Pole-based Vehicle Localization with Vector Maps: A Camera-LiDAR Comparative Study

Maxime Noizet, Philippe Xu, Philippe Bonnifait

Pole-based Vehicle Localization with Vector Maps: A Camera-LiDAR Comparative Study

Maxime Noizet[1] Philippe Xu[1] Philippe Bonnifait[1]

***Abstract*— For autonomous navigation, accurate localization with respect to a map is needed. In urban environments, infrastructure such as buildings or bridges cause major difficulties to Global Navigation Satellite Systems (GNSS) and, despite advances in inertial navigation, it is necessary to support them with other sources of exteroceptive information. In road environments, many common furniture such as traffic signs, traffic lights and street lights take the form of poles. By geo-referencing these features in vector maps, they can be used within a localization filter that includes a detection pipeline and a data association method. Poles, having discriminative vertical structures, can be extracted from 3D geometric information using LiDAR sensors. Alternatively, deep neural networks can be employed to detect them from monocular cameras. The lack of depth information induces challenges in associating camera detections with map features. Yet, multi-camera integration provides a cost-efficient solution. This paper quantitatively evaluates the efficacy of these approaches in terms of localization. It introduces a real-time method for camera-based pole detection using a lightweight neural network trained on automatically annotated images. The proposed methods' efficiency is assessed on a challenging sequence with a vector map. The results highlight the high accuracy of the vision-based approach in open road conditions.**

I. Introduction

In the field of autonomous driving, achieving a reliable and accurate localization solution is crucial to ensure safe and efficient navigation when using a navigation map. For example, localization is essential for tasks such as planning, crossing intersections, aiding perception, cooperative navigation, etc. Depending on the context and the requirements of the localization task, this can be particularly challenging. Even on rather favourable operational domains like highways, non-differential multi-constellation GNSS (Global Navigation Satellite Systems) aided by Dead-Reckoning (DR) sensors is insufficient when lane-level positioning is needed [1].

To improve the localization performance, exteroceptive sensors such as LiDARs or cameras can be added to handle the mentioned limitations in complex environments. In this case, a vector map can be an efficient and scalable means to manage geo-referenced features such as traffic signs, lane markings or other road features. In this paper, we focus on High-Definition vector maps (HD maps) with a cm-level accuracy.

To reach lane-level positioning, lane markings and curbs are now very well detected by cameras and associated with HD maps to improve cross-track accuracy and integrity [2], [3]. Yet, road information for localization is sensitive to environmental factors as degradation, occlusion, and variation which can lead to unreliable and inconsistent measurements. Besides, it requires an HD map containing a geometric, e.g., polyline-based, description of all road markings, which can be costly to produce and maintain.

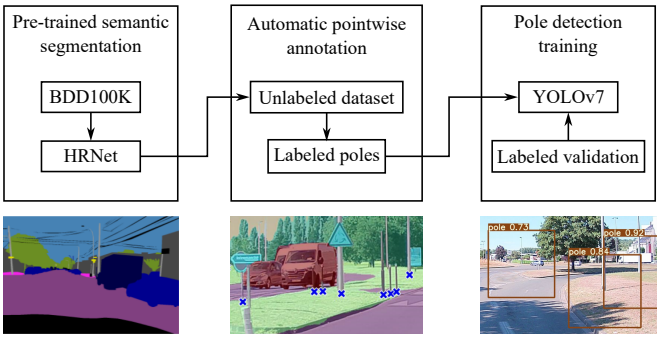


Fig. 1: Image-based poles dectector training. A pre-trained semantic segmentation network is used to annotate an unlabeled custom dataset. The pixel-wise annotations are transformed into pointwise labels at the bases of the poles. A YOLOv7 is trained using bounding boxes centered at the pointwise labels. A small amount of manually labeled data are used for validation.

There are other widespread road infrastructure elements that can be used as additional sources of information as they are quite easy to detect. For example, using a LiDAR point cloud, traffic signs can be easily extracted with intensity filtering and used to improve localization [4]. Traffic signs represent only a small part of all the features available in a road environment. They belong to a broader widespread class which is that of poles or vertical signage including in addition traffic lights and streetlights. They can provide absolute localization information when detected by on-board sensors. They have shown to improve deeply localization performance [5], [6].

However, LiDAR sensors have some limitations due to the sparse nature of point clouds. Detecting fine or distant structures can be complex, and the cost of such a sensor can also be a barrier for large-scale deployment. In contrast, monocular cameras do not have these drawbacks, although their field of view is necessarily narrower and they are more sensitive to illumination conditions.

Many methods can be applied to detect objects in images captured by cameras. To the best of our knowledge, there is limited research on pole detection in images [7], [8], particularly from a localization perspective [9]. In a previous work [10], we proposed a pole bases detector using a neural

[1]The authors are with the Université de technologie de Compiègne, CNRS, Heudiasyc, France. name.surname@hds.utc.fr

network trained on automatically annotated data using HD-maps.

