CEA Comité Español de Automática
XLVII JORNADAS DE AUTOMÁTICA CÓRDOBA 2026
UNIVERSIDADE DA CORUÑA

Jornadas de Automática

Mapeo semántico de vocabulario abierto mediante 3D Gaussian Splatting

Serrano-Mesa, J.*, Moncada-Ramirez, J., Ojeda, P., Ambrosio-Cestero, G., Ruiz-Sarmiento, J.R., Gonzalez-Jimenez, J.

Grupo de Percepción Artificial y Robótica Inteligente (MAPIR), Dept. de Ingeniería de Sistemas y Automática, Instituto Universitario en Ingeniería Mecatrónica y Sistemas Ciberfísicos (IMECH.UMA), Universidad de Málaga, Blvr. Louis Pasteur, 35, 29071 Málaga, España.

To cite this article: Serrano-Mesa, J., Moncada-Ramirez, J., Ojeda, P., Ambrosio-Cestero, G., Ruiz-Sarmiento, J.R., Gonzalez-Jimenez, J., 2026. Open-Vocabulary Semantic Mapping via 3D Gaussian Splatting. Jornadas de Automática, 47. https://doi.org/10.17979/ja-cea.2026.47.13803

Resumen

Los mapas semánticos de vocabulario abierto son clave para la autonomía de los robots móviles en entornos complejos no estructurados. Comparados con las técnicas tradicionales, los mapas semánticos de vocabulario abierto representan una mejora, ya que no están limitados por un conjunto predefinido de categorías de objetos. Para su construcción, las representaciones geométricas basadas en 3D Gaussian Splatting con descriptores semánticos de bajo nivel han ganado popularidad. Sin embargo, la operativa robótica exige etiquetas de alto nivel, como descripciones textuales y categorías. En este trabajo presentamos una evolución de OpenSplat3D que integra Modelos de Visión-Lenguaje a Gran Escala (LVLMs) para la descripción y categorización de instancias de objetos, cuya coherencia es validada en el espacio latente de Modelos de Visión-Lenguaje (VLMs). Adicionalmente, optimizamos la reconstrucción geométrica mediante una inicialización densa y el uso de una función de pérdida basada en la profundidad. Ambas propuestas se validan empleando los conjuntos de datos Replica y ScanNet, obteniendo mapas precisos enriquecidos semánticamente.

Palabras clave: Robots móviles, Percepción y sensado, Construcción de mapas, Robótica inteligente, Aprendizaje automático

Open-Vocabulary Semantic Mapping via 3D Gaussian Splatting

Abstract

Open-vocabulary semantic maps are key to the autonomy of mobile robots in complex, unstructured environments. Compared to traditional techniques, open-vocabulary semantic maps represent a significant improvement, as they are not limited by a predefined set of categories. For their construction, geometric representations based on 3D Gaussian Splatting enriched with low-level semantic descriptors have gained popularity. However, robotic operations demand high-level labels, such as textual descriptions and categories. This work presents an evolution of OpenSplat3D that integrates Large Vision-Language Models (LVLMs) for the description and categorization of object instances, whose consistency is validated within the latent space of Vision-Language Models (VLMs). Additionally, the geometric reconstruction has been optimized through dense initialization and the use of a depth-based loss function. Both proposals are validated using the Replica and ScanNet datasets, yielding accurate, semantically enriched maps.

Keywords: Autonomous Mobile Robots, Perception and sensing, Map building, Intelligent robotics, Machine Learning

1. Introducción

Actualmente, existe un interés creciente por integrar robots móviles en entornos compartidos con humanos, tales como hogares, centros logísticos, entornos sanitarios o de restauración, para asistir de manera segura y eficiente en las actividades que se desarrollan (Rubio et al., 2019). En estos escenarios, surge la necesidad de que el robot posea una representación del entorno que no sea exclusivamente geométrica,

*Autor para correspondencia: sm15jorge@uma.es

sino que le permita interpretar y razonar sobre el mismo para completar con éxito sus tareas. La estrategia predominante para alcanzar estas capacidades es la utilización de *mapas semánticos* (Ruiz-Sarmiento et al., 2017), modelos del entorno de trabajo donde se representan tanto los elementos físicos (obstáculos, objetos, etc.) como su semántica (propiedades, funcionalidades, relaciones, etc.). Por ejemplo, en una planta industrial, un operario podría ordenar al robot realizar un inventario de las herramientas disponibles. Empleando el mapa semántico, el robot sería capaz de representar información asociada a las herramientas como sus propiedades (tipo, estado operativo, fabricante, etc.), sus funcionalidades (útil para embalaje, sujeción, manipulación, etc.) y sus relaciones con otros elementos del entorno, como las estaciones de trabajo donde se suelen usar. Esta información le permitiría generar inventarios de manera autónoma y notificar la presencia de herramientas fuera de lugar, faltantes, en mal estado, etc.

