

## Research Article

# Laser-Based Obstacle Detection at Railway Level Crossings

Vítor Amaral,<sup>1</sup> Francisco Marques,<sup>2</sup> André Lourenço,<sup>2</sup> José Barata,<sup>2</sup> and Pedro Santana<sup>1,3</sup>

<sup>1</sup>ISCTE-Instituto Universitario de Lisboa (ISCTE-IUL), 1649-026 Lisboa, Portugal

<sup>2</sup>CTS-UNINOVA, Universidade Nova de Lisboa (UNL), 2829-516 Caparica, Portugal

<sup>3</sup>Instituto de Telecomunicações (IT), 1049-001 Lisboa, Portugal

Correspondence should be addressed to Pedro Santana; [pedro.santana@iscte.pt](mailto:pedro.santana@iscte.pt)

Received 6 September 2015; Accepted 12 January 2016

Academic Editor: Jesus Corres

Copyright © 2016 Vítor Amaral et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper presents a system for obstacle detection in railway level crossings from 3D point clouds acquired with tilting 2D laser scanners. Although large obstacles in railway level crossings are detectable with current solutions, the detection of small obstacles remains an open problem. By relying on a tilting laser scanner, the proposed system is able to acquire highly dense and accurate point clouds, enabling the detection of small obstacles, like rocks laying near the rail. During an offline training phase, the system learns a background model of the level crossing from a set of point clouds. Then, online, obstacles are detected as occupied space contrasting with the background model. To reduce the need for manual on-site calibration, the system automatically estimates the pose of the level crossing and railway with respect to the laser scanner. Experimental results show the ability of the system to successfully perform on a set of 41 point clouds acquired in an operational one-lane level crossing.

## 1. Introduction

Railway level crossings (e.g., see Figure 1) are a safety weak point in the railway infrastructure. To reduce the risk of collision between trains and people, vehicles, or any other goods, these are coordinated by half-barriers and traffic signalling. However, there is always the risk that people do not respect signalling or that an unexpected event results in the unwanted presence of people or goods in the train's passageway. CCTV systems are often used to allow remote operators to identify these situations and react promptly. Despite the advantages of relying on human-based decision making, the high number of trains and level crossings to control posit this strategy error-prone. Thus, it is preferable to have the human acting as a complement to sensor-based automated solutions.

Traditionally, obstacle detection at level crossings is done by means of inductive sensors laid on the pavement detecting metallic structures. Another approach is to check if any object interrupts the transmission of a radio or light signal between an emitter and a receptor, strategically and accurately installed on the level crossing. Despite detecting a large set of obstacles to a train's passage, these solutions

will disregard nonmetallic and small obstacles, such as the ones depicted in Figure 2. A collision with such objects, although not necessarily damaging to the train, may project them off the railway endangering people or goods present nearby. The aforementioned obstacle detection techniques also require time-consuming installation and calibration of the equipment. More importantly, even if the sensor detects the obstacle, it will not be able to provide sufficiently discriminatory data for a proper assessment of the alarm's severity. More details on the use of traditional obstacle detection techniques in level crossings can be found in [1].

In recent years, techniques for higher acuity in obstacle detection have been proposed. However, these present several limitations in the detection of obstacles such as the ones depicted in Figure 2, which include a basket, a laid down jerrican, and a small rock. Of particular interest to this work are the challenges imposed by the rock, which due to its low volume ( $\approx 10 \text{ dm}^3$ ) is largely occluded by the rail, rendering it highly inconspicuous to most detection techniques. The small portion of the rock that stands out from the rail is insufficient for a RADAR system (e.g., [2]) to detect without incurring into unacceptable false alarm rates. A finer alternative to RADAR is the use of static 2D laser scanners (e.g., [3, 4]).



FIGURE 1: The level crossing considered for the field trials.

These provide a set of range radial measurements on a plane. However, to provide long range data without returning hits from the ground, these sensors must be accurately aligned with it. Moreover, the sensor must be very close to the ground in order to detect low obstacles, which are easily occluded by any ground's unevenness. Detailed analysis of the level crossing can be done, in principle, by processing high resolution imagery, provided by surveillance cameras (e.g., [5, 6]). However, the robustness of image-based solutions is considerably challenged by the presence of shadows, sudden light variations, and unknown objects. A volumetric analysis of the level crossing should reduce the sensitivity to these environmental factors, provided that enough detail is available. Binocular vision is a low-cost solution to obtain such a rich volumetric information about the environment and, as a result, has been already studied in the context of obstacle detection in level crossings (e.g., [7–9]). However, in binocular vision, the accuracy and completeness of the volumetric data depend heavily on lighting conditions, materials' textures, and distance to scene elements. The latter limitation is particularly important when the goal is to detect partially occluded small obstacles.

This paper presents a system that uses volumetric sensory data to detect small obstacles in level crossings. For this purpose it relies on the use of dense 3D point clouds produced by a tilting 2D laser scanner. This way the system is not as dependent on lighting conditions and object's appearance as it would be if relying on binocular vision, and it is also not as dependent on ground planarity as are solutions based on static 2D laser scanners. The system is composed of a software layer that processes the 3D point cloud preceded by an acquisition hardware layer. Obstacle detection is carried out exploiting the assumption that a quasiplanar dominant ground plane exists. A background model of the level crossing is learned from a set of training point clouds. Obstacles are the objects that show up in the point clouds captured online that are inexistent in the training dataset. To avoid cumbersome and time-consuming manual on-site calibration, the proposed system includes an automatic calibration process that estimates the pose of both level crossing and railway with respect to the laser scanner.

A commercial product from IHI Corporation uses tilting 2D laser scanners for obstacle detection in railway level crossings [10]. The obstacle detection method operates on a scanwise basis; that is, it does not integrate information

into a volumetric map, which limits its ability to engage on a detailed and robust volumetric analysis. Moreover, the method relies on the existence of a perfectly planar ground, whose pose with respect to the sensor must be known beforehand. This limits the ability of the method in detecting small obstacles in uneven ground. As a result, according to IHI, the product is designed to detect obstacles whose volume is greater than  $1 \text{ m}^3$ . However, a rock as small as the one depicted in Figure 2 occupies considerably less than  $1 \text{ m}^3$ . Moreover, the method presented in [10] does not report the ability to automatically estimate the pose of both level crossing and railway with respect to the sensor. Hence, our work is expected to contribute towards an increased detection resolution and faster installation of current laser-based obstacle detectors for railway level crossings.

This paper is organised as follows. Section 2 describes the proposed system. Then, a set of experimental results are presented in Section 3. Finally, in Section 4, some conclusions are drawn and future work avenues are suggested.

## 2. The Proposed System

*2.1. System Overview.* The system's software layer is composed of three main stages: (1) calibration; (2) training; and (3) online operation. The calibration phase is responsible for determining, from a single reference point cloud, the pose of the level crossing (see Section 2.3) and the pose of the railway (see Section 2.4). The system assumes that the sensor will not be moved until further calibration.

From the estimated pose of the level crossing, a set of three zones is defined longitudinal to the railway (see Figures 1 and 3). The goal is to allow separable system's parameterisation and, as a result, a better handling of the speed-accuracy trade-off. Every point cloud to be processed in both training and online operation stages is first segmented into these three zones (see Section 2.5). Then, to define a region of interest (ROI) for subsequent processing steps, the points in each segment of little value for obstacle detection are rejected (see Section 2.6). These points are those at the ground level and too far from the railway medial axis.

In the training phase, the system learns a volumetric map for each zone from the ROI of a set of acquired training point clouds (see Section 2.7). This map includes all static objects present in the level crossing and vicinities. During learning, the level crossing must be free of obstacles. The system must be retrained if the disposition of static obstacles changes.

