




Review

A Comprehensive Survey on the Integrity of Localization Systems

Elias Maharmeh ^{1,2,*} , Zayed Alsayed ^{1,2}  and Fawzi Nashashibi ² 

¹ Valeo Mobility Tech Center (VMTC), 6 Rue Daniel Costantini, 94000 Créteil, France; zayed.alsayed@valeo.com

² Inria-ASTRA Team, 48 Rue Barrault, 75013 Paris, France; fawzi.nashashibi@inria.fr

* Correspondence: elias.maharmeh@inria.fr

Abstract: This survey extends and refines the existing definitions of integrity and protection level in localization systems (localization as a broad term, i.e., not limited to GNSS-based localization). In our definition, we study integrity from two aspects: quality and quantity. Unlike existing reviews, this survey examines integrity methods covering various localization techniques and sensors. We classify localization techniques as optimization-based, fusion-based, and SLAM-based. A new classification of integrity methods is introduced, evaluating their applications, effectiveness, and limitations. Comparative tables summarize strengths and gaps across key criteria, such as algorithms, evaluation methods, sensor data, and more. The survey presents a general probabilistic model addressing diverse error types in localization systems. Findings reveal a significant research imbalance: 73.3% of surveyed papers focus on GNSS-based methods, while only 26.7% explore non-GNSS approaches like fusion, optimization, or SLAM, with few addressing protection level calculations. Robust modeling is highlighted as a promising integrity method, combining quantification and qualification to address critical gaps. This approach offers a unified framework for improving localization system reliability and safety. This survey provides key insights for developing more robust localization systems, contributing to safer and more efficient autonomous operations.

Keywords: integrity; protection level; localization; SLAM; fault/outlier detection; robust optimization; factor graph; fault detection and exclusion (FDE); model-based FDE; coherence-based FDE



Received: 2 December 2024

Revised: 18 December 2024

Accepted: 26 December 2024

Published: 9 January 2025

Citation: Maharmeh, E.; Alsayed, Z.; Nashashibi, F. A Comprehensive Survey on the Integrity of Localization Systems. *Sensors* **2025**, *25*, 358. <https://doi.org/10.3390/s25020358>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Integrity is a critical evaluation criterion for localization systems. It complements traditional performance metrics such as accuracy, availability, and reliability [1,2]. According to the **Federal Radionavigation Plan** [3], integrity is defined as follows: “The measure of the trust that can be placed in the correctness of the information supplied by a positioning, navigation, and timing (PNT) system. Integrity includes the ability of the system to provide timely warnings to users when the system should not be used for navigation”.

This definition highlights two key aspects of integrity: system trustworthiness and its ability to warn users of potential discrepancies. These aspects are particularly vital for high-stakes applications such as aviation and autonomous driving systems (ADS), where localization errors can have catastrophic consequences.

While this definition provides a solid foundation, new advances in localization technologies and increasing safety demands require a refinement of integrity concepts. High levels of automation, as defined by SAE [4], depend on accurate localization and guaranteed

integrity. This paper explores revised definitions and methods for integrity, as discussed in Section 4.

Localization systems are affected by various error types (Section 2.1). Methods to handle these errors include fault detection and exclusion (FDE) (Section 6) and error quantification through the Protection Level (PL) (Section 5).

The PL “represents an upper bound on the localization error” [5]. It is widely used to indicate the error level in system estimates and remains one of the most prominent integrity metrics in the literature.

Integrity is critical for safety. In the United States alone, motor vehicle crashes cause over 40,000 deaths and 2 million injuries annually [6]. Localization errors can result in incorrect navigation decisions, leading to accidents [7]. Integrity also fosters consumer confidence, essential for the widespread adoption of autonomous vehicles [8]. It assures users of system reliability and enhances efficiency in navigation, route planning, and control, especially in challenging environments such as extreme weather or limited vision.

This survey focuses on integrity challenges related to sensors and algorithms in robotic systems, including autonomous vehicles. Sensor and algorithm failures compromise localization integrity. *Where no single method can address all faults and/or error types.* Hence, multiple techniques have been developed (Sections 6 and 7).

Several existing surveys focus primarily on integrity monitoring (IM) for Global Navigation Satellite Systems (GNSSs). For example, ref. [9] discusses GNSS IM techniques, including Receiver Autonomous Integrity Monitoring (RAIM), fault detection, exclusion methods, and PL computation. Similarly, ref. [10] addresses GNSS-based IM for urban transport applications, noting challenges and open research areas compared to aviation.

Other reviews, such as [11], explore IM methods for GNSSs, INS, map-assisted, and wireless-augmented systems. It covers measurement errors, faults from various data sources, and the integration of sensors with GNSSs to improve navigation reliability. The review identifies challenges and highlights the need for advances in fault detection, exclusion, error modeling, and real-time processing. It also explores IM techniques for GNSS/INS with map-matching, discussing map/map-matching error handling and map constraints.

In contrast, our survey covers a wider range of localization methods. These include Simultaneous Localization and Mapping (SLAM), fusion-based, and optimization-based approaches. We address integrity challenges for diverse sensors such as LiDAR, cameras, HD maps, and INS. Our review emphasizes integrity in perception-based localization systems and highlights gaps in PL methods for these sensors.

Based on data from [9,10,12–16], 73.3% of the surveyed papers focus on GNSS-based integrity methods, while only 26.7% explore non-GNSS approaches. This imbalance highlights a research gap in perception-based localization systems (Sections 5–7). Figure 1 shows the distribution of surveyed papers and reveals the need for further exploration in PL methods for these systems.

To address this gap and advance the field, this paper makes the following contributions:

1. **Overview of integrity methods:** A thorough review of integrity methods for localization systems, covering sensors like LiDAR, cameras, HD maps, and INS.
2. **A new classification framework:** Introduction of a new categorization of integrity methods (Figure 2).
3. **Refined definitions:** Updated definitions of integrity and PL specific to localization systems, clarifying key concepts and metrics.
4. **In-depth review and comparative analysis:** A detailed analysis of robust modeling, PL computation techniques, and FDE methods.

5. **Detailed comparisons:** Comparisons of techniques, metrics, data types, sensors, and integrity enhancements.

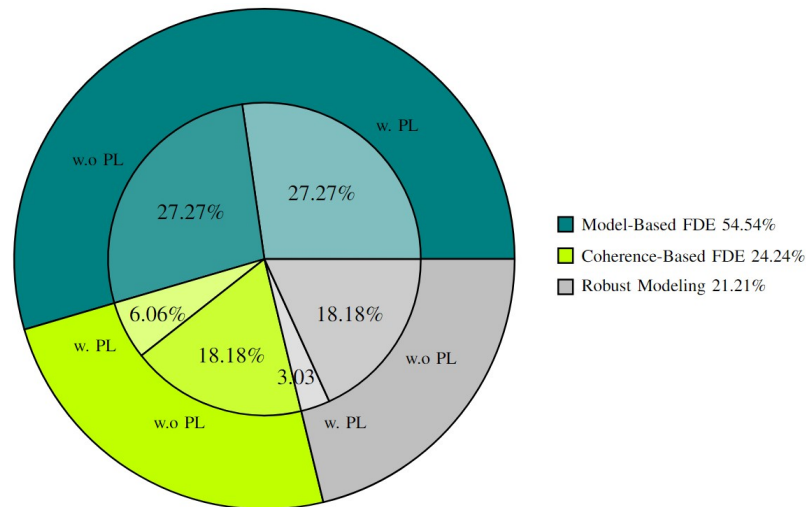


Figure 1. Percentage of surveyed literature on integrity methods, categorized as being w. (with) PL and w.o (without) PL.

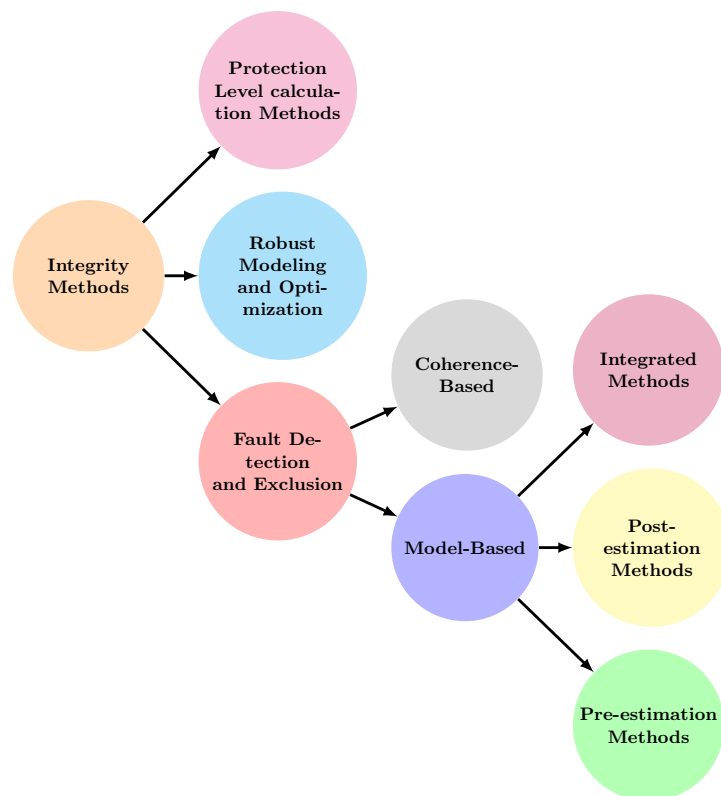


Figure 2. Classification of integrity methods.

In conclusion, integrity is a critical aspect of localization systems across various technologies, not just GNSSs. Existing research shows a lack of focus on PL methods for perception-based systems. This paper addresses this gap by providing a comprehensive framework, redefining integrity concepts, and advancing the understanding of integrity methods in localization systems.

The remainder of this paper is organized as follows. In Section 2, we introduce the key concepts related to error types and protection level parameters, providing a self-contained foundation for the subsequent sections. Section 3 offers a brief review of GNSS-

IM systems, summarizing their relevance in localization frameworks. In Section 4, we revisit and discuss various integrity definitions proposed in the literature, concluding with our proposed integrity definition, which better aligns with the objectives of this work. Similarly, Section 5 reviews and analyzes existing definitions of the protection level and culminates with our proposed definition that addresses identified limitations. Section 6 introduces fault detection and exclusion methods, categorizing them into various approaches and subcategories to highlight their roles in ensuring robust localization. Lastly, Section 7 focuses on robust modeling and optimization techniques, presenting their qualitative and quantitative aspects and demonstrating their importance in enhancing localization system performance. Together, these sections aim to provide a comprehensive understanding of integrity and robustness within localization systems while presenting our contributions and findings.

2. Background and Foundational Concepts

This section defines the vocabulary used to describe the integrity of localization systems. First of all, this section introduces various error types that localization systems encounter. Then, the different perspectives on the integrity definition that appear in the literature are discussed. Finally the section concludes with a proposed integrity definition that encompasses the different dimensions of the definitions found in the literature.

We propose the following concepts and terms to make the whole discussion self-contained and easy to follow. These terms will help newcomers to the topic of integrity and localization in general to understand the concepts more easily and clearly.

2.1. Error Types

A localization system can be affected by various error types. These can significantly compromise the integrity of the system. The error types can be classified into four main categories: *uncertainty*, *bias*, *drift*, and *outliers*, as illustrated in Figure 3.

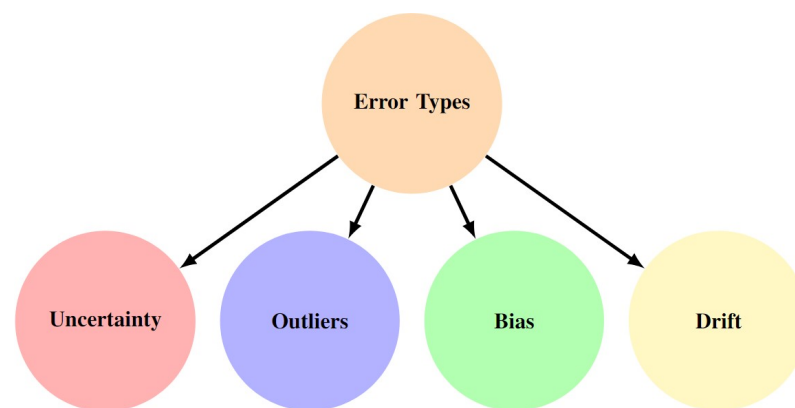


Figure 3. Classification of error types.

Each of these error types has *distinct characteristics and impacts* on the system's performance, and understanding them is crucial for developing methods to improve localization integrity.

Uncertainty, often referred to as random error, is “a short-term scattering of values around a mean value” [17]. This type of error can be expressed using a probabilistic density function, such as the Gaussian distribution. It is depicted in the left image of Figure 4, where the measurement errors are distributed symmetrically around the true value, indicating random fluctuations over time.

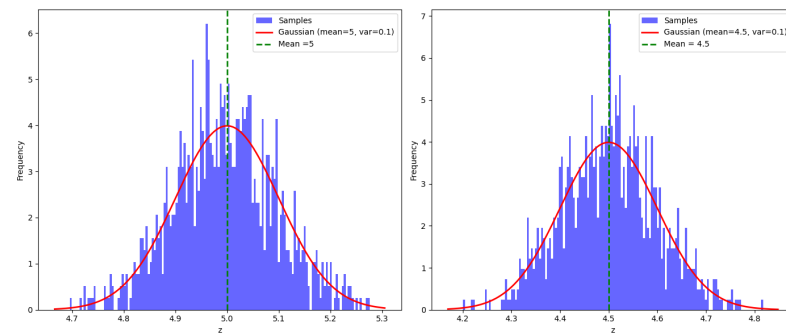


Figure 4. (Left): Histogram of LiDAR measurements at the true distance (5 m). (Right): Histogram of LiDAR measurements for the same true distance (5 m) with $b = 0.5$ m bias error (mean = 4.5 m).

Bias, or systematic error, is “a permanent deflection in the same direction from the true value” [17]. Unlike uncertainty, bias is not random but consistently skews measurements in one direction. The right image in Figure 4 shows a histogram of LiDAR measurements, where a bias of $b = -0.5$ m is added to the true value of 5 m, resulting in a shift in the mean of the measurement distribution.

Drift refers to “errors that grow slowly over time” [11], often due to cumulative sensor inaccuracies or environmental factors. These errors can cause the estimated trajectory of a vehicle to deviate progressively from the true path, as shown in Figure 5. The figure demonstrates how the estimated path, affected by drift, diverges from the true path as time progresses.

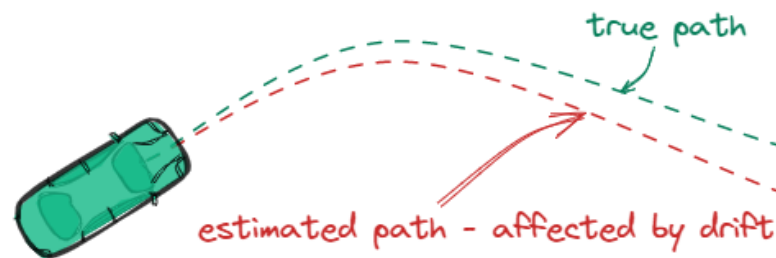


Figure 5. Due to the drift error, the vehicle’s estimated path is constantly deviating from the true path.

Outlier is “an observation that deviates so much from other observations as to arouse suspicions that it was generated by a different mechanism” [18]. This indicates that they might have been generated by a different mechanism. Outliers can occur due to sensor malfunctions, environmental disturbances, or other external factors that influence the measurement process. In the context of a LiDAR point cloud observation, an outlier is considered to be any data point that does not belong to the assumed population of true measurements. The presence of outliers is illustrated in Figure 6, where a uniform distribution of outlier values is combined with the Gaussian distribution of true values.

We adopt the generative models used in [18,19], which utilize a generic observation model to characterize the sensor behavior. We use a LiDAR sensor as an example. Each measurement, like a single LiDAR beam, is independent given the sensor’s pose. This assumption is important for the model’s simplicity and efficiency. It lets us model each beam’s measurement separately and then combine them to create a complete sensor model.

Consider a point cloud data observation from a LiDAR at time t , $Z_t = \{z_t^1, z_t^2, \dots, z_t^k, \dots, z_t^N\}$, where N is the number of LiDAR beams. This observed point cloud does not contain only the true point cloud data, $\bar{Z}_t = \{\bar{z}_t^1, \bar{z}_t^2, \dots, \bar{z}_t^k, \dots, \bar{z}_t^N\}$, but also different error types.

Starting with a single LiDAR beam, the measurement at time t , denoted as z_t^k , can be written as

$$z_t^k = \underbrace{\bar{z}_t^k}_{\text{true value}} + \underbrace{\mathbf{b}}_{\text{bias value}} + \underbrace{\mathbf{d}(t)}_{\text{drift value}} + \underbrace{\nu}_{\text{uncertainty}} \quad (1)$$

Here, z_t^k represents the raw measurement from the LiDAR sensor, which consists of several components:

- The true value \bar{z}_t^k , which is the actual distance to the target;
- The bias \mathbf{b} , which represents systematic errors that shift the measurement consistently [20–23];
- The drift $\mathbf{d}(t)$, representing errors that change over time, typically due to sensor aging or environmental influences;
- The uncertainty ν , which accounts for random noise in the measurement process.

Outliers, which are measurements that fall far outside the expected range of values, are accounted for separately in the model. In particular, the outlier distribution is modeled using a combined probability density function (PDF), which accounts for both the true measurements and the outliers. The probability of encountering an outlier is represented by δ , and the combined PDF for a LiDAR measurement is expressed as

$$p(z_t^k) = (1 - \delta) p_{\text{basic}}(z_t^k) + \delta p_{\text{outlier}}(z_t^k) \quad (2)$$

This equation combines the likelihood of the measurement being a true value with the likelihood of it being an outlier. Here,

- $p_{\text{basic}}(z_t^k)$ is the PDF for an observation, which is a PDF of the random variable z_t^k in Equation (1);
- $p_{\text{outlier}}(z_t^k)$ is outlier PDF.

The combined PDF allows us to model the probability that each measurement is either a true value or an outlier, which is important for robust localization in the presence of sensor errors.