Tradicionalmente, la construcción de estos mapas ha consistido en vincular información semántica a primitivas geométricas básicas, tales como nubes de puntos, cajas contenedoras (*bounding boxes*), mallas o vóxeles (Matez-Bandera et al., 2024). No obstante, estas presentan algunas limitaciones que restringen la capacidad del robot para interactuar con el entorno con la robustez y precisión necesarias. Por un lado, las nubes de puntos carecen de continuidad topológica, lo que dificulta el razonamiento sobre los límites precisos de los objetos. Por otro lado, las cajas contenedoras y los vóxeles tienden a una simplificación excesiva del entorno mediante formas ortogonales con una cierta resolución, resultando en una pérdida de fidelidad en zonas con geometrías irregulares.

Recientemente, han surgido métodos basados en *3D Gaussian Splatting* (3DGS) (Kerbl et al., 2023) para mitigar esta falta de detalle geométrico. El 3DGS representa el entorno mediante una densa nube de elipsoides 3D (Gaussianas) con propiedades de color, opacidad, posición, rotación y escala. A diferencia de las representaciones discretas tradicionales, el 3DGS permite una reconstrucción diferenciable que, mediante un proceso iterativo de optimización de una función de pérdida, alcanza una representación geométrica altamente fiel a la escena real. Dada esta capacidad para capturar geometrías complejas con precisión, el 3DGS se postula como una representación ideal para anclar información semántica, abriendo la puerta a la creación de mapas semánticos mucho más detallados. En este contexto, han aparecido propuestas que integran información semántica de bajo nivel en 3DGS como OpenGaussian (Wu et al., 2024), InstanceGaussian (Li et al., 2025) y ObjectGS (Zhu et al., 2025).

Entre los métodos del estado del arte basados en esta tecnología destaca OpenSplat3D (Piekenbrinck et al., 2025), el cual extiende el 3DGS añadiendo segmentación a nivel de instancias de objetos. Este método introduce vectores de características de bajo nivel (referidos en este trabajo como descriptores visuales o *embeddings*) asociados a cada instancia provenientes de Modelos de Visión-Lenguaje (VLMs) (p. ej., CLIP (Radford et al., 2021) o MasQCLIP (Xu et al., 2023)). Sin embargo, OpenSplat3D, en su estado actual, presenta algunas limitaciones a la hora de aprovechar las representaciones resultantes en aplicaciones robóticas. La limitación más crítica radica en que las instancias segmentadas carecen de etiquetas de alto nivel en lenguaje natural (descripciones textuales y categorías) que las identifiquen (p. ej., microondas, sofá, etc.) lo que dificulta procesos como la planificación de tareas o la Interacción Humano-Robot (HRI). Además, el método emplea *Structure from Motion* (SfM) para inicializar las posiciones de las Gaussianas, resultando en un mapa con una notable falta de detalles geométricos complejos. Finalmente, la optimización geométrica depende únicamente del color de las imágenes, provocando la aparición de Gaussianas flotantes (*floaters*) que no corresponden con ninguna superficie real del entorno.

Este artículo presenta un método para la creación de mapas semánticos que, partiendo de la base de OpenSplat3D, aborda las limitaciones mencionadas anteriormente para generar representaciones geométricas precisas donde las instancias de objetos están enriquecidas con etiquetas de alto nivel. En primer lugar, se ha extendido el flujo de trabajo original incorporando una etapa de generación de hipótesis sobre estas etiquetas mediante Modelos de Visión-Lenguaje de Gran Escala (LVLM), incluyendo descripciones textuales y categorías correspondientes a cada una de las instancias (p.ej., `A white upright appliance with vertical handles for cooling food : Refrigerator`) y permitiendo una identificación de objetos mediante un *vocabulario abierto* de categorías. Para la selección de la hipótesis más representativa por instancia y el filtrado de posibles resultados erróneos, se comprueba la similitud entre el descriptor visual de la instancia y los descriptores textuales de las hipótesis en el espacio latente de algún VLM, garantizando la coherencia entre la descripción textual y la información visual codificada en el mapa. En segundo lugar, se ha incorporado una fase de preprocesamiento en la que se genera una nube de puntos densa utilizando información de profundidad de la escena (procedente de una cámara RGB-D o un modelo de estimación de profundidad), lo que proporciona una inicialización de las Gaussianas altamente fiel a la escena real. Finalmente, se ha modificado la función de optimización original para integrar una pérdida de profundidad, mejorando la precisión geométrica en la reconstrucción de la escena.

Para validar la propuesta se han realizado pruebas con los conjuntos de datos Replica (Straub et al., 2019) y ScanNet (Dai et al., 2017), construyéndose mapas donde cada instancia detectada está caracterizada por una descripción y una categoría que es útil para su aprovechamiento por parte de un robot. La implementación del método con las mejoras propuestas se encuentra disponible públicamente en `https://github.com/jrgserrano/semantic-gaussians`.