Once training is concluded, the system can be used online. In this stage, a point cloud is acquired every time the traffic management system needs to ensure that the level crossing is free of unexpected obstacles. Then, the ROI of the acquired point cloud is compared with the volumetric map learned during the training phase to check for the presence of obstacles (see Section 2.8). Afterwards, a set of simple fast to compute metrics are obtained per obstacle (see Section 2.9). This information aims at providing the traffic management system with context information to assess the alarm severity.

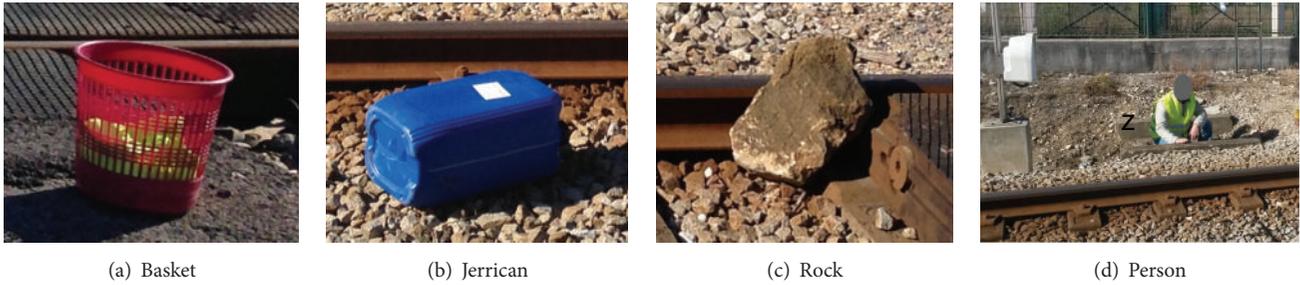


FIGURE 2: Some of the obstacles used to validate the proposed system. The volumes occupied by the basket, jerrican, and rock are approximately  $20 \text{ dm}^3$ ,  $50 \text{ dm}^3$ , and  $10 \text{ dm}^3$ , respectively.

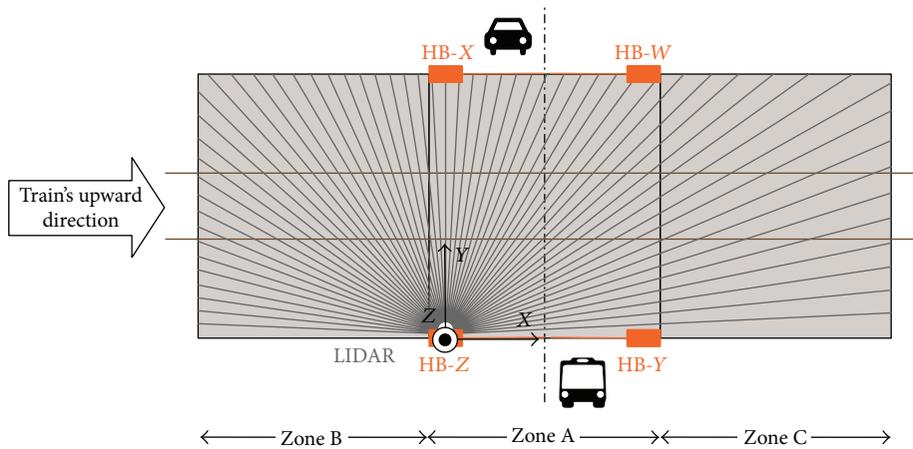


FIGURE 3: The level crossing model. The two horizontal lines represent the railway. The radial lines stemming from the LIDAR position are representative of the laser beams generated when the sensor is not tilted. These lines give an intuition regarding the loss of point density as a function of the distance to the sensor. The frame of reference represented is the one of the LIDAR. The four filled rectangles represent the four half-barriers present in the site. Note that, in most situations, the barriers disposed in a nonsquare quadrilateral shape.

**2.2. Hardware Setup.** The equipment used to capture the point cloud is based on a LIDAR sensor from SICK, LMS III. This laser scanner provides, at 25 Hz, 0.5 m–20 m range measurements with an angular resolution of  $0.25^\circ$  over a field of view of  $270^\circ$ . The LIDAR is coupled with an Inertial Measurement Unit (IMU) 1056-3/3/3, from PhidgetSpatial, and is tilted by means of a MX-64 servo, from Dynamixel (see Figure 4). Assuming that the tilt angle is zero when the laser beams are parallel to the ground plane, the tilting range during a full cycle is defined between  $-15^\circ$  and  $30^\circ$ . The estimation of the laser's pose is performed through an optimal estimator [11] based on the IMU output. The laser's pose information, at every moment, is used to register the set of points captured at any given instant with the previously captured points.

To ensure a proper registration of the several 2D laser scans, the full 3D point cloud acquisition process is set to take on average 5 sec. This acquisition rate can be reduced with a better synchronisation between IMU and servo, ideally in hardware. Nevertheless, actual future deployments of the proposed system would rely on a multibeam laser scanner, such as a Velodyne LIDAR, which is able to provide full 3D point clouds at 15 Hz. Figure 5 depicts a typical 3D point cloud acquired with this setup at the level crossing.

The acquired point cloud encompasses  $\approx 250 \text{ k}$  3D points. Note the decrease in point density as a function of the distance to the LIDAR. Note also that the point density in the half-barriers (HB) changes considerably. All these aspects impose serious challenges to the level crossing's pose estimation problem and obstacle detection thereafter.

The estimator, servo controller, and laser data registration to produce a complete point cloud run as separate nodes on the Robot Operating System (ROS) [12]. The detection system was tested on a 2.66 GHz Intel Core 2 Duo-based MacBook Pro with 8 Gb of RAM, running the operating system OSX 10.9 Maverick. The system was fully implemented in C++ with extensive use of the Point Cloud Library (PCL) [13] for low-level point cloud processing tasks.

**2.3. Level Crossing's Pose Estimation.** To detect obstacles and classify their relevance, it is crucial to estimate from the reference point cloud,  $C_R$ , the pose of the level crossing, that is, its position and orientation in the sensor's frame. The level crossing's pose is estimated from the configuration of the four half-barriers (HB), which are assumed to be upwards in the reference point cloud (see Figures 1 and 5). An HB in the upwards position is expected to exhibit a height with respect

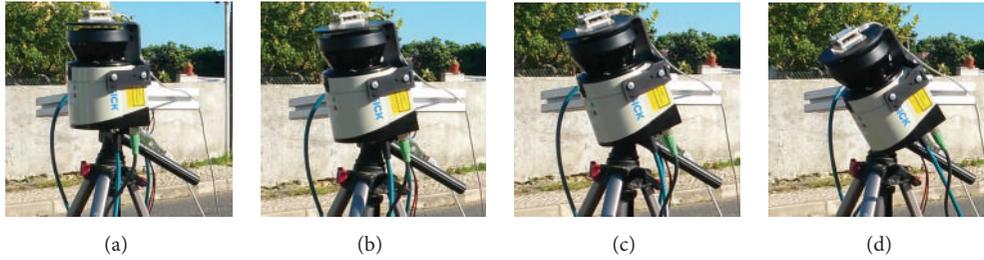


FIGURE 4: The laser scanner being actively tilted by the servo so as to acquire a complete 3D point cloud (time flows from (a) to (d)). The laser scanner tilting angle is estimated at each time step via the IMU mounted on the top of the sensor.

