Dynamic reconstruction in simultaneous localization and mapping based on the segmentation of high variability point zones

Brayan Andru Montenegro, Juan Fernando Flórez and Elena Muñoz

Faculty of Electronic Engineering and Telecommunications, University of Cauca, Popayán, Colombia

ABSTRACT
Dynamic scene reconstruction in real environments is still an ongoing research challenge; moving objects affect the performance of static environment-based simultaneous localization and mapping and impede a correct scene reconstruction. This paper proposes a method for dynamic scene reconstruction using sensor fusion for dynamic simultaneous localization and mapping. It employs two-dimensional LIDAR statistical behaviour to detect and segment high variability point cloud areas containing a dynamic object. The method is computationally low-cost, allowing a 6.6 Hz execution rate. It obtains point cloud reconstruction of a static scene by reducing, segmenting, and concatenating successive point clouds of a dynamic environment. The tests were in real indoor environments with a robotic vehicle and a person traversing a scene. The correlation between the static environment point cloud and successive reconstructed point clouds demonstrates that the proposed method reconstructs different environments in the presence of dynamic objects.

1. Introduction
Simultaneous localization and mapping (SLAM) seek to reconstruct online environments by 3D point cloud generation (PCL) without prior knowledge of the scene (Sahin et al., 2020). A SLAM system should be able to detect and track objects immersed in an environment, whether static or dynamic (Jinyu et al., 2019). Researchers have proposed processing algorithms using highly complex mathematical and statistical methods (Mazurek et al., 2020) (Huynh et al., 2020). Implementing these multiple algorithms should be computationally low-cost to ensure real-time processing as far as possible.

Computational cost is a determining factor for reconstructing real-time environments (Woźniak et al., 2015), primarily when SLAM systems use complete and complex PCLs. For computational cost reduction, several authors have proposed PCL segmentation methods. For example, (Siddiqua and Fan, 2019) applied a supervisor for feature extraction for segmentation, (Ye et al., 2021) implemented monocular depth extraction, and (Liu and Wang, 2021) processed an occupancy matrix by segmenting it into constant surfaces. However, each segmented element present in the scene involves resource consumption. (Delanoy et al., 2019) demonstrated how a reconstruction process by static object prediction requires between 0.15 and 0.35 s, even after applying a voxel grid reduction. (Cao et al., 2019) proposed a Fast and robust feature tracking for the 3D reconstruction method with a low computational cost for 3D reconstruction, the execution time was 17.73 s versus the worst times of Kanade Lucas Tomasi, ETH-3D and Sparse Bundler (Snavely et al., 2006) of 19.56, 93.94 and 162.32 s, respectively. These investigations show how offline reconstruction is a highly complex process with high computational weight.
For online reconstruction, particularly on mobile platforms with limited load and power consumption (Burak et al., 2020), it is necessary to apply PCL simplification in conjunction with low computational weight techniques. (Belkin et al., 2021) proposed a 3D LIDAR reconstruction method with online execution at 2 Hz over static environments. (Vazquez-Arellano et al., 2018) proposed a reconstruction method that captures the static environment data at 5 Hz, but the applied clustering and reconstruction process took 18.45 s. In (Li et al., 2020), a GNSS-denied method obtained reconstructions with periods from 5 s to 60 s in static scenes. These results indicate that 3D reconstruction in static environments with online execution is still far from a real-time implementation, especially in real conditions with dynamic objects present in the environment, where the reconstruction must remove the objects from the scene.

Research on dynamic object detection methods is still an open topic, and it is fundamental to achieve the interaction of autonomous systems in real environments. (Chacon et al., 2019) presented an overview of dynamic object detection techniques in two-dimensional (2D) image sequences highlighting difficulties and advances. The challenge is more critical in dynamic 3D object detection. Dynamic objects are at different times and positions, generating occlusion and therefore making it impossible to reconstruct with a single PCL or by the consecutive joining of PCLs, as they generate distortion in the reconstruction (Ai et al., 2021). For example, (Wang et al., 2020) performed an offline detection of moving objects whose velocities were limited to 10 Hz by using instruments such as 3D LIDAR. (Cordeiro and Pedrino, 2019) established the trajectory of a dynamic object with a maximum velocity of 3.125 m/s. (Zhao et al., 2019) presented an onboard diagnostics method to estimate trajectories of dynamic objects using 3D LIDAR with a 2 Hz run. (Sun et al., 2018) performed dynamic object segmentation by applying dense optical flow estimation with an 8 s per frame run. (Fan et al., 2019) proposed an offline method of dynamic object removal through image fusion on datasets. These new contributions aim to generate a PCL with the smallest number of 3D points belonging to a dynamic object and to achieve a total reconstruction of the scene. Online execution with low computational weight has not yet achieved this goal.

