Failure Detection for Laser-based SLAM in Urban and Peri-Urban Environments

Zayed Alsayed, Guillaume Bresson, Anne Verroust-Blondet, Fawzi Nashashibi

Failure Detection for Laser-based SLAM in Urban and Peri-Urban Environments

Zayed Alsayed*†, Guillaume Bresson*, Anne Verroust-Blondet† and Fawzi Nashashibi†

*Institut VEDECOM
Versailles, France
firstname.name@vedecom.fr

†Inria Paris
Paris, France
firstname.name@inria.fr

***Abstract*—Simultaneous Localization And Mapping (SLAM) is considered as one of the key solutions for making mobile robots truly autonomous. Based mainly on perceptive information, the SLAM concept is assumed to solve localization and provide a map of the surrounding environment simultaneously. In this paper, we study SLAM limitations and we propose an approach to detect *a priori* potential failure scenarios for 2D laser-based SLAM methods. Our approach makes use of raw sensor data, which makes it independent of the underlying SLAM implementation, to extract a relevant descriptors vector. This descriptors vector is then used together with a decision-making algorithm to detect failure scenarios. Our approach is evaluated using different decision algorithms through three realistic experiments.**

I. Introduction

The Simultaneous Localization And Mapping (SLAM) concept is widely considered to be the solution for making mobile robots truly autonomous. The idea behind SLAM is for a mobile robot to be able to incrementally build a map of its surroundings while estimating its pose (position and orientation) within this map. Since the formulation of its theoretical concept and its probabilistic solution, SLAM has been quickly adopted and developed by the scientific community. Various implementations of SLAM have been proposed, based on a variety of perception-sensors and applied to different fields for different types of platforms.

SLAM algorithms are designed to handle robot localization based on features extracted from the environment, also called landmarks, which are assumed to satisfy certain criteria, such as being unique, static and of sufficient number to estimate the robot's relative displacement.

If there appears to be an insufficient number of landmarks, it is put down to the fact that landmarks had been badly defined - which is not necessarily the case as the environment need not be limited to only one configuration. Autonomous cars could navigate in a wide range of scenarios where we must not be limited to exploiting a single or one set of specific types of information. In other words, the type of information content in the environment can change according to the local scenario.

Furthermore, SLAM approaches based on raw data, such as maximum-likelihood SLAM, which consider raw data, without extracting specific features, cannot avoid failure in non-salient (ambiguous) configurations, for example the tunnel passage scenario, where the environment configuration does not propose a unique/distinctive configuration to estimate the displacement. These ambiguous configurations also apply to landmark-based approaches.

Work done on SLAM generally focuses on combining other positioning information with SLAM in different ways, although to our knowledge, specifically detecting these odd scenarios a priori in order to ensure the operability of SLAM independently of any other information sources has not yet been explored. Work on failure detection has been carried out, although it focuses on comparing different sources of information rather than studying the limitations of the techniques themselves.

In this paper we propose an approach to detect potential SLAM failure scenarios based only on information perceived from the environment (SLAM input) only. Our approach is designed for 2D laser-based SLAM methods.

The rest of this paper is organized as follows: Section II presents the state of the art regarding SLAM algorithms and Fault Detection and Isolation (FDI) systems applied to localization. Then, Section III introduces limitations of the SLAM concept, illustrated by examples of 2D laser-based SLAM. Section IV presents our method to detect possible failure of laser-based approaches by processing laser data. Finally, Section V presents the results of our experiments and we conclude by giving some perspectives in Section VI.

II. State of the art

Initial work on SLAM led to the foundation of a theoretical solution [1][2][3], although many issues emerged from practical implementations, and various methods have been proposed. These methods can essentially be classified according to the map representation, the estimation processes or the type of sensor used.

The first SLAM approaches used landmark-map representations and were based on Extended Kalman Filters [2] and Particle Filters [4]. Other solutions used grid map representations [5][6][7] and were based on Likelihood Maximization. In these different approaches, perception data (the input of SLAM) are implicitly supposed to be static and unique in order to allow SLAM to operate properly.

The static world assumption is handled using strategies called SLAM-DATMO [8][9][7], which integrate mobile object detection and tracking together with the SLAM process. Another trend is based on the transferable belief model [10]

where the map representation makes it possible to deal directly with ambiguous information (dynamic/static). This method does not attempt to detect or track mobile objects, rather it treats them as ambiguous information to be very low weighted while maximizing the likelihood.

Error accumulation can be dealt with using a backward correction algorithm such as graph-SLAM [11], which rectifies the previous estimations when detecting loop-closing. Such approaches can correct drift caused by longterm operation and therefore reduce ambiguity, but require more information from the environment before rectifying or recovering from errors.

