Vehicles that do not maintain a repository perform the check only on the currently-captured image.

The process of the querying vehicle is schematically described in Fig. 1. After receiving the images and the attached navigation data from other vehicles in the group, two best images are chosen and, together with the querying image, are used for formulating the three-view constraints (Section III). These constraints are then transformed into a measurement and are used for updating the navigation system of the querying vehicle, as described in Section IV. Since the navigation data attached to the chosen three images may be correlated, a graph-based technique is applied for calculating the required cross-covariance terms for the fusion process. This technique is discussed in Section V. The overall protocol for information sharing among the vehicles in the group is presented in Section VI.

Throughout this paper, the following coordinate systems are used:

- L - Local-level, local-north (LLLN) reference frame, also known as a north-east-down (NED) coordinate system. Its origin is set at the location of the navigation system. X_L points north, Y_L points east and Z_L completes a Cartesian right hand system.
- B - Body-fixed reference frame. Its origin is set at the vehicle's center-of-mass. X_B points towards the vehicle's front, Y_B points right when viewed from above, and Z_B completes the setup to yield a Cartesian right hand system.

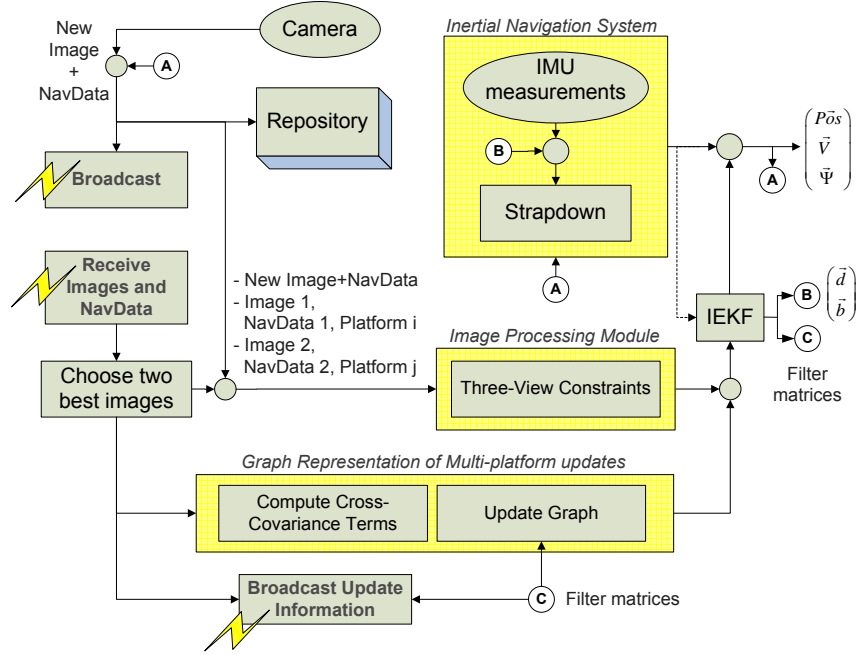


Fig. 1. Multi-vehicle navigation aiding - querying vehicle scheme. “A” denotes the corrected navigation solution, “B” represents the estimated parameterization of IMU errors, and “C” denotes the filter matrices calculated in the update step.

- C - Camera-fixed reference frame. Its origin is set at the camera center-of-projection. Z_C points toward the FOV center, X_C points toward the right half of the FOV when viewed from the camera center-of-projection, and Y_C completes the setup to yield a Cartesian right hand system.

III. THREE-VIEW GEOMETRY CONSTRAINTS

Assume that some general scene is observed from three different views, captured by different vehicles. Fig. 2 depicts such a scenario, in which a static landmark \mathbf{p} is observed in the three images I_1 , I_2 and I_3 . Image I_3 is the currently-captured image of the third vehicle, while I_1 and I_2 are two images captured by the first two vehicles. These two images can be the currently-captured images of these vehicles, but they could also be captured in the past and stored in the repository of each vehicle, as illustrated in the figure. Alternatively, I_1 and I_2 could also be captured by the same vehicle.

Denote by \mathbf{T}_{ij} the translation vector from the i th to the j th view, with $i, j \in \{1, 2, 3\}$ and $i \neq j$. Let also \mathbf{q}_i and λ_i be a line of sight (LOS) vector and a scale parameter, respectively, to the landmark \mathbf{p} at time t_i , such that $\|\lambda_i \mathbf{q}_i\|$ is the range to this landmark. In particular, if \mathbf{q}_i is a unit LOS vector, then λ_i is the range to \mathbf{p} . The constraints resulting from observing the same landmark from three different views can be formulated as follows [18], [19]:

$$\mathbf{q}_1^T (\mathbf{T}_{12} \times \mathbf{q}_2) = 0 \quad (1a)$$

$$\mathbf{q}_2^T (\mathbf{T}_{23} \times \mathbf{q}_3) = 0 \quad (1b)$$

$$(\mathbf{q}_2 \times \mathbf{q}_1)^T (\mathbf{q}_3 \times \mathbf{T}_{23}) = (\mathbf{q}_1 \times \mathbf{T}_{12})^T (\mathbf{q}_3 \times \mathbf{q}_2) \quad (1c)$$

All the parameters in Eqs. (1) should be expressed in the same coordinate system using the appropriate rotation matrices taken from the navigation systems of the vehicles. It is assumed that this coordinate system is the LLLN system of the vehicle that captured the second image at t_2 . Since the navigation parameters involved in formulating the three-view constraints (1) (such as rotation matrices from LLLN to camera frames) contain errors, these constraints

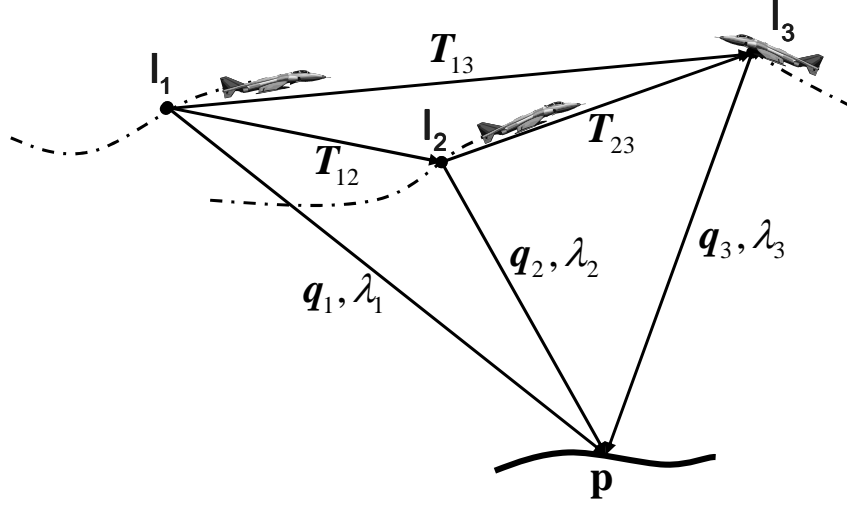


Fig. 2. Three-view geometry: a static landmark \mathbf{p} observed by three different vehicles. Images I_1 and I_2 were captured by the first two vehicles in the past, while image I_3 , is the currently-acquired image by the third vehicle.

will not be satisfied in practice. The next section discusses how to estimate these errors and subsequently perform navigation aiding.

Recall that Eqs. (1a) and (1b) are the well-known epipolar constraints [22], forcing the translation vectors to be co-planar with the LOS vectors. Thus, given multiple matching features, the translation vectors \mathbf{T}_{12} and \mathbf{T}_{23} can be determined only up to scale. The two scale parameters are different in the general case. Eq. (1c) relates between these two scale parameters, thereby allowing to calculate one of the translation vectors given the other [18], [19].

In practice, the three views will have more than a single landmark in common. Therefore, three different sets of matching LOS vectors are defined: A set of matching pairs of features between the first and second view, another set of matching pairs of features between the second and third view, and a set of matching triplets between all the three views. All the LOS vectors are expressed in an LLLN system, as mentioned above. These sets are denoted by $\{\mathbf{q}_{1_i}, \mathbf{q}_{2_i}\}_{i=1}^{N_{12}}$, $\{\mathbf{q}_{2_i}, \mathbf{q}_{3_i}\}_{i=1}^{N_{23}}$ and $\{\mathbf{q}_{1_i}, \mathbf{q}_{2_i}, \mathbf{q}_{3_i}\}_{i=1}^{N_{123}}$, respectively, where N_{12} , N_{23} and N_{123} are the number of matching features in each set, and \mathbf{q}_{j_i} is the i th LOS vector in the j th view, $j \in (1, 2, 3)$.

Taking into consideration all the matching pairs and triplets, the constraints (1) turn into [18]:

$$\begin{bmatrix} U \\ F \\ 0 \end{bmatrix}_{N \times 3} \mathbf{T}_{23} = \begin{bmatrix} W \\ 0 \\ G \end{bmatrix}_{N \times 3} \mathbf{T}_{12} \quad (2)$$

where $N \doteq N_{12} + N_{23} + N_{123}$ and

$$\begin{aligned} U &= [\mathbf{u}_1 \ \dots \ \mathbf{u}_{N_{123}}]^T & W &= [\mathbf{w}_1 \ \dots \ \mathbf{w}_{N_{123}}]^T \\ F &= [\mathbf{f}_1 \ \dots \ \mathbf{f}_{N_{23}}]^T & G &= [\mathbf{g}_1 \ \dots \ \mathbf{g}_{N_{12}}]^T \end{aligned}$$

while the vectors $\mathbf{f}, \mathbf{g}, \mathbf{u}, \mathbf{w} \in \mathbb{R}^{3 \times 1}$ are defined as

$$\begin{aligned} \mathbf{f}^T &\doteq (\mathbf{q}_2 \times \mathbf{q}_3)^T \\ \mathbf{g}^T &\doteq (\mathbf{q}_1 \times \mathbf{q}_2)^T \\ \mathbf{u}^T &\doteq (\mathbf{q}_1 \times \mathbf{q}_2)^T [\mathbf{q}_3]_{\times} = \mathbf{g}^T [\mathbf{q}_3]_{\times} \\ \mathbf{w}^T &\doteq (\mathbf{q}_2 \times \mathbf{q}_3)^T [\mathbf{q}_1]_{\times} = \mathbf{f}^T [\mathbf{q}_1]_{\times} \end{aligned}$$

Let \mathbf{x}_i denote the navigation solution calculated by the i th vehicle

$$\mathbf{x}_i = [\mathbf{Pos}_i^T \quad \mathbf{V}_i^T \quad \mathbf{\Psi}_i^T]^T \quad (3)$$

where \mathbf{Pos} , \mathbf{V} and $\mathbf{\Psi}$ are the position, velocity and attitude of the vehicle, respectively. Denote by \mathbf{x}_i^t the (unknown) true navigation solution and let $\mathbf{y}_{i,IMU}$ represent the measurements of the i th vehicle's inertial navigation sensors. The errors in $\mathbf{y}_{i,IMU}$ are modeled by an unknown vector of parameters β_i^t . Denote by β the calculated model of inertial sensor errors, used for correcting the measurements $\mathbf{y}_{i,IMU}$ (cf. Fig. 1). In this paper, a simple parameterization for β is used:

$$\beta = [\mathbf{d}_{IMU}^T \quad \mathbf{b}_{IMU}^T]^T \quad (4)$$

where $\mathbf{d}_{IMU} \in \mathbb{R}^3$ is the gyro drift, and $\mathbf{b}_{IMU} \in \mathbb{R}^3$ is the accelerometer bias.

Letting

$$\zeta_i(t_k) \doteq \begin{bmatrix} \mathbf{x}_i(t_k) \\ \beta_i(t_k) \end{bmatrix}, \quad \zeta_i^t(t_k) \doteq \begin{bmatrix} \mathbf{x}_i^t(t_k) \\ \beta_i^t(t_k) \end{bmatrix} \quad (5)$$

the navigation solution of the i th vehicle is given by

$$\zeta_i(t_{k+1}) = \mathbf{f}(\zeta_i(t_k), \mathbf{y}_{i,IMU}(t_k)) \quad (6)$$

The following navigation error state vector is defined

$$\mathbf{X}_i(t) \doteq \zeta_i(t) - \zeta_i^t(t) \quad (7)$$

The evolution of the state vector \mathbf{X} can be modeled by the linear time-varying stochastic model [31], [32]:

$$\dot{\mathbf{X}}_i(t) = \Phi^i(t)\mathbf{X}_i(t) + \omega^i(t) \quad (8)$$

where Φ^i is the continuous system matrix and ω^i is the process noise, which is assumed to be white and zero-mean Gaussian. This continuous time model can be replaced by the discrete model

$$\mathbf{X}_i(t_b) = \Phi_{t_a \rightarrow t_b}^i \mathbf{X}_i(t_a) + \omega_{t_a:t_b}^i \quad (9)$$

where $\Phi_{t_a \rightarrow t_b}^i$ is the discrete system matrix relating the state between any two time instances t_a and t_b , $t_b > t_a$, and $\omega_{t_a:t_b}^i$ is the equivalent discrete process noise. The state vector of each vehicle is defined as

$$\mathbf{X} = [\Delta \mathbf{P}^T \quad \Delta \mathbf{V}^T \quad \Delta \mathbf{\Psi}^T \quad \mathbf{d}^T \quad \mathbf{b}^T]^T$$

where $\Delta \mathbf{P} \in \mathbb{R}^3$, $\Delta \mathbf{V} \in \mathbb{R}^3$, $\Delta \mathbf{\Psi} = (\Delta \phi, \Delta \theta, \Delta \psi)^T \in [0, 2\pi) \times [0, \pi) \times [0, 2\pi)$ are the position, velocity and attitude errors, respectively, and \mathbf{d} and \mathbf{b} are the residual gyro drift and accelerometer bias, respectively:

$$\mathbf{d} \doteq \mathbf{d}_{IMU} - \mathbf{d}_{IMU}^t, \quad \mathbf{b} \doteq \mathbf{b}_{IMU} - \mathbf{b}_{IMU}^t \quad (10)$$

with \mathbf{d}_{IMU}^t , \mathbf{b}_{IMU}^t being the unknown true values of \mathbf{d}_{IMU} , \mathbf{b}_{IMU} . The position and velocity errors are expressed in the NED system, while \mathbf{d} and \mathbf{b} are given in the body-fixed reference frame. The corresponding transition matrix $\Phi_{t_a \rightarrow t_b}^i$ satisfying Eq. (9) for the i th vehicle is given in Appendix A.

Let $\mathbf{y}_i(t_i)$ denote the measurements of the external sensors of the i th vehicle at some time instant t_i (as opposed to $\mathbf{y}_{i,IMU}(t_i)$ that denotes the IMU measurements). These measurements are corrupted by a Gaussian white noise $\mathbf{v}_i(t_i)$. Let $\mathbf{y}_j^t(t_a) \doteq \mathbf{y}_j(t_a) - \mathbf{v}_j(t_a)$.

A general multi-vehicle measurement involving information from r vehicles can therefore be written as [21]:

$$\mathbf{z}(t) = \mathbf{h}(\{\zeta_i(t_i), \mathbf{y}_i(t_i)\}_{i=1}^r) \quad (11)$$

While the vehicles can be equipped with various external sensors, in this work we assume that each vehicle is equipped with a single camera only, and focus on a specific multi-vehicle measurement, dictated by the three-view constraints (2). Thus, $\mathbf{y}_i(t_i)$ in Eq. (11) represents the pixel coordinates of the image captured by the i th vehicle at t_i .