This paper proposes a low computational cost method with online execution for dynamic segmentation and 3D reconstruction in SLAM using the sensory fusion of a 2D LIDAR and an RGB-D camera. The method generates an online reconstruction at a rate of 6.6 Hz of the static environment while a dynamic object performs a walkthrough. The results present the detection of the object using statistical data trending and the resulting reconstruction of the environment by segmentation and concatenations of PCLs.

2. Methodology

The proposed dynamic reconstruction’s base method is high variability point zone segmentation (Rec-HV). The method uses the standard deviation as a measure of the variability of each LIDAR point in a window of n cycles. The presence of dynamic objects causes high variability. RGB-D PCL segmentation uses the areas of higher variability. Rec-HV detects and removes dynamic zones in a scene. Rec-HV is composed of three stages (Figure 1).

Rec-HV performs a LIDAR data statistical analysis when potential dynamic zones are detected. It applies a depth point reduction to the RGB-D PCL. Subsequently, it merges data to segment and remove dynamic zones to obtain the static zone’s PCL and reconstruct a scene by concatenation.

2.1. Sensor fusion stage

The first process consists of preprocessing, a statistical treatment, and a detection subprocess performed every
LIDAR data $k$-th measurement cycle. LIDAR data are depth vector ($r$), start-up sampling angle ($\theta_{\text{min}}$), and increment angle ($\Delta \theta$); these values conform to an angle vector ($\theta$).

The depth vector $r$ presents distance measurements; however, some data are undefined or infinity. The interpolation and linear extrapolation of the two nearest neighbouring points filter these values. This method obtains the filtered polar coordinate vectors $r$ and $\theta$. The statistical treatment subprocess applies a moving average with window size $n$ cycles, the vector of $r$ averages of cycle $k$-th ($\bar{r}_k$), to depth vector $r$ and calculates a standard deviation vector $d\bar{r}$ for each LIDAR point in the window of size $n$.

For each LIDAR data $k$-th cycle, the statistical treatment provides three data vectors: average distance $\bar{r}$, angle $\bar{\theta}$, and standard deviation $d\bar{r}$, as presented in (1):

$$
\bar{r}_k = [\bar{r_1}, \bar{r_2}, \bar{r_3} \ldots \bar{r_i}]
$$

$$
\bar{\theta} = [\theta_1, \theta_2, \theta_3 \ldots \theta_i]
$$

$$
d\bar{r}_k = [d\bar{r}_1, d\bar{r}_2, d\bar{r}_3 \ldots d\bar{r}_i]
$$

Finally, the detection subprocess identifies positions that determine the boundaries of a dynamic zone during each cycle using a maximum values analysis $d\bar{r}_k$.

Rec-HV sets a threshold given by half the $d\bar{r}$ maximum value. It classifies any value exceeding that threshold as a high variability point. It selects two maximum values from this set of values. In this way, Rec-HV divides the $d\bar{r}$ data into zones separated by high variability points since these correspond to abrupt changes due to the presence of dynamic objects.

Rec-HV classifies zones as static or dynamic using $d\bar{r}_{\text{max1}}$ and $d\bar{r}_{\text{max2}}$. Dynamic zones are zones surrounded by maximum variability values. $d\bar{r}_{\text{max1}}$ and $d\bar{r}_{\text{max2}}$ are associated with distances $l_1$ and $l_2$, so the dynamic zone angular limits $\alpha_1$ and $\alpha_2$.

With $\alpha_1$ and $\alpha_2$, Rec-HV determines an angular segment to apply to the PCL.

The reduction process captures the PCL after processing the LIDAR data $k$-th cycle. It contains an extensive data volume that implies a high computational weight for online processing, so a point reduction by a voxel grid is necessary. Thus, Rec-HV obtains a reduced PCL that preserves environmental characteristics with lower density 3D points.