2. Descripción del método para mapeo semántico

El método propuesto, basado en OpenSplat3D (Piekenbrinck et al., 2025), toma como entrada un conjunto de imágenes RGB-D junto con sus correspondientes poses de cámara y devuelve como salida un mapa semántico del entorno basado en 3DGS. Este mapa está formado por un conjunto de elipsoides 3D (Gaussianas), típicamente decenas de miles, agrupados en instancias de objetos. Cada instancia de objeto, además, viene ampliada con una etiqueta de alto nivel que incluye una descripción textual y su categoría.

El flujo de trabajo se divide en una serie de etapas secuenciales (ver Fig. 1). Inicialmente, se generan nubes de puntos

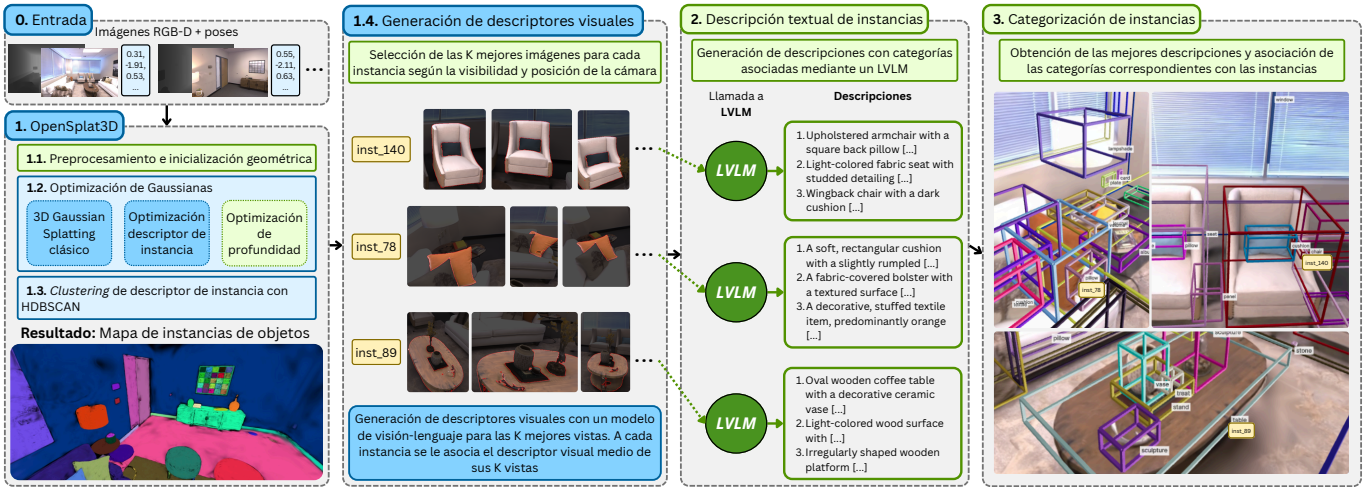

Figura 1: Visión general del método propuesto basado en OpenSplat3D, ilustrado sobre la escena `room0` del conjunto de datos Replica. Las etapas resaltadas en verde han sido modificadas y/o añadidas para incluir las propuestas presentadas con respecto al trabajo original de OpenSplat3D.

a partir de las imágenes RGB-D sobre las cuales se inicializan las Gaussianas (Sección 2.1). Posteriormente, se realiza un proceso de optimización en el que se refinan las propiedades de las Gaussianas (posición, escala, rotación, opacidad y color) y se aplican estrategias de densificación y poda (*adaptive control*) con el fin de obtener una representación geométricamente fiel a las imagenes captadas (Sección 2.2). De forma paralela a la optimización geométrica, mediante el uso de máscaras de segmentación 2D y una pérdida contrastiva, se optimizan descriptores de instancia de cada Gaussiana, lo que permite que las pertenecientes a una misma instancia de objeto en la escena presenten descriptores similares (Sección 2.3). Una vez finalizada la optimización, se aplica un algoritmo de agrupamiento (*clustering*) a dichos descriptores de instancia para segmentar las instancias de objetos. A partir de estas, se seleccionan las imágenes más informativas en las que aparece cada instancia bajo criterios de mayor visibilidad y diversidad geométrica, y se extraen descriptores visuales mediante un VLM que son promediados a nivel de instancia. De este modo, cada instancia se asocia con un descriptor visual de bajo nivel (Sección 2.4). Finalmente, las imágenes anteriores se procesan mediante un LVLM para generar hipótesis sobre las etiquetas (descripciones y categorías de los objetos), que son validadas posteriormente comparando el descriptor visual con descriptores textuales obtenidos aplicando el VLM a cada etiqueta (ver Sección 2.5).

