



Detección y seguimiento de entidades de alta reflectividad en tiempo real mediante clustering adaptativo de nubes de puntos de LiDAR 3D

Luis Alfredo Calle^{1,3}, Felipe Jiménez², Pascual Campoy³

¹ Grupo de Investigación en Simulación, Optimización y Toma de Decisiones, Universidad Politécnica Salesiana, lcallea@ups.edu.ec

² Instituto Universitario de Investigación del Automóvil. Universidad Politécnica de Madrid, felipe.jimenez@upm.es

³ Computer Vision and Aerial Robotics Group (CVAR-UPM). Centre for Automation and Robotics C.A.R. (UPM-CSIC). Universidad Politécnica de Madrid, pascual.campoy@upm.es

RESUMEN

Los algoritmos modernos de SLAM integran información semántica para mejorar la comprensión del entorno. La detección y segmentación de entidades características clave permiten una localización y un mapeo más precisos en entornos complejos como los que se enfrenta un vehículo parcial o totalmente automatizado.

Los algoritmos de SLAM semántico actuales no aprovechan completamente las características de los sensores LiDAR 3D como la reflectividad. En este trabajo se presenta un sistema integral para la detección y seguimiento en tiempo real de entidades de alta reflectividad como letreros, matrículas y señales de tráfico, utilizando datos de nubes de puntos provenientes de sensores LiDAR 3D. El uso de esta información para filtrar y rastrear objetos con alta reflectividad ha sido poco explorado en la literatura. Integrar la reflectividad mejora la precisión en la detección de objetos en entornos complejos a un coste computacional relativamente bajo.

El enfoque se fundamenta en procesar los datos de LiDAR para identificar y rastrear clústeres a lo largo del tiempo. Inicialmente, se aplica un filtrado a los datos de la nube de puntos basado en un umbral de reflectividad. Este proceso elimina los puntos que no tienen interés disminuyendo notablemente la densidad de la muestra original para el análisis a objetos de alta reflectividad, mejorando la eficiencia y precisión del procesamiento. Además, se emplea el algoritmo DBSCAN adaptado para funcionar en una segmentación regional para optimizar la identificación de clústeres en diversas zonas del espacio el cual se divide en regiones mediante umbrales de distancia, permitiendo ajustar dinámicamente los parámetros de clustering para mejorar la detección tanto de objetos cercanos como lejanos en los cuales la distancia y cantidad de puntos varían en función de la distancia al sensor. Para el seguimiento de objetos, el sistema asocia clústeres detectados en secuencias temporales basándose en la proximidad espacial. Se utiliza la marca de tiempo de la nube de puntos para determinar de manera precisa la diferencia de tiempo entre mediciones permitiendo estimar la velocidad instantánea de los objetos que han sido rastreados durante un número suficiente de escaneos LiDAR; esto permite determinar la persistencia de cada objeto, asegurando que solo los objetos relevantes sean considerados, eliminando falsos clústeres.

Los resultados demuestran la efectividad del sistema en entornos dinámicos reales de carretera, siendo capaz de detectar y seguir múltiples objetos de alta reflectividad en tiempo real, demostrando ser eficiente y robusto al manejar correctamente los casos sin detecciones, evitando conflictos en la gestión de identificadores de clústeres. A su vez, tras esta identificación, se pueden plantear clasificaciones más finas para incrementar la precisión en las estimaciones de posicionamiento y mejorar la calidad de la reconstrucción del entorno.

1. Introducción

Los avances constantes en las tecnologías de percepción han permitido que los sistemas autónomos aborden situaciones complejas de manera más efectiva. En este sentido la localización y el mapeado simultáneos (SLAM por sus siglas en inglés), basados en LiDAR se han consolidado como herramientas vitales para aplicaciones que necesitan una comprensión precisa del espacio como la conducción autónoma y la navegación robótica [1][2].

Recientemente se ha observado un aumento en el uso de la reflectividad del LiDAR como una herramienta beneficiosa para mejorar la segmentación y detección de objetos. Por ejemplo, en investigaciones como la referenciada en [3], se destaca cómo la reflectividad puede potenciar el desempeño de los algoritmos de segmentación semántica en entornos complejos. Además, hay trabajos actuales que han creado conjuntos de datos avanzados como DurLAR, incluyendo información panorámica sobre reflectividad para aplicaciones multimodales en conducción autónoma [4]. Otros métodos investigan la división de marcas viales y superficies reflectantes en aplicaciones industriales para resaltar la flexibilidad de esta tecnología [5].