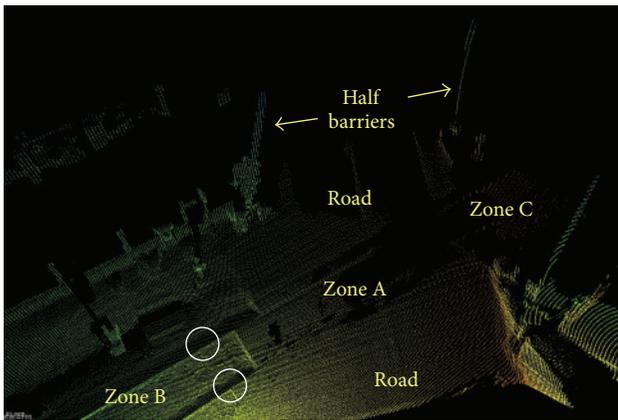


FIGURE 5: A typical 3D point cloud acquired at the level crossing.

to the ground plane between 2 m and 6.5 m, depending on the road's width. Hence, the four HB are expected to stand out considerably from the other elements in the scene.

The first step to detect the four HB is to determine the set of points in the reference point cloud  $C_R$ ,  $G$ , that corresponds to the ground plane and, hence, is surely not belonging to any HB. This set of points are those whose signed orthogonal distance to the ground plane is higher than an empirically defined scalar,  $\eta$ . This value is typically high (e.g., 1.5 m) so as to ensure that the dominant ground plane is found, despite potential ground's unevenness. Formally, a point  $\mathbf{p} = (p_x, p_y, p_z)$  in the reference point cloud,  $\mathbf{p} \in C_R$ , is added to the set of ground points  $G$ ,  $G \leftarrow G \cup \{\mathbf{p}\}$ , if

$$d(\mathbf{p}, \pi) < \eta, \quad \text{with } d(\mathbf{p}, \pi) = \frac{ap_x + bp_y + cp_z + d}{\sqrt{a^2 + b^2 + c^2}}, \quad (1)$$

where the tuple  $\pi = (a, b, c, d)$  contains the set of coefficients of the plane equation  $ax + by + cz + d = 0$  that defines the ground plane.

The installation of the sensor may not ensure its strict verticality and, thus, the ground plane's orientation and distance to the sensor are unknown without calibration. This means that the ground plane's coefficients need to be estimated directly from the reference point cloud  $C_R$ . The level crossing is not strictly planar, the input point cloud is noisy, and there are many objects standing on the ground plane. Therefore, the ground plane estimation process must be robust to the strong presence of outliers. This is attained by

recurring to random sample consensus method (RANSAC) [14]. In short, this method generates  $n$  plane hypotheses, that is, tuples  $\pi_i = (a_i, b_i, c_i, d_i)$ ,  $i = 1 \dots n$ , and test which of them induces a higher number of inliers in the data. An inlier is a 3D point whose distance to the putative plane is below threshold  $\eta$  (see (1)). The best hypothesis, called simply  $\pi$  henceforth, is then refined with all data using a weighted least-squares approach. Refer to [15] for further details on the application of the RANSAC method for ground plane estimation.

The unorganised set of points belonging to objects present in the level crossing is composed of all nonground points,  $O = C_R \setminus G$ . As this set is unorganised, there is no distance function implicit in the order of the points in the set and, thus, there is no spatial connectivity as well. To segregate the various objects present in the scene, a segmentation procedure based on the Euclidean distance between the points is applied [16]. The method clusters points in a way that each point within a cluster is close to at least another point in the cluster and it is far from any point from all other clusters. Closeness is empirically defined in an Euclidean sense by a scalar  $d$ . At this point, all nonground points contained in  $O$  are separated into a set of clusters (segments),  $S$ . This set of clusters excludes clusters that are unreasonably too small ( $<10$  points) or too large ( $>2500$  points). Too small clusters are often caused by the presence of noise whereas too large clusters are typically resulting from faulty laser scans registration.

The set of segments  $S$  contains all objects that are present in the level crossing, which includes walls, debris, traffic control equipment, and HB themselves. A first screening to separate the HB from all other objects is done by exploiting the knowledge that the four HB are oriented upwards and are surely taller than a given threshold  $l$ . In this sense, and assuming that the sensor is roughly vertically aligned, a segment  $s$  is added to the set of putative HB segments,  $P \leftarrow P \cup \{s\}$ , if the height of the cluster is higher than  $l$ ,  $h(s) > l$ , with

$$h(s) = \max_{\forall \mathbf{p} \in s} \{p_z\} - \min_{\forall \mathbf{p} \in s} \{p_z\}. \quad (2)$$

At this point, the set  $P$  is populated with an unordered set of segments corresponding to the four HB present in the scene and of segments potentially corresponding to non-HB objects and noise that could not be rejected as a function of its height. To reduce complexity in the next steps, these segments will be represented by their centroids projected onto

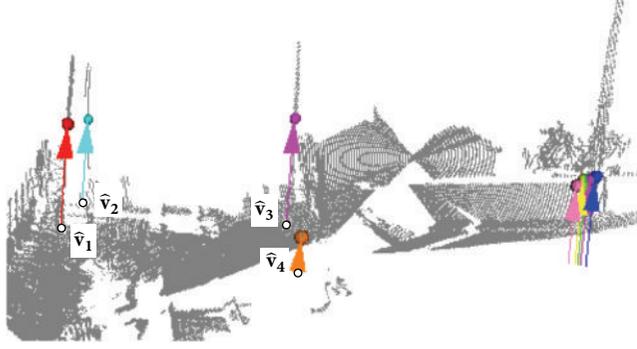


FIGURE 6: The several objects detected when searching for the HB in the same point cloud as the one depicted in Figure 5 (view aligned with the railway). The coloured spheres on top of arrows represent the centroid of each detected object and the unfilled circles at the starting points of the associated arrows correspond to the projection of the centroids on the ground plane. Note that only the objects detected in left hand side of the figure correspond to the actual HB.

the ground plane. The set of these two-dimensional vectors representing the position of a given object on the ground plane is denominated by  $P'$ . Formally, for each segment  $s \in P$  with unprojected centroid  $c(s) = (1/|s|) \sum_{p \in s} \mathbf{p}$ , a two-dimensional vector is added to  $P'$ ,  $P' \leftarrow P' \cup \{\Psi(c(s), \pi)\}$ :

$$\Psi(\mathbf{c}, \pi) = \mathbf{c} - ((\mathbf{c} - \mathbf{o}_\pi) \cdot \mathbf{n}_\pi) \mathbf{n}_\pi, \quad (3)$$

where  $\mathbf{n}_\pi = (a, b, c) / \|(a, b, c)\|$  is a unitary vector normal to the ground plane defined by  $\pi = (a, b, c, d)$  and  $\mathbf{o}_\pi = (0, 0, -d/c)$  is an arbitrary point on plane  $\pi$ .

The next step is to check which objects represented by their two-dimensional positions in  $P'$  actually do correspond to each of the four HB. That is, the previous step potentially detects many more objects than the ones corresponding to the actual HB (see Figure 6). This association problem is solved based on a model of the level crossing. This model assumes that the four HB are disposed as a quadrilateral laid on the ground plane, whose shape and dimensions are defined based on prior knowledge available at the system's commissioning phase. This quadrilateral model is used to check how closely the shape of a given quadrilateral, whose vertices are four of the elements in  $P'$ , matches the shape of a quadrilateral model.