When real navigation and imagery data are considered, the constraints (2) will not be satisfied. Thus, a residual measurement is defined as

$$\mathbf{z} \doteq \begin{bmatrix} U \\ F \\ 0 \end{bmatrix}_{N \times 3} \mathbf{T}_{23} - \begin{bmatrix} W \\ 0 \\ G \end{bmatrix}_{N \times 3} \mathbf{T}_{12} \doteq \mathcal{A}\mathbf{T}_{23} - \mathcal{B}\mathbf{T}_{12} \quad (12)$$

Since \mathbf{z} is comprising of navigation and imagery data of three views, it can be expressed in a similar manner to Eq. (11) with $r = 3$:

$$\mathbf{z}(t) = \mathbf{h}(\zeta_1(t_1), \mathbf{y}_1(t_1), \zeta_2(t_2), \mathbf{y}_2(t_2), \zeta_3(t_3), \mathbf{y}_3(t_3)) \quad (13)$$

Noting that $\mathbf{T}_{12} = \mathbf{Pos}_2(t_2) - \mathbf{Pos}_1(t_1)$, $\mathbf{T}_{23} = \mathbf{Pos}_3(t_3) - \mathbf{Pos}_2(t_2)$, and the matrices F, G, U, W are functions of the LOS vectors, the residual measurement \mathbf{z} is a nonlinear function of the following parameters:

$$\mathbf{z} = \mathbf{h}\left(\mathbf{Pos}_3(t_3), \Psi_3(t_3), \mathbf{Pos}_2(t_2), \Psi_2(t_2), \mathbf{Pos}_1(t_1), \Psi_1, \left\{\mathbf{q}_{1_j}^{C_1}, \mathbf{q}_{2_j}^{C_2}, \mathbf{q}_{3_j}^{C_3}\right\}\right) \quad (14)$$

The notation $\left\{\mathbf{q}_{1_j}^{C_1}, \mathbf{q}_{2_j}^{C_2}, \mathbf{q}_{3_j}^{C_3}\right\}$ in Eq. (14) refers to the fact that LOS vectors from all the three images are used for calculating the residual measurement \mathbf{z} . Each of the matrices F, G, U, W is a function of a different set of matching points. These LOS vectors can be calculated from $\mathbf{y}_1(t_1), \mathbf{y}_2(t_2)$ and $\mathbf{y}_3(t_3)$, provided that the camera calibration parameters are known.

Linearizing Eq. (13) about $\zeta_i^t(t_k)$ and $\mathbf{y}_i^t(t_i)$ gives

$$\mathbf{z}(t) \approx H_3(t_3)\mathbf{X}_3(t_3) + H_2(t_2)\mathbf{X}_2(t_2) + H_1(t_1)\mathbf{X}_1(t_1) + D\mathbf{v} \quad (15)$$

where $D \doteq [D_1(t_1) \ D_2(t_2) \ D_3(t_3)]$ and

$$H_i(t_i) = \nabla_{\zeta_i^t(t_i)} \mathbf{h} \quad , \quad D_i(t_i) = \nabla_{\mathbf{y}_i^t(t_i)} \mathbf{h} \quad , \quad i = 1, 2, 3 \quad (16)$$

while $\mathbf{v} \doteq [\mathbf{v}_1^T(t_1) \ \mathbf{v}_2^T(t_2) \ \mathbf{v}_3^T(t_3)]^T$, with $\mathbf{v}_i(t_i)$ representing the image noise in the LOS vectors from the i th view. In addition, we denote $R = E[\mathbf{v}\mathbf{v}^T]$. Since $\zeta_i^t(t_k)$ and $\mathbf{y}_i^t(t_i)$ are unknown, the Jacobian matrices are approximated by

$$H_i(t_i) = \nabla_{\zeta_i(t_i)} \mathbf{h} \quad , \quad D_i(t_i) = \nabla_{\mathbf{y}_i(t_i)} \mathbf{h} \quad , \quad i = 1, 2, 3 \quad (17)$$

Explicit expressions for the Jacobian matrices $H_i(t_i)$ and $D_i(t_i)$ can be found in Appendix B. The residual measurement \mathbf{z} is a function of all the three state vectors, which in the general case can be correlated. Ignoring this correlation may result in an inconsistent and over-confident estimations [23], [24]. Assuming the cross-covariance terms relating the three state vectors are known, all the vehicles involved in the measurement can be updated. This can be performed by applying an implicit extended Kalman filter (IEKF), as described later.

The following augmented state vector is defined:

$$\mathcal{X} \doteq [\mathbf{X}_3^T(t_3) \ \mathbf{X}_2^T(t_2) \ \mathbf{X}_1^T(t_1)]^T \quad (18)$$

with an augmented covariance matrix $\mathcal{P} \doteq E[\tilde{\mathcal{X}}\tilde{\mathcal{X}}^T]$, where $\tilde{\mathcal{X}}$ denotes the estimation error of \mathcal{X} . Let also $\mathcal{H} \doteq [H_3(t_3) \ H_2(t_2) \ H_1(t_1)]$. The augmented Kalman gain matrix

$$\mathcal{K} \doteq [\check{K}_3^T \ \check{K}_2^T \ \check{K}_1^T]^T \quad (19)$$

is computed as [25]

$$\mathcal{K} = \mathcal{P}^{-1} \mathcal{H}^T (\mathcal{H} \mathcal{P}^{-1} \mathcal{H}^T + D R D^T)^{-1} \quad (20)$$

The a posteriori estimation error of the augmented state vector \mathcal{X} is

$$\tilde{\mathcal{X}}^+ = [I - \check{K} \check{H}] \tilde{\mathcal{X}}^- - \check{K} D \mathbf{v} \quad (21)$$

while the augmented covariance matrix is updated according to

$$\mathcal{P}^+ = [I - \mathcal{K} \mathcal{H}] \mathcal{P}^- [I - \mathcal{K} \mathcal{H}]^T + [\mathcal{K} D] R [\mathcal{K} D]^T \quad (22)$$

The a posteriori estimation errors of the three state vectors in \mathcal{X} can be explicitly written based on Eq. (21) as:

$$\tilde{\mathbf{X}}_3^+ = [I - \check{K}_3 H_3] \tilde{\mathbf{X}}_3^- - \check{K}_3 H_2 \tilde{\mathbf{X}}_2^- - \check{K}_3 H_1 \tilde{\mathbf{X}}_1^- - \check{K}_3 D \mathbf{v} \quad (23)$$

$$\tilde{\mathbf{X}}_2^+ = [I - \check{K}_2 H_2] \tilde{\mathbf{X}}_2^- - \check{K}_2 H_3 \tilde{\mathbf{X}}_3^- - \check{K}_2 H_1 \tilde{\mathbf{X}}_1^- - \check{K}_2 D \mathbf{v} \quad (24)$$

$$\tilde{\mathbf{X}}_1^+ = [I - \check{K}_1 H_1] \tilde{\mathbf{X}}_1^- - \check{K}_1 H_3 \tilde{\mathbf{X}}_3^- - \check{K}_1 H_2 \tilde{\mathbf{X}}_2^- - \check{K}_1 D \mathbf{v} \quad (25)$$

Having stated how all the vehicles can be updated when all the cross-covariance terms of Eq. (20) are known, we now explain how these terms can be actually calculated. The main difficulty in calculating the cross-covariance terms is that it is a priori unknown which vehicles and what time instances will appear in a three-view measurement. The common approach to mitigate this difficulty is to maintain the cross-covariance terms between all the possible pairs of vehicles using an augmented covariance matrix. This approach is suitable whenever the multi-vehicle measurement, \mathbf{z} , involves only concurrent information from different vehicles, i. e., $t_1 = t_2 = t_3 = t$, as in the case of relative pose measurements between pairs of vehicles (cf., e. g., [6]).

Given M vehicles in the group, and an $m \times m$ covariance matrix P_i for each vehicle i , the total covariance matrix of the group, containing also all the cross-covariance terms among vehicles in the group, is the $Mm \times Mm$ matrix

$$\mathcal{P}_{\text{Total}}(t) = \begin{bmatrix} P_1 & P_{12} & \cdots & P_{1M} \\ P_{21} & P_2 & \cdots & P_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ P_{M1} & P_{M2} & \cdots & P_M \end{bmatrix} \quad (26)$$

where $P_i(t) = E[\tilde{\mathbf{X}}_i(t) \tilde{\mathbf{X}}_i^T(t)]$ and $P_{ij}(t) = E[\tilde{\mathbf{X}}_i(t) \tilde{\mathbf{X}}_j^T(t)]$. The matrix $\mathcal{P}_{\text{Total}}$ can be efficiently calculated in a distributed manner (i. e. by every vehicle in the group) [6]. Yet, the measurement model in Eq. (15), involves data from different vehicles and from different time instances. Maintaining a total covariance matrix $\mathcal{P}_{\text{Total}}$ containing a covariance for every vehicle and cross-covariance terms between each pair of vehicles for any two time instances is impractical. Thus, an alternative technique should be used.

As a solution to the aforementioned problem, it is proposed to explicitly calculate the required cross-covariance terms based on an approach developed in [21] for a general multi-vehicle measurement model. This approach represents the multi-vehicle updates in a directed acyclic graph, locally maintained by every vehicle in the group. The required cross-covariance terms are computed based on this graph representation, going back and forth in the time domain according to the history of the so-far performed multi-vehicle updates.

Another possible approach for consistent data fusion has been recently developed in [3], considering relative pose and two-view measurements between pairs of vehicles. However, in this approach a smoothing over the navigation solution of all the vehicles is performed

whenever any kind of measurement is received. In contrast to this, the computationally-efficient approach used in this paper allows on-demand calculation of the required cross-covariance terms without refining the navigation history.

The graph needs to be acyclic, since otherwise a measurement might trigger recursive updates in past measurements. In a general scenario involving three-view measurements among different vehicles at different time instances, the graph is guaranteed to be acyclic, particularly if only vehicles that contributed their *current* (and not past) image and navigation data are updated. For simplicity, in this paper we consider only one such vehicle, which is the querying vehicle, i. e. the vehicle that broadcasts the query image to the rest of the vehicles in the group. Moreover, without loss of generality, it is assumed that the querying vehicle captures the *third* image, as illustrated in Fig. 2.

Thus, referring to Eq. (15), only $\mathbf{X}_3(t_3)$ is estimated, while $\mathbf{X}_2(t_2)$ and $\mathbf{X}_1(t_1)$ are modeled as random parameters. As a consequence, instead of applying Eqs. (18)-(25), the actual update equations are as follows. The a posteriori estimation error of the querying vehicle is given by:

$$\tilde{\mathbf{X}}_3^+ = (I - K_3 H_3) \tilde{\mathbf{X}}_3^- - K_3 H_2 \tilde{\mathbf{X}}_2^- - K_3 H_1 \tilde{\mathbf{X}}_1^- - K_3 D \mathbf{v} \quad (27)$$

while the a posteriori covariance is calculated as

$$P_3^+ = (I - K_3 H_3) P_3^- (I - K_3 H_3)^T + K_3 \mathcal{R} K_3^T \quad (28)$$

where

$$\mathcal{R} \doteq [H_2 \ H_1] \begin{bmatrix} P_2 & P_{21} \\ P_{21}^T & P_1 \end{bmatrix} [H_2 \ H_1]^T + D R D^T \quad (29)$$

and K_3 is the Kalman filter gain of the querying vehicle, which is calculated as

$$K_3 = P_{\mathbf{X}_3(t_3)\mathbf{z}(t_3,t_2,t_1)} P_{\mathbf{z}(t_3,t_2,t_1)}^{-1} \quad (30)$$

with

$$P_{\mathbf{X}_3(t_3)\mathbf{z}(t_3,t_2,t_1)} = P_3^- H_3^T + P_{32}^- H_2^T + P_{31}^- H_1^T \quad (31)$$

$$P_{\mathbf{z}(t_3,t_2,t_1)} = H_3 P_3^- H_3^T + \mathcal{R} \quad (32)$$

Referring to Eqs. (19), (20) and (30), one should note that $K_3 \neq \check{K}_3$.

Since only $\mathbf{X}_3(t_3)$ is estimated, $H_2 \mathbf{X}_2(t_2) + H_1 \mathbf{X}_1(t_1) + D \mathbf{v}$ is modeled as measurement noise (cf. Eq. (15)). However, $H_2 \mathbf{X}_2(t_2) + H_1 \mathbf{X}_1(t_1) + D \mathbf{v}$ is statistically dependent on $\mathbf{X}_3(t_3)$. Thus, one of the basic assumptions of the Kalman filter is violated. Eqs. (31) and (32) constitute an ad-hoc approach for accommodating the statistical dependence within the Kalman filter framework. If all the three state vectors, $\mathbf{X}_3(t_3)$, $\mathbf{X}_2(t_2)$ and $\mathbf{X}_1(t_1)$, were to be estimated (cf. Eqs. (18)-(25)), the measurement noise $D \mathbf{v}$ would still be statistically dependent on the state vectors. However, this dependence would only be due to the Jacobian D , as modeled by a standard IEKF formulation [25], [26].

It is worth mentioning that there are specific cases, in which *all* the vehicles participating in the measurement can be updated, since it is guaranteed that the graph will always be acyclic. In these cases, the filter formulation changes to Eqs. (18)-(25). An example of such a scenario is given in Section VII-B.

V. CROSS-COVARIANCE CALCULATION

In this section, we discuss how the cross-covariance terms can be calculated upon demand using a graph-based method. Recall that it is unknown a priori what vehicles and which time instances will participate in each three-view measurement. First, the development of expressions for the cross-covariance terms is presented in a simple scenario. Then, the concept of a graph-based technique for automatic calculation of these terms in general scenarios is discussed.

A. Simple Scenario

Consider the scenario shown in Fig. 3. Vehicle III is the querying vehicle, and therefore only this vehicle is updated. Two three-view updates are performed. In each of these updates, the first two images are transmitted by Vehicle I. For example, the first measurement is formulated using images and navigation data denoted by a_1, a_2 and a_3 , where a_1, a_2 are obtained from Vehicle I, and the second measurement is similarly formulated based on imagery and navigation data b_1, b_2 and b_3 (as shown in the figure). It is assumed that the vehicles do not apply any updates from other sources. As will be seen in Section VI-C, these restrictions are not necessary, and are undertaken here for clarifying the basics of the approach.

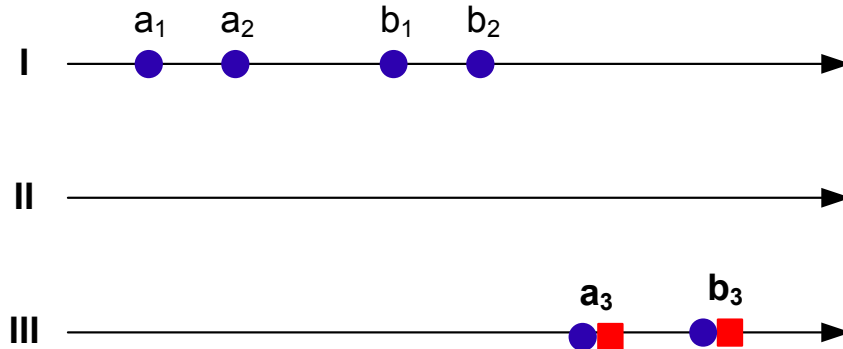


Fig. 3. Measurement schedule example. Vehicle III is updated based on images transmitted by vehicle I. The filled circle marks denote images participating in the measurement, square marks indicate update events.

The cross-covariance terms are computed in the following recursive way. Assume the first measurement, comprised of a_1, a_2 and a_3 was carried out, and that the a priori and a posteriori covariance and cross-covariance terms are known. Now, it is required to calculate the cross-covariance terms $E[(\tilde{\mathbf{X}}_{b_3}^-)(\tilde{\mathbf{X}}_{b_2}^-)^T]$, $E[(\tilde{\mathbf{X}}_{b_3}^-)(\tilde{\mathbf{X}}_{b_1}^-)^T]$ and $E[(\tilde{\mathbf{X}}_{b_2}^-)(\tilde{\mathbf{X}}_{b_1}^-)^T]$ for performing the second three-view update.

