Solution Separation-based Integrity Monitor for Integrated GNSS/IMU/Camera Navigation: Constraining the Hypothesis Space With Deep Learning

1st Vasileios Bosdelekidis
Department of Engineering Cybernetics
Norwegian University of Science and Technology
Trondheim, Norway
vasileios.bosdelekidis@ntnu.no

2nd Tor A. Johansen
Department of Engineering Cybernetics
Norwegian University of Science and Technology
Trondheim, Norway
tor.arne.johansen@ntnu.no

3rd Nadezda Sokolova
SINTEF
and
Department of Engineering Cybernetics
Norwegian University of Science and Technology
Trondheim, Norway
nadia.sokolova@sintef.no

4th Torleiv H. Bryne
Department of Engineering Cybernetics
Norwegian University of Science and Technology
Trondheim, Norway
torleiv.h.bryne@ntnu.no

***Abstract*—This article introduces a method utilizing deep learning in typical Multiple Hypothesis Solution Separation (MHSS)-based Integrity Monitors (IMs) of autonomous vehicle navigation systems, when conventional sensors, such as GNSS and Inertial Measurement Unit (IMU), are integrated with a camera. It is an innovative methodology to reduce the hypothesis space of sensor faults. In the proposed method, the measurement subsets to evaluate in MHSS are generated from the IMU/GNSS measurement set only, so that Fault Detection and Exclusion (FDE) in camera measurements takes place separately. In the investigated approach, anomaly prediction in state estimate error due to camera faults is performed based on raw images with a Deep Neural Network (DNN), and IM input is required only during the online refinement of predicted anomaly locations to reflect anomalies in the IM test statistic. This opens for the possibility to evaluate environment features and conditions that cause specific detected or undetected sensor faults. Experiments on the IMU/GNSS/Camera integration demonstrated that Protection Level (PL) bounding performance of the proposed IM, with limited hypothesis space and the individual camera FDE, is comparable to the MHSS IM informed by the full set of fault hypotheses. Despite the use case of the camera, the method can be directly extended to integrations with multiple auxiliary sensors, where each auxiliary sensor is evaluated individually for faults.**

***Index Terms*—Integrity Monitor, Solution Separation, Deep Learning, Camera Navigation, Visual SLAM**

I. Introduction

Multi-sensor navigation systems that combine traditional GNSS/IMU integration with auxiliary sensors, like cameras, are prominent today. However, the development of IMs has mostly been limited to specific navigation algorithms or sensors [1], whereas MHSS has been used in a few cases for sensor-agnostic [2], [3], or even for navigation algorithm-agnostic IM [4]. Such approaches rely on the monitored navigation algorithm itself for estimation of consistent position error covariance due to each sensor in the fault-free case. However, covariances are often not readily available, and someone would need to undergo complex internal modifications in the navigation algorithm. Furthermore, the extension of traditional MHSS-based IMs to the examined multi-sensor systems might be unrealizable due to a large and intractable number of fault sources, resulting in many fault-tolerant hypotheses to evaluate, or in the development of IMs that are undertested under challenging environmental conditions. In fact, the reliability of Visual Navigation (VN) algorithms in non-ideal situations is a topic which is researched very sparsely. One work from Bednář, Petrlík, Vivaldini and Saska [5] concluded that different algorithms perform very differently in real environments depending on the orientation of the camera, the presence of sunlight and the type of vehicle motion. Yet, an extensive set of conditions that characterize camera performance is still not available.

A possible solution to overcome the above mentioned limitations is to limit the hypothesis space of assumed fault-free sensor measurement subsets in MHSS, by evaluating the navigation faults that each auxiliary sensor causes, independently from other sensors. We refer to a navigation fault or anomaly as the event where the true error of a navigation

The work is funded by the Research Council of Norway and Centre for Autonomous Marine Operations and Systems (RCN grant number: 305051).

solution is not drawn from the typical distribution of errors. We denote as IM anomaly the event that the true error of a navigation solution exceeds a calculated PL. IM anomalies are usually happening due to violation of assumptions during PL computation, e.g. the presence of non-linearities, or due to insufficient compensation for every sensor's noise and faults. The main objective of the presented method is to prevent IM anomalies originating from insufficient compensation due to unknown noise of some sensors or due to the limited fault hypothesis space. The method proves that navigation anomaly prediction, based on individual auxiliary sensor's measurements, can be applied for this purpose.

A. Prior work

The utilization of deep learning for anomaly detection in multivariate time-series data has shown impressive results in past works [6]. Specifically, Deep Neural Networks are more appropriate than previous methods to capture temporal dependencies as well as correlations between observations in the data. Zhang, Chen, Wang and Pan [7] highlighted that this can be achieved with a combination of Long Short Term Memory (LSTM) and convolutional layers, where in their work they introduced this architecture in an auto encoder-based approach for anomaly detection. Sun [8] utilized multi-layer LSTM to classify an input sequence of GNSS measurements as faulty or fault-free, based on irregularities in the distribution of observations of an IM test statistic. The model is trained with labeled data which consist of raw GNSS measurements and the fault-alarm or fault-free label. Kim and Cho [9] utilized a Time Delayed Neural Network to detect dissimilarities of the trend of an IM statistic compared to a past trend, where the system was operating under normal conditions. Gupta and Gao [10] utilized deep learning to produce position error and error covariance estimates, that can be used in Protection Level calculation, for a camera + LiDAR navigation system. This is done by comparing static images with a local depth map produced around the current state estimate and given the LiDAR map. A shortcoming of the mentioned research is that they are very specific to the sensors or the IM used. In addition they do not focus on understanding the direct effect that environmental conditions have on navigation and IM anomalies due to sensory faults.

It is worth noting that the majority of existing anomaly detection methods are trained to find the best representation of normal data, requiring the availability of large datasets that are free from anomalies. Methods that try to learn representations of normal data might ignore the important features that differentiate normal from anomalous samples, especially when learned features are redundant, noisy or refer to a specific subset of the training set [11]. Research on supervised learning of anomalies are very few, due to the difficulty of obtaining enough anomaly examples for training, and even in those cases learning of anomalies would be limited to the already labeled anomalies. Here, we build the presented IM method upon our previous work [12], where training data was labeled in few trajectories as normal or anomalous using statistical methods. We leave as future work the automatic detection of more unseen anomalies in presently unlabeled data [13].

B. Contribution

At the start, the method utilizes a supervised Deep Convolution Neural Network model with a Time Distributed Layer, which is trained offline, for the prediction of anomalies in the state estimate error, based on features in the raw sensor input. Then, the method refines the predicted anomalies based on an IM test statistic, which is directly related to the sensor of concern, without the need to consider other auxiliary sensor measurements.

The first main contribution of the presented work is that, despite the extension to auxiliary sensors, the MHSS module computes the PLs informed solely by conventional sensor measurements as before. In this way the IM scales to auxiliary sensor measurements, without modification of existing IM methods and without important computation burdens. An assumption is that the integration of more sensors in a navigation system results in smaller estimation uncertainty, with the latter being consistent in case of absence of sensor faults. Then, the PL will reliably bound the estimate error of the monitored algorithm, which is informed by the full IMU/GNSS/Auxiliary sensor measurement set. The second contribution is that the method can directly capture the relation of anomalies in IM to raw inputs of individual sensors. This allows the evaluation of environment features and conditions, including dynamic changes, that cause specific sensor faults. At the same time, the initial prediction quality is independent from the quality of the IM, allowing a more objective evaluation of different IMs. The eventual fault exclusion is informed by the IM and, therefore, the DNN training can tolerate the presence of certain amount of image noise and incorrect anomaly labels.

We use the proposed method to monitor the integrity of a navigation system that integrates the GNSS and IMU sensors with a camera. It uses car sensor data collected in urban environments. Figure 1 shows example images from the evaluated environments. The method employs a camera-only navigation algorithm to estimate camera poses that are used in a loose integration with the other sensors. The results show that, in the absence of faults in GNSS/IMU, for a system where the PL computation is informed from GNSS and IMU noise only, the detected anomalies in camera-only navigation correlate with IM anomalies. In addition, the results confirm an improvement of the performance of the IM system in terms of the Relaxed Bound Tightness (RBT) metric, when comparing it to the method that does not use camera FDE. Also, it remains computationally efficient.