The quadrilateral model is defined in terms of four two-dimensional vertices,  $\mathbf{q} = (\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \mathbf{v}_4)$ , each corresponding to the position of an HB. The similarity between the quadrilateral model  $\mathbf{q}$  and a putative quadrilateral  $\mathbf{q}' = (\mathbf{v}'_1, \mathbf{v}'_2, \mathbf{v}'_3, \mathbf{v}'_4)$  with geometric centre  $\bar{\mathbf{v}}' = (1/4) \sum_{i=1}^4 \mathbf{v}'_i$ ,  $\Phi(\mathbf{q}, \mathbf{q}')$  is taken as the cumulative Euclidean distance between the  $i$ th vertexes in both quadrilaterals. The quadrilateral model is by definition centred in the origin, whereas the putative quadrilateral is not. As a consequence, the similarity metric  $\Phi(\mathbf{q}, \mathbf{q}')$  must be computed after bringing the quadrilateral to the origin by means of translating its vertices by  $\bar{\mathbf{v}}'$ :

$$\Phi(\mathbf{q}, \mathbf{q}') = \sum_{i=1}^4 \left\| (\mathbf{v}'_i - \bar{\mathbf{v}}') - \mathbf{v}_i \right\|. \quad (4)$$

The combination of four object locations stored in  $P'$  that render the quadrilateral that best fits the quadrilateral model is the one, over the set all possible combinations  $Q$ , that builds the quadrilateral minimising the similarity metric:

$$\hat{\mathbf{q}}_A = \arg \min_{\mathbf{q}' \in Q} \Phi(\mathbf{q}, \mathbf{q}'). \quad (5)$$

Figure 7 provides an illustration of the search procedure. The vertices of the quadrilateral  $\hat{\mathbf{q}}_A$  correspond to the estimated locations of the four HB and, as a result, to the pose of the railway level crossing. The points defining this quadrilateral for a specific point cloud are depicted as  $\hat{\mathbf{v}}_1 \hat{\mathbf{v}}_2 \hat{\mathbf{v}}_3 \hat{\mathbf{v}}_4$  in Figure 6.

**2.4. Railway's Pose Estimation.** The quadrilateral defining the level crossing's pose,  $\hat{\mathbf{q}}_A$ , is valuable information to determine where obstacles cannot be present for safety reasons. Further detail on which objects in the site can potentially be hit by the train can be obtained by determining the pose of the railway. A coarse approximation to the railway's pose can be geometrically obtained from the pose of the level crossing. Concretely, the railway is expected to cross the level crossing at its middle axis. To get an accurate estimate, the coarse railway's pose hypothesis must be refined according to local sensory data.

Specifically, the railway's middle axis is coarsely defined as the line  $\mathcal{L}$  that splits the quadrilateral  $\hat{\mathbf{q}}_A$  into two equal halves. Let us define the quadrilateral representing zone A,  $\hat{\mathbf{q}}_A$ , by centroids  $\hat{\mathbf{v}}_1 \hat{\mathbf{v}}_2 \hat{\mathbf{v}}_3 \hat{\mathbf{v}}_4$ . Then,  $\mathcal{L}$  is the line that connects the points  $(1/2)(\hat{\mathbf{v}}_1 + \hat{\mathbf{v}}_3)$  and  $(1/2)(\hat{\mathbf{v}}_2 + \hat{\mathbf{v}}_4)$  (see Figure 8).

In actual level crossings, the quadrilateral defined by the half-barriers does not predict accurately enough the location of the railway. So as to compensate for this fact,  $\mathcal{L}$  is used as a starting point for an accurate search for both rails' signature in the point cloud. Rails are assumed to be distinguishable from their close vicinities in terms of point density. Figure 5 shows that rails exhibit a higher point density than their surroundings (see the areas indicated by the circles in the figure). This is explained by the fact that a given surface is sampled more densely if its normal vector is parallel with respect to the laser beams than if it is perpendicular. The figure also shows that the rail casts a laser-shadow onto the vicinity ground, which lowers considerably the point density.

To estimate the lines that best represent both rails,  $\mathcal{L}_+$  and  $\mathcal{L}_-$ , two points on each must be found based on the rail's point density signature. The search for these points starts from two points sufficiently distant from each other and from the central area of the level crossing (the pavement in the central area renders the railway volumetrically inconspicuous) along lines  $\mathcal{L}$ ,  $\mathbf{a}_\mathcal{L}$ , and  $\mathbf{b}_\mathcal{L}$  (see Figure 8). Then, four search procedures start from these two points to look for the rails' signature (see Figure 9).

The search procedure proceeds as follows. From the two points  $\mathbf{a}_\mathcal{L}$  and  $\mathbf{b}_\mathcal{L}$ , two emanating search direction unitary vectors are created perpendicular to  $\mathcal{L}$ ,  $\mathbf{d}$ , and  $-\mathbf{d}$ , such that  $\mathbf{d} \cdot (\mathbf{a}_\mathcal{L} - \mathbf{b}_\mathcal{L}) = 0$  (see Figure 10). Each of these direction vectors is used to define a parametric semiline of the form