Generally, Fault Detection and Isolation (FDI) systems exploit information redundancy and measure the coherence of different sources together with a motion model for prediction to detect faults or failure of the sources. In [12] the authors analyze the residuals at the output of a bank of Kalman filters by thresholding, while in [13] they use a Neural Network to detect failure. In [14] the authors provide some improvements to the method by measuring the coherence of different estimations. The authors of [15] combine fault detection with outlier detection based on the Normalized Innovation Squared (NIS) test. [16] use neural networks as approximators to predict the future state, and to detect failure. Their approach is designed for food transportation applications.

Wei et al.[17] present a fusion method for redundant localization information in order to build a consistent localization system. They start by a selection and validation step based on measuring the information coherence from every sensor, then the filtering is done using KF-like techniques (EKF, UKF, IF, UIF). Although promising, the evaluation method used does not give a clear idea about the accuracy of the method.

Localization could be achieved based on several techniques depending on each algorithm's limitations and operating range. Bresson et al. [18] present a cooperative system based on a selection between several localization techniques, where the operating range of each algorithm is defined a priori on the test circuit. Experimental results prove the validity of such a methodology to achieve localization for autonomous driving.

In this paper, we study SLAM failure and ambiguity scenarios when considering 2D laser-based approaches. Then we propose to detect a priori these possible failure situations based on extracting relevant descriptors from raw perception data. Our approach does not put any assumption on how the underlying SLAM works.

III. SLAM Failure Description

By defining failure and non failure of SLAM we intend to analyze the limitations of SLAM algorithms and consequently define their operating range (where they are supposed to operate properly).

We define the term **current scene** as the configuration of the local environment perceived by SLAM. In practice, it is represented by the current sensor observation Z_t jointly with its projection on the local map $Z_t \cup M_{t-1}$ where M_{t-1} is the local SLAM map at previous time step. We define **saliency** as the general term which qualifies a landmark or a set of landmarks as being sufficient information for the SLAM process to calculate a unique solution that is coherent with the real displacement. Mobile objects are excluded from this definition as they invalidate the estimation.

Along the same lines, a **salient scene** is characterized either by containing at least one salient landmark that is both present in the current observation and on the map of the previous time step, or by a salient configuration of a set of landmarks that are shared between the current observation and the previous time step map. In contrast, a **non-salient scene** contains one or more landmarks that, taken as a whole, are not salient.

For the sake of clarity, Figure 1 illustrates some important failure and non-failure scenarios of 2D laser-based SLAM operating on a ground vehicle in urban and peri-urban environments.

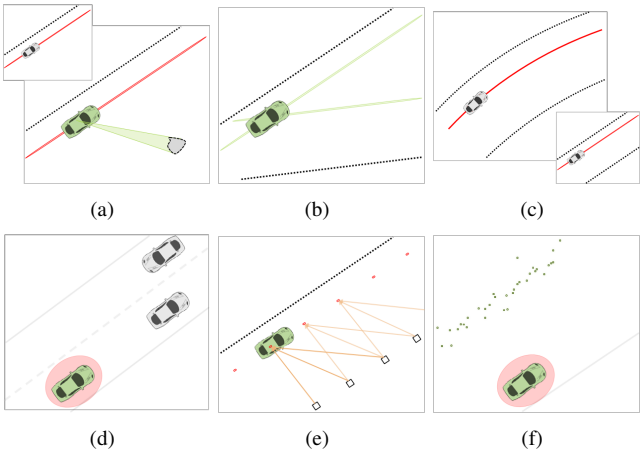

Fig. 1. 2D laser-based SLAM failure and non failure scenarios from a ground vehicle in urban and peri-urban environments. Red ellipses represent the uncertainty area on position estimation induced from the present scenario.

The roadside scenario in Figure 1(a) presents a potential non-salient landmark, although the presence of the boulder which is salient on the right side saves the situation and the scenario becomes salient. In Figure 1(c) adding a parallel roadside does not give any more information and the whole configuration remains non-salient. However, Figure 1(b) shows non equidistant roadsides where each side proposes a set of possible solutions, but the two together intersect at only one point. If salient landmarks are identical and regularly distributed, this may result in a non-salient scene, as we can see in Figure 1(e). Actually, this last scenario is tricky because it depends on the distances between the landmarks and the vehicle's maximum possible velocity. The presence of mobile objects in the scene, as shown in Figure 1(d), may mislead SLAM, as may vegetation or dust, as shown in Figure 1(f), due to the imprecision/vagueness of their position/measurement.

