Traffic sign detection in MLS acquired point clouds for geometric and image-based semantic inventory.

Mario Soilán[a,*], Belén Riveiro[b], Joaquín Martínez-Sánchez[a], Pedro Arias[a]

[a]*Department of Natural Resources and Environmental Engineering, School of Mining Engineering, University of Vigo, 36310, Vigo, Spain*

[a]*Department of Materials Engineering, Applied Mechanics and Construction, School of Industrial Engineering, University of Vigo, 36310, Spain*

Abstract

Nowadays, mobile laser scanning has become a valid technology for infrastructure inspection. This technology permits collecting accurate 3D point clouds of urban and road environments and the geometric and semantic analysis of data became an active research topic in the last years. This paper focuses on the detection of vertical traffic signs in 3D point clouds acquired by a LYNX Mobile Mapper system, comprised of laser scanning and RGB cameras. Each traffic sign is automatically detected in the LiDAR point cloud, and its main geometric parameters can be automatically extracted, therefore aiding the inventory process. Furthermore, the 3D position of traffic signs are reprojected on the 2D images, which are spatially and temporally synced with the point cloud. Image analysis allows for recognizing the traffic sign semantics using machine learning approaches. The presented method was tested in road and urban scenarios in Galicia (Spain). The recall results for traffic sign detection are close to 98%, and existing false positives can be easily filtered after point cloud projection. Finally, the lack of a large, publicly available Spanish traffic sign database is pointed out.

Keywords: Mobile Mapping; Laser scanning; Traffic sign inventory; Traffic sign recognition; Point Cloud Segmentation;

1. Introduction

The visibility and quality of traffic signs is one of the most important factors for road safety. In Spain, driver distractions are the leading cause of fatal car accident, (RACE et al., 2013). The development of intelligent vehicles that can warn and inform the driver are crucial to increasing driver safety. Furthermore, the European Directive on Road Infrastructure Safety Management (European Commission, 2013) remarks that Member States should ensure periodic inspections of roads in operation. Currently, road safety inspection is typically done by a qualified inspector who evaluates a number of safety parameters and writes up a report where the elements that need maintenance are remarked. Therefore, decisions are based upon the knowledge of the inspector and a subjective diagnosis. A semi-automatic or fully automatic inspection will reduce its subjectivity and will save public resources while improving the road safety.

The most common source of information for this inventory task are RGB images taken by one or several cameras installed in a vehicle. The literature on traffic sign detection and recognition on images is numerous, and a broad variety of computer vision techniques have been applied to this problem. Illumination changes, occlusions, cluttered scenes or vandalized traffic signs are some of the challenging problems to deal with. Different color spaces have been used, for instance HSI-HSV (Fleyeh, 2006; Gomez-Moreno et al. 2010), YUV (Shadeed et al., 2003) or Gaussian color model (Li et al., 2015) as a visual feature to define a traffic sign region. Shape features have also been studied, such as Hough Transform (Barrile et al., 2012), Local Contour Pattern

* Corresponding author.

E-mail address: msoilan@uvigo.es

(Landesa-Vázquez et al., 2010), or Local Binary Patterns (Liu et al., 2014). Usually, both color and shape features are combined in order to obtain a better description. Contextually, there are two possible objectives in these works: traffic sign detection, this is, computing the regions on an image that contain a traffic sign (Salti et al. 2015; Wu et al. 2013); and traffic sign recognition, where the specific semantics of the detected traffic signs are obtained using complex machine learning techniques, such as Convolutional Networks (Sermanet and Lecun 2011) or Support Vector Machines (Wang et al., 2013a). Some of the cited articles are based on the German Traffic Sign Detection Benchmark (GTSDB) and the German Traffic Sign Recognition Benchmark (Houben et al. 2013; Stallkamp et al. 2012) which are respectively a single-image detection and a multi-class classification challenge. Wang et al. (2013b) obtained the best results for the GTSDB with detection rates of almost 100%. For GTSRB, Cireşan et al. (2012) got a recognition accuracy of 99.5% with a committee of Convolutional Neural Networks, improving the human performance. However, geometric inventory parameters and contextual properties of traffic signs (inclination of the panel, angle and distance with respect to the trajectory of a vehicle, etc.), which are relevant features to take into account during inventories, are not considered.

Nowadays, Mobile Mapping Systems (MMS) equipped with laser scanners are able to collect accurate and reliable 3D point clouds. A survey of an urban or a road environment provides geometric and radiometric information of infrastructure facilities. Research is focused on the automation of detection and classification processes for these elements. For example, Zhou and Vosselman (2012) detect curbstones and therefore road sides in urban areas, Yu et al. (2015) detect street lights using a pairwise 3D shape context that classifies clustered 3D points, and Serna and Marcotegui (2014) detect and classify 20 urban objects (cars, bollards, traffic lights, etc.) using mathematical morphology and Support Vector Machines (SVM), and they present an extensive review of methodologies employed for urban object classification in point clouds. Recently, Yang et al. (2015) classified several objects (including traffic signs) segmenting multi-scale supervoxels and applying an hierarchical and heuristic classification process.

This paper intends to combine two sources of information, namely 3D point clouds and RGB imagery, both installed in a Mobile Mapping System. In our previous work (Riveiro et al., 2015), only 3D data was processed and the imagery collected by the MMS was omitted. Consequently, in this work the previous methodology is improved not only by adding RGB imagery but by slightly modifying the point cloud processing workflow. Therefore, the proposed method has proven to be efficient for the task of detection, classification and extraction of geometric properties of vertical traffic signs. In Section 2, the proposed methodology is described. In Section 3, the MMS features and the study case are presented. Section 4 will summarize the results and their discussion by comparing them with the results in (Riveiro et al., 2015), and Section 5 will outline the conclusions.

2. Methodology

The proposed method aims to identify both geometric and semantic properties of traffic signs in urban and highway environments. For that purpose, two main sources of data are used: point clouds collected with a mobile laser scanner and imagery collected with RGB cameras, both of them integrated in a MMS.

