High accuracy high integrity train positioning based on GNSS and image processing integration

Alessandro Neri *Roma Tre University, RadioLabs*
Federica Battisti *University of Padova*
Sara Baldoni, Michele Brizzi, Luca Pallotta *Roma Tre University*
Agostino Ruggeri *RadioLabs*
Gianluigi Lauro *Hitachi Rail STS*

BIOGRAPHY

Alessandro Neri is full professor in Telecommunications at Roma Tre University. Since 2009, he is the President of RadioLabs, a not-for-profit research center based on the partnership between universities and industries. His research activity has mainly been focused on information theory, signal and image processing, location and navigation technologies.

Federica Battisti is currently tenure-track Assistant Professor in the Department of Information Engineering at University of Padova. Her main research interests are in the field of signal and image processing.

Sara Baldoni is a Ph.D. student in Applied Electronics in the Department of Engineering at Roma Tre University. Her main research interests are in the area of Communication Security and Navigation and Localization Systems.

Michele Brizzi is a Ph.D. student in Applied Electronics in the Department of Engineering at Roma Tre University. His main research interests are in the field of image processing and imaging sensors for visual navigation.

Luca Pallotta is a non-tenured Assistant Professor at Department of Engineering of Roma Tre University. His research interests lie in the field of radar target detection, automatic target recognition, polarimetric SAR image classification, and statistical signal processing with emphasis on radar/SAR signal processing.

Agostino Ruggeri is a senior network engineer at Radiolabs, Italy. He has experience in applied research programs and in the field of GNSS and TLC applications for railway safety-related systems. He participated to several international projects funded by ESA and the EC concerning the employment of Next Generation Communication Systems in ERTMS.

Gianluigi Lauro is a Senior Innovation engineer at Hitachi Rail STS. He is technical coordinator for international projects funded by ESA and the European commission on topics related to innovative train positioning systems.

ABSTRACT

One of the major challenges in the design of high accuracy, high integrity localization procedures for rail applications based on Global Navigation Satellite Systems is represented by the local hazards that cannot be mitigated by resorting to augmentation networks. By fact, combining smoothed code pseudoranges with (differential) carrier phase and/or with Inertial Measurement Unit's outputs is ineffective against multipath low frequency components. These issues can be mitigated by processing images, depth maps and/or pointclouds provided by imaging sensors placed on board. The absolute position of the train can be determined by combining its relative position with respect to georeferenced rail infrastructure elements (e.g., panels, signals, signal gantries) provided by the visual localization processing unit with the landmark absolute position. In addition, the visual input can be exploited for determining on which track the train is located and can be used as complementary odometry source. Moreover, the information provided by the visual localization processing unit can be used to monitor integrity and compute the protection levels. In this contribution we present a localization system that integrates a Global Navigation Satellite System receiver, Inertial Measurement Units, and video sensors (such as monocular and stereo video camera, Time of Flight camera and LIDAR), and has the potential to overcome some of the operational and economical limitations of the current train localization system employed in the European Railway Traffic Management System.

I. INTRODUCTION

The exact knowledge of trains' position is one of the most crucial information needed by modern rail traffic management systems. This information is needed for many purposes such as reducing the safety distance between two trains travelling on the same track or discriminating the track on which a train is running. Currently, the European Railway Traffic Management System (ERTMS) Train Control System (TCS) allows to localize the train through equipment deployed on the tracks. Each track is usually split

into sections, the blocks. Thanks to track circuits deployed along the track, it is possible to know if a block is occupied by a train or not. This information is important since, as a rule, each block cannot be occupied by more than one train at the same time. A track circuit is composed by two elements: a low-power feed and a relay connected by the rails. When the block is not occupied by a train, the relay is energized, while when a train gets in the block, the wheels and the axles of the rolling stocks short out the electrical circuit and, consequently, de-energize the relay. In principle the data acquired by the odometer, inertial sensors and the Global Navigation Satellite System (GNSS) receiver could be jointly processed in order to provide location services to the TCS, as in the solution proposed in [1] for Low Density Lines in China. However in ERTMS the GNSS based estimation of the train location is currently decoupled from odometry. To properly handle slipping and sliding phenomena, advanced odometry jointly processes tightly coupled mechanical odometers and inertial sensor packages [2]. Currently, the objective is to reduce the signaling cost while maximizing the train traffic through the virtualization of the signaling equipment. Moreover, concerning the positioning task, the trend is to remove the wayside component, and to replace it with on-board localization sensors. In this paper we present a train positioning system based on the integration of GNSS, Inertial Measurement Unit (IMU), and data collected from vision sensors (depth and intensity image acquired with a depth camera, pointclouds recorded with a Time of Flight (ToF) camera and/or a LIDAR). This system is under study and development in the framework of the ESA NAVISP 2 project "VOLIERA-Video Odometry with LIDAR and EGNSS for ERTMS applications" [3].