The following equations can be written for the state propagation:

$$\begin{aligned}\tilde{\mathbf{X}}_{b_3}^- &= \Phi_{a_3 \rightarrow b_3} \tilde{\mathbf{X}}_{a_3}^+ + \omega_{a_3:b_3} \\ \tilde{\mathbf{X}}_{b_i}^- &= \Phi_{a_2 \rightarrow b_i} \tilde{\mathbf{X}}_{a_2}^- + \omega_{a_2:b_i}, \quad i = 1, 2\end{aligned}$$

where $\omega_{i:j}$ is the equivalent process noise between the time instances t_i and t_j of the appropriate vehicle¹. The cross-covariance terms $E[(\tilde{\mathbf{X}}_{b_3}^-)(\tilde{\mathbf{X}}_{b_i}^-)^T]$ with $i = 1, 2$, may be calculated as:

$$E \left[\left(\Phi_{a_3 \rightarrow b_3} \tilde{\mathbf{X}}_{a_3}^+ + \omega_{a_3:b_3} \right) \left(\Phi_{a_2 \rightarrow b_i} \tilde{\mathbf{X}}_{a_2}^- + \omega_{a_2:b_i} \right)^T \right] \quad (33)$$

The a posteriori estimation error $\tilde{\mathbf{X}}_{a_3}^+$ is given, according to Eq. (27), by:

$$\begin{aligned}\tilde{\mathbf{X}}_{a_3}^+ &= (I - K_{a_3} H_{a_3}) \tilde{\mathbf{X}}_{a_3}^- - K_{a_3} H_{a_2} \tilde{\mathbf{X}}_{a_2}^- - \\ &\quad - K_{a_3} H_{a_1} \tilde{\mathbf{X}}_{a_1}^- - K_{a_3} D_a \mathbf{v}_a\end{aligned} \quad (34)$$

¹From now on, explicit notations of vehicle identities and time instances are omitted for conciseness, since these may be concluded by context.

Since $\omega_{a_2:b_2}$ is statistically independent of $\tilde{\mathbf{X}}_{a_3}^-, \tilde{\mathbf{X}}_{a_2}^-, \tilde{\mathbf{X}}_{a_1}^-$, and since $\omega_{a_3:b_3}$ is statistically independent of $\tilde{\mathbf{X}}_{a_2}^-$ and $\omega_{a_2:b_2}$ (cf. Fig. 3):

$$\begin{aligned} E \left[\tilde{\mathbf{X}}_{a_3}^+ \omega_{a_2:b_2}^T \right] &= 0 \\ E \left[\omega_{a_3:b_3} \left(\Phi_{a_2 \rightarrow b_2} \tilde{\mathbf{X}}_{a_2}^- + \omega_{a_2:b_2} \right)^T \right] &= 0 \end{aligned}$$

Denoting $P_{ab} \doteq E[(\tilde{\mathbf{X}}_a)(\tilde{\mathbf{X}}_b)^T]$ and incorporating the above into Eq. (33) yields

$$\begin{aligned} P_{b_3b_2}^- &= \Phi_{a_3 \rightarrow b_3} \left\{ (I - K_{a_3} H_{a_3}) P_{a_3a_2}^- - \right. \\ &\quad \left. - K_{a_3} H_{a_2} P_{a_2a_2}^- - K_{a_3} H_{a_1} P_{a_1a_2}^- \right\} \Phi_{a_2 \rightarrow b_2}^T \end{aligned} \quad (35)$$

where the expectation terms involving a_3, a_2, a_1 are known (from previous update). In a similar manner we get

$$\begin{aligned} P_{b_3b_1}^- &= \Phi_{a_3 \rightarrow b_3} \left\{ (I - K_{a_3} H_{a_3}) P_{a_3a_2}^- - \right. \\ &\quad \left. - K_{a_3} H_{a_2} P_{a_2a_2}^- - K_{a_3} H_{a_1} P_{a_1a_2}^- \right\} \Phi_{a_2 \rightarrow b_1}^T \end{aligned} \quad (36)$$

while $P_{b_2b_1}^- \doteq E[(\tilde{\mathbf{X}}_{b_2}^-)(\tilde{\mathbf{X}}_{b_1}^-)^T]$ is given by

$$P_{b_2b_1}^- = \Phi_{b_1 \rightarrow b_2} P_{b_1b_1}^- \quad (37)$$

B. Graph-Based Cross-Covariance Calculation

A directed acyclic graph (DAG) $G = (V, E)$ is locally maintained by every vehicle in the group, where V is the set of nodes and E is the set of directed arcs connecting between the nodes in V . Two type of nodes exist in this graph: nodes that represent images and the attached navigation data that participated in some multi-vehicle update, and update-event nodes. The nodes are connected by directed weighted arcs. The weight of each arc reflects the information flow between the two connected nodes. Each node in G can be connected to another node by a transition relation, and, in addition, it may be involved in a three-view measurement, in which case it would be also connected to an update-event node by an update relation.

The transition relation is given by Eq. (9), relating between the state vectors of the i th vehicle at two different time instances t_a and t_b . G will contain two nodes, representing these two time instances, only if each of them participates in some three-view measurement. In this case, these two nodes will be connected by an arc, weighted by the transition matrix $\Phi_{t_a \rightarrow t_b}^i$. The noise process covariance matrix $Q_{t_a:t_b}^i \doteq E[\omega_{t_a:t_b}^i (\omega_{t_a:t_b}^i)^T]$ is associated to this arc as well.

The update relation is given by Eq. (27):

$$\tilde{\mathbf{X}}_3^+ = [I - K_3 H_3] \tilde{\mathbf{X}}_3^- - K_3 H_2 \tilde{\mathbf{X}}_2^- - K_3 H_1 \tilde{\mathbf{X}}_1^- - K_3 D \mathbf{v}$$

Thus, G will contain 4 nodes representing the above equation. Let the nodes β_i represent the a priori estimation errors $\tilde{\mathbf{X}}_i^-$, with $i = 1, 2, 3$, and the node α represent the a posteriori estimation error $\tilde{\mathbf{X}}_3^+$. Then, the arc weights connecting the nodes β_1, β_2 and β_3 with the node α are $-K_3 H_1, -K_3 H_2$ and $I - K_3 H_3$, respectively. Each such arc is also associated with the relevant measurement noise covariance matrix [21]. It is assumed that the a priori and a posteriori covariance and cross-covariance terms between the nodes that participated in the *same* multi-vehicle update, which had already been carried out, are known (this information may be stored in the nodes themselves).

Consider, for example, the simple scenario discussed in the previous section (cf. Fig. 3). The equivalent graph is given in Fig. 4(a). As seen, two update events are carried out, both

on Vehicle III. At each update, the first two images of the three are transmitted by Vehicle I, while the third image is the currently-captured image by the querying Vehicle III. Vehicle II has not transmitted any images and therefore has no nodes in the graph. The transmission action is denoted by a dashed arc in the graph. Nodes of the first type are denoted as circle nodes, while the second-type nodes are designated by a square notation. The arc weights are not explicitly specified in the graph (for clarity of presentation). For example, the weight of an arc connecting between the nodes a_1^- and a_2^- is the transition matrix $\phi_{a_1 \rightarrow a_2}$, since no measurement updates were performed between these two time instances. On the other hand, the arcs connecting a_1^-, a_2^- and a_3^- to a_3^+ are weighted, according to Eq. (34), as $-K_{a_3} H_{a_1}$, $-K_{a_3} H_{a_2}$ and $I - K_{a_3} H_{a_3}$, respectively. In addition, each arc is also associated with the appropriate noise covariance, as mentioned above.

The graph representation suggests a convenient approach for computing the correlation terms. Assume we need to calculate the cross-covariance between some two nodes c and d in the graph, representing \tilde{X}_c and \tilde{X}_d , respectively. The first step is to construct two inverse-trees, containing all the possible routes in the graph G to each of the nodes c and d . This is performed as follows. The first tree, T_c , is initialized with the node c . Each next level is comprised of the parents of nodes that reside in the previous level, as determined from the graph G . Thus, for example, the second level of T_c contains all the nodes in G that are directly connected to c . The same process is executed for constructing a tree T_d for the node d . Fig. 4(b) shows an example of two trees with $c \doteq b_3^-$ and $d \doteq b_1^-$, constructed based on the graph, given in Fig. 4(a), for calculating the cross-covariance $P_{b_3 b_1}^-$. This term and the terms $P_{b_3 b_2}^-, P_{b_2 b_1}^-$ are required for carrying out the measurement update b_3^+ .

Note that each node in T_c and T_d has only one child but may have one or three parents. In the latter case, the node represents an update event.

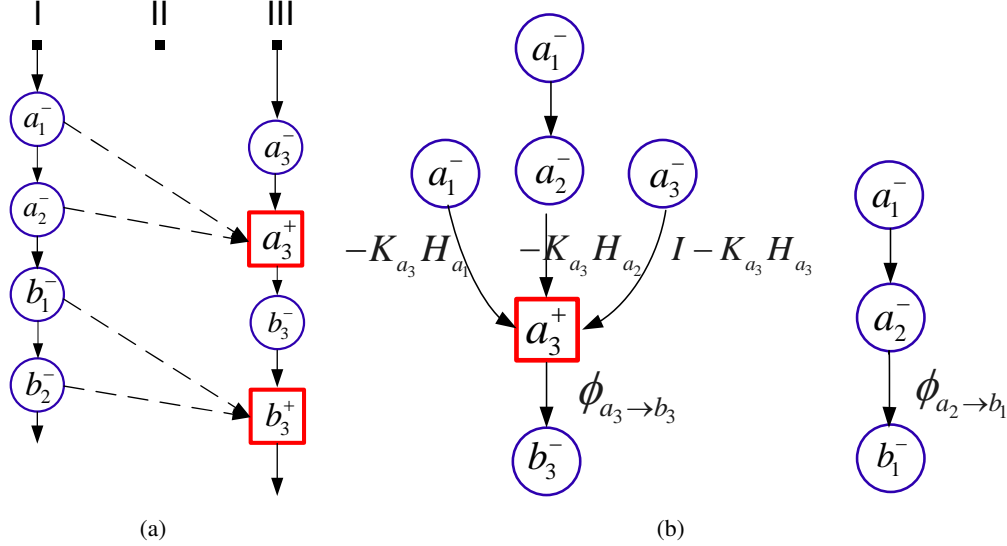


Fig. 4. (a) Graph representation for the scenario shown in Fig. 3. (b) The two inverse-trees $T_{b_3^-}$ and $T_{b_1^-}$ required for calculating $P_{b_3 b_1}^-$.

The concept of the proposed graph-based approach for calculating the cross-covariance terms is as follows. We start with the two nodes c and d , which are the first-level nodes in the trees T_c and T_d , respectively. Since the term $E[\tilde{X}_c \tilde{X}_d^T]$ is unknown, we proceed to the parents of these nodes. As noted above, two types of relations exist for a general graph topology. Assume, for example, that both of the nodes c and d are related to the nodes in the next level by a transition relation, and therefore have only one parent. This assumption is made

only for clarity of explanation². Denote the parents of c and d as c_2 and d_2 , respectively. The nodes c_2 and d_2 constitute the second level in the trees T_c and T_d , respectively. For example, c_2 and c are connected via $\tilde{\mathbf{X}}_c = \Phi_{c_2 \rightarrow c} \tilde{\mathbf{X}}_{c_2} + \boldsymbol{\omega}_{c_2 \rightarrow c}$. The convention used here is that if some node a_i has several parents, the j th parent is denoted as a_{i+1}^j . Also, $a \equiv a_1$.

Now, the required cross-covariance term $P_{cd} \doteq E[\tilde{\mathbf{X}}_c \tilde{\mathbf{X}}_d^T]$ may be written in several forms:

$$\begin{aligned} & E \left[\tilde{\mathbf{X}}_c \left(\Phi_{d_2 \rightarrow d} \tilde{\mathbf{X}}_{d_2} + \boldsymbol{\omega}_{d_2:d} \right)^T \right] \\ & E \left[\left(\Phi_{c_2 \rightarrow c} \tilde{\mathbf{X}}_{c_2} + \boldsymbol{\omega}_{c_2:c} \right) \tilde{\mathbf{X}}_d^T \right] \\ & E \left[\left(\Phi_{c_2 \rightarrow c} \tilde{\mathbf{X}}_{c_2} + \boldsymbol{\omega}_{c_2:c} \right) \left(\Phi_{d_2 \rightarrow d} \tilde{\mathbf{X}}_{d_2} + \boldsymbol{\omega}_{d_2:d} \right)^T \right] \end{aligned}$$

Since the expression from the previous level (the first level) was already checked, it is now required to check whether any of the expressions involving nodes from the current level are known. In other words, the question is whether any of the pairs P_{cd_2} , P_{c_2d} and $P_{c_2d_2}$ are known. In addition, it is also required to know the correlation between the noise terms and the state vectors.

Assuming none of the pairs is known, we proceed to the next level, the third level. *Each* node in the second level may have either transition or update relation, given by Eqs. (9) and (27), respectively, with the nodes in the third level. In this case, since the second level contains only a single node in each tree (c_2 and d_2), there are four possible cases: transition relation for c_2 and update relation for d_2 ; update relation for c_2 and transition relation for d_2 ; update relations for c_2 and d_2 ; transition relations for c_2 and d_2 . At this point, we choose to analyze the first case. Other cases are treated in a similar manner.

Thus, c_2 has a single parent, denoted by c_3 , while d_2 has three parents denoted by d_3^1 , d_3^2 and d_3^3 (in this case d_2 represents an update event, while d_3^1 , d_3^2 , d_3^3 represent the three participating images and the attached navigation data). The relations for c_2 and d_2 can be written as

$$\begin{aligned} \tilde{\mathbf{X}}_{c_2} &= \Phi_{c_3 \rightarrow c_2} \tilde{\mathbf{X}}_{c_3} + \boldsymbol{\omega}_{c_3:c_2} \\ \tilde{\mathbf{X}}_{d_2} &= A_{d_3^3} \tilde{\mathbf{X}}_{d_3^3} + A_{d_3^2} \tilde{\mathbf{X}}_{d_3^2} + A_{d_3^1} \tilde{\mathbf{X}}_{d_3^1} + A_{d_3^{123}} \mathbf{v}_{d_3^{123}} \end{aligned}$$

where $A_{d_3^3} \doteq (I - K_{d_3^3} H_{d_3^3})$, $A_{d_3^2} \doteq -K_{d_3^2} H_{d_3^2}$, $A_{d_3^1} \doteq -K_{d_3^1} H_{d_3^1}$ and $A_{d_3^{123}} \doteq -K_{d_3^{123}} D_{d_3^{123}}$.

Having reached a new level, the third level, new expressions for the required term $E[\tilde{\mathbf{X}}_c \tilde{\mathbf{X}}_d^T]$ may be written utilizing nodes from *this* level and *lower* levels. Note that all the expressions from the previous (second) level were already analyzed.

Assume that some term, for example, $E[\tilde{\mathbf{X}}_{c_3} \tilde{\mathbf{X}}_{d_3^3}^T]$, is known, which means that the nodes c_3 and d_3^3 in T_c and T_d , respectively, either represent images that participated in the same three-view update in the past, or that these two nodes are identical ($c_3 \equiv d_3^3$). In any case, the known term $E[\tilde{\mathbf{X}}_{c_3} \tilde{\mathbf{X}}_{d_3^3}^T]$, accordingly weighted, is part of $E[\tilde{\mathbf{X}}_c \tilde{\mathbf{X}}_d^T]$.

Having a known term also means that there is no need to proceed to nodes of higher levels that are related to this term. In the case of a known $E[\tilde{\mathbf{X}}_{c_3} \tilde{\mathbf{X}}_{d_3^3}^T]$, we would not proceed to the parents of c_3 and d_3^3 , unless this is required by the *unknown* terms in the current level. In this example, if the unknown terms are $E[\tilde{\mathbf{X}}_{c_3} \tilde{\mathbf{X}}_{d_3^2}^T]$ and $E[\tilde{\mathbf{X}}_{c_3} \tilde{\mathbf{X}}_{d_3^1}^T]$, then we would proceed to the parents of c_3 in T_c , and of d_3^1 and d_3^2 in T_d , but *not* to the parents of d_3^3 in T_d .