Monocular cameras only provide angular information resulting in a bearing-only localization problem. Bearing information has been used in various studies as in tracking applications in aviation or submarine fields, but also in vehicle localization. Bearing-only Simultaneous Localization And Mapping (SLAM) have been studied [11]. Some camera-based methods [12], [13] have been proposed where the visual features used are low-level features such as SIFT [14] or Harris-Laplace points [15]. In [16], an omni-camera and four landmarks are used to localize an automated agricultural vehicle with distinguishable landmarks optimally placed.

Pole-based localization is very challenging for data association due to the non-discernability of road features. Incorrect associations of pole detections with vector map features can therefore lead to poor localization. Moreover, relying solely on bearing information makes it difficult to accurately estimate a vehicle's pose since angular measurements are made relatively to the heading of the vehicle.

In this paper, the objective is to study how a localization system based on GNSS and DR sensors can be improved by incorporating pole-like feature detections that are matched with a vector map. We consider detections obtained from both a LiDAR and a multi-camera system, and compare their performance in terms of accuracy improvements.

The article is organized as follows. In Section II, we propose a monocular image-based pole detection method using a semantic segmentation network to generate pseudo labels to train an object detector. The LiDAR-based geometric pole detector is also described. The pole-based localization framework is given in Section III. Finally, experimental results using real data are detailed and analyzed in Section IV. Finally, Section V presents conclusions and future work.

II. Pole detection

A. Problem statement

The thickness of pole-like features with respect to the scale of an urban HD vector map implies that they are usually mapped as points. The coordinates of such a point typically represent the base of a pole at the ground level. In the rest of the paper, we consider the general case where the poles are mapped as 2D points without further information about their types, their height or geometry. From the map perspective, all the poles are considered as being indistinguishable. The vector map is therefore a set of georeferenced landmarks

$$\boldsymbol{\mathcal{M}} = \left\{ {}^{(\mathrm{O})}\boldsymbol{m}_j \in \mathbb{R}^2 \,\middle|\, j = 1, \dots \right\} \tag{1}$$

where each map feature ${}^{(\mathrm{O})}\boldsymbol{m}_j$ is a 2D point expressed in a local working frame (O) using East-North-Up (ENU) coordinates. For clarity, in the rest of the paper, the left exponent will be omitted when the coordinates are expressed in the (O) frame.

The aim is to detect these features using cameras and LiDAR. In the image frame, it consists in detecting the pixel coordinates of the bases of the poles. In the LiDAR point cloud, it comes to compute the 2D coordinates of the projections of the poles onto the ground plane.

B. Camera-based detection

The visual characteristics of poles make them ill suited to be detected from an object-based bounding box point of view. Indeed, poles are thin and are often truncated in the image field-of-view. Therefore, poles are most of the time considered at the pixel-level within a semantic segmentation framework. The main drawback of dense semantic segmentation is its computational cost compared to modern object detection such as YOLO [17].

In our prior work [10], we have demonstrated how to formalize the detection of poles in images with an object detection pipeline using bounding boxes centered at the bases of the poles, *i.e.*, the contact point between the poles and the ground. The labels for the images were automatically generated by the joint use of a vector map and a LiDAR in order to compute the projection of the map features onto the image frame. For this purpose, it was necessary to estimate the ground plane as well as determining whether or not the feature was visible, *e.g.*, not occluded by some obstacles. One of the limitations of this solution is that it requires a LiDAR in addition to the cameras as well as a localization ground truth.

We propose to extend the solution in [10] by making use of a pre-trained semantic segmentation neural network to generate pointwise annotation. The process follows the steps pictured in Fig. 1:

1) Train a semantic segmentation neural network on a labeled dataset that includes pole-like classes;
2) Use the neural network to generate pixel-wise pseudo-labels in an unlabeled dataset and compute point-wise labels at the bases of the poles;
3) Use bounding boxes centered at the poles to train an object detector using a small dataset of manually annotated images for validation.

For the first step, we use the High-Resolution Network (HRNet) proposed in [18] with a model pre-trained on the BDD100K dataset [19]. Among multiple networks trained on BDD100K dataset, it is one of the most effective for segmenting pixels related to pole-like classes. In this dataset, the pole-like features correspond to three classes, namely "pole", "traffic sign" and "traffic light".

For the second step, all connected pole-like pixels are grouped and for such a cluster the lowest pixel lying on ground pixels, if it exists, is considered as the pole base and is labeled as such. Contrary to what has been proposed in [10] the pixel semantic labels are not from the ground truth but are pseudo-labels computed from a neural network.

Finally, for the last step, we transform the pointwise labels into squared fixed-sized bounding boxes centered on the labeled points. We then feed these data to a YOLOv7 [20] object detector as in [10] while tuning the size of the bounding boxes on a validation set containing a small amount manually annotated images. At the inference stage, the detected bounding boxes are converted back into points by

(a) Front RGB image (b) Left grayscale image (c) Right grayscale image

Fig. 2: Examples of segmented images and the obtained annotations (blue crosses).