The data fusion process requires an alignment of the LIDAR and RGB-D camera geometric centres. The LIDAR presents an angular aperture of 90° in the XY plane (purple zone), while the RGB-D camera has a 57° of aperture (green zone). The LIDAR and RGB-D camera apertures present a difference corresponding to a previous LIDAR detection angle. Rec-HV determines the dynamic zone borders before an object enters the RGB-D camera detection zone. The detection angles $\alpha_1$ and $\alpha_2$ determine the segmented dynamic zone on the reduced PCL (Figure 2).

2.2. Segmentation stage

The segmentation eliminates all PCL 3D points within the area delimited by the detection angles ($\alpha_1$ and $\alpha_2$) and the origin of the coordinates (Figure 2). For segmentation, it is necessary to obtain the angle of each PCL point to the Y-axis in the XY plane following:

Where $(x_{\text{pts}}, y_{\text{pts}}, z_{\text{pts}})$ is a spatial coordinate, $r_{\text{pts}}$ is a radius, and $\alpha_{\text{pts}}$ is an angle of the Y-axis in the XY plane. The segmentation eliminates the dynamic zone’s 3D points between $\alpha_1$ and $\alpha_2$. Likewise, the 3D points outside this angular zone are considered 3D points of the static zone, which will constitute the PCL of cycle $k(PCL_k)$.

2.3. Reconstruction stage

The reconstruction stage involves completing the segmented $PCL_k$ with the PCL obtained from the $k$-1 cycle ($PCL_{k-1}$) by (2):

$$
PCL_{kT-1} = \begin{bmatrix}
    x_1 & y_1 & z_1 \\
    \vdots & \vdots & \vdots \\
    x_p & y_p & z_p
\end{bmatrix}
\quad PCL_k = \begin{bmatrix}
    x_1 & y_1 & z_1 \\
    \vdots & \vdots & \vdots \\
    x_q & y_q & z_q
\end{bmatrix}
$$

$$
PCL_{\text{trans.}} = \begin{bmatrix}
    x_1 & y_1 & z_1 \\
    \vdots & \vdots & \vdots \\
    x_{p+q} & y_{p+q} & z_{p+q}
\end{bmatrix}
= PCL_{kT-1} || PCL_k
$$

The current cycle calculates a transient PCL ($trans$ PCL). This $PCL_{Trans}$ has overlapping zones where $PCL_{kT-1}$ and $PCL_k$ have 3D points in the scene’s same zone (Figure 3). These overlapping areas increase the point density of $PCL_{Trans}$, so a second voxel grid is applied to obtain $PCL_{kT}$.
This paper proposes a correlation measure using a scene comparison between reference PCL \( (PCL_{\text{ref}}) \) captured in one frame of a static scene without dynamic objects and \( PCL_{kT} \). For this purpose, Rec-HV projects both the \( PCL_{\text{ref}} \) and the \( PCL_{kT} \) on a 2D matrix. Each point coordinates \( (y_{\text{pts}}, z_{\text{pts}}) \) are positions in the matrix. The voxel grid size corresponds to pixel extent for each point, and it normalizes the x-coordinate to the maximum depth detected in \( PCL_{\text{ref}} \). This way, it is possible to obtain a grayscale correspondence to draw each point, generating two images. The image comparison is a measure of the correlation (Figure 4).

2.4. Materials

To develop this research, we used a robotic operating system (ROS V1.0), in Ubuntu 18.06, with a DELL Inspiron N5110 computer, a 3.2 Gb core i7 processor with four real cores and four virtual cores, a 16 Gb RAM, and a 512 MB GPU. The sensors used were an RGB-D type Kinect V1.0 (with 57° horizontal, 43° vertical, a depth of 4.5 m, a depth image of \( 320 \times 240 \), and an RGB image of \( 640 \times 480 \)) with a USB Type-A adapter and a LIDAR RPLIDAR A1 (with a maximum depth of 8 m, 360°, and a cycle time of 143 ms), which were static in the test. The sensors’ installation matches the LIDAR and RGB-D camera geometric centres; the Y-axis of the Kinect coincided with the 90° angle of the LIDAR (Figure 5(a)). The testing environment comprised objects of different materials spaced against a wall. The research uses two types of dynamic objects, a mobile robot consisting of a V5 Robot Brain controller and two V5 Smart Motor motors with 36:1 gearbox and configured at 80% of power, with which the robot reached an average speed of 2.78 m/s. This was equipped with a side screen of \( 64 \text{ cm} \times 27 \text{ cm} \), (Figure 5(b)) and a person who made trips at an average speed of 4 m/s.