The procedure proceeds to higher levels until either all the terms required for calculating the cross-covariance $E[\tilde{\mathbf{X}}_c \tilde{\mathbf{X}}_d^T]$ are known, or reaching the top level in both trees. In the latter case, the unknown terms of cross-covariance are actually zero.

²In practice, c and d will usually represent images that are going to participate in a three-view update event, and therefore c and d will indeed have only one parent each.

The process noise terms are assumed to be statistically independent of each other, $E[\omega_{i_1:i_2} \omega_{j_1:j_2}^T] = 0$, if $\omega_{i_1:i_2}$ and $\omega_{j_1:j_2}$ belong to different vehicles, or in the case the two noise terms belong to the same vehicle but $(t_{i_1}, t_{i_2}) \cap (t_{j_1}, t_{j_2}) = \phi$. The measurement noise is assumed to be statistically independent with the process noise. On the other hand, the process and measurement noise terms are *not* necessarily statistically independent with the involved state vectors. Their contribution to the required cross-covariance $E[\tilde{\mathbf{X}}_c \tilde{\mathbf{X}}_d^T]$ is analyzed in [21]. Thus, in the general case, the process and measurement noise covariance matrices are part of the calculated cross-covariance terms, while the measurement noise covariance matrices participate also in Eqs. (27)-(32).

VI. OVERALL DISTRIBUTED SCHEME

Given M cooperative vehicles, each, or some, of these vehicles maintain a repository of captured images attached with navigation data. All the vehicles maintain a local copy of the graph, that is updated upon every multi-vehicle update event. This graph contains M threads, one thread for each vehicle in the group. The graph is initialized to M empty threads. The formulation of a single multi-vehicle update event is as follows.

The querying vehicle broadcasts its currently-captured image and its navigation solution to the rest of the vehicles. A vehicle that receives this query, performs a check in its repository to see whether it has previously captured images of the same region. Vehicles that do not maintain such a repository perform this check over the currently captured image only. Different procedures for performing this query may be devised. One possible alternative is to check only those images in the repository that have a reasonable navigation data attached, e. g. images that were captured from a vicinity of the transmitted position of the querying vehicle.

Among the chosen images, only images that have a smaller uncertainty in their attached navigation data, compared to the uncertainty in the transmitted navigation data of the querying vehicle, are transmitted back to the querying vehicle. Specifically, denote by P_Q the covariance matrix of the querying vehicle, and by P the covariance matrix attached to one of the chosen images from a repository of some other vehicle in the group. Then, in our current implementation, this image is transmitted back to the querying vehicle only if its position uncertainty is smaller than the position uncertainty of the querying vehicle, i. e.:

$$(P)_{ii} < \alpha(P_Q)_{ii} \quad , \quad i = 1, 2, 3 \quad (38)$$

where $(A)_{ij}$ is the ij entry of some matrix A , and α is a constant satisfying $0 < \alpha \leq 1$. Other criteria may be applied as well.

The chosen images, satisfying the above condition, are transmitted to the querying vehicle, along with their attached navigation data. In addition, transition matrices between the transmitted images are sent, as well as the image noise covariance matrices. In case the replying vehicle has already participated in at least one multi-vehicle update of any vehicle in the group, its thread in the graph will contain at least one node. Therefore, transition matrices bridging the navigation data attached to the images being transmitted in the current multi-vehicle update to the closest nodes in this thread are also sent.

As an example, consider the scenario shown in Fig. 5. Fig. 6 presents the construction details of the graph for this scenario, for each of the executed three-view measurement updates. Assume the first update, a_3^+ , was performed, and focus on the second update, b_3^+ . As shown in Fig. 6(b), vehicle I transmits two images and navigation data, denoted by the nodes b_1^- and b_2^- in the graph. This navigation data comprises the navigation solution and the covariance matrix that were attached to each image that was stored in the vehicle's repository (cf. Section II). However, in addition to the transmitted transition matrix and process noise covariance matrix between these two nodes, $\phi_{b_1 \rightarrow b_2}$ and $Q_{b_1 \rightarrow b_2}$, the transition matrix and noise covariance matrix between the nodes b_2^- and a_3^+ , $\phi_{b_2 \rightarrow a_3}$ and $Q_{b_2 \rightarrow a_3}$, are transmitted as well.

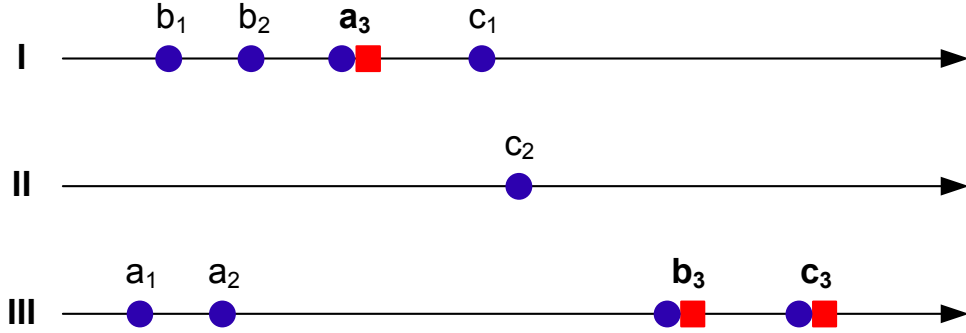


Fig. 5. Three-vehicle scenario

TABLE I
COMMUNICATION BETWEEN THE VEHICLES FOR UPDATING VEHICLE III AT TIME INSTANT t_{b_3} IN THE EXAMPLE
CONSIDERED IN FIG. 6(B).

#	Description	What is transmitted
1	Vehicle III broadcasts its current image, navigation solution and covariance	$I_1(t_{b_3})$ $\mathbf{x}_1(t_{b_3}), P_{b_3 b_3}^- \doteq P_1^-(t_{b_3})$
2	Vehicle I transmits imagery and navigation data from t_{b_1} and t_{b_2} to Vehicle III	$I_1(t_{b_1}), \mathbf{x}_1(t_{b_1}), P_{b_1 b_1}^- \doteq P_1^-(t_{b_1}), R_{b_1} \doteq R_1(t_{b_1}),$ $I_1(t_{b_2}), \mathbf{x}_1(t_{b_2}), P_{b_2 b_2}^- \doteq P_1^-(t_{b_2}), R_{b_2} \doteq R_1(t_{b_2})$
3	Vehicle I transmits additional data	$\phi_{b_1 \rightarrow b_2}, \phi_{b_2 \rightarrow a_3}, Q_{b_1 \rightarrow b_2}, Q_{b_2 \rightarrow a_3}$
4	Vehicle III performs an update	-
5	Vehicle III broadcasts update information: Identities of involved vehicles and time instances Additional required matrices Filter matrices Involved a priori covariance Calculated a priori cross-covariance Calculated a posteriori covariance and cross-covariance	Vehicles I and III, $t_{b_1}, t_{b_2}, t_{b_3}$ $\phi_{b_1 \rightarrow b_2}, \phi_{b_2 \rightarrow a_3}, Q_{b_1 \rightarrow b_2}, Q_{b_2 \rightarrow a_3}$ $K_3, H_i(t_{b_i}), D_i(t_{b_i}), R_i(t_{b_i}), i = 1, 2, 3$ $P_{b_1 b_1}^-, P_{b_2 b_2}^-, P_{b_3 b_3}^-$ $P_{b_1 b_2}^-, P_{b_1 b_3}^-, P_{b_2 b_3}^-$ $P_{b_3 b_3}^+$ (and $P_{b_3 b_2}^+, P_{b_3 b_1}^+$ if Vehicles I and II were updated)

Upon receiving the transmitted images and the navigation data, two best images are selected³ and the cross-covariance terms are calculated based on the local graph, as discussed in Section V, followed by computation of all the relevant filter matrices: $H_3, H_2, H_1, \mathcal{A}, \mathcal{B}, D$. Next, the update of the querying vehicle is carried out based on Section IV.

Now, it is only required to update the local graphs of all the vehicles in the group by the performed update event. The querying vehicle broadcasts the following information: *a*) identity of the involved vehicles in the current update; *b*) time instances (or some other identifiers) of the involved images; required transition matrices of the involved images; *c*) a priori covariance and a posteriori cross-covariance matrices; *d*) filter matrices K_3, H_3, H_2 and H_1 and the required noise covariance matrices. Then, each vehicle updates its own graph representation. This communication procedure is elaborated in Table I, which presents what is being actually transmitted for the second update of Vehicle III in the example considered in Fig. 6(b).

The process described above is summarized in Algorithms 1 and 2. Algorithm 1 contains a protocol of actions carried out by the querying vehicle, while Algorithm 2 provides the protocol of actions for the rest of the vehicles in the group. The computational complexity

³The selection is according to some criteria, e. g., Eq. (38). Alternatively, the proposed approach may be also applied on more than three images [18], [19].

for updating a single vehicle can be bounded by $O(n^2 \log(n))$, where n is the number of the thus-far performed three-view measurement updates that are represented in the graph [19], [24]. Moreover, the involved communication cost is bounded by $O(Nn^2)$, with N being the number of vehicles in the group.¹⁶

It is important to mention, that the distributed algorithm does not make any further approximations other than those mentioned in Section IV: The algorithm is exact⁴ if all the vehicles involved in the three-view measurements can be updated while keeping the graph acyclic. If keeping the graph acyclic means that only part of the involved vehicles can be updated, an approximated solution is obtained due to the applied ad-hoc approach for calculating the filter's gain matrix (cf. Section IV). The performance of the algorithm in such a scenario is provided and analyzed in Section VII-B.

While it is tempting to compare the proposed approach with bundle adjustment (BA) methods, we believe this comparison is premature since extending the method presented herein to a BA framework is outside the scope of this paper.

However, it is important to note that one key difference between the proposed method and approximated BA methods, such as sliding-window BA (e. g. [34]), is the ability to handle loops, in which a vehicle re-observes the same scene after some a priori unknown time: Such scenarios cannot be handled by sliding-window BA methods if some of the involved information is not represented in the sliding-window state vector. The proposed method, on the other hand, allows to handle loops, as well as multi-vehicle measurements that involve similar challenges⁵, by: a) storing only essential information in the graph (cf. Section V-B), while the rest is stored in the local hierarchical repositories (cf. Section VI-B); b) computing an approximated solution for only part of the involved vehicles (or time instances) if the acyclic graph topology requires so. These two aspects allow calculating the cross-covariance terms by going over the graph, without being forced to go through all the information that would be processed in a smoothing approach (i.e. the navigation history stored in the hierarchical repositories). Naturally, applying smoothing is expected to produce better results at the cost of additional computations.

A. Vehicles Joining or Leaving the Group

Whenever a vehicle joins an existing group of cooperative vehicles, it must obtain the graph describing the history of multi-vehicle updates among the vehicles in the group. This graph may be transmitted to the joining vehicle by one of the vehicles in the group. Departure of a vehicle from the group does not require any specific action.

An interesting scenario is one in which there are *several* groups of cooperative vehicles, and a vehicle has to migrate from one group to another. Refer the former and the latter groups as the source and destination groups. For example, this might be the case when each cooperative group operates in a distinct location and there is a need to move a vehicle within these groups. In these scenarios the migrating vehicle has already a local graph representing the multi-vehicle events of the source group, while the destination group has its own graph.

These two graphs have no common threads only when each vehicle is assigned only to one group, and, in addition, no migration between the groups have occurred in the past. In any case, upon receiving the graph of the destination group, the joining vehicle may fuse the two graphs and broadcast the updated graph to all the vehicles in the destination group.

⁴Up to the standard linearization approximations involved with the extended Kalman filter framework.

⁵For example, multi-vehicle three-view measurements that involve considerably different time instances, between which additional multi-vehicle measurements were executed.

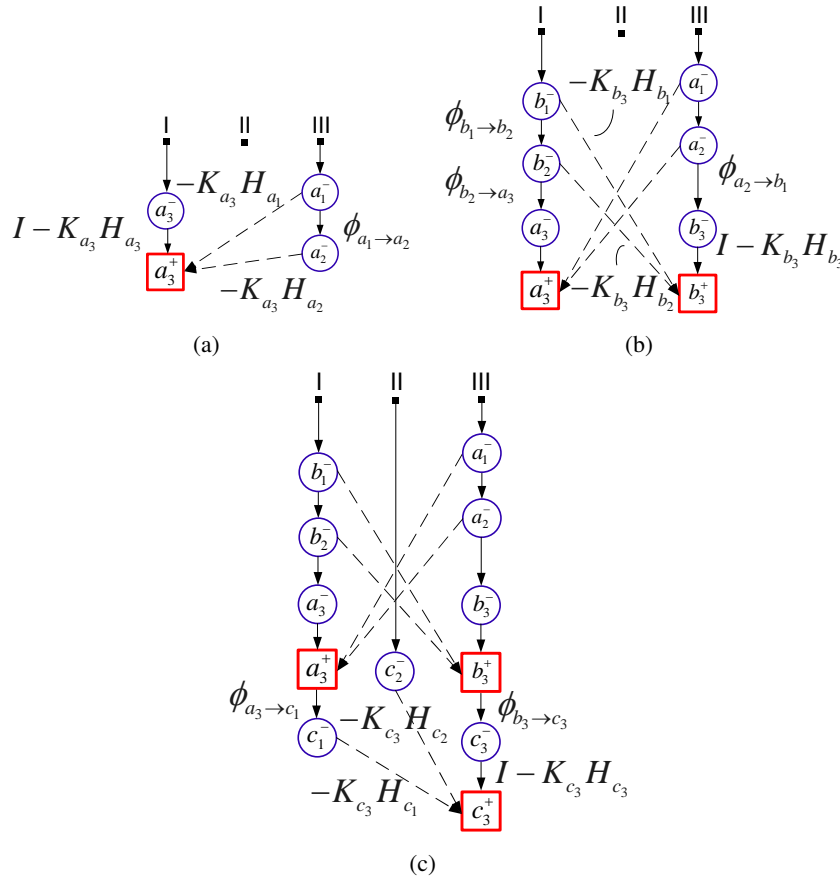


Fig. 6. Graph update process: **a**) update event a_3^+ ; **b**) update event b_3^+ ; **c**) update event c_3^+ .

B. Efficient Calculation of Transition and Process Noise Covariance Matrices

The problem this section refers to is calculating the transition matrix and the process noise covariance matrix between some two time instances that are unknown a priori. These matrices participate in calculation of the cross-covariance terms, as explained in Section V. We first handle calculation of transition matrices. Recall that each image stored in the vehicle repository is associated with navigation parameters taken when the image was captured.

A naive approach for calculating the transition matrix $\phi_{i \rightarrow j}$ between some image i to some other image j in the repository would be based on

$$\phi_{i \rightarrow j} = \phi_{j-1 \rightarrow j} \cdot \dots \cdot \phi_{i \rightarrow i+1} \quad (39)$$

However, a much more time-efficient alternative is to calculate $\phi_{i \rightarrow j}$ using transition matrices bridging between several time instances. For example, if we had the matrix $\phi_{i \rightarrow j-1}$, the computation of $\phi_{i \rightarrow j}$ would require multiplication of only two matrices: $\phi_{i \rightarrow j} = \phi_{j-1 \rightarrow j} \cdot \phi_{i \rightarrow j-1}$. This concept may be realized by maintaining a skip list [27] type database. The lowest level is comprised of the stored images and the associated navigation data, including the transition matrices between adjacent stored images. This level is a possible implementation of the repository maintained by all/some vehicles. Each next level is constructed by skipping several nodes in the lower level, and assigning the appropriate transition matrix, transferring from previous node to next node in the same level. No other data are stored outside the first level nodes.