In the definition of the VOLIERA architecture we selected the following types of visual sensors:

- **Stereo camera:** a pair of calibrated, synchronized image sensors acquire the same scene from two different viewpoints. The displacement (in pixels) introduced by parallax on the position of corresponding points is measured by a stereo matching algorithm and used to compute the distance (in meters) of that point. The output of a stereo camera is an image whose pixels represent the distance of 3D points in camera coordinates, the depth map.
- **LIDAR:** the LIDAR sensor collects through consecutive scans a pointcloud (i.e., a collection of 3D points in the sensor's reference system) with x, y, and z coordinates of the surrounding objects by measuring the round-trip delay of a transmitted light pulse. For each point, a reflectivity value is also estimated based on the measured attenuation of the received echo. It can work day and night without the influence of lighting conditions and in almost all weather conditions. Moreover, it is characterized by high accuracy and reliability, which is the major reason that LIDARs are more and more used as a critical component of self-driving vehicle sensor systems.
- **ToF camera:** a ranging device similar to the LIDAR, it provides a measure of the distance of objects by measuring the phase difference of a modulated infrared light source. Using a camera sensor, it provides a 2D depth map instead of a pointcloud. However, a pointcloud can be extracted by using the camera projection parameters.

The data collected through the selected sensors allows the VOLIERA to provide three capabilities:

- **Track Discrimination**: VOLIERA allows to discriminate the track on which the train is located. The recorded depth map is converted to the corresponding pointcloud, then a registration step is performed to represent the depth map-based pointcloud and the original pointcloud in the same reference system. After registration, the two pointclouds are merged to obtain an augmented pointcloud which is used as input to the track discrimination algorithm. This last exploits the information provided by GNSS and IMU (i.e. position and position estimate confidence interval) to speed up and simplify the track identification task. More in details, given the area in which the train is located, the information about the number of tracks can be obtained, thus reducing the complexity of the track discrimination algorithm.
- **Absolute Positioning**: it provides the 3D position of the train with respect to an external reference frame, as well as the arclength of the track with respect to an initial reference point, usually denoted as train mileage. Conspicuous points such as railway infrastructure elements whose position is known are identified and used to compute the train's position by relying on well established techniques for object detection and recognition based on deep learning.
- **Enhanced Odometry**: it aims at computing the relative pose transformation between consecutive acquisitions of the visual sensors, in order to complement the information provided by the mechanical odometer or the solution provided by inertial sensors. The estimation of the train pose update is performed through a late fusion between the pose update provided by the stereo odometry algorithm which exploits the depth map and the intensity image, and the one obtained through the odometry algorithm based on the pointcloud.

II. VOLIERA ARCHITECTURE

As previously mentioned, the visual data needed in the VOLIERA approach can be recorded through largely available hardware such as depth cameras, LIDARs and ToF cameras.

The sensors employed for the VOLIERA project are shown in Figure 1. For our purposes we selected the Livox Horizon LIDAR [4], the Basler Blaze ToF camera [5], and the Nerian Karmin3 stereo camera [6].

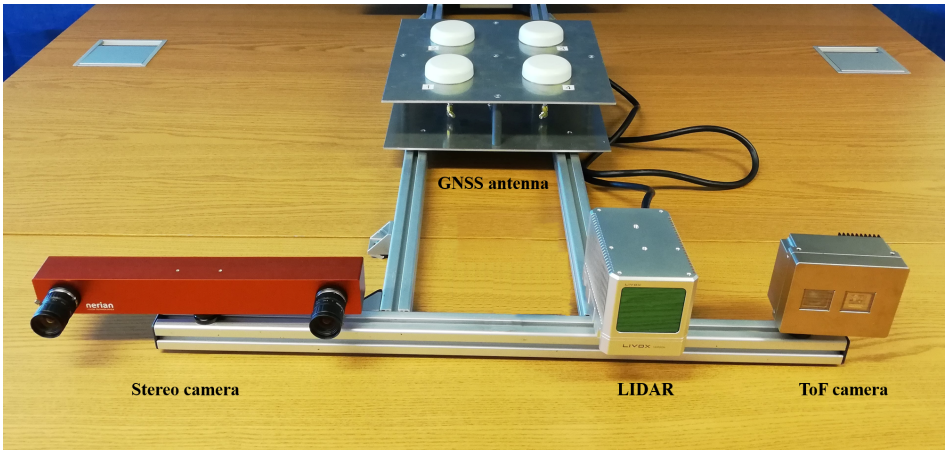

Figure 1: VOLIERA sensors.

As shown in VOLIERA system architecture (Figure 2), there is some redundancy in the type of information acquired by these sensors. In fact, both the stereo camera and the ToF camera provide a depth map, and both the ToF and the LIDAR provide a pointcloud of the scene. This is an intentional design choice pushed by the need of building a system which is robust against sensor's failures. Moreover, despite acquiring similar data, the differences in the characteristics and working principles of the sensors (e.g., maximum range, accuracy, robustness against harsh environmental conditions) contribute to the system's ability to withstand unexpected operational scenarios.

In addition to pointclouds and depth maps, in order to provide the required functionalities of Track Discrimination, Absolute Positioning and Enhanced Odometry, we also make use of the rectified images extracted from the stereo camera (intensity image in Figure 2).

Additionally, in order to maintain the sensors synchronized, in our architecture a GNSS receiver acts as a master clock and provides the reference time, while the IEEE 1588 Precision Time Protocol (PTP) is used for clock distribution.

A tightly-coupled integration of IMU and GNSS (IMU/GNSS module in the figure) provides a coarse navigation solution that is used as a starting point by the other modules.