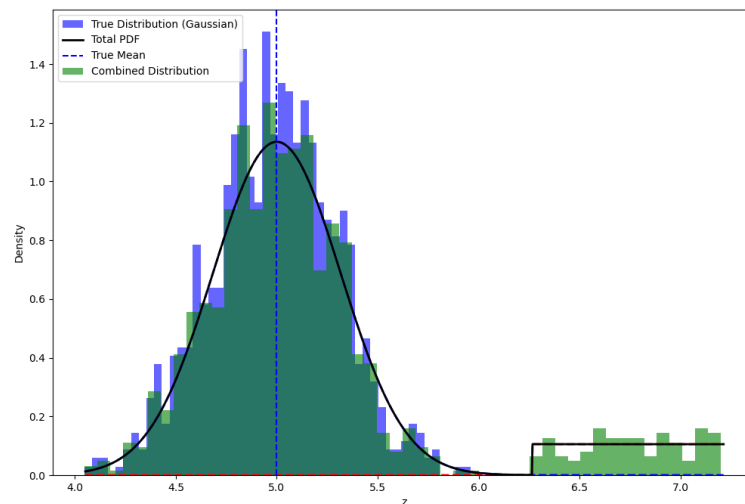


Figure 6. The true and outlier distributions along with their combined PDF. The true distribution (blue) is a Gaussian with mean 5 and variance 0.1. Outlier distribution (green) is a uniform PDF, shifted to the right. Probability of encountering an outlier δ is set to 0.1.

Introducing more information about outliers into the model will *improve its ability to handle and account for them*. This leads to more accurate results. The model's outlier handling mechanism is crucial. It ensures that the localization system can handle erroneous data points and provide accurate estimates, even with sensor faults or disturbances.

However, **no single method can handle all these error types and remove their effect on the final estimate**. Different methods are required for this, which are discussed in Sections 6 and 7.

2.2. Protection Level Related Parameters

To facilitate the understanding of the protection level concept, several related terms and vocabulary are essential. The **position error**, denoted as E , represents the difference between the estimated pose, x , and the true pose, x_{true} , i.e., $E = \|x - x_{true}\|$. In systems like RTK-GNSS, the true pose is estimated accurately, yet the localization system often lacks knowledge of the true path. The position error is typically modeled by a probability distribution, $P(E)$, which can be influenced by various error types, Section 2.1, and approximations. In an ideal scenario, assuming a linear system, Gaussian distributions, and the absence of bias, drift, and outliers, position error would be normally distributed, i.e., $E \sim \mathcal{N}(0, \Sigma_E)$.

Accuracy is defined as the degree to which the estimated position of the system approaches the actual position, as described by [10,11]. The **Alarm Limit (AL)** “represents the largest position error allowable for safe operation” [10]. When the error exceeds the AL, the localization system is deemed unsafe to rely on.

Integrity risk (IR) is the probability that the position error, E , exceeds the AL, $E > AL$, which can be expressed as

$$IR_{AL} = P(E > AL) = \int_{AL}^{-\infty} P(E = e)de \quad (3)$$

However, since the AL can change over time and in different contexts, as noted in [24], the PL is used as a more stable alternative. This leads to an updated definition of IR in terms of PL, as presented in [25–27]:

$$IR_{PL} = P(E > PL) = \int_{PL}^{-\infty} P(E = e)de \quad (4)$$

IR can be intuitively understood as the likelihood that the position error exceeds the AL, typically quantified per hour or per mile. This probability reflects the likelihood of undetected failures within the system that may lead to inaccurate or unsafe pose estimates. It is crucial for evaluating the robustness of localization systems, particularly in critical applications where safety is paramount.

Target Integrity Risk (TIR) is defined as the maximum acceptable level of IR, essentially an upper bound on IR [25–27]. This threshold is determined based on industry standards and safety requirements, and it ensures that the localization system remains safe within specified operational contexts. The IR must be continuously monitored to ensure that the system operates within acceptable limits. Additionally, the IR is calculated using system performance data, error models, and the prevailing operating conditions. The relationship between IR and TIR can be expressed as

$$IR_{PL} < TIR \quad (5)$$

Figure 7 shows how the PL is evaluated at a specific time t , given an arbitrary error distribution. For a given PL, we can compute the probability that the error is below this level, based on its probability distribution at time t . Similarly, for a given AL, we can determine the probability that the error is below the AL. This allows us to calculate the IR for the PL.

This is the forward approach. We use a given PL to check if it satisfies the integrity risk criterion. However, the main goal is to find the PL that ensures a specified integrity risk for

a given context and period. This is performed based on the error distribution at that time. Thus, we aim to determine the PL that guarantees the desired level of integrity risk.

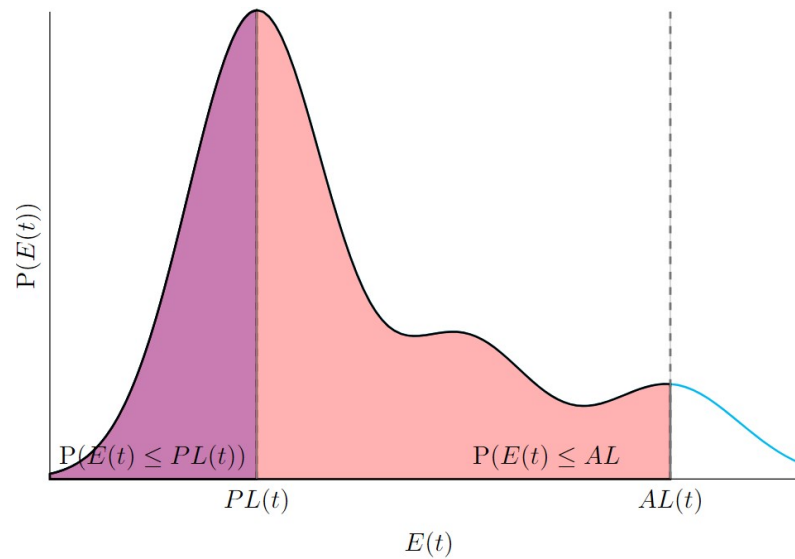


Figure 7. An arbitrary error distribution for a localization system at time t . The shaded regions indicate the probability that the error $E(t) = e$ falls within those areas.

Usually, TIR is used to refine and adjust the PL by tuning its parameters to fit the entire trajectory of the localization system. This is typically performed offline, using a learning approach with training and validation datasets, as described in [26]. The goal is to find the parameters that best fit the whole trajectory. In contrast, our focus is on estimating the PL in real time at each time step based on the current error distribution for a given IR.

3. Integrity Methods in GNSSs

Understanding integrity in localization systems is essential for many applications. Global Navigation Satellite Systems (GNSSs) provide the foundational methods for achieving integrity in localization. Building upon this foundation, this paper focuses on **extending** these integrity concepts to *perception-based* localization systems. This section presents key integrity methods used in GNSSs. It highlights fundamental techniques and concepts. **The overview is not exhaustive.** It does not cover all aspects of GNSS integrity. For a more detailed review, readers are encouraged to consult the surveys [9,10].

GNSSs use various integrity methods, mainly categorized into **Receiver Autonomous Integrity Monitoring (RAIM)** and **(PL)**.

RAIM checks the consistency of multiple satellite signals by computing the position using different subsets of satellites and comparing the results. It requires a minimum of five satellites to detect faults. However, it is typically limited to handling only one faulty satellite at a time.

Different RAIM variants use various measurements, like code or carrier measurements. They also differ in their fault detection capabilities. Advanced RAIM, for instance, can handle multiple faults more effectively compared to traditional RAIM. For a detailed comparison of measurement types and fault detection capabilities in these variants, refer to the surveys [9,10]. In this section, the results from these surveys are summarized and illustrated in Figure 8, with a focus on the RAIM variants.

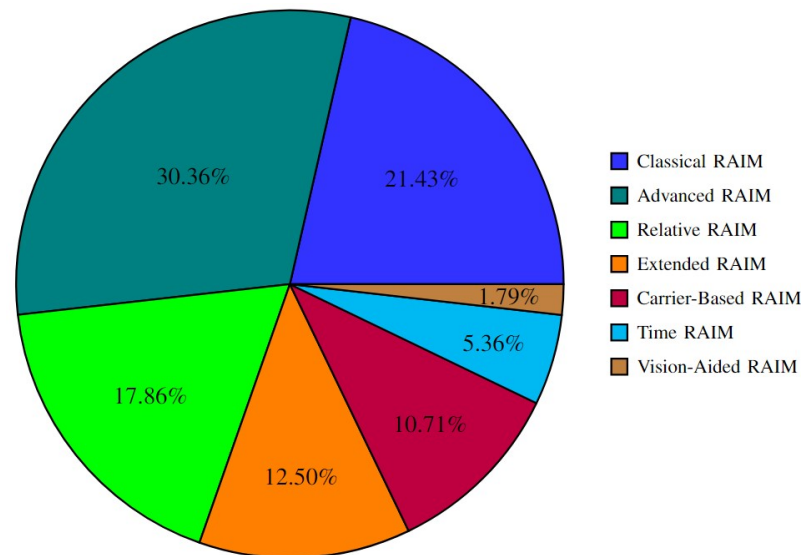


Figure 8. RAIM method classification.

PL depends on satellite-user geometry and expected pseudorange error. For example, in SBAS (Satellite-Based Augmentation System), PL is calculated as [28]:

$$PL = K\sigma \quad (6)$$

where K is an inflation constant and σ represents the confidence in the estimated position, measured in meters. Accurate PL computation requires knowledge of the distribution of residual position or range errors [29–31]. While PL has several formal definitions, discussed in Section 5, this informal description captures its essence and primary use in GNSSs.

4. Revisiting Integrity: Review, Enhancement, and New Definition

In this section, various definitions of “integrity” as presented in the literature on localization systems are reviewed and analyzed. Each definition’s approach to integrity is examined, where strengths and limitations are highlighted. It is important to note that this paper revisits the definitions of integrity, primarily in the context of non-GNSS localization systems. We begin by presenting the standard definition used in GNSS-based systems, such as the one outlined in the **Federal Radionavigation Plan**, and then analyze the integrity definitions in non-GNSSs that rely on perception sensors, such as LiDAR and cameras. This analysis seeks to complement and expand upon the existing literature in this area.

Tossaint et al. (2007) [32] define integrity for GNSS-based localization as “the system’s ability to provide warnings to the user when the system is not available for a specific operation”. Similarly, the Federal Radionavigation Plan defines integrity as “the measure of the trust that can be placed in the correctness of the information supplied by a positioning, navigation, and timing (PNT) system. Integrity includes the ability of the system to provide timely warnings to users when the system should not be used for navigation”. While these definitions emphasize providing warnings, they rely heavily on the context of specific user applications and operational requirements. The concept of integrity is reduced to the system’s ability to issue warnings, which, as a standalone metric, is vague and insufficient. Furthermore, neither definition provides a framework to quantify terms like “correctness” or “trust,” which are central to a robust understanding of integrity.

Ochieng et al. (2003) [33] and Larson (2010) [34] similarly define integrity as “the navigation system’s ability to provide timely and valid warnings to users when the system must not be used for the intended operation or phase of flight”. However, their definitions focus only on giving a warning but don’t explain much about what the problem is or how serious it

could be. That's why we need definitions that not only give warnings but also provide clear and useful (actionable) information.

Li et al. (2019, 2020) [35,36] describe integrity as *"the degree of trust that can be placed on the correctness of the localization solution, and compared it with the 3σ used in visual navigation."* While this perspective provides a quantitative approach, it is unclear whether the comparison to 3σ is limited to vision-based systems or can be generalized to other localization approaches. This lack of clarity reduces its applicability across different system types.

In contrast, AlHage et al. (2021, 2022, 2023) [25–27] define integrity as *"the ability to estimate error bounds in order to address uncertainty in the localization estimates in real time."* This definition shifts the focus toward quantifying uncertainty through error bounds, emphasizing that these bounds should include the true position. However, their work implicitly incorporates FDE methods, referred to as *"internal integrity,"* to enhance the system's overall performance. This implicit connection is not clearly articulated in their definition, which could lead to ambiguity.

Arjun et al. (2020) [37] define integrity as *"the measures of overall accuracy and consistency of data sources."* Although this definition highlights data source consistency, it does not address fault impacts or provide mechanisms for evaluating how faults affect system integrity quantitatively. As such, it lacks the necessary depth for a comprehensive understanding of system performance.

Bader et al. (2017) [38] describe integrity as *"the absence of improper system alterations"*. This definition adopts a rigid view, equating any fault with a complete loss of integrity. By failing to distinguish between the varying impacts of different faults, this perspective oversimplifies the concept. A more nuanced approach would involve quantifying fault impacts to better evaluate system integrity.

Wang et al. (2022) [39] describe integrity as *"an important indicator for ensuring the driving safety of vehicles"*. However, this definition lacks specificity, as it does not elaborate on how integrity relates to localization systems or explain its role in ensuring safety.

Quddus et al. (2006) [40] focus on a narrower context, defining integrity as *"the degree of trust that can be placed in the information provided by the map matching algorithm for each position."* This definition restricts the scope of integrity to map matching algorithms, ignoring other critical components of localization systems.

Marchand et al. (2010) [41,42], Sriramy (2021) [43], and Shubh (2023) [44] describe integrity as *"the measure of trust which can be placed in the correctness of the information supplied by the total system"*, *"the measure of trust that can be placed in the correctness of the estimated position by the navigation system"*, and *"the measure of trust that can be placed in the accuracy of the information supplied by the navigation system"*, respectively. While these definitions focus on trust and correctness, they fail to frame integrity as an evaluation criterion for the entire localization system. Vague terms like *"correctness"* and *"trust"* are left undefined, creating gaps in understanding how they relate to assessing system performance. This oversight also neglects the broader need to evaluate how well the system manages errors and deviations, which is essential for a comprehensive assessment of its integrity.

The reviewed definitions of integrity provide valuable insights but reveal several critical gaps that necessitate a more comprehensive framework:

- **Overemphasis on warnings:** Existing definitions, such as those by Tossaint et al. [32], Ochieng et al. [33], and Larson [34], focus heavily on issuing warnings without addressing broader aspects like error management or quantification.
- **Lack of quantification:** Terms like *"trust"* and *"correctness,"* central to definitions by Tossaint et al. [32], Marchand et al. [41], and others, are vague and unmeasurable, limiting their practical applicability.

- **Limited scope:** Definitions such as Quddus et al.'s [40] focus narrowly on specific components (e.g., map matching) rather than the entire localization system.
- **Fault impact and error management:** Works like AlHage et al. [25,27] implicitly address FDE but fail to clearly articulate its connection to integrity as a measurable concept.
- **Oversimplification:** Definitions like Bader et al.'s [38] equate any fault with a complete loss of integrity, oversimplifying the varied impacts of different fault types.
- **Misalignment with real-time systems:** While some works, such as AlHage et al. [27], propose real-time error estimation, they lack clarity in connecting these estimates to actionable metrics like PL.
- **Insufficient robustness considerations:** Few definitions explicitly address robustness or outlier handling, a crucial aspect of real-world localization systems.

This paper addresses these gaps by proposing a new definition of integrity that combines both qualitative and quantitative dimensions. Integrity is redefined as the system's alignment with reality, encompassing robustness, outlier handling, and deviation measurements. The proposed definition:

Definition 1 (Integrity). *Integrity refers to the quality of a system being coherent with reality.*

Definition 2 (Integrity for Localization Systems). *In the context of a localization system, integrity serves as an important evaluation criterion, encompassing both **qualifying** and **quantifying** aspects:*

- **Qualifying aspect:** *Integrity represents the system's ability to remain unaltered and effectively handle outliers and errors;*
- **Quantifying aspect:** *Integrity also involves providing an overbounding measure of **how far** the system's outputs can deviate from reality.*

*The term **how far** will be formally quantified in Section 5.*

The proposed framework evaluates robustness and reliability comprehensively. Qualitative methods focus on ensuring system reliability and managing outliers effectively, while FDE methods (Sections 6.1 and 6.2) play a key role in mitigating errors. Quantitative methods, like PL (Section 5), measure deviations between the system's outputs and reality.

New robust modeling and optimization techniques, discussed in Section 7 and illustrated in Figure 2, enhance the system's ability to handle outliers. These techniques provide probabilistic interpretations of errors, improving the assessment of localization systems.

The paper reviews integrity methods across various localization systems and sensors. It also redefines PL as a core metric for quantifying integrity, aligning it with the proposed definition.

5. Protection Level: Current Definitions and New Perspectives

In the following discussion, multiple definitions of PL found in the literature will be outlined. These definitions capture various meanings and applications of PL in the context of integrity for localization systems. Following this review, a proposed definition of PL will be presented to broaden and enhance our understanding of this crucial topic.

Li et al. (2019, 2020) [35,36] describe PL as "*the highest translational error resulting from an outlier that outlier detection systems cannot detect*". This definition is limited to translational errors and does not fully address how PL should encompass all types of uncertainties, including those from various sources beyond undetected outliers. Moreover, it fails to capture the complete error region within which the true position is guaranteed and does not consider the full scope of errors from all system components and algorithms.

Marchand et al. (2010) [41,42] define PL as “the result of a single undiscovered fault on the positioning error”. Similar to the previous definition, this one is confined to undetected faults and does not account for multiple faults or the broader uncertainty inherent in sensor measurements.

The importance of PL as “a statistical bound on position error, E , that guarantees that IR does not exceed TIR” is highlighted by AlHage et al. (2021, 2022, 2023) [25–27]. In a similar way, Arjun et al. (2020) [37] and Sriramya (2021) [43] define PL as “an error bound linked to a pre-defined risk”. While these definitions connect PL to localization system requirements, where TIR is used to check for undetected faults, they do not fully address how PL should account for all uncertainties from various system components.

Shubh (2023) [44] defines PL as “the range within which the true position lies with a high degree of confidence”, while Wang et al. (2022) [39] describe it as “an upper bound on positioning error”. Larson (2010) [34] states PL as “ensuring that position errors remain within allowable boundaries, even with faults”. While Wang’s use of “upper bound” is overly general, Larson’s focus on error boundaries relates more to system error minimization and handling and does not clearly separate PL from the system’s accuracy.