En este trabajo, se propone un sistema integral que aproveche las capacidades de los sensores LiDAR 3D para detectar y rastrear entidades de alta reflectividad en tiempo real. Este enfoque presenta optimizaciones por medio de clustering adaptativo y segmentación semántica, combinando los datos espaciales con los de reflectividad para superar las limitaciones presentes en los enfoques existentes, ya que a diferencia de otros enfoques en el estado del arte que se centran en la segmentación semántica general o en la integración de datos para mapas estáticos, este sistema utiliza reflectividad como un atributo clave para mejorar la detección y seguimiento en entornos dinámicos. Además, introduce un algoritmo de clustering adaptativo basado en DBSCAN, ajustado a regiones espaciales, y un filtrado inicial que reduce la densidad de datos y optimiza el rendimiento computacional, así también incorpora un mecanismo de asociación temporal para estimar velocidades y garantizar la persistencia de objetos detectados, algo que no es común en estudios previos.

2. Estado del Arte

El SLAM basado en LiDAR ha evolucionado significativamente desde sus inicios, integrando algoritmos más sofisticados y datos semánticos para mejorar la percepción y el mapeo. Tradicionalmente, los enfoques se centran en la geometría de las nubes de puntos, como lo describen Durrant-Whyte y Bailey en sus estudios clásicos [1][2]. Sin embargo, el uso de la reflectividad ha emergido como un complemento crítico.

Estudios recientes han demostrado que la reflectividad permite segmentar objetos de manera más precisa en entornos complejos. Zhao et al. [3] propusieron el uso de picos de intensidad y reflejos para mejorar la clasificación en mapas SLAM, abordando los desafíos de superficies reflectantes. En un enfoque similar, el conjunto de datos DurLAR integra información de reflectividad y ambiente, permitiendo aplicaciones multimodales de alta precisión en vehículos autónomos [4]. Además, Certad et al. [5] introdujeron un método para segmentar marcas viales utilizando reflectividad, lo que mejora la detección de carriles bajo condiciones adversas.

Otro avance notable es la aplicación de algoritmos de clustering optimizados para clasificar puntos en entornos no estructurados. Estudios como los de Viswanath et al. [6] destacan la combinación de reflectividad y segmentación geométrica para mejorar la precisión en mapas semánticos. Adicionalmente, Dubé et al. [7] desarrollaron SegMap, una solución basada en segmentos para la representación eficiente de mapas 3D utilizando descriptores impulsados por datos. Por último, Cao et al. [8] propusieron un método iterativo de puntos más cercanos (ICP) que combina características semánticas para mejorar la precisión de la odometría LiDAR.

Este trabajo aporta un enfoque integral que combina prefiltrado por reflectividad, clustering adaptativo basado en DBSCAN y seguimiento temporal de clústeres, mejorando la detección y persistencia de entidades relevantes, como señales de tráfico y patentes, al tiempo que filtra entidades irrelevantes. Además, incorpora el cálculo de velocidades mediante ventanas deslizantes, optimizando el procesamiento en tiempo real y superando las limitaciones de los métodos existentes.

3. Metodología

El enfoque propuesto combina segmentación adaptativa, clustering dinámico y seguimiento temporal, lo que permite manejar datos LiDAR en tiempo real con una baja carga computacional luego del filtrado de entidades por criterios de alta reflectividad para eliminar ruido, adaptar parámetros y seguir objetos a través de escaneos garantizando su utilidad en entornos dinámicos y aplicaciones críticas como la navegación autónoma. En la figura 1 se muestra el diagrama de flujo de la implementación de algoritmo.

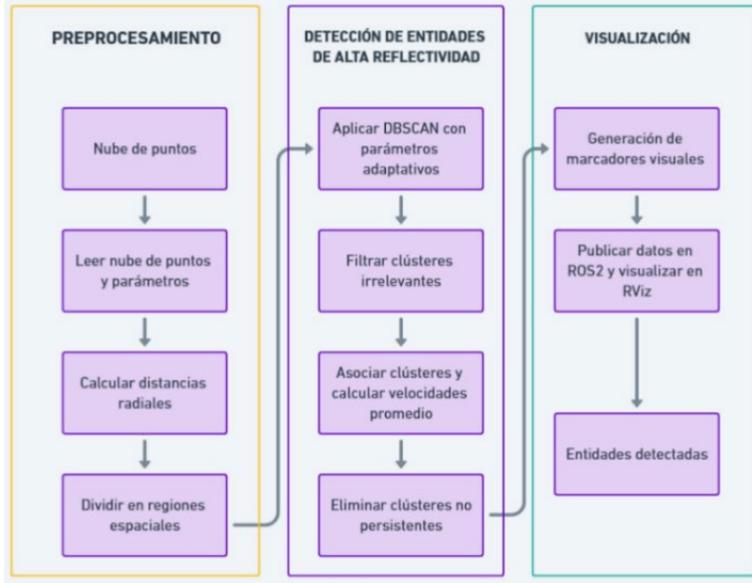


Figura 1: Diagrama de flujo para seguimiento de entidades reflectivas por persistencia de clústeres.