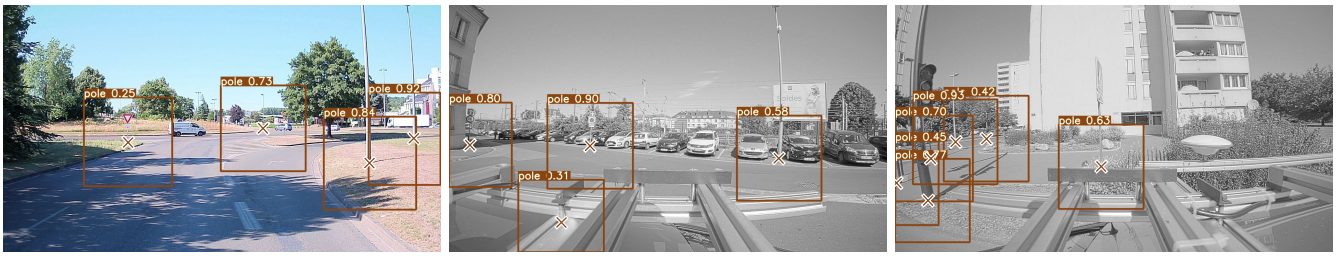


Fig. 3: Examples of detections obtained from YOLOv7-based pole detectors on RGB and grayscale images. Each bounding box is displayed with its detection score and its center corresponding to a pole base is highlighted by a cross. For the grayscale side cameras, a final filtering is applied to remove detections on the vehicle roof.

computing the center point. For a given image I at time k, the output of the detection is a set of measurements:

$$ {}^{(\mathrm{I})}\boldsymbol{Y}_k^{\mathrm{I}} = \left\{ {}^{(\mathrm{I})}\boldsymbol{y}_{k,i}^{\mathrm{I}} = (u_{k,i}, v_{k,i}) \middle| \, i = 1, \dots \right\}, \qquad (2) $$

where each measurement ${}^{(\mathrm{I})}\boldsymbol{y}_{k,i}^{\mathrm{I}}$ is the pixel coordinates u, v of a pole base expressed in the image frame (I) of the camera.

We apply this strategy to a multi-camera system composed of a front color camera and two wide-angle grayscale cameras directed on the sides. Examples of segmented images with the annotations obtained are illustrated in Fig. 2. Even though the images in the BDD100K dataset are more similar to the color camera, the performance on the wide-angle grayscale ones are reasonable. Fig. 3 pictures the detection results on the three cameras. For the side cameras, a final filtering is applied to remove detections on the vehicle roof.

C. LiDAR-based detection

In a LiDAR point-cloud, each point is characterized by its Cartesian position in the vehicle frame. Consequently pole-like features can be extracted using geometric-based techniques. Firstly, ground points are removed and remaining points are grouped into clusters using the method proposed by Zermas et al. [21]. Then, each obtained cluster is classified as a pole or not using a Principal Component Analysis (PCA) strategy. For each cluster, the principal components characterized by the eigenvectors v_1, v_2, v_3 and eigenvalues λ_1, λ_2, λ_3 of the covariance matrix sorted in descending order are computed. Then, thresholds are defined to consider a cluster as a pole:

- Linearity $l = (\lambda_1 - \lambda_2)/\lambda_1$: quantifies the predominance of the main component compared to the others. A pole should have a high linearity.

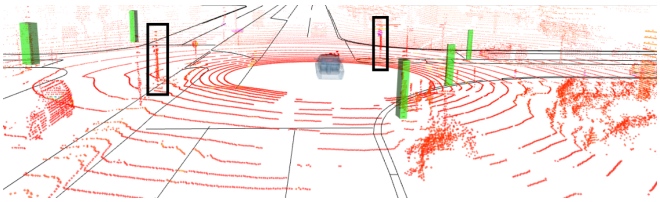


Fig. 4: Examples of pole detections obtained from LiDAR. The bounding boxes of detected poles are visible in green. Two examples of missed detections are highlighted with black rectangles. Other examples are visible in point cloud distribution.

- Orientation β: quantifies the angle between the z-axis and the main component direction v_1. A pole should be vertical, *i.e.*, have a low value for β.
- Height h: a pole is typically a tall cluster.
- Thickness t: a pole is typically a thin cluster.

For each of these quantities, we define a threshold and we consider a cluster as being a pole if the following condition is met:

$$ (l > l_{\min}) \,\&\, (\beta < \beta_{\max}) \,\&\, (h > h_{\min}) \,\&\, (t < t_{\max}) \qquad (3) $$

The detection output from the LiDAR L is a set of 2D measurements

$$ {}^{(\mathrm{L})}\boldsymbol{Y}_k^{\mathrm{L}} = \left\{ {}^{(\mathrm{L})}\boldsymbol{y}_{k,i}^{\mathrm{L}} = \left({}^{(\mathrm{L})}x_{k,i}^{\mathrm{L}}, {}^{(\mathrm{L})}y_{k,i}^{\mathrm{L}} \right) \middle| \, i = 1, \dots \right\}, \qquad (4) $$

where each measurement ${}^{(\mathrm{L})}\boldsymbol{y}_{k,i}^{\mathrm{L}}$ corresponds to the 2D coordinates of the centroid of the cluster i expressed in the LiDAR frame (L). An example of poles detection using LiDAR data is illustrated in Fig. 4.

