



UNIVERSIDAD
DE MÁLAGA



Universidad de Málaga

ESCUELA DE INGENIERÍAS INDUSTRIALES

DEPARTAMENTO DE INGENIERÍA DE SISTEMAS Y AUTOMÁTICA

Trabajo Fin de Grado

GENERACIÓN DE MODELOS 3D DE NAVEGACIÓN PARA ROBOTS DE EXPLORACIÓN

GRADO EN INGENIERÍA EN TECNOLOGÍAS INDUSTRIALES

ITINERARIO DE AUTOMÁTICA

Autor: Alfonso Martínez Petersen

Tutor: Carlos Pérez del Pulgar Mancebo

Cotutor: Levin Gerdes

26 de mayo de 2025

Resumen

Generación de modelos 3D de navegación para robots de exploración

Autor: Alfonso Martínez Petersen

Tutor: Carlos Pérez del Pulgar

Cotutor: Levin Gerdes

Departamento: Departamento de Ingeniería de Sistemas y Automática

Titulación: Grado de Ingeniería en Tecnologías Industriales

Palabras clave: Reconstrucción tridimensional, Gaussian Splatting, Neural Radiance Fields (NeRF), Fotogrametría, SLAM, Modelado 3D, COLMAP, Nerfstudio, Robótica espacial, Rover

En los últimos años, los sistemas robotizados han adquirido un papel cada vez más relevante en las misiones de exploración espacial. La creciente complejidad de los objetivos, junto con el desarrollo de nuevas tecnologías en inteligencia artificial, percepción y computación embarcada, ha consolidado a los robots como herramientas clave para un amplio abanico de operaciones, tanto planetarias como orbitales. Factores como el retardo en las comunicaciones y la ausencia de sistemas de posicionamiento global hacen imprescindible mejorar su autonomía e integrar información sensorial más rica que les permita situarse con precisión y tomar decisiones informadas en tiempo real. En este contexto, cobran especial interés los últimos desarrollos en reconstrucción tridimensional, que destacan por la riqueza de la información visual y espacial que aportan a los robots para el desempeño de sus tareas. En especial, tecnologías como los Campos de Radiancia Neuronal (*NeRF*) y los métodos de reconstrucción mediante Salpicaduras Gaussianas (*Gaussian Splatting*) suponen un avance significativo por su capacidad de generar modelos 3D de alta calidad a partir de información visual limitada, gracias a sus novedosos enfoques de representación, mediante redes neuronales en el caso del primero y mediante la optimización directa de primitivas gaussianas en el segundo.

Este Trabajo Fin de Grado investiga métodos avanzados de reconstrucción tridimensional y localización visual aplicados a un rover espacial operando en entornos semejantes a los encontrados en misiones espaciales. Se propone un pipeline automatizado que, a partir de datos registrados por el propio rover, permite generar reconstrucciones precisas mediante distintas variantes de NeRF y Gaussian Splatting, con el objetivo de evaluar sus capacidades y viabilidad para tareas de exploración planetaria.

El proceso se estructura en tres fases: extracción y adaptación de los datos almacenados por el rover; reconstrucción geométrica guiada mediante un proceso de *Structure-from-Motion*; y generación de modelos tridimensionales mediante entrenamiento o ajuste de sus respectivas representaciones. Además, se abordan aspectos clave como la automatización del flujo de trabajo, la integración en un entorno unificado de todas las herramientas (ROS 2, COLMAP, Nerfstudio) y su contenedorización completa con Docker.

Este trabajo aspira a establecer un punto de partida sólido para el uso de técnicas de reconstrucción 3D en futuras aplicaciones de robótica espacial, facilitando su incorporación como estándar en entornos donde los métodos clásicos resultan insuficientes.

Abstract

3D model generation for exploration robots

Author: Alfonso Martínez Petersen

Supervisor: Carlos Pérez del Pulgar

Cosupervisor: Levin Gerdes

Departament: Departamento de Ingeniería de Sistemas y Automática

Degree: Grado de Ingeniería en Tecnologías Industriales

Keywords: 3D Reconstruction, Gaussian Splatting, Neural Radiance Fields (NeRF), Photogrammetry, SLAM, 3D Modeling, COLMAP, Nerfstudio, Space Robotics, Rover

In recent years, robotic systems have taken on an increasingly important role in space exploration missions. The growing complexity of mission objectives, combined with the development of new technologies in artificial intelligence, perception, and embedded computing, has consolidated robots as key tools for a wide range of operations, both planetary and orbital. Factors such as communication delays and the absence of global positioning systems make it essential to enhance their autonomy and integrate richer sensory information that allows them to localize precisely and make informed decisions in real time. In this context, recent advances in 3D reconstruction have gained particular relevance due to the richness of visual and spatial information they provide for robotic tasks. In particular, technologies such as Neural Radiance Fields (NeRF) and reconstruction methods based on Gaussian Splatting represent a significant leap forward in their ability to generate high-quality 3D models from limited visual data, thanks to their novel representational approaches: neural networks in the former case, and direct optimization of Gaussian primitives in the latter.