3.1. Definición de la Nube de Puntos

Consideremos un sensor LiDAR con N_L capas y N_P puntos por capa. Una única exploración del sensor produce una nube de puntos P_A definida como un conjunto $N_L \times N_P$ de puntos p_{ij} en \mathbb{R}^8 .

$$p_{ij} = [x_{ij} \ y_{ij} \ z_{ij} \ t_{ij} \ r_{ij} \ signal_{ij} \ R_{ij} \ near_ir_{ij}] \in \mathbb{R}^8 \quad (1)$$

Componentes del Punto p_{ij} :

- $x_{ij} \ y_{ij} \ z_{ij}$: Coordenadas espaciales tridimensionales del punto, medidas en el marco de referencia del sensor LiDAR.
- t_{ij} : Marca temporal asociada al punto, indicando el tiempo exacto de adquisición dentro de la exploración.
- r_{ij} : Distancia desde el marco de referencia del sensor hasta el punto.
- $signal_{ij}$: Intensidad de la señal reflejada, una medida de la energía de retorno detectada por el sensor.
- R_{ij} : Reflectividad del punto, que caracteriza las propiedades del material reflejado.
- $near_ir_{ij}$: Valor asociado al infrarrojo cercano, útil para análisis específicos del material.
- i : Índice de la capa en la que se encuentra el punto, un entero que varía entre 1 y N_L , el número total de capas del sensor.
- j : Índice del punto dentro de la capa i , un entero entre 1 y N_P , el número total de puntos por capa.

3.2. Filtrado por reflectividad:

En el filtrado por reflectividad se utiliza un umbral predefinido para seleccionar únicamente los puntos de la nube proporcionada por el sensor LiDAR que presentan niveles específicos de reflectividad, con el objetivo de reducir la densidad de datos a ser procesados, enfocándose en aquellos puntos que tienen una mayor probabilidad de representar objetos de interés, como señales de tráfico, matrículas u otros elementos altamente reflectantes.

Se define un umbral de reflectividad basado en las características esperadas de los objetos relevantes. Por ejemplo, objetos como señales de tráfico o letreros suelen tener valores de reflectividad superiores debido a su material retro reflectante.

Los puntos con reflectividad inferior al umbral son descartados, mientras que aquellos que lo superan son retenidos para su posterior procesamiento.

Dado un umbral inferior R_L y un umbral superior R_U , definimos el conjunto filtrado P_B a partir de la nube de puntos P_A como:

$$P_B = \{p_{ij} \in P_A : R_L \leq R_{ij} \leq R_U ; 0 \leq i < N_L ; 0 \leq j < N_P\} \quad (2)$$

En la Figura 2, se observa la aplicación del filtrado por criterio de umbral de reflectividad según la ecuación (2).