An example of this concept is given in Fig. 7, in which every two nodes in some level contribute a node in the next level. Thus, for instance, calculation of $\phi_{2 \rightarrow 5}$ may be performed

-
- 1: Notations: Q - Querying vehicle; A, B - two other vehicles.
 - 2: Broadcast current image I^Q and current navigation data.
 - 3: Receive a set of images and associated navigation data from other vehicles. See steps 2-11 in Algorithm 2.
 - 4: Choose two best images I^A, I^B transmitted by vehicles A and B , respectively.
 - 5: **First graph update:**
 - Add a new node for each image in the appropriate thread (A, B and Q) in the graph.
 - Denote these three new nodes in threads A, B and Q by β_1, β_2 and β_3 , respectively.
 - Connect each such node to previous and next nodes (if exist) in its thread by directed arcs associated with the transition matrices and with the process noise covariance matrices.
 - 6: Calculate cross-covariance terms based on the local graph (cf. Section V).
 - 7: Calculate the measurement \mathbf{z} and the filter matrices K_3, H_3, H_2, H_1, D based on the three images I^A, I^B, I^Q and the attached navigation data.
 - 8: Perform navigation update on vehicle Q .
 - 9: **Final graph update:**
 - Add an update-event node, denoted by α , in the thread Q .
 - Connect the nodes β_1, β_2 and β_3 to the update-event node α by directed arcs weighted as $-K_3H_1, -K_3H_2$ and $I - K_3H_3$, respectively. Associate also measurement noise covariance matrix to each arc.
 - Store a priori and a posteriori covariance and cross-covariance terms (e. g. in the nodes $\beta_1, \beta_2, \beta_3$ and α).
 - 10: Broadcast update event information.
-

Algorithm 2 Replying Vehicle Protocol

-
- 1: Notations: Q - Querying vehicle; A - current vehicle.
 - 2: **if** a query image and its navigation data are received **then**
 - 3: Search repository for images containing the same scene.
 - 4: Choose images that satisfy the navigation uncertainty criteria (38).
 - 5: For each chosen image, captured at some time instant k , look among all the nodes in thread A in the local graph, for two nodes with time l and m that are closest to k , such that $l < k < m$.
 - 6: Calculate transition matrices $\phi_{l \rightarrow k}$ and $\phi_{k \rightarrow m}$ and noise covariance matrices $Q_{l:k}$ and $Q_{k:m}$.
 - 7: **if** more than one image was chosen in step 4 **then**
 - 8: Calculate transition matrices and noise covariance matrices between the adjacent chosen images.
 - 9: **end if**
 - 10: Transmit the chosen images, their navigation data and the calculated transition and noise covariance matrices to the querying vehicle Q .
 - 11: **end if**
 - 12: **if** update message is received from Q **then**
 - 13: Update local graph following steps 5 and 9 in Algorithm 1.
 - 14: **end if**
-

by searching for the appropriate route in the skip list formation, which will yield $\phi_{2 \rightarrow 5} = \phi_{3 \rightarrow 5} \phi_{2 \rightarrow 3}$, instead of carrying out the three matrix multiplications $\phi_{2 \rightarrow 5} = \phi_{4 \rightarrow 5} \phi_{3 \rightarrow 4} \phi_{2 \rightarrow 3}$.

The process noise covariance matrix between any two time instances may be efficiently¹⁹ calculated following a similar approach. For example, $Q_{i:j}$, for general i and j , $i < j$ is given by

$$Q_{i:j} = Q_{j-1:j} + \Phi_{j-1 \rightarrow j} Q_{j-2:j-1} \Phi_{j-1 \rightarrow j}^T + \dots + \Phi_{i+1 \rightarrow j} Q_{i:i+1} \Phi_{i+1 \rightarrow j}^T$$

However, if each node in the skip list database contains the noise covariance matrix between the previous node in the same level, $Q_{i:j}$ may be also calculated, for instance, as

$$Q_{i:j} = Q_{j-1:j} + \Phi_{j-1 \rightarrow j} Q_{i:j-1} \Phi_{j-1 \rightarrow j}^T \quad (40)$$

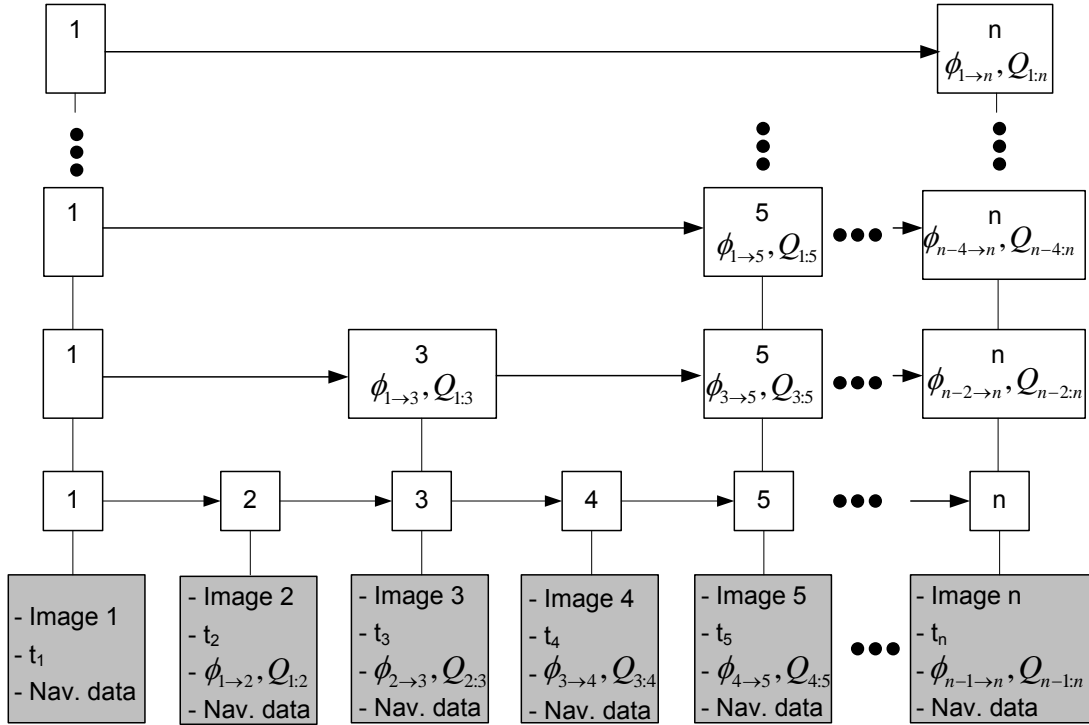


Fig. 7. Skip list repository database example.

C. Incorporating Other Measurements

The proposed graph-based technique for calculating cross-covariance terms may be also applied when, in addition to the multi-vehicle three-view updates, other measurements should be incorporated as well. For instance, each vehicle can apply epipolar-geometry constraints based on images captured by its *own* camera (e. g. [28]). Moreover, some of the vehicles may be equipped with additional sensors, or additional information might be available (e. g. DTM).

For simplicity, we assume at this point a standard measurement model for these additional measurement types, i. e. $\mathbf{z} = H\mathbf{X} + \mathbf{v}$. These measurement updates will be referred to as *basic measurement updates*. Next, it is shown how the basic measurements may be incorporated into the proposed approach for cooperative localization and navigation.

Since a standard measurement model was assumed, the a posteriori estimation error is given by

$$\tilde{\mathbf{X}}^+ = (I - KH)\tilde{\mathbf{X}}^- - K\mathbf{v} \quad (41)$$

Going back to the three-view measurement model, consider the simple scenario shown in Fig. 3. Assume a single basic update was performed between the first update event, at a_3 , and the second update event, at b_3 . Denote by γ the time instant of this additional update, $\gamma \in (a_3, b_3)$. $\tilde{\mathbf{X}}_{b_3}^-$ is no longer inertially propagated from $\tilde{\mathbf{X}}_{a_3}^+$, but instead may be expressed as

$$\tilde{\mathbf{X}}_{b_3}^- = \phi_{\gamma \rightarrow b_3} \tilde{\mathbf{X}}_{\gamma}^+ + \boldsymbol{\omega}_{\gamma:b_3} \quad (42)$$

Based on Eq. (41), $\tilde{\mathbf{X}}_{b_3}^-$ may be expressed as

$$\phi_{\gamma \rightarrow b_3} \left[(I - K_{\gamma} H_{\gamma}) \left(\phi_{a_3 \rightarrow \gamma} \tilde{\mathbf{X}}_{a_3}^+ + \boldsymbol{\omega}_{a_3:\gamma} \right) - K_{\gamma} \mathbf{v} \right] + \boldsymbol{\omega}_{\gamma:b_3}$$

or, alternatively:

$$\tilde{\mathbf{X}}_{b_3}^- = \phi_{a_3 \rightarrow b_3}^* \tilde{\mathbf{X}}_{a_3}^+ + \boldsymbol{\omega}_{a_3:b_3}^* \quad (43)$$

where

$$\phi_{a_3 \rightarrow b_3}^* \doteq \phi_{\gamma \rightarrow b_3} (I - K_{\gamma} H_{\gamma}) \phi_{a_3 \rightarrow \gamma} \quad (44)$$

is the equivalent transition matrix and

$$\boldsymbol{\omega}_{a_3:b_3}^* \doteq \phi_{\gamma \rightarrow b_3} (I - K_{\gamma} H_{\gamma}) \boldsymbol{\omega}_{a_3:\gamma} - \phi_{\gamma \rightarrow b_3} K_{\gamma} \mathbf{v} + \boldsymbol{\omega}_{\gamma:b_3} \quad (45)$$

is the equivalent noise term with noise covariance $Q_{a_3:b_3}^*$ given by

$$Q_{a_3:b_3}^* \doteq \phi_{\gamma \rightarrow b_3} (I - K_{\gamma} H_{\gamma}) Q_{a_3:\gamma} [\phi_{\gamma \rightarrow b_3} (I - K_{\gamma} H_{\gamma})]^T + \phi_{\gamma \rightarrow b_3} K_{\gamma} R K_{\gamma}^T \phi_{\gamma \rightarrow b_3}^T + Q_{\gamma:b_3} \quad (46)$$

Thus, for example, $P_{b_3 b_1}^-$ is given by (cf. Eq. (36)):

$$P_{b_3 b_1}^- = \Phi_{a_3 \rightarrow b_3}^* \left\{ (I - K_{a_3} H_{a_3}) P_{a_3 a_2}^- - K_{a_3} H_{a_2} P_{a_2 a_2}^- - K_{a_3} H_{a_1} P_{a_1 a_2}^- \right\} \Phi_{a_2 \rightarrow b_1}^T$$

In the general case, there might be a number of basic updates in each of the vehicles. However, these updates are treated in a similar manner, by calculating the equivalent transition matrix Φ^* and noise covariance matrix Q^* between the time instances that participate in the three-view measurement and updating accordingly the repository database (cf. Section VI-B).

VII. SIMULATION AND EXPERIMENTAL RESULTS

In this section the proposed approach for vision-aided cooperative navigation is studied in two different scenarios. First, a formation flying scenario is considered, involving two vehicles, a leader and a follower. Statistical results, based on simulated navigation data and synthetic imagery are presented. Next, a holding pattern scenario is demonstrated in an experiment using real imagery and navigation data.

A. Implementation Details

1) *Navigation Simulation:* The navigation simulation for *each* of the two vehicles consists of the following steps [28]: (a) Trajectory generation; (b) velocity and angular velocity increments extraction from the created trajectory; (c) IMU error definition and contamination of pure increments by noise; and (d) strapdown calculations. The strapdown mechanism provides, at each time step, the calculated position, velocity and attitude of the vehicle.

Each vehicle is handled independently based on its own trajectory. Once a vehicle obtains three images with a common overlapping area, the developed algorithm is executed: cross-covariance terms are computed, followed by estimation of the state vector. The estimated state vector is then used for updating the navigation solution and the IMU measurements (cf. Fig. 1). Next, the update information is stored and delivered to the second vehicle in the group.

2) *Image Processing Module*: Given three images with a common overlapping area, the image processing phase includes features extraction from each image using the SIFT algorithm [29] and computation of sets of matching pairs between the first two images, $\{\mathbf{x}_1^i, \mathbf{x}_2^i\}_{i=1}^{N_{12}}$, and between the last two images, $\{\mathbf{x}_2^i, \mathbf{x}_3^i\}_{i=1}^{N_{23}}$, where $\mathbf{x}^i = (x^i, y^i)^T$ are the image coordinates of the i th feature. This computation proceeds as follows. First, the features are matched based on their descriptor vectors (that were computed as part of the SIFT algorithm), yielding the sets $\{\mathbf{x}_1^i, \mathbf{x}_2^i\}_{i=1}^{N_{12}}$, $\{\mathbf{x}_2^i, \mathbf{x}_3^i\}_{i=1}^{\tilde{N}_{23}}$. Since this step occasionally produces false matches (outliers), the RANSAC algorithm [30] is applied over the fundamental matrix [22] model in order to reject the existing false matches, thus obtaining the refined sets $\{\mathbf{x}_1^i, \mathbf{x}_2^i\}_{i=1}^{N_{12}}$ and $\{\mathbf{x}_2^i, \mathbf{x}_3^i\}_{i=1}^{N_{23}}$. The fundamental matrices are not used in further computations.

The next step is to use these two sets for calculating matching triplet features, i. e. matching features in the three given images. This step is performed by matching all $\mathbf{x}_1 \in \{\mathbf{x}_1^i, \mathbf{x}_2^i\}_{i=1}^{N_{12}}$ with all $\mathbf{x}_3 \in \{\mathbf{x}_2^i, \mathbf{x}_3^i\}_{i=1}^{N_{23}}$, yielding a set of matching triplets $\{\mathbf{x}_1^i, \mathbf{x}_2^i, \mathbf{x}_3^i\}_{i=1}^{N_{123}}$. The matching process includes the same steps as described above.

When using synthetic imagery data, a set of points in the real-world is randomly drawn. Then, taking into account the camera motion, known from the true vehicle trajectory, and assuming specific camera calibration parameters, the image coordinates of the observed real-world points are calculated using a pinhole projection [22] at the appropriate time instances. See, for example, Ref. [26] for further details. Consequently, a list of features for each time instant of the three time instances, which are manually specified, is obtained: $\{\mathbf{x}_1^i\}$, $\{\mathbf{x}_2^i\}$ and $\{\mathbf{x}_3^i\}$. The mapping between these three sets is known, since these sets were calculated using the pinhole projection based on the same real-world points. Thus, in order to find the matching sets $\{\mathbf{x}_1^i, \mathbf{x}_2^i, \mathbf{x}_3^i\}_{i=1}^{N_{123}}$, $\{\mathbf{x}_1^i, \mathbf{x}_2^i\}_{i=1}^{N_{12}}$ and $\{\mathbf{x}_2^i, \mathbf{x}_3^i\}_{i=1}^{N_{23}}$ it is only required to check which features are within the camera field of view at the appropriate time instances.

Finally, the calculated sets of matching features are transformed into sets of matching LOS vectors. A LOS vector, expressed in the camera system for some feature $\mathbf{x} = (x, y)^T$, is calculated as $\mathbf{q}^C = (x, y, f)^T$, where f is the camera focal length. As a result, three matching LOS sets are obtained: $\{\mathbf{q}_{1i}^{C_1}, \mathbf{q}_{2i}^{C_2}, \mathbf{q}_{3i}^{C_3}\}_{i=1}^{N_{123}}$, $\{\mathbf{q}_{1i}^{C_1}, \mathbf{q}_{2i}^{C_2}\}_{i=1}^{N_{12}}$ and $\{\mathbf{q}_{2i}^{C_2}, \mathbf{q}_{3i}^{C_3}\}_{i=1}^{N_{23}}$. When handling real imagery, the camera focal length, as well as other camera parameters, are found during the camera calibration process. In addition, a radial distortion correction [22] was applied to camera-captured images, or alternatively, to the extracted feature coordinates.