C. Paper organization

The paper is organized as follows. In section II we present the methodology of predicting the IM anomaly due to camera faults. It describes briefly the DNN model and the IM for collaborative learning of anomalies (II-A), introduces the camera measurement samples tested for faults (II-B), defines the hypotheses tailored to test for camera-faults (II-C) and,

Fig. 1: Example images from the urban navigation datasets examined here. The locations are San Francisco (USA), Oxford (UK) and Dongtan (South Korea). The images are from the UrbanLoco [14], Robotcar [15] and KAIST [16] datasets.

finally, defines the ordinary distribution of IM test statistics and the ordinality tests (II-D and II-E). The evaluated datasets are presented in section III-A, section III-B presents the anomaly prediction results for multiple real trajectories in those datasets, and section III-C discusses the performance of the anomaly prediction and exclusion method with numerical results of how it affects the performance of the IM in bounding the true position errors. Section IV is the conclusion and a discussion of future works.

II. Hypothesis testing for anomaly detection in VN, informed by IM

A. Detection of anomalies in the uncertainty of visual SLAM and in the IM test statistic

Hazardous Misleading Information, which is quantified by an IM, refers to the event where an undetected state estimation error deviates from the true value by more than an alert limit. Here the interest is to monitor state estimation faults that are caused by measurement faults beyond the nominal sensor noise. Usually, there is no knowledge of the magnitude or the direction of a potential fault $\boldsymbol{f}$ in a measurement set and we pursue the computation of the worst-case fault. This is done by employing the concept of fault slope:

$$\boldsymbol{Slope}_f = \frac{\text{effect of } \mathbf{f} \text{ on state estimation error}}{\text{effect of } \mathbf{f} \text{ on the IM test statistic}} \tag{1}$$

Using that definition, in the specific investigated case, the fault $slope$ for camera measurements can be computed as the ratio of the state estimate error caused by camera measurement faults to the IM test statistic measure affected by the camera measurement faults. In MHSS-based IM, the test statistic to detect faults in sensor measurements is the difference in the state estimate under all-source hypothesis H_0 and under hypothesis H_j, which is fault-tolerant to the sensor's measurements:

$$|\boldsymbol{x}_p^{(j)} - \boldsymbol{x}_p^{(0)}| \leq \boldsymbol{T}_{p,\Delta_j} \tag{2}$$

with $\boldsymbol{x}_p^{(j)}$ and $\boldsymbol{x}_p^{(0)}$ denoting the position state under fault hypothesis H_j and fault-free hypothesis H_0, respectively. We use the subscript p to highlight that the vectors refer to the elements of the position state and we drop the subscript when the quantities refer to one direction of interest in the position state. $\boldsymbol{T}_{p,\Delta_j}$ is the detection threshold, which takes into account the continuity risk assigned to H_j. In one direction of interest, the equation for the fault $slope$ in SS RAIM can be simplified to [17]:

$$Slope_{f_j} = \sigma^2_{\Delta_j} \tag{3}$$

where $\sigma^2_{\Delta_j} = \sigma^2_j - \sigma^2_0$, σ^2_j and σ^2_0 are the estimated error covariances under H_j and H_0, respectively.

So:

$$T_{\Delta_j} = K_{fa,j}\sigma^2_{\Delta_j} \tag{4}$$

where $K_{fa,j}$ a factor that accounts for the false alarm rate.

For clarity, let AU be the set of indices for hypotheses under which the measurement set excludes the measurements from one sensor, auxiliary or conventional, but contains measurements from other auxiliary sensors. In this work we consider that estimation under hypotheses H_j, for all $j \in AU$ is computationally inefficient, at least when the sensor fusion algorithm of such integrations has a high computational load. In addition, we do not assume anything of the internal structure of the monitored navigation algorithm which is informed by the measurement set under the all-source hypothesis H_0. In this case, the covariance of position estimates is assumed unavailable, something that prevents the calculation of the detection threshold in (4). Therefore, the method performs fault detection in each auxiliary sensor's measurements individually. This is done via anomaly prediction in the estimate error with a sensor-specific DNN model and refinement based on an IM test statistic which is computed for a specialized measurement set for the sensor of concern, that excludes all other auxiliary sensor measurements. The latter procedure is implemented in a new module called Deep Hypothesis Testing (DHT). The next part of this section describes briefly the DNN model, while section II-C describes the hypotheses that are directly related to the camera and how they are used in the DHT module to detect IM anomalies.

Our previous work [12] described the DNN model for anomaly prediction in the distribution of SLAM state estimate errors. Any VN algorithm can be used instead of SLAM. According to that work, an input sample consists of M sequential images. During training, the SLAM position estimate errors are assumed to be normally distributed around 0 with a covariance matrix R, in the nominal case. Then, the DNN model parameters are optimized to predict that specific low-dimensional image features and dynamic changes within the

image sequence will cause the SLAM position estimate error at the end of the sequence to fall outside the normal distribution of errors.

The DNN combines the Time Distributed Layer with convolutional and max pooling layers, and the LSTM layer. In this way the network learns the effect of non-recurrent features as well as of transient changes to the predicted class output, via the extraction of features from all images in a sample and by associating them chronologically. This is an important characteristic in the context of IM as the model is powerful in capturing fault dynamics.

A diagram of the overall method is shown in figure 2. The figure and the text uses the term Common Sensor Set (CSS) to refer to the set of conventional sensors, i.e. GNSS/IMU. In hypothesis H_{CSS}, all CSS sensors are assumed fault free, while the remaining non-CSS sensors are assumed faulty and are excluded, allowing the execution of standard Error State Kalman Filter (ESKF), as detailed in [4]. The "All-source" hypothesis H_0 assumes the whole set of sensors in CSS and auxiliary sensors as fault-free. In the new method proposed here, the monitored navigation algorithm needs to be executed only once, for that sensor set. The new DHT module is introduced in the general IM architecture, runs in parallel with the IM of CSS sensors, and executes one instance of a specific navigation algorithm for each auxiliary sensor. For example, this is a visual SLAM algorithm in the case of the camera. Although the diagram in figure 2 illustrates the general architecture, where many auxiliary sensors may be integrated with GNSS/IMU, the rest of the article refers specifically to fault detection of camera measurements, as the camera is the only auxiliary sensor evaluated herein. The detailed explanation of the DHT module follows.

B. Hypotheses fault-tolerant to image frames

A predicted navigation anomaly from the previously described DNN model will initiate DHT to detect the specific anomalous frames that cause IM hazards due to the camera. This and the next section describe the new hypothesis set.