3. Results

3.1. Relationship between the computational cost and point cloud size

We tested the dependency relationship among the voxel grid size, PCL size, and computational cost of the Rec-HV method during 100 cycles. For this purpose, we applied a variable voxel grid of 15 values and a specific segmentation angle between 30° and 60° to the PCL. This test was developed in a scenario with a background perpendicular to the x-axis of the sensors, an average depth of 3 m, and highly reflective materials (Figure 6(a)). The research applies a 0.08-m voxel grid to the frames captured in this scenario (Figure 6(b)).

The Rec-HV method exhibited changes in the average run times for different voxel grid values. However, the LIDAR cycle time limited the minimum Rec-HV processing time, which averaged 0.146s and occurred for voxel grids of 0.05 m or larger (depicted by the blue line in Figure 7(a)). The decrease in PCL size directly impacted the computational cost. The decrease in PCL was more relevant during the segmentation processes (as shown by the orange line in Figure 7(a)). The variation of the voxel grid values changed the size of the reduced PCL
processed online (see Figure 7(b)), thereby reducing the computational cost. A voxel grid of 0.08 m for the Rec-HV method reduced the density from 142,681 average 3D points in the PCL to 1,901 3D points in the reduced PCL, thereby reducing the average execution time from 4.36 s to 0.146 s. A voxel grid of 0.08 m reduces the average execution time of the reduced PCL and the segmentation time, which on average took 3.03 s, and its new value is 0.032 s.

3.2. Test with a mobile robot

The Rec-HV method had a 0.08-m voxel grid and a moving average with a window of $n = 4$ cycles. In this test, the dynamic object was a mobile robot. The sensors traversed an indoor environment at 1.5 m (Figure 8).

In this test, two experiments with a static and dynamic scene, respectively, were performed to evaluate the static environment reconstruction. Each experiment included
ten repetitions of 26 cycles. In Experiment 1, the scene remained static; the test obtained a reduced PCL of the initial cycle and calculated its correlation with 25 successive reconstructions. In Experiment 2, the test obtained a static scene with reduced PCL in the first cycle and calculated a reduced PCL in the following 25 cycles with a dynamic object crossing the scene. The experiment concatenated different PCLs in each cycle. We determined the correlation between the static scene PCL and reconstructed PCL in successive cycles in both experiments. In Experiment 1, with a static scene, an average correlation of 85.8% was achieved (Figure 9(a)). The averages obtained from the two correlations were similar.

In Experiment 2, with a dynamic scene, the static environment was reconstructed (Figure 10). The reconstruction is visually related in dimension and distribution to the static scene environment reconstruction, despite some noise resulting from the Rec-HV process (Figure 6(b)).

3.3. Test with a person

The test development was into five scenes with different characteristics (Table 1).

Five hundred execution cycles were captured in each scene while a person performed several round trips at
an average speed of 4 m/s. In each cycle, Rec-HV segmented the dynamic zone (red-coloured dots) from the static zone (smaller multicoloured dots), as shown in the segmentation column in Figure 11, and performed the environment reconstruction with and without the Rec-HV method. In the reconstruction column in Figure 11, the distortion generated by the dynamic object is observable where it was not removed.

The average test duration time for the execution of the 500 cycles was 75.443 s; thus, each cycle had an average execution time of 0.152 s (Table 2). The research only applied segmentation in cycles where high variability points were detected. Depending on the environment and configuration, the segmentation can occur in 50.8% to 93.0% of the cycles.

The RGB-D camera presented different input PCL sizes, ranging from 148,634–239,082 3D points. A 0.08-m voxel grid reduced the PCL between 95% and 99.06%.

The execution test processed between 1,000,650 and 3,706,861 3D points in 500 cycles. Of these, 6% and 34% were segmented and eliminated because they belonged to a dynamic zone (Table 3).

The PCL reconstruction process presented a progressive increase of the occluded 3D points or areas not captured in previous cycles. This increase in reconstructed
Table 4. Metrics of the reconstructed environment percentage increase of 3D points and number of 3D points of PCL.

| Scene | Percentage increase in reconstructed PCL | Reconstruction 3D points |
|-------|-----------------------------------------|--------------------------|
| A     | 63%                                     | 4370                     |
| B     | 61%                                     | 3476                     |
| C     | 264%                                    | 6823                     |
| D     | 69%                                     | 14306                    |
| E     | 147%                                    | 6828                     |

PCLs presented percentages between 61% and 264% (Table 4).