III. POLE-BASED LOCALIZATION FILTER

We build our localization solution using a standard extended Kalman filter formalism. The system uses GNSS, wheel speed sensors, a gyro, a LiDAR and multiple cameras for pole detection and a vector HD map. At a time k, the state vector $\boldsymbol{x}_k$ is expressed as follows:

$$\boldsymbol{x}_k = \left[x_k, y_k, \theta_k, v_k, \dot{\theta}_k, b_{k,x}, b_{k,y}\right]^\top \tag{5}$$

The component $\boldsymbol{q}_k = (x_k, y_k, \theta_k)$ is the vehicle pose, *i.e.*, position and heading, defined at the center of the vehicle rear axle. The components v_k and $\dot{\theta}_k$ correspond to the longitudinal speed and the yaw rate, respectively. To handle GNSS bias exhibited by the receiver, a random constant GNSS 2D bias $(b_{k,x}, b_{k,y})$ is added to the state vector.

A. GNSS and DR

The GNSS measurements $\boldsymbol{z}_k^{\mathrm{G}}$ provide the 2D coordinates of the antenna, its observation model is derived as follows:

$$\boldsymbol{z}_k^{\mathrm{G}} = \begin{bmatrix} x_k + b_{k,x} \\ y_k + b_{k,y} \end{bmatrix} + \begin{bmatrix} \cos\theta_k & -\sin\theta_k \\ \sin\theta_k & \cos\theta_k \end{bmatrix} \begin{bmatrix} t_x \\ t_y \end{bmatrix} + \boldsymbol{\beta}_k^{\mathrm{G}} \tag{6}$$

where (t_x, t_y) is the antenna lever arm with respect to the vehicle frame and $\boldsymbol{\beta}_k^{\mathrm{G}}$ the GNSS observation noise.

For the dead reckoning the observation models of the left and right rear wheel speeds, $z_k^{\mathrm{W}_l}$ and $z_k^{\mathrm{W}_r}$, are given in $\mathrm{m{\cdot}s^{-1}}$ and expressed as follows:

$$z_k^{\mathrm{W}_l} = v_k - \frac{\ell}{2}\dot{\theta}_k + \beta_k^{\mathrm{W}_l}\ , \quad z_k^{\mathrm{W}_r} = v_k + \frac{\ell}{2}\dot{\theta}_k + \beta_k^{\mathrm{W}_r} \tag{7}$$

where ℓ the distance separating the two wheels and $\beta_k^{\mathrm{W}_l}$, $\beta_k^{\mathrm{W}_r}$ the observation noises. Finally, the gyro provides a straightforward measurement of the yaw rate $z_k^{\mathrm{Y}} = \dot{\theta}_k + \beta_k^{\mathrm{Y}}$.

B. Poles measurements

To build the observation model for the poles detected by the LiDAR or the cameras, two steps are necessary. First, the detections from the LiDAR and the cameras need to be expressed in a common space with respect to the map features. And second, the detections and the map features need to be associated to each other.

For the LiDAR, a detection is represented by a 2D point similar to the map representation. We can either move the map features from the (O) frame to the LiDAR (L) frame or the opposite for solving the data-association. Because there are often less detections than map features, the latter is less computationally demanding.

At a given time k, an estimate of the vehicle pose $\hat{\boldsymbol{q}}_{k|k-1}$ is predicted from the previous state estimate $\hat{\boldsymbol{x}}_{k-1}$. This pose estimate is then used to transform the LiDAR detection set ${}^{(\mathrm{L})}\boldsymbol{Y}_k^{\mathrm{L}}$ from the (L) frame to the (O) frame: $\boldsymbol{Y}_k^{\mathrm{L}}$.

For the pole detection in the image frame, because of the lack of depth information from monocular cameras, it is not possible to compute the 2D coordinates of the detected poles in the map frame. Instead, we use a bearing only approach to encode the poles detection. Given the camera intrinsic calibration parameters, the image detection set ${}^{(\mathrm{I})}\boldsymbol{Y}_k^{\mathrm{I}}$ is transformed into a set of bearing angles expressed in the camera frame (C):

$${}^{(\mathrm{C})}\boldsymbol{Y}_k^{\alpha} = \left\{ {}^{(\mathrm{C})}y_{k,i}^{\alpha} = \alpha_{k,i} \in [-\pi;\pi) \,\middle|\, i = 1, \dots \right\} \tag{8}$$

where $\alpha_{k,i}$ corresponds to the angle of the i-th detection with respect to the direction pointed by the camera which is aligned with the vehicle heading in the case of the front color camera.