Consider that a navigation anomaly is predicted by the DNN at a step k^*. The method searches for IM anomalies around k^* in a horizon with pre-selected size N_{vis}. Let us denote the full set of image frames around the anomaly as $Y_{cam,k^*-b}^{k^*+a}$, with $a-b=N_{vis}$. Then, we will denote each image frame obtained in the horizon in any step k as $c_k \in Y_{cam,k^*-b}^{k^*+a}$. In a typical VN algorithm, the estimation of the camera pose w.r.t. to the first frame is achieved by tracking features observed in subsequent frames. Let us denote the camera pose:

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{p}_C^{(W)} \\ \boldsymbol{r} \end{bmatrix} \in \mathbb{R}^{3+r_m} \tag{5}$$

with $\boldsymbol{p}_C^{(W)} \in \mathbb{R}^3$ being the camera position in the global reference frame (W) and $\boldsymbol{r}$ being the camera attitude vector parametrized with $r_m = 3$ or 4 elements, depending on the representation used. A 2D projection of a feature l with coordinates $\boldsymbol{p}_l^{(W)}$ to the image plane is symbolized as $\pi_l(\boldsymbol{x})$ and the pinhole camera model can be used for the formula of $\pi_l(\boldsymbol{x})$ [18]. It is worth noting that the variable $\boldsymbol{x}$ is there because the projection depends on $\boldsymbol{p}_C^{(W)}$ and the rotation matrix between camera body frame and global frame. Then the noisy measurement of a feature point extracted from the measurement image is given as:

$$\boldsymbol{z}_l = \pi_l(\boldsymbol{x}) + \boldsymbol{v}_l + \boldsymbol{b}_l \tag{6}$$

where $\boldsymbol{v}_l$ is zero-mean Gaussian noise, and $\boldsymbol{b}_l$ is a bias in the feature location. Given N_L feature correspondences, in total, tracked in N_{vis} subsequent frames, the camera pose estimate $\hat{\boldsymbol{x}}$ can be found by solving the nonlinear optimization problem:

$$\hat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}} ||\boldsymbol{z} - \pi(\boldsymbol{x})||^2_{\boldsymbol{\Sigma}_{\boldsymbol{v}}^{-1}} \tag{7}$$

where $\boldsymbol{z} = \begin{bmatrix} \boldsymbol{z}_1 \\ \vdots \\ \boldsymbol{z}_L \end{bmatrix}$, $\pi(\boldsymbol{x}) = \begin{bmatrix} \pi_1(\boldsymbol{x}) \\ \vdots \\ \pi_L(\boldsymbol{x}) \end{bmatrix}$, and $\boldsymbol{\Sigma}_{\boldsymbol{v}}$ is the covariance matrix of the stacked noise vector $\boldsymbol{v} = \begin{bmatrix} \boldsymbol{v}_1 \\ \vdots \\ \boldsymbol{v}_L \end{bmatrix}$.

For simplicity, by grouping the features into their image frames, the measurement vector can be stacked as

$$\boldsymbol{Z} = \begin{bmatrix} \boldsymbol{Z}_1 \\ \vdots \\ \boldsymbol{Z}_{N_{vis}} \end{bmatrix} \tag{8}$$

where $\boldsymbol{Z}_k$, $k = 1, ..., N_{vis}$ is a vector with feature points found in image frame $\boldsymbol{c}_k$. This notation will help us define the hypotheses with clear association to image frames (i.e. groups of feature correspondences that might contain faults due to faults in the respective image frame) instead of individual features.

C. Hypotheses definition

The input measurement set to the "DHT for camera" module in figure 2 consists of camera and CSS measurements only. As the DHT module is part of the overall IM, let us define as H_{CC} the hypothesis that is associated with the all-source measurement set in the DHT module, i.e. H_{CC} is the hypothesis that all auxiliary sensors other than the camera are faulty. In DHT, a VN system utilizes the image frames to estimate camera poses, and those are compared with CSS measurements to detect inconsistencies. Such inconsistencies guide the IM anomaly detection process.

For IM anomaly detection due to camera measurements, the method utilizes $N_{vis}+1$ hypotheses, where under hypothesis $H_{CC,j}(j = 0, 1, \ldots, N_{vis})$ from 0 to N_{vis} image frames are assumed to be associated with an IM anomaly. Hypothesis testing is initiated if an image frame $\boldsymbol{c}_k$ is associated with an anomaly in the true error (after employing the DNN predictor) with high probability. Then, under hypothesis $H_{CC,j}$, the subset of frames that inform the navigation solution is $Y_{vis} = \{y_k | 1_k = 1, \forall k \in [1, N_{vis}]\}$ with 1_k defined as:

$$1_k = \begin{cases} 0 \quad if \quad (j+k-N_{vis}) \leq 0 \\ 1, otherwise \end{cases} \tag{9}$$

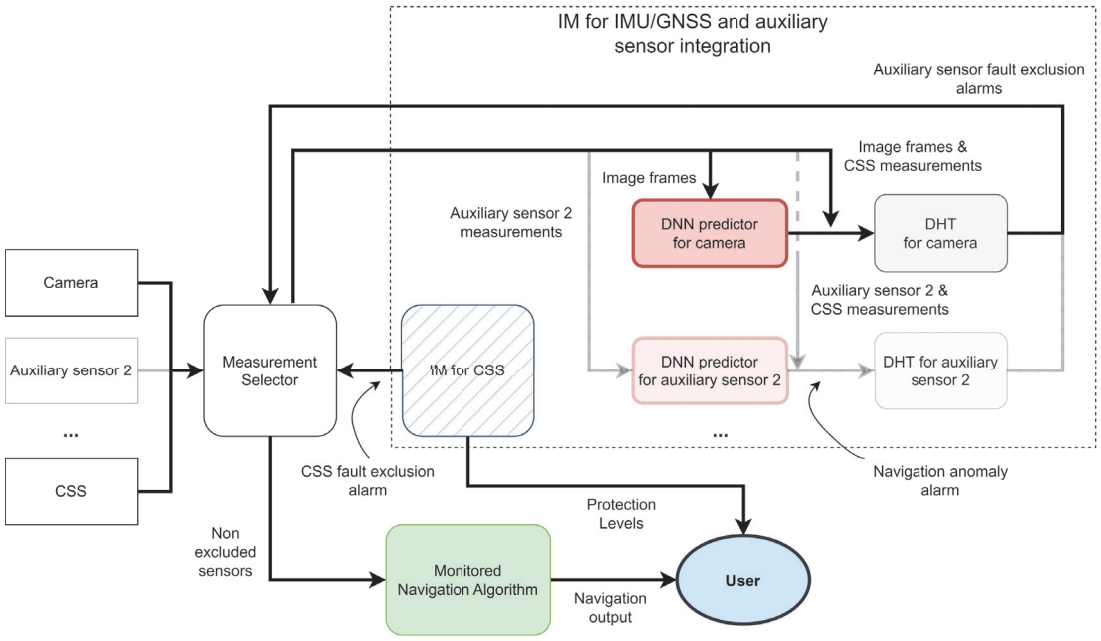


Fig. 2: Diagram of the proposed IM architecture for integration of GNSS/IMU with auxiliary sensors. Although this paper investigates only the case of the camera, the diagram shows the general case, where the parts that refer to other auxiliary sensors are visualized with higher transparency.

For example, under $H_{CC,0}$ zero frames are included, as $0 + k - N_{vis} \leq 0$ for any $k = 1, \ldots, N_{vis}$. Under $H_{CC,1}$ only the N_{vis}-th image frame is included, while under $H_{CC,N_{vis}}$ all image frames from indexes 1 to N_{vis} are included. This hypothesis is identical to H_{CC}.

Figure 3 shows the diagram of the DHT module for camera measurements. Importantly, when developing the DHT module for other auxiliary sensors, only the VN algorithm needs to change to another sensor-specific algorithm.

Each hypothesis $H_{CC,j}$ is associated with a sequence with N_{vis} observations $\boldsymbol{y}^{(CC,j)} = (y_1^{(CC,j)}, \ldots, y_{N_{vis}}^{(CC,j)})$, $j = 0, \ldots, N_{vis}$ of the values of an IM test statistic, which we denote as a process under hypothesis $H_{CC,j}$. A new child process is possibly created after N_{vis} steps, if the system has not returned to ordinality (section II-D).

A subset of observations does not include a fault when the observations follow the ordinary distribution λ of IM test statistics, while, if there is one or more faults, the observation at any step $k \in [1, N_{vis}]$ cannot be drawn from the ordinary distribution with large likelihood. The next section introduces the ordinary distribution and the ordinality test for the process under each hypothesis.