2.1. *Preprocesamiento e inicialización geométrica*

A diferencia de OpenSplat3D, que emplea nubes de puntos dispersas provenientes de SfM, el método propuesto introduce una reconstrucción densa inicial que utiliza la profundidad de imágenes RGB-D para posicionar desde el comienzo las Gaussianas sobre superficies reales. Para cada imagen RGB-D i, el proceso emplea el modelo de cámara Pinhole para construir su nube de puntos asociada, P_i^C. Esta nube de puntos se expresa en el sistema de referencia del mapa empleando la pose de la cámara correspondiente a dicha imagen y el operador composición: $P_i^M = C_i^M \oplus P_i^C$, consolidando los centroides iniciales para el proceso de 3DGS. Para reducir la demanda de recursos computacionales, en la inicialización se aplica un submuestreo aleatorio limitado por un presupuesto máximo de puntos que garantiza una cobertura global eficiente. De esta forma, la optimización se centra en el refinado de la apariencia y la semántica, incrementando la precisión geométrica y reduciendo significativamente el tiempo de convergencia.

2.2. *Fundamentos: 3D Gaussian Splatting*

El núcleo del método utiliza 3DGS para representar la escena mediante un conjunto de N elipsoides 3D (Gaussianas truncadas) definidos por su posición (μ), covarianza (Σ), color mediante armónicos esféricos (c) y opacidad (o). El renderizado de la escena se realiza mediante el proceso de *Splatting*, que consiste en transformar las primitivas 3D en una representación 2D final a través de una secuencia de operaciones. En primer lugar, se proyectan los elipsoides 3D (Gaussianas) sobre el plano de la imagen para obtener distribuciones 2D denominadas *splats*. En segundo lugar, estos *splats* se ordenan por profundidad respecto al punto de vista de la cámara y se combinan mediante *alpha blending*, obteniendo la contribución α_n de cada Gaussiana por píxel p que permite calcular su intensidad final en la imagen renderizada $I(p)$.

El modelo se optimiza minimizando la pérdida de reconstrucción fotométrica $\mathcal{L}_{\text{RGB}}$ entre las imágenes observadas y las renderizadas, que combina el error absoluto medio ($\mathcal{L}_{l_1}$) y el índice de similitud estructural ($\mathcal{L}_{\text{SSIM}}$) (Kerbl et al., 2023):

$$\mathcal{L}_{\text{RGB}} = (1-\beta)\mathcal{L}_{l_1} + \beta\mathcal{L}_{\text{SSIM}} \tag{1}$$

Esta base permite reconstruir la apariencia visual y la geométrica de la escena con alta precisión. Las secciones siguientes detallan cómo se enriquece el mapa incorporando información semántica a distintos niveles.

2.3. *Aprendizaje de instancias*

Para codificar la información de instancia de objeto, OpenSplat3D extiende cada Gaussiana con un descriptor de instancia $\mathbf{f}_n \in \mathbb{R}^d$. Estos se inicializan aleatoriamente y se renderizan mediante *Splatting* para generar un mapa de características $F(p)$. Utilizando máscaras extraídas por un modelo de segmentación de imágenes (p.ej., Segment Anything Model (Kirillov et al., 2023)), para cada máscara $\mathcal{M}_i$ se calcula

su prototipo de instancia $\mathbf{z}_i$ mediante el promedio de los descriptores de instancia proyectados en ella.

Este prototipo se utiliza para definir una *pérdida contrastiva* compuesta por dos términos. Por un lado, la componente positiva ($\mathcal{L}_{\text{pos}}$) busca minimizar la varianza interna de la instancia, atrayendo todos los descriptores de instancia dentro de la misma máscara hacia su prototipo. Por otro, la componente negativa ($\mathcal{L}_{\text{neg}}$) que fomenta la separabilidad entre instancias mediante una pérdida basada en un margen de distancia γ, forzando a que los prototipos de distintas instancias se alejen entre sí. La pérdida contrastiva final a nivel de instancia se define como la combinación ponderada de ambos términos:

$$\mathcal{L}_{\text{inst2D}} = w_p \cdot \mathcal{L}_{\text{pos}} + w_n \cdot \mathcal{L}_{\text{neg}} \tag{2}$$

donde w_p y w_n controlan el equilibrio entre la compacidad de la instancia y su separabilidad.