Contrary to the LiDAR case, for the camera, it is the map features that are transformed into the camera frame. Given $\hat{\boldsymbol{q}}_{k|k-1}$, the map features within a limited radius around the pose position, are transformed into the camera frame and their relative angles with respect to the camera are computed. This leads to a camera map composed of angles relative to the camera frame:

$${}^{(\mathrm{C})}\boldsymbol{\mathcal{M}}_k^{\alpha} = \left\{ {}^{(\mathrm{C})}m_{k,j}^{\alpha} = \alpha_{k,j} \in [-\pi;\pi) \,\middle|\, j = 1, \dots \right\} \tag{9}$$

The same process is done for each of the three cameras.

C. From measurements to map-matched observations

Map-matching consists in associating the measurements of the detected features with landmarks retrieved from the map. Because we have considered the features to be indistinguishable, we use geometric distances to associate the data. The Mahalanobis distance can be used to measure the proximity of a LiDAR measurement $\boldsymbol{y}_{k,i}^{\mathrm{L}}$ and a map feature $\boldsymbol{m}_j$ as follows:

$$D_{k,i,j}^{\mathrm{L}} = \sqrt{(\boldsymbol{m}_j - \boldsymbol{y}_{k,i}^{\mathrm{L}})^\top {R_{k,i}^{\mathrm{L}}}^{-1} (\boldsymbol{m}_j - \boldsymbol{y}_{k,i}^{\mathrm{L}})^\top} \tag{10}$$

where $R_{k,i}^{\mathrm{L}}$ is the covariance matrix associated to the LiDAR measurement $\boldsymbol{y}_{k,i}^{\mathrm{L}}$ computed from the covariance matrix of the pose estimate $\hat{\boldsymbol{q}}_{k|k-1}$.

In the camera case, we manipulate angular quantities, for a camera measurement ${}^{(\mathrm{C})}y_{k,i}^{\alpha} \in [-\pi;\pi)$ and a map feature ${}^{(\mathrm{C})}m_{k,j}^{\alpha} \in [-\pi;\pi)$, their difference $\delta_{k,i,j}$ is mapped onto the $[-\pi;\pi)$ interval as follows:

$$\delta_{k,i,j} = \left[\left({}^{(\mathrm{C})}m_{k,j}^{\alpha} - {}^{(\mathrm{C})}y_{k,i}^{\alpha} + \pi\right) \bmod 2\pi\right] - \pi \tag{11}$$

where mod 2π is the modulo operator providing the result within $[0; 2\pi)$. The final distance is then defined as the squared difference

$$D_{k,i,j}^{\mathrm{C}} = \delta_{k,i,j}^2 \tag{12}$$

The map-matching problem is solved as an assignment problem using the Hungarian method [22]. This method finds in a polynomial time the optimal sets of pairs

$$\boldsymbol{Z}_k^{\mathrm{L}} = \left\{ \boldsymbol{z}_{k,i,j}^{\mathrm{L}} = \left(\boldsymbol{y}_{k,i}^{\mathrm{L}},\ \boldsymbol{m}_j\right) \right\} \tag{13}$$

$$\boldsymbol{Z}_k^{\mathrm{C}} = \left\{ \boldsymbol{z}_{k,i,j}^{\mathrm{C}} = \left({}^{(\mathrm{C})}y_{k,i}^{\alpha},\ {}^{(\mathrm{C})}m_{k,j}^{\alpha} \right) \right\} \tag{14}$$

that minimize the sum of the associated distances

$$\boldsymbol{Z}_k^{\mathrm{L}} = \underset{\boldsymbol{y}_{k,i}^{\mathrm{L}} \in \boldsymbol{Y}_k^{\mathrm{L}}, \boldsymbol{m}_j \in \boldsymbol{\mathcal{M}}}{\arg\min} \sum_{i,j} D_{k,i,j}^{\mathrm{L}} \tag{15}$$

$$\boldsymbol{Z}_k^{\mathrm{C}} = \underset{{}^{(\mathrm{C})}y_{k,i}^{\alpha} \in {}^{(\mathrm{C})}\boldsymbol{Y}_k^{\alpha}, {}^{(\mathrm{C})}m_{k,j}^{\alpha} \in {}^{(\mathrm{C})}\boldsymbol{\mathcal{M}}_k^{\alpha}}{\arg\min} \sum_{i,j} D_{k,i,j}^{\mathrm{C}} \tag{16}$$

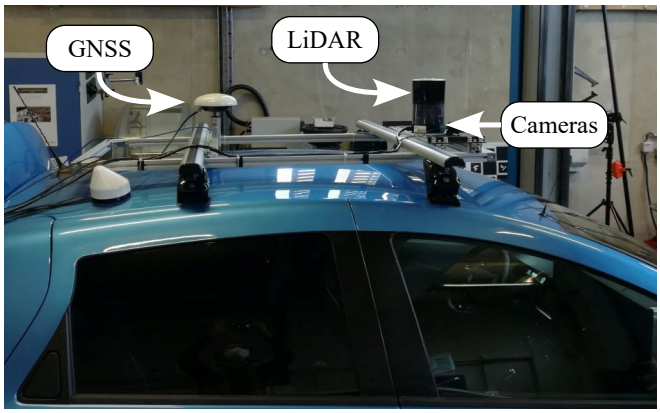


Fig. 5: Roof of the experimental Renault ZOE vehicle equipped showing the GNSS antenna and the Hesai Pandora sensor combining a LiDAR with several cameras.