The general workflow is shown in Figure 1. First, the 3D point cloud is segmented using ground removal and intensity filter techniques. The segmented cloud is organized using a DBSCAN-based clustering, and further filtered in order to isolate each traffic sign in the road. Second, the geometric parameters of each detected sign are extracted. Subsequently, the 3D points of each traffic sign are reprojected on 2D images, using the trajectory of the vehicle, the time stamp of both 3D points and images, and the relative orientation between the

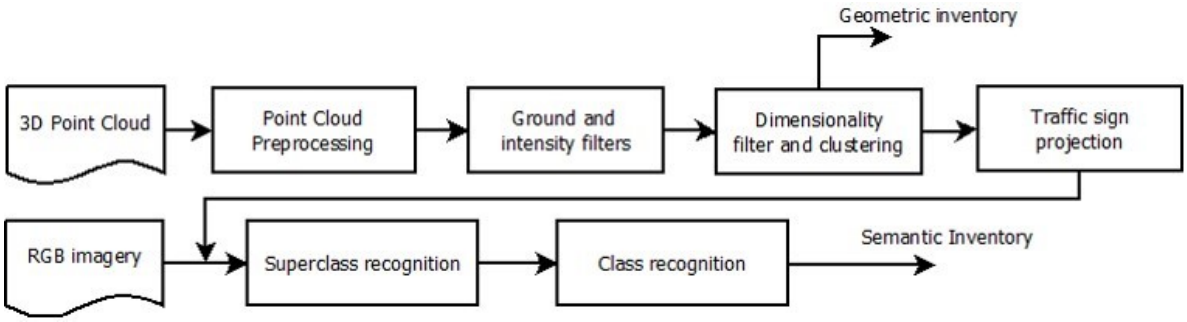

Figure 1. General Workflow

vehicle and the cameras. Finally, a hierarchical classification method is carried out to obtain the meaning of the traffic sign.

2.1. Point cloud segmentation

González-Jorge et al. (2013) state that the intensity attribute of a point cloud can be used for segmentation processes, and therefore for the evaluation of road signs. Furthermore, Pu et al. (2011) classify basic traffic sign shapes using the intensity attribute to support the process. Similarly to these methods, our point cloud segmentation method focuses on using the intensity data recorded by the scanner.

First, the original point cloud is preprocessed. Using the trajectory data, the distance from the 3D points to the sensor is computed, and points further than 20 meters are filtered out. This way, only the information of the road or street where the MMS is traveling will be kept for further processing.

Let P = (**x, y, z, I, ts**) be the point cloud after preprocessing, where **x**,**y**,**z** are the 3D coordinates, **I** is the intensity, and **ts** is the time stamp of each point. A ground removal process is used to separate off-ground points from ground points in P. This step is motivated by two reasons: ground removal significantly reduces the number of points in the cloud, consequently reducing the computational load and saving memory resources; the second advantage is that road markings, which present high intensities in the cloud, are removed too so that they do not distort the filtering based on intensity. Note that the present methodology focuses only on vertical traffic sign detection, and this way the problem of dealing with road marking detections is avoided.

The ground removal process consists of a rasterization of P onto the horizontal plane. This space reduction is motivated by two requirements: improving the computational performance of the algorithm; and, given that mobile laser scanning point clouds are leveled, the projection of 3D points onto the XY plane produce an appropriate plan view of the environment. During rasterization, it is necessary to define a grid size, g_s, that defines the resolution of the raster structure. Grid sizes from 20cm to 1m were tested and trade-off between efficiency and quality of the segmentation was obtained for $g_s = 0.5\ m$ (for smaller sizes, the quality is similar while the execution time grows exponentially, and bigger sizes lose local information). For each point p_i in the cloud, a cell coordinate (x_i^R, y_i^R) and an index id_i are assigned following Equations 1:

$$x_i^R = \frac{x_i - \max(\boldsymbol{x})}{g_s}$$
$$y_i^R = \frac{y_i - \max(\boldsymbol{y})}{g_s}$$
$$id_i = y_i^R + \frac{\text{ceil}(\max(\boldsymbol{y}) - \min(\boldsymbol{y}))}{g_s} x_i^R \quad (1)$$

Where **x** and **y** are the corresponding point coordinates in *P*.

Different features can be computed using the attributes of the points in each cell. For each feature or combination of features, the raster structure can be visualized as an intensity image. Each pixel is related with the points in the cloud with the cell coordinates or with id_i.

For each cell (x^R, y^R) that contains a set of points $p^{XY} = (\boldsymbol{x}^{XY}, \boldsymbol{y}^{XY}, \boldsymbol{z}^{XY}, \boldsymbol{I}^{XY})$ the number of points (Equation 2a), the accumulated height (Equation 2b), the vertical variance (Equation 2c), and a height based image (Equation 2d) are computed.

$$n^{XY} = |p^{XY}| \quad (2a)$$
$$height^{XY} = \sum_{i=1}^{n^{XY}} (z_i^{XY} - min(\boldsymbol{z}^{XY})) \quad (2b)$$
$$\sigma_z^{XY} = \sqrt{\frac{1}{n^{XY}} \sum_{i=1}^{n^{XY}} (z_i^{XY} - mean(\boldsymbol{z}^{XY}))} \quad (2c)$$
$$I_{height}(x^R, y^R) = \frac{\sigma_z^{XY} height^{XY}}{n^{XY}} \quad (2d)$$

The feature I_{height}, which represents the accumulated height weighted with the vertical variance, discriminates between ground and not-ground pixels. It is first normalized to the range [0, 1] and its nonzero values are partitioned in 1000 bins. The height image is binarized using the upper bound of the biggest bin as threshold. Normally, this threshold removes points in cells such that $0 \leq I_{height} \leq 0.001$ with sufficiently good results.

The same raster structure is used for computing the average intensity of the points in each cell (Equation 3), and the image I_{int} is analogously binarized, using the mean of the nonzero elements of I_{int} as threshold.

$$I_{int}(x, y) = \frac{\sum_{i=1}^{n^{XY}} int(p_i^{XY})}{n^{XY}} \quad (3)$$

The result of an AND operation between both binary images is used as a coarse filtering where ground is removed and points within cells with low intensities are filtered.

At this point, the cloud is composed not only of traffic signs but also of light poles, façades, or walls. Given the retro-reflective properties of the traffic sign panels, the intensity attribute is used to classify whether or not a point may belong to a traffic sign. Assuming that the intensity distribution of both non-reflective and retro-reflective points follow a normal distribution (Riveiro et al., 2015), a Gaussian Mixture Model (GMM) with two components is estimated. The first component, with higher area, covers non-reflective points, whereas the second component contains a small number of reflective points. Therefore, each point in the cloud is assigned to the component with the largest posterior probability for its intensity, and those points assigned to the component with the smallest mean are filtered out from the cloud. It should be noticed that two intensity-based filters have been applied: using I_{int} first; and GMM afterwards (aiming to obtain a finer segmentation). The first filter performs a coarse segmentation, reducing the number of input points within the GMM component estimation, which reduces computation time and increases the intensity mean of the filtered point cloud, therefore reducing noise and improving the efficiency of the following clustering process.

Once the remaining points in the cloud have high intensity values, they are clustered using the DBSCAN algorithm (Ester et al., 1996). This algorithm groups together points with a certain density and discards outliers, and it can be easily implemented. DBSCAN uses two parameters, the minimum number of points in the neighbourhood of each point to be part of a cluster, *k*, and the radius of the neighborhood, *eps*. For the results obtained in Section 4, these parameters have been experimentally set to $k = 25$ (the number of points of a traffic sign in the surveyed area is always greater than *k*) and $eps = 0.2m$ (covers the points of a traffic sign but not the surroundings).