In the following section we present our approach to detecting SLAM failure.

IV. Failure detection

Detecting SLAM failure and non-failure comes down to analyzing the saliency of the current scene. Our approach

is suited to 2D laser-based SLAM, so we will draw on the different scenarios we presented in Section III. The general block diagram of SLAM with SLAM's Failure detector is illustrated in Figure 2.

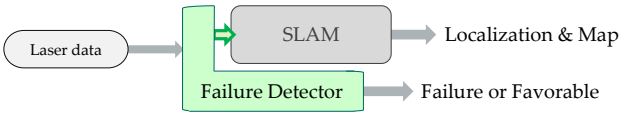

Fig. 2. Block diagram of Failure Detector algorithm with SLAM

The diagram shows that the failure detection and SLAM module are independent and operate simultaneously. Therefore, the failure detection decision does not affect the operation of the SLAM algorithm but should guide the later processing.

A. Approach outline

To ensure the operability of 2D laser-based SLAM algorithms, it is necessary to first study the characteristics of different elements present in the scene. Our strategy consists of the following steps:

1. Clustering of the laser observation: the 2D laser points Z_t will be distributed onto elements $S_t = \{E_t^1, E_t^2, ..., E_t^J\}$ where each E_t^j corresponds to a subset of the observation reflected from one element in the scene. The elements with non sufficient information will be considered as isolated points $Z_t = S_t \bigcup I_t$ where I_t groups all the isolated points from the current laser scan.

2. Detecting, tracking and filtering mobile objects S_t^M. In this way, we keep a list of remaining elements S_t^R with $S_t = S_t^M \bigcup S_t^R$. The issue of mobile object detection is beyond the scope of this paper, and we do not deal with it here.

3. Characterizing remaining elements: where the aim is to characterize and find a 2D geometric approximation for each remaining element $E_t^r \in S_t^R$. We approximate the element E_t^r either by ***line*** (first degree polynomial), ***circle*** (quadratic polynomial) or a ***cubic-spline*** (set of third degree polynomials). Following the order they are presented in, only **one** approximation will be validated based on two consistency measures.

4. Extracting current scene descriptors: based on element characterization, their point cloud and their approximation function, we define a set of descriptors $\mathcal{D}_t$ to describe the whole scene and the relations between different non-salient elements.

5. Decision-making: the information extracted from the previous steps serves as input for failure detection algorithm. Several strategies are proposed.

B. laser observation clustering

An appropriate clustering method should consider the distance between points as a similarity criterion. In addition, the Euclidean distance between a couple of points separated by the same angle varies depending on how far the couple of points is from the sensor origin. In order to group points reflected from the same element, we use a modified version based on a connected component labeling algorithm [19]. This derived version uses a circular neighboring window of variable size. Based on ρ which is the distance of the point considered to the sensor center, the window size is calculated as follows:

$$\omega(\rho) = c \times \sqrt{\rho} \tag{1}$$

where c is an amplification coefficient determined experimentally.

Finally, elements with non sufficient information i.e. three or less points, will be considered as isolated points as they do not provide enough information. Thus $Z_t = (\bigcup_{j=1}^{J} E_t^j) \bigcup I_t$ where I_t groups all the isolated points from the current laser scan, and $S_t = \{E_t^1, E_t^2, ..., E_t^J\}$ is the set of clusters.

C. Geometric characterization of the remaining elements

In order to extract an appropriate description of the scene for failure detection, we need to characterize the different elements composing the scene. A relevant characterization should focus on the geometric representation of salient and non-salient elements.

The saliency of each scene element E_t^r is directly related to its 2D approximation. Non-salient point configurations are commonly observed as long *line-segments* or an arc of a circle of large radius, hereafter referred to as a *large-circle-arc*. In contrast, salient elements have more complex forms, which we refer to as *curves*, such as open polygons e.g. a building with one or several corners, or some other non-regular but basically consistent shape.

Consequently, every element E_t^r in the scene will then be approximated by either:

1. **a line-segment:** using an algebraic linear regression [20]. Line-segments are non-salient elements, although they provide useful information but only along one axis, i.e. the distance between the vehicle and the roadside cf. Figure 1(a).

2. **a circle-arc:** using an algebraic circular regression following Taubin [21] [22]. laser points distributed along an arc could suggest a non-salient configuration cf. Figure 1(c).

3. **a cubic spline curve:** using a smoothing cubic spline algorithm [23], the aim is to handle all other shapes in order to qualify the element as {*smooth-curve, noisy-curve or non-qualified-curve*} according to the consistency of its point-cloud silhouette.