D. Ordinary distribution

According to equation (2), for the new hypothesis set tailored to camera measurements, at each step k we define as test statistic the difference between the position solution under H_{CC} and hypotheses $H_{CC,j}$:

$$\boldsymbol{\tau}_{pCC,j} = |\boldsymbol{x}_p^{(CC,j)} - \boldsymbol{x}_p^{(CC)}| \tag{10}$$

Zhang, Wang and Gao [19] analyzed the distributions of the MHSS test statistic for different fault hypotheses and suggested that they vary due to the differences in the underlying measurement subsets. The MHSS test statistics under each fault hypothesis are normally distributed [20]. For one position direction of interest the distribution of the IM test statistic is:

$$\tau_{CC,j} = \mathcal{N}(f_{\Delta_{CC,j}}, \sigma^2_{\Delta_{CC,j}}) \tag{11}$$

Here

$$\sigma^2_{\Delta_{CC,j}} = \sigma^2_{CC,j} - \sigma^2_{CC}$$

where

$\sigma^2_{CC,j}$, σ^2_{CC} are the estimated variances in the direction of interest for the solutions under $H_{CC,j}$ and H_{CC} respectively, and

$$f_{\Delta_{CC,j}} = f_{CC} - f_{CC,j}$$

where

f_{CC} is the contribution of measurement faults to the position state of interest estimated under H_{CC}, and

$f_{CC,j}$ is the contribution of faults of non-excluded measurements under hypothesis $H_{CC,j}$ to the corresponding position state of interest.

This paper deals with a set of processes, where the measurement set informing each process is different among the simultaneous processes running at each step as well as between a parent and a child process possibly created after the test of

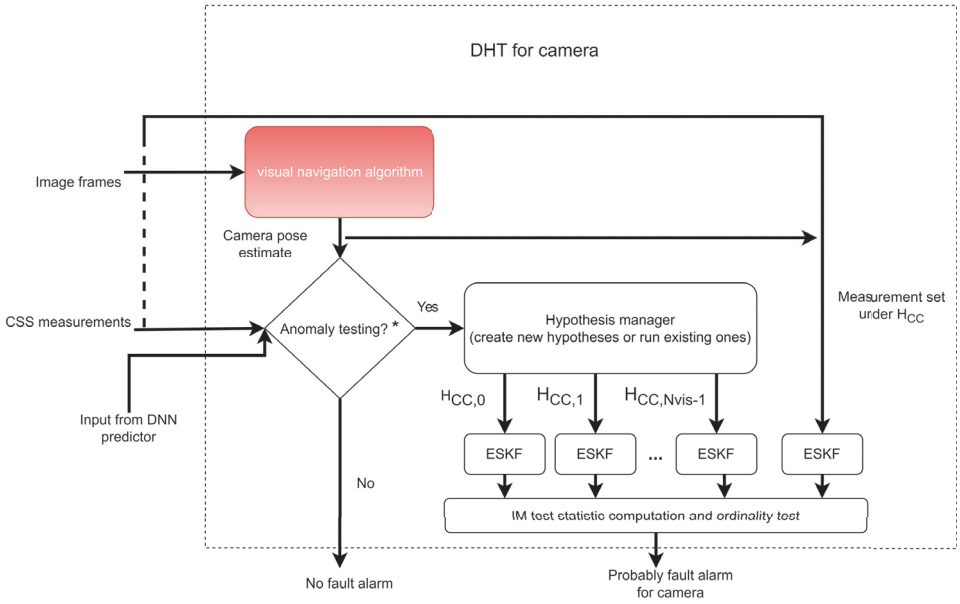


Fig. 3: Diagram of the DHT module for fault detection in camera measurements. Different instances of ESKF are running for each hypothesis, although the implementation of the filter is always the same.
* Anomaly testing runs if either the DNN predicted a navigation anomaly or if the DHT is still evaluating a previous anomaly.

ordinality in equation (13). This means that a test statistic distribution determined to fit the samples of a process does not necessarily represent a distribution of ordinality that can be used for reliably testing the samples of any other new process. However, under fault-free conditions, the distributions for all-hypotheses are expected to have zero mean (according to equation (11), although with different variances. In this paper, only one distribution of ordinality is utilized per direction, computed with samples from all processes. This distribution is found to follow a zero-mean normal distribution:

$$\lambda \sim \mathcal{N}(0, \sigma^2_{\lambda_a}) \tag{12}$$

where the subscript a refers to the direction of interest. The experimentation in this work is for 2D navigation and uses North-East-Down (NED) as frame of reference for the vehicle positions, so the two directions of interest are North and East.

To obtain the ordinary samples and the ordinary distribution in the case study, the complete IM system with the monitored navigation algorithm is run for driving road vehicle trajectories with labeled sections where the outputs appear ordinary, after utilizing statistical methods. The labeling of ordinary sections is accomplished offline based on a sliding Z-score metric $Z = (\hat{e}_S - \mu_S)/\sigma_S$ of the SLAM error estimates $\hat{\hat{e}}_S$ in one direction to a known reference trajectory. The mean μ_S and standard deviation σ_S refer to the normal distribution of errors computed in a horizon around the sample. The difference of an error to the mean is compared with a given number of standard deviations to decide if the corresponding sample is ordinary. This is a statistical method to detect anomalies that was also used to label anomalies in the training set in [12]. Understandably from the above, the definition of the ordinary distribution is an open topic for more research in the future. Figure 4 depicts the computed ordinary distribution that was utilized in the experiments to decide anomalous samples.

E. Test for anomalies

The method identifies the end of an anomaly by executing a test at the finishing step $k_{1,\kappa} = \kappa N_{vis}$ of the current running processes, with $\kappa \in \mathcal{Z}^+$ a positive integer. Starting from the hypothesis $H_{CC,N_{vis}-1}$, that assumes that only the 0th image frame is faulty, until hypothesis $H_{CC,0}$, that assumes all N_{vis} image frames are faulty, the test is:

$$\begin{aligned} y_i^{(CC,j)} &\in [F_\lambda^{-1}(0.05), F_\lambda^{-1}(0.95)] \\ &\forall i \in [1, N_{vis}],\ \forall j \in [0, N_{vis}-1] \end{aligned} \tag{13}$$

where:

F_λ^{-1}, F_λ^{-1} the inverse of the Cumulative Distribution Function (CDF) of the ordinarily distributed variable λ.

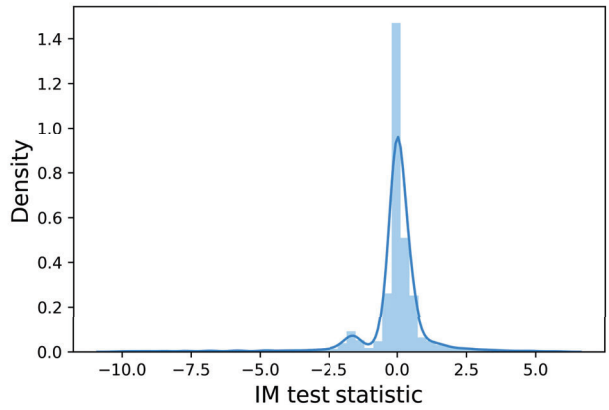
$f(y_i^{(CC,j)}; \lambda)$ the probability density function of sample $y_i^{(CC,j)}$ drawn from the ordinary distribution λ.

A new set of processes is created at time $k_{1,\kappa}$, until the ordinality test in equation (13) holds.