The segmentation process finishes with a dimensionality analysis. For each cluster, Principal Component Analysis (PCA) is performed on the covariance matrix of its points, and the eigenvalues $(\lambda_1, \lambda_2, \lambda_3)$ are used to compute planarity as $a_{2D} = \frac{\sqrt{\lambda_2}-\sqrt{\lambda_3}}{\sqrt{\lambda_1}}$. If $a_{2D} < 1/3$, that is, the cluster cannot be labelled as a plane (Gressin et al., 2013) it is filtered out. Non-planar metallic parts or reflective clothing from pedestrians are effectively removed. Finally, those clusters whose height is less than 25cm are removed too, avoiding to deal with license plate detections. Figure 2 shows the process of the point cloud segmentation.

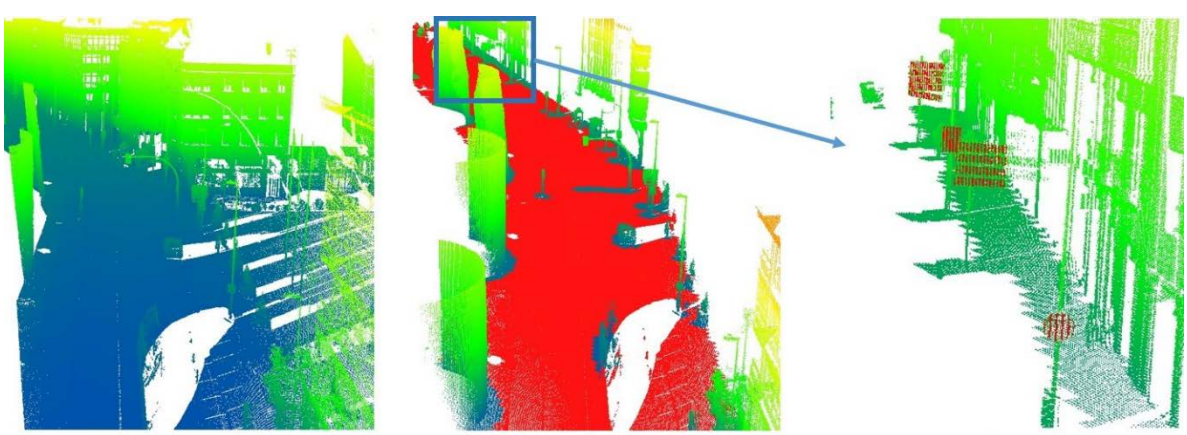

Figure 2. Point cloud segmentation. (a) Raw point cloud of an urban area. (b) Ground removal. Points colored in red belong to the ground. (c) Intensity filtering and point clustering. Points in red belong to traffic sign panels.

2.2. Geometric inventory

Several geometric parameters can be computed for each detected sign. Traffic signs are typically held on a pole, but they can be placed on walls, lampposts or directly over the ground. The first step of the geometric inventory is to determine whether or not the traffic sign is held on a pole. A region growing approach is used for the reconstruction of the surroundings of each traffic sign. For each cluster $C_i = (\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}, \boldsymbol{I}, \boldsymbol{ts}, \boldsymbol{x^R}, \boldsymbol{y^R}, \boldsymbol{id})_i$, that contains a traffic sign panel, the raster indices $\boldsymbol{id}$ are retrieved and an auxiliary cloud Ca_i that contains the points in the original cloud *P* that share the indices in the vector $\boldsymbol{id}$ is created. Ca_i is the region from which the cloud C_i grows iteratively using a neighborhood radius of 0.2m so that a reconstructed cloud, Cr_i is obtained (Figure 3a).

Subsequently, several segments $S_1, S_2, \dots, S_n$ are extracted from Cr_i as shown in Figure 3. Each segment S_j contains points between two heights $h_b - h_o(j)$ and $h_b + h_o(j)$, where h_b has been set experimentally to 75cm over the lowest elevation in Cr_i, and h_o is a vector of heights between 15cm and 50cm with steps of 5cm. Then, the linearity of each segment is computed as $a_{1D} = \frac{\sqrt{\lambda_1}-\sqrt{\lambda_2}}{\sqrt{\lambda_1}}$, with λ_1, λ_2, being the eigenvalues obtained from PCA of the covariance matrix. If the linearity coefficient is smaller than 0.5 for every S_j, the traffic sign is considered *not held on a pole* and the following parameters are computed as shown in Figure 4:

1) *Traffic sign position* (p_s): It is defined as the centroid of each cluster C_i. $p_s(C_i) = (\overline{\boldsymbol{x}}, \overline{\boldsymbol{y}}, \overline{\boldsymbol{z}})$.

2) *Traffic sign height* (h_s): Height difference between $p_s(C_i)$ and the lowest point in Cr_i.

3) *Distance between the trajectory and the traffic sign* (d_t)*:* It is defined as the minimum Euclidean distance between the traffic sign position and the trajectory of the vehicle. $d_t = \min(dist(p_s, traj))$.

4) *Angle between the traffic sign and the trajectory* (α_t)*:* It is defined as the minimum angle between the normal of the traffic sign panel (obtained as the eigenvector corresponding to the smallest eigenvalue after applying PCA on C_i ($\boldsymbol{n}_s$)), and the tangent to the trajectory at the point used for computing d_t ($\boldsymbol{t}_t$). $\alpha_t = \widehat{\boldsymbol{n}_s, \boldsymbol{t}_t}$.

If at least one linearity coefficient is bigger than 0.5 (as in Figure 3b), the traffic sign is considered to be *held on a pole* and the segment that corresponds to the biggest a_{1D} is used for computing the pole vector $\boldsymbol{v}_p$ as the eigenvector corresponding to the biggest eigenvalue after applying PCA on the points of the segment. Finally, the four previous parameters are computed together with:

5) Inclination of the pole in a front view of the sign (α_v)*:* It is defined as the angle between $\boldsymbol{v}_p$ and the vertical $\boldsymbol{v}_z$, both projected in the plane defined by the traffic sign panel.

6) Inclination of the pole in a profile view of the sign (α_e)*:* It is defined as the angle between $\boldsymbol{v}_z$ and the normal of the traffic sign $\boldsymbol{n}_s$.

All these parameters are gathered together, giving spatial and geometric information of each traffic sign that can be used, for instance, as input for a Geographic Information System (GIS) layer.