B. Formation Flying Scenario - Statistical Results

In this section the proposed method for vision-aided cooperative navigation is applied on a formation flying scenario, comprised of a leader vehicle and a single follower vehicle. Each vehicle is equipped with a camera and an IMU. In this scenario, the leader's IMU is of a better quality than the follower's IMU. It is also assumed that the leader's initial navigation errors are small compared to those of the follower. Table II summarizes the assumed 1σ values of the initial navigation errors and IMU bias and drift for the two vehicles.

Both vehicles perform the same trajectory, which is a straight and level north-headed flight at a velocity of 100 m/s. The mean height above ground level is 2000 m. The distance between the leader and follower vehicles is 2000 meters (the follower is behind the leader), i. e. 20 seconds delay. The initial navigation errors, as well as the accelerometers bias and gyroscopes drift, were assumed to be of a zero-mean Gaussian distribution, with standard deviation values specified in Table II. In each simulation run, the drawn values of initial navigation errors were used for calculating the actual initial navigation solution, while the drawn drift and bias were applied to corrupt the ground-truth IMU measurements. The IMU measurements were also contaminated by a zero-mean Gaussian noise with a standard deviation of $100 \mu g/\sqrt{Hz}$ and $0.001 \text{ deg}/\sqrt{hr}$ for the accelerometers and gyroscopes, respectively.

INITIAL NAVIGATION ERRORS AND IMU ERRORS IN THE FORMATION FLYING SCENARIO

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

The synthetic imagery data were obtained by assuming a $20^0 \times 30^0$ field of view, focal length of 1570 pixels, and image noise of 0.5 pixel. The ground landmarks were randomly drawn with a height variation of ± 200 meters relative to the mean ground level.

The follower was updated using the proposed method every 10 seconds, applying the same measurement schedule as in Fig. 3 (Vehicle I in the figure is the leader, Vehicle III is the follower). The first update was carried out after 27 seconds of inertial flight, while the leader vehicle performed an inertial flight the whole time duration. The true translation motion between any three views participating in the same measurement is $T_{12} = 200$ meters and $T_{23} = 400$ meters, in the north direction.

In each update, two of the three images⁶ that participate in the measurement were taken from the leader. Since the two vehicles perform the same trajectory, with a 20 seconds time delay, these two images have been acquired by the leader 20 seconds before the measurement. Therefore they were stored in the leader's repository and retrieved upon request. The cross-covariance terms were calculated in each update according to Eqs. (35)-(37).

The Monte-Carlo results (1000 runs) for the follower vehicle are presented in Figs. 8-10, in terms of the mean navigation error (μ), standard deviation (σ) and square root covariance of the filter. In addition, the results are compared to inertial navigation of the follower and to the navigation errors of the leader vehicle (only standard deviation of these errors is presented; the mean error is zero in all cases). As seen, the position and velocity errors (Fig. 8) are significantly reduced, compared to the inertial scenario, in *all* axes. The bias state is estimated also in *all* axes, while the drift state is only partially estimated (Fig. 10). The updates yielded a mild reduction in Euler angle errors as well (Fig. 9).

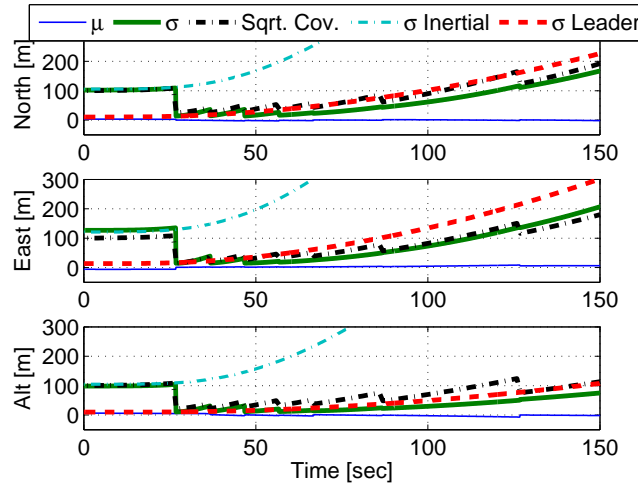
A comparison of the follower navigation errors to the leader navigation errors, reveals further insight. The position errors (and velocity errors to a less extent) of the follower are *lower* than those of the leader, despite the fact that the leader has a considerably better navigation system, and the follower is updated solely based on the leader's navigation data. The reason for this phenomenon is that the measurement \mathbf{z} , given in Eq. (14), is a function of both the follower's and the leader's navigation parameters, while only the follower is actually updated (cf. Section IV). Carrying out the updates on both vehicles, using a similar filter formulation to the formulation discussed in Section IV, will yield an improvement also in the leader's navigation errors [6]. Assuming the measurement schedule given in Fig. 3, it is guaranteed that the graph will remain acyclic even if both of the vehicles are updated each measurement⁷.

The fact that the obtained filter is unbiased and consistent, whereas neglecting the cross-covariance terms renders the filter biased and inconsistent [19], suggests that the calculated values for the cross-covariance terms are close to the true values.

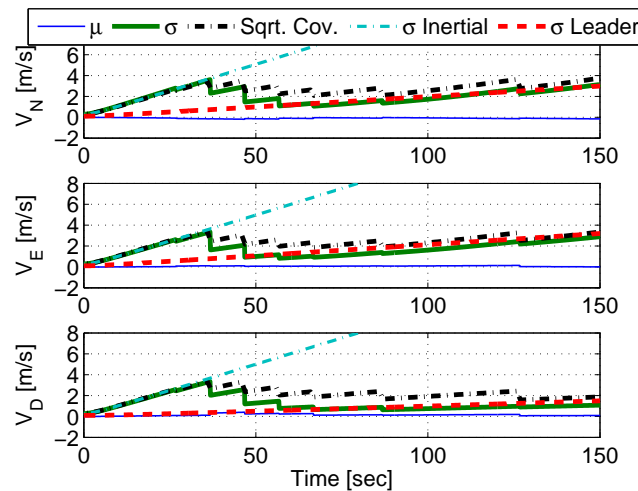
⁶Since in this section a synthetic imagery data is used, the term "image" refers to a synthetic data, e. g. features coordinates.

⁷Provided that the leader is updated only at the time instant of the second image (e. g. referring to Fig. 3, Vehicle I that represents the leader, can be only updated at a_2 and b_2 , and not at a_1 and b_1).

It is also worth mentioning that should the leader perform self-updates based on the available sensors and information (e. g. epipolar constraints, GPS, DTM), improved navigation errors will be obtained not only in the leader but also in the follower navigation system.



(a) Position errors.



(b) Velocity errors.

Fig. 8. Formation flying scenario - Monte Carlo (1000 runs) results; Follower navigation errors compared to inertial navigation and to navigation errors of the Leader: Reduced position and velocity errors in *all* axes. Position and velocity errors are reduced below the Leader level of errors.

C. Holding Pattern Scenario - Experiment Results

In this section the proposed method is demonstrated in an experiment. The experiment setup consists of a single ground vehicle, equipped with a 207MW Axis network camera⁸ and MTi-G Xsens IMU/INS⁹. The vehicle was manually commanded using a joystick, while the camera captured images perpendicular to the motion heading. As in [18], the IMU and imagery data was recored for post-processing at 100 Hz and 15 Hz, respectively. These two sources of data were synchronized.

The vehicle performed two different trajectories. The IMU and the camera were turned off between these two trajectories, thereby allowing to treat each trajectory as if it were performed

⁸http://www.axis.com/products/cam_207mw/index.htm.

⁹<http://www.xsens.com/en/general/mti-g>.

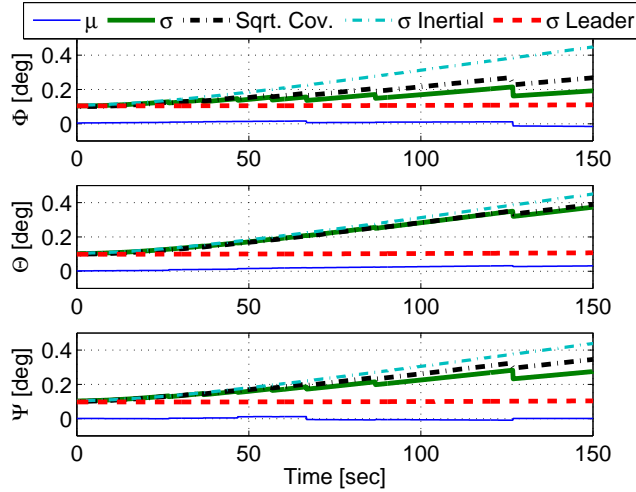


Fig. 9. Formation flying scenario - Monte Carlo (1000 runs) results; Follower navigation errors compared to inertial navigation and to navigation errors of the Leader: Euler angles errors. Euler angles are mildly reduced, however do not reach the Leader's levels due to poor estimation of the drift state (cf. Fig. 10(a)).

by a different vehicle, equipped with a similar hardware (IMU and camera), as opposed to Section VII-B, where one of the vehicles was assumed to be equipped with a better navigation system. Thus, we have two ground vehicles, each performing its own trajectory and recording its own IMU and imagery data.

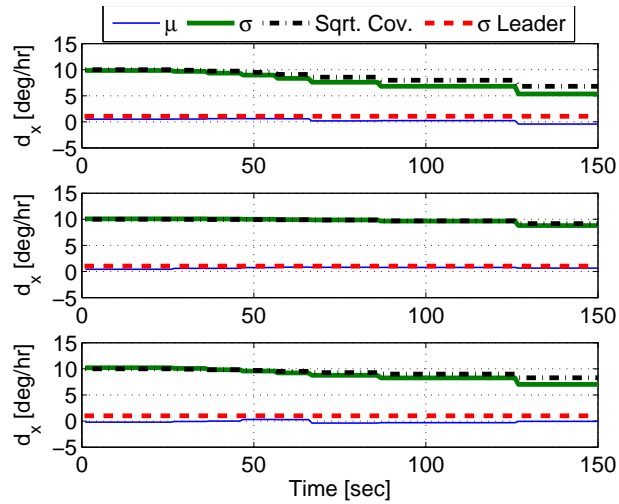
The only available ground-truth data is the manually measured trajectories, since the experiment was carried out indoors and GPS was therefore unavailable. The two trajectories represent a holding pattern scenario. Each vehicle performs the same basic trajectory: Vehicle I performs this basic trajectory twice, while Vehicle II performs the basic trajectory once, starting from a different point along the trajectory, and reaching the starting point of Vehicle I after about 26 seconds. The reference trajectories of Vehicle I and II are shown in Fig. 11. The diamond and square marks denote the manual measurements of the vehicles position. Each two adjacent marks of the same vehicle are connected using a linear interpolation.

The proposed method for multi-vehicle (MV) three-view based updates was applied several times in the experiment. In addition, the method was executed in a self-update mode, in which all the images are captured by the same vehicle. The cross-covariance terms in this case were computed exactly as in the case of multi-vehicle updates. A schematic sketch of the measurements schedule is given in Fig. 12. Table III provides further information, including the time instances of each participating triplet of images in the applied measurements.

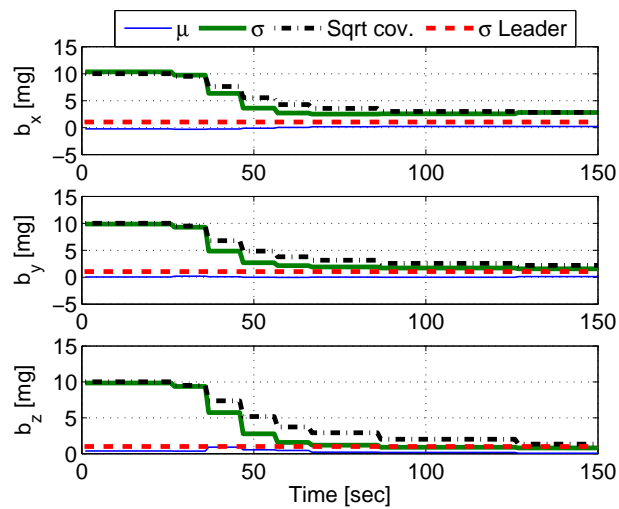
As seen, Vehicle I is updated twice using data obtained from Vehicle II (measurements c and e), and four times based on its own images (measurements f, g, h and i). Vehicle II is updated three times utilizing the information received from Vehicle I (measurements a, b and d). The vehicles performed inertial navigation elsewhere, by processing the recorded IMU data.

The images participating in each three-view update were manually identified and chosen. Fig. 13 shows, for example, the three images of measurement a: images 13(a) and 13(b) were captured by Vehicle I, while image 13(c) was captured by Vehicle II. Features that were found common to all the three images (triplets) are also shown in the figure. Note that two objects (a bottle, and a bag) that appear in images 13(a) and 13(b) are missing in image 13(c). These two objects were removed between the two trajectories. Therefore, as seen in Fig. 13, these two objects are not represented by matched triplets of features (but can be represented by matched pairs of features between the first two views).

The experiment results are given in Figs. 14-15: Figs. 14(a) and 14(b) show the position



(a) Drift estimation errors.



(b) Bias estimation errors.

Fig. 10. Formation flying scenario - Monte Carlo (1000 runs) results: Poor drift estimation. Bias estimation to the Leader's bias levels (1 mg).

errors for Vehicle I and II, while Figs. 15(a) and 15(b) show the velocity errors. Each figure consists of three curves: navigation error, square root covariance of the filter, and navigation error in an inertial scenario (given for reference). The measurement type (MV-update or self-update) is also denoted in the appropriate locations.

The position error was calculated by subtracting the navigation solution from the true trajectories (cf. Fig. 11). In a similar manner, the velocity error was computed by subtracting the navigation solution from the true velocity profiles. However, since velocity was not measured in the experiment, it was only possible to obtain an approximation thereof. The approximated velocity was calculated assuming that the vehicles moved with a constant velocity in each phase¹⁰.

As seen from Fig. 14(a), the position error of Vehicle I was nearly nullified in all axes as the result of the first update, which was of MV type. The next update (also MV) caused to another reduction in the north position error. After completing a loop in the trajectory, it became possible to apply the three-view updates in a self-update mode for Vehicle I, i. e.

¹⁰The phase duration and the translation that each vehicle has undergone in each phase are known from analyzing the IMU measurements and from the true trajectories.

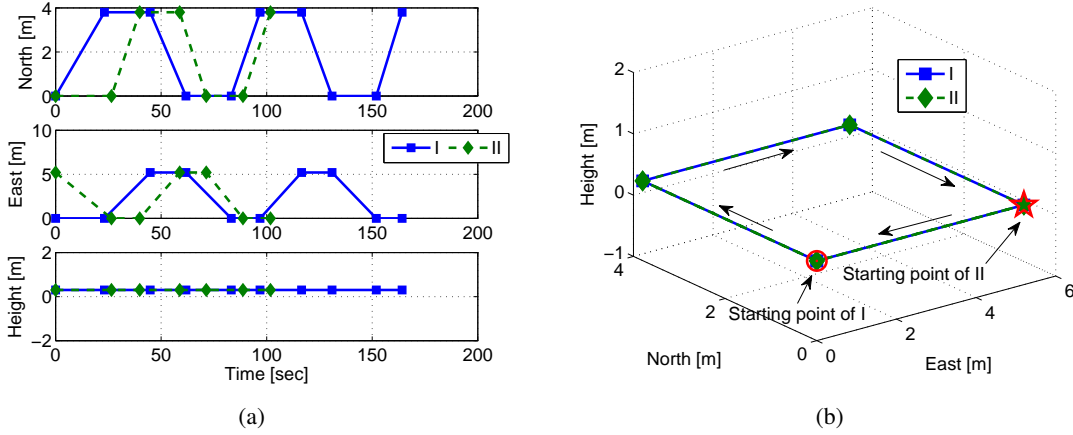


Fig. 11. Trajectories of Vehicles I and II in the experiment. Diamond and square marks indicate manually-measured vehicle locations. Circle and star marks in (b) denote the starting point of each vehicle.