Depending on the train operating conditions, the sensors might record unreliable data. To this aim, after acquisition a Fault Detection and Exclusion (FDE) step is performed to detect sensor faults and, if necessary, correct or exclude them by appropriately configuring the processing chain. In the following, the algorithms designed and implemented for performing the three tasks of Track Discrimination, Absolute Positioning and Enhanced Odometry are presented.

III. TRACK DISCRIMINATION

This section is devoted to the description of the selected rail tracks discrimination method. In particular, it is based on the exploitation of the augmented pointcloud of the observed scene comprising one or more tracks. This method is an adaptation of the one designed in [7] which allows the detection of railway tracks over a more complex scene. The algorithm consists of few steps and exploits the differences in elevation of rail tracks with respect to the other objects and the terrain. A block scheme synthetically describing the pointclouds-based track discrimination method is reported in Figure 3.

In particular, the first step consists in applying a sliding window of size 1×1 m to analyze the heights of the points that it contains. This processing is performed over the entire acquired pointcloud. By doing so, a first coarse indication about the position of potential rail tracks is performed. More precisely, multiple thresholding procedures are applied to cancel all the points that certainly do not belong to a railway track (e.g., points above the rail tracks could be associated with other scattering points in the scene such as wires).

After the coarse selection of points in the augmented pointclouds is performed, a procedure aimed at rejecting outliers while extracting linear structures is applied. In this respect, reasonably assuming that there is only one rail track within each sliding window, the line fitting is performed using the well-known RANSAC algorithm [8]. RANSAC algorithm is essentially tuned in order to account for the expected percentage of inliers as well as their maximum distance to the line. It is worth to recall herein that the RANSAC, differently from the Least Squares (LS), is capable to better follow the general behavior of the inliers rejecting the outliers that inevitably are contained in the considered pointcloud. After lines are traced within the pointclouds, the

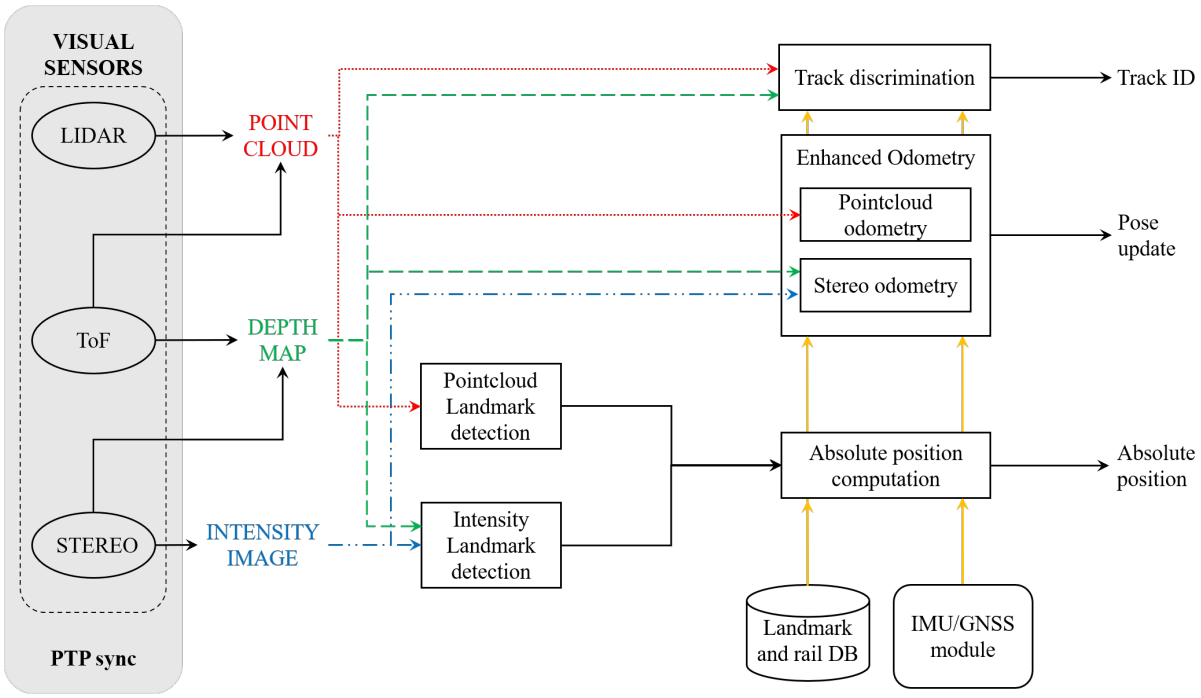

Figure 2: VOLIERA system architecture.

third step of the considered algorithm performs searches for couples of parallel lines associated with the same track. Therefore, each line detected by RANSAC is associated with another one searched within a specific distance from it, e.g., between 1.3 m and 1.6 m.

Finally, the last step of the algorithm performs rail tracks number counting. This operation is needed to acquire knowledge about the number of tracks on the left and right side of the train, to correctly identify the rail track on which the train is. Additionally, it is worth to emphasize the possibility of improving the detection performances by using information contained in the rail database to remove residual outliers, false detected tracks as well as to avoid missing track detection. This is done, thanks to the detailed position information provided by IMU and GNSS.