Following the order they were presented in, only one geometric approximation is chosen to represent an element. For this purpose, the approximation is validated through two consistency measures (δ_1, δ_2) together with a couple of thresholds (τ_1, τ_2) set experimentally. These consistency measures calculate the quantity and quality of dispersion of the point-cloud from the suggested approximation.

Assuming the element E_t^r, which is formed by a point-cloud of N laser impacts $E_t^r = \{z_t^{1,r}, z_t^{2,r}, \dots z_t^{N,r}\}$. Let us also assume that its 2D approximation is represented by $\mathcal{A}(i, E_t^r)$ where $i \in \{1 : line - segment, 2 : circle - arc, 3 : curve\}$.

The function $dist(z_t^{n,r}, \mathcal{A}(i, E_t^r))$ calculates the shortest distance between a laser endpoint $z_t^{n,r}$ and a 2D approximation $\mathcal{A}(i, E_t^r)$. Hence the consistencies are calculated as follows:

The dispersion quality δ_1 is the mean of the cumulative distance of each point in the point-cloud from the approximated shape:

$$\delta_1(E_t^r, \mathcal{A}(i, E_t^r)) = \frac{1}{N} \sum_{n=1}^{N} dist(z_t^{n,r}, \mathcal{A}(i, E_t^r)) \quad (2)$$

The dispersion quantity δ_2 is the percentage of good fittings i.e. the normalized number of points which are under a certain distance σ (fixed according to the laser measurement accuracy) from the approximated shape:

$$\delta_2(E_t^r, \mathcal{A}(i, E_t^r)) = \frac{1}{N} \times card\Big(\{z_t^{n,r} \in E_t^r |\ dist(z_t^{n,r}, \mathcal{A}(i, E_t^r)) < \sigma\}\Big) \quad (3)$$

To summarize, at the end of this step, each element E_t^r will have, in addition to its point cloud, an attributed geometric label in {*line-segment, circle-arc, smooth-curve, noisy-curve, non-qualified-curve*}, its approximation function and its dispersion values (δ_1, δ_2) to the selected approximation. This information will be used in the next step to extract a relevant description vector of the scene.

D. Descriptor extraction

Once the different elements that make up the scene have been characterized (a geometric representation and an appropriate function have been determined), we still need to determine how their organization in the environment may affect the SLAM process. As we have seen in Figure 1(c), when the scene contains non-salient elements of the same class (i.e. only *line-segments* or only *circle-arcs*) it is important to calculate how *parallel* the line-segments are, and how *concentric* the circle-arcs are.

According to this and the previous description, we propose a set of 20 descriptors $\mathcal{D}_t = \{d_t^1, d_t^2, \dots d_t^{20}\}$ to decide whether the current scenario corresponds to SLAM failure or not. These descriptors are listed below:

1) *Maximal difference between line-segments slopes*: designed to measure the parallelism between line-segments. Let us consider having a number of Q line-segments sorted according to their slope values. Each element $q \in \{1, 2, \dots Q\}$ is approximated by $\mathcal{A}(1, E_t^q) \cong y = a^q x + b^q$.

$$d_t^1 = f(a^Q) - f(a^1) \quad (4)$$

with $f : \Re \to \Re$ is the function that takes the slope value as an argument and returns the corresponding incline angle.

2) *Diameter of the centers of the large-circle-arcs*: this descriptor measures the concentricity between circle-arcs. Let us consider having a number of K circle-arcs where element $k \in \{1, 2, \dots K\}$ is approximated by $\mathcal{A}(2, E_t^k) \cong (x - x_0^k)^2 + (y - y_0^k)^2 = r^2$.

$$d_t^2 = \underset{p,k \in \mathcal{C}}{\operatorname{argmax}}\ g(p, k) \quad (5)$$

where $\mathcal{C}$ is the set of center points of circle-arcs and $g : \Re \to \Re$ is the function that returns the Euclidean distance between two points.

3) *Raw observation number of points*: $d_t^3 = card(Z_t)$.

4) *Number of isolated laser points*: $d_t^4 = card(I_t)$.