Given that this research resulted in datasets, code, instructions, and online execution videos of the Rec-HV method, a repository is available to the SLAM research community to encourage the development of this line of research.

This research presents the Rec-HV system with online execution for the reconstruction of static environments with the presence of dynamic objects, the reduction of execution time has tremendous importance, as reported by other authors. This paper shows how the factors of execution time and PCL size depend on the value of the voxel grid, which configured at 0.08 m allowed a reconstruction of the environment at a rate of 6.6 Hz, a higher value than reported by authors such as Belkin, Abramenko, & Yudin 2021 that perform it at 2 Hz, without the presence of dynamic objects. The present research also performs...
detection, segmentation, and removal of dynamic objects with online execution. This factor is the most relevant contribution to the research, given that other research has reported offline removal methods (Fan et al., 2019; Wang et al., 2020).

This research presented the behaviour of Rec-HV against different real scenarios with diverse conformation and configuration. In these scenarios, the system managed to perform the reconstruction. However, Rec-HV is susceptible to factors such as the change in environments since this can increase the size of the PCL, generating a possible increase in the computational cost and execution time. The speed of the dynamic object can influence the Rec-HV results since dynamic objects at high or low speeds do not generate enough disturbance for detection.

4. Conclusions

Rec-HV demonstrated that by employing high variability points identified in the LIDAR data, it is possible to establish the limits of the dynamic zone and thus segment and eliminate the dynamic object in each cycle. Given that the average reconstruction correlation is 84.8%, this research corroborates that the Rec-HV method enables the recovery of static scenes even when dynamic objects are present.

The results show that the PCL reduction in Rec-HV by a voxel grid directly impacts its execution’s computational cost. A voxel grid of 0.08 m in Rec-HV permitted a reduction of the average execution time of 96.65% and an average segmentation time of 98.94%; these factors are of great importance for its online execution.

The tests evidenced that even when Rec-HV only preserves between 0.77% and 5.00% of the PCL, the scenario features are not lost, thus allowing its reconstruction.

According to the PCL increase percentages in the reconstruction with values between 61% and 264%, the Rec-HV method enables reconstructing sectors of a scene that, due to the presence of dynamic objects, were occluded or not captured in previous cycles.

The data that support the findings of this study are openly available on Github at https://github.com/MAB144-Python/SLAM-REC-HV-version-1.0-test.

Disclosure statement

No potential conflict of interest was reported by the author(s).

ORCID

Brayan Andru Montenegro http://orcid.org/0000-0001-5409-5998
Juan Fernando Flórez http://orcid.org/0000-0003-1646-4419
Elena Muñoz http://orcid.org/0000-0002-3919-583X

References

Ai, Y., Rui, T., Yang, X., He, J., Fu, L., Li, J., & Lu, M. (2021). Visual SLAM in dynamic environments based on object detection. Defence Technology, 17(5), 1712–1721. https://doi.org/10.1016/j.dt.2020.09.012

Belkin, I., Abramenko, A., & Yudin, D. (2021). Real-Time lidar-based localization of mobile ground robot. Procedia Computer Science, 186, 440–448. https://doi.org/10.1016/j.procs.2021.04.164

Bura, A., Kadir, A., & Yakut, O. (2020). Four-legged hunter (FLH) robot: Design and shooting control to moving targets with SMC. Simulation Modelling Practice and Theory, 104, 102117. https://doi.org/10.1016/j.simpat.2020.102117.

Cao, M., Jia, W., Lv, Z., Li, Y., Xie, W., Zheng, L., & Liu, X. (2019). Fast and robust feature tracking for 3D reconstruction. Optics & Laser Technology, 110, 120–128. https://doi.org/10.1016/j.optlastec.2018.05.036

Chacon, M. I., Guzman, A., Ramirez, G., & Ramirez, J. A. (2019). A novel instrument to compare dynamic object detection algorithms. Image and Vision Computing, 88, 19–28. https://doi.org/10.1016/j.imavis.2019.04.006

Cordeiro, N., & Pedrino, E. (2019). A new methodology applied to dynamic object detection and tracking systems for visually impaired people. Computers & Electrical Engineering, 77, 61–71. https://doi.org/10.1016/j.cpleeng.2019.05.003