IV. ABSOLUTE POSITIONING

The proposed solution for computing the train absolute position is an extension of the method proposed in [9]. More in details, the positioning procedure is organized in three steps. First, a coarse estimation of the train location is performed based on the position estimate provided by the IMU/GNSS module. Then, landmarks are detected in the images and pointclouds acquired by stereo camera, ToF camera and LIDAR sensors. Finally, a fine estimation of the train position is computed by exploiting the landmarks' known position.

1. Landmark database

The aim of the first step is to determine which are the traffic lights and the rail signs potentially in view of the imaging subsystem. This is accomplished with the support of the digital map of the railway, which stores the position coordinates as well as the type of the landmarks. Starting from the coarse position solution provided by the IMU/GNSS module, the elements in the database located inside the area of the confidence interval are obtained. The visible landmarks are extracted based on the semantic labels provided by the landmark detection algorithm. The use of the IMU/GNSS module output eases the searching procedure in the database limiting the search area thus decreasing the computational cost [10].

2. Landmark detection

Given the peculiarities of the railway environment, we focus on the detection of traffic lights, traffic signs, and in general the signalling equipment used for train operations.

Depending on the availability of the signals after the FDE procedure, it is possible to perform the detection of the landmarks

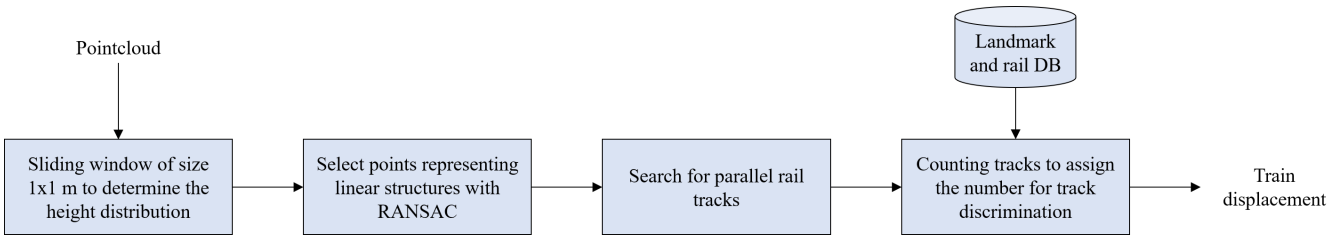

Figure 3: Block scheme of the track discrimination algorithm using pointclouds.

from the images (and related depth maps) acquired by the stereo camera, or from the pointclouds acquired by the LIDAR and ToF sensors.

a). 2D object detection

Typical object detection techniques output a bounding box containing the detected objects along with the semantic IDs associated to the object type. However, to estimate the relative direction of the landmark's Lines of Sight (LoS) with respect to the train, as well as the distance between the two, we need to accurately evaluate the location of the pixels belonging to the landmarks. To practically compute the LoS between the train and the landmark, we define it as the segment whose end-point is the detected landmark's centroid. To do so, we exploit the masks provided by the landmark detection algorithm. More in details, we compute the centroids for all the blobs identified in the mask. Given the displacement between the optical center and the landmark's centroid in pixels, the corresponding value in degrees can be obtained considering the horizontal Field of View (FoV) of the camera and the sensor's resolution. We used Mask R-CNN [11], which represents an extension of the state-of-the-art Faster R-CNN [12], with the inclusion of a semantic segmentation branch for predicting object masks inside the bounding boxes.

Therefore, the object detection algorithm provides two outputs: a 2D binary mask and the 2D semantics information. The former is a binary image with pixel values of 1 for pixels which belong to a landmark, and 0 elsewhere.

2D object detection aims at detecting objects from a 2D image, therefore the resulting bounding boxes or segmentation masks are expressed in the local camera coordinates on the image plane. If the position of the detected object in world coordinates is needed, the depth information is also required, provided for instance by a stereo or ToF camera, and the camera needs to be accurately calibrated.

b). 3D object detection

With respect to the 2D case, 3D object detection methods compute 3D bounding boxes which inherently encode size and position of the object in world coordinates. This information allows a better understanding of the environment. However, the performance gap between 2D and 3D object detection methods is still significant, mainly due to the additional computational complexity and to the need for 3D labelled datasets for training. The task of detecting objects from a pointcloud is challenging because of the sparsity of the data, distortions caused by motion, and occlusions. Nevertheless. the pointclouds acquired by the LIDAR sensor can be used to improve the performances of the landmark detection and supplement the camera information in the case of bad visibility conditions (e.g., presence of haze, dirty sensor) or at night time.

Our approach is based on the PointNet classification model proposed in [13], which addresses many of the aforementioned problems by learning a robust representation of the pointcloud, which is encoded into a compact feature vector.

PointNet has also been shown to be robust against missing points, which is a problem affecting LIDAR scans with low point density (either due to low angular resolution or short acquisition time). When there are 50% points missing, the measured accuracy only drops by 2.4% and 3.8% depending on the task [13].

The output of the 3D object detection algorithm is a binary pointcloud mask, in which each point is equal to 1 if it belongs to a landmark and 0 otherwise.

This procedure is performed when the images and depth maps acquired by the stereo camera are not available and the 2D object detection cannot be performed. On the contrary, if the 2D bounding boxes are available, it is possible to compute the 3D bounding boxes from the 2D ones according to the measured depth, since the registration parameters are known.