5) 6) *Number of current scene elements and their cardinal.*

7) 8) *Number of line-segment elements and their cardinal.*

9) 10) *Number of circle-arc elements and their cardinal.*

11) 12) *The number of smooth-curve elements and their cardinal.*

13) 14) *Number of noisy-curve elements and their cardinal.*

15) 16) *The number of non-qualified elements and their cardinal.*

17) 18) *Number of parallel line-segments and their cardinal.*

19) 20) *Number of concentric arc elements and their cardinal.*

E. Decision making

All the previous processing results in transforming the raw observation Z_t into a set of descriptors $\mathcal{D}_t$ appropriate for decision making algorithms. For this purpose we propose to compare two strategies to detect SLAM failure:

*1) **Inference rules for SLAM failure detection**:* Based on a set of rules defined thanks to the previously described scenarios (Section III). The aim is to match the current scene description to know whether it could provoke a failure for SLAM or not.

- ***Failure***: $if\ \exists$ **only parallel** *line-segements*
- ***Failure***: $if\ \exists$ **only concentric** *circle-arcs*
- ***Failure***: $if\ \exists$ **only** *noisy elements*
- ***Failure***: $if\ \exists$ **only** *non-qualified elements*
- ***Favorable***: $if\ \exists$ **at least one** *smooth curve*
- ***Favorable***: $if\ \exists$ **at least two elements in** {*line-segment, circle-arc, non qualified*}
- $Otherwise$: ***Failure***.

This strategy supposes that mobile objects were properly removed and noisy elements were correctly detected. In addition it favors the decision *failure* when the situation in question is not clear enough.

*2) **Supervised machine learning for SLAM failure detection**:* Machine learning algorithms require a *training phase* where the learning algorithm modifies the model's weights in order to learn how to take a decision when faced with different possible scenarios. A labeled database is mandatory in order to teach these models. The training examples for these algorithms are formed by the set of descriptors $\mathcal{D}_i$ together with an adequate label:

$$T = \{(\mathcal{D}_i, l_i) \mid l_i \in \{+1, -1\}\}$$

where $l_i = \{+1\}$ indicates that the current scene is labeled favorable for SLAM, and $l_i = \{-1\}$ indicates a failure for SLAM.

Once the training phase has been completed, the learnt model can be used for failure detection. In our study we have considered standard machine learning methods used

for classification [24][25]. The algorithms we tested are the following:

1) Logit regression. 2) Multilayer perceptron (MLP). 3) Ensemble MLP. 4) Decision forests. 5) Adaptive Boosting.

V. Datasets, experiments and results

A. Labeling and Datasets

The approach we are presenting in this paper has not been explored before, consequently, benchmarks with appropriate data (i.e. labeled failure and favorable for SLAM) to evaluate our algorithm do not exist. Furthermore, automatic labeling cannot be an option due to the undefined behavior of SLAM when in a failure scenario. For these reasons we labeled our experimental data manually, frame by frame. This was done based on current laser scan views together with a local-map view. In practice, objective labeling is not an easy task due to the noise accompanying laser observations, the diversity of elements present in the environment, their geometric shape and identifying some specific configuration. For example, a human expert cannot estimate the precise incline difference between roadsides, and consequently the real impact it can have on the SLAM estimation. To mitigate this kind of bias the labeling was done by an expert in SLAM algorithms.

Several data sequences were acquired using a moving vehicle in a variety of urban and peri-urban environments. This car is equipped with five IBEO-Lux sensors deployed to cover a 360°field of view around the vehicle. Table I gives a quick statistical overview of the acquired datasets and test sequences.

TABLE I
Acquired datasets for failure detection learning and evaluation, showing the number of frames annotated failure and favorable in each dataset

	Full FD	Training	FD Test	Plaisir	Amsterdam
Failure	10045	8337	1708	4413	8263
Favorable	8926	7349	1577	14139	5445
Total	18971	15686	3285	20501	15911

Failure Detection dataset built upon a variety of subsequences without mobile objects in the view. This database is intended mainly for training and contains only clear passages which required less difficulty to be labeled.

Plaisir & Amsterdam Circuits are two long and difficult sequences covering a diversity of scenarios containing mobile objects that have not been removed. The first circuit, acquired at Plaisir (France), is 14Km long, starting from a car park, passing round-abouts, dusty roads and rural areas with only one roadside visible in the laser view. The second circuit is more tricky for SLAM as the car was driving mainly on highways around Amsterdam. In order to avoid subjective labeling, confusing passages were omitted (not labeled) from these sequences.

In addition to these datasets, we use the sequence (01) from the KITTI odometry dataset [26] that contains potential failure passages for 2D horizontal laser-based SLAM approaches (i.e. a highway with mobile cars).

B. Experiments

The validity of our approach was experimented through three test scenarios:

1. The first experiment is based on our *Failure Detection dataset* which is divided as follows: four fifths for training and one fifth for evaluation. It is important to mention that subsequences of the test-set and training-set were acquired from different environments and do not contain mobile elements.