Para evitar que los descriptores de instancia diverjan debido al *alpha blending* durante el renderizado, se introduce una pérdida de regularización de varianza:

$$\mathcal{L}_{\text{var}} = \text{Var}(F(p)) = \left(\sum_{n=1}^{N} \mathbf{f}_n^2 \alpha_n \prod_{j=1}^{n-1} (1 - \alpha_j) \right) - F(p)^2 \tag{3}$$

Además, como propuesta de este trabajo para mejorar la consistencia geométrica del mapa, se introduce una pérdida de profundidad $\mathcal{L}_{\text{depth}}$ que penaliza la discrepancia entre el mapa de profundidad renderizado mediante *Splatting*, $\hat{D}(p)$, y la profundidad real proporcionada por el sensor RGB-D, $D(p)$, en cada píxel válido p:

$$\mathcal{L}_{\text{depth}} = \frac{1}{|P|} \sum_{p \in P} \|\hat{D}(p) - D(p)\|_1 \tag{4}$$

La función de pérdida de la optimización final combina la reconstrucción fotométrica (Eq. (1)), la pérdida contrastiva (Eq. (2)), la regularización de varianza (Eq. (3)) y la pérdida de profundidad (Eq. (4).

$$\mathcal{L} = \mathcal{L}_{\text{RGB}} + \lambda_{\text{inst2d}} \mathcal{L}_{\text{inst2d}} + \lambda_{\text{var}} \mathcal{L}_{\text{var}} + \lambda_{\text{depth}} \mathcal{L}_{\text{depth}}$$

Tras la optimización, se aplica el algoritmo de agrupamiento HDBSCAN sobre los descriptores resultantes, lo que permite recuperar un número arbitrario de instancias de forma discreta, gestionando el ruido y la incertidumbre inherentes a la extensión de las instancia y la representación Gaussiana.

2.4. Caracterización semántica de bajo nivel

Para dotar a las instancias de información semántica de bajo nivel, el método propuesto asocia un único *embedding* visual a cada instancia discretizada previamente mediante HDBSCAN. Este descriptor se extrae a partir de un proceso de selección de las imágenes más informativas en las que aparece el objeto, garantizando que el VLM reciba la máxima cantidad de atributos visuales desde diferentes puntos de vista.

Para ello, se renderizan máscaras binarias de cada instancia desde las diferentes perspectivas, de tal modo que las Gaussianas que pertenezcan a una misma instancia se renderizan de color blanco, mientras que el resto lo hacen en negro. A continuación, se aplica un umbral para la obtención de una máscara binaria y, finalmente, para la selección de los puntos de vista con más información de la instancia de objeto i, se calcula un valor de visibilidad $s_{v,i}$ por punto de vista v con un método similar al empleado en OpenMask3D (Takmaz et al., 2023):

$$s_{v,i} = \frac{|M_{v,i}|}{|I_v|} \cdot \frac{|G_{v,i}|}{|G_i|}$$

donde $|M_{v,i}|$ es el número de píxeles de la instancia, $|I_v|$ el tamaño de la imagen, $|G_i|$ el total de Gaussianas de la instancia y $|G_{v,i}|$ las Gaussianas visibles desde el punto de vista v.

Para mejorar la etapa de caracterización, se ha implementado un algoritmo de selección basado en diversidad geométrica en el que tras filtrar un conjunto de puntos de vista candidatos v_{pool} con los mayores índices de visibilidad, se seleccionan iterativamente las K vistas que maximizan la disparidad angular. Siendo $\mathbf{d}_v$ el vector unitario de dirección de la vista v y $\mathcal{S}$ el conjunto de vistas seleccionadas hasta el momento, la siguiente vista v^* se elige según el siguiente criterio sobre la distancia coseno:

$$v^* = \arg\max_{v \in v_{\text{pool}} \setminus \mathcal{S}} \left(\min_{s \in \mathcal{S}} \text{d}_{\cos}(\mathbf{d}_v, \mathbf{d}_s) \right)$$

donde $\text{d}_{\cos}(\mathbf{d}_v, \mathbf{d}_s) = 1 - \frac{\mathbf{d}_v \cdot \mathbf{d}_s}{\|\mathbf{d}_v\| \|\mathbf{d}_s\|}$. Este enfoque garantiza una cobertura multicontextual de la instancia mediante la observación desde perspectivas complementarias. En consecuencia, se mitigan errores por oclusiones parciales o condiciones de iluminación desfavorables en ángulos específicos.

Para optimizar la entrada a los VLMs, se ha implementado un proceso de extracción de recortes orientado al objeto. Inicialmente, se recuperan las máscaras binarias $M_{k,i}$ de una instancia i para las K mejores vistas. Posteriormente, se calcula un factor de expansión dinámico según las dimensiones de la instancia en $M_{k,i}$ que permite recortar la imagen de la vista k centrada en la instancia i. Este recorte proporciona el contexto necesario para la comprensión del entorno sin comprometer los detalles de las instancias. Finalmente, las K vistas recortadas de cada instancia se introducen en el codificador visual de un VLM para obtener K descriptores visuales, los cuales son promediados para obtener el descriptor visual definitivo.