3. Positioning

The third step of the absolute positioning procedure consists in computing the train position based on the detected landmarks. Once the landmarks have been correctly identified in the database, their coordinates are extracted and exploited for estimating the train position. More specifically, we identified three processing workflows which will be automatically selected based on the

number of visible landmarks, as well as on the availability of the measurement sensors. Moreover, if no landmark is detected, the position information provided by the IMU/GNSS module is provided as output of the absolute positioning procedure.

4. Workflow 1

When at least three landmarks are visible, the method proposed in [9] can be applied. For sake of clarity, we briefly describe the positioning procedure in the following. Starting from the masks provided by the landmark detection step, the directions of the LoS of the selected landmarks with respect to the train are computed. Once the LoS has been determined, the location of the train is estimated as the intersection of the loci of points for which the differences of the viewing angles corresponding to each landmark pair remain constant. Given three visible landmarks, $\{LM_1, LM_2, LM_3\}$, without loss of generality we assume that LM_1 is the landmark used as pivot to compute the difference of LoS directions α_i. Given LM_i, the locus of points for which the viewing angle α_i stays constant is a portion of the circle passing through the points $\{LM_1, LM_i\}$ and the train location $P = [X_T, Y_T]$. As detailed in [9], the circle center coordinates are given by

$$\mathbf{C_i} = \begin{bmatrix} x_i^C \\ y_i^C \end{bmatrix} = \begin{bmatrix} \frac{x_i+x_1}{2} \pm \frac{y_i-y_1}{2}\cot(\alpha_i) \\ \frac{y_i+y_1}{2} \mp \frac{x_i-x_1}{2}\cot(\alpha_i) \end{bmatrix}, \tag{1}$$

where $[x_i, y_i]$ are the coordinates of the i^{th} landmark. Moreover, the radius of the circles is

$$\rho_i = \frac{d_i}{(2sin(\alpha_i))}, \tag{2}$$

where d_i is the length of the segment $\overline{LM_1 LM_i}$. It is worth noticing that for each landmark pair, two circles can be defined. In Equation 1 the different signs are associated to the two possible centers $C_{(i,1)}$ and $C_{(i,2)}$. As suggested in [9], the ambiguity about the center location can be solved by exploiting the diretion of motion which in case of train positioning is assumed to be known.

Collecting the angle measurements for all the available landmark pairs, the intersections between the circles can be computed by solving the non-linear system

$$(X_T - x_i^C)^2 + (Y_T - y_i^C)^2 = \rho_i^2, \quad i = 2, 3. \tag{3}$$

Let us underline that only two circles are needed since the other intersection point between the circles is represented by the pivot location, which can be automatically excluded as train position candidate.

Let us now discuss the applicability of the workflow 1. Differently from [9], here we use as input of the LoS estimation both the 2D landmark mask provided by the stereo camera and the 3D landmark mask provided by the LIDAR and/or ToF. If all the inputs are available, the angular measurements can be integrated thus providing a more robust solution. In case of failure of one or more sensors, however, the availability of a single input is sufficient for performing the positioning task. The main limitation of the workflow 1, however, is the need for three landmarks in a single acquisition. The first workflow is summarized in Figure 4.

5. Workflow 2

Workflow 2 can be applied when only two landmarks are available. More specifically, under these circumstances, a single angle measurement, and thus a single angle-based circle, can be computed. Thanks to the depth information provided by the installed sensors, however, it is possible to compute the distance (D_i) between the detected landmarks and the train. As a consequence, two circles centered at the landmark locations and whose radius equal the distances between the landmarks and the train, can be defined. As a consequence, the train position can be computed as the intersections between the three circles, which can be obtained by solving the non-linear system:

$$\begin{cases} (X_T - x_\alpha^C)^2 + (Y_T - y_\alpha^C)^2 = \rho_i^2, \\ (X_T - LM_i^x)^2 + (X_T - LM_i^y)^2 = D_i^2 \quad i = 1, 2. \end{cases} \tag{4}$$

In Equation 4, $[LM_i^x, LM_i^y]$ are the i-th landmark coordinates, and $[x_\alpha^C, y_\alpha^C]$ are the coordinates of the center of the angle-based circle.

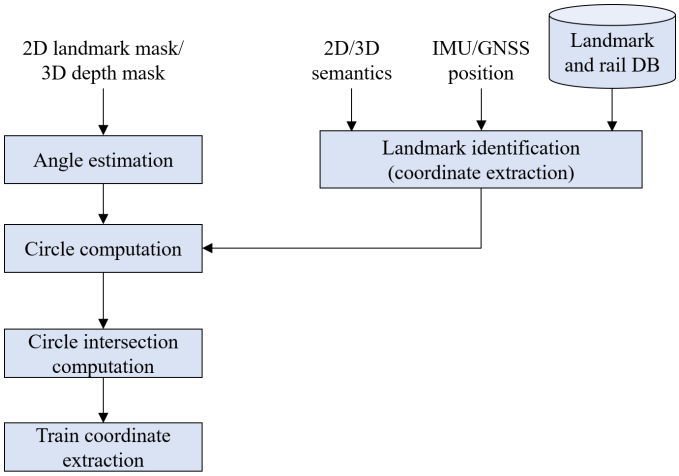

Figure 4: Workflow 1.