This Bachelor's Thesis investigates advanced methods for 3D reconstruction and visual localization applied to a space rover operating in environments similar to those found in real missions. It proposes an automated pipeline that, using data recorded by the rover itself, enables the generation of accurate reconstructions using different variants of NeRF and Gaussian Splatting, with the aim of evaluating their capabilities and feasibility for planetary exploration tasks.

The process is structured into three main phases: extraction and adaptation of the data stored by the rover; geometry reconstruction guided by a Structure-from-Motion approach; and generation of 3D models through training or adjustment of their respective representations. In addition, the project addresses key aspects such as workflow automation, integration of heterogeneous tools (ROS 2, COLMAP, Nerfstudio) within a unified environment, and full containerization using Docker.

This work aims to establish a solid starting point for the use of 3D reconstruction techniques in future space robotics applications, supporting their adoption as a standard in environments where classical methods prove insufficient.

A mi abuelo José Carlos.

Agradecimientos

Quisiera expresar mi más sincero agradecimiento a todas las personas que me han acompañado y apoyado durante la realización de este Trabajo de Fin de Grado.

En primer lugar, agradezco profundamente a mi tutor, Carlos Pérez del Pulgar, por su guía, disponibilidad permanente y por haber contado conmigo desde el principio en el Laboratorio de Robótica Espacial. Me has descubierto un mundo que no conocía y que, sin duda, ha marcado un antes y un después en mi camino académico. Igualmente, quiero agradecer a mi cotutor, Levin Gerdes, por su ayuda inestimable y su paciencia conmigo.

A mi familia, pilar fundamental de mi vida, que a pesar de todo siempre ha creído en mí. Sin vuestro apoyo, nada de esto habría sido posible.

Finalmente, a mis amigos, gracias por acompañarme en todas las etapas de este proceso sin ser conscientes.

A todos vosotros, gracias de corazón.

Índice

1.	Introducción	1
1.1.	Contexto y motivación del trabajo	1
1.2.	Problema abordado	4
1.3.	Objetivos del trabajo	5
1.4.	Estructura del documento	5
2.	Estado del Arte	7
2.1.	Sistemas tradicionales de localización para robots sin GNSS	7
2.2.	Structure from Motion (SfM):	9
2.3.	Métodos basados en campos de radiancia: Nerf y Gaussian Splatting	12
2.4.	Comparativa entre MRFs y métodos tradicionales de localización y mapeo	21
3.	Metodología	26
3.1.	Herramientas y tecnologías utilizadas	26
3.2.	Proceso de experimentación con NeRF y Gaussian Splatting	28
3.3.	Diseño del software para extracción y manipulación de datos de los <i>rosbags</i>	32
3.4.	Reconstrucciones a partir de datos de los <i>rosbags</i>	33
3.5.	Valoración de la viabilidad de la realización de reconstrucciones sin SfM previo: programación de software adaptador de formato	34
3.6.	Mejoras en el proceso de Structure from Motion	35
3.7.	Optimización del flujo de trabajo	37
4.	Resultados	42
4.1.	Reconstrucciones 3D realizadas	42
4.2.	Análisis de tiempo y precisión en las reconstrucciones	52
5.	Discusión	55
5.1.	Evaluación de los resultados obtenidos	55
5.2.	Impacto potencial	59
5.3.	Propuestas para investigaciones futuras	60
6.	Conclusión	61

7.	Anexos	62
7.1.	Configuración técnica de las herramientas utilizadas	62

Índice de Figuras

1.	<i>rover</i> Perseverance. Fuente: [1].	1
2.	Módulo de aterrizaje Chang'e-4 en la cara oculta de la Luna, captado por el <i>rover</i> Yutu-2. Fuente: [2].	2
3.	Ariane 6 durante pruebas de infraestructura en el puerto espacial europeo, 2024. Fuente: [3].	2
4.	Sistema ORB-SLAM2 sobre imágenes en el campus de la Universidad Austral. Crédito: Alejandro Silvestri – CC BY-SA 4.0. Fuente: [4].	8
5.	Captura de modelo 3D LiDAR del Pelourinho (Salvador de Bahía), generada por un sistema SLAM. Crédito: CyArk – CC BY-SA 3.0. Fuente: [5].	9
6.	Esquema del método Structure from Motion (SfM). Partiendo de imágenes desde distintas perspectivas de un mismo objeto 3D, a partir de la posición sobre los planos imagen de determinadas características se estiman la localización de las cámaras y la estructura del cuerpo 3D. Adaptado de [6].	11
7.	Comparativa conceptual entre γ y Gaussian Splatting. En el enfoque NeRF (a), se muestrean puntos a lo largo de un rayo y se evalúan mediante una red neuronal (MLP) para obtener color y densidad. Este tipo de representación se considera un modelo implícito, ya que la geometría y la apariencia de la escena no están directamente almacenadas como estructuras espaciales discretas, sino codificadas en los parámetros de la red neuronal (F