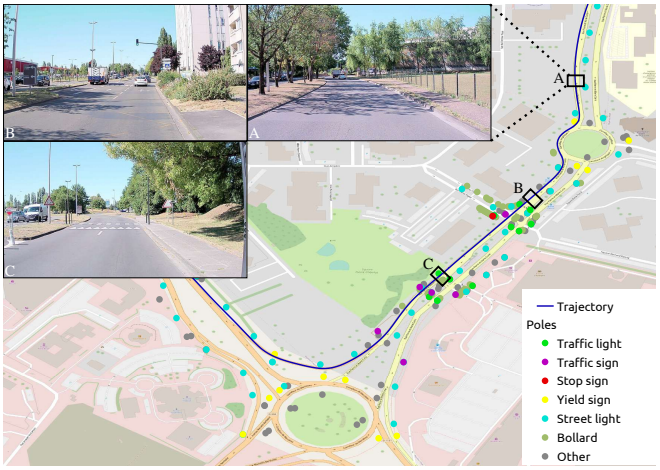


Fig. 6: Experimental trajectory. The mapped pole-like features are displayed. The pictures illustrate the experimental conditions.

under the constraint that at most one measurement can be associated to a map feature and conversely.

Once the map-matching step is done, the observations from the LiDAR and the cameras are injected into the localization filter.

IV. Experimental results

A. Experimental setup

The experiments were conducted with a Renault Zoe experimental vehicle equipped with several sensors:

- Wheel speed sensors [100 Hz]
- Septentrio mosaic X5 GNSS receiver with an automotive grade IMU [1 Hz]
- Hesai Pandora sensor combining a 40-layer LiDAR and 5 monocular cameras (4 grayscale cameras with a horizontal FOV of 129° and one front RGB camera with a vertical FOV of 52°). The front and back grayscale cameras were not used in this study [10 Hz]
- Novatel SPAN-CPT GNSS/IMU with post-processed PPK computations for localization ground truth [50 Hz]

The combination of sensors tested are:

- **GNSS+DR** only uses receiver, wheel speeds sensors and yaw rate.
- **Front** uses GNSS+DR and the bearing measurements obtained from Pandora front color camera.
- **Left/Right** uses GNSS+DR and the bearing measurements obtained from Pandora left and right grayscale cameras.
- **All cameras** uses GNSS+DR and the bearing measurements obtained from Pandora left and right grayscale cameras and front color camera.
- **LiDAR** uses GNSS+DR and the pole measurements obtained from Pandora LiDAR.

Fig. 5 shows the roof of the experimental vehicle with the Hesai Pandora sensor. We evaluated our pole-based localization framework on a 600 m-long section visible in Fig. 6 extracted from datasets covering the city of Compiègne, France.

B. Results and discussion

The table I summarizes the Root Mean Square (RMS) errors obtained from all various combinations on multiple datasets.

It is worth noting that the datasets were carefully acquired ensuring similar weather and traffic conditions for the majority of sequences, with the exception of the 05-19 sequence, which experienced higher traffic density. Then, the performance gap of detection methods between sequences due to variation in driving conditions is minimized.

Different behaviors occurred during these sequences. Firstly, for the 05-10 and 07-06 sequences, the performances obtained with all cameras are similar to the LiDAR. Yet, on the 05-10 sequence, the Left/Right is the combination obtaining the best results due to Front degradation which also affects the all-camera combination. On the 05-19 and 06-28 sequences, the LiDAR reached better performance than any camera combination. This is probably due to wrong associations when using camera measurements. Finally, on the 05-24 sequence, using all cameras is better than the LiDAR due also to miss-associations when using LiDAR measurements.

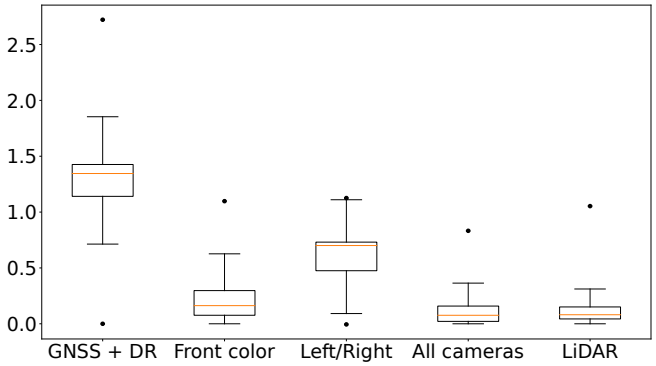
Globally, the position obtained using only GNSS and DR sensors is deeply improved except when miss-associations with the LiDAR occurred on the 05-24 sequence. The achieved performance is comparable with the LiDAR performance and adding all the cameras together instead of using only the Front or the Left/Right cameras improves globally the localization performance.