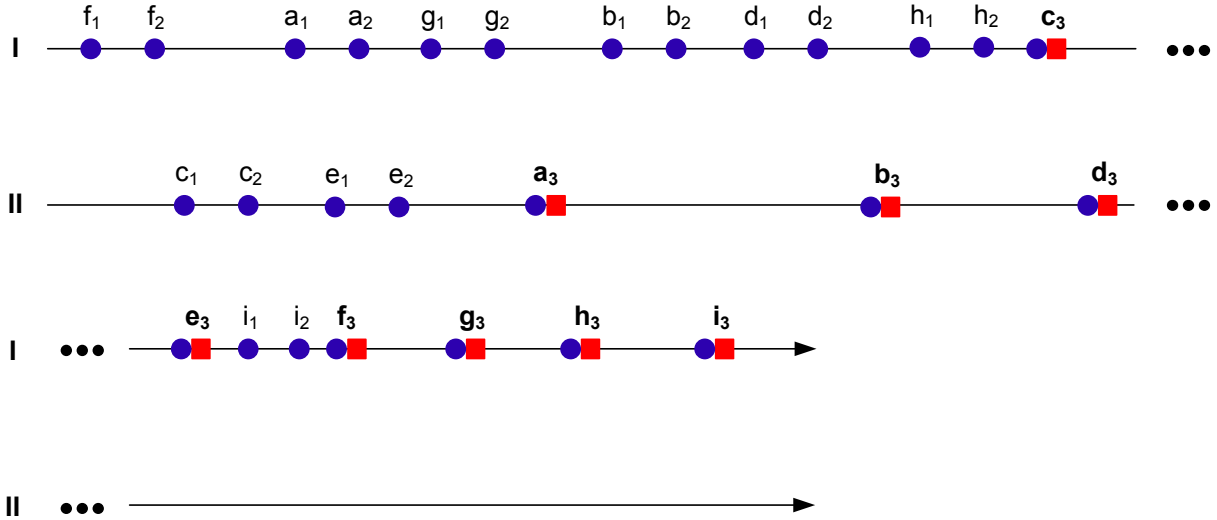


Fig. 12. Schematic sketch of the measurement schedule in the experiment. Further information regarding each measurement is given in Table III.

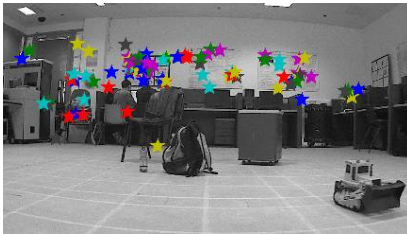
all the three images were captured by Vehicle I. Overall, due to the applied 6 three-view updates, the position error of Vehicle I has been confined to around 50 meters in the north and east directions, and 10 meters in altitude. As a comparison, the position error of Vehicle I in an inertial scenario reaches 900, 200 and 50 meters in the north, east and down directions, respectively, after 150 seconds of operation. The position error of Vehicle II (cf. Fig. 14(b)) is also dramatically reduced as the result of the three-view multi-vehicle updates. For example, after the third update ($t \approx 60$ seconds), the position error was nearly nullified in north direction and reduced from 50 to 20 meters in the east direction. The velocity errors are also considerably reduced in all axes (cf. Fig. 15).

VIII. CONCLUSIONS

This paper presented a new method for distributed vision-aided cooperative navigation based on three-view geometry constraints. Whereas traditional approaches for cooperative localization and navigation utilize relative pose measurements, in the newly-proposed method a measurement is formulated whenever the same general scene is observed by different vehicles.

TABLE III
MEASUREMENT DETAILS IN THE EXPERIMENT.

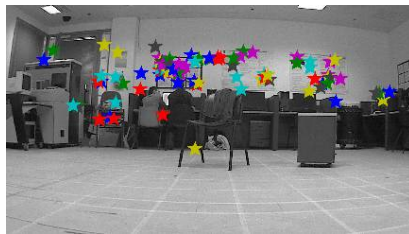
Notation	Type	Querying vehicle	t_3 [sec]	Replying vehicle	t_1, t_2 [sec]
a	MV update	II	32.6	I	8.4, 14.2
b	MV update	II	53.2	I	35.9, 39.1
c	MV update	I	60.0	II	2.3, 5.6
d	MV update	II	60.6	I	47.9, 49.2
e	MV update	I	66.8	II	10.3, 12.1
f	Self update	I	81.1	I	0.3, 1.3
g	Self update	I	97.0	I	22.8, 24.3
h	Self update	I	124.7	I	54.3, 55.6
i	Self update	I	142.0	I	70.8, 72.1



(a)



(b)

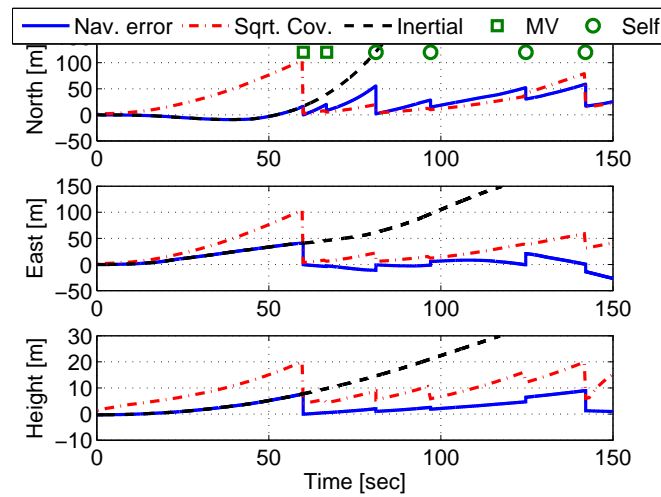


(c)

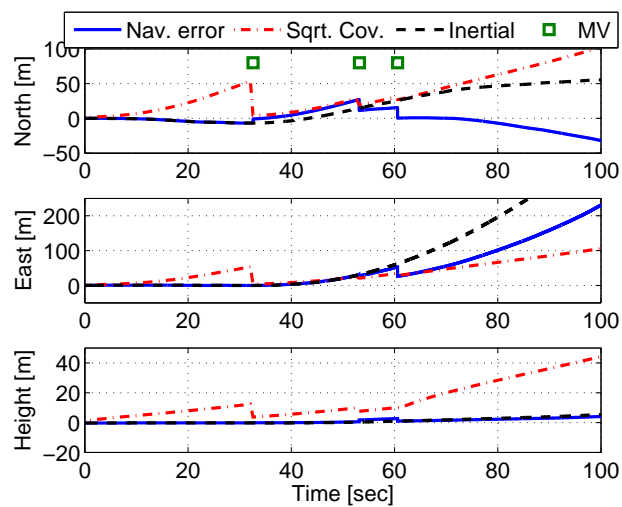
Fig. 13. Images participating in measurement a and matched triplets of features. Images (a) and (b) were captured by Vehicle I; Image (c) was captured by Vehicle II. The images (a),(b) and (c) are represented in Fig. 12 as a_1, a_2 and a_3 .

Three images of a common region are required for each measurement. These images are not necessarily captured at the same time. All, or some, of the vehicles maintain a local repository of captured images, that are associated with some navigation parameters. In a typical scenario, a vehicle captures an image and broadcasts it, along with its current navigation solution, to other vehicles in the group, inquiring if they have previously captured images containing the same region. Upon receiving such a query, each vehicle performs a check in its repository looking for appropriate images. Among these images, only images with a smaller navigation uncertainty compared to the uncertainty in the navigation data of the query image, are transmitted back.

The proposed method was studied in a simulated environment and in an experiment. Statistical results based on simulated navigation and synthetic imagery data were presented for a leader-follower scenario, in which the leader is equipped with a higher quality INS compared to the follower's INS. The developed method reduced the rapidly-developing navigation errors



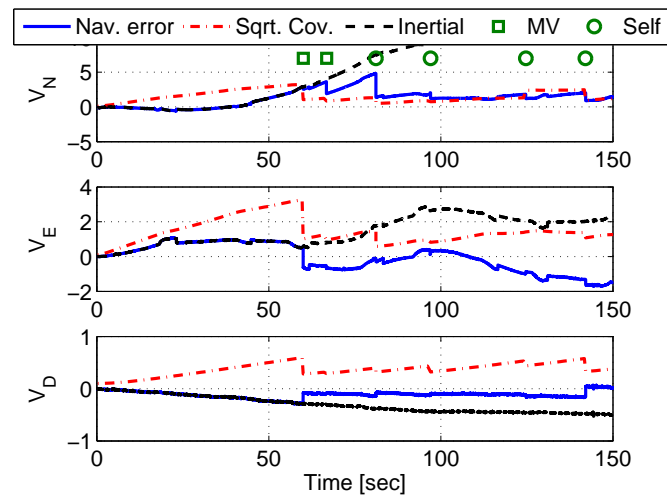
(a) Position errors - vehicle I



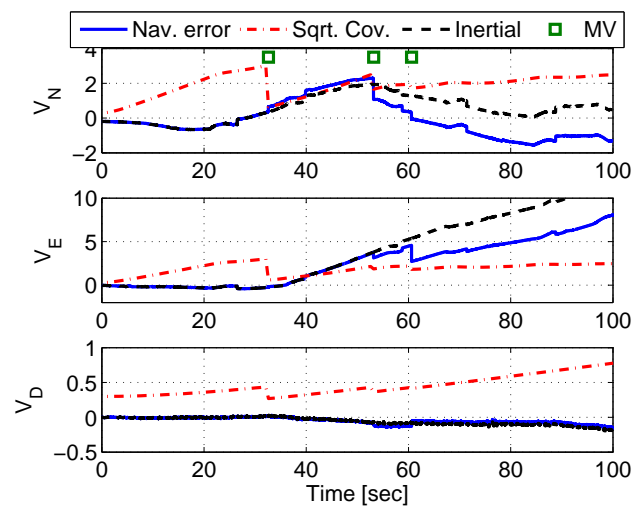
(b) Position errors - vehicle II

Fig. 14. Position errors of Vehicles I and II in the experiment.

of the follower to the level of errors of the leader. A holding pattern scenario was demonstrated in an experiment involving two ground vehicles equipped with identical inertial measurement units and cameras. Significant reduction in the navigation errors of both vehicles was obtained as a result of activating the newly-developed method.



(a) Velocity errors - vehicle I



(b) Velocity errors - vehicle II

Fig. 15. Velocity errors of Vehicles I and II in the experiment.

APPENDIX A

The transition matrix $\Phi_{t_a \rightarrow t_b}$ satisfying Eq. (9) is calculated according to

$$\Phi_{t_a \rightarrow t_b} = e^{\Phi_c \Delta t} \quad (47)$$

where $\Delta t \doteq t_b - t_a$ and Φ_c is the following continuous system matrix

$$\Phi_c = \begin{bmatrix} 0_{3 \times 3} & I_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & A_s & 0_{3 \times 3} & C_L^B \\ 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & -C_L^B & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} \end{bmatrix} \quad (48)$$

In the above equation, the matrix C_L^B is a directional cosine matrix transforming from body system to LLLN system and A_s is a skew-symmetric matrix of the specific force vector $\mathbf{f} = (f_x \ f_y \ f_z)^T$, measured by the accelerometers and expressed in the NED system:

$$A_s = \begin{bmatrix} 0 & -f_D & f_E \\ f_D & 0 & -f_N \\ -f_E & f_N & 0 \end{bmatrix} \quad \begin{bmatrix} f_N \\ f_E \\ f_D \end{bmatrix} = C_L^B \begin{bmatrix} f_x \\ f_y \\ f_z \end{bmatrix} \quad (49)$$

The expression for the continuous system matrix Φ_c , given in Eq. (48), is valid for short periods of operation, significantly smaller than the Schuler period (around 84 minutes) [32]. While the scenario examples considered in this paper indeed satisfy this condition, one could adopt less degenerated process models (for medium-term and long-term scenarios [32]) as well. It is worth noting that a similar model of the system matrix is widely used also in the SLAM community (e. g. [33]).

APPENDIX B

In this appendix we present explicit expressions for the Jacobian matrices

$$H_i(t_i) = \nabla_{\zeta_i(t_i)} \mathbf{h} \quad , \quad D_i(t_i) = \nabla_{\mathbf{y}_i(t_i)} \mathbf{h} \quad , \quad i = 1, 2, 3 \quad (50)$$

The reader is referred to [18] and [19] for further details.

Expressions for $H_i(t_i)$

The matrices $H_i(t_i)$, $i = 1, 2, 3$ are of the following form:

$$H_i = [H^{\text{Pos}(t_i)} \quad 0 \quad H^{\Psi(t_i)} \quad 0 \quad 0] \quad (51)$$

with

$$H^{\text{Pos}(t_3)} = \mathcal{A} \quad (52)$$

$$H^{\text{Pos}(t_2)} = -(\mathcal{A} + \mathcal{B}) \quad (53)$$

$$H^{\text{Pos}(t_1)} = \mathcal{B} \quad (54)$$

and

$$H^{\Psi(t_3)} = \sum_{i=1}^{N_{123}} \mathbf{e}_i \left[\mathbf{T}_{12}^T \frac{\partial \mathbf{w}_i}{\partial \mathbf{q}_{3_i}} - \mathbf{T}_{23}^T \frac{\partial \mathbf{u}_i}{\partial \mathbf{q}_{3_i}} \right] C_{L_2}^{L_3} [\mathbf{q}_{3_i}^{L_3}]_{\times} - \sum_{i=1}^{N_{23}} \mathbf{e}_{N_{123}+i} \mathbf{T}_{23}^T \frac{\partial \mathbf{f}_i}{\partial \mathbf{q}_{3_i}} C_{L_2}^{L_3} [\mathbf{q}_{3_i}^{L_3}]_{\times} \quad (55)$$

$$\begin{aligned}
H^{\Psi(t_2)} &= \sum_{i=1}^{N_{123}} \mathbf{e}_i \left[\mathbf{T}_{12}^T \frac{\partial \mathbf{w}_i}{\partial \mathbf{q}_{2_i}} - \mathbf{T}_{23}^T \frac{\partial \mathbf{u}_i}{\partial \mathbf{q}_{2_i}} \right] [\mathbf{q}_{2_i}^{L_2}]_{\times} - \sum_{i=1}^{N_{23}} \mathbf{e}_{N_{123}+i} \mathbf{T}_{23}^T \frac{\partial \mathbf{f}_i}{\partial \mathbf{q}_{2_i}} [\mathbf{q}_{2_i}^{L_2}]_{\times} + \\
&+ \sum_{i=1}^{N_{12}} \mathbf{e}_{N_{123}+N_{23}+i} \mathbf{T}_{12}^T \frac{\partial \mathbf{g}_i}{\partial \mathbf{q}_{2_i}} [\mathbf{q}_{2_i}^{L_2}]_{\times}
\end{aligned} \tag{56}$$

$$H^{\Psi(t_1)} = \sum_{i=1}^{N_{123}} \mathbf{e}_i \left[\mathbf{T}_{12}^T \frac{\partial \mathbf{w}_i}{\partial \mathbf{q}_{1_i}} - \mathbf{T}_{23}^T \frac{\partial \mathbf{u}_i}{\partial \mathbf{q}_{1_i}} \right] C_{L_2}^{L_1} [\mathbf{q}_{1_i}^{L_1}]_{\times} + \sum_{i=1}^{N_{12}} \mathbf{e}_{N_{123}+N_{23}+i} \mathbf{T}_{12}^T \frac{\partial \mathbf{g}_i}{\partial \mathbf{q}_{1_i}} C_{L_2}^{L_1} [\mathbf{q}_{1_i}^{L_1}]_{\times} \tag{57}$$

In the above expressions, \mathbf{e}_j is a $N \times 1$ vector that is comprised of zero entries except for the j th element which is equal to one.