Let us now discuss the applicability of the workflow 2. Based on the available inputs, the angle estimation can be performed exploiting the 2D landmark mask, the 3D depth mask or both as described for workflow 1. As for the distance computation, both the 2D depth mask and 3D depth mask can be employed. If both are available, the result of the integration of the distance information can be used to provide a more robust solution. The second workflow is summarized in Figure 5.

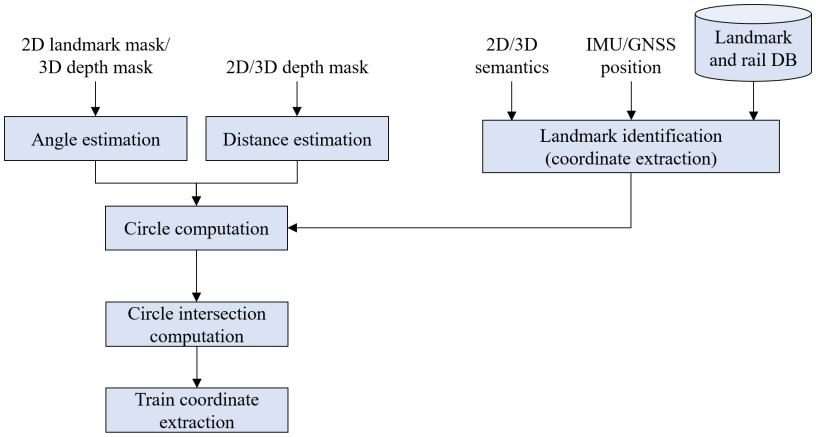

Figure 5: Workflow 2.

6. Workflow 3

Workflow 3 is applied when a single landmark is available by exploiting the distance information. As in workflow 2, a circle centered at the landmark location whose radius equals the distance, D, between the train and the landmark can be computed. The equation of that circle is:

$$(X_T - LM^x)^2 + (X_T - LM^y)^2 = D^2. \tag{5}$$

Assuming that the train path coordinates are recorded in the landmark and rail database, it is possible to compute the train position as the intersection between the aforementioned circle and the train path.

Let us now discuss the applicability of workflow 3. As previously mentioned, both the 2D depth mask and the 3D depth mask can be exploited for computing the distance between the landmark and the train. Workflow 3 is summarized in Figure 6.

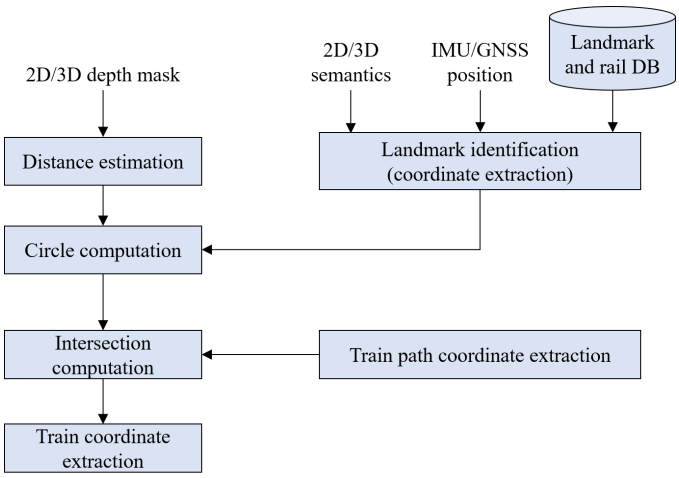

Figure 6: Workflow 3.

V. ODOMETRY

Visual Odometry (VO) can be defined as the estimation of the ego-motion of a moving object performed through the acquired visual input. More in details, given a set of consecutive frames or pointclouds, it is possible to estimate the movement of the objects in the scene from one acquisition to the other. Based on this, the displacement of the moving object can be inferred. In the VOLIERA project two types of visual odometry will be performed: the stereo and the pointcloud odometry. The former relies on the images and depthmaps provided by the stereo camera, wehereas the second takes as input the pointlocuds acquired by the LIDAR or the ToF. The train displacement estimates provided by the two odometry procedures are then fused together with the output of the IMU/GNSS module to obtain the train pose update. The use of a late fusion approach is the result of a trade-off between performances and computational cost.

1. Stereo Visual Odometry

The selected stereo VO approach [14] is based on the one proposed in [15]. The processing workflow is shown in Figure 7.

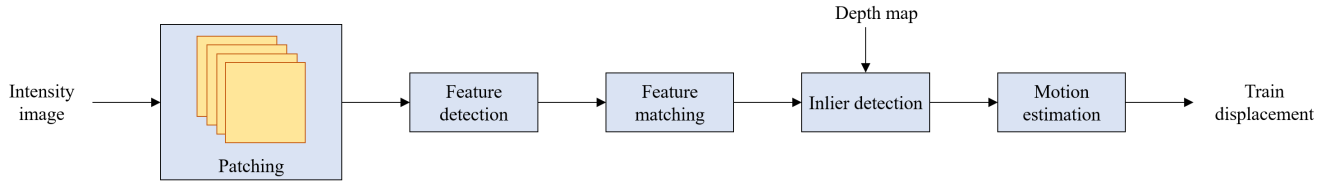

Figure 7: Stereo odometry workflow.