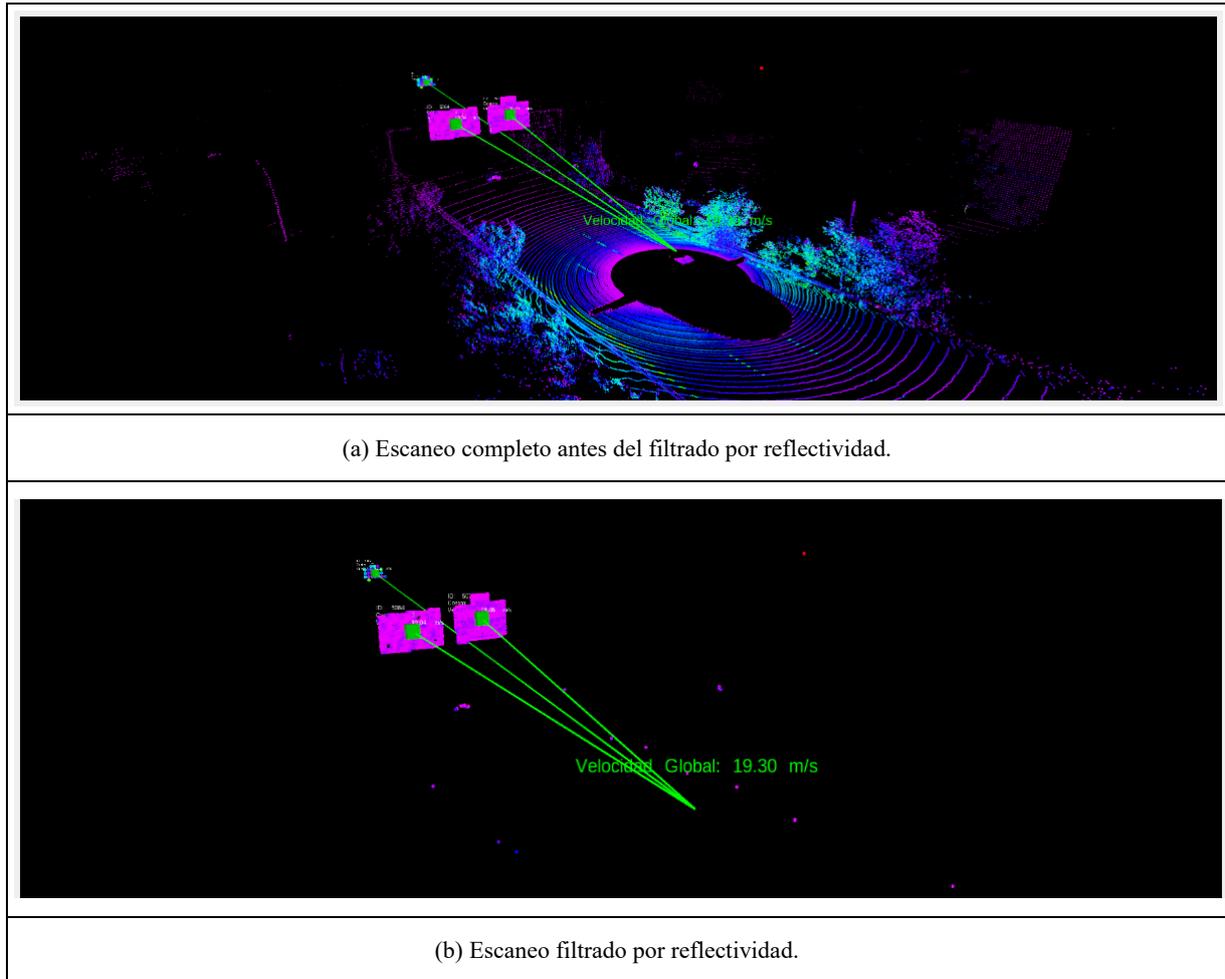


Figura 2: Aplicación del filtrado por umbral de reflectividad.

3.3. Preprocesamiento:

El algoritmo comienza con la adquisición de una nube de puntos LiDAR y la configuración de los parámetros necesarios para el procesamiento. Cada punto de la nube se preprocesa calculando su distancia radial al sensor, mediante la ecuación (3), lo que permite clasificar los puntos según su proximidad. Esta operación inicial proporciona una base para dividir el espacio en regiones específicas, optimizando las etapas posteriores.

$$d_i = \sqrt{x_i^2 + y_i^2 + z_i^2}, \quad \forall i \in P \quad (3)$$

3.4. Segmentación del Espacio:

La nube de puntos se divide en múltiples regiones radiales, cada una con características propias de densidad de puntos. Estas regiones son:

Cercanas al sensor, donde la densidad de puntos es mayor.

$$R1: d_i < r_1 \quad (4)$$

$$R2: r_1 \leq d_i < r_2 \quad (5)$$

Intermedias y lejanas, donde la densidad disminuye progresivamente. La división permite ajustar los parámetros de análisis a las características de cada región, mejorando la precisión y reduciendo el ruido en los datos.

$$R3: r_2 \leq d_i < r_3 \quad (6)$$

$$R4: d_i \geq r_3 \quad (7)$$

En la Figura 3 se puede observar las regiones radiales descritas en las restricciones condicionales (4), (5), (6) y (7).

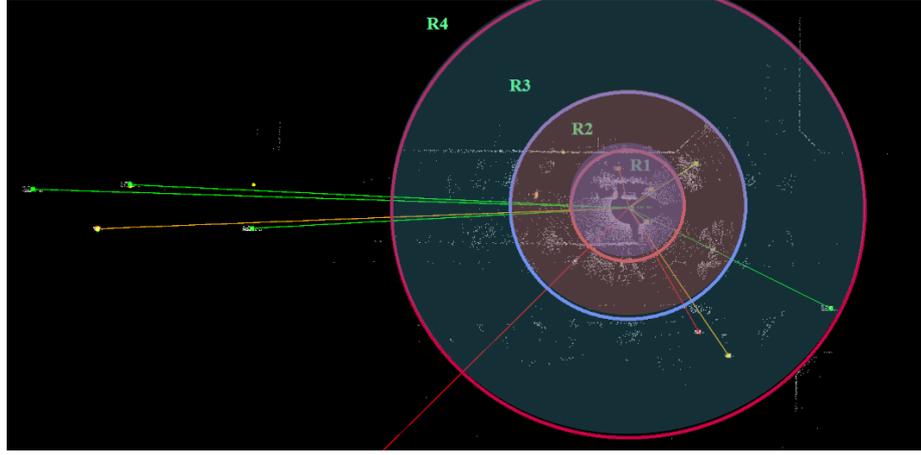


Figura 3: Segmentación de nube de puntos por regiones.