Overall, current definitions of PL provide useful insights but reveal several critical gaps:

- **Limited scope of errors considered:** Definitions by Li et al. [35,36] and Marchand et al. [41,42] focus narrowly on undetected faults, failing to account for multiple simultaneous faults or uncertainties from system components, such as sensor noise or environmental factors.
- **Lack of comprehensive uncertainty coverage:** While definitions by AlHage et al. [25–27] and Sriramya [43] connect PL to statistical bounds, they do not fully encompass uncertainties from all sources, including sensor noise, dynamic conditions, and processing errors.
- **Generalization without specificity:** Definitions by Wang et al. [39] and Larson [34] use vague terms like “upper bound”, which fail to distinguish PL as a distinct metric from accuracy or precision.
- **Inconsistent real-time relevance:** Shubh’s [44] focus on confidence lacks a connection to real-time adaptability, which is crucial for ensuring integrity in dynamic environments.
- **Separation from integrity assessment:** Many definitions fail to explicitly link PL as a core metric for evaluating and maintaining system integrity, limiting their practical applicability.

To address these gaps, the proposed definition of PL is

Definition 3 (Protection Level). *Protection Level is the real-time estimate or calculation of the error region within which the true position is guaranteed to lie.*

By assigning PL to each state estimate, the localization system can effectively adapt to changing environments, sensor conditions, and vehicle dynamics. As a result, system integrity is properly assessed and maintained in real time.

6. Fault Detection and Exclusion

FDE is crucial for enhancing the integrity of the localization system. FDE ensures accurate outputs despite faults or deviations in sensors or algorithms intended behavior. It works by identifying and removing faulty data. “Faults,” “failures,” and “outliers” often refer to deviations that can negatively impact estimation accuracy. The FDE process

addresses the qualifying aspect of integrity by making the localization resilient to various error types.

The literature distinguishes between FDE and Fault Detection and Isolation (FDI). FDE focuses on detecting and excluding anomalies to maintain integrity, **without identifying the specific error types that caused the deviation from the true value**. However, FDI aims to identify the specific cause of the problem, which is more relevant in the control engineering and software industries. For localization systems, the key objective is detecting and excluding abnormalities, regardless of their cause. Therefore, this discussion considers all approaches under the category of FDE.

Extensive literature analysis reveals two possible main categories of FDE techniques: model-based and coherence-based approaches. Model-based techniques, Figure 9, utilize mathematical models to predict the system's behavior, identifying deviations as potential faults. Coherence-based techniques, Figure 10, leverage the consistency among various sensors or measurements of the same quantity, flagging incoherent data points as potential faults.

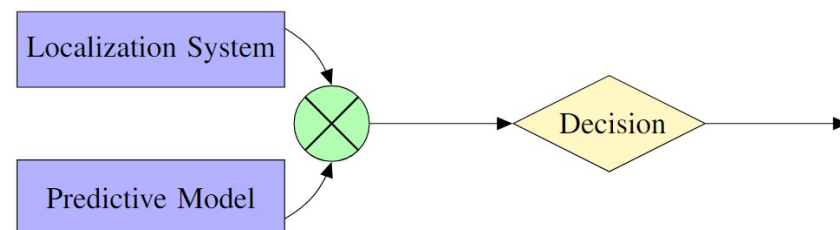


Figure 9. Model-based fault detection and exclusion: A schematic representation illustrating the integration of predictive models for system behavior and localization systems.

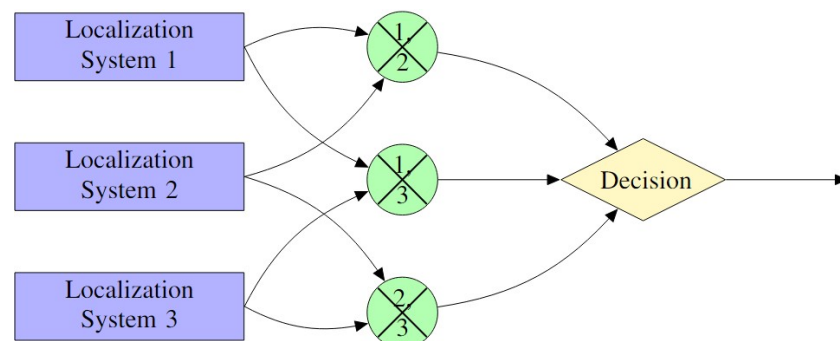


Figure 10. Coherence-based fault detection and exclusion: The figure illustrates three localization systems (1, 2, and 3) undergoing coherence checks. Comparative analyses are performed between systems 1 and 2, 2 and 3, and 1 and 3 to identify any non-coherent behavior.

The following sections explore each category in detail, including methods for computing the protection level. We will use some illustrative figures from the reviewed references. Not all figures will be included; only those that help clarify the process will be selected.

6.1. Model-Based FDE

In the field of FDE in localization and navigation systems, Model-Based FDE, or MB-FDE, is a vital component, providing reliable solutions through the use of predictive models of system behavior; see Figure 9. These predictive models could be sensor models, system models, or machine learning models like Convolutional Neural Networks (CNNs).

MB-FDE techniques identify discrepancies between expected and observed values to detect and exclude faulty data. MB-FDE techniques identify faults by analyzing discrepancies between predicted and observed values.

As an illustrative example following Figure 9, the mathematical derivation involves calculating the residuals $\mathbf{r}(t)$. These residuals represent the difference between the predicted output $\hat{\mathbf{y}}(t)$ and the observed output $\mathbf{y}(t)$:

$$\mathbf{r}(t) = \mathbf{y}(t) - \hat{\mathbf{y}}(t) \quad (7)$$

where

- $\mathbf{r}(t)$ is the residual at time t ;
- $\mathbf{y}(t)$ is the observed value at time t ;
- $\hat{\mathbf{y}}(t)$ is the predicted value based on the system model.

To determine if the discrepancy is significant enough to indicate a fault, the residuals are compared against a threshold. This threshold is typically derived from the statistical properties of the residuals, often using a Chi-square test. The Chi-square statistic quantifies the discrepancy between the residual vector and its expected distribution:

$$\chi^2 = \mathbf{r}(t)^T \mathbf{R}^{-1} \mathbf{r}(t) \quad (8)$$

where

- $\mathbf{r}(t)$ is the residual vector at time t ;
- \mathbf{R} is the covariance matrix of the residuals, which models the expected variability of the residuals under normal operating conditions.

For a properly functioning system, the Chi-square statistic follows a known distribution. The threshold γ_α is selected based on a desired confidence level, $1 - \alpha$, where α is the significance level. This corresponds to a critical value from the Chi-square distribution with appropriate degrees of freedom, typically the number of residuals.

If the calculated Chi-square statistic exceeds this critical value, the discrepancy is deemed too large to have occurred under normal conditions, indicating a fault. Then, the fault exclusion rule is

$$\gamma_\alpha = \chi_{\alpha, m}^2$$

If the test statistic satisfies $\chi^2 > \gamma_\alpha$, the measurement is flagged as faulty and excluded from the estimation process. Otherwise, the measurement is considered valid:

$$\chi^2 > \gamma_\alpha \Rightarrow \text{Fault Detected}$$

By dynamically generating the threshold based on the statistical properties of the residuals, this method ensures that the system remains robust to normal variations while being sensitive enough to detect faults. The previous example illustrates the general framework of the MB-FDE methodology. As will be illustrated in the following sections, variations in approaches within this domain arise primarily from differences in the computation of residuals and the selection of thresholds, which are often based on specific statistical distributions. The localization algorithm and input number and type affect these variations.

In the process of this review, a wide range of techniques will be examined, each of which will provide special insights for improving the integrity and PL calculation.

Based on the surveyed papers, MB-FDE is further categorized into three types:

1. Post-estimation MB-FDE (Section 6.1.1);
2. Pre-estimation MB-FDE (Section 6.1.2);
3. Integrated (or Embedded) MB-FDE (Section 6.1.3).

Tables 1–3 provides a summary of all MB-FDE methods, comparing them across various criteria. The first two tables focus on ground vehicles, while the last table covers

multi-ground vehicles and micro aerial vehicles. Each of these categories will be explained in the following sections.

Table 1. Summary of model-based fault detection and exclusion methods ¹.

Reference	Algorithm	Fault Detection	Fault Exclusion	PL	Evaluation	Data	Sensor
[25,26]	EIF	Residual calculation using Mahalanobis distance	Compare residual with Chi-square distribution	Adjust covariance for estimated error using Student t -distribution	Calculate IR and compare with TIR	Data gathered for city Rambouillet	Odometry, GNSS, camera, HD map
[27]	t -EIF	Residual calculation using Kullback–Leibler Divergence	Compare residual with Chi-square and F-distributions	Compute PL with minimum degree of freedom	Calculate IR and compare with TIR	Data gathered for town of Compiègne	Odometry, GPS, camera, HD map
[45]	Particle filter	Use selection vector to vote for faulty measurement	Exclude faulty measurements	Use GMM to calculate error probability	Compare PE with PL	Simulation and Chemnitz data	GNSS, odometry
[41,42]	EKF	Compute state residual error and compare with Chi-square distribution	Weight sensors based on residual error	Use Chi-square distribution for misdetection probability	N/A	Data acquired in urban context	Wheel speed sensors, yaw rate gyroscope, GPS
[39]	EKF	Feature-based approach	Dynamic thresholding	Use EKF error bound for error covariance	Miss detection and false alarm rate	Data acquired in urban canyons in Beijing	GNSS, INS, LiDAR
[46,47]	GraphSLAM	Test statistic computation, RANSAC	Batch test statistic computation	Perform worst-case failure slope analysis	Compare PE with PL	Data collected in alleyway of Stanford and semi-urban area of Champaign, Illinois	GPS, fish eye camera
[48]	UKF	Hotelling's T^2 test, Student t -distribution	Compare with a threshold	N/A	Compare with the standard UKF, adaptive UKF, adaptive UKF with the proposed FDE, and t -student adaptive UKF with the proposed FDE	Simulation and Highway experimental scenario	GNSS, IMU, velocity wheel sensor, steer angle, and position and azimuth using a SLAM
[34]	EKF	Parity space test	Compare residual with Chi-square distribution	Perform worst-case failure slope analysis	Compare slope of position error with respect to test statistics of parity space test	Simulation data	INS, camera
[49]	CMRNet	Outlier weighting	N/A	Cumulative distribution function of a GMM	Bound gap, false alarm rate and failure rate	KITTI visual odometry dataset [50]	Camera

¹ All vehicles are ground vehicles.

Table 2. Summary of model-based fault detection and exclusion methods ¹.

Reference	Algorithm	Fault Detection	Fault Exclusion	PL	Evaluation	Data	Sensor
[51]	RBPF + CNN	Detection of localization failure using CNN	N/A	N/A	Compare the results with AMCL ²	Simulation data and real indoor experiments	LiDAR
[52]	RBPF + CNN + LFM	Detection of localization failure using CNN	N/A	N/A	Compare the results with AMCL	Experimental environment and simulation environment	LiDAR
[53]	Free-space feature + MCL + IS	Detection of localization failure using MAE	N/A	N/A	Compare the results with AMCL	Simulation environment, robotics 2D-Laser datasets ³	LiDAR
[54]	VO-EKF	FDE using hierarchical clustering	Distance threshold between the pseudo range and the class center	N/A	Compare the result to the same system but without the FDE step	Field test Nanjing, Jiangsu, China, with the raw GNSS measurement	GPS, IMU, and binocular depth stereo camera
[55]	B-CIIF	Decision tree	Random forest	N/A	Accuracy of the decision tree and random forest	Experimental environment	Wheel encoders, IMU, LiDAR, and Marvelmind system

¹ All vehicles are ground vehicles. ² Augmented MCL [19,56,57]. ³ <https://www.ipb.uni-bonn.de/datasets/> (accessed on: 1 December 2024)

Table 3. Summary of model-based fault detection and exclusion methods ¹.

Reference	Algorithm	Fault Detection	Fault Exclusion	PL	Evaluation	Data	Sensor
[35,36]	ORB-SLAM2	Parity space test	Compare residual with Chi-square distribution	Weighted covariance for sensor noise	Compare PL with 3σ	EuRoC dataset	Camera
[58]	B-CIIF	MLP	MLP	N/A	Accuracy of the MLPs	Experimental environment	Wheel encoders, IMU, LiDAR and Marvelmind system
[59]	EIF	GKLD measure between prediction and update distribution	Use EIF bank for fault exclusion	N/A	Compare FDE with ground truth trajectory	Indoor environment	Wheel encoders, gyroscope, Kinect, LiDAR
[60]	EIF	Jensen Shannon divergence compared to Youden index of ROC curve	Signature matrix-based exclusion	Not specified	Data acquired by three Turtlebot3	Experimental environment	Wheel encoders, IMU, LiDAR, Marvelmind system

¹ All vehicles are multi-ground vehicles except for [35,36], which are micro aerial vehicles.

6.1.1. Post-Estimation MB-FDE

In the post-estimation scheme, FDE is applied after the localization system has produced an estimate, such as the pose. First, the localization algorithm performs data fusion, Bayesian updates, or optimization. Then, faults or outliers are detected. This means measurements are used as they are, known as the sensor level, and fault detection happens at the system level, like the state or pose. Therefore, detection of faults or outliers happens after the localization processing is complete; see Figure 11. The following provides an in-depth review of post-estimation FDE methods found in the literature.

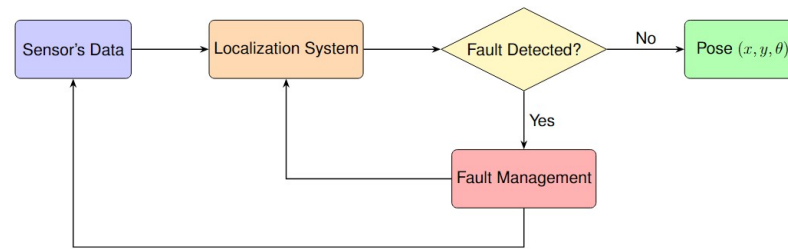


Figure 11. Post-estimation FDE process in localization systems.

In [25,26], an Extended Information Kalman Filter (EIF) is introduced. Banks of EIFs estimate the state using sensors like GNSS, cameras, and odometry. Each filter's output is compared to a main filter that combines, fuses, all outputs. Residuals, calculated as the Mahalanobis distance, were used to identify and exclude deviant filters and their sensor data.

PL is calculated by over-bounding the EIF error covariance with a Student's t -distribution. The degree of freedom for this distribution is adjusted offline during a training phase. However, this method has limitations. It assumes Gaussian noise, which doesn't accurately represent noise during faults or outliers. The thresholding also depends on this assumption, using Mahalanobis distance compared to a Chi-square distribution. Additionally, the PL calculation is adjusted offline to find the best degree of freedom for the t -distribution for a specific trajectory or scenario.

In [27], a Student t -distribution EIF (t-EIF) is utilized, akin to [25,26], but with different residual generation methods. Instead of Mahalanobis distance, it employs Kullback–Leibler Divergence (KLD) between updated and predicted distributions. Residual values adaptively adjust the t -distribution's degree of freedom, enhancing robustness against outliers. Larger residuals indicate noisy measurements, necessitating thicker tails and lower degrees of freedom, while smaller residuals justify higher degrees of freedom. This adaptation is governed by a negative exponential model, ensuring flexibility and optimization for various measurement conditions. PL calculation depends on degrees of freedom at the prediction and update steps. Errors are adjusted based on the minimum degrees of freedom between these steps. The final PL formula mirrors that of [25,26]. Figure 12 shows a general diagram of how the EIF is applied for FDE.

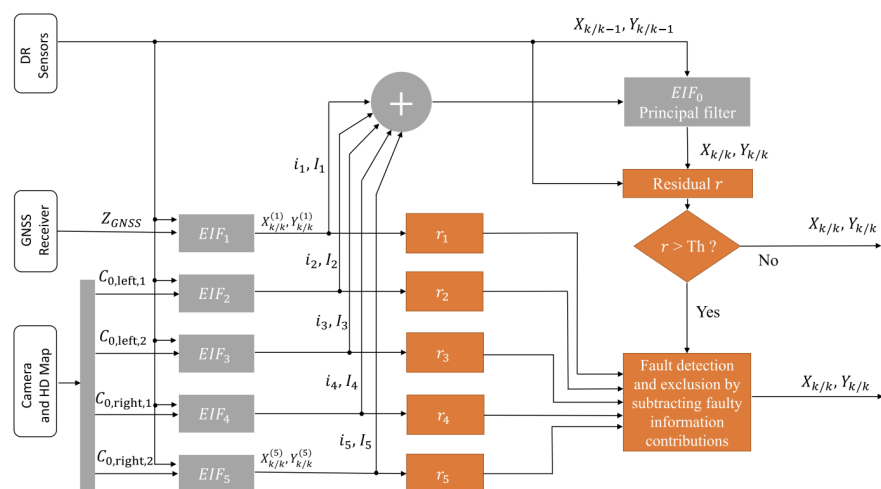


Figure 12. Multisensor fusion with FDE uses camera measurements and the HD map as inputs. In the context of the information filter for multisensor fusion: Z denotes observations from GNSS or cameras, $X = (x, y, \theta)$ is the state vector, Y is the information matrix, and y is the information vector. $I_{i,k}$ and $i_{i,k}$ represent the information contributions from observation Z_i . This figure was adopted from [26].

A multirobot system with an FDE step is addressed in [59]. The approach utilizes an EIF-based multisensor fusion system. The Global Kullback–Leibler Divergence (GKLD) between the a priori and a posteriori distributions of the EIF is computed as a residual. This residual, dependent on mean and covariance matrices, is utilized to detect and exclude faults from the fusion process. First, the GKLD is used to detect faults. Next, an EIF bank is designed to exclude faulty observations. The Kullback–Leibler Criterion (KLC) is then used to optimize thresholds in order to achieve the optimal false alarm and detection probability.

In [60], a multirobot system uses EIF for localization. Each robot performs local fault detection based on its updated and predicted states. The Jensen-Shannon divergence (JSD) is applied to generate the residuals between distributions. Fault detection thresholds are set using the Youden index from ROC curves. When a fault is detected, JSD is computed for predictions versus corrections from Gyro, Marvelmind, and LiDAR. Residuals are categorized based on their sensitivity to different error types and errors from nearby vehicles. If thresholds are exceeded, residuals are activated using a signature matrix, which helps detect and exclude faults and detects simultaneous errors. Faulty measurements are then removed from the fusion process.