First of all, a patching procedure is applied to the input images. In more details, the image is divided into several non-overlapping portions and a maximum number of feature points are selected for each of them. This process has two advantages:

- the features are properly distributed throughout the image;
- the computational complexity of the following processing chain decreases thanks to the reduction of the input image dimensions.

As for feature extraction, several methods could be used, such as Scale Invariant Feature Transform (SIFT) [16], Speeded Up Robust Features (SURF) [17] and Features from Accelerated Segment Test (FAST). However, for the VOLIERA project we made a choice based on the reduction the computational complexity, thus resorting to FAST feature detection [18].

The features detected for frames acquired at time T and $T+1$ are provided as input to the feature matching algorithm. In principle, the feature matching could be performed by associating to each feature a descriptor, and then comparing the descriptors across the images. To efficiently perform this task, the Kanade-Lucas-Tomasi (KLT) tracker can be employed [19]. In addition, a pyramid approach can be exploited to deal with displacements of different sizes. In this way large motions can be transformed into smaller ones. The KLT tracker provides the coordinates of the features in the frame acquired at time $T+1$ corresponding

to the ones detected in the frame at time T. Moreover, it provides an accuracy and error measure so that low quality features are dropped from further processing.

After feature matching, an inlier detection algorithm is applied [15]. To do so, exploiting the depth map, the 2D matched features are transformed in the corresponding 3D points in world coordinates. The core idea behind the inlier detection procedure is that although the absolute positions of two features are not related, the relative distance between them remains constant in time. As a consequence, a feature pair is considered consistent if the distance between the features at time T (in world coordinates) equals the distance between the corresponding features at time $T+1$. Denoting as f_T the features at time T, and as $f_{(T+1)}$ the ones at time $T+1$, a pair of matches $(f_T, f_{(T+1)})$ and $(f'_T, f'_{(T+1)})$ are considered consistent if:

$$|w_T - w'_T| - |w_{(T+1)} - w'_{(T+1)}| < \delta, \tag{6}$$

where w represent the world coordinates, and δ is a fixed threshold.

At last, the train motion is estimated by finding the transformation Δ which minimizes the image re-projection error for all the matched feature points [15]. The re-projection error can be defined as:

$$\epsilon = \sum_{(f_T, f_{(T+1)}) \in Q} (j_T - P\Delta w_{T+1})^2 + (j_{T+1} - P\Delta^{-1} w_T)^2, \tag{7}$$

where j and w are the homogeneous image and world coordinates for features $(f_T, f_{(T+1)})$, respectively, P is the camera projection matrix, and Q is the largest set of mutually consistent matches. The minimization procedure is performed using the Levenberg-Marquardt least square optimization.

2. Pointcloud odometry

The main solution for pointcloud-based navigation is represented by scan-matching using the Iterative Closest Point (ICP) algorithm [20, 21] followed by pose graph optimization.

The scan matching is performed every time a new pointcloud is acquired and provides estimated relative poses of the sensor between consecutive scans, obtained by aligning the two pointclouds. However, ICP can converge to local minima, therefore an initial estimate is needed. This is provided by the IMU/GNSS module. The pose graph optimization runs on a separate thread at lower rate and optimizes these relative poses to obtain the pose of the sensor in the navigation frame.

Given two consecutive pointclouds $\mathbf{P} = \{p_i\}$ and $\mathbf{Q} = \{q_j\}$, the following steps are performed:

- Pointcloud de-warping: as the train moves forward, points in the pointcloud are warped towards the sensor. In order to correct this geometric distortion, each point (expressed using homogeneous coordinates $p_i = [p_i^x, p_i^y, p_i^z, 1]^T$ is transformed as follows:
$$p_i^t = \mathbf{T} p_i^{(t+dt)} \tag{8}$$
where dt is the difference between the current time (acquisition time of the scan) and the timestamp of p_i, and $\mathbf{T} \in SE(3)$ is the estimated rigid body transformation between the pose at time t and the pose at time $t + dt$.
- Pointcloud filtering: the pointcloud is filtered to remove outliers outside of the specified range, using the same observation model for pointcloud measurements adopted for performing fault detection. Points characterized by very low values of reflectivity are also excluded.
- Pointcloud downsampling: pointcloud downsampling helps improving the speed as well as the accuracy of the pointcloud registration step. It consists in removing points in order to have uniform point spatial density.
- Pointcloud registration: the relative pose (translation t and rotation R) between consecutive pointclouds is estimated by using the ICP algorithm. More specifically, point correspondences are estimated by selecting the closest points using a point-to-plane metric. Then, the solution that minimizes the sum of squared Euclidean distances between corresponding points (with index n),
$$(\mathbf{R}^*, \mathbf{t}^*) = \arg\min_{(\mathbf{R},\mathbf{t})} \sum_n \|p_n - \mathbf{R} q_n - \mathbf{t}\|^2, \tag{9}$$
is computed. Both steps are iterated until the difference between iterations of the mean squared error becomes smaller than a pre-defined threshold.