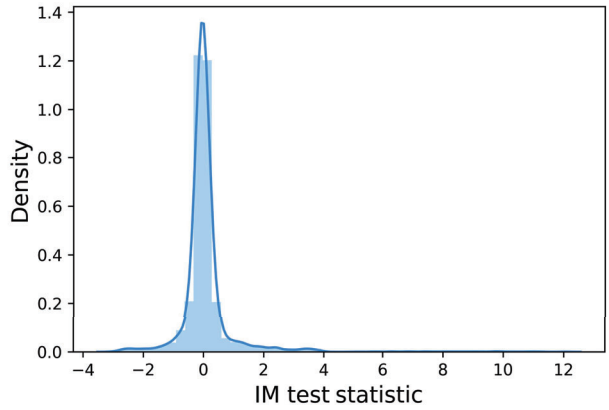
III. Evaluation

A. Trajectories and system setup

The performance of the proposed method is evaluated by utilizing datasets from the UrbanLoco [14], Robotcar [15] and KAIST [16]. The datasets provide camera, GNSS, IMU and other sensor data collected during car drives in urban environments with dynamic objects, illumination changes and



North axis



East axis

Fig. 4: Ordinary distribution of IM test statistics, on each axis. It is computed from samples of the metric that were selected as ordinary based on statistical tests. The samples were obtained during IM of real trajectories in the case study.

repetitive patterns. In each dataset, the car completes one trajectory. We selected two datasets from UrbanLoco, referred to as UrbanLoco1 and UrbanLoco2 in the text, one dataset from Robotcar and one dataset from KAIST. The trajectories in the datasets are split into sections of around 1500 to 2250 steps (150s - 225s) and the evaluation is done for one section from each trajectory, excluding some steps at the beginning to allow initialization. The sections are:

- UrbanLoco1: Steps 0-1500 in the dataset CAColiTower20190828184706, collected near Coit Tower, San Francisco: Very busy area at the beginning, many pedestrians and pedestrian crossings, illumination challenges.
- UrbanLoco2: Steps 0-1500 in the dataset CALombardStreet2019082819041, collected near Lombard street, San Francisco: Not that busy area, a lot of shadows / dark areas, hilly terrain.
- Robotcar: Steps 1000-3000 in the dataset 2015-08-14-14-54-57, collected in Oxford, UK: not busy area, roadworks, severe illumination challenges.
- KAIST: Steps 4500-6750 Suburban area, a lot of car traffic, High rising buildings with repetitive patterns, severe illumination challenges.

The datasets provide reference trajectories; UrbanLoco provides data from a SPAN-CPT module. Robotcar provides a ground truth generated using post-processed raw GPS, IMU, and static GNSS base station recordings [21]. KAIST provides data from a pose-graph SLAM solution that fuses various sensors. Although we use the provided reference data for all trajectories, the accuracy of the reference for the KAIST trajectory might vary and be significantly deteriorated under conditions with GNSS unavailability [16]. In addition, we found out that the reference does not cover the whole trajectory, and we filled those gaps by the estimates from a simple integration of GNSS/IMU in ESKF. Therefore, the reference for that trajectory is inaccurate in some parts.

We utilize the ORBSLAM2 algorithm [22] for camera pose estimation for the UrbanLoco and KAIST trajectories. For the Robotcar trajectory we use estimation data that are already provided with the dataset and were obtained using a Visual Odometry solution. In all cases, the Umeyama method [23] is utilized to find the optimal transformation between the reference and the corresponding camera pose estimates. In the following, the text uses the same term VN for both camera pose estimation algorithms.

At this stage, splitting of the trajectories was important to limit learning navigation anomalies occurring due to previous drifts induced by visual-only navigation. There is future work to do to label the exact frames that associate with anomalies. This will require to go from statistical methods in the overall trajectory error to relative error methods and handling of the drift.

The number of frames to test each time was selected $N_{vis} = 10$. This number should be selected to be at least equal to the number of image frames M per input sample that is fed to DNN. That number is 7, as can be seen in our previous work [12]. A key finding of that work was that larger samples lead to better anomaly learning than smaller ones, although there is an upper limit for the sample size due to limited hardware resources. A selection $N_{vis} > M$, as in this work, is done to account for probably anomalous segment that begins a bit before a predicted anomaly from the DNN.

Figure 5 shows the reference trajectories, the evaluated sections and the aligned estimated sections from the VN system.

The IM for CSS, informed by the IMU and GNSS noise, computes the PLs based on an integrity requirement I_{REQ} that was preselected to be in the order of 10^{-8}.The experiments employ a formula for the computation of PLs that was previously utilized for Receiver Autonomous Integrity Monitoring (RAIM) [17]. The PL in each direction, for the fault-free case, is computed as:

$$p_L = Q^{-1}\{I_{REQ}/2\}\sigma_{CSS} \tag{14}$$

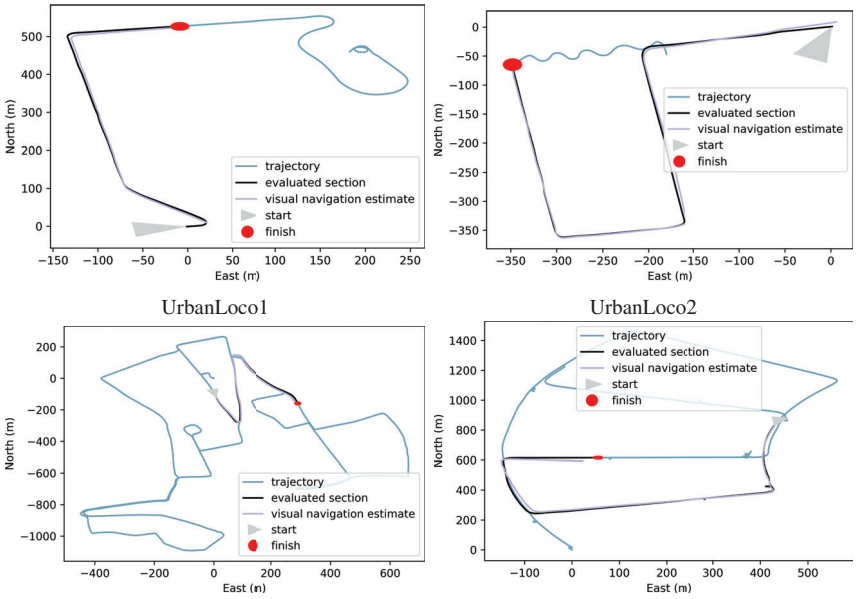


Fig. 5: Evaluated trajectories. The figures show the full reference trajectories, the evaluated sections of each trajectory and the aligned camera poses for those sections, estimated from the VN system.

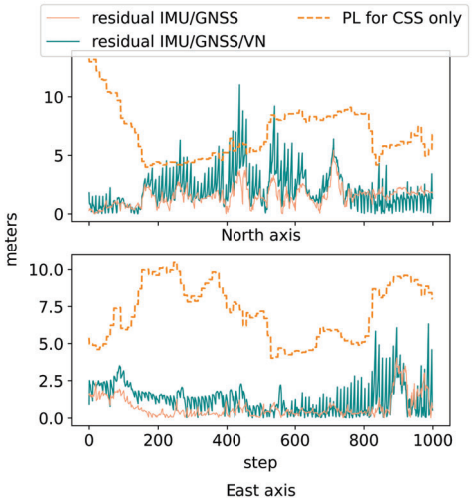


Fig. 6: PLs, informed by IMU and GNSS nominal noise, and residual errors before and after integrating IMU/GNSS with the VN system, for UrbanLoco1.

where σ_{CSS} is the estimated variance of the state error under H_{CSS} in the respective direction, and Q^{-1} is the inverse tail probability of the standard normal distribution. As there is no redundancy of the measurements in the utilized CSS, the PLs do not compensate for possible faults in the measurements of those sensors. Nevertheless, we tuned their noise parameters and confirmed that there are no significant faults by utilizing consistency checks and IM.

B. Anomaly detection results

Here, we analyze the performance of IM anomaly detection after executing the DHT module, with or without utilization of the DNN. If the DNN is not utilized, the DHT runs uninterrupted for the detection of anomalies. Uninterrupted execution means that:

a Anomaly testing does not stop even if the ordinality test in equation (13) succeeds.
b Camera measurements are not excluded even after the detection of anomaly.

For UrbanLoco1, figure 6 shows the result of the IM for the integrations IMU/GNSS and IMU/GNSS/VN. In both cases, the PLs are only informed from GNSS and IMU nominal noise, and are computed based on equation (14). We observe an increase of the error at the North direction after the integration with VN, especially after step 180. This results in many IM anomalies that last until approximately step 500. For the same trajectory, figure 7 shows the minimum log Probability Density Function (log PDF), as defined in equation (15) below. It also shows the DHT processes, that the ordinality test in equation (13) identifies as anomalous, under hypotheses $H_{CC,0}$, $H_{CC,4}$ and $H_{CC,9}$. In this evaluation, the DHT module runs uninterrupted.