2. The second experiment aims to evaluate the proposed approach in realistic difficult scenarios (a complete sequence) and its sensitivity to the presence of mobile objects (not removed). In this experiment we take the whole *Failure Detection dataset* as a learning-set and we evaluate our approach on the *Plaisir* and *Amsterdam* sequences.

3. The third experiment aims to evaluate the impact of *Failure Detector* on SLAM's estimation. We applied our Failure Detector using the best rated method from the previous experiments together with a PML-SLAM [5] implementation. This SLAM implementation is a maximum likelihood 2D laser-based SLAM approach that uses an occupancy grid representation.

C. Results and Discussion

The results of the first and second experiments are shown in Table II which summarizes the best correct classification rates per method over the different sequences. Table III gives more detailed information, as it shows the confusion matrix which is organized as follows:

$$\mathbb{C} = \begin{pmatrix} True\ Negative & False\ Positive \\ False\ Negative & True\ Positive \end{pmatrix}$$

where *Negative* refers to SLAM failure and *Positive* refers to SLAM non-failure. *True Negative* represents the percentage of frames which were annotated failure and classified failure by the decision making method. *False Positive* refers to failure frames that were wrongly classified favorable, while *False Negative* refers to the percentage of Favorable frames which were wrongly classified failure. Finally *True Positive* refers to favorable frames which were properly classified.

The results show that AdaBoost gives the best results over the other methods, with considerably better correct classification rates on the FD-Test dataset and Amsterdam circuit. Random Forests, MLP and Ensemble MLP give comparable results. The logistic regression and Inference rules methods give more modest scores.

The FD-Test dataset evaluation gives considerably more satisfactory classification rates than on the Plaisir circuit. This is due to the presence of mobile objects, which can result in a higher *False Positive* rate, and to noisy passages that induce a considerably higher *False Negative* rate. These arguments are also valid for the Amsterdam circuit evaluation, except that there are many fewer noisy passages and less vegetation areas.

The results of the third experiment are illustrated in Figure 3 where the histogram shows the errors in estimating the

TABLE II
TABLE OF CORRECT CLASSIFICATION RATES OF EACH DECISION MAKING METHOD ON THE DIFFERENT DATASETS

	Inference Rules	Logistic Regression	Multilayer perceptron	Decision Forests	Ensemble MLP	Adaptive Boosting
FD Testset	65.69%	70.75%	79.70%	80.67%	78.72%	85.57%
Plaisir Circuit	45.35%	66.76%	68.35%	67.76%	68.66%	69.66%
Amsterdam Circuit	60.18%	56.08%	65.39%	66.77%	63.54%	74.61%

TABLE III
CONFUSION MATRIX OF THE PROPOSED DECISION MAKING METHODS AND DATASETS. EACH CELL CONTAINS FOUR VALUES, FROM TOP LEFT TO BOTTOM RIGHT: TN: TRUE NEGATIVE, FP: FALSE POSITIVE, FN: FALSE NEGATIVE AND TP TRUE POSITIVE. *Positive* REFERS TO FRAMES ANNOTATED FAVORABLE FOR SLAM AND *Negative* REFERS TO FAILURE

	Inference Rules		Logistic Regression		Multilayer perceptron		Decision Forests		Ensemble MLP		Adaptive Boosting	
FD Testset	43.9574	26.2709	31.0198	20.9741	41.4612	10.5327	43.5921	8.4018	40.7001	11.2938	43.8356	8.1583
	8.0365	21.7352	8.2801	39.7260	9.7717	38.2344	10.9285	37.0776	9.9848	38.0213	6.2709	41.7352
Plaisir	23.4961	54.3607	11.2117	12.5755	11.4974	12.2898	12.3006	11.4866	10.8075	12.9797	10.3493	13.4379
	0.2911	21.8521	20.6662	55.5466	19.3564	56.8564	20.7579	55.4549	18.3646	57.8482	16.8984	59.3144
Amsterdam	51.6997	31.2445	30.3254	29.9460	42.0120	18.2594	45.1415	15.1299	37.6423	22.6291	51.8019	8.4695
	8.5789	8.4768	13.9845	25.7441	16.3554	23.3732	18.0989	21.6297	13.8313	25.8973	16.9171	22.8115

displacement at each SLAM time step, with and without our Failure Detector.