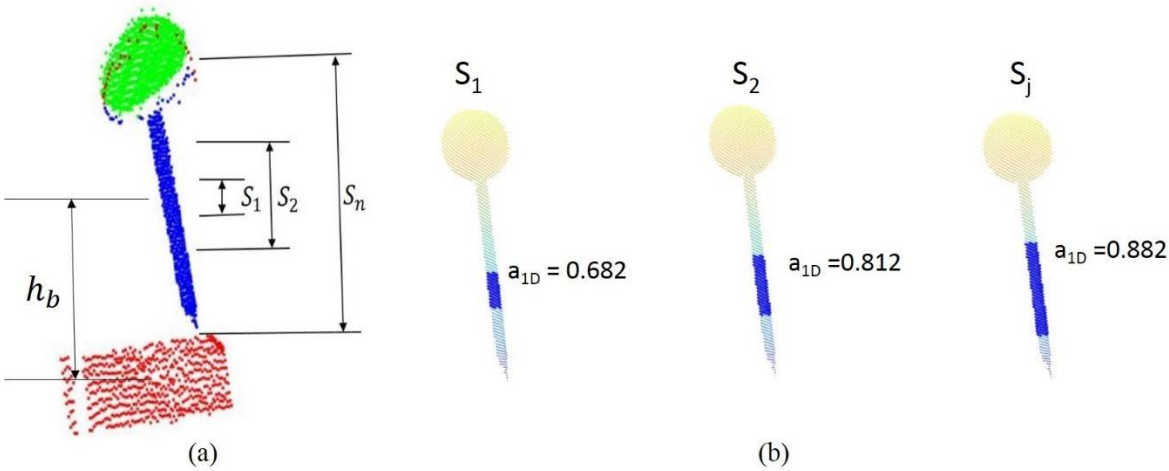

Figure 3. (a) Reconstruction of traffic signs from an auxiliary cloud (Ca_i). Green points (C_i) together with blue points form the reconstructed cloud (Cr_i). Red points are removed after the reconstruction process. Segments $S_1 \dots S_n$ are extracted in order to check if the panel is held on a pole. (b) For each segment S_j (highlighted in blue) the linearity coefficient is evaluated. If at least one linearity coefficient is bigger than 0.5, the segment that corresponds to the biggest a_{1D} is used to compute the pole director vector.

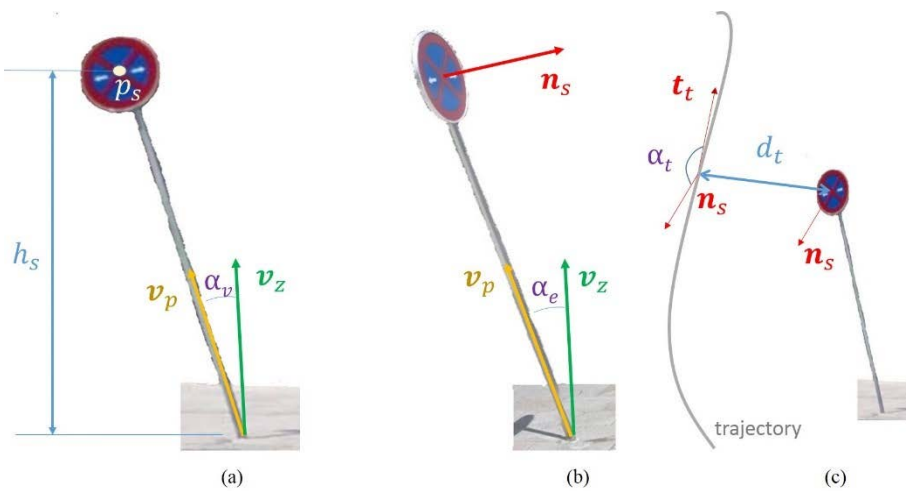

Figure 4. Illustration of the computed geometric parameters. (a) Front view of a traffic sign. (b) Side view of a traffic sign. (c) View of the trajectory together with the traffic sign.

2.3. Traffic sign projection onto 2D images.

The geometric inventory provides essential information for the definition of the traffic sign network. However, the spatial resolution of a point cloud is not enough to recognize the specific meaning of each traffic sign. The best source of information that the MMS provides for the recognition task are four RGB cameras, whose internal calibration and external orientation parameters with respect to the vehicle are known (Puente et al., 2013). Furthermore, the metadata of each photo includes the trajectory – position and orientation – of the

vehicle and a time stamp synced with the point cloud. A time stamp is extracted from each detected sign as the average time stamp of the points in the cluster that defines the sign panel, $t_{sign}(C_i) = \bar{t_s}$. Only the images with a time stamp of $t_{sign}(C_i) \pm 5s$ are analysed for each traffic sign (Figure 5a).

Let A be the global coordinate system, B the GPS coordinate system (a North-East-Down coordinate system), C the vehicle coordinate system and D_j, $(j = 1 \dots 4)$ the coordinate systems of the four cameras. For each vehicle position to be analysed, the points of a traffic sign panel in homogeneous coordinates C_{ih}^A are transformed from the system A to the system C, so both the inner orientation of the camera and the points to be reprojected are in the same coordinate system:

$$C_{ih}^C = (T_{ab}T_{bc})^{-1}C_{ih}^A \tag{4}$$

Where T_{ab} and T_{bc} are the transformation matrix between systems A-B and B-C respectively.

Given the position of the optical centre with respect to the vehicle (x_j, y_j, z_j), together with the unit vector of the principal axis, $\boldsymbol{v_j}$, and the focal distance of each camera f_j, both the plane π_j that contains the camera sensor and the principal point p_j of each camera can be defined:

$$\pi_j \equiv \boldsymbol{v_j} \cdot [(x, y, z) - (x_j, y_j, z_j)] = 0$$
$$p_j = -f_j(x_j, y_j, z_j)^T \boldsymbol{v_j} \tag{5}$$

Each point in the cloud C_{ih}^C is projected to π_j as shown in Figure 5b. The coordinates of the projection (d_u, d_v) are already in the camera frame $(\boldsymbol{u}, \boldsymbol{v})$, and they can be transformed to pixel coordinates (x_{pix}, y_{pix}) (Figure 5c). First, lens distortion is corrected using the distortion model described in Equation 6. Radial distortion parameters k_1 and k_2 are known beforehand, and tangential distortion is considered to be negligible.

$$d'_u = d_u(k_1 r^2 + k_2 r^4)$$
$$d'_v = d_v(k_2 r^2 + k_2 r^4) \tag{6}$$

Where (d_u, d_v) is the undistorted image point, $(d'_u, d_v{}')$ is the distorted image point, and r is the distance between (d_u, d_v) and the principal point p_j.

Finally, the pixel coordinates are retrieved, given the pixel size (s_{pix}) and the image coordinates of the principal point (c_x, c_y) obtained from the calibration, following Equation 7:

$$x_{pix} = \frac{d'_u + c_x}{s_{pix}}$$
$$y_{pix} = \frac{d'_v + c_y}{s_{pix}} \quad (7)$$

Once an entire cluster is projected into an image, the bounding box of the projected points is computed, and the original image is cropped with a margin of 25% of the bounding box size, to deal with calibration errors and add some background to the detection (Figure 5d).