The experiments use the minimum log PDF metric, that is defined here, to visualize the likelihood that the various DHT processes under different hypotheses are ordinary. Previously, we used the symbol $\boldsymbol{y}^{(CC,j)}$ to denote the sequence of IM test statistics that are obtained during a process executed under hypothesis $H_{CC,j}$. The minimum log PDF of the samples in $\boldsymbol{y}^{(CC,j)}$, given the ordinary distribution λ, is:

$$l_{CC,j}(\boldsymbol{y}^{(CC,j)};\lambda) = \min\{ln(f(y_1^{(CC,j)};\lambda)),...,\\ ln(f(y_{N_{vis}}^{(CC,j)};\lambda))\} \tag{15}$$

with $f(y_i^{(CC,j)};\lambda)$, $i = 1,...,N_{vis}$ being the PDF around the sample $y_i^{(CC,j)}$ when it is drawn from the ordinary distribution λ.

By comparing with figure 6, we observe that the sharp drops of the minimum log PDF in the plots are quite precise in indicating the location of IM anomalies. In addition, processes that do not include any or include few image frames (i.e. processes under $H_{CC,0}$ and $H_{CC,4}$) appear anomalous more often than $H_{CC,9}$, that includes most of the image frames in the sequence. This is expected as for processes, like $H_{CC,9}$, where the fault-free measurement set is very similar to that under H_{CC}, the IM test statistic (equation (10)) is generally closer to zero. Additionally, the plots indicate that the effect of individual image frames on the build-up of an anomaly may vary. An example is around step 450, where the plot for $H_{CC,9}$

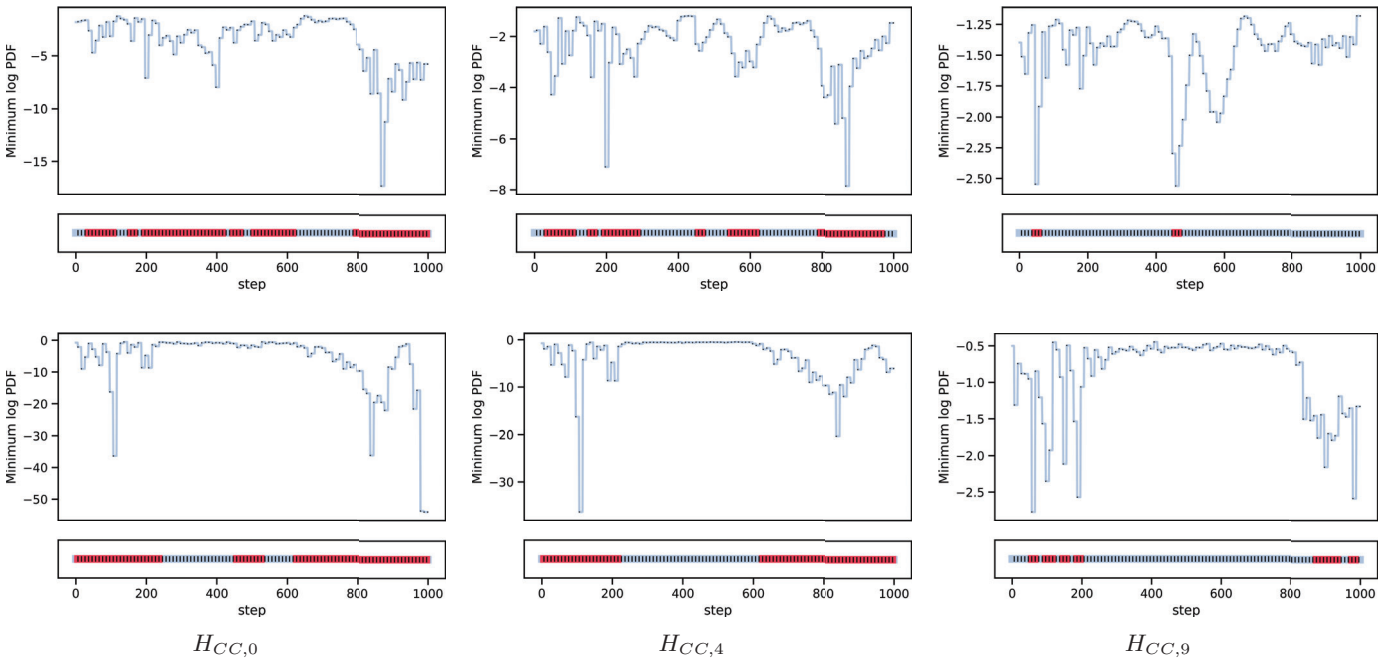


Fig. 7: For UrbanLoco1, the plots show the values $l_{CC,j}(\boldsymbol{y}^{(CC,j)};\lambda)$ for $j = 0, 4, 9$ over time and the processes that are detected anomalous (red) or pass the ordinality test (light blue). Processes are shown as subsequent segments separated by a vertical line at the bottom of each plot.

in North direction shows that one of the tested processes is anomalous. The sharp decrease in the min log PDF is likely caused by the only image frame assumed non-faulty in the anomalous process. This is not the case for the anomaly at around step 850. The anomaly is clearly visible in the plots for $H_{CC,0}$ and $H_{CC,4}$ but not in the plot for $H_{CC,9}$. The anomaly seems to build up over more than one image frames in the processes.

The next experiment introduces the DNN in the anomaly detection procedure. This means that DHT initiates testing only after receiving a navigation anomaly alarm from the DNN. Anomaly testing stops when the ordinality test succeeds. No fault exclusion is implemented at this experiment. Figure 8 shows, for all evaluated trajectory sections, the PLs per axis as computed in equation (14). Additionally, it plots the residual of position state estimates under H_0 to reference positions. These may be compared with the PLs for bounding issues. The figure also annotates the VN anomalies detected by the DNN. The detected IM anomalous segments are marked on the residual. The DNN does not differentiate between anomalies in the North and East direction, as training was done based on the maximum of the VN system's estimate errors in the two directions. Therefore, the VN anomalies are the same in both directions. The y-axis location of VN anomalies indicates the magnitude of the VN estimate error (maximum between the two axes). Lastly, a detection for IM anomaly may happen for any of the two directions, but the method will alarm for an anomalous segment regardless.

Figure 9 shows, for UrbanLoco1, the result of fault exclusion using the proposed method. The PLs and the residual can be compared with figure 6. The fault exclusion leads to a reduction of the residual error and the prevention of some IM anomalies. However, the exclusion is insufficient to prevent all anomalies.

A general conclusion from the figures is that the proposed method performs well in identifying navigation and IM anomalies, in the vast majority of cases where there is anomaly present. The DHT module was able to mitigate for false alarms from the DNN, although there were also few cases where it falsely ignored them. The method was unsuccessful to exclude enough camera measurements and prevent all IM anomalies, as we observed in the experimentation with fault exclusion. Therefore, despite the very promising results, we observe that further tuning is required to adequately prevent all anomalies, and limit false positives, i.e. detection alarms for camera measurements that do not actually cause anomalies. The quantitative evaluation of the method follows in the next section.