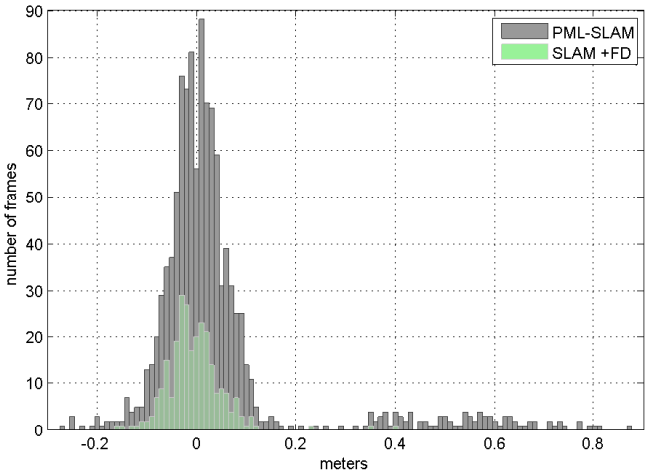

Fig. 3. Histogram of errors in displacement estimation by frame, with and without Failure Detection

This KITTI test sequence contained initially 1101 frames which were annotated using AdaBoost as follows: 411 (37.33%) as Favorable and 690 (62.67%) as Failure, which is coherent as the car mainly navigated on a highway with several exits/junctions along the way. Note that for the failure detection evaluation, we only consider the estimation of SLAM for frames annotated Favorable which were immediately preceded by a Favorable frame. This reduces the number of frames considered to 266 (24.18%) estimations out of 1100.

SLAM failure can be seen in the histogram in Figure 3 represented mainly by the aberrant displacement estimations (i.e. errors of large value). If we consider that SLAM failure occurs when an error in displacement is greater than some tolerance value, Table IV shows the number of aberrant SLAM estimations with and without our Failure Detector. We observe that our Failure Detector identifies most of these aberrant estimations at a reasonable tolerance value, i.e. 4 aberrant estimations if the tolerated range of error in displacement is bounded by $15cm$.

TABLE IV
NUMBER OF ABERRANT ESTIMATIONS OF SLAM VS. SLAM +FD CONSIDERING DIFFERENT TOLERANCE ON DISPLACEMENT ERRORS

Tolerance(cm)	5	10	15	20	30	40	50	65	80
SLAM	439	176	119	105	93	74	52	17	2
SLAM +FD	78	13	4	3	2	0	0	0	0

In contrast, considering the same criterion, i.e. a $15cm$ tolerance value on estimation errors, we see that the Failure Detector also eliminates many frames which seem to give acceptable estimations i.e. $690 - 119 = 571$ frames. The reasons mentioned earlier about the subjective nature of labeling are still valid here. In addition, having a reasonable error does not necessarily imply that the scenario is favorable for SLAM, particularly with the presence of misleading elements such as mobile objects which were not removed.

VI. CONCLUSION AND PERSPECTIVES

In this paper we have presented an approach to detect and isolate odd configurations of the surrounding environment which could potentially mislead SLAM. As an example, the well-known tunnel scenario where localization using SLAM becomes a difficult, if not impossible task. State-of-the-art methods do not attempt to detect such limitations of SLAM algorithms.

Our approach is designed for 2D laser-based SLAM algorithms based solely on the analysis of the sensor data. First we cluster the laser observation into elements. Then each element is characterized by a 2D geometric approximation which is used together with laser points to build a relevant features vector. Finally, various standard decision-making algorithms are used and compared. The validity of our approach was proved through three experiments. AdaBoost learner succeeded in properly classifying (failure or favorable)

85.57% of scenarios. The last experiment showed that using Failure Detector together with SLAM helped to isolate almost all the aberrant estimations given by SLAM. In contrast, many other estimations were omitted even though they had a reasonable error range.

In future work, we plan to extend our technique and study the internal elements of the SLAM process in order to detect possible failure. In addition, it is worth studying the impact of using a training set which contains noisy passages.

REFERENCES

[1] T. Bailey and H. Durrant-Whyte, "Simultaneous localization and mapping (SLAM): Part II," *IEEE Robotics and Automation Magazine*, vol. 13, no. 3, pp. 108–117, 2006.

[2] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: part I," *IEEE Robotics & Automation Magazine*, vol. 13, no. 2, pp. 99–110, 2006.

[3] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1309–1332, 2016.

[4] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, "FastSLAM: A factored solution to the simultaneous localization and mapping problem," in *Proc. of 8th National Conference on Artificial Intelligence/14th Conference on Innovative Applications of Artificial Intelligence*, vol. 68, no. 2, 2002, pp. 593–598.

[5] Z. Alsayed, G. Bresson, F. Nashashibi, and A. Verroust-Blondet, "PML-SLAM: a solution for localization in large-scale urban environments," in *Proceedings of the 7th Workshop on Planning, Perception and Navigation for Intelligent Vehicles*, Hambourg, 2015.