2.5. Descripción textual y categorización de instancias

Una vez consolidado el descriptor visual de cada instancia, se inicia la etapa de caracterización semántica de alto nivel. El proceso comienza proporcionando las K mejores vistas recortadas a un LVLM. Estas vistas son procesadas delimitando el contorno del objeto mediante una línea de color rojo y aplicando un filtro de atenuación sobre el fondo. Este ajuste visual, junto con un *prompt* diseñado específicamente para la tarea, permite al modelo centrar su atención en la instancia objetivo y generar un conjunto de hipótesis descriptivas candidatas $H = \{h_1, h_2, \ldots, h_m\}$, donde cada hipótesis incluye una descripción textual y una categoría para la identificación de la instancia correspondiente empleando un vocabulario abierto.

Para garantizar la coherencia semántica de la descripción textual asignada a cada instancia, se emplea el descriptor visual ($\mathbf{v}_i$) como referencia de validación. Entre el conjunto de hipótesis generadas por el LVLM, la hipótesis final h^* se selecciona como aquella cuyo descriptor textual $\mathbf{t}_j$ presente mayor similitud coseno con $\mathbf{v}_i$:

$$h^* = \arg\max_{h_j \in H} \left(\frac{\mathbf{t}_j \cdot \mathbf{v}_i}{||\mathbf{t}_j|| ||\mathbf{v}_i||} \right)$$

donde $\mathbf{t}_j$ se obtiene codificando cada descripción generada por el LVLM mediante el mismo modelo empleado en la extracción de descriptores visuales (Sección 2.4). Este criterio de selección asegura que las categorías ancladas al mapa de Gaussianas sean las que presentan la mayor afinidad semántica con la información visual 3D almacenada, permitiendo una interpretación del entorno de vocabulario abierto aprovechable para tareas robóticas de alto nivel.

3. Experimentos

Con el objetivo de validar el método de mapeo semántico presentado en este trabajo, se han realizado una serie de experimentos empleando escenas extraídas de dos conjuntos de datos: Replica (Straub et al., 2019) y ScanNet (Dai et al., 2017). El primero proporciona entornos sintéticos de alta fidelidad, permitiendo validar el método en situaciones ideales. El otro, ScanNet, ofrece capturas del mundo real, lo que permite evaluar la robustez del método ante el ruido sensorial y la complejidad geométrica. A continuación, se detallan los aspectos de implementación (Sección 3.1) y se analizan los resultados obtenidos (Secciones 3.2–3.3).

3.1. Detalles de implementación

El despliegue del método descrito se ha realizado mediante la arquitectura CSAR (Ambrosio-Cestero et al., 2026) , basada en la tecnología de contenedores LXC (*Linux Containers*). Este esquema modular permite externalizar las etapas de alta carga computacional, como la optimización de las Gaussianas y la inferencia de modelos de gran escala, hacia nodos de computación en el borde (*edge computing*), evitando así comprometer los limitados recursos de procesamiento locales de una plataforma robótica móvil. Esta arquitectura está desplegada en un equipo de alto rendimiento en el que destacan las tres tarjetas NVIDIA RTX6000 ADA con 48GB de memoria VRAM GDDR6.

Respecto a los modelos integrados en el flujo de trabajo, para la fase de segmentación de instancias en las imágenes 2D (Sección 2.3) se ha empleado SAM2 (*Segment Anything Model 2*) (Ravi et al., 2024), aprovechando su avanzada capacidad de obtención de máscaras. Para la generación de los descriptores visuales de las instancias (Sección 2.4) se ha utilizado MasQCLIP (Xu et al., 2023), un modelo que permite extraer descriptores alineados con el lenguaje natural que mantienen una estrecha correspondencia con la geometría de la máscara, facilitando la validación de coherencia semántica descrita en la Sección 2.5. En este caso, se ha utilizado $K = 3$, es decir, para cada instancia se han seleccionado las 3 imágenes más informativas para generar los descriptores semánticos de las instancias. Finalmente, para el razonamiento semántico y la generación de categorías de vocabulario abierto (Sección 2.5) se ha seleccionado Gemma 4 31B[1] como LVLM. La elección de este modelo se fundamenta en su elevada capacidad de comprensión contextual, permitiendo obtener descripciones detalladas y categorías precisas a partir de las perspectivas de los objetos seleccionadas. De nuevo, se ha optado por que el modelo genere 3 descripciones y categorías distintas.

3.2. Validación de la coherencia semántica con Replica

Para evaluar la precisión de la categorización, se realizaron pruebas exhaustivas en 4 escenas de Replica. La aplicación del método propuesto comienza con la optimización de las Gaussianas y la extracción de instancias (recuérdese la Fig. 1). Una vez discretizadas las instancias, se seleccionan las perspectivas más informativas siguiendo el criterio de disparidad angular y valor de visibilidad descrito en la Sección 2.4. Estas imágenes se procesan a través de Gemma 4 31B, el cual genera tres hipótesis con distintas descripciones y categorías para cada instancia. Como se observa en la Fig. 2, para una instancia específica (un cojín en la escena), el modelo propone diversas interpretaciones basadas en su contexto visual.