C. Numerical evaluation of the IM performance

This section evaluates the performance of the proposed camera FDE when monitoring the GNSS/IMU/VN system and PLs are only informed by GNSS and IMU noise. Four metrics are evaluated and these are:

1) The RBT metric, proposed by Li and Waslander [24] to quantify how much of the time the error is sufficiently bounded, as well as the tightness of the bound. This metric is calculated as follows:

$$RBT = \sqrt{\frac{\sum_{i=1}^{N} \rho\left(\frac{p_{L_i} - |e_i|}{\sigma_i}\right)^2}{N}} \tag{16}$$

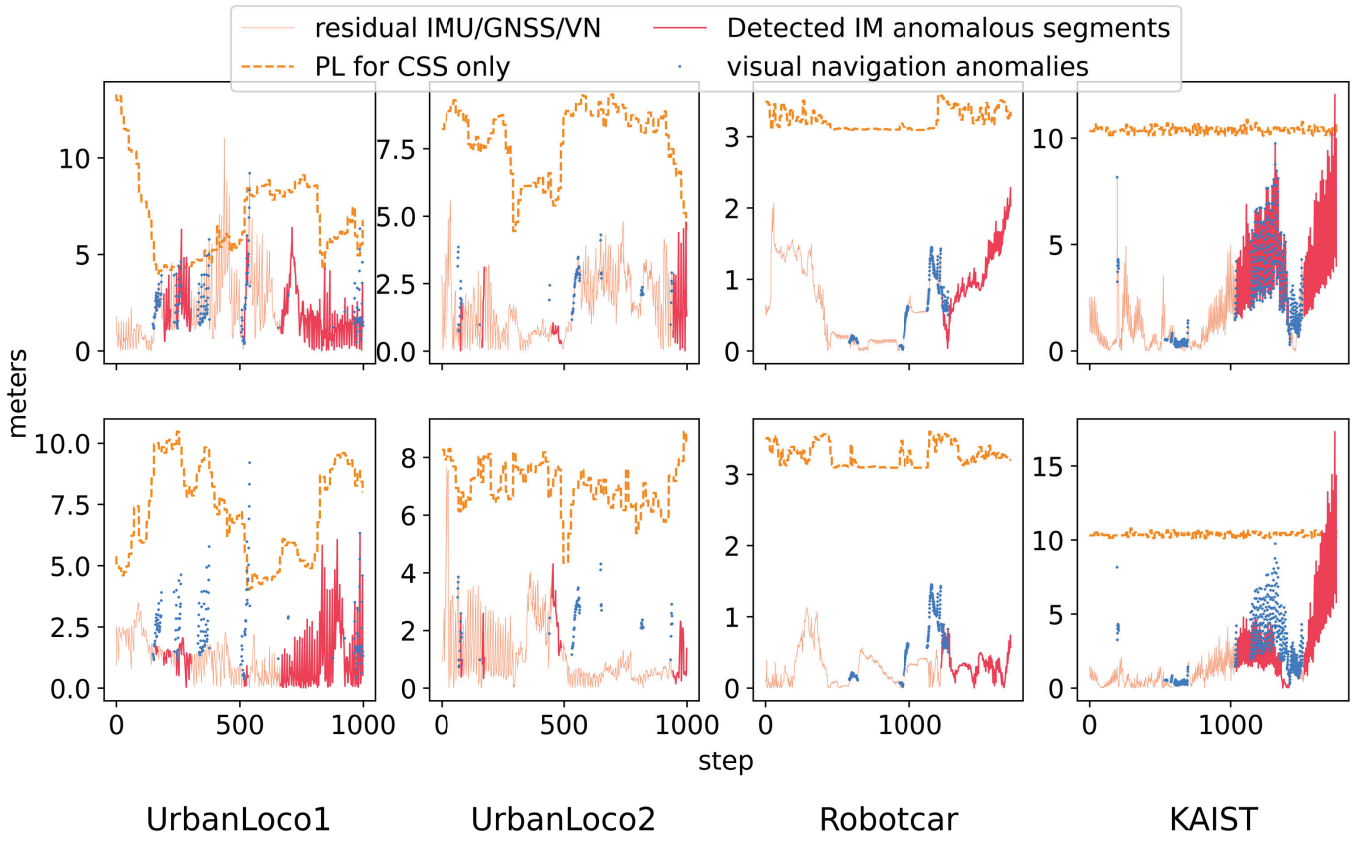


Fig. 8: Detected anomalies, true position error of GNSS/IMU/VN integration and PLs informed by the GNSS/IMU only. Each column shows the results for one trajectory. The top row shows results for the North direction and the bottom row shows results for the East direction.

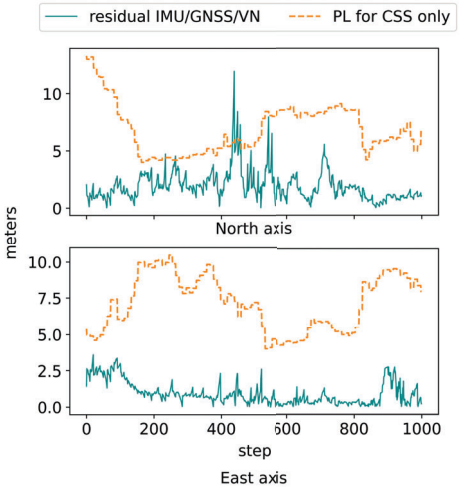


Fig. 9: Results of IM for the IMU/GNSS/VN integration after utilizing the DHT module for VN fault exclusion. The results are for the UrbanLoco1 trajectory.

where p_{L_i} and $|e_i|$ are respectively the PL and the error for a sample time i in one direction, N is the number of samples, σ_i is the error variance for the sample. ρ is a weight function that should penalize bounding failures more than loose bounds. In the experiments $\rho = 64$.

2) The percentage of time that the errors are bounded by the PLs.
3) The minimum position alert limit for 100% availability of the system (it is the maximum protection level).
4) The average execution time.

As described in section II-E, DHT runs until the process under $H_{CC,0}$ is found ordinary. In the experiment, new anomalies are ignored if DHT is already running by the time of their detection.

Table I summarizes the results for the various trajectory sections. As the table compares the performance of our method against different IM setups, it is worth noting that the table and the text symbolizes as "IM for CSS" the IM that is informed only by GNSS and IMU noise, while it symbolizes as "GNSS/IMU/VN MHSS" the MHSS-based IM that is informed by GNSS + IMU + VN noise and faults. In the latter case, the IM does a simplistic assumption of white noise in the VN estimates. The IM compensates for

the contributions of failures in the GNSS measurements or VN estimates by defining the $N_{SS} = 2$ fault hypotheses H_j ($j = 1, 2$), respectively fault tolerant to GNSS measurements or VN estimates. It assumes equal distribution of the integrity risk I_{REQ} and continuity risk to the fault-free hypothesis H_0 and the two fault hypotheses. The PL under each fault hypothesis, in each direction, is computed as:

$$p_{L_j} = Q^{-1}\{\frac{I_{REQ}}{P(H_j)(N_{SS}+1)}\}\sigma_j + T_{\Delta_j} \tag{17}$$

where σ_j is the estimated variance of the state error under hypothesis j and T_{Δ_j} is given in equation (4). Under fault-free hypothesis H_0 the PL compensates for the nominal noise of all three inputs:

$$p_{L_0} = Q^{-1}\{\frac{I_{REQ}}{2(N_{SS}+1)}\}\sigma_0 \tag{18}$$

The IM selects the maximum of the three PL bounds as the final PL.

The first row in the table for each trajectory shows the performance of our method, which utilizes IM for CSS and the DHT module for camera FDE. It compares it against the method where the DHT module runs continuously to find IM anomalies and performs fault exclusion, independently if the DNN predicts anomalies ("DHT, no DNN" at second row), as well as against the method which does not use the DHT module ("IM for CSS" method at third row). The last row shows the performance of the MHSS where PLs are also informed from camera faults ("GNSS/IMU/Camera MHSS"). The table shows the average metrics across both North and East axes.

The method based on IM for CSS without camera FDE (third row) shows the worst performance in terms of bounding the error, while, by using the IM for CSS + DHT (our method), we obtain comparable results to the GNSS/IMU/VN MHSS (fourth row) in terms of the execution time and the percentage of time with bounded error. The integration of more sensors results in lower uncertainty which is consistent when the sensors do not contain faults. The results are even better, in terms of prevention of IM anomalies, when the method runs DHT and fault exclusion, without using alarms from the DNN (second row), although with significantly higher computation time than when using the DNN (our method). The GNSS/IMU/VN MHSS shows the worst performance on the RBT metric due to very loose PLs. However, one should pay attention when interpreting the differences in the RBT metrics obtained for the IM setups compared here; The RBT metric rewards tighter PLs. It might be that the exclusion of the camera, due to a false alarm, results in larger estimate error, still bounded by the PL, and the estimated uncertainty increases too. RBT will indicate that the result is better, although the navigation algorithm is less accurate.