In [55,58], the setup is extended with a new FDE approach and batch covariance intersection informational filter (B-CIIF). Fault detection and exclusion are based on JSD between predicted and updated states from all sensors.

In [55], a decision tree is employed for fault detection, while a random forest classifier is used for fault exclusion. Both methods use JSD residuals and a prior probability of the no-fault hypothesis for training. In contrast, ref. [58] employs two Multi-layer Perceptron (MLP) models. One for fault detection and the other for fault exclusion. The input to the MLPs includes residuals and the prior probability of the no-fault hypothesis.

Training data for these machine learning techniques include various fault categories like gyroscope drift, encoder data accumulation, Marvelmind data bias, and LiDAR errors. However, the limitation of generalizability remains significant. Since the training data are specific to certain scenarios, the models may struggle to detect and address faults in new or unfamiliar environments. This limitation can compromise the overall reliability and integrity of the system when deployed beyond the scope of the training data.

6.1.2. Pre-Estimation MB-FDE

In the pre-estimation scheme, FDE is applied before the localization estimate is made at the sensor level. Sensor measurements, such as from LiDAR, cameras, odometry, or GNSS, are first checked for faults or outliers. This means faults are detected and excluded before the data are used in the localization system, whether it involves fusion, Bayesian updates, or optimization; see Figure 13. An in-depth review of pre-estimation FDE methods from the literature is presented next.

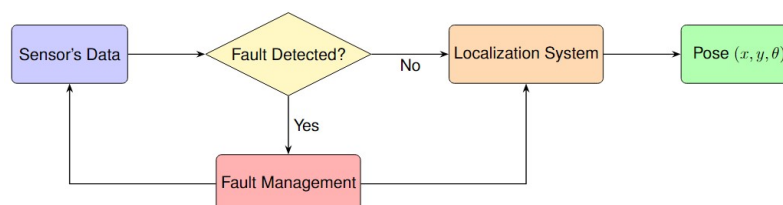


Figure 13. Pre-estimation FDE process in localization systems.

In GNSSs, ref. [45] presents a method to reduce residuals between predicted and measured pseudo-range data from satellites. They use a Gaussian Mixture Model (GMM) to handle errors that have multiple modes. Instead of a single linearization point, they use a distribution over this point, managed by a particle filter. Each particle has a vector

indicating the weight from each satellite. The particle with the highest total weight is favored; this is called the voting step. This method relies on data from multiple satellites and GNSS receiver correlations over time. If any data are missing, the method's accuracy declines. Integrity is measured by calculating the likelihood that the estimated pose exceeds the AL region. Figure 14 illustrates the framework adopted in this work.

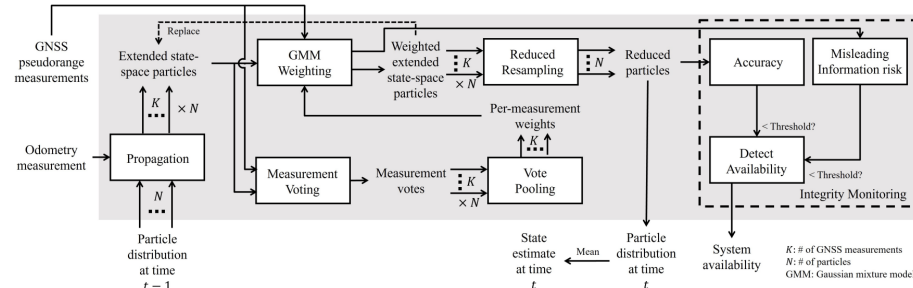


Figure 14. The framework addresses GNSS faults with GMM weighting with the voting scheme. This figure was adopted from [45].

Integrity is assessed using two metrics: hazardous operation risk and accuracy. Accuracy measures the probability that the estimated pose is outside the AL, while hazardous operation risk examines if the estimated position has at least a 50% probability of containing the true position. An alarm is triggered if either metric exceeds a set threshold, indicating a loss of integrity.

In [35,36], a vision-based localization system enhances ORB-SLAM2 with FDE techniques. It uses a parity space test to detect faults, addressing one fault at a time by comparing expected and observed measurements. Faulty features are removed iteratively until the error hits a threshold. An adaptive residual error calculation accounts for uncertainties. The modular design allows easy integration with existing SLAM techniques. PL considers noise from observations and maximum deviation from undetected faults, calculated as a weighted sum of covariance elements. Improved covariance matrix elements boost computation accuracy by removing inaccurate features or outliers iteratively. A threshold ensures a sufficient number of inliers for SLAM operations; if not met, the location estimate is deemed unsafe.

The technique used for FDE in the image-based navigation system described in [34] is similar to that of GPS RAIM [61–63], which is based on performing a parity test for FDE as in [35,36]. PL is calculated based on the maximum slope of the worst-case failure model, similar to concepts discussed in [46,47]. The author of this work uses a feature-based tracking method for localization.

Refs. [41,42] present a method to reduce errors by treating GNSS and odometry data separately. They improve state estimation by using trajectory monitoring and a short-term memory buffer instead of relying on the standard Markovian assumption. Their technique estimates states over a finite horizon and uses sensor residuals and variances to weight GNSS and odometry data.

For fault detection, they calculate sensor residuals, squared errors, and variances. They then weight GNSS and odometry data based on the ratio of each residual to its standard deviation. A Chi-square distribution threshold is used for detecting faults. The sensor with the highest weight is prioritized for exclusion.

This approach relies on hyperparameters for its operation. These include the misdetection probability for the protection level calculation and the Chi-square distribution for fault detection. Additionally, the size of the state buffer is a hyperparameter that should be selected based on the environment and scenario.

The technique presented in [39] is customized for FDE in real time within LiDAR mapping and odometry algorithms. This technique uses a feature-based sensor model to compute the mean and variance of the latest k innovations for the Extended Kalman Filter (EKF) setting. This technique adaptively establishes a threshold for fault detection by using the Chi-square distribution. Thus, the technique can adapt to environmental changes and sensor conditions. Feature-based FDE technique, dynamic thresholding, and real-time noise estimates are combined in this technique to effectively detect and exclude faults in LiDAR odometry and mapping algorithms. The technique lacks a particular formula for estimating the protection level. Its integrity is, however, evaluated by analyzing critical parameters, including error boundaries, missed detection rate, and false alarm rate. Among these parameters, the error bound stands out as a critical signal for integrity assessment, showing the maximum possible pose error.

In [48], each sensor's output is evaluated using Hotelling's T^2 test, based on expected sensor output and covariance. This allows for accurate fault detection by examining the correlation within the same sensor's data. By employing the Student t -distribution to overbound measurement noise and measurement innovation sample covariance to inflate it, where measurement noise is adaptively updated. The adaptive updating of the measurement noise covariance is similar to [39]. It takes into account faults and other outliers adaptively. This application uses a UKF localization-based methodology.

The method in [54] uses raw GNSS measurements for FDE; see Figure 15. Vehicle pose is predicted using IMU and Visual Odometry (VO). The error between this prediction and the GPS receiver pose estimate is computed. FDE is implemented using Hierarchical Clustering [64] to detect and exclude faulty satellite signals. Satellites are divided into three clusters based on estimated errors: multipath, Non-Line of Sight (NLOS), and Line of Sight (LOS) without errors. Initially, each satellite signal is considered a cluster, which is then clustered based on similarity. The three resulting clusters represent the main types of GPS errors. The LOS cluster, presumed to have the most samples, is used to compare expected pseudo-range errors. If this error exceeds a preset threshold, the associated satellite is excluded. The remaining measurements are used to calculate the GPS receiver's position and velocity. However, selecting and adjusting thresholds lacks standardization, and the method to calculate the threshold in [54] is not specified.

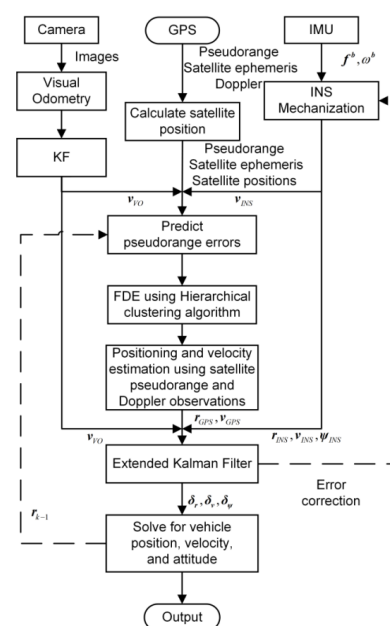


Figure 15. The algorithm framework of using hierarchical clustering for FDE. This figure was adopted from [54].

6.1.3. Integrated MB-FDE

In the integrated (or embedded) FDE scheme, FDE is incorporated within the localization process itself, suggesting that the fault detection and exclusion steps occur simultaneously with the localization algorithm, rather than as a separate phase. As sensor data are collected, they are immediately checked for faults or outliers, often using a weighting or selection method, before being used for localization. This means that the system continuously adjusts how each measurement affects the final output, such as the pose. For example, if a sensor reading is found to be unreliable, it is down-weighted or ignored during the data fusion or optimization steps. Unlike pre-estimation or post-estimation methods, which handle faults before or after main processing, integrated FDE handles faults dynamically as data are processed; see Figure 16. Next, a detailed examination of integrated FDE methods in the literature is provided.

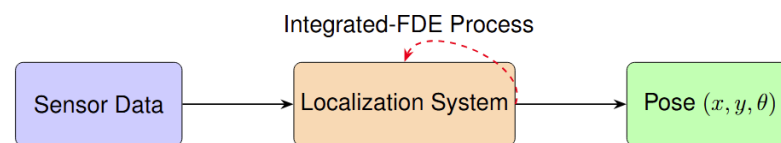


Figure 16. Integrated FDE process in localization systems.

A GraphSLAM-based FDE technique is used in [65]. GPS satellites are treated as 3D landmarks by combining GPS data with the vehicle motion model in the GraphSLAM framework; see Figure 17. The algorithm creates a factor graph using pseudo-ranges, a motion model, and broadcast signals. The difference between predicted and measured pseudo-ranges is used to compute residuals, which are weighted in the graph optimization to detect and reject faulty measurements. The sample mean and covariance of these residuals form an empirical Gaussian distribution, used for FDE. New residuals must fall within a 25% region of this distribution to update the mean and covariance. The algorithm iteratively localizes the vehicle and satellites, eliminating faulty measurements, and updates the 3D map through local and global optimization steps.

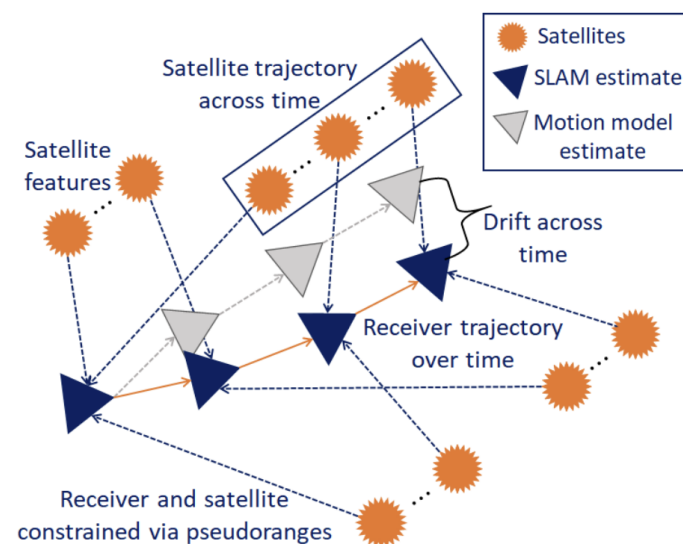


Figure 17. GraphSLAM-based FDE algorithm detects and excludes multiple GPS faults. Orange stars represent GPS satellite landmarks. Blue triangles show the GPS receiver trajectory estimated by the GraphSLAM-based FDE. Gray triangles depict the vehicle trajectory estimated using only its motion model. This figure was adopted from [65].

Building on the previous technique, the authors of [46,47] combine GPS and visual data for integrity monitoring in a GraphSLAM framework. They use GPS pseudoranges,

vision pixel intensities, vehicle motion models, and satellite ephemeris to construct a factor graph. Temporal analysis of GPS residuals and spatial analysis of vision data are performed. Superpixel-based intensity segmentation [66,67], using RANSAC [68,69], labels pixels as sky or non-sky to remove vision faults. Residuals for all inputs are included in the graph optimization, with weights to detect faults and/or outliers. GPS FDE relies on temporal correlation, while vision FDE uses spatial correlation. A batch test statistic, the sum of weighted squared residuals, is computed to assess integrity. This statistic follows a Chi-squared distribution, and the protection level is calculated using the non-centrality parameter and the worst-case failure mode slope. This approach is illustrated in Figure 18.

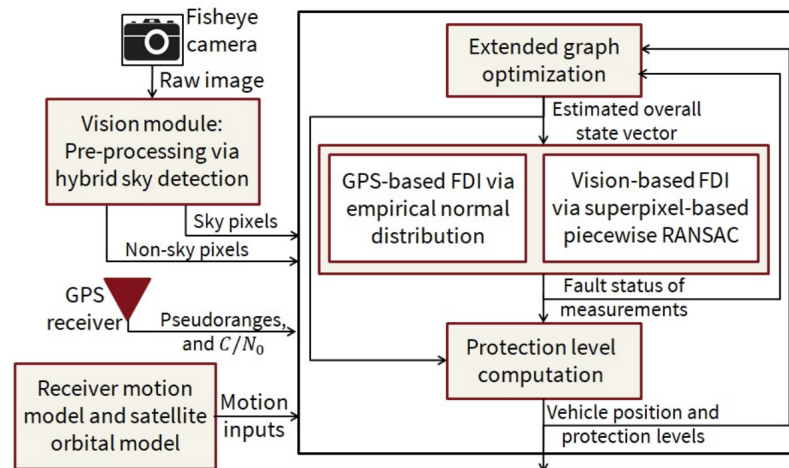


Figure 18. An illustration of the GraphSLAM-based integrity monitoring approach combining GPS and visual data. This figure was adopted from [47].

The method in [51] enhances system integrity by estimating both the robot's position and its reliability. According to [51], reliability is defined as the probability that the estimated pose error falls within an acceptable range. A modified CNN from [70] identifies localization failures by learning from successful and failed localizations. It converts CNN output into a probabilistic distribution using a Beta distribution [71], based on a reliability variable.

The conventional Dynamics Bayesian Network (DBN) model is updated with two new variables, Figure 19: the reliability variable and the CNN output. The DBN uses a Rao-Blackwellized Particle Filter (RBPF) to estimate both reliability and position. Over time, the reliability variable decays if no observations are made. To improve efficiency, ref. [52] introduces a Likelihood-Field Model (LFM) for calculating particle likelihoods. The CNN output uses a sigmoid function to indicate localization success. Positive and negative decisions are modeled with Beta and uniform distributions, respectively, with constants optimized experimentally.

Despite the LFM's efficiency benefits, it may reduce data representation, affecting failure detection accuracy. The method also faces challenges due to the computational demands of visual data analysis. Domain-specific knowledge and computational resources are needed to determine the constants for optimal performance.

In [53], a more advanced method extends pose and reliability estimation by incorporating observed sensor measurements' class. This approach includes three latent variables: localization state (successful or failed), measurement class (mapped or unmapped), and vehicle state. A modified LFM is used for observation sensors, and the proposed model integrates information about observed obstacles using conditional class probability [72].

In contrast to the previous methods [51,52], which employed a CNN to make decisions, this strategy uses a basic classifier based on the Mean Absolute Error (MAE) of the residual

errors. The residual is the difference between the observed beams and the closest obstacle in the occupancy grid map. The final decision is computed using a threshold. This method includes both global localization using the free-space feature proposed in [73] and local pose tracking using the MCL presented in [19]. As such, it can perform relocalization due to pose tracking failure. The method in [74] is used to fuse the local and global localization approaches via importance Importance Sampling (IS).

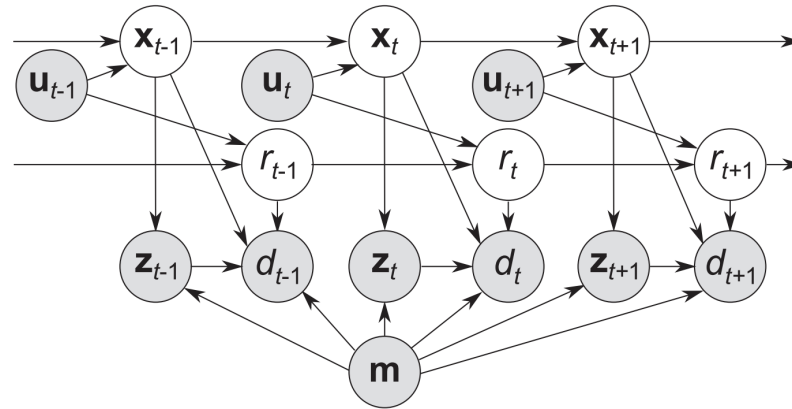


Figure 19. Graphical model for estimating both the robot's current pose x_t and the reliability, r_t , of this estimate. White nodes represent hidden variables, and gray nodes represent observable variables. The CNN uses sensor observations z_t , the map m and the pose x_t to make a decision d_t . Reliability is treated as a hidden variable and is estimated using the CNN's decision d_t and the control input u_t . This figure was adopted from [51].

In [44,49], a method is proposed to enhance camera-based localization in GNSS-limited areas by improving PL calculations. The approach leverages CNNs and 3D point cloud maps from LiDAR to estimate location error distributions, capturing both epistemic and aleatoric uncertainties [75,76].

The CNN has two components, Figure 20: one estimates position error, and the other calculates covariance matrix parameters. It uses CMRNet [77] with correlation layers [78] for error estimation and a model similar to [79] for covariance.

To handle CNN fragility, the method applies outlier weighting using robust Z-scores. A GMM is built from weighted error samples to represent position errors and calculate PL from the GMM's cumulative distribution function.