3.5. Clustering Adaptativo por región con DBSCAN:

El algoritmo empleado para la generación de clústeres es DBSCAN (Density-Based Spatial Clustering of Applications with Noise), un método de agrupamiento basado en densidad que segmenta la nube de puntos en función de las regiones densamente pobladas en el escaneo. Este enfoque resulta especialmente adecuado en el contexto de vehículos autónomos, donde la cantidad de clústeres presentes en el entorno es desconocida y varía según las características del espacio explorado. A diferencia de otros métodos de agrupamiento, DBSCAN no requiere una configuración previa del número de clústeres a buscar, lo que lo hace robusto y adaptable a escenarios dinámicos. Las entidades de alta reflectividad como señales de tráfico o carteles suelen concentrar una mayor densidad de puntos en su región geométrica central, mientras que el soporte o poste asociado tiene una densidad significativamente menor. Esto permite que el algoritmo defina una vecindad centrada en la figura geométrica de la entidad, delimitada por un radio ϵ , para identificar y agrupar puntos relevantes. De esta forma, DBSCAN asegura que ciertos elementos no sean clasificados como señales de tráfico ni considerados en el cálculo de la persistencia de los clústeres.

Este método a pesar de que en el peor de los escenarios su implementación puede presentar una complejidad de una alta carga computacional y uso de memoria, al realizar un pre filtrado de acuerdo con la reflectividad la cantidad de puntos de la escena disminuye de ciento treinta y un mil a una media de quinientos puntos por lo que el tiempo de procesamiento de la creación de los clústeres es menor al periodo de muestreo de los escaneos proporcionados por el Lidar, lo que beneficia al sistema ya que no genera latencia en este proceso.

Para cada región definida, se aplica el algoritmo de clustering DBSCAN, adaptado con parámetros específicos como ϵ (radio de vecindad) y $min_samples$ (número mínimo de puntos para formar un clúster). Estos parámetros varían según la región.

En áreas cercanas, se usan valores más restrictivos debido a la alta densidad de puntos, regiones 1 y 2. En áreas lejanas, regiones 3 y 4, los valores son más permisivos para adaptarse a la menor densidad. Esta individualización asegura que los clústeres representen con precisión los objetos relevantes en cada zona.

Tras identificar los clústeres, se realiza un filtrado basado en criterios específicos. Se descartan clústeres que no cumplen con un umbral mínimo de altura, $h = z_{max} - z_{min}$ con el criterio $h < u_{alt}$, ya que suelen representar ruido o elementos no relevantes, como el suelo. Además, se eliminan clústeres cuya distancia promedio entre puntos no sea significativa, lo que contribuye a optimizar el procesamiento.

3.6. Seguimiento Temporal:

El seguimiento de clústeres persistentes se basa en asociar clústeres detectados en un escaneo actual con clústeres de escaneos previos, utilizando la proximidad espacial y la continuidad temporal, para esto se realiza lo siguiente:

Con el conjunto de clústeres en el escaneo t :

$$C_t = \{c_1^t, c_2^t, c_3^t, \dots, c_{N_t}^t\} \quad (8)$$

Donde c_i^t representa el i -ésimo clúster en el escaneo t y N_t es el número total de clústeres de los cuales se calcula su centroide en términos de coordenadas (x, y, z) mediante la siguiente definición:

$$\mu(c_i^t) = \left(\frac{1}{|c_i^t|} \sum_{k=1}^{|c_i^t|} x_k, \frac{1}{|c_i^t|} \sum_{k=1}^{|c_i^t|} y_k, \frac{1}{|c_i^t|} \sum_{k=1}^{|c_i^t|} z_k \right) \quad (9)$$

El centroide del clúster, denotado como $\mu(c_i^t)$, se calcula como el promedio aritmético de las coordenadas espaciales de todos los puntos (x_k, y_k, z_k) que pertenecen al clúster. Siendo $|c_i^t|$ es el numero total de puntos en clúster c_i^t .

Posteriormente el centroide de un clúster $\mu(c_i^t)$ se compara con el centroide del clúster del escaneo anterior $\mu(c_j^{t-1})$.

$$\text{dist}(\mu(c_i^t), \mu(c_j^{t-1})) \leq \delta \quad (10)$$

Donde δ es un umbral de proximidad.

Luego de la asociación de clústeres se realiza el cálculo de la velocidad promedio, para esto se utiliza una ventana deslizante para suavizar variaciones y garantizar estabilidad en las estimaciones a lo largo del tiempo.

Para cada clúster c_i^t , se mantiene un historial $V(c_i^t)$ que almacena las velocidades calculadas en escaneos anteriores:

$$V(c_i^t) = \{v(c_i^{t-k}), v(c_i^{t-k+1}), \dots, v(c_i^t)\} \quad (11)$$

Donde k es el tamaño de la ventana deslizante.