$$\mathbf{p}(t) = \mathbf{s} + t\mathbf{v}, \quad (6)$$

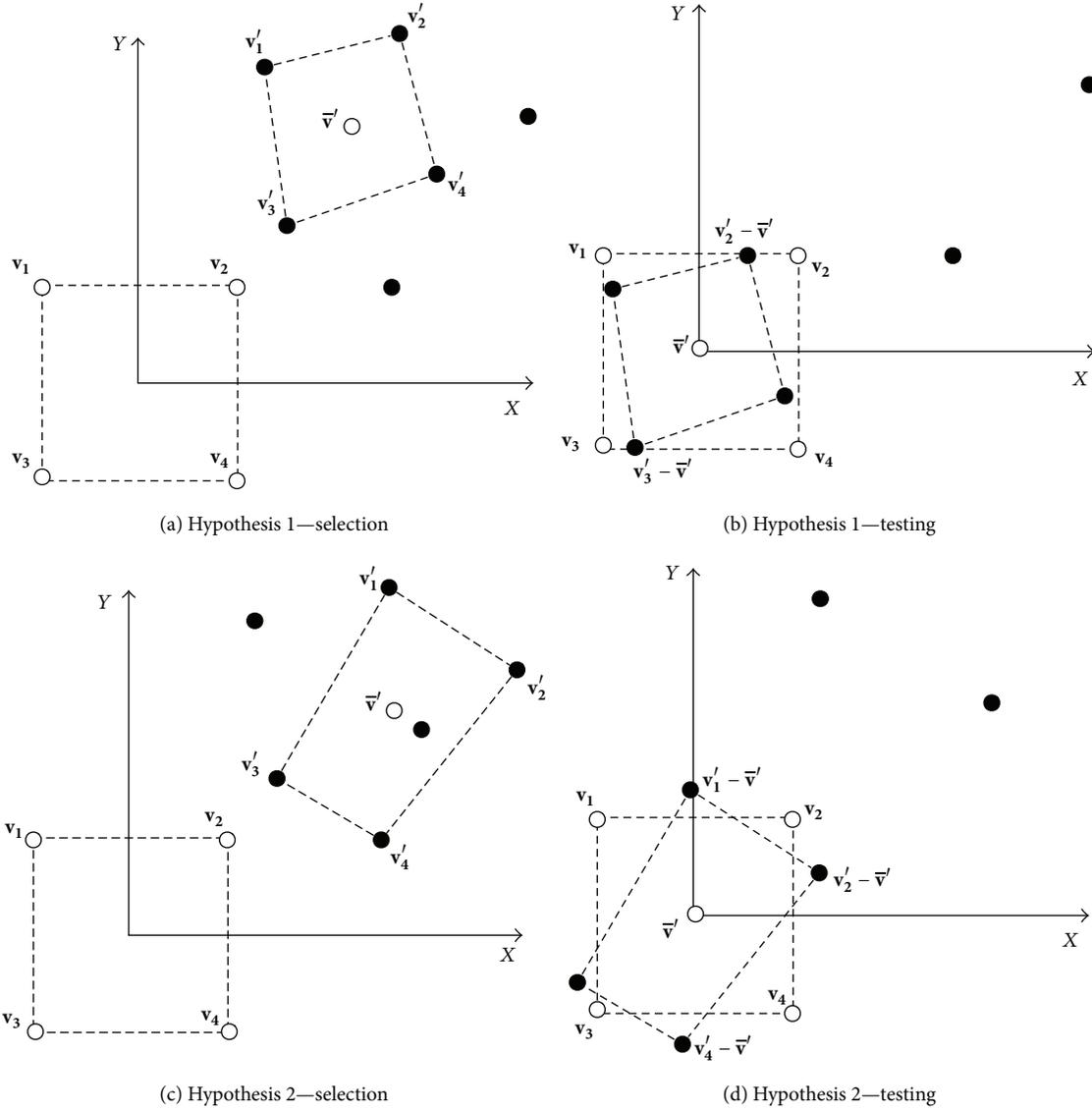


FIGURE 7: An illustration of the process employed to determine which detected objects represent the half-barriers (HB) in the point cloud. The coordinate system represents the 2D projection of the LIDAR frame of reference. The filled circles correspond to detected objects described in terms of the LIDAR frame of reference, that is, putative HB. (a) and (c) Four putative HB were randomly selected to form quadrilateral  $v'_1 v'_2 v'_3 v'_4$ , whose centroid is  $\bar{v}'$ . (b) and (d) The quadrilaterals picked in (a) and (c) are centred in the origin by translating each vertex by the corresponding quadrilateral's centroid vector. After this transformation it is possible to compare each putative HB position, a given vertex  $v'_i$ , with the corresponding expected position  $v_i$  (unfilled circles). After averaging these distances, it is possible to state that the hypothesis picked in (a) is most likely to be correct one when compared to the one picked in (c).

where  $\mathbf{s} \in \{\mathbf{a}_{\mathcal{L}}, \mathbf{b}_{\mathcal{L}}\}$  represents the search's starting point,  $\mathbf{v} \in \{\mathbf{d}, -\mathbf{d}\}$  represents the unitary search direction vector perpendicular to the rail middle axis, and  $t \in \mathbb{R}^+$  is a control parameter that can be used to sample the semiline.

The parametric semilines are sampled by applying  $\delta$  increments to the control parameter  $t$  in (6),  $t \leftarrow t + \delta$ . At each sample point  $\mathbf{p}(t)$ , the set of neighbour points within a radius of  $2\delta$  is computed,  $N(\mathbf{p}(t))$ . If the cardinality of this set shows to be  $\alpha$  times higher than the cardinality of the set obtained in the previous sampling point,  $|N(\mathbf{p}(t))| \geq \alpha|N(\mathbf{p}(t - \delta))|$ , then the search procedure stops and returns the centroid of the neighbour points as the result of the search.

For instance, let us consider the search procedure that starts in  $\mathbf{a}_{\mathcal{L}}$  towards direction  $\mathbf{d}$ . Let us assume that the search stops (the trail is found) when  $t$  in (6) takes the value  $t_a^+$ . In this case, the point on the rail resulting from this search procedure is taken as  $c(N(\mathbf{a}_{\mathcal{L}} + t_a^+ \mathbf{d}))$  (see (6) and Figure 10).

A  $k$ - $d$  tree is built from the input point cloud and used in the computation of  $N(\cdot)$ , that is, in the search for the neighbours of a given point. Formally, a point  $\mathbf{p}'$  is added to the set of neighbours of  $\mathbf{p}$ ,  $N(\mathbf{p}) \leftarrow N(\mathbf{p}) \cup \{\mathbf{p}'\}$ , if  $\|\mathbf{p} - \mathbf{p}'\| \leq r$ . The use of a  $k$ - $d$  tree speeds up the nearest-neighbour search process considerably.

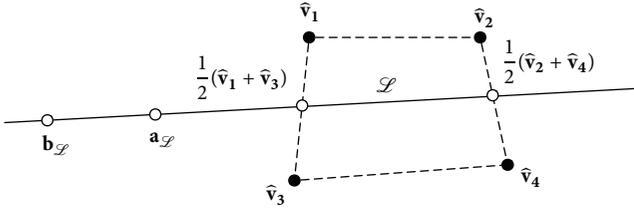


FIGURE 8: Diagram representing the process used to obtain a first rough approximation of the railway middle axis line,  $\mathcal{L}$ , given the quadrilateral  $\hat{v}_1\hat{v}_2\hat{v}_3\hat{v}_4$  formed by the half-barriers.

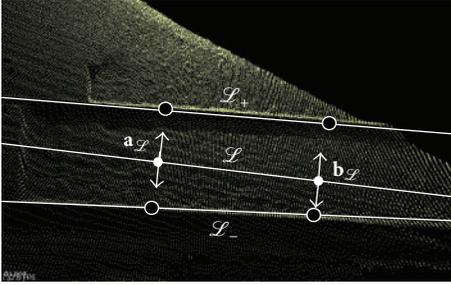


FIGURE 9: An illustrative representation of the rails search from two points laid on the rails' middle axis rough approximation,  $\mathcal{L}$ . The search direction is represented by the white arrows. The unfilled circles represent the four points found on the rails as a result of the search and, thus, that defines the estimated rails' lines  $\mathcal{L}_+$  and  $\mathcal{L}_-$ . These two lines represent more accurately the railway's middle axis than the first approximation  $\mathcal{L}$ .

**2.5. Zones Segmentation.** Point clouds acquired by a tilting laser scanner exhibit a considerable decrease in point density as a function of depth, which means that obstacles nearby the sensor are represented much more densely than obstacles distant from the sensor. To account for this factor, obstacle detection is parameterised differently depending on the distance to the sensor. Concretely, three zones are defined. One of the zones,  $Z_A$ , corresponds to the level crossing core, that is, the area covered by the quadrilateral  $\hat{q}_A$ . The two other zones,  $Z_B$  and  $Z_C$ , are both side-by-side with  $Z_A$  along both railway's directions. The quadrilaterals defining these zones are denominated as  $\hat{q}_B$  and  $\hat{q}_C$ , respectively. These two zones represent the periphery of the level crossing.  $Z_B$  is assumed to be the closest to the sensor and, thus, the zone with higher density of points. Conversely,  $Z_C$  is the zone with the least point density.

For a given input point cloud obtained during either training or online operation phases,  $Z_A$ ,  $Z_B$ , and  $Z_C$  are the sets of 3D points encompassed by the cuboids whose base are the quadrilaterals defining the respective zones. A point  $\mathbf{p} \in \mathcal{R}^3$  is said to belong to a given zone  $i \in \{A, B, C\}$ ,  $\mathbf{p} \in Z_i$ , if its 2D projection onto the ground plane,  $\mathbf{p}^* \in \mathcal{R}^2$ , is encompassed by the quadrilateral  $\hat{q}_i$ . By assuming that the sensor is roughly vertically aligned,  $\mathbf{p}^*$  is  $\mathbf{p}$  without its third coordinate.

Let us assume that one of the quadrilateral sides is defined by the two-dimensional vertices  $(x_1, y_1)$  and  $(x_2, y_2)$  and, consequently, by the implicit function  $f(x, y) = (y_1 - y_2)x + (x_2 - x_1)y + x_1y_2 - x_2y_1$ . A two-dimensional point

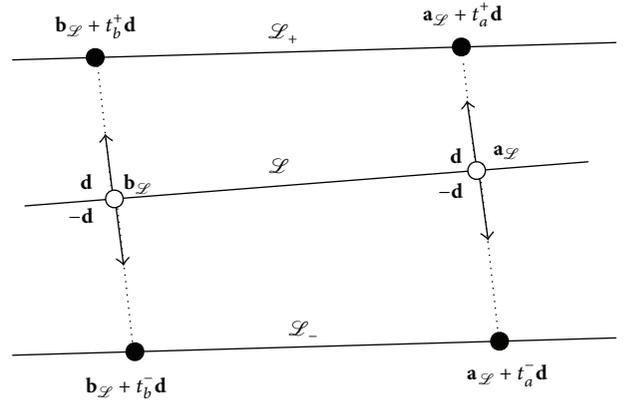


FIGURE 10: A diagram representative of the rail search procedure.