However, the method's effectiveness is limited by the need for large, high-quality training datasets, which can be labor-intensive and affect accuracy due to variability in input data and dataset quality.



Figure 20. Architecture of the deep neural network for estimating error distribution using CMRNet and correlation layers. A similar architecture, CovarianceNet, is used to produce covariance matrix parameters based on the translation error output. This figure was adopted from [49].

6.1.4. Model-Based FDE Methods: Summary and Insights

MB-FDE algorithms are promising for fault handling in localization systems but face scalability and reliability challenges. The reliance on correct sensor data, as well as the assumption of specific noise distributions, may limit their application in real-world scenarios with a wide range of environments and sensors. Practical implementation is further complicated by the computational complexity of analyzing several failure hypotheses and the combinatorial explosion in the number of measurements. Furthermore, the efficiency of these strategies is strongly reliant on the proper modeling of temporal and spatial correlations, which is not always simple or precise. Overall, while these FDE techniques are significant advances in maintaining the integrity of localization systems, more research is needed to overcome their shortcomings and increase their robustness in a variety of operating environments.

6.2. Coherence-Based Techniques

These techniques use the consistency of data from various sensors or localization systems to perform FDE. Fundamentally, coherence-based FDE, or CB-FDE, techniques take advantage of the idea that, in typical operational environments, several estimates of the same quantity should show coherence or agreement. These estimates can be acquired by different sensors, systems, or algorithms. Usually, the coherence check is carried out by weighing the estimates from each source and accepting the set of estimates that satisfy a threshold test. Alternatively, a test between each pair of estimate sources can be performed to check for inconsistencies.

To perform a coherence check, as shown in Figure 10, we calculate the pairwise residuals and use a coherence measure to identify the faulty measurement among \mathbf{y}_1 , \mathbf{y}_2 , and \mathbf{y}_3 . First, we define the residuals as the differences between each pair of measurements. These residuals are

$$\begin{aligned}\mathbf{r}_{12} &= \mathbf{y}_1 - \mathbf{y}_2, \\ \mathbf{r}_{23} &= \mathbf{y}_2 - \mathbf{y}_3, \\ \mathbf{r}_{13} &= \mathbf{y}_1 - \mathbf{y}_3.\end{aligned}\tag{9}$$

These residuals represent the discrepancies between each pair of measurements. The coherence between two measurements (or residuals) can be calculated using a similarity measure, such as the cosine similarity or correlation coefficient. For simplicity, we define the coherence between \mathbf{y}_i and \mathbf{y}_j as the normalized dot product of their residuals. The coherence between \mathbf{y}_i and \mathbf{y}_j is

$$\text{Coherence}_{ij} = \frac{\mathbf{r}_{ij}^T \mathbf{r}_{ij}}{\|\mathbf{r}_{ij}\| \|\mathbf{r}_{ij}\|}\tag{10}$$

This coherence measure ranges from 0 to 1, where 1 indicates perfect agreement between the measurements (i.e., no fault), and 0 indicates a large discrepancy. To detect a faulty measurement, we compare the pairwise coherencies. Let the coherencies be computed as

$$\begin{aligned}\text{Coherence}_{12} &= \frac{\mathbf{r}_{12}^T \mathbf{r}_{12}}{\|\mathbf{r}_{12}\| \|\mathbf{r}_{12}\|} \\ \text{Coherence}_{23} &= \frac{\mathbf{r}_{23}^T \mathbf{r}_{23}}{\|\mathbf{r}_{23}\| \|\mathbf{r}_{23}\|} \\ \text{Coherence}_{13} &= \frac{\mathbf{r}_{13}^T \mathbf{r}_{13}}{\|\mathbf{r}_{13}\| \|\mathbf{r}_{13}\|}\end{aligned}$$

Now, we apply a threshold for coherence, γ , where measurements with coherence values below this threshold indicate a faulty measurement.

The faulty measurement is detected based on its incoherence with the others. We define a decision rule to flag the faulty measurement:

$$\text{Faulty Measurement} = \begin{cases} y_1 & \text{if Coherence}_{12} < \gamma \text{ and Coherence}_{13} < \gamma \\ y_2 & \text{if Coherence}_{12} < \gamma \text{ and Coherence}_{23} < \gamma \\ y_3 & \text{if Coherence}_{23} < \gamma \text{ and Coherence}_{13} < \gamma \end{cases}$$

Here, the measurement with the lowest pairwise coherence compared to the others is flagged as the faulty one.

Once the faulty measurement is identified, it can be excluded from further processing or estimation.

$$\text{Remaining Measurements} = \begin{cases} \{y_2, y_3\} & \text{if } y_1 \text{ is faulty} \\ \{y_1, y_3\} & \text{if } y_2 \text{ is faulty} \\ \{y_1, y_2\} & \text{if } y_3 \text{ is faulty} \end{cases}$$

The above example is just one case of many variations in coherence-based methods, which lie in the way coherency checks are generally performed. These variations depend on the localization algorithm or approach and the types of input utilized. The following in-depth analysis will examine a wide range of algorithms and approaches, each providing unique perspectives and techniques for CB-FDE techniques. An innovative approach for detecting localization failures is introduced by [80]. It examines the coherency between sensor readings and the map by analyzing latent variables. The model uses Markov Random Fields (MRF) [81,82] with fully connected latent variables [83] to find misalignments. These latent variables can be aligned, misaligned, or unknown, based on residual errors between measurements and the map.

The method integrates with the localization module and uses the 3D Normal Distribution Transform (3D-NDT) scan-matching technique [84,85]. It estimates the posterior distribution of latent variables from residual errors and applies a probabilistic likelihood model. The model includes a hyperparameter and selection bias. Failure probability is approximated using sample-based methods, though the precision of this approximation is not verified. The model's multimodality, due to latent variables, may not capture all possible outcomes.

The technique from [37] integrates data from LiDAR, cameras, and maps into a unified model, as shown in Figure 21. It maintains data consistency through redundancy and weighting. The method aligns sensor data with map data using GPS positions. A Feature Grid (FG) is used to label physical areas and assign weights based on distance. This FG model overcomes limitations of traditional geometrical models by representing different features with labels and evaluating coherence between feature grids; see Figure 22.

The particle filter from [86] is adapted for map matching, creating a uniform integrity testing framework. This approach avoids specific thresholds and does not rely on particular error noise models. The PL, including the Horizontal Protection Level (HPL), is calculated using the variances of particle distributions from sensor combinations, focusing on average standard deviation.

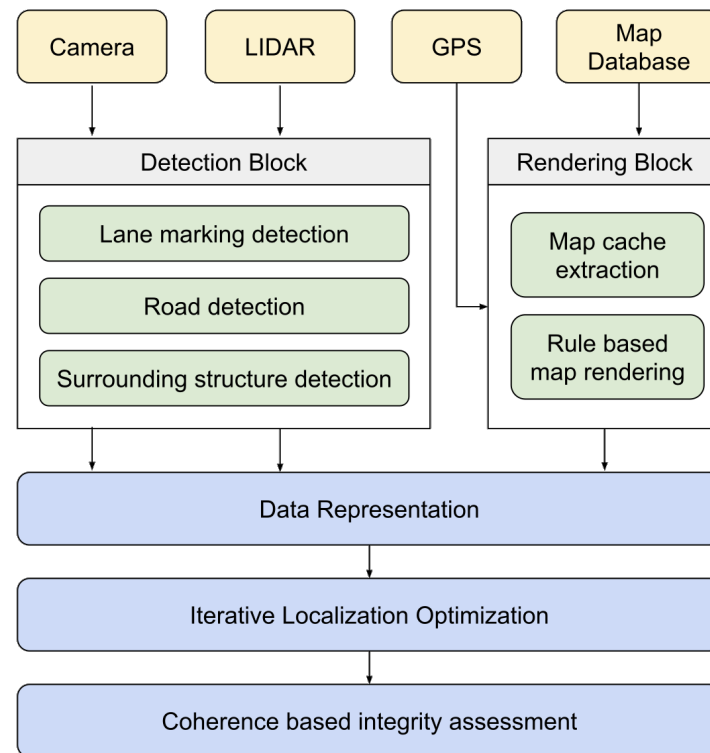


Figure 21. Framework for assessing integrity by ensuring consistency across multiple data sources. This figure was adopted from [37].

The technique in [87] uses multiple localization systems to detect and recover from faults. It combines an EKF with a Cumulative Sum (CUSUM) test [88] to detect faults and estimate the time when they occur. The EKF tracks outputs from various systems, comparing them to find deviations. The CUSUM test helps reduce false alarms by monitoring these deviations. When a fault is detected, the system uses stored sensor data for position estimation based on the fault time. This method does not need a specific fault model, simplifying implementation. However, it lacks the ability to exclude faults and relies on assumptions about system consistency, which may lead to false alarms or missed detections.

The approach in [89] uses EKF for sensor fusion in SLAM. It integrates multiple sensors, including MarvelMind for localization, gyroscopes, encoders, and LiDAR, with two EKFs: EKF-SLAM1 processes encoder and LiDAR data, while EKF-SLAM2 handles encoder and gyro data. Faults are detected by comparing Euclidean distances between EKF positions and MarvelMind data against a threshold. A large residual indicates a fault if it exceeds this threshold. Faults are identified by specific residual subsets, simplifying fault exclusion in sensors like gyroscopes and indoor GPS. When both the encoder and laser rangefinder fail, angular velocities are compared to find the faulty sensor. This method requires a global position estimate from Marvelmind to function correctly.

Research by [38] introduces a method for enhancing fault tolerance in multisensor data fusion systems. It uses two separate data fusion systems to ensure that only one fault occurs at a time. When a fault is detected, the system compares outputs from duplicated sensors and data fusion blocks. It measures the Euclidean distance between two EKF outputs and checks it against a preset threshold. If the distance is too high, it indicates a fault. The process involves two steps: comparing residuals and sensor outputs. Hardware faults are identified by comparing sensor outputs, while faulty localization systems are detected through residual comparison. The system recovers by using error-free localization values. However, setting detection thresholds can be expensive and context-specific.

In [90], a combined approach of model-based and hardware redundancy addresses drift-like failures in wheels and sensors. Model-based redundancy uses a mathematical model to mimic component behavior. The technique employs a bank of EKF and three gyroscopes, each assigned to specific faults, producing distinct residual signatures for fault detection. Residuals are used to identify faults in wheels, encoders, and gyroscopes. A fault is flagged if the residual exceeds three times its standard deviation for a set number of times. Simulations demonstrate the technique's effectiveness in detecting sensor and actuator failures. However, the initial thresholds for fault detection, based on the three-sigma rule, do not adapt well to varying conditions. This can lead to false alarms or missed detections, affecting the system's accuracy in dynamic environments.

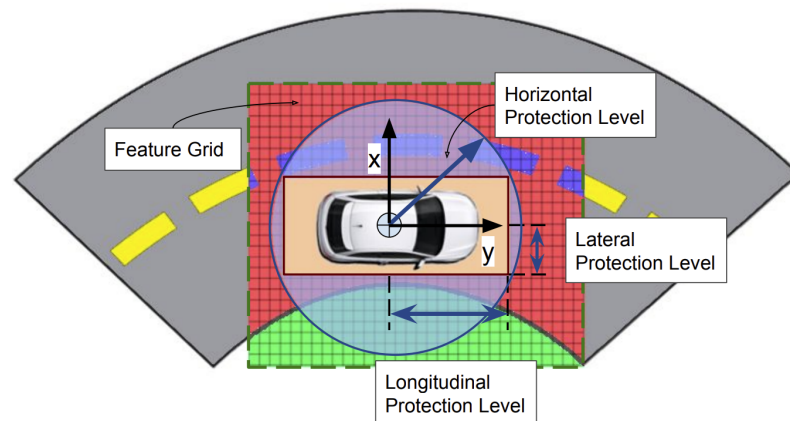


Figure 22. Feature grid representing the vehicle's localization. The Feature Grid illustrates data consistency across LiDAR, camera, and map sources. It includes the detection of road surfaces (red), lane markings (blue), other surfaces (green), and unclassified points (black). PL is indicated based on the variances of particle distributions. This figure was adopted from [37].

In [91], a process model predicts the initial pose by integrating data from stereoscopic systems, LiDAR, and GPS for vehicle localization. GPS provides the absolute position, while stereoscopic systems and LiDAR estimate ego-motion. The method uses the extended Normalized Innovation Squared (NIS) test to ensure sensor coherence before data fusion. Faulty sensor observations are removed before integration. LiDAR accuracy is improved with Iterative Closest Point (ICP) and outlier rejection. Sensor coherence is checked using the extended NIS test. An Unscented Information Filter (UIF) integrates data from multiple sensors, minimizing error accumulation. Parity relations [92], calculated with Mahalanobis distance, help detect faults.

However, this method assumes only one fault occurs at a time, which may not reflect real-world scenarios. It also assumes Gaussian sensor readings, which may not be accurate in complex situations. Moreover, the Gaussian assumption may not handle noise or outliers efficiently, potentially leading to incorrect fault detection. Accurate sensor modeling and calibration are challenging, and computational costs increase with more sensors.

In [93,94], the maximum consensus algorithm localizes the vehicle. Ref. [93] uses LiDAR data, while [94] converts 3D LiDAR point clouds into 2D images. An approximate pose is aligned with a georeferenced map point cloud. The search for the vehicle's position is limited to a predefined range, ensuring the true position is within it. Candidate positions are discretized, and a consensus set is created for each cell by counting matches between map points and sensor scan points, using a distance threshold and classical ICP cost function [95–98].

An exhaustive search finds the global optimum by identifying the transformation with the highest consensus. Despite the exponential cost with more dimensions, the search

becomes constant with fixed dimensions. The algorithm covers the entire objective function distribution, and parallel processing is facilitated by simple count operations. Real-time applications benefit from discrete optimization techniques like branch-and-bound [99,100], which speed up computations.

However, ref. [93,94] have limitations due to their simplistic counting approach. It does not fully account for the importance of correspondences in vehicle pose estimation, especially in urban environments where most matches are from the ground and facades. Ref. [94] introduces a new objective function that uses normal vectors for point-to-surface matching, improving constraints, especially longitudinally. Errors are measured using the covariance matrix of position parameters, with a smaller trace (The trace of a matrix is the sum of its diagonal elements.) indicating fewer errors. Helmert's point error, the inverse of the matrix trace, scores solution quality, guiding localization. The algorithm uses a physical beam model [19] to create a probability grid from LiDAR data, defining the PL from grid cells with a probability $p > 1 \times 10^{-7}$. A significant drawback is the grid-based search constraint, which limits precision to the resolution of the search space. Handling irregular point cloud distributions remains challenging, even with point-to-surface mapping.

Coherence-Based FDE Methods: Summary and Insights

CB-FDE techniques have a number of shortcomings. As the number of sources rises, scalability problems occur, and comparisons grow quadratically. For example, 10 sources require 45 comparisons, whereas 20 sources demand 190. This leads to increased computational complexity, making real-time problem detection difficult owing to processing delays. Additionally, these techniques rely on redundancy, which is less successful in systems with fewer sources because it requires multiple sources to provide the same value. Furthermore, as CB-FDE performs best with errors that consistently affect all sources, it is less effective with irregular error patterns.

Table 4 summarizes all MB-FDE methods and compares them using various criteria.

Table 4. Summary of coherence-based fault detection and exclusion methods ¹.

Reference	Algorithm	Coherency Check	PL	Evaluation	Data	Sensor
[37]	Particle filter-based map-matching	FG cell's weight is used to weight each source and a threshold is applied to detect the incoherent source	HPL determined by the variances of particle distributions from each sensor combination used in the localization algorithm	Compare the HPL calculated by this method with historical values in [101]	KITTI with different scenarios	Cameras, LiDAR, GPS
[80]	3D-NDT	MRF that exploits the full correlation between the sensor measurement	N/A	Root Mean Square error	SemanticKITTI dataset, and data acquired on Japanese public roads	LiDAR
[87]	EKF	Cumulative Sum test	N/A	Observing when a faulty localization system is detected	Data acquired by the vehicle	Odometry and LiDAR
[89]	EKF-SLAM	Euclidean distances between positions from the EKFs and MarvelMind are measured and compared to a predetermined threshold	N/A	Compare the obtained pose with the ground truth trajectory	Acquired data through an experimental environment	EKF1 (encoder and LiDAR), EKF2 (encoder and gyroscope), Marvelmind

Table 4. Cont.

Reference	Algorithm	Coherency Check	PL	Evaluation	Data	Sensor
[38]	EKF	Euclidean distance between the two EKF systems output is calculated and compared with a threshold	N/A	False positive rate, no detection rate, detected errors rate, etc.	Data acquired in the city of Compiègne, France; also, a simulation environment by using real data processed offline	INS, GPS, wheel sensors, gyrometer, and steering angle sensor
[90]	EKF	8 residuals are generated and the one that exceeds 3σ for a specific number of time steps is excluded	N/A	False alarm	Simulation models	Gyroscopes and wheel encoders
[91]	UIF	Extended NIS test where the sensor of increased residual is excluded	N/A	Compare with the ground truth trajectory	Real data acquired by an experimental vehicle	GPS, stereoscopic system, LiDAR
[93,94]	Maximum consensus	A consensus set for each pose candidate	Subset of grid cells that together account for a probability $p > 1 - 10^{-7}$	N/A	Measurement data were recorded in an inner-city area with a dense building structure	LiDAR

¹ All Vehicles are Ground Vehicle.

7. Robust Modeling and Optimization

FDE methods primarily focus on **qualifying** system integrity by detecting and managing faults and outliers. Qualification in this context means determining whether the system is functioning correctly by identifying when and where things go wrong. However, FDE methods often fall short in the **quantification** of integrity, which involves measuring the extent or impact of these faults. Without quantification, it is difficult to assess how errors affect the overall system performance. To address this gap, additional techniques, such as PL, are required to provide a numerical measure of system integrity.

Robust modeling and optimization techniques address both qualification and quantification. These methods do not depend on specific fault models; instead, they use general approaches to handle a wide variety of faults and noise. This allows for both a thorough assessment of whether the system is performing correctly (qualification) and a measurement of how well it is performing (quantification). By providing probabilistic interpretations of error distributions, robust methods give a more complete picture of system integrity.