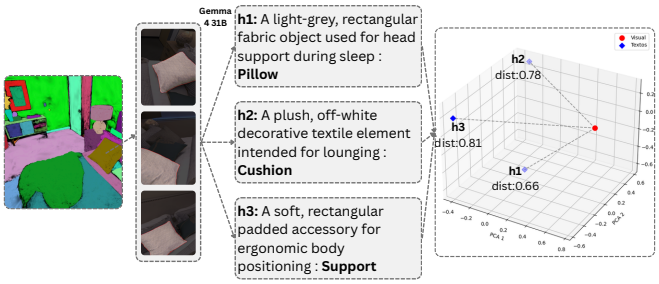

Figura 2: Ejemplo de caracterización de alto nivel. A la izquierda, mapa de instancias obtenido de la escena `room1` de Replica. Al centro, selección de las mejores perspectivas de una instancia y generación de hipótesis por Gemma 4 31B. A la derecha, representación mediante proyección PCA de descriptores MasQCLIP donde se aprecian las distancias entre el *embedding* visual (punto rojo) y los de las descripciones textuales (diamantes azules). La menor distancia (mayor similitud) confirma la descripción más coherente.

La selección de la categoría definitiva se realiza mediante el cálculo de la distancia en el espacio latente de MasQCLIP. Como se muestra en la proyección tridimensional mediante Análisis de Componentes Principales (PCA) de la Fig. 2, suele existir una correlación directa entre la coherencia lógica de la descripción y su proximidad al descriptor visual de la instancia. En este experimento, h_1 ("Pillow") y h_2 ("Cushion") presentan las distancias más reducidas respecto al descriptor visual, siendo la categoría "Cushion" la que mejor describe al objeto en el contexto de la escena en que se encuentra. Por el contrario, categorías más genéricas, como h_3 ("Support"), quedan más lejanas en el espacio latente.

Este resultado confirma que el uso del codificador de un modelo VLM como validador es un mecanismo eficaz para filtrar las alucinaciones potenciales del LVLM o hipótesis menos precisas, asegurando que la descripción y categoría almacenadas en el mapa semántico final sean lo más fieles posible a la apariencia visual del objeto.

3.3. Evaluación en entornos reales con ScanNet

A continuación, se procedió a evaluar la robustez del método propuesto en el conjunto de datos ScanNet, el cual presenta desafíos significativos derivados de capturas con sensores reales, incluyendo desenfoque por movimiento y variaciones bruscas de iluminación. Como se observa en la Fig. 3, el método logra segmentar instancias coherentes a pesar del ruido inherente a la reconstrucción. Los resultados cualitativos

[1] `https://huggingface.co/google/gemma-4-31b`

indican que el método mantiene una precisión considerable en la identificación de objetos incluso ante oclusiones parciales.

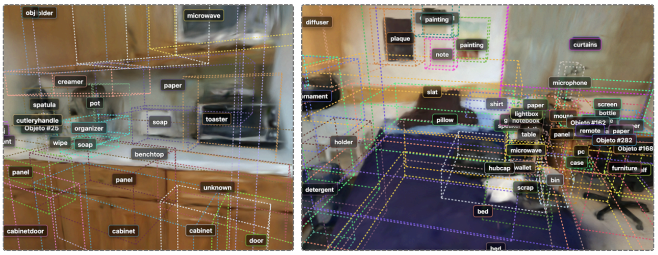

Figura 3: Resultados de reconstrucción y categorización en la secuencia `scene0000_01` de ScanNet. Mejor ver con zoom 250 %.

Como evidencia de la estructura de datos generada, el método produce un mapa semántico en formato JSON que organiza la información geométrica y lingüística de cada objeto. El siguiente fragmento muestra la salida para una instancia de la secuencia `scene0000_01` de ScanNet:

```
"doormat_289": {
    "bbox": {
        "center": [6.92, 5.18, 0.10],
        "size": [0.64, 0.91,0.30]
    },
    "n_observations": 816,
    "results": {
        "caption": "A rectangular, light-colored fabric
             mat typically used at a doorway",
        "similarity": 0.28248
    }
}
```

Este nivel de detalle permite que el mapa resultante no solo sirva para visualización, sino que sea directamente utilizable para tareas de alto nivel como la planificación o la HRI.

Figura 4: Comparación geométrica en la secuencia `scene0000_01` de ScanNet. A la izquierda, imagen de la escena con inicialización aleatoria. A la derecha, imagen de la escena con las mejoras geométricas propuestas.

Finalmente, se ha evaluado el resultado de las mejoras geométricas propuestas (Sección 2.3) en una secuencia de ScanNet. La Fig. 4 desmuestra una reducción de *floaters* no deseables y una mejora en la precisión geométrica. Además, se ha realizado una comparación cuantitativa usando la métrica PSNR, obteniendo 28,3 dB con nuestro método y 26,3 dB con el original, lo que supone una mejora notable.