$\mathbf{p}^*$  is said to be below this line if  $f(\mathbf{p}^*) < 0$ . Let us now assume the existence of four implicit functions,  $f_{i \in \{1,2,3,4\}}$ , one representing each side of the quadrilateral. Then, a point  $\mathbf{p}$  is said to be encompassed by the quadrilateral if  $f_i(\mathbf{p}^*) < 0$ ,  $\forall i \in \{1, 2, 3, 4\}$ .

**2.6. Region of Interest.** Obstacles are those objects that can potentially interfere with train's motion. A set of metrics can then set the alarm relevance depending on, for instance, the obstacle's distance to the railway. This means that the region of interest for the detector is not limited to the train's workspace; it must also encompass its vicinities. In this sense, a point in a given zone  $i \in \{A, B, C\}$ ,  $\mathbf{p} \in Z_i$ , is labelled as *relevant* if its orthogonal distance to the railway middle line  $\mathcal{L}$  is below half the train's width plus a safety margin; the aggregate is represented by the scalar  $\omega_i$ :

$$\frac{\|(\mathbf{p} - \mathbf{a}_{\mathcal{L}}) \times (\mathbf{a}_{\mathcal{L}} - \mathbf{b}_{\mathcal{L}})\|}{\|\mathbf{a}_{\mathcal{L}} - \mathbf{b}_{\mathcal{L}}\|} < \omega_i, \quad (7)$$

where  $\mathbf{a}_{\mathcal{L}}$  and  $\mathbf{b}_{\mathcal{L}}$  are two points along the line  $\mathcal{L}$  (see Section 2.4). The index  $i$  in  $\omega_i$  states that the width of the region of interest changes according to the level crossing's zone.

Small obstacle detection is a challenging task due to potential unevenness of the ground, sparsity of the point cloud, and its noisy nature. To reduce sensitivity to noise, the outliers present in all processed point clouds are filtered out. A point is discarded from further processing if the distribution of the distances between itself and its  $k$  nearest neighbours is too dissimilar from the distribution computed over all points in the point cloud [17]. That is, if the point exhibits a local distribution that detaches from all other local distributions then it must be an outlier. Concretely, let  $\bar{d}_{\mathbf{p}}$  represent the average Euclidean distance between a given point  $\mathbf{p}$  and its  $k$  nearest neighbours. Let  $\mu$  and  $\sigma$  be the mean and standard deviation, respectively, of these average distances computed over the whole point cloud  $C_n$ ,  $\mu = (1/|C_n|) \sum_{\mathbf{p} \in C_n} \bar{d}_{\mathbf{p}}$ . With this principle in mind, a point  $\mathbf{p} \in Z_i$  already labelled as *relevant* keeps its label if [17]

$$\mu - \alpha\sigma \leq \bar{d}_{\mathbf{p}} \leq \mu + \alpha\sigma, \quad (8)$$

where  $\alpha$  is an empirically defined scalar. Due to efficiency purposes, the nearest neighbours search is performed on the top of a  $k$ - $d$  tree.

Points already labelled *relevant* are passed through a final filtering process before being analysed for the presence of obstacles. Ground points are by definition nonobstacles and, thus, should be discarded using a RANSAC-like procedure. The ground plane  $\pi$  has been determined for the whole point cloud with a high distance threshold  $\eta$  (see Section 2.3), meaning that it is just a rough approximation of the local ground planarity. Therefore, the ground plane  $\pi$  is not well suited for segmenting ground points from potentially small obstacles. Instead, a ground plane is estimated for each zone based only on the points present therein. Formally, for a given zone  $i \in \{A, B, C\}$ , a RANSAC procedure is run on the top of the subpoint cloud  $Z_i$  with distance threshold  $\eta_i$  to determine the zone's local ground plane  $\pi_i$ . Then, a given point  $\mathbf{p} \in Z_i$  previously labelled as *relevant* only keeps its label if its orthogonal distance to the ground plane  $\pi_i$  is above a given threshold  $\eta_i$ ,  $d(\mathbf{p}, \pi_i) > \eta_i$  (see (11)). The subscript  $i$  in  $\eta_i$  denotes the fact that the threshold can be set separately for each zone. This is useful for instant to reduce the number of false positives in the farthest zone with respect to the sensor. In this zone, the low point density and signal-to-noise ratio induces large false positives if the threshold is low. Conversely, a low threshold operates well in the closest zones. As a result, distant obstacles must be large to be detected. This is a limitation imposed by the sensor that can be easily removed by registering point clouds acquired by complementary laser scanners.

Let us call  $Z_i^R$  the *relevant* point cloud, that is, the subset of  $Z_i$  that only contains points labelled *relevant*. In short, these are points of zone  $i$  that are nonoutliers, are included in the zone's region of interest, and are above the ground. Therefore,  $Z_i^R$  potentially contains a few undetected ground points and objects standing on the ground. Not all these objects should be reported as obstacles. For example, a traffic control equipment is a static object present in the level crossing that is not an obstacle to the train, not even potential. Obstacles are defined as unexpected objects present in the level crossing and, thus, above-plane is not a sufficient condition to report the presence of an obstacle. The next section presents an additional step to identify unexpected objects in the region of interest of each zone.

**2.7. Training.** To detect obstacles, a background model is first learned during a training phase for each zone  $i \in \{A, B, C\}$ ,  $Z_i^T = Z_{i,0} \cup \dots \cup Z_{i,k}$ , where  $k$  is the number of clouds acquired for training. Hence, the background model is simply defined as the union of all point clouds in the training dataset (see Section 2.6). This assumes that only static elements (e.g., ground, traffic control equipment, and walls) are present in the level crossing during training and, therefore, represented in the background model.

The training phase occurs during the installation of the obstacle detector in the level crossing. It may be repeated whenever new static obstacles are added to or removed from the level crossing or when the sensor is moved.

**2.8. Obstacle Detection.** After training, the system enters in the online operation stage, meaning that, prior to each train passage, it obtains a new point cloud, builds a *relevant* point cloud  $Z_i^R$  per zone  $i \in \{A, B, C\}$  (see Section 2.6), and detects the presence of obstacles in it (see below).

Obstacles are defined as objects present in the current point cloud that were not present in the training dataset. Therefore, a potential obstacle is present whenever a difference between point cloud,  $Z_{i \in \{A, B, C\}}^R$ , and the corresponding background model,  $Z_i^T$ , is found. Comparing the point clouds directly is impractical as these are unorganised; that is, there is no obvious implicit spatial order in their vector structures, and their point densities differ considerably.

To reduce computation, a fine regular grid with cell size  $\nu$  is superposed onto each point cloud and all the points encompassed by each grid cell are substituted by their centroids. This results in two new downsampled point clouds per zone  $i \in \{A, B, C\}$ ,  $\mathcal{V}(Z_i^R)$  and  $\mathcal{V}(Z_i^T)$ . Then, the two downsampled point clouds in each zone are compared according to the method proposed by Kammerl et al. [18]. In short, the first point cloud,  $\mathcal{V}(Z_i^R)$ , is used to build the initial structure of a double-buffered octree. Then, the double-buffered octree's structure is altered in order to accommodate the contents of the second point cloud,  $\mathcal{V}(Z_i^T)$ . During this process, new octree's leaf nodes are added to the octree's structure so as to represent elements of the environment that were not represented by the initial octree's structure. The centroids of these octree's new leaf nodes are used to create a point cloud representing the set of potential obstacle points. This process is repeated so as to create a point cloud  $Z_i^*$  per zone  $i \in \{A, B, C\}$ .