In localization tasks, the common assumption that errors follow a Gaussian distribution often fails. This is not only because of linear approximations in algorithms like factor graphs and Bayesian methods but also due to other factors such as the presence of unknown outlier distributions; see Section 2.1.

These outliers can arise from various front-end [102] processes involved in building the factor graph, such as the following:

- **Image or LiDAR Scan Matching Errors** [103–105]: In odometry, mismatches in image sequences or LiDAR scans can introduce significant outliers.
- **Loop Closure Detection** [106]: In SLAM-based localization, incorrect identification of loop closures can distort the graph and lead to substantial errors.
- **Erroneous Map Queries** [107–109]: In map-based localization, errors can occur during the process of querying the map, particularly in the absence of accurate GPS data.
- **Mapping Errors** [105,110–112]: Outliers can also arise due to inaccuracies in the map itself, which may result from errors accumulated during the map generation

process. These mapping errors can propagate through the system, leading to further mismatches during map matching and adding additional outliers.

These front-end issues, if not properly handled, can weaken both the qualification and quantification of system integrity.

Robust modeling techniques are particularly valuable in these scenarios because they can manage diverse sources of error without needing detailed models of every possible fault. Unlike traditional methods, robust algorithms dynamically adapt to uncertainties in real time, which is crucial in complex, changing environments. These algorithms effectively mitigate the impact of various uncertainties, making them essential for both qualifying and quantifying system performance.

In localization, robust modeling is often applied using a factor graph model [113–115], as shown in Figure 23. In this model, a graph representation is adopted, where nodes correspond to different states or poses, and edges represent constraints between them. These constraints can come from various sources, such as the following:

- **Odometry:** Using methods like ICP from image sequences or LiDAR scans, or motion models from IMU data;
- **GPS:** Providing positional constraints based on satellite data;
- **Map Matching:** Aligning sensor data with a known map;
- **Landmarks Observations:** Constraints from observing known landmarks;
- **Calibration Parameters:** Constraints related to sensor calibration.

These constraints generate residuals, and the sum of these residuals represents the total energy or loss of the graph. The goal of optimization is to minimize this loss, thereby refining the graph's configuration to best fit the sensory information. Factor graph optimization can be performed either online, as data are received, or offline, using a batch of data.

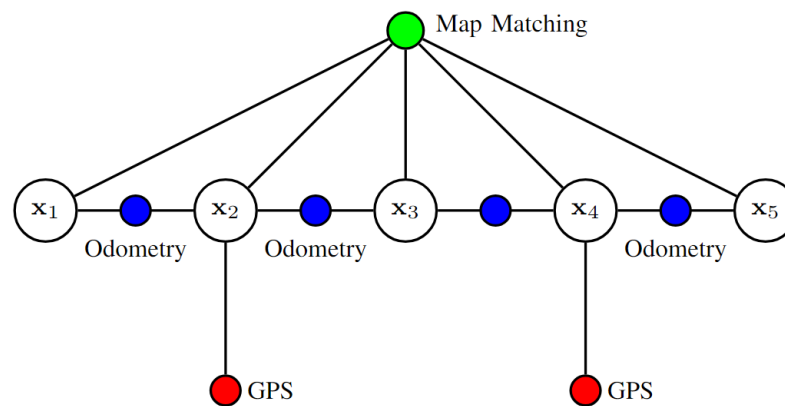


Figure 23. The factor graph illustrates the pose estimation problem with various constraints. The poses x_0 to x_5 are represented as circular nodes, connected by different types of factors, indicated by colored filled circles. Odometry constraints (blue) are binary factors linking successive poses, representing the motion model between consecutive states. GPS constraints (red) are unary factors applied to specific poses, reflecting time-dependent GPS measurements. The map matching constraint (green) is a multi-node factor connecting all poses, ensuring alignment with a known map. In this figure we ignore the prior over initial state.

To mathematically model this process, the optimization problem can be formalized as follows:

$$\mathbf{x}^* = \min_{\mathbf{x}} \sum_i \rho(\|\mathbf{r}(\mathbf{x})\|_1) \quad (11)$$

where

- $\mathbf{r}_i(\mathbf{x})$ is the residual for the i -th constraint;
- $\rho_i(\cdot)$ is a robust kernel (e.g., Huber, Cauchy) that reduces the influence of large residuals caused by outliers.

The residuals are defined generically over manifolds as the abstract difference between measurements \mathbf{z}_i and predictions $\hat{\mathbf{z}}_i(\mathbf{x})$. This abstract difference is expressed using the *boxminus* operator (\boxminus) [116] to ensure proper handling of the manifold structure. The residual for factor i is given by

$$\mathbf{r}_i(\mathbf{x}) = \hat{\mathbf{z}}_i(\mathbf{x}) \boxminus \mathbf{z}_i \quad (12)$$

where

- $\hat{\mathbf{z}}_i(\mathbf{x})$ is the predicted measurement based on the current estimate of the state \mathbf{x} ;
- \mathbf{z}_i is the observed measurement for factor i ;
- \boxminus denotes the manifold-aware difference, which accounts for the non-Euclidean nature of the state space.

Solving Equation 12 using the iteratively reweighted least squares (IRLS) [117–120]:

$$\mathbf{x}^* = \min_{\mathbf{x}} \sum_i w_i \|\mathbf{r}_i(\mathbf{x})\|_2 \quad (13)$$

where the weights w_i are defined as

$$w_i = \frac{\partial \|\mathbf{r}_i(\mathbf{x})\|_1}{\partial \mathbf{x}} \frac{\partial \rho(\|\mathbf{r}_i(\mathbf{x})\|_1)}{\partial \|\mathbf{r}_i(\mathbf{x})\|_1} \quad (14)$$

The optimization process alternates between solving the weighted least squares problem and updating the weights w_i based on the residuals $\mathbf{r}_i(\mathbf{x})$ of the current solution. This iterative process continues until convergence criteria (e.g., residual reduction or parameter update threshold) are satisfied.

By addressing both the qualification and quantification of errors, robust modeling ensures that the system not only identifies faults but also accurately measures their impact, leading to a resilient and precise localization system. This approach is particularly crucial in environments where the quality of the sensor's data can significantly influence the overall performance.

This section reviews the primary algorithms and techniques used to create robust localization algorithms.

7.1. Analysis of Robust Modeling and Optimization Techniques

Current localization methods often use least squares optimization but face challenges when dealing with outliers like data association errors and false positive loop closures.

To address these issues, ref. [121] introduced a solution that improves the back-end optimization process. Instead of keeping the factor graph structure or topology fixed, this method allows the graph topology to change dynamically during optimization. This flexibility helps the system detect and reject outliers in real time, making SLAM more robust. The method uses *switch variables*, see Figure 24, for each *potential* outlier constraint or edge. These variables help the system decide which constraints to include or exclude based on their accuracy. Essentially, switch constraints (SC) act as adjustable weights for each factor in the factor graph. These weights are optimized along the map for SLAM [121] and the pose for GNSS-based localization [122,123].

This approach is similar to FDE because it automatically identifies and removes erroneous data associations or pseudo measurements, ensuring the system uses only reliable data [121–123]. However, this approach introduced extra switch variables for each

potential outlier, which increased both the computational cost and complexity of each iteration. As a result, the system could become less efficient and slower to converge.

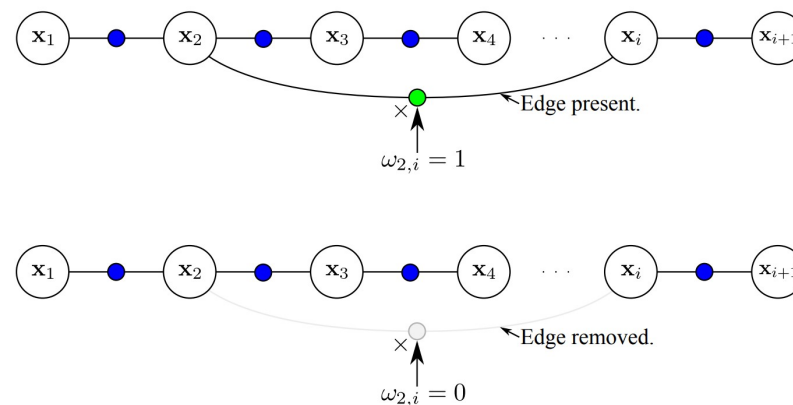


Figure 24. A binary weight $\omega_{2,j} \in \{0, 1\}$ is assigned to each loop closure constraint. When $\omega_{2,j} = 1$, the constraint remains active (top). When $\omega_{2,j} = 0$, the constraint is either disabled or removed (bottom). If these weights are **treated as variables in the optimization process**, the constraints can be adjusted or excluded during optimization. This figure was adopted from [124].

In contrast, Dynamic Covariance Scaling (DCS) [125–128] offers a more efficient method for managing outliers in SLAM without adding extra computational load. DCS adjusts the covariance of constraints based on their error terms, changing the information matrix without needing additional variables. This makes the optimization process more efficient and speeds up convergence. The scaling function in DCS is determined analytically and is related to weight functions in M -estimation [129], which reduces the number of parameters to estimate compared to the SC approach, since DCS does not require iterative optimization of the scaling function.

The earlier methods using SC and DCS had challenges with tuning the scaling function based on error and also required manual adjustment of parameters [130]. The method in [131] addresses this issue with self-tuning M -estimators. This approach directly adjusts the parameters of M -estimator cost functions, which simplifies the tuning process.

The self-tuning M -estimators method connects M -estimators with elliptical probability distributions, as introduced in [131]. This means that M -estimators can be chosen based on the assumption that errors follow an elliptical distribution. The algorithm then automatically adjusts the parameters of the M -estimators during optimization, selecting the best one based on the data's likelihood.

A broader approach to robust cost functions is introduced in [132]. This method improves algorithm performance for tasks like clustering and registration by treating *robustness as a continuous parameter*. The robust loss function in this framework can handle a wide range of probability distributions, including normal and Cauchy distributions, by using the negative log of a univariate density. **By incorporating robustness as a latent variable in a probabilistic framework, this approach automatically determines the appropriate level of robustness during optimization, which eliminates the need for manual tuning and provides a more flexible solution.** Building on this, ref. [133] presents a method that dynamically adjusts robust kernels based on the residual distribution during optimization. This dynamic tuning improves performance compared to static kernels and previous methods. The key difference from [132] is that the new method covers a wider range of probability distributions by extending the robust parameter's range. The shape of the robust kernel is controlled by a hyperparameter that adjusts in real time, enhancing both performance and robustness. Figure 25 represents various robust kernels and the

switching between them during optimization. *The switching between these kernels during optimization occurs due to the fact that robustness is modeled by an optimizable latent variable.*

In contrast, ref. [134] introduces a probabilistic approach to improve convergence. This method fits the error distribution of a sensor fusion problem using a multimodal GMM. In real-time applications like GNSS localization, the adaptive mixing method adjusts to the actual error distribution and reduces reliance on prior knowledge. This approach effectively handles non-Gaussian measurements by accurately accounting for their true distribution during estimation. In sensor fusion, where asymmetric or multimodal distributions are common, this method provides a probabilistically accurate solution. It uses a factor graph-based sensor fusion approach and optimizes the GMM adaptation with the Expectation Maximization (EM) algorithm [135,136].

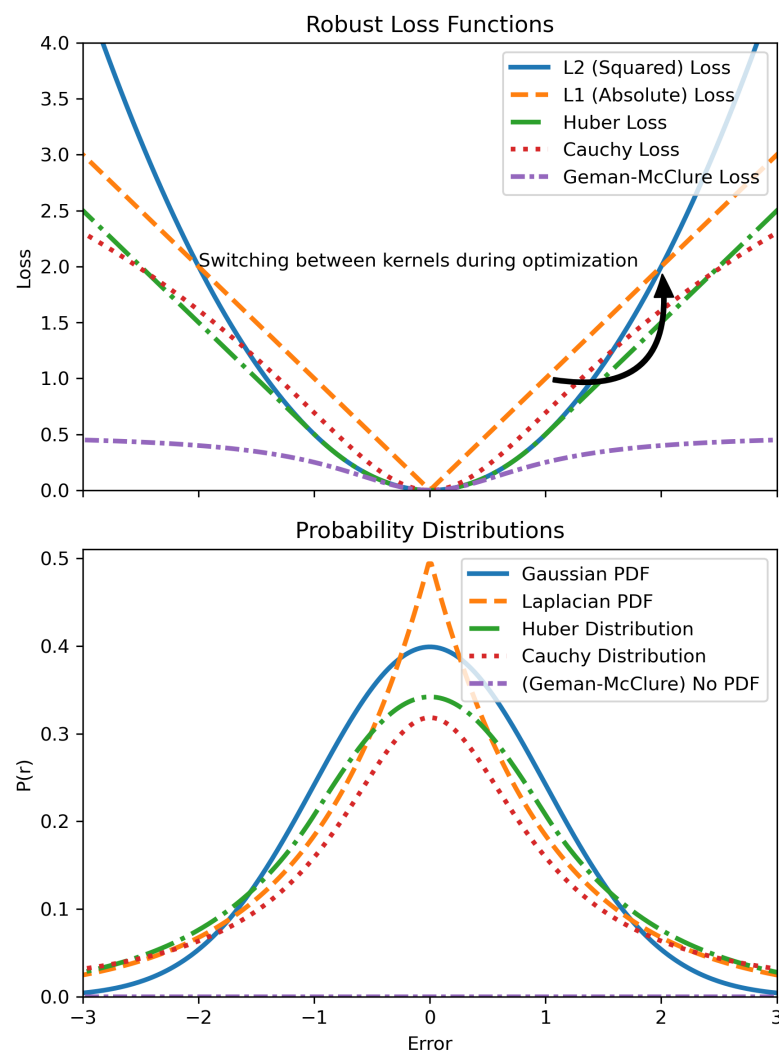


Figure 25. The top plot displays the robust loss functions: L2 (Squared) loss, L1 (Absolute) loss, Huber loss, Cauchy loss, and Geman–McClure loss, each depicted with distinct line styles. The bottom plot shows the associated probability distributions, including Gaussian PDF for L2 loss, Laplacian PDF for L1 loss, and the specific distributions for the other kernels. Notably, Geman–McClure loss has no associated probability distribution, represented by a horizontal zero line.

In [44,137], a robust multisensor state estimation method combines a Particle Filter (PF) with a robust Extended Kalman Filter (R-EKF) using RBPF. It replaces the Gaussian likelihood with robust cost functions like Huber or Tukey biweight loss [129] to better handle non-Gaussian errors and outliers unlike standard EKF methods [138,139]. The

approach uses PF for linearization points and R-EKF for state estimation, integrating estimates across points with resampling [140]. Position error bounds are estimated using a GMM and Monte Carlo integration [19], addressing orientation uncertainty and providing robust probabilistic bounds. Limitations include sensitivity to initial parameters, potential convergence to local optima, and added complexity in tuning and balancing loss terms, which can introduce biases or errors.

7.2. Robust Modeling and Optimization: Summary and Insights

In conclusion, this section underscores the importance of both FDE techniques and robust modeling approaches for enhancing the integrity of localization systems. FDE qualifies system integrity by managing faults and deviations but doesn't measure their impact on performance. Robust modeling qualifies and quantifies integrity by handling errors and providing probabilistic error bounds. Specifically, robust modeling offers two key features: it associates a probabilistic distribution over the error and dynamically accounts for variations in system uncertainty as described in [132,133]. Thus, to fully address the integrity of the localization system, integrating FDE or robust modeling with PL is essential.

Lastly, while PL quantifies integrity, the evaluation of integrity varies across applications. This typically involves computing the Integrity Risk (IR) and comparing it with the Total Integrity Risk (TIR) to determine the likelihood of the true position exceeding the provided PL, as described in Section 5.

Table 5 summarizes all the robust methods and compares them using various criteria.

Table 5. Summary of Robust Algorithms ¹.

Reference	Algorithm	PL	Evaluation	Data	Sensor
[137]	RBPF R-EKF	Compute the maximum quantile for a specific TIR	Compare with base lines such as EKF with GNSS measurements only, EKF with MD-VO ² only and EKF with GNSS measurements and MD-VO	Hong Kong UrbanNav dataset [141]	GNSS, LiDAR, camera, and IMU
[121]	Factor Graph + SC	N/A	Compare with the ground truth	Synthetic, like Manhattan, and real-world datasets, e.g., Intel	Odometry and Camera
[122]	Factor Graph + SC	N/A	Compare with the ground truth	real dataset	Odometry and GNSS receiver
[125]	Factor Graph + DCS	N/A	Compare with the results of SC [122]	Synthetic, like Manhattan, and real-world datasets, e.g., Intel	Odometry and Camera
[131]	Factor Graph + Self tuning M-Estimator	N/A	Compare the normalized squared error for different M-estimators	real data	Four monocular fish-eye cameras
[133]	ICP + Bundle Adjustment	N/A	Compare with static kernels, with [132] as well as SuMa ³ [142]	KITTI for ICP and CARLA simulator [143] for bundle adjustment	LiDAR for ICP, and camera for bundle adjustment
[134]	Factor Graph + EM	N/A	Chemnitz City and smartLoc [144] dataset	N/A	GNSS receiver and odometry

¹ All Vehicles are Ground Vehicle. ² Monocular-Depth Visual Odometry (MD-VO) ³ A dense mapping approach called Surfel-based Mapping (SuMa).

8. Conclusions

In conclusion, this survey paper presents several significant contributions to the field of integrity methods in localization systems. It identifies a crucial gap in the research on

integrity methods for non-GNSS-based systems, highlighting the need for more efforts in this area.

While 73.3% of surveyed literature focuses on GNSS-based systems, only 26.7% covers non-GNSSs that use various sensors and approaches, such as cameras, LiDAR, fusion, optimization, or SLAM. Furthermore, among these, only a small fraction specifically explores protection level calculations.

This paper introduces a unified definition of integrity that encompasses both qualitative and quantitative aspects, cf. Section 4. The new definition integrates robustness, outlier management, and deviation measures, providing a holistic evaluation of localization systems. The proposed framework improves upon existing definitions by offering a comprehensive view that includes the system's alignment with reality and detailed error handling.