4. Conclusiones

Este trabajo presenta una evolución del método OpenSplat3D para la creación de mapas semánticos de vocabulario abierto mediante 3D Gaussian Splatting. En concreto, se ha extendido el flujo original incorporando una etapa de generación de hipótesis sobre las descripciones textuales y las categorías de las instancias de objetos empleando un LVLM, y un mecanismo de validación de dichas hipótesis. Además, partiendo de la segmentación en instancias ya consolidada, se han introducido mejoras en la precisión geométrica mediante una inicialización basada en la generación de nubes de puntos y la integración de la pérdida de profundidad. Los experimentos realizados demuestran cualitativamente la efectividad de la estrategia. Sin embargo, el método presenta una serie de limitaciones. Por un lado, la segmentación de SAM en entornos reales pueden dificultar la categorización posterior por parte del LVLM. Por otro lado, el tiempo de ejecución del método (3 horas de media en las pruebas realizadas) limita su aplicación actual a escenarios *offline*. Como trabajo futuro se plantea una evaluación cuantitativa de la propuesta y su aprovechamiento en entornos reales por parte de un robot.

Agradecimientos

Este trabajo ha sido desarrollado en el contexto de los proyectos MINDMAPS (PID2023-148191NB-I00) y Voxeland (PPRO-B1-2023-017, JA.B1-09), financiados por el Ministerio de Ciencia e Innovación y la Universidad de Málaga, respectivamente, así como del Plan Andaluz de Investigación, Desarrollo e Innovación (PPRO-TEP960-G-2023) y del programa de becas FPU (FPU24/02974).

Referencias

Ambrosio-Cestero, Gregorio, Andrades, G., Cipriano, Gonzalez-Jimenez, Javier, Ruiz-Sarmiento, Jose-Raul, 2026. Csar: Containerized system architecture for robotics.

Dai, A., Chang, A. X., Savva, M., Halber, M., Funkhouser, T., Nießner, M., 2017. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In: Proc. CVPR.

Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G., July 2023. 3d gaussian splatting for real-time radiance field rendering. ACM Trans. Graph. 42 (4).

Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., et al., 2023. Segment anything. arXiv:2304.02643.

Li, H., Wu, Y., Meng, J., Gao, Q., Zhang, Z., Wang, R., Zhang, J., 2025. Instancegaussian: Appearance-semantic joint gaussian representation for 3d instance-level perception. In: Proc. CVPR.

Matez-Bandera, J.-L., Ojeda, P., Monroy, J., Gonzalez-Jimenez, J., Ruiz-Sarmiento, J.-R., 2024. Voxeland: Probabilistic instance-aware semantic mapping with evidence-based uncertainty quantification. arXiv:2411.08727.

Piekenbrinck, J., Schmidt, C., Hermans, A., Vaskevicius, N., Linder, T., Leibe, B., 2025. OpenSplat3D: Open-Vocabulary 3D Instance Segmentation using Gaussian Splatting. In: Proc. CVPR. pp. 5246–5255.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., et al., 2021. Learning transferable visual models from natural language supervision. In: Proc. ICML. PmLR, pp. 8748–8763.

Ravi, N., Gabeur, V., Hu, Y.-T., Hu, R., Ryali, C., Ma, T., et al., 2024. Sam 2: Segment anything in images and videos. arXiv:2408.00714.

Rubio, F., Valero, F., Llopis-Albert, C., 2019. A review of mobile robots: Concepts, methods, theoretical framework, and applications. Int. J. Adv. Robot. Syst. 16.

Ruiz-Sarmiento, J.-R., Galindo, C., Gonzalez-Jimenez, J., 2017. Building multiversal semantic maps for mobile robot operation. Knowl. Based Syst. 119, 257–272.
DOI: `https://doi.org/10.1016/j.knosys.2016.12.016`

Straub, J., Whelan, T., Ma, L., Chen, Y., Wijmans, E., Green, S., et al., 2019. The Replica dataset: A digital replica of indoor spaces. arXiv:1906.05797.

Takmaz, A., Fedele, E., Sumner, R. W., Pollefeys, M., Tombari, F., Engelmann, F., 2023. OpenMask3D: Open-Vocabulary 3D Instance Segmentation. In: Proc. NeurIPS.

Wu, Y., Meng, J., Li, H., Wu, C., Shi, Y., Cheng, X., et al., 2024. Opengaussian: Towards point-level 3d gaussian-based open vocabulary understanding. In: Proc. NeurIPS. pp. 19114–19138.

Xu, X., Xiong, T., Ding, Z., Tu, Z., October 2023. Masqclip for open-vocabulary universal image segmentation. In: Proc. ICCV. pp. 887–898.

Zhu, R., Yu, M., Xu, L., Jiang, L., Li, Y., Zhang, T., et al., 2025. Objectgs: Object-aware scene reconstruction and scene understanding via gaussian splatting. In: Proc. ICCV.