La velocidad instantánea de c_i^t se calcula mediante:

$$v_{inst}(c_i^t) = \frac{\mu(c_i^t) - \mu(c_j^{t-1})}{\Delta t} \quad (12)$$

Donde:

- $\mu(c_i^t)$ es el centroide actual.
- $\mu(c_j^{t-1})$ es el centroide del clúster asociado en el escaneo previo.
- Δt es el intervalo de tiempo entre los fotogramas t y $t - 1$.

Finalmente para el cálculo de la velocidad promedio con ventana deslizante se ocupa la siguiente definición:

$$v_{prom}(c_i^t) = \frac{1}{\min(k, t)} \sum_{n=\max(1, t-k+1)}^t v_{inst}(c_i^n) \quad (13)$$

Durante el seguimiento temporal, se calcula la velocidad promedio de cada clúster a partir de las posiciones actuales y previas, teniendo en cuenta la diferencia temporal entre fotogramas. Estas velocidades se almacenan en un historial con una ventana deslizante para suavizar variaciones y proporcionar estimaciones más estables eliminando fluctuaciones abruptas al promediar las velocidades instantáneas reduciendo así el impacto de los errores o mediciones atípicas.

Si un clúster actual coincide con uno previo dentro de un umbral, se actualizan su posición y su historial de velocidades, tal y como se describe en la ecuación (14).

$$C_t^{persistentes} = \{c_i^t \in C^t : \exists c_j^{t-1} \in C_{t-1} \text{ tal que } \text{dist}(\mu(c_i^t), \mu(c_j^{t-1})) \leq \delta \} \quad (14)$$

Los clústeres no detectados en un fotograma incrementan un contador de pérdida, lo que permite gestionar su persistencia en el tiempo. Si un clúster no se detecta durante un número definido de escaneos consecutivos, se elimina.

Esta representación matemática, ecuación (14), define cómo se rastrean y gestionan los clústeres a través de múltiples escaneos, asegurando la persistencia de objetos relevantes y eliminando ruido o clústeres transitorios.

En la Figura 4, se puede observar la imagen de proyección esférica y el escaneo Lidar con la detección de las entidades de alta reflectividad.

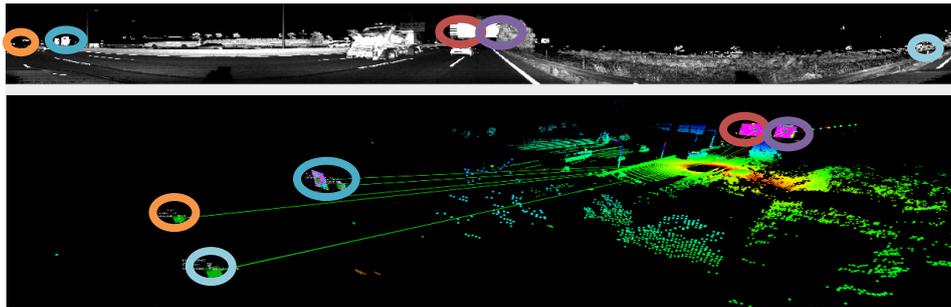


Figura 4: Segmentación de nube de puntos por regiones.

4. Resultados

Los experimentos se llevaron a cabo en entornos dinámicos reales de operación de un vehículo autónomo. Estos entornos incluían alta variabilidad en factores, como la densidad del tráfico, la presencia de vehículos en movimiento y elementos estacionarios, así como peatones, señales de tráfico y carteles. Los escenarios fueron seleccionados para evaluar la robustez del algoritmo en la detección, segmentación y seguimiento de clústeres en tiempo real bajo condiciones cambiantes y complejas. En cuanto a los escenarios de prueba, se incluyeron tanto tramos urbanos como de carretera, con características geométricas diversas, como rectas, curvas y pendientes. Se evaluó el comportamiento del algoritmo en la detección de entidades tanto en movimiento como estacionarias, así como su desempeño en intersecciones y zonas de cruce. Estos entornos dinámicos permitieron medir la capacidad del sistema para identificar clústeres persistentes en presencia de múltiples objetos en movimiento. En la Figura 5 se puede observar la detección de las entidades altamente reflectivas tanto en zona urbana como en carretera.