This way, several images of the same sign can be obtained, and the recognition process can be carried out using RGB images with the following advantages with respect to approaches that only use images (Soheilian et al. 2013, De Souza et al. 2013): 1) The detection problem is almost solved, as the location of the traffic signs is known beforehand. 2) Images of the same traffic sign can be stored and analysed together, hence a single classification result is expected for each set of input images (Figure 6). 3) There is a very low probability of finding false positives.

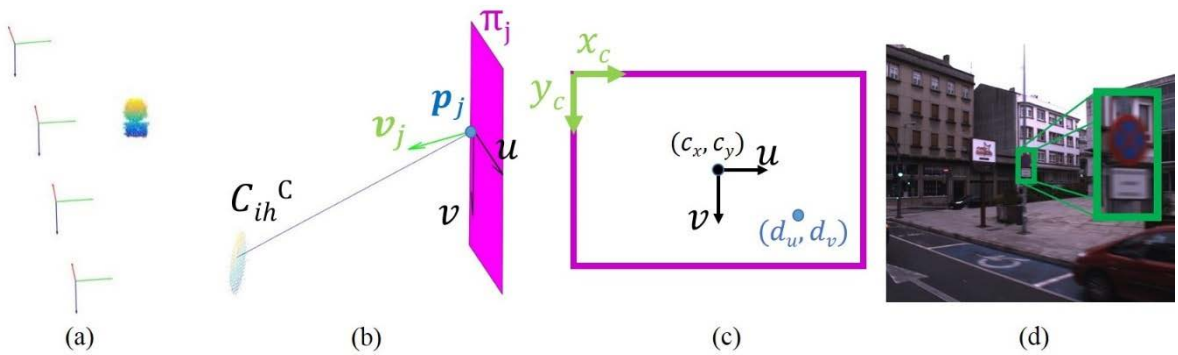

Figure 5. Traffic sign projection onto 2D images. (a) The coordinate system of the vehicle is drawn for four instants whose time stamps are close to the average time stamp of the traffic sign 3D points. (b) Each point of the 3D point cloud C_{ih}^{c} is projected on a plane π_j that contains the camera sensor. (c) The coordinates (d_u, d_v) obtained after the projection are transformed to pixel coordinates. (d) The bounding box of all the computed pixel coordinates is extracted from the input image.

2.4. Traffic sign recognition

The last step of the proposed method consists of defining the meaning of each detected traffic sign given one or several images of the same sign. Although Traffic Sign Recognition (TSR) is not the main objective of this paper, and a vast literature in this regard has been published – see Section 1 –, a general pipeline is proposed.

The TSR workflow is shown in Figure 6. First, seven superclasses are defined in a first level of hierarchy; prohibition, danger, give way, no entry, stop, indication and obligation. Information signs are not considered for this TSR. With this first classification, the color and shape combinations of each superclass is easily distinguished from the others. For each input image, two color bitmaps (red and blue) are computed. The image is transformed from Red-Green-Blue (RGB) colorspace to Hue-Luminance-Saturation (HLS). Each color bitmap is a binary image result of the classification of each pixel in the image. A regularized logistic regression

was trained for this classification using over 200 thousand red and blue pixel values. This way, color can be correctly detected in a variety of illumination conditions. Subsequently, the shape is defined and classified for each color bitmap using Histogram of Oriented Gradients (HOG) (Dalal and Triggs, 2004) and Support Vector

Machines (SVM). Several HOG parameters were tested and the best results were obtained with *Cell Size = 8*, *Block Size = 2*, *bins = 9*, *signed*, a 576-element feature classified by a linear SVM. Over 1500 images were used for training and cross validating the models.

Finally, each traffic sign is classified within its superclass. Some superclasses already define a specific sign – give way, no entry and stop -, while others contain a different number of classes. Each image to be classified is resized to 40x40px and HOG features are computed again with *Cell Size = 5*, *Block Size = 2*, *bins = 8*. In order to maximize the color information extracted by HOG algorithm, the features of each color channel are concatenated. Creusen et al. (2010) state that CIELab and YCbCr color spaces provide the best performance for TSR. The results in section 4 have been obtained using CIELab color space. As in the first hierarchy level, linear SVM are used for training the traffic sign models. They are trained in the usual manner, where the negative set is made up from random background samples and a set of samples of other classes within the current superclass.

The traffic sign class information will be integrated together with the geometric parameters, obtaining a spatial and semantic definition of each traffic sign. This information can be applied to an Intelligent Transportation System (ITS), or used for improving the road network information (parking zones, direction of traffic, speed limit, etc.).

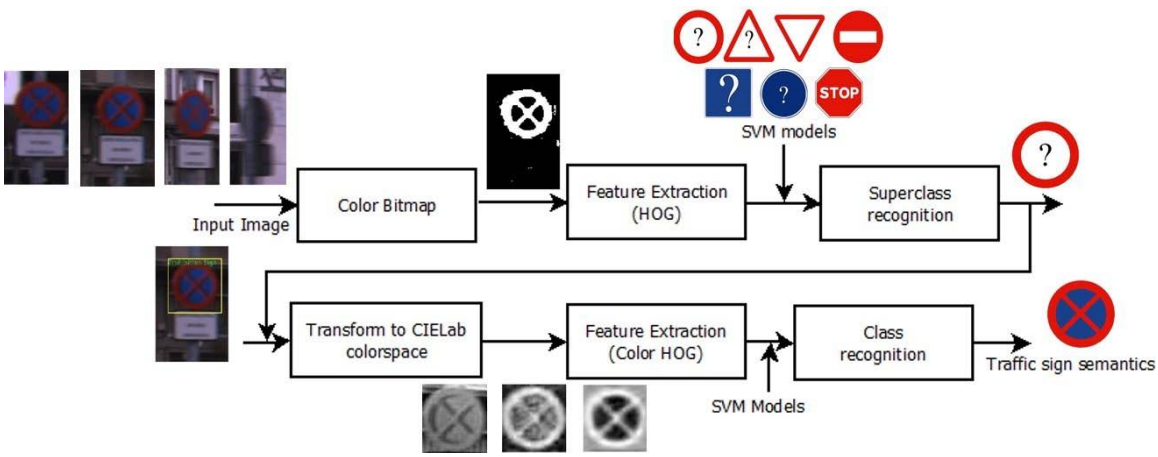

Figure 6. Traffic sign recognition workflow. One or several images of the same sign are obtained for each detected traffic sign on the 3D point cloud. Each image is hierarchically classified in order to obtain the meaning of the traffic sign.