Expressions for D and R

Instead of calculating $D_i(t_i)$, we present expressions for $E[D\mathbf{v}(D\mathbf{v})^T] \doteq DRD$. Recall that

$$D \doteq \nabla_{\{\mathbf{q}_{1_i}^{C_1}, \mathbf{q}_{2_i}^{C_2}, \mathbf{q}_{3_i}^{C_3}\}} \mathbf{h} \tag{58}$$

$$R \doteq \text{cov}(\{\mathbf{q}_{1_i}^{C_1}, \mathbf{q}_{2_i}^{C_2}, \mathbf{q}_{3_i}^{C_3}\}) \tag{59}$$

Let $R_{\mathbf{v}}$ be a 3×3 covariance matrix of the image noise

$$R_{\mathbf{v}} = \begin{bmatrix} R_x & 0 & 0 \\ 0 & R_y & 0 \\ 0 & 0 & R_f \end{bmatrix} \tag{60}$$

with $R_x = E(v_x v_x^T)$ and $R_y = E(v_y v_y^T)$. R_f represents the uncertainty in the camera focal length. Assuming the focal length is known precisely, it can be chosen as zero.

Let ΔN_{12} and ΔN_{23} be the number of additional pairs in $\{\mathbf{q}_{1_i}, \mathbf{q}_{2_i}\}_{i=1}^{N_{12}}$ and $\{\mathbf{q}_{2_i}, \mathbf{q}_{3_i}\}_{i=1}^{N_{23}}$ that are not present in $\{\mathbf{q}_{1_i}, \mathbf{q}_{2_i}, \mathbf{q}_{3_i}\}_{i=1}^{N_{123}}$: $N_{12} = N_{123} + \Delta N_{12}$ and $N_{23} = N_{123} + \Delta N_{23}$. Although the overall number of matches in the sets

$$\{\mathbf{q}_{1_i}, \mathbf{q}_{2_i}\}_{i=1}^{N_{12}}, \quad \{\mathbf{q}_{2_i}, \mathbf{q}_{3_i}\}_{i=1}^{N_{23}}, \quad \{\mathbf{q}_{1_i}, \mathbf{q}_{2_i}, \mathbf{q}_{3_i}\}_{i=1}^{N_{123}} \tag{61}$$

is $N = N_{123} + N_{12} + N_{23}$, the actual number of *different* matches is $N_{123} + \Delta N_{12} + \Delta N_{23}$.

The matrix DRD is given by:

$$\begin{aligned}
DRD^T &= \sum_{i=1}^{N_{123}+\Delta N_{12}} \frac{\partial \mathbf{h}}{\partial \mathbf{q}_{1_i}^{C_1}} R_{\mathbf{v}} \frac{\partial \mathbf{h}^T}{\partial \mathbf{q}_{1_i}^{C_1}} + \sum_{i=1}^{N_{123}+\Delta N_{12}+\Delta N_{23}} \frac{\partial \mathbf{h}}{\partial \mathbf{q}_{2_i}^{C_2}} R_{\mathbf{v}} \frac{\partial \mathbf{h}^T}{\partial \mathbf{q}_{2_i}^{C_2}} + \\
&+ \sum_{i=1}^{N_{123}+\Delta N_{23}} \frac{\partial \mathbf{h}}{\partial \mathbf{q}_{3_i}^{C_3}} R_{\mathbf{v}} \frac{\partial \mathbf{h}^T}{\partial \mathbf{q}_{3_i}^{C_3}}
\end{aligned} \tag{62}$$

where

$$\begin{aligned} \frac{\partial \mathbf{h}}{\partial \mathbf{q}_{1_i}^{C_1}} \Big|_{N \times 3} &= \left[\frac{\partial \mathcal{A} \mathbf{T}_{23}}{\partial \mathbf{u}_i} \frac{\partial \mathbf{u}_i}{\partial \mathbf{q}_{1_i}^{L_2}} - \frac{\partial \mathcal{B} \mathbf{T}_{12}}{\partial \mathbf{w}_i} \frac{\partial \mathbf{w}_i}{\partial \mathbf{q}_{1_i}^{L_2}} - \frac{\partial \mathcal{B} \mathbf{T}_{12}}{\partial \mathbf{g}_i} \frac{\partial \mathbf{g}_i}{\partial \mathbf{q}_{1_i}^{L_2}} \right] \frac{\partial \mathbf{q}_{1_i}^{L_2}}{\partial \mathbf{q}_{1_i}^{C_1}} = \\ &= \begin{cases} \left\{ \mathbf{e}_i \left[\mathbf{T}_{23}^T \frac{\partial \mathbf{u}_i}{\partial \mathbf{q}_{1_i}^{L_2}} - \mathbf{T}_{12}^T \frac{\partial \mathbf{w}_i}{\partial \mathbf{q}_{1_i}^{L_2}} \right] - \mathbf{e}_{N_{123}+N_{23}+i} \mathbf{T}_{12}^T \frac{\partial \mathbf{g}_i}{\partial \mathbf{q}_{1_i}^{L_2}} \right\} C_{L_2}^{C_1} & i \leq N_{123} \\ -\mathbf{e}_{N_{123}+N_{23}+i} \mathbf{T}_{12}^T \frac{\partial \mathbf{g}_i}{\partial \mathbf{q}_{1_i}^{L_2}} C_{L_2}^{C_1} & N_{123} < i \leq \check{N}_{12} \end{cases} \quad (63) \end{aligned}$$

$$\begin{aligned} \frac{\partial \mathbf{h}}{\partial \mathbf{q}_{2_i}^{C_2}} \Big|_{N \times 3} &= \begin{cases} \mathbf{e}_i \left[\mathbf{T}_{23}^T \frac{\partial \mathbf{u}_i}{\partial \mathbf{q}_{2_i}^{L_2}} - \mathbf{T}_{12}^T \frac{\partial \mathbf{w}_i}{\partial \mathbf{q}_{2_i}^{L_2}} \right] C_{L_2}^{C_2} + \\ + \left[\mathbf{e}_{N_{123}+i} \mathbf{T}_{23}^T \frac{\partial \mathbf{f}_i}{\partial \mathbf{q}_{2_i}^{L_2}} - \mathbf{e}_{N_{123}+N_{23}+i} \mathbf{T}_{12}^T \frac{\partial \mathbf{g}_i}{\partial \mathbf{q}_{2_i}^{L_2}} \right] C_{L_2}^{C_2} & i \leq N_{123} \\ \mathbf{e}_{N_{123}+i} \mathbf{T}_{23}^T \frac{\partial \mathbf{f}_i}{\partial \mathbf{q}_{2_i}^{L_2}} C_{L_2}^{C_2} & N_{123} < i \leq \check{N}_{23} \\ -\mathbf{e}_{2N_{123}+i} \mathbf{T}_{12}^T \frac{\partial \mathbf{g}_i}{\partial \mathbf{q}_{2_i}^{L_2}} C_{L_2}^{C_2} & \check{N}_{23} < i \leq \check{N}_{23} + \Delta N_{12} \end{cases} \quad (64) \end{aligned}$$

$$\begin{aligned} \frac{\partial \mathbf{h}}{\partial \mathbf{q}_{3_i}^{C_3}} \Big|_{N \times 3} &= \begin{cases} \left\{ \mathbf{e}_i \left[\mathbf{T}_{23}^T \frac{\partial \mathbf{u}_i}{\partial \mathbf{q}_{3_i}^{L_2}} - \mathbf{T}_{12}^T \frac{\partial \mathbf{w}_i}{\partial \mathbf{q}_{3_i}^{L_2}} \right] + \mathbf{e}_{N_{123}+i} \mathbf{T}_{23}^T \frac{\partial \mathbf{f}_i}{\partial \mathbf{q}_{3_i}^{L_2}} \right\} C_{L_2}^{C_3} & i \leq N_{123} \\ \mathbf{e}_{N_{123}+i} \mathbf{T}_{23}^T \frac{\partial \mathbf{f}_i}{\partial \mathbf{q}_{3_i}^{L_2}} C_{L_2}^{C_3} & N_{123} < i \leq \check{N}_{23} \end{cases} \quad (65) \end{aligned}$$

with $\check{N}_{12} \doteq N_{123} + \Delta N_{12}$ and $\check{N}_{23} \doteq N_{123} + \Delta N_{23}$.

REFERENCES

- [1] Madhavan, R., Fregene, K., and Parker, L. E., "Distributed Cooperative Outdoor Multirobot Localization and Mapping," *Autonomous Robots*, Vol. 17, 2004, pp. 23–29.
- [2] Nettletona, E., Thrun, S., Durrant-Whyte, H., and Sukkarieh, S., "Decentralised SLAM with Low-Bandwidth Communication for Teams of Vehicles," *Proceedings of the International Conference on Field and Service Robotics*, Lake Yamanaka, Japan, 2003.
- [3] Kim, B., Kaess, M., Fletcher, L., Leonard, J., Bachrach, A., Roy, N., and Teller, S., "Multiple Relative Pose Graphs for Robust Cooperative Mapping," *Proceedings of the IEEE International Conference on Robotics and Automation*, Anchorage, Alaska, May 2010.
- [4] Lazaro, M. T. and Castellanos, J. A., "Localization of Probabilistic Robot Formations in SLAM," *Proceedings of the IEEE International Conference on Robotics and Automation*, Anchorage, Alaska, May 2010.
- [5] Smali, C., Najjar, M. E. E., and Charpillat, F., "Multi-sensor Fusion Method Using Bayesian Network for Precise Multi-vehicle Localization," *Proceedings of the IEEE International Conference on Intelligent Transportation Systems*, Beijing, China, 2008, pp. 906–911.
- [6] Roumeliotis, S. I. and Bekey, G. A., "Distributed Multirobot Localization," *IEEE Transactions on Robotics and Automation*, Vol. 18, No. 5, 2002, pp. 781–795.
- [7] Kurazume, R., Nagata, S., and Hirose, S., "Cooperative Positioning with Multiple Robots," *Proceedings of the IEEE International Conference on Robotics and Automation*, San Diego, CA, May 1994, pp. 1250–1257.
- [8] Martinelli, A., Pont, F., and Siegwart, R., "Multi-Robot Localization Using Relative Observations," *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, Barcelona, Spain, 2005, pp. 2797–2802.
- [9] Caglioti, V., Citterio, A., and Fossati, A., "Cooperative, Distributed Localization in Multi-robot Systems: a Minimum-entropy Approach," *Proceedings of the IEEE Workshop on Distributed Intelligent Systems*, 2006, pp. 25–30.
- [10] Knuth, J. and Barooh, P., "Distributed collaborative localization of multiple vehicles from relative pose measurements," *Forty-Seventh Annual Allerton Conference*, Illinois, USA, 2009, pp. 314–321.
- [11] Nerurkar, E. D., Roumeliotis, S. I., and Martinelli, A., "Distributed Maximum A Posteriori Estimation for Multi-robot Cooperative Localization," *Proceedings of the IEEE International Conference on Robotics and Automation*, Kobe, Japan, 2009, pp. 1402–1409.
- [12] Nerurkar, E. D. and Roumeliotis, S. I., "Multi-Centralized Cooperative Localization under Asynchronous Communication," *Department of Computer Science and Engineering, University of Minnesota, Technical Report*, March 2010.
- [13] Howard, A., Mataric, M. J., and Sukhatme, G. S., "Putting the 'I' in 'Team' - an Ego-Centric Approach to Cooperative Localization," *Proceedings of the IEEE International Conference on Robotics and Automation*, Taipei, Taiwan, 2003, pp. 868–874.
- [14] Karam, N., Chausse, F., Aufrere, R., and Chapuis, R., "Localization of a Group of Communicating Vehicles by State Exchange," *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, Beijing, China, 2006, pp. 519–524.

- [15] Sharma, R. and Taylor, C., "Cooperative Navigation of MAVs In GPS Denied Areas," *Proceedings of the IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, Seoul, Korea, 2008, pp. 481–486.
- [16] Mariottini, G. L., Pappas, G., Prattichizzo, D., and Daniilidis, K., "Vision-based Localization of Leader-Follower Formations," *Proceedings of the IEEE International Conference on Decision and Control*, Seville, Spain, 2005, pp. 635–640.
- [17] Montesano, L., Gaspar, J., Santos-Victor, J., and Montano, L., "Fusing vision-based bearing measurements and motion to localize pairs of robots," *Proceedings of the ICRA Workshop on Cooperative Robotics*, Barcelona, Spain, 2005.
- [18] Indelman, V., Gurfil, P., Rivlin, E., and Rotstein, H., "Real-Time Vision-Aided Localization and Navigation Based on Three-View Geometry," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 48, No. 2, April 2012, to appear.
- [19] Indelman, V., "Navigation Performance Enhancement Using Online Mosaicking," *PhD Thesis*, Technion, Israel, 2011.
- [20] Merino, L., Wiklund, J., Caballero, F., Moe, A., Ramiro, J., Forsen, E., Nordberg, K., and Ollero, A., "Vision-Based Multi-UAV Position Estimation," *IEEE Robotics and Automation Magazine*, September 2006, pp. 53–62.
- [21] Indelman, V., Gurfil, P., Rivlin, E., and Rotstein, H., "Graph-based Distributed Cooperative Navigation," *Proceedings of the IEEE International Conference on Robotics and Automation*, Shanghai, China, May 2011.
- [22] Hartley, R. and Zisserman, A., *Multiple View Geometry*, Cambridge University Press, 2000.
- [23] Bahr, A., Walter, M. R., and Leonard, J. J., "Consistent Cooperative Localization," *Proceedings of the IEEE International Conference on Robotics and Automation*, Kobe, Japan, May 2009, pp. 3415–3422.
- [24] Indelman, V., Gurfil, P., Rivlin, E., and Rotstein, H., "Distributed Vision-Aided Cooperative Localization and Navigation based on Three-View Geometry," *Proceedings of the IEEE Aerospace Conference*, Montata, USA, March 2011.
- [25] Soatto, S., Frezza, R., and Perona, P., "Motion Estimation via Dynamic Vision," *IEEE Transactions on Automatic Control*, Vol. 41, No. 3, March 1996, pp. 393–413.
- [26] Gurfil, P. and Rotstein, H., "Partial Aircraft State Estimation from Visual Motion Using the Subspace Constraints Approach," *Journal of Guidance, Control and Dynamics*, Vol. 24, No. 5, July 2001, pp. 1016–1028.
- [27] Pugh, W., "Skip Lists: A Probabilistic Alternative to Balanced Trees," *Communications of the ACM*, Vol. 33, No. 6, 1990, pp. 668–676.
- [28] Indelman, V., Gurfil, P., Rivlin, E., and Rotstein, H., "Navigation Aiding Based on Coupled Online Mosaicking and Camera Scanning," *Journal of Guidance, Control and Dynamics*, Vol. 33, No. 6, 2010, pp. 1866–1882.
- [29] Lowe, D., "Distinctive Image Features from Scale-Invariant Keypoints," *International Journal of Computer Vision*, Vol. 60, No. 2, November 2004, pp. 91–110.
- [30] Fischler, M. and Bolles, R., "Random sample consensus: a paradigm for model fitting with application to image analysis and automated cartography," *Commun. Assoc. Comp. Mach.*, Vol. 24, 1981, pp. 381–395.
- [31] Farrel, J. A. and Barth, M., *The Global Positioning System and Inertial Navigation*, McGraw-Hill, 1998.
- [32] Titterton, D. H. and Weston, J. L., *Strapdown Inertial Navigation Technology*, AIAA, 2004.
- [33] Bryson, M. and Sukkarieth, S., "Observability Analysis and Active Control for Airborne SLAM," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 44, No. 1, January 2008, pp. 261–280.
- [34] Mourikis, A. and Roumeliotis, I., "A Multi-State Constraint Kalman Filter for Vision-aided Inertial Navigation" *Proceedings of the IEEE International Conference on Robotics and Automation*, Roma, Italy, April 2007.