The survey reviews and refines the definitions of PL, cf. Section 5. It points out that current definitions do not account for all uncertainties and limitations from system components and algorithms. The new definition of PL provided here addresses these gaps by requiring a real-time estimate of PL. This definition facilitates effective adjustment to changing environments and sensor conditions, ensuring real-time system integrity assessment.

The survey provides a detailed review of FDE methods. The FDE techniques are categorized into model-based and coherence-based approaches, examining their applications, effectiveness, and limitations. Model-based FDE methods are further divided into post-estimation, pre-estimation, and integrated-processing categories. While these methods are promising for fault handling, they face challenges related to scalability, reliability, computational complexity, model selection, and accurate modeling. Coherence-based FDE techniques also encounter issues with scalability and effectiveness, particularly with irregular error patterns.

Moreover, the paper introduces robust modeling and optimization as essential methods for integrity. Unlike traditional FDE methods that focus primarily on qualification, robust modeling addresses both qualification and quantification. It provides probabilistic error bounds and adapts to variations in system performance, offering a more comprehensive view of integrity. This approach allows for a thorough assessment of system performance and measurement of how well it is functioning.

Finally, the survey includes comparative tables that summarize and evaluate various integrity methods, highlighting their strengths and limitations. This comparative analysis provides a clearer understanding of how different methods can be applied across various localization systems.

Overall, this paper offers a valuable reference for researchers and practitioners, presenting a detailed review of integrity methods, new definitions, and a comprehensive classification framework. It sets a foundation for future research and development, aiming to enhance the safety and efficiency of localization technologies by addressing key gaps and offering a more complete understanding of integrity and protection levels.

Author Contributions: Conceptualization, E.M., Z.A. and F.N.; formal analysis, E.M.; visualization, E.M., Z.A. and F.N.; writing—original draft preparation, E.M. and Z.A.; writing—review and editing, E.M.; supervision, F.N. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Allen, M.; Baydere, S.; Gaura, E.; Kucuk, G. Evaluation of localization algorithms. In *Localization Algorithms and Strategies for Wireless Sensor Networks: Monitoring and Surveillance Techniques for Target Tracking*; IGI Global: Hershey, PA, USA, 2009; pp. 348–379.
2. Shan, X.; Cabani, A.; Chafouk, H. A Survey of Vehicle Localization: Performance Analysis and Challenges. *IEEE Access* **2023**, *11*, 107085–107107. [\[CrossRef\]](#)
3. Esper, M.; Chao, E.L.; Wolf, C.F. *2019 Federal Radionavigation Plan*; Technical Report; Department of Defense: Arlington County, VA, USA, 2020.
4. *3016-2018*; Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles. SAE International: Warrendale, PA, USA, 2018.
5. Tossaint, M.; Samson, J.; Toran, F.; Ventura-Traveset, J.; Sanz, J.; Hernandez-Pajares, M.; Juan, J. The stanford-ESA integrity diagram: Focusing on SBAS integrity. In Proceedings of the 19th International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GNSS 2006), Fort Worth, TX, USA, 26–29 September 2006; pp. 894–905.
6. Wansley, M.T. Regulating Driving Automation Safety. *Emory Law J.* **2024**, *73*, 505.
7. Patel, R.H.; Härrri, J.; Bonnet, C. Impact of localization errors on automated vehicle control strategies. In Proceedings of the 2017 IEEE Vehicular Networking Conference (VNC), Turin, Italy, 27–29 November 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 61–68.
8. Bosch, N.; Baumann, V. Trust in Autonomous Cars. In Proceedings of the I: Seminar Social Media and Digital Privacy, Toronto, ON, Canada, 19–21 July 2019.
9. Zabalegui, P.; De Miguel, G.; Pérez, A.; Mendizabal, J.; Goya, J.; Adin, I. A review of the evolution of the integrity methods applied in GNSS. *IEEE Access* **2020**, *8*, 45813–45824. [\[CrossRef\]](#)
10. Zhu, N.; Marais, J.; Bétaille, D.; Berbineau, M. GNSS position integrity in urban environments: A review of literature. *IEEE Trans. Intell. Transp. Syst.* **2018**, *19*, 2762–2778. [\[CrossRef\]](#)
11. Jing, H.; Gao, Y.; Shahbeigi, S.; Dianati, M. Integrity monitoring of GNSS/INS based positioning systems for autonomous vehicles: State-of-the-art and open challenges. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 14166–14187. [\[CrossRef\]](#)
12. de Oliveira, F.A.C.; Torres, F.S.; García-Ortiz, A. Recent advances in sensor integrity monitoring methods—A review. *IEEE Sens. J.* **2022**, *22*, 10256–10279. [\[CrossRef\]](#)
13. Hassan, T.; El-Mowafy, A.; Wang, K. A review of system integration and current integrity monitoring methods for positioning in intelligent transport systems. *IET Intell. Transp. Syst.* **2021**, *15*, 43–60. [\[CrossRef\]](#)
14. Hewitson, S.; Wang, J. GNSS receiver autonomous integrity monitoring (RAIM) performance analysis. *Gps Solut.* **2006**, *10*, 155–170. [\[CrossRef\]](#)
15. Angrisano, A.; Gagliano, S.; Gioia, C. RAIM algorithms for aided GNSS in urban scenario. In Proceedings of the 2012 Ubiquitous Positioning, Indoor Navigation, and Location Based Service (UPINLBS), Helsinki, Finland, 3–4 October 2012; IEEE: Piscataway, NJ, USA, 2012; pp. 1–9.
16. Bhattacharyya, S.; Gebre-Egziabher, D. Kalman filter-based RAIM for GNSS receivers. *IEEE Trans. Aerosp. Electron. Syst.* **2015**, *51*, 2444–2459. [\[CrossRef\]](#)
17. Hofmann, D. 48: Common Sources of Errors in Measurement Systems. In *Common Sources of Errors in Measurement Systems*; Wiley: Hoboken, NJ, USA, 2005.
18. Hawkins, D.M. *Identification of Outliers*; Springer: Berlin/Heidelberg, Germany, 1980; Volume 11.
19. Thrun, S. Probabilistic robotics. *Commun. ACM* **2002**, *45*, 52–57. [\[CrossRef\]](#)
20. Laconte, J.; Deschênes, S.P.; Labussière, M.; Pomerleau, F. Lidar measurement bias estimation via return waveform modelling in a context of 3D mapping. In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 8100–8106.
21. Holst, C.; Artz, T.; Kuhlmann, H. Biased and unbiased estimates based on laser scans of surfaces with unknown deformations. *J. Appl. Geod.* **2014**, *8*, 169–184. [\[CrossRef\]](#)
22. Naus, T. Unbiased LiDAR data measurement (draft). *Retrieved Sept.* **2008**, *20*, 2018.
23. Kodors, S. Point distribution as true quality of lidar point cloud. *Balt. J. Mod. Comput* **2017**, *5*, 362–378. [\[CrossRef\]](#)
24. Bonnifait, P. Localization Integrity for Intelligent Vehicles: How and for what? In Proceedings of the 33rd IEEE Intelligent Vehicles Symposium (IV 2022), Aachen, Germany, 5–9 June 2022.
25. Hage, J.A.; Xu, P.; Bonnifait, P.; Hage, J.A.; Xu, P.; Bonnifait, P.; Integrity, H.; With, L.; Hage, J.A.; Xu, P.; et al. High Integrity Localization with Multi-Lane Camera Measurements. In Proceedings of the 2019 IEEE Intelligent Vehicles Symposium (IV), Paris, France, 9–12 June 2019.
26. Hage, J.A.; Xu, P.; Bonnifait, P.; Ibanez-Guzman, J. Localization Integrity for Intelligent Vehicles Through Fault Detection and Position Error Characterization. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 2978–2990. [\[CrossRef\]](#)
27. Hage, J.A.; Bonnifait, P. High Integrity Localization of Intelligent Vehicles with Student's t Filtering and Fault Exclusion. In Proceedings of the 2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC), Bilbao, Spain, 24–28 September 2023; pp. 24–28. [\[CrossRef\]](#)

28. RTCA DO-229; Minimum Operational Performance Standards for Global Positioning System/Wide Area Augmentation System Airborne Equipment. RTCA: Washington, DC, USA, 1996.
29. Brown, R.G. *GPS RAIM: Calculation of Thresholds and Protection Radius Using Chi-Square Methods; A Geometric Approach*; Radio Technical Commission for Aeronautics: Washington, DC, USA, 1994.
30. Walter, T.; Enge, P. Weighted RAIM for precision approach. In *Proceedings of Ion GPS*; Institute of Navigation: Manassas, VA, USA, 1995; Volume 8, pp. 1995–2004.
31. Young, R.S.; McGraw, G.A.; Driscoll, B.T. Investigation and comparison of horizontal protection level and horizontal uncertainty level in FDE algorithms. In *Proceedings of the 9th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GPS 1996)*, Kansas City, MO, USA, 17–20 September 1996; pp. 1607–1614.
32. Tossaint, M.; Samson, J.; Toran, F.; Ventura-Traveset, J.; Hernández-Pajares, M.; Juan, J.; Sanz, J.; Ramos-Bosch, P. The Stanford–ESA integrity diagram: A new tool for the user domain SBAS integrity assessment. *Navigation* **2007**, *54*, 153–162. [[CrossRef](#)]
33. Ochieng, W.Y.; Sauer, K.; Walsh, D.; Brodin, G.; Griffin, S.; Denney, M. GPS integrity and potential impact on aviation safety. *J. Navig.* **2003**, *56*, 51–65. [[CrossRef](#)]
34. Larson, C.D. *An Integrity Framework for Image-Based Navigation Systems*. Ph.D. Thesis, Air Force Institute of Technology, Wright-Patterson AFB, OH, USA, 2010.
35. Li, C.; Waslander, S.L. Visual Measurement Integrity Monitoring for UAV Localization. In *Proceedings of the 2019 IEEE International Symposium on Safety, Security, and Rescue Robotics, SSRR 2019*, Würzburg, Germany, 2–4 September 2019; pp. 22–29. [[CrossRef](#)]
36. Li, C. *Two Methods for Robust Robotics Perception: Visual Measurement Integrity Monitoring for Localization and Confidence Guided Stereo 3D Object Detection*. Master’s Thesis, University of Toronto, Toronto, ON, Canada, 2020.
37. Balakrishnan, A.; Florez, S.R.; Reynaud, R. Integrity monitoring of multimodal perception system for vehicle localization. *Sensors* **2020**, *20*, 4654. [[CrossRef](#)]
38. Bader, K.; Lussier, B.; Schön, W. A fault tolerant architecture for data fusion: A real application of Kalman filters for mobile robot localization. *Robot. Auton. Syst.* **2017**, *88*, 11–23. [[CrossRef](#)]
39. Wang, Z.; Li, B.; Dan, Z.; Wang, H.; Fang, K. 3D LiDAR Aided GNSS/INS Integration Fault Detection, Localization and Integrity Assessment in Urban Canyons. *Remote Sens.* **2022**, *14*, 4641. [[CrossRef](#)]
40. Quddus, M.A.; Ochieng, W.Y.; Noland, R.B. Integrity of map-matching algorithms. *Transp. Res. Part C Emerg. Technol.* **2006**, *14*, 283–302. [[CrossRef](#)]
41. Le Marchand, O.; Bonnifait, P.; Ibañez-Guzmán, J.; Bétaille, D. Automotive localization integrity using proprioceptive and pseudo-ranges measurements. In *Proceedings of the Accurate Localization for Land Transportation*, Paris, France, 16 June 2009; Volume 125, pp. 7–12.
42. Le Marchand, O.; Bonnifait, P.; Ibañez-Guzmán, J.; Bétaille, D. Vehicle localization integrity based on trajectory monitoring. In *Proceedings of the 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, St. Louis, MO, USA, 10–15 October 2009; IEEE: Piscataway, NJ, USA, 2009; pp. 3453–3458.
43. Dissertation, S.B. *Risk-Resilient Gps-Based Positioning, Navigation, and Timing Using Sensor Fusion and Multi-Agent Platforms*. Ph.D. Thesis, University of Illinois at Urbana-Champaign, Champaign, IL, USA, 2021.
44. Gupta, S. *High-Integrity Urban Localization: Bringing Safety in Aviation to Autonomous Driving*. Ph.D. Thesis, Stanford University, Stanford, CA, USA, 2023.
45. Gupta, S.; Gao, G.X. Reliable GNSS Localization Against Multiple Faults Using a Particle Filter Framework. *arXiv* **2021**, arXiv:2101.06380.
46. Bhamidipati, S.; Gao, G.X. Integrity monitoring of Graph-SLAM using GPS and fish-eye camera. *NAVIGATION J. Inst. Navig.* **2020**, *67*, 583–600. [[CrossRef](#)]
47. Bhamidipati, S.; Gao, G.X. SLAM-based integrity monitoring using GPS and fish-eye camera. In *Proceedings of the Proceedings of the 32nd International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GNSS+ 2019)*, Miami, FL, USA, 16–20 September 2019; pp. 4116–4129.
48. Mori, D.; Sugiura, H.; Hattori, Y. Adaptive sensor fault detection and isolation using unscented kalman filter for vehicle positioning. In *Proceedings of the 2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, Auckland, New Zealand, 27–30 October 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 1298–1304.
49. Gupta, S.; Gao, G. Data-driven protection levels for camera and 3D map-based safe urban localization. *Navigation* **2021**, *68*, 643–660. [[CrossRef](#)]
50. Geiger, A.; Lenz, P.; Urtasun, R. Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, Providence, RI, USA, 16–21 June 2012.
51. Akai, N.; Morales, L.Y.; Murase, H. Simultaneous pose and reliability estimation using convolutional neural network and Rao–Blackwellized particle filter. *Adv. Robot.* **2018**, *32*, 930–944. [[CrossRef](#)]

52. Akail, N.; Morales, L.Y.; Murase, H. Reliability Estimation of Vehicle Localization Result. In Proceedings of the 2018 IEEE Intelligent Vehicles Symposium (IV), Changshu, China, 26–30 June 2018; Institute of Electrical and Electronics Engineers Inc.: Piscataway, NJ, USA, 2018; Volume 10; pp. 740–747. [[CrossRef](#)]
53. Akai, N. Reliable Monte Carlo localization for mobile robots. *J. Field Robot.* **2023**, *40*, 595–613. [[CrossRef](#)]
54. Wang, Y.; Sun, R. A novel gps fault detection and exclusion algorithm aided by imu and VO data for vehicle integrated navigation in urban environments. *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.* **2023**, *48*, 1147–1153. [[CrossRef](#)]
55. El Mawas, Z.; Cappelle, C.; El Najjar, M.E.B. Decision Tree based diagnosis for hybrid model-based/data-driven fault detection and exclusion of a decentralized multi-vehicle cooperative localization system. *IFAC-PapersOnLine* **2023**, *56*, 7740–7745. [[CrossRef](#)]
56. Gutmann, J.S.; Fox, D. An experimental comparison of localization methods continued. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Lausanne, Switzerland, 30 September–4 October 2002; IEEE: Piscataway, NJ, USA, 2002; Volume 1, pp. 454–459.
57. Gutmann, J.S.; Burgard, W.; Fox, D.; Konolige, K. An experimental comparison of localization methods. In Proceedings of the 1998 IEEE/RSJ International Conference on Intelligent Robots and Systems. Innovations in Theory, Practice and Applications (Cat. No. 98CH36190), Victoria, BC, Canada, 17 October 1998; IEEE: Piscataway, NJ, USA, 1998; Volume 2, pp. 736–743.
58. Mawas, Z.E.; Cappelle, C.; Najjar, M.E.B.E. Hybrid Model/data-Driven Fault Detection and Exclusion for a Decentralized Cooperative Multi-Robot System. In Proceedings of the European Workshop on Advanced Control and Diagnosis, Nancy, France, 16–18 November 2022; Springer: Cham, Switzerland, 2022; pp. 261–271.
59. Hage, J.A.; Najjar, M.E.E.; Pomorski, D. Multi-sensor fusion approach with fault detection and exclusion based on the Kullback–Leibler Divergence: Application on collaborative multi-robot system. *Inf. Fusion* **2017**, *37*, 61–76. [[CrossRef](#)]
60. El Mawas, Z.; Cappelle, C.; El Najjar, M.E.B. Fault tolerant cooperative localization using diagnosis based on Jensen Shannon divergence. In Proceedings of the 2022 25th International Conference on Information Fusion (FUSION), Linköping, Sweden, 4–7 July 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 1–8.
61. Pullen, S.; Joerger, M. GNSS integrity and receiver autonomous integrity monitoring (RAIM). In *Position, Navigation, and Timing Technologies in the 21st Century: Integrated Satellite Navigation, Sensor Systems, and Civil Applications*; Wiley: Hoboken, NJ, USA, 2020; Volume 1, pp. 591–617.
62. Liu, W.; Papadimitratos, P. Extending RAIM with a Gaussian Mixture of Opportunistic Information. In Proceedings of the Proceedings of the 2024 International Technical Meeting of The Institute of Navigation, Long Beach, CA, USA, 23–25 January 2024; 2024; pp. 454–466.
63. Li, R.; Li, L.; Jiang, J.; Du, F.; Na, Z.; Xu, X. Improved protection level for the solution-separation ARAIM based on worst-case fault bias searching. *Meas. Sci. Technol.* **2024**, *35*, 046303. [[CrossRef](#)]
64. Nielsen, F.; Nielsen, F. Hierarchical clustering. In *Introduction to HPC with MPI for Data Science*; Springer: Cham, Switzerland, 2016; pp. 195–211.
65. Bhamidipati, S.; Gao, G.X. Multiple GPS fault detection and isolation using a Graph-SLAM framework. In Proceedings of the Proceedings of the 31st International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2018), Miami, FL, USA, 24–28 September 2018; pp. 2672–2681.
66. Ren; Malik. Learning a classification model for segmentation. In Proceedings of the Ninth IEEE International Conference on Computer Vision, Nice, France, 13–16 October 2003; IEEE: Piscataway, NJ, USA, 2003; pp. 10–17.
67. Barcelos, I.B.; Belém, F.D.C.; João, L.D.M.; Patrocínio Jr, Z.K.D.; Falcão, A.X.; Guimarães, S.J.F. A comprehensive review and new taxonomy on superpixel segmentation. *ACM Comput. Surv.* **2024**, *56*, 1–39. [[CrossRef](#)]
68. Martínez-Otzeta, J.M.; Rodríguez-Moreno, I.; Mendiádua, I.; Sierra, B. Ransac for robotic applications: A survey. *Sensors* **2022**, *23*, 327. [[CrossRef](#)]
69. Derpanis, K.G. Overview of the RANSAC Algorithm. *Image Rochester* **2010**, *4*, 2–3.
70. Zagoruyko, S.; Komodakis, N. Learning to compare image patches via convolutional neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 4353–4361.
71. Gupta, A.K.; Nadarajah, S. *Handbook of Beta Distribution and Its Applications*; CRC Press: Boca Raton, FL, USA, 2004.
72. Akai, N.; Morales, L.Y.; Murase, H. Mobile robot localization considering class of sensor observations. In Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 1–5 October 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 3159–3166.
73. Millane, A.; Oleynikova, H.; Nieto, J.; Siegwart, R.; Cadena, C. Free-space features: Global localization in 2D laser SLAM using distance function maps. In Proceedings of the 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Macau, China, 3–8 November 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 1271–1277.
74. Akai, N.; Hirayama, T.; Murase, H. Hybrid localization using model-and learning-based methods: Fusion of Monte Carlo and E2E localizations via importance sampling. In Proceedings of the 2020 IEEE International Conference on Robotics and Automation (ICRA), Paris, France, 31 May–31 August 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 6469–6475.