a). Pose graph optimization

The objective of the pose graph optimization step is the estimation of a trajectory (in the form of a collection of poses) from relative pose measurements obtained by the ICP algorithm. Every node x_i in the graph represents a pose, and each edge between two nodes corresponds to a spatial constraint between them. When the train moves from x_i to $x_{(i+1)}$, the edge corresponds to the transformation estimated from the odometry measurement according to the IMU/GNSS module. An edge can also exist between two arbitrary nodes x_i and x_j observing the same landmark, and is inserted by creating virtual measurements about a pose x_j seen from x_i. The objective of pose graph optimization thus becomes finding the most-likely node configuration that minimizes the error introduced by the violations of these constraints. This is accomplished by using the Levenberg-Marquardt algorithm. At each iteration, the correction $\Delta\mathbf{x}$ applied to the state vector resulting from the concatenation of the estimated poses, $\mathbf{x}$, is given by solving:

$$(\mathbf{H} + \lambda\mathbf{I})\Delta\mathbf{x} = \mathbf{b} \tag{10}$$

where $\mathbf{H}$ is the Gauss-Newton approximation of the Hessian of the system, $\mathbf{b}$ is the vector of the residuals, and λ is a damping parameter.

Usually, a key component for optimizing the graph is given by the loop closure detection, which identifies loop closures (locations already visited by the vehicle) and adds constraints between their nodes. However, in our case there will be no loop closures because the train only moves forward and, at least for the total duration of one ride, does not pass by the same place twice. For this reason, we expect the matrix $\mathbf{H}$ to be very sparse, and use an appropriate sparse solver to find the solution to the system.

VI. CONCLUSIONS

In this paper we presented an innovative framework for enhancing the performances of the current positioning techniques in the railway scenario. We aim at going beyond the state-of-the-art thanks to the adoption of visual sensors that can complement the information already available from the GNSS and IMU modules. The proposed system relies on the latest advancements in signal and image processing for providing the functionalities of track discrimination, absolute positioning, and enhanced odometer, thus extending the capabilities currently offered by the information provided by GNSS and IMU. The selected visual sensors are characterized by their ability of working in different environmental conditions in order to grant their functionality in the largest number of operative cases. The architecture presented in this paper is the core of an ESA funded project and will be tested on a real scenario. Unfortunately, due to the current pandemic situation, the foreseen rail field trials are experiencing delays and for this reason they will be addressed in future works.

ACKNOWLEDGMENTS

This work was supported in part by the ESA (European Space Agency) within the framework of project Ref. 4000128511/19/NL/MP "VOLIERA (Video Odometry with Lidar and EGNSS for ERTMS Applications)".

REFERENCES

[1] W. Jiang, S. Chen, B. Cai, J. Wang, W. ShangGuan, and C. Rizos, "A Multi-Sensor Positioning Method-Based Train Localization System for Low Density Line," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 11, pp. 10 425–10 437, 2018.

[2] B. Allotta, P. D'Adamio, M. Malvezzi, L. Pugi, A. Ridolfi, A. Rindi, and G. Vettori, "An Innovative Localisation Algorithm for Railway Vehicles," *Vehicle System Dynamics*, vol. 52, no. 11, pp. 1443–1469, 2014.

[3] VOLIERA. [Online]. Available: https://navisp.esa.int/project/details/85/show

[4] Livox Horizon. [Online]. Available: https://www.livoxtech.com/horizon

[5] Basler Blaze. [Online]. Available: https://www.baslerweb.com/en/products/cameras/3d-cameras/basler-blaze/

[6] Nerian Karmin3. [Online]. Available: https://nerian.com/products/karmin3-3d-stereo-camera/

[7] S. Oude Elberink, K. Khoshelham, M. Arastounia, and D. Diaz Benito, "Rail Track Detection and Modelling in Mobile Laser Scanner Data," *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. II-5/W2, pp. 223–228, 2013. [Online]. Available: https://www.isprs-ann-photogramm-remote-sens-spatial-inf-sci.net/II-5-W2/223/2013/

[8] M. A. Fischler and R. C. Bolles, "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[9] S. Baldoni, F. Battisti, M. Brizzi, and A. Neri, "A Hybrid Position Estimation Framework based on GNSS and Visual Sensor Fusion," in *2020 IEEE/ION Position, Location and Navigation Symposium (PLANS)*, 2020, pp. 979–986.

[10] P. Groves, *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems, Second Edition*. Artech, 2013.

[11] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2980–2988.

[12] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," 2016.

[13] C. Qi, H. Su, K. Mo, and L. Guibas, "PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 77–85, 2017.

[14] C. Garg. [Online]. Available: https://github.com/cgarg92/Stereo-visual-odometry

[15] A. Howard, "Real-Time Stereo Visual Odometry for Autonomous Ground Vehicles," in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2008, pp. 3946–3952.

[16] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[17] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-Up Robust Features (SURF)," *Computer vision and image understanding*, vol. 110, no. 3, pp. 346–359, 2008.

[18] E. Rosten and T. Drummond, "Machine Learning for High-Speed Corner Detection," in *ECCV*, 2006.

[19] J. Shi and Tomasi, "Good Features to Track," in *1994 Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 1994, pp. 593–600.

[20] Y. Chen and G. Medioni, "Object Modeling by Registration of Multiple Range Images," in *Proceedings. 1991 IEEE International Conference on Robotics and Automation*, 1991, pp. 2724–2729 vol.3.

[21] P. Besl and N. D. McKay, "A Method for Registration of 3-D Shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 239–256, 1992.