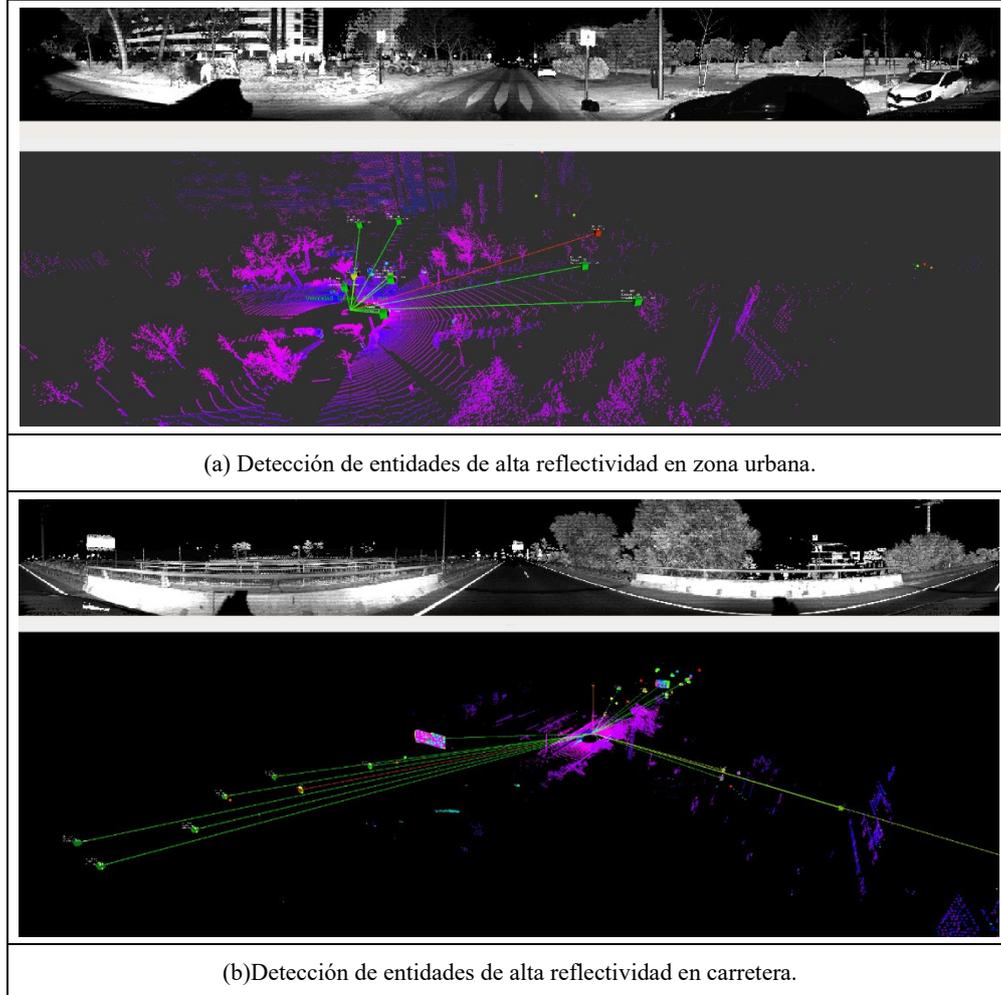


Figura 5: Aplicación del filtrado por umbral de reflectividad.

En [5], se utilizan diferentes métricas como: Precision, Recall y F1-score, para evaluar la efectividad y robustez del algoritmo. A continuación se presentan las definiciones matemáticas de cada una de ellas:

$$Precision = \frac{TP}{TP+FP} \quad (15)$$

$$Recall = \frac{TP}{TP+FN} \quad (16)$$

$$F1 \text{ score} = 2 * \frac{Precision * Recall}{Precision + Recall} \quad (17)$$

Donde:

- *TP*: son las detecciones verdaderas.
- *FP*: Son detecciones erróneas.
- *FN*: Son detecciones falsas

El experimento se diseñó para evaluar la capacidad del algoritmo en la detección y segmentación precisa de entidades positivas y negativas presentes en nubes de puntos LiDAR obtenidas en entornos dinámicos urbanos y de carretera. El objetivo principal fue medir el desempeño del sistema en la clasificación correcta de objetos relevantes para la navegación autónoma, como letreros, carteles, señales de tráfico y patentes de vehículos, mientras se filtran entidades irrelevantes o ruidos, como reflejos en materiales de paredes u otras superficies.

Como entidades positivas se considera:

- **Letreros y carteles:** Los cuales incluyen señales informativas, publicitarias o direccionales, con alta reflectividad y geometrías regulares. Estas entidades suelen estar ubicadas a los costados de las carreteras o en intersecciones estratégicas.
- **Señales de Tráfico:** Se caracterizan por su alta densidad de puntos en las áreas reflectivas y formas definidas.
- **Patentes de Vehículos:** Placas de matrícula detectadas en la parte frontal o trasera de los vehículos y presentan propiedades de reflectividad moderada, con tamaños y ubicaciones predecibles en el espacio.