75. Hüllermeier, E.; Waegeman, W. Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods. *Mach. Learn.* **2021**, *110*, 457–506. [\[CrossRef\]](#)
76. Sanchez, T.; Caramiaux, B.; Thiel, P.; Mackay, W.E. Deep learning uncertainty in machine teaching. In Proceedings of the 27th International Conference on Intelligent User Interfaces, Helsinki, Finland, 22–25 March 2022; pp. 173–190.
77. Cattaneo, D.; Vaghi, M.; Ballardini, A.L.; Fontana, S.; Sorrenti, D.G.; Burgard, W. Cmrnet: Camera to lidar-map registration. In Proceedings of the 2019 IEEE Intelligent Transportation Systems Conference (ITSC), Auckland, New Zealand, 27–30 October 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 1283–1289.
78. Dosovitskiy, A.; Fischer, P.; Ilg, E.; Hausser, P.; Hazirbas, C.; Golkov, V.; Van Der Smagt, P.; Cremers, D.; Brox, T. FlowNet: Learning optical flow with convolutional networks. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 2758–2766.
79. Russell, R.L.; Reale, C. Multivariate uncertainty in deep learning. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *33*, 7937–7943. [\[CrossRef\]](#) [\[PubMed\]](#)
80. Akai, N.; Akagi, Y.; Hirayama, T.; Morikawa, T.; Murase, H. Detection of Localization Failures Using Markov Random Fields with Fully Connected Latent Variables for Safe LiDAR-Based Automated Driving. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 17130–17142. [\[CrossRef\]](#)
81. Koller, D.; Friedman, N. *Probabilistic Graphical Models: Principles and Techniques*; MIT Press: Cambridge, MA, USA, 2009.
82. Jordan, M.I. *An introduction to Probabilistic Graphical Models*; University of Pittsburgh: Pittsburgh, PA, USA, 2003.
83. Bishop, C.M. Pattern recognition and machine learning. *Springer Google Sch.* **2006**, *2*, 1122–1128.
84. Magnusson, M.; Lilienthal, A.; Duckett, T. Scan registration for autonomous mining vehicles using 3D-NDT. *J. Field Robot.* **2007**, *24*, 803–827. [\[CrossRef\]](#)
85. Biber, P.; Straßer, W. The normal distributions transform: A new approach to laser scan matching. In Proceedings of the 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No. 03CH37453), Las Vegas, NV, USA, 27–31 October 2003; IEEE: Piscataway, NJ, USA, 2003; Volume 3, pp. 2743–2748.
86. Sandhu, R.; Dambreville, S.; Tannenbaum, A. Particle filtering for registration of 2D and 3D point sets with stochastic dynamics. In Proceedings of the 2008 IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, AK, USA, 23–28 June 2008; IEEE: Piscataway, NJ, USA, 2008; pp. 1–8.
87. Sundvall, P.; Jensfelt, P. Fault detection for mobile robots using redundant positioning systems. In Proceedings of the 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006, Orlando, FL, USA, 15–19 May 2006; IEEE: Piscataway, NJ, USA, 2006; pp. 3781–3786.
88. Feng, J.; Gossmann, A.; Pirracchio, R.; Petrick, N.; Pennello, G.A.; Sahiner, B. Is this model reliable for everyone? Testing for strong calibration. In Proceedings of the International Conference on Artificial Intelligence and Statistics, Valencia, Spain, 2–4 May 2024; PMLR: Westminster, UK, 2024; pp. 181–189.
89. Kellalib, B.; Achour, N.; Demim, F. Sensors Faults Detection and Isolation using EKF-SLAM for a Mobile Robot. In Proceedings of the 2019 International Conference on Advanced Electrical Engineering (ICAEE), Algiers, Algeria, 19–21 November 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 1–7.
90. Mellah, S.; Graton, G.; El Mostafa, E.; Ouladsine, M.; Planchais, A. On fault detection and isolation applied on unicycle mobile robot sensors and actuators. In Proceedings of the 2018 7th International Conference on Systems and Control (ICSC), Valencia, Spain, 24–26 October 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 148–153.
91. Wei, L.; Cappelle, C.; Ruichek, Y. Camera/laser/gps fusion method for vehicle positioning under extended nis-based sensor validation. *IEEE Trans. Instrum. Meas.* **2013**, *62*, 3110–3122. [\[CrossRef\]](#)
92. Lu, Y.; Collins, E.; Selekwa, M. *Parity Relation Based Fault Detection, Isolation and Reconfiguration for Autonomous Ground Vehicle Localization Sensors*; FAMU-FSU College of Engineering: Tallahassee, FL, USA, 2004.
93. Axmann, J.; Brenner, C. Maximum consensus localization using lidar sensors. In Proceedings of the ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences V-2-2021 (2021), Online, 5–9 July 2021; Volume 2, pp. 9–16.
94. Axmann, J.; Brenner, C. Maximum Consensus based Localization and Protection Level Estimation using Synthetic LiDAR Range Images. In Proceedings of the 2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC), Bilbao, Spain, 24–28 September 2023; IEEE: Piscataway, NJ, USA, 2023; pp. 5917–5924.
95. Besl, P.J.; McKay, N.D. Method for registration of 3-D shapes. In Proceedings of the Sensor fusion IV: Control Paradigms and Data Structures, Boston, MA, USA, 12–15 November 1991; Spie: St Bellingham, WA, USA, 1992; Volume 1611, pp. 586–606.
96. Arun, K.S.; Huang, T.S.; Blostein, S.D. Least-squares fitting of two 3-D point sets. *IEEE Trans. Pattern Anal. Mach. Intell.* **1987**, *5*, 698–700. [\[CrossRef\]](#)
97. Hähnel, D.; Burgard, W. Probabilistic matching for 3d scan registration. In Proceedings of the VDI-Conference Robotik, Ludwigsburg, Germany, 19–30 June 2002; Citeseer: Princeton, NJ, USA, 2002; Volume 2002.
98. Segal, A.; Haehnel, D.; Thrun, S. Generalized-icp. *Robot. Sci. Syst.* **2009**, *2*, 435.

99. Androulakis, I.P. MINLP: Branch and bound global optimization algorithm. In *Encyclopedia of Optimization*; Springer: Berlin/Heidelberg, Germany, 2024; pp. 1–7.
100. Das Gupta, S.; Van Parys, B.P.; Ryu, E.K. Branch-and-bound performance estimation programming: A unified methodology for constructing optimal optimization methods. *Math. Program.* **2024**, *204*, 567–639. [[CrossRef](#)]
101. European Geostationary Navigation Overlay Service (EGNOS). Position Level Specific Historical Performance. Available online: <https://egnos.gsc-europa.eu/services/safety-of-life-service/historical-performance/position-level-specific> (accessed on 1 December 2024).
102. Cadena, C.; Carlone, L.; Carrillo, H.; Latif, Y.; Scaramuzza, D.; Neira, J.; Reid, I.; Leonard, J.J. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Trans. Robot.* **2016**, *32*, 1309–1332. [[CrossRef](#)]
103. Jiang, X.; Ma, J.; Jiang, J.; Guo, X. Robust feature matching using spatial clustering with heavy outliers. *IEEE Trans. Image Process.* **2019**, *29*, 736–746. [[CrossRef](#)] [[PubMed](#)]
104. Wang, G.; Chen, Y. Robust feature matching using guided local outlier factor. *Pattern Recognit.* **2021**, *117*, 107986. [[CrossRef](#)]
105. Galstyan, T.; Minasyan, A.; Dalalyan, A.S. Optimal detection of the feature matching map in presence of noise and outliers. *Electron. J. Stat.* **2022**, *16*, 5720–5750. [[CrossRef](#)]
106. Latif, Y.; Cadena, C.; Neira, J. Robust loop closing over time for pose graph SLAM. *Int. J. Robot. Res.* **2013**, *32*, 1611–1626. [[CrossRef](#)]
107. Viswanathan, A.; Pires, B.R.; Huber, D. Vision based robot localization by ground to satellite matching in gps-denied situations. In Proceedings of the 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, Chicago, IL, USA, 14–18 September 2014; IEEE: Piscataway, NJ, USA, 2014; pp. 192–198.
108. Brubaker, M.A.; Geiger, A.; Urtasun, R. Map-based probabilistic visual self-localization. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *38*, 652–665. [[CrossRef](#)]
109. Jagadeesh, G.; Srikanthan, T.; Zhang, X. A map matching method for GPS based real-time vehicle location. *J. Navig.* **2004**, *57*, 429–440. [[CrossRef](#)]
110. Schaffrin, B.; Uzun, S. Errors-in-variables for mobile mapping algorithms in the presence of outliers. *Arch. Fotogram. Kartogr. I Teledetekci* **2011**, *22*, 377–387.
111. Maldaner, L.F.; Molin, J.P.; Spekken, M. Methodology to filter out outliers in high spatial density data to improve maps reliability. *Sci. Agric.* **2021**, *79*, e20200178. [[CrossRef](#)]
112. Zhong, Q.; Groves, D. Outlier detection for 3D-mapping-aided GNSS positioning. In Proceedings of the 35th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2022), Denver, CO, USA, 19–23 September 2022; ION: West Palm Beach, FL, USA, 2022.
113. Dellaert, F.; Kaess, M. Factor graphs for robot perception. *Found. Trends[®] Robot.* **2017**, *6*, 1–139. [[CrossRef](#)]
114. Dellaert, F. Factor graphs: Exploiting structure in robotics. *Annu. Rev. Control. Robot. Auton. Syst.* **2021**, *4*, 141–166. [[CrossRef](#)]
115. Dellaert, F. Planning and Factor Graphs. 2020. Available online: <https://dellaert.github.io/20S-8803MM/Readings/Planning.pdf> (accessed on 25 December 2024).
116. Hertzberg, C.; Wagner, R.; Frese, U.; Schröder, L. Integrating generic sensor fusion algorithms with sound state representations through encapsulation of manifolds. *Inf. Fusion* **2013**, *14*, 57–77. [[CrossRef](#)]
117. Giannelli, C.; Imperatore, S.; Kreusser, L.M.; Loayza-Romero, E.; Mohammadi, F.; Villamizar, N. A general formulation of reweighted least squares fitting. *Math. Comput. Simul.* **2024**, *225*, 52–65. [[CrossRef](#)]
118. Chen, C.; He, L.; Li, H.; Huang, J. Fast iteratively reweighted least squares algorithms for analysis-based sparse reconstruction. *Med. Image Anal.* **2018**, *49*, 141–152. [[CrossRef](#)]
119. Bosse, M.; Agamennoni, G.; Gilitchenski, I. Robust estimation and applications in robotics. *Found. Trends[®] Robot.* **2016**, *4*, 225–269. [[CrossRef](#)]
120. Zhang, Z. Parameter estimation techniques: A tutorial with application to conic fitting. *Image Vis. Comput.* **1997**, *15*, 59–76. [[CrossRef](#)]
121. Sünderhauf, N.; Protzel, P. Switchable constraints for robust pose graph SLAM. In Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, Vilamoura-Algarve, Portugal, 7–12 October 2012; IEEE: Piscataway, NJ, USA, 2012; pp. 1879–1884.
122. Sünderhauf, N.; Obst, M.; Lange, S.; Wanielik, G.; Protzel, P. Switchable constraints and incremental smoothing for online mitigation of non-line-of-sight and multipath effects. In Proceedings of the 2013 IEEE Intelligent Vehicles Symposium (IV), Gold Coast, QLD, Australia, 23–26 June 2013; IEEE: Piscataway, NJ, USA, 2013; pp. 262–268.
123. Sünderhauf, N.; Obst, M.; Wanielik, G.; Protzel, P. Multipath mitigation in GNSS-based localization using robust optimization. In Proceedings of the 2012 IEEE Intelligent Vehicles Symposium, Madrid, Spain, 3–7 June 2012; IEEE: Piscataway, NJ, USA, 2012; pp. 784–789.
124. Sünderhauf, N. Robust Optimization for Simultaneous Localization and Mapping. Ph.D. Thesis, Technische Universität Chemnitz, Chemnitz, Germany, 2012.

125. Agarwal, P.; Tipaldi, G.D.; Spinello, L.; Stachniss, C.; Burgard, W. Robust map optimization using dynamic covariance scaling. In Proceedings of the 2013 IEEE International Conference on Robotics and Automation, Karlsruhe, Germany, 6–10 May 2013; IEEE: Piscataway, NJ, USA, 2013; pp. 62–69.
126. Agarwal, P.; Tipaldi, G.; Spinello, L.; Stachniss, C.; Burgard, W. Dynamic covariance scaling for robust map optimization. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, China, 31 May–7 June 2014.
127. Agarwal, P.; Grisetti, G.; Tipaldi, G.D.; Spinello, L.; Burgard, W.; Stachniss, C. Experimental analysis of dynamic covariance scaling for robust map optimization under bad initial estimates. In Proceedings of the 2014 IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, China, 31 May–7 June 2014; IEEE: Piscataway, NJ, USA, 2014; pp. 3626–3631.
128. Agarwal, P. Robust Graph-Based Localization and Mapping. Ph.D. Thesis, Albert-Ludwigs-Universität Freiburg, Baden-Württemberg, Germany, 2015.
129. Grisetti, G.; Guadagnino, T.; Aloise, I.; Colosi, M.; Della Corte, B.; Schlegel, D. Least squares optimization: From theory to practice. *Robotics* **2020**, *9*, 51. [[CrossRef](#)]
130. Pfeifer, T.; Protzel, P. Robust sensor fusion with self-tuning mixture models. In Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 1–5 October 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 3678–3685.
131. Agamennoni, G.; Furgale, P.; Siegwart, R. Self-tuning M-estimators. In Proceedings of the 2015 IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA, USA, 26–30 May 2015; IEEE: Piscataway, NJ, USA, 2015; pp. 4628–4635.
132. Barron, J.T. A general and adaptive robust loss function. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 4331–4339.
133. Chebrolu, N.; Läbe, T.; Vysotska, O.; Behley, J.; Stachniss, C. Adaptive robust kernels for non-linear least squares problems. *IEEE Robot. Autom. Lett.* **2021**, *6*, 2240–2247. [[CrossRef](#)]
134. Pfeifer, T.; Protzel, P. Expectation-maximization for adaptive mixture models in graph optimization. In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 3151–3157.
135. Murphy, K.P. *Probabilistic Machine Learning: An Introduction*; MIT Press: Cambridge, MA, USA, 2022.
136. Murphy, K.P. *Probabilistic Machine Learning: Advanced Topics*; MIT Press: Cambridge, MA, USA, 2023.
137. Gupta, S.; Mohanty, A.; Gao, G. Getting the best of particle and Kalman filters: GNSS sensor fusion using rao-blackwellized particle filter. In Proceedings of the 35th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2022), Denver, CO, USA, 19–23 September 2022; pp. 1610–1623.
138. Bailey, T.; Durrant-Whyte, H. Simultaneous localization and mapping (SLAM): Part II. *IEEE Robot. Autom. Mag.* **2006**, *13*, 108–117. [[CrossRef](#)]
139. Whyte, H.D. Simultaneous localisation and mapping (SLAM): Part I the essential algorithms. *IEEE Robot. Autom. Mag.* **2006**, *13*, 99–110. [[CrossRef](#)]
140. Fox, D.; Thrun, S.; Burgard, W.; Dellaert, F. Particle filters for Mobile robot localization. In *Sequential Monte Carlo Methods in Practice*; Springer: Berlin/Heidelberg, Germany, 2001; pp. 401–428.
141. Hsu, L.T.; Kubo, N.; Wen, W.; Chen, W.; Liu, Z.; Suzuki, T.; Meguro, J. UrbanNav: An open-sourced multisensory dataset for benchmarking positioning algorithms designed for urban areas. In Proceedings of the 34th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2021), St. Louis, MI, USA, 20–24 September 2021; pp. 226–256.
142. Behley, J.; Stachniss, C. Efficient Surfel-Based SLAM using 3D Laser Range Data in Urban Environments. *Robot. Sci. Syst.* **2018**, *2018*, 59.
143. Dosovitskiy, A.; Ros, G.; Codevilla, F.; Lopez, A.; Koltun, V. CARLA: An open urban driving simulator. In Proceedings of the Conference on Robot Learning, Mountain View, CA, USA, 13–15 November 2017; PMLR: New York, NY, USA, 2017; pp. 1–16.
144. Reisdorf, P.; Pfeifer, T.; Breßler, J.; Bauer, S.; Weissig, P.; Lange, S.; Wanielik, G.; Protzel, P. The problem of comparable gnss results—an approach for a uniform dataset with low-cost and reference data. In Proceedings of the Advances in Vehicular Systems, Technologies and Applications (VEHICULAR), Barcelona, Spain, 13–17 November 2016.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.