Finally, all points in  $Z_i^*$  are segmented into a set of spatially separate objects according to their Euclidean distance (see Section 2.3), that is, potential obstacles,  $O = \{o_0, \dots, o_m\}$ , as in Section 2.3, where  $m$  is the number of separated objects found in the process. This set of objects does not include too small segments as these are likely to be caused by the presence of noise. The size of the smallest acceptable segment depends on the point density and range accuracy and, hence, it must be defined per zone,  $\psi_i$ ,  $i \in \{A, B, C\}$ . As the segments are built upon a downsampled point cloud, the point density of the clusters is roughly homogeneous, which means that the number of points in the segment is a rough approximation of the segment's surface area. As a result, small segments can be safely, that is, in a density-invariant way, rejected if it is supported by less than  $\zeta$  points.

**2.9. Obstacle Characterisation.** To be able to generate meaningful alarms it is essential to characterise the obstacles found in the train's workspace and vicinity. This section presents two simple metrics that can be computed rapidly. General purpose object recognition techniques adequate for point clouds (e.g., [19]) could also be applied to endow the alarming system with additional semantics, at the expense of additional computation.

The first metric is defined as the height and diameter of a bounding cylinder of the obstacle. This provides easy-to-interpret volume information that can be used by

the alarming system to determine the relevance of the obstacle from its approximate volume and outline. The bounding cylinder is assumed to lay on the ground surface assuming that the lower part of the obstacle may be occluded. Moreover, the cylinder is perpendicular to the ground plane and its position is defined as the ground plane projected centroid of the obstacle points:

$$c_c(o, \pi) = (\Psi(c(o), \pi)). \quad (9)$$

The height and diameter of the cylinder with respect to ground plane  $\pi$  for obstacle  $o$ ,  $h_c(o, \pi)$ , and its radius,  $r_c(o, \pi)$ , are

$$h_c(o, \pi) = \max_{\forall \mathbf{p} \in o} \left\{ \frac{ap_x + bp_y + cp_z + d}{\sqrt{a^2 + b^2 + c^2}} \right\}, \quad (10)$$

$$r_c(o, \pi) = \max_{\forall \mathbf{p}, \mathbf{q} \in o} \{\|\mathbf{p} - \mathbf{q}\|\}.$$

The third metric is the distance the cylinder is from the railway's middle axis,  $\mathcal{L}$ . This information is crucial to determine the likelihood that the obstacle may actually be hit by the train. Formally, the third metric for object characterisation is computed as follows:

$$d_c(o, \pi) = \frac{\|(c_c(o, \pi) - \mathbf{a}_{\mathcal{L}}) \times (\mathbf{a}_{\mathcal{L}} - \mathbf{b}_{\mathcal{L}})\|}{\|\mathbf{a}_{\mathcal{L}} - \mathbf{b}_{\mathcal{L}}\|}. \quad (11)$$

### 3. Experimental Results

To validate the proposed system a set of point clouds were acquired in the level crossing of Sabugo, Portugal (see Figure 1), with the setup presented in Section 2.2. To estimate the level crossing (see Section 2.3) and railway (see Section 2.4) poses, the calibration point cloud depicted in Figure 5 was used. A set of similar 9 other point clouds were used for learning the background model (see Section 2.7). Another set of 32 point clouds was used to determine the ability of the detection system to report the presence of the obstacles depicted in Figure 2. In this set, 28 clouds contain obstacles and 4 other clouds are free of obstacles.

The system's parameters were empirically set as follows (see Section 2):  $n = 1000$ ,  $\eta = 1.5$  m,  $\eta_A = 20$  cm,  $\eta_B = 15$  cm,  $\eta_C = 70$  cm,  $\psi_A = 32$ ,  $\psi_B = 35$ ,  $\psi_C = 55$ ,  $\omega_A = 4.25$  m,  $\omega_B = \omega_C = 1.75$  m,  $d = 10$  cm,  $l = 2.5$  m,  $\delta = 3.5$  cm,  $\alpha = 2.5$ , and  $k = 30$ . Finally, the volume of the minimum octrees' leaf size was set to 5 cm  $\times$  5 cm  $\times$  5 cm.

**3.1. Pose Estimation Results.** Figure 11 depicts the pose estimation result obtained with the calibration point cloud. The figure shows that the system managed to detect the half-barriers properly (represented by the upwards directed arrows), as well as the railway (represented by the parallel red lines). The distance between these two lines was measured to be 1755 mm, which deviates only 5% from the expected distance, which is 1668 mm in Portugal and Spain. A little offset between the estimated railway orientation and the actual one is visible in the figure. This owes to the radius used to check point density variations along sampled points. A smaller radius would reduce the deviation but at the cost of potentially failing to detect the density change.

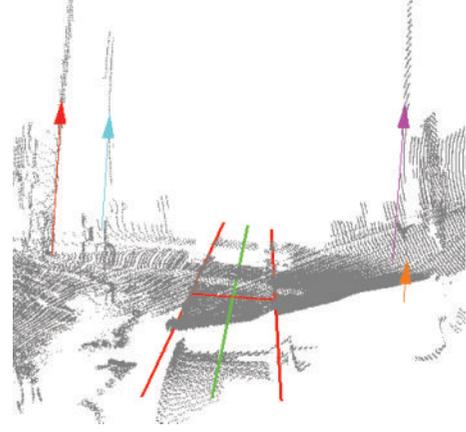


FIGURE 11: Pose estimation results. Level crossing's pose estimation is represented by the arrows overlaid on the estimated half barrier positions. Railway's pose estimation is represented by the lines overlaid on the rails' regions. The red and green lines represent the estimated pose of the rails and their medial axis, respectively.

TABLE 1: System's partial average timings.

Processing step	Section	Phase	Duration
Point cloud acquisition	2.2	Both	5000 ms
Level crossing's pose estimation	2.3	Offline	1930 ms
Railway's pose estimation	2.4	Offline	574 ms
Training	2.7	Offline	210 ms
Region of interest segmentation	2.6	Online	515 ms
Obstacle detection	2.8	Online	100 ms

**3.2. Detection Results.** Figure 12 presents the detection results in four of the 32 tested point clouds. The results show the ability of the system to detect a large variety of obstacles, from large car parts to small rocks laid down near one of the rails. Small-sized obstacles were disposed only on zones A and B as they are by definition inconspicuous in zone C due to its very low point density. A parameterisation that would enable the system to detect small obstacles in zone C would also result in unacceptable false alarm rates. The deployment of a complementary laser scanner would suffice to remove this constraint, meaning that the system would be able to detect both large and small obstacles in the whole area of the level crossing. The system detected all obstacles in the full set of tested point clouds without generating false alarms.