Each camera appears to contribute more to a specific component of the localization error. In fact, when focusing on cross-track (CT) errors summarized in Fig. 7a for the 07-06 sequence. All cameras performs better than LiDAR combination and this is, as expected, mainly due to the front color camera leading to an average error of less than 40 cm.

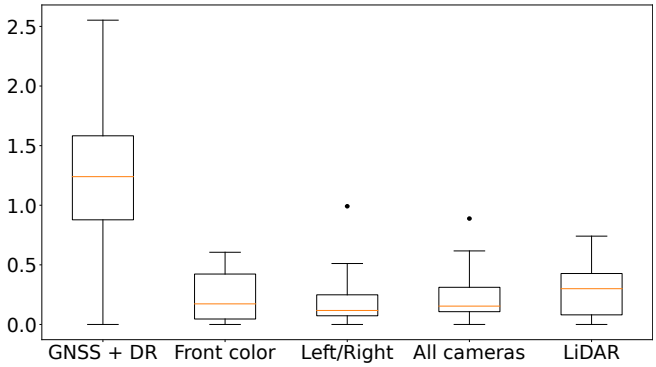
In terms of along-track (AT) errors, as visible in Fig. 7b, even if Front is capable of improving AT accuracy,

TABLE I: RMS obtained for several combination of sensors for multiple datasets acquired under similar weather and traffic conditions

Date	GNSS+DR	Front	Left/Right	All cameras	LiDAR
05-10	1.13	0.82	**0.40**	0.46	0.53
05-19	2.43	1.11	2.29	0.96	**0.39**
05-24	2.32	0.92	1.03	**0.70**	3.16
06-28	2.29	1.13	0.88	0.73	**0.36**
07-06	1.95	0.48	0.72	**0.39**	0.44



(a) Cross-track errors.



(b) Along-track errors.

Fig. 7: Boxplots of the cross-track and along-track errors with different sensor setup obtained on the 07-06 sequence.

Fig. 8: Biases in ENU frame obtained using the all cameras solution on the 07-06 sequence. Biases on x and y axes are respectively on top and bottom. The observations timestamps provided by the different cameras are summarized in the middle (Front camera in green, left camera in yellow and right camera in red).

main improvement comes from the Left/Right combination, improving deeply all cameras solution, although extreme error values are still higher than LiDAR errors.

When focusing on biases estimation obtained during the 07-06 sequence using all cameras as shown in Fig. 8, the filter seems capable of estimating them even if some jumps are visible on the curves. Some of these jumps seems to be due to miss-associations between map features and detected poles. For example, a jump on b_y occurs around 30s after reception of right camera observations. A jump on b_x happens at the end of the sequence around 110s and seems to be correlated with reception of front camera observations. Because the map-matching algorithm rely on an initial pose estimate, it is essential to correct the GNSS bias during the estimation.

Moreover, as shown in this figure, the results are primarily driven by the input from the front and left cameras. The right camera, detects fewer elements due to the majority of features being on the left side of the vehicle in this section. Removing the right camera would have had minimal impact on the overall solution.

Overall, our different detection methods face distinct challenges depending on the approach used, directly impacting localization accuracy. LiDAR may experience limitations in performance due to its simplistic detection process relying on multiple thresholding during a PCA procedure, which can result in a notable number of false positives and negatives.

On the other hand, cameras also encounter challenges in detection. These challenges primarily arise from the quality of training, particularly for the left and right cameras. The performance of the camera's detector is heavily influenced by the segmentation neural network used for image annotation, which is not optimal for wide-angle grayscale images.

V. Conclusion

In this paper, we proposed to enhance a localization system based on GNSS and Dead Reckoning sensors by integrating pole-like feature detections and associations with a vector map. We proposed two different detection approaches using different sensors: one based on LiDAR sensor data where geometric filtering techniques are used and one based on object detection in camera images using automatically annotated data obtained from an image segmentation network.

We compared the performance of LiDAR and multi-camera integration in terms of localization accuracy improvements on a complex peri-urban section containing multiple road features, mapped or not, and potential false positive detection sources.

We showed that adding a front camera capable of detecting pole-like features can improve cross track positioning deeply. This result was expected due to similarities with lane-marking based localization for lane-level positioning. Adding side cameras improves greatly the along-track positioning. Consequently, the combination of all these cameras provides localization performance similar to LiDAR integration.

These results suggest that a multi-camera system is promising to replace or complete a LiDAR system, although further exploration is required to assess the robustness of the association process with the map, mitigate potential miss-associations and guarantee the integrity of the localization solution.

In future work, our perception pipeline will be improved to enhance detection capabilities of the sensors and avoid false positives. A comprehensive study will be undertaken to investigate the various factors influencing detection performance, including weather and traffic conditions, as well as the inherent characteristics of the detection methods themselves. Then, a particular attention will be given to the robustness of the data association and the estimation process. A study will be conducted to investigate the complementarity of LiDAR and cameras, the data association of multiple sources regarding the same features, and the benefits of such a system for localization.

Acknowledgment

This work has been funded by the European project ERASMO [23] (GSA/GRANT/03/2018) in the framework of the SIVALab laboratory between Renault and Heudiasyc.