[6] J. Xie, F. Nashashibi, M. Parent, and O. G. Favrot, "A real-time robust global localization for autonomous mobile robots in large environments," *11th International Conference on Control, Automation, Robotics and Vision, ICARCV 2010*, vol. 1, pp. 1397–1402, 2010.

[7] T.-D. Vu, "Vehicle perception: Localization, mapping with detection, classification and tracking of moving objects," Ph.D. dissertation, 2009.

[8] C.-C. Wang, C. Thorpe, and A. Suppe, "Ladar-based detection and tracking of moving objects from a ground vehicle at high speeds," in *Intelligent Vehicles Symposium, 2003. Proceedings. IEEE*. IEEE, 2003, pp. 416–421.

[9] C.-C. Wang, C. Thorpe, and S. Thrun, "Online simultaneous localization and mapping with detection and tracking of moving objects: Theory and results from a ground vehicle in crowded urban areas," in *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on*, vol. 1. IEEE, 2003, pp. 842–849.

[10] G. Trehard, Z. Alsayed, E. Pollard, B. Bradai, and F. Nashashibi, "Credibilist Simultaneous Localization and Mapping with a LIDAR," in *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*. IEEE, 2014, pp. 2699–2706.

[11] G. Grisetti, R. Kummerle, C. Stachniss, and W. Burgard, "A Tutorial on Graph-Based SLAM," *IEEE Intelligent Transportation Systems Magazine*, vol. 2, no. 4, pp. 31–43, 2010.

[12] S. Roumeliotis, "Sensor Fault Detection and Identification in a Mobile Robot," in *Intelligent Robots and Systems*, vol. 3. IEEE, 1998, pp. 2–7.

[13] P. Goel, G. Dedeoglu, S. Roumeliotis, and G. Sukhatme, "Fault detection and identification in a mobile robot using multiple model estimation and neural network," in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, vol. 3. IEEE, 2000, pp. 2302–2309.

[14] P. Sundvall and P. Jensfelt, "Fault detection for mobile robots using redundant positioning systems," in *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.* IEEE, 2006, pp. 3781–3786.

[15] Y. Morales, E. Takeuchi, and T. Tsubouchi, "Vehicle localization in outdoor woodland environments with sensor fault detection," in *Proceedings - IEEE International Conference on Robotics and Automation*. IEEE, 2008, pp. 449–454.

[16] A. Jabbari, R. Jedermann, and W. Lang, "Application of Computational Intelligence for Sensor Fault Detection and Isolation," in *Proceedings of World Academy of Science, Engineering and Technology*, vol. 22, no. July. Citeseer, 2007, pp. 503–508.

[17] L. Wei, C. Cappelle, and Y. Ruichek, "Camera/laser/gps fusion method for vehicle positioning under extended nis-based sensor validation," *IEEE Transactions on Instrumentation and Measurement*, vol. 62, no. 11, pp. 3110–3122, 2013.

[18] G. Bresson, M. C. Rahal, D. Gruyer, M. Revilloud, and Z. Alsayed, "A Cooperative fusion architecture for robust localization: Application to autonomous driving," in *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, 2016, pp. 859–866.

[19] H. Samet and M. Tamminen, "Efficient component labeling of images of arbitrary dimension represented by linear bintrees," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 10, no. 4, pp. 579–586, 1988.

[20] D. Freedman, *Statistical Models: Theory and Practice*. cambridge university press, 2009.

[21] G. Taubin, "Estimation of planar curves, surfaces, and nonplanar space curves defined by implicit equations with applications to edge and range image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 11, pp. 1115–1138, 1991.

[22] A. Al-Sharadqah and N. Chernov, "Error analysis for circle fitting algorithms," *Electronic Journal of Statistics*, vol. 3, pp. 886–911, 2009.

[23] C. H. Reinsch, "Smoothing by Spline Functions. II," *Numer. Math.*, vol. 16, no. 3, pp. 451–454, 1971.

[24] E. Alpaydin, *Ethem Alpaydin. Introduction to Machine Learning (Adaptive Computation and Machine Learning Series)*. MIT Press, 2008, vol. 14, no. 01.

[25] W. Yoo, B. A. Ference, M. L. Cote, and A. Schwartz, "A comparison of logistic regression, logic regression, classification tree, and random forests to identify effective gene-gene and gene-environmental interactions," *International journal of applied science and technology*, vol. 2, no. 7, p. 268, 2012.

[26] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.