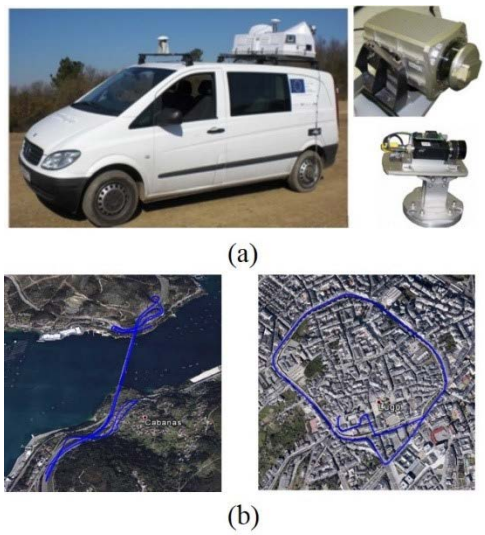

Figure 7. (a) Mobile Mapping System (MMS) equipped with sensors such as laser scanners and JAI cameras. (b) Trajectories for the study case. Rande Bridge and surroundings (road environment, left), and Lugo city centre (urban environment, right).

3. Case study

All the information – point clouds and images - in this work were collected using the LYNX Mobile Mapper by Optech. Inc., (2012), Figure 7a.The system has two sensor heads with a field of view (FOV) of 360º, placed with an angle of 90º between their rotational axes and 45º with respect to the trajectory. The imagery data is registered by four 5-MPix JAI cameras timestamped with GPS time, and integrated with the Lynx Survey software. The navigation system integrates an Inertial Measurement Unit (IMU) with a two-antenna heading measurement system (GAMS). An extensive accuracy study of this system can be found in Puente et al. (2013).

The point clouds used for the validation of the method comprised an urban area and a motorway. Challenging scenarios have been chosen for both cases. The urban area is located in the city of Lugo, in northwest Spain. The area of analysis is a 2.5km three-lane street that encircles a roman wall in the city center (Figure 7b). The average speed of the vehicle is around 20km/h. The road area extends along 7.5km of the AP9 motorway and N-552, N-554 roads, crossing Rande Bridge between the cities of Vigo and Pontevedra, Figure 7b, and it was surveyed at an average speed of 80km/h. The traffic flow is dense in both areas, and the scenes in the collected images are cluttered in the case of the city, and under strong backlights in the case of the motorway.

To avoid excessively large point clouds, the survey was conducted using consecutive strips. They were inspected manually, and the traffic signs were annotated for further validation of the algorithms. Note that bollards, guide posts, or angular milestones are not considered in this study. The information of both study areas is summarized in Table 1.

Table 1. Case study data.

Area	Strips	Points	Traffic Signs
Lugo (Urban)	5	129553905	65
Rande (Road)	3	145759301	78

4. Results and discussion

The described methodology was tested in the road and city scenarios presented in Section 3. The algorithms were implemented using MATLAB software, and the validation was performed in an Intel Core i7-4771 CPU at 3.5GHz. The same parameters were used for urban and road scenarios, although there are differences between both environments. Namely, point clouds of urban environments have higher density of points due to speed limitations, and the probability of false detections is higher as there are more reflective objects – license plates, metallic parts, pedestrian clothing, etc.

4.1. Traffic sign detection

Tables 2 and 3 show the results for traffic sign detection. Each point cloud strip was analyzed and the clusters of points (C_i in Section 2.2) obtained were matched against a ground truth that contains the number and the position of the traffic signs in each strip. The total number of detections, the number of true positives, false positives and false negatives can be computed and are shown in Table 2. Precision, recall, and F1 score performance metrics are shown in Table 3. The global performance averages the results for urban and road environments. A comparison is made between the performance of the proposed method and the results obtained with Riveiro et al. (2015) method.

Table 2. Traffic sign detection results.

Area	Strip	Number of Signs	Number of Detections	True Positives	False Positives	False Negatives
Lugo (Urban)	1	4	6	4	2	0
	2	8	7	7	0	1
	3	16	15	15	0	1
	4	10	15	10	5	0
	5	27	27	26	3	1
	Total	65	70	62	10	3
Rande (Road)	1	23	26	23	3	0
	2	46	49	46	3	0
	3	9	9	9	0	0
	Total	78	84	78	6	0

Table 3. Precision, Recall and F1 score for traffic sign detection.

Area	Precision (%)	Recall (%)	F1 score (%)
Lugo (Urban)	86.1%	95.4%	90.5%
Rande (Road)	92.8%	100%	96.3%
Global performance	89.7%	97.9%	93.4%
Global performace (Riveiro et al., 2015)	91.3%	90.9%	91.1%

A global precision value of 89.7% and a global recall value of 97.9% are obtained for the validation of the algorithm. Although the proposed algorithm applies several filters, which aim to remove false positives, there are still falsely detected objects. A visual analysis of these false positives shows that planar metallic surfaces and some pedestrian reflective clothing are the main sources of error. The results for the urban environment are slightly worse than for the road environment, due to the larger number of reflective objects in the city streets. However, the global recall value is very promising, as only three traffic signs are lost for the whole study case. The main cause of false negatives lies in the filtering stage. Traffic signs placed directly over the ground, or whose reflectivity properties are partially lost due to material deterioration may be erroneously filtered out by the algorithm. This latter cause of false negatives can be meaningful in a road inspection context, as the absence of a traffic sign may indicate maintenance needs. As for the false positives, the recognition stage should be robust enough to discard the images that correspond to erroneous detections in the point cloud. Examples of false negatives and false positives can be seen in Figure 8. Furthermore, the results show the improvement of this method with respect to the previous work, in terms of global performance.

4.2. Traffic sign recognition

The projection of the detected traffic signs from the 3D point cloud on the RGB images precedes the proper recognition stage. As introduced in Section 2.3, different projections of the same sign can be obtained from different images as the MMS drives through. The four cameras are programmed to take one picture every two seconds. Therefore, the number of available images for the analysis of a single traffic sign will be modified by two parameters: the speed of the vehicle, and the relative orientation between the camera and the traffic sign front panel. The output of the projection stage will contain images not only with the front panel but also with the reverse of the traffic signs.

For the performance analysis, only the traffic signs that belong to one of the seven superclasses defined in Figure 6b are considered as positives. Direction and information signs, images resulting from false positives in the previous stage, and images of the reverse side of the signs will be considered as negatives.

The result of the superclass classification is summarized in Table 4. Two performance measurements are computed. First, the number of traffic signs correctly classified (TP_{ts}) is defined as the number of image sets that have at least one image with a true positive, and no false positives. FP_{ts} is defined as the number of image sets with at least one false positive. They represent the overall performance, as the final output is the class of the sign, independently of the number of correct detections. Finally, TP_{im} is simply defined as the number of images correctly classified, and FP_{im} is the total number of false positives.