References

[1] J. Laconte, A. Kasmi, R. Aufrère, M. Vaidis, and R. Chapuis, "A Survey of Localization Methods for Autonomous Vehicles in Highway Scenarios," *Sensors*, vol. 22, no. 1, p. 247, Dec. 2021.

[2] G. Frisch, P. Xu, and E. Stawiarski, "High integrity lane level localization using multiple lane markings detection and horizontal protection levels," in *15th International Conference on Control, Automation, Robotics and Vision*, 2018, pp. 1496–1501.

[3] J. Al Hage, P. Xu, and P. Bonnifait, "High integrity localization with multi-lane camera measurements," in *IEEE Intelligent Vehicles Symposium*, 2019, pp. 1232–1238.

[4] F. Ghallabi, G. El-Haj-Shhade, M.-A. Mittet, and F. Nashashibi, "LIDAR-Based road signs detection For Vehicle Localization in an HD Map," in *IEEE Intelligent Vehicles Symposium*, Paris, France, June 2019, pp. 1484–1490.

[5] L. Li, M. Yang, L. Weng, and C. Wang, "Robust localization for intelligent vehicles based on pole-like features using the point cloud," *IEEE Transactions on Automation Science and Engineering*, pp. 1–14, 2021.

[6] R. Spangenberg, D. Goehring, and R. Rojas, "Pole-based localization for autonomous vehicles in urban scenarios," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Daejeon, South Korea, Oct. 2016, pp. 2161–2166.

[7] H. Sharma, V. Adithya, T. Dutta, and P. Balamuralidhar, "Image Analysis-Based Automatic Utility Pole Detection for Remote Surveillance," in *International Conference on Digital Image Computing: Techniques and Applications*, Adelaide, SA, Nov. 2015, pp. 1–7.

[8] W. Zhang, C. Witharana, W. Li, C. Zhang, X. Li, and J. Parent, "Using Deep Learning to Identify Utility Poles with Crossarms and Estimate Their Locations from Google Street View Images," *Sensors*, vol. 18, no. 8, p. 2484, Aug. 2018.

[9] B. H. G. Barbosa, N. P. Bhatt, A. Khajepour, and E. Hashemi, "Soft constrained autonomous vehicle navigation using gaussian processes and instance segmentation," *ArXiv*, vol. abs/2101.06901, 2021.

[10] B. Missaoui, M. Noizet, and P. Xu, "Map-aided annotation for pole base detection," in *IEEE Intelligent Vehicles Symposium Workshop*, June 2023.

[11] K. Bekris, M. Click, and E. Kavraki, "Evaluation of algorithms for bearing-only SLAM," in *IEEE International Conference on Robotics and Automation*, Orlando, FL, 2006, pp. 1937–1943.

[12] P. Jensfelt, D. Kragic, J. Folkesson, and M. Bjorkman, "A framework for vision based bearing only 3D SLAM," in *IEEE International Conference on Robotics and Automation*, Orlando, FL, USA, 2006.

[13] T. Lemaire, C. Berger, I.-K. Jung, and S. Lacroix, "Vision-Based SLAM: Stereo and Monocular Approaches," *International Journal of Computer Vision*, vol. 74, no. 3, July 2007.

[14] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the seventh IEEE international conference on computer vision*, vol. 2. Ieee, 1999, pp. 1150–1157.

[15] C. G. Harris and M. J. Stephens, "A combined corner and edge detector," in *Alvey Vision Conference*, 1988.

[16] Y. Hoshino, L. Yang, and S. Suzuki, "Self-localization method using a single omni-directional camera based on landmark positions and arrangement," in *IEEE/SICE International Symposium on System Integration*, 2016, pp. 580–585.

[17] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.

[18] J. Wang, K. Sun, T. Cheng, B. Jiang, C. Deng, Y. Zhao, D. Liu, Y. Mu, M. Tan, X. Wang, W. Liu, and B. Xiao, "Deep high-resolution representation learning for visual recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 10, pp. 3349–3364, oct 2021.

[19] F. Yu, H. Chen, X. Wang, W. Xian, Y. Chen, F. Liu, V. Madhavan, and T. Darrell, "BDD100K: A diverse driving dataset for heterogeneous multitask learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2020, pp. 2636–2645.

[20] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," *arXiv preprint:2207.02696*, 2022.

[21] D. Zermas, I. Izzat, and N. Papanikolopoulos, "Fast segmentation of 3d point clouds: A paradigm on lidar data for autonomous vehicle applications," in *IEEE International Conference on Robotics and Automation*, 2017, pp. 5067–5073.

[22] H. W. Kuhn, "The hungarian method for the assignment problem," *Naval research logistics quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.

[23] L. Vilalta Estrada, C. Muñoz García, E. Domínguez Tijero, M. Noizet, P. Xu, S. Y. Voon, S. Guerassimov, and W. W. Cox, "ERASMO – Enhanced Receiver for AutonomouS MObility," in *Proceedings of the 15th ITS European Congress*, May 2023.