Delanoy, J., Coeurjolly, D., Lachaud, J., & Bousseau, A. (2019). Combining voxel and normal predictions for multi-view 3D sketching. Computers & Graphics, 82, 65–72. https://doi.org/10.1016/j.cag.2019.05.024

Fan, Y., Han, H., Tang, Y., & Zhi, T. (2019). Dynamic objects elimination in SLAM based on image fusion. Pattern Recognition Letters, 127, 191–201. https://doi.org/10.1016/j.patrec.2018.10.024

Huynh, T., Hua, C., Anh Tu, N., & Kim, D. (2020). Learning 3D spatiotemporal gait feature by convolutional network for person identification. Neurocomputing, 397, 192–202. https://doi.org/10.1016/j.neucom.2020.02.048

Jinyu, L., Bangbang, Y., Danpeng, C., Nan, W., Guofeng, Z., & Hujun, B. (2019). Survey and evaluation of monocular visual-inertial SLAM algorithms for augmented reality. Virtual Reality & Intelligent Hardware, 1(4), 386–410. https://doi.org/10.1016/j.vrih.2019.07.002

Li, S., Li, G., Wang, L., & Qin, Y. (2020). SLAM integrated mobile mapping system in complex urban environments. ISPRS Journal of Photogrammetry and Remote Sensing, 166, 316–332. https://doi.org/10.1016/j.isprsjprs.2020.05.012

Liu, H., & Wang, L. (2021). Collision-free human–robot collaboration based on context awareness. Robotics and Computer-Integrated Manufacturing, 67, 101997. https://doi.org/10.1016/j.rcim.2020.101997.

Mazurek, P., Wagner, J., Mießkina, A., & Morawski, R. Z. (2020). Comparison of sixteen methods for fusion of data from impulse-radar sensors and depth sensors applied for monitoring of elderly persons. Measurement, 154, 107455. https://doi.org/10.1016/j.measurement.2019.107455

Sahin, C., Garcia, G., Sock, J., & Kim, T.-K. (2020). A review on object pose recovery: From 3D bounding box detectors to full 6D pose estimators. Image and Vision Computing, 96, 103898. https://doi.org/10.1016/j.imavis.2020.103898

Siddiqua, A., & Fan, G. (2019). Semantics-enhanced supervised deep autoencoder for depth image-based 3D model retrieval.
Snavely, N., Seitz, S. M., & Szeliski, R. (2006). Photo tourism: Exploring photo collections in 3D. *ACM Transactions on Graphics*, 25(3), 835–846. https://doi.org/10.1145/1141911.1141964

Sun, Y., Liu, M., & Meng, M. Q. (2018). Motion removal for reliable RGB-D SLAM in dynamic environments. *Robotics and Autonomous Systems*, 108, 115–128. https://doi.org/10.1016/j.robot.2018.07.002

Vázquez M., R., Paraforosa, D., Garrido, D., C, M., Burce, M. E., & Griepentrog, H. W. (2018). 3-D reconstruction of maize plants using a time-of-flight camera. *Computers and Electronics in Agriculture*, 145, 235–247. https://doi.org/10.1016/j.compag.2018.01.002

Wang, L., Zhao, D., Wu, T., Fu, H., Wang, Z., Xiao, L., ... Dai, B. (2020). Drosophila-inspired 3D moving object detection based on point clouds. *Information Sciences*, 534, 154–171. https://doi.org/10.1016/j.ins.2020.05.006

Woźniak, M., Paszynski, M., Pardo, D., & Dalcin, L. D. (2015). Computational cost of isogeometric multi-frontal solvers on parallel distributed memory machines. *Computer Methods in Applied Mechanics and Engineering*, 284, 971–987. https://doi.org/10.1016/j.cma.2014.11.020

Ye, X., Chen, S., & Xu, R. (2021). DPNet: Detail-preserving network for high quality monocular depth estimation. *Pattern Recognition*, 109, 107578. https://doi.org/10.1016/j.patcog.2020.107578

Zhao, J., Xu, H., Liu, H., Wu, J., Zheng, Y., & Wu, D. (2019). Detection and tracking of pedestrians and vehicles using roadside LiDAR sensors. *Transportation Research Part C: Emerging Technologies*, 100, 68–87. https://doi.org/10.1016/j.trc.2019.01.007