**3.3. Timings.** Table 1 presents the average partial time (over 10 repetitions) spent by the system in each of the processing steps that occur offline (calibration and training), online (detection), or in both offline and online phases. Point cloud acquisition time, which includes the mechanical tilting of the sensor, is rather high. However, as already explained in Section 2.2, it could be easily reduced to a small fraction of the current one with state-of-the-art multibeam laser scanners. Although this is not an issue in the calibration and training phases, it can be problematic while operating online. The time spent in pose estimation is clearly acceptable given the low

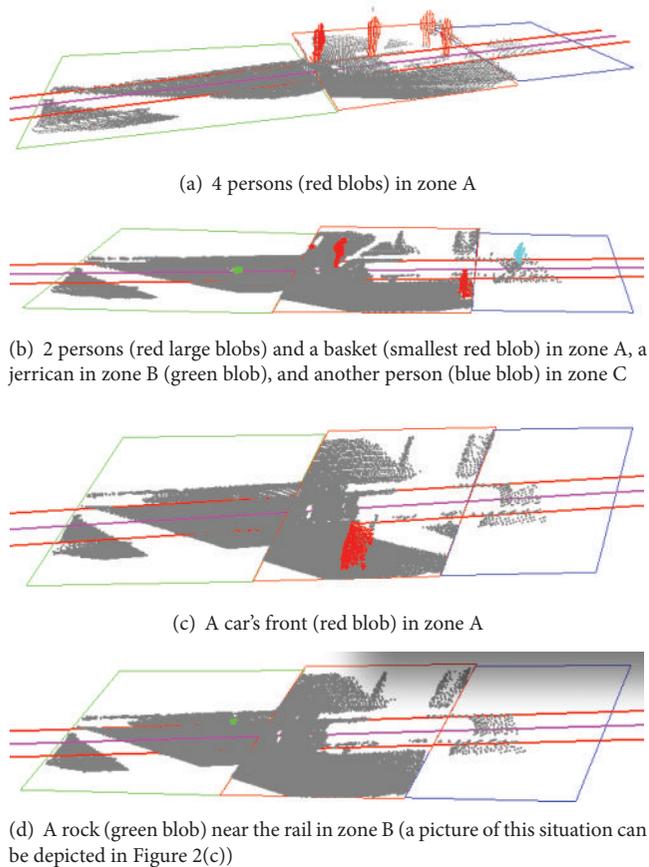


FIGURE 12: Detection results in typical point clouds from the testing data set. The three quadrilaterals correspond to the quadrilaterals that were estimated as defining zones A, B, and C. The grey points correspond to the ground points present in the region of interest. Coloured points are obstacle points (refer to subcaptions for color coding).

time constraints of an offline calibration procedure. Finally, although further code optimisations could be considered, current detection time is acceptable for current safety standards. In sum, the system's responsiveness is in line with the requirements of level crossing safety standards.

#### 4. Conclusions

A system for obstacle detection in level crossings from 3D point clouds acquired with tilting 2D laser scanners was presented. By allowing the detection of large and small obstacles ( $\geq 10 \text{ dm}^3$ ) despite lighting conditions, the system contributes to reducing the number of casualties in level crossings without reducing traffic flow. In addition to solving the detection problem, the proposed system includes an automatic mechanism to estimate the poses of both level crossing and railway, which reduces installation time. A set of experimental results on several point clouds acquired on an operational level crossing shows the viability of the system. As future work we intend to reduce point cloud acquisition time, use more than one laser scanner, and include object classification capabilities to trigger semantic alarms.

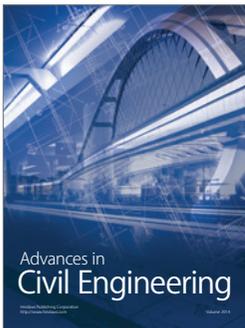
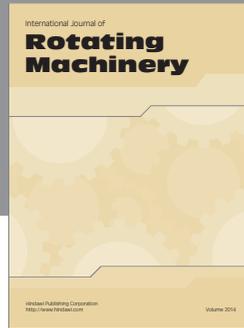
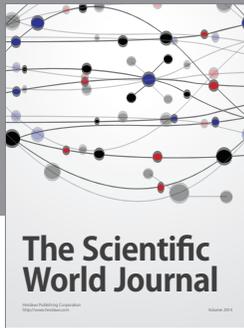
#### Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

#### References

- [1] A. D. Little, "Obstacle detection at level crossing (t522)," Tech. Rep., Rail Safety and Standard Board (RSSB), 2006.
- [2] A. H. Narayanan, P. Brennan, R. Benjamin, N. Mazzino, G. Bochetti, and A. Lancia, "Railway level crossing obstruction detection using MIMO radar," in *Proceedings of the European Radar Conference (EuRAD '11)*, pp. 57–60, IEEE, Manchester, UK, October 2011.
- [3] K. Iwata, K. Nakamura, H. Zhao, R. Shibasaki, and H. Takeuchi, "Object detection with background occlusion modeling by using multiple laser range scanners," in *Proceedings of the 6th International Conference on ASIA GIS*, March 2006.
- [4] H. Takeuchi and T. Shozawa, "Development of advanced obstacle-detecting methods for level crossings," in *Proceedings of the World Congress on Railway Research (WCRR '06)*, Montreal, Canada, June 2006.
- [5] G. L. Foresti, "A real-time system for video surveillance of unattended outdoor environments," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 8, no. 6, pp. 697–704, 1998.
- [6] H. Salmene, L. Khoudour, and Y. Ruichek, "Improving safety of level crossings by detecting hazard situations using video based processing," in *Proceedings of the IEEE International Conference on Intelligent Rail Transportation (ICIRT '13)*, pp. 179–184, IEEE, Beijing, China, September 2013.
- [7] M. Ohta, "Level crossings obstacle detection system using stereo cameras," *Quarterly Report of RTRI*, vol. 46, no. 2, pp. 110–117, 2005.
- [8] I. Yoda, K. Sakaue, and D. Hosotani, "Multi-point stereo camera system for controlling safety at railroad crossings," in *Proceedings of the 4th IEEE International Conference on Computer Vision Systems (ICVS '06)*, p. 51, IEEE, New York, NY, USA, January 2006.
- [9] N. Fakhfakh, L. Khoudour, E. El-Koursi, J. L. Bruyelle, A. Dufaux, and J. Jacot, "3D objects localization using fuzzy approach and hierarchical belief propagation: application at level crossings," *EURASIP Journal on Image and Video Processing*, vol. 2011, Article ID 548604, 4 pages, 2011.
- [10] K. Sekimoto, K. Nagata, and Y. Hisamitsu, "3D laser radar for traffic safety system," in *Industrial Applications of Laser Remote Sensing*, pp. 143–152, Bentham Science, 2012.
- [11] S. O. H. Madgwick, A. J. L. Harrison, and R. Vaidyanathan, "Estimation of IMU and MARG orientation using a gradient descent algorithm," in *Proceedings of the IEEE International Conference on Rehabilitation Robotics (ICORR '11)*, pp. 1–7, IEEE, Zurich, Switzerland, June–July 2011.
- [12] M. Quigley, B. Gerkey, K. Conley et al., "ROS: an open-source robot operating system," in *Proceedings of the Open-Source Software Workshop of the International Conference on Robotics and Automation (ICRA '09)*, 2009.
- [13] R. B. Rusu and S. Cousins, "3D is here: point cloud library (PCL)," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '11)*, pp. 1–4, IEEE, Shanghai, China, May 2011.
- [14] 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.
- [15] P. Santana, M. Guedes, L. Correia, and J. Barata, “Stereo-based all-terrain obstacle detection using visual saliency,” *Journal of Field Robotics*, vol. 28, no. 2, pp. 241–263, 2011.
- [16] R. B. Rusu, *Semantic 3D object maps for everyday manipulation in human living environments [Ph.D. Dissertation]*, Computer Science Department, Technische Universität München, Munich, Germany, 2009.
- [17] R. B. Rusu, Z. C. Marton, N. Blodow, M. Dolha, and M. Beetz, “Towards 3D point cloud based object maps for household environments,” *Robotics and Autonomous Systems*, vol. 56, no. 11, pp. 927–941, 2008.
- [18] J. Kammerl, N. Blodow, R. B. Rusu, S. Gedikli, M. Beetz, and E. Steinbach, “Real-time compression of point cloud streams,” in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '12)*, pp. 778–785, IEEE, Saint Paul, Minn, USA, May 2012.
- [19] J. Knopp, M. Prasad, G. Willems, R. Timofte, L. Van Gool, and J. Knopp, “Hough transform and 3D SURF for robust three dimensional classification,” in *Proceedings of the 11th European Conference on Computer Vision: Part VI (ECCV '10)*, pp. 589–602, Springer, 2010.



**Hindawi**

Submit your manuscripts at  
<http://www.hindawi.com>