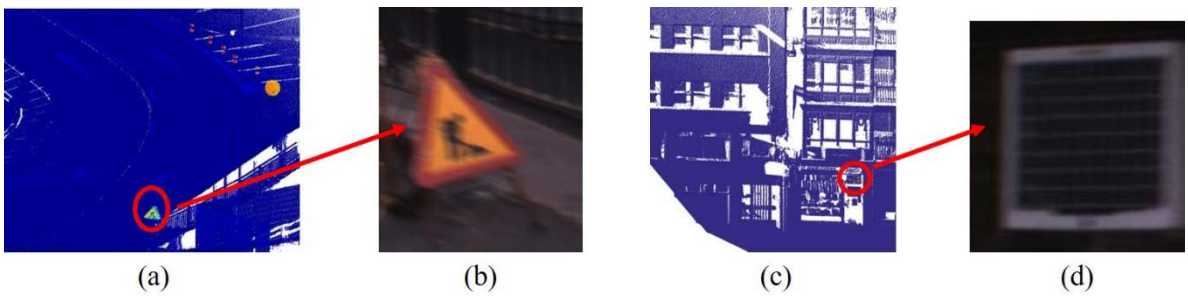

Figure 8. Example of false negatives and false positives. (a)-(b). False negative: A traffic sign that is placed directly over the ground is not detected by the algorithm. (c)-(d). False positive: A metallic part is falsely detected as a traffic sign.

Table 4. Traffic sign superclass recognition results.

	Lugo (Urban)	Rande (Road)
Total number of images	301	544
Number of traffic signs	43	60
TP_{ts}	39	47
FP_{ts}	2	4
Precision	95.1%	93.8%
Recall	90.7%	78.3%
Number of positive images	86	191
TP_{im}	70	157
FP_{im}	2	9
Precision	97.2%	94.57%
Recall	81.4%	82.19%

Two main conclusions can be inferred from these results. First, it is possible to deal with the false positives obtained in the detection stage, as the corresponding images will not contain any traffic sign. This fact makes the whole process very robust against false positives. Second, the proposed method for superclass classification should be modified to improve the recall value, replicating algorithms with better performances in the literature (Wang et al., 2013a) or improving the feature selection and training method for our method.

The last step involves the classification of each traffic sign within its superclass. Only the two more populated class-specific SVM models were trained, for pedestrian crossing and no parking signs. Further training could not be performed because the number of samples for a larger number of classes was not enough to train reliable models (more discussion about this fact in Section 5). Results in Table 5 are, therefore, merely illustrative as the TSR was not the main objective of this work.

Table 5. Traffic sign class recognition results for two classes: Pedestrian crossing and no parking.

		
Samples	9	38
TP	9	27
FP	1	1

5. Conclusions and future work

In this paper a method for an automated detection, inventory and classification of vertical traffic signs from point clouds and images collected by a Mobile Mapping System was presented. The reflective properties of the traffic signs are essential as the method relies on the intensity attribute of the point cloud. A rasterization process is used for the segmentation of the ground and filtering the cloud in a coarse manner. A Gaussian Mixture Model is computed together with DBSCAN and a dimensionality analysis to further segment and cluster the points belonging to the same traffic sign. After this processing, different geometric parameters are computed for each detected traffic sign, namely centroid, height, distance to the trajectory and inclination angles with respect to the vertical.

This represents a novel, robust algorithm for traffic sign detection in point clouds, obtaining a global recall score of almost 98%. The semantics of each sign can be also retrieved. With that purpose, the points that belong to traffic signs are reprojected on RGB images taken by the four cameras of the MMS, and they are hierarchically classified into specific traffic sign classes. However, the number of instances of each class (i.e. the number of images of the same traffic sign) was not enough to train reliable models, and the results obtained were discouraging. While our database is composed of approximately 1500 images with around a 50-100 classes of signs, the best performance results in TSR were obtained with much more populated datasets. The already mentioned German Traffic Sign Recognition Benchmark has more than 50,000 images of 43 classes, the BeligumTS dataset has over 15,000 images, the Summer Swedish Traffic Sign dataset contains over 20,000 images, and Hazelhoff et al. (2012) use over 55,000 images on their classification methodology from panoramic images. However, there are small differences between traffic signs in each country, and there is not a sufficiently large database of Spanish traffic signs images publicly available. The development or the publication of a traffic sign benchmark similar to other countries could represent an improvement on our TSR approach.

The novelty of this method can be summarized in three thematic blocks. Firstly, the point cloud processing workflow has been improved with respect to previous work (Riveiro et al., 2015), adding trajectory-dependent preprocessing steps, a height-based filter which segments the ground (as future work, the ground segment may be processed in order to detect road markings), and a different application of the GMM for fine segmentation of the point cloud. Secondly, the computation of the aforementioned geometric parameters, which may enrich a semiautomatic inventory process, and finally the combination of LiDAR data and RGB imagery in order to boost the potential of a MMS for semantic endowing tasks.

Future studies should focus on adapting our method depending on the environment, as the results on the urban scenario of the case study were slightly worse than on the road environment. Moreover, future research should add more complex semantic descriptions of road networks, using geometric and contextual features and therefore not depending on the reflectivity in order to detect objects such as traffic sign poles. Furthermore, the

performance of our TSR approach was clearly insufficient; besides the lack of a consistent database, any of the state-of-the-art methods should be replicated to get a fully robust algorithm. Finally, the research should move forward with integration in road network spatial information databases.

Acknowledgements

This work has been partially supported by the Spanish Ministry of Economy and Competitiveness through the project HERMES:S3D - Healthy and Efficient Routes in Massive Open-Data based Smart Cities (Ref.: TIN201346801-C4-4-R), Xunta de Galicia (Grant No.CN2012/269), Dirección General de Tráfico (Ministerio del Interior – Gobierno de España) (SPIP20151500) and Human Resources program FPI (Grant BES-2014-067736).

References

Barrile, V., Meduri, G.M., Cuzzocrea, D., 2012. Automatic Recognition of Road Signs by Hough Transform: Road-GIS. J. Earth Sci. ….

Cireşan, D., Meier, U., Masci, J., Schmidhuber, J., 2012. Multi-column deep neural network for traffic sign classification. Neural Networks 32, 333–338. doi:10.1016/j.neunet.2012.02.023

Creusen, I.M., Wijnhoven, R.G.J., Herbschleb, E., De With, P.H.N., 2010. Color exploitation in HOG-based traffic sign detection. Proc. - Int. Conf. Image Process. ICIP 2669–2672. doi:10.1109/ICIP.2010.5651637

Dalal, N., Triggs, W., 2004. Histograms of Oriented Gradients for Human Detection. doi:10.1109/CVPR.2005.177

De Souza, A.F., Fontana, C., Mutz, F., De Oliveira, T.A., Berger, M., Forechi, A., De Oliveira Neto, J., De Aguiar, E., Badue, C., 2013. Traffic sign detection with VG-RAM weightless neural networks. Proc. Int. Jt. Conf. Neural Networks. doi:10.1109/IJCNN.2013.6706809

Ester, M., Kriegel, H.-P., Sander, J., Xu, X., 1996. A density-based algorithm for discovering clusters. Proc. 2nd Int. Conf. Knowl. Discov. Data Min. 226 – 231.