Como entidades negativas se considera:

- **Reflejos en Paredes y Superficies:** reflejos especulares generados por materiales lisos o metálicos presentes en el entorno. Estas superficies producen puntos dispersos y de baja densidad que no corresponden a entidades deseables.
- **Materiales Reflectivos no Relevantes:** Elementos como ventanas, espejos, o superficies brillantes que generan datos inconsistentes. Se caracterizan por distribuciones irregulares y geometrías no asociadas a señales relevantes.

Luego de la aplicación del experimento tanto en zona urbana como en carretera, se contabilizó las entidades positivas y negativas, para luego aplicar las ecuaciones (15), (16) y (17), obteniendo los resultados presentados en la Tabla 1:

Tabla 1: Resultados obtenidos

	Columnas		
	Precision	Recall	F1 score
Carretera	94.44%	95.03%	94.74%
Zona urbana	93.75%	91.84%	92.78%

5. Conclusiones

El sistema desarrollado mostró un desempeño robusto y eficiente en la detección y clasificación de entidades relevantes en nubes de puntos LiDAR, como señales de tráfico, carteles y patentes de vehículos, mientras filtra eficazmente elementos irrelevantes como reflejos en superficies.

En situaciones de tráfico en carretera, el sistema logró una puntuación de precisión y cobertura F1 del 94, 74 %, demostrando su capacidad para equilibrar ambas características en entornos menos complejos. En áreas urbanas, donde la densidad y variabilidad del entorno son mayores, se obtuvo un porcentaje de F1 del 92, 78 %, lo que confirma la capacidad de adaptación del sistema ante desafíos en entornos más complejos con más elementos reflectivos. Estas mediciones respaldan la efectividad del algoritmo en contextos de navegación autónoma, donde una detección precisa y constante es fundamental. El empleo de técnicas tales como el filtrado previo por reflectividad y el agrupamiento adaptativo basado en DBSCAN resultó crucial para lograr estos resultados; facilitando una segmentación precisa y eficiente de la información. Asimismo, el seguimiento temporal de clústeres persistentes aseguró la consistencia en la identificación de entidades a lo largo de varios escaneos temporales; reduciendo al mínimo los falsos positivos y mejorando la persistencia de los clústeres pertinentes. Aunque el sistema demostró un rendimiento ligeramente superior en carreteras; los resultados en áreas urbanizadas evidencian su solidez y capacidad para adaptarse ante escenarios dinámicos y complejos. Estos resultados confirman que puede ser una herramienta útil para la detección de entidades altamente reflectiva para vehículos autónomos en tiempo real.

Como línea futura de investigación, se propone desarrollar un sistema avanzado de reconocimiento de entidades que permita clasificarlas de manera precisa, reduciendo detecciones falsas y optimizando la precisión global del sistema. Asimismo, se plantea integrar información semántica para identificar y filtrar objetos dinámicos presentes en el entorno, con el objetivo de mitigar su impacto en la estimación de la odometría, mejorando la robustez y confiabilidad del sistema en escenarios complejos.

6. Agradecimientos

Los resultados de este artículo forman parte del proyecto PID2022-140554OB-C31 financiado por MCIN/AEI /10.13039/501100011033/ y FEDER 'Una manera de hacer Europa'.

7. Referencias

- [1] Durrant-Whyte H. and Bailey T. “Simultaneous localization and mapping: Part I,” *IEEE Robotics & Automation Magazine*, **13**(2), 99–110, (2006).
- [2] Yue X. *et al.*, “LiDAR-based SLAM for robotic mapping: state of the art and new frontiers,” College of Engineering and Design, Hunan Normal University, (2022).
- [3] Zhao X. *et al.*, “Mapping with reflection: Detection and utilization of reflection in 3D LiDAR scans,” *Actas del IEEE International Conference on Robotics and Automation*, Hong Kong, 2021.
- [4] Li L. *et al.*, “DurLAR: A High-Fidelity 128-Channel LiDAR Dataset with Panoramic Ambient and Reflectivity Imagery for Multi-Modal Autonomous Driving Applications”, *Actas del International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission*, London, 2021.
- [5] Certad N. *et al.*, “Road Markings Segmentation from LiDAR Point Clouds using Reflectivity Information” *Actas del IEEE International Conference on Vehicular Electronics and Safety (ICVES)*, Bogotá 2022.
- [6] Viswanath K. *et al.*, “Reflectivity Is All You Need!: Advancing LiDAR Semantic Segmentation,” Texas A&M University, arXiv:2403.13188, (2024).
- [7] Dubé R. *et al.*, “SegMap: 3D Segment Mapping Using Data-Driven Descriptors,” *Actas del Robotics: Science and Systems*, Pennsylvania, 2018.
- [8] Cao Q. *et al.*, “An Iterative Closest Point Method for Lidar Odometry with Fused Semantic Features”, *Applied Sciences*, **13**, 12741, 1–12, (2023).