We believe that the results are motivating, since, compared to the GNSS/IMU/VN MHSS method, the proposed method scales better to integrations with more auxiliary sensors. Also, it does not assume knowledge of the noise of auxiliary sensor measurements. Finally, it achieves 100% availability for a lower Alert Limit, as it does not compensate for noise and faults of the auxiliary sensors. Therefore, the proposed method should be considered during the development of computationally efficient and sensor-agnostic IMs, although further tuning is required to prevent anomaly misdetections and insufficient camera measurement exclusion. A possible further research will be around PL inflation, in order to compensate for camera faults, still based on the presented limited hypothesis space of measurement faults.

IV. Conclusion

The presented method constrains the space of sensor fault hypotheses in MHSS-based IMs for multi-sensor navigation systems and is suitable for integrations of the conventional GNSS and IMU sensors with auxiliary sensors. There are two important modifications in the MHSS-based IM architecture of previous works. PL computation is only informed from the conventional sensors IMU/GNSS as before, while the IM evaluates each auxiliary sensor for FDE individually, assisted by a DNN. In this work the evaluated system is, specifically, the integration IMU/GNSS/Camera. The method is able to detect the vast majority of IM anomalies, although there are some misdetections and fault alarms. It shows comparable results, in terms of PL bounding performance, to the typical MHSS-based IM, that is informed by the noise and faults of all sensors. In contrast to that IM, the proposed method is scalable to integrations with multiple auxiliary sensors and does not rely on the availability of a noise model for each auxiliary sensor. An additional contribution is that the initial anomaly prediction in the distribution of navigation faults caused by the camera is done based on raw measurements. Hence, the method can facilitate future researches to quantify the direct effect of various environmental conditions on the performance of IM, via the effect on individual sensors. This may allow the development of robust IMs that are tested in a predetermined set of challenging simulated or real environmental conditions.

References

[1] T. Hassan, A. El-Mowafy, and K. Wang, "A review of system integration and current integrity monitoring methods for positioning in intelligent transport systems." *IET Intell Transp Syst.*, 2021.

[2] A. Appleget, R. C. Leishman, and J. Gipson, "Evaluation of sensor agnostic all-source residual monitoring for navigation." *Proceedings of the 2021 International Technical Meeting of The Institute of Navigation*, 2021.

[3] Q. Meng and L.-T. Hsu, "Integrity monitoring for all source navigation enhanced by kalman filter based solution separation." *IEEE Sensors Journal*, 2020.

[4] V. Bosdelekidis, T. H. Bryne, N. Sokolova, and T. A. Johansen, "Navigation algorithm-agnostic integrity monitoring based on solution separation with constrained computation time and sensor noise overbounding." *J Intell Robot Syst*, 2022.

[5] J. Bednář, M. Petrlík, K. C. T. Vivaldini, and M. Saska, "Deployment of reliable visual inertial odometry approaches for unmanned aerial vehicles in real-world environment," *International Conference on Unmanned Aircraft Systems (ICUAS)*, 2022.

[6] K. Choi, J. Yi, C. Park, and S. Yoon, "Deep learning for anomaly detection in time-series data: Review, analysis, and guidelines." *IEEE Access*, 2021.

TABLE I: Results of the proposed IM for the GNSS/IMU/Camera navigation system and comparison with other IM setups.

		RBT	Percentage of Time available with bounded error (%)	Available for horizontal alert limit (m)	Average execution time (s)
UrbanLoco1	IM for CSS + DHT (our method)	3.76	98.1	13.28	0.192
	DHT, no DNN	3.6	98.5	13.28	0.250
	IM for CSS	5.42	97.3	13.28	0.077
	GNSS/IMU/VN MHSS	18.94	100	36.96	0.129
UrbanLoco2	IM for CSS + DHT (our method)	3.99	100	9.50	0.142
	DHT, no DNN	3.56	100	9.50	0.215
	IM for CSS	4.22	100	9.50	0.077
	GNSS/IMU/VN MHSS	9.55	100	27.76	0.106
Robotcar	IM for CSS + DHT (our method)	3.645	100	3.6	0.097
	DHT, no DNN	3.645	100	3.6	0.155
	IM for CSS	3.65	100	3.6	0.066
	GNSS/IMU/VN MHSS	19.245	100	21.92	0.147
KAIST	IM for CSS + DHT (our method)	4.14	100	10.85	0.149
	DHT, no DNN	4.15	100	10.85	0.217
	IM for CSS	3.99	98.5	10.85	0.071
	GNSS/IMU/VN MHSS	6.535	99.9	17.18	0.101

[7] Y. Zhang, Y. Chen, J. Wang, and Z. Pan, "Unsupervised deep anomaly detection for multi-sensor time-series signals." *IEEE Transactions on Knowledge and Data Engineering*, 2021.

[8] Y. Sun, "RAIM-NET: A deep neural network for receiver autonomous integrity monitoring." *Remote Sens.*, 2020.

[9] D. Kim and J. Cho, "Improvement of anomalous behavior detection of GNSS signal based on TDNN for augmentation systems." *Sensors*, 2018.

[10] S. Gupta and G. Gao, "Data-driven protection levels for camera and 3D map-based safe urban localization." *NAVIGATION: Journal of the Institute of Navigation*, 2021.

[11] B. Sofman, B. Neuman, A. Stentz, and J. A. Bagnell, "Anytime online novelty and change detection for mobile robots." *J. Field Robotics*, 2011.

[12] V. Bosdelekidis, T. A. Johansen, and N. Sokolova, "DNN-based anomaly prediction for the uncertainty in visual SLAM," *17th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, 2022.

[13] G. Pang, A. van den Hengel, C. Shen, and L. Cao, "Toward deep supervised anomaly detection: Reinforcement learning from partially labeled anomaly data." *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery Data Mining*, 2021.

[14] W. Wen, Y. Zhou, G. Zhang, S. Fahandezh-Saadi, X. Bai, W. Zhan, M. Tomizuka, and L.-T. Hsu, "UrbanLoco: A full sensor suite dataset for mapping and localization in urban scenes," *IEEE International Conference on Robotics and Automation (ICRA)*, 2020.

[15] W. Maddern, G. Pascoe, C. Linegar, and P. Newman, "1 year, 1000km: The Oxford RobotCar dataset," *The International Journal of Robotics Research (IJRR)*, vol. 36, no. 1, pp. 3–15, 2017.

[16] J. Jeong, Y. Cho, Y.-S. Shin, H. Roh, and A. Kim, "Complex Urban Dataset with multi-level sensors from highly diverse urban environments," *International Journal of Robotics Research*, 2019.

[17] M. Joerger, F.-C. Chan, and B. Pervan, "Solution separation versus residual-based RAIM." *J Inst Navig*, 2014.

[18] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, 2004.

[19] S. Zhang, Z. Wang, and Z. Gao, "Research on exact thresholds for ARAIM MHSS fault monitoring." *Mathematical Problems in Engineering*, 2019.

[20] M. Joerger, S. Stevanovic, F.-C. Chan, S. Langel, and B. Pervan, "Integrity risk and continuity risk for fault detection and exclusion using solution separation ARAIM." *26th International Technical Meeting of the Satellite Division of the Institute of Navigation, ION GNSS*, 2013.

[21] W. Maddern, G. Pascoe, M. Gadd, D. Barnes, B. Yeomans, and P. Newman, "Real-time kinematic ground truth for the Oxford RobotCar dataset," *arXiv preprint arXiv: 2002.10152*, 2020.

[22] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: an open-source SLAM system for monocular, stereo and RGB-D cameras," *IEEE Transactions on Robotics*, 2017.

[23] S. Umeyama, "Least-squares estimation of transformation parameters between two point patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1991.

[24] W. S. L. Li, Chengyao, "Visual measurement integrity monitoring for uav localization." *2019 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, 2019.