European Commission, 2013. EU Transport in figures. Statistical pocketbook, European Commission. doi:10.2832/19314

Fleyeh, H., 2006. Shadow and highlight invariant colour segmentation algorithm for traffic signs. 2006 IEEE Conf. Cybern. Intell. Syst. doi:10.1109/ICCIS.2006.252225

Gomez-Moreno, H., Maldonado-Bascon, S., Gil-Jimenez, P., Lafuente-Arroyo, S., 2010. Goal evaluation of segmentation algorithms for traffic sign recognition. IEEE Trans. Intell. Transp. Syst. 11, 917–930. doi:10.1109/TITS.2010.2054084

González-Jorge, H., Riveiro, B., Armesto, J., Arias, P., 2013. Evaluation of road signs using radiometric and geometric data from terrestrial LiDAR. Opt. Appl. 43, 421–433. doi:10.5277/oa130302

Gressin, A., Mallet, C., Demantké, J., David, N., 2013. Towards 3D lidar point cloud registration improvement using optimal neighborhood knowledge. ISPRS J. Photogramm. Remote Sens. 79, 240–251. doi:10.1016/j.isprsjprs.2013.02.019

Hazelhoff, L., Creusen, I., De With, P.H.N., 2012. Robust detection, classification and positioning of traffic signs from street-level panoramic images for inventory purposes. Proc. IEEE Work. Appl. Comput. Vis. 313–320. doi:10.1109/WACV.2012.6163006

Houben, S., Stallkamp, J., Salmen, J., Schlipsing, M., Igel, C., 2013. Detection of traffic signs in real-world images: The German traffic sign detection benchmark. Proc. Int. Jt. Conf. Neural Networks. doi:10.1109/IJCNN.2013.6706807

Landesa-Vázquez, I., Parada-Loira, F., Alba-Castro, J.L., 2010. Fast Real-Time Multiclass Traffic Sign Detection based on Novel Shape and Texture Descriptors. IEEE Annu. Conf. Intell. Transp. Syst. 1388 – 1395.

Li, H., Sun, F., Liu, L., Wang, L., 2015. A novel traffic sign detection method via color segmentation and robust shape matching. Neurocomputing 169, 77–88. doi:10.1016/j.neucom.2014.12.111

Liu, C., Chang, F., Chen, Z., 2014. Rapid Multiclass Traffic Sign Detection in High-Resolution Images. IEEE Trans. Intell. Transp. Syst. 2394 – 2403.

Optech Inc., (2012). *Homepage of the Company Optech Inc.* [Online]. Available: http://www.optech.ca, accesed on Dec. 11, 2015.

Pu, S., Rutzinger, M., Vosselman, G., Oude Elberink, S., 2011. Recognizing basic structures from mobile laser scanning data for road inventory studies. ISPRS J. Photogramm. Remote Sens. 66, S28–S39. doi:10.1016/j.isprsjprs.2011.08.006

Puente, I., González-Jorge, H., Riveiro, B., Arias, P., 2013. Accuracy verification of the Lynx Mobile Mapper system. Opt. Laser Technol. 45, 578–586. doi:10.1016/j.optlastec.2012.05.029

RACE, Castrol, BP, 2013. Primer barómetro de las distracciones en la conducción.

Riveiro, B., Diaz-Vilarino, L., Conde-Carnero, B., Soilan, M., Arias, P., 2015. Automatic Segmentation and Shape-Based Classification of Retro-Reflective Traffic Signs from Mobile LiDAR Data. IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. pp. doi:10.1109/JSTARS.2015.2461680

Salti, S., Petrelli, A., Tombari, F., Fioraio, N., Di Stefano, L., 2015. Traffic sign detection via interest region extraction. Pattern Recognit. 48, 1039–1049. doi:10.1016/j.patcog.2014.05.017

Sermanet, P., Lecun, Y., 2011. Traffic sign recognition with multi-scale convolutional networks. Proc. Int. Jt. Conf. Neural Networks 2809–2813. doi:10.1109/IJCNN.2011.6033589

Serna, A., Marcotegui, B., 2014. Detection, segmentation and classification of 3D urban objects using mathematical morphology and supervised learning. ISPRS J. Photogramm. Remote Sens. 93, 243–255. doi:10.1016/j.isprsjprs.2014.03.015

Shadeed, W.G., Abu-Al-Nadi, D.I., Mismar, M.J., 2003. Road traffic sign detection in color images. 10th IEEE Int. Conf. Electron. Circuits Syst. 2003. ICECS 2003. Proc. 2003 2, 890–893. doi:10.1109/ICECS.2003.1301930

Soheilian, B., Paparoditis, N., Vallet, B., 2013. Detection and 3D reconstruction of traffic signs from multiple view color images. ISPRS J. Photogramm. Remote Sens. 77, 1–20. doi:10.1016/j.isprsjprs.2012.11.009

Stallkamp, J., Schlipsing, M., Salmen, J., Igel, C., 2012. Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition. Neural Networks 32, 323–332. doi:10.1016/j.neunet.2012.02.016

Wang, G., Ren, G., Wu, Z., Zhao, Y., Jiang, L., 2013a. A hierarchical method for traffic sign classification with support vector machines. Int. Jt. Conf. Neural Networks 1 – 6.

Wang, G., Ren, G., Wu, Z., Zhao, Y., Jiang, L., 2013b. A robust, coarse-to-fine traffic sign detection method. 2013 Int. Jt. Conf. Neural Networks 1–5. doi:10.1109/IJCNN.2013.6706812

Wu, Y., Liu, Y., Li, J., Liu, H., Hu, X., 2013. Traffic Sign Detection based on Convolutional Neural Networks. Int. Jt. Conf. Neural Networks 1–7. doi:10.1109/IJCNN.2013.6706811

Yang, B., Dong, Z., Zhao, G., Dai, W., 2015. Hierarchical extraction of urban objects from mobile laser scanning data. ISPRS J. Photogramm. Remote Sens. 99, 45–57. doi:10.1016/j.isprsjprs.2014.10.005

Yu, Y., Li, J., Guan, H., Wang, C., Yu, J., 2015. Semiautomated Extraction of Street Light Poles From Mobile LiDAR Point-Clouds. IEEE Trans. Geosci. Remote Sens. 53, 1374–1386. doi:10.1109/TGRS.2014.2338915

Zhou, L., Vosselman, G., 2012. Mapping curbstones in airborne and mobile laser scanning data. Int. J. Appl. Earth Obs. Geoinf. 18, 293–304. doi:10.1016/j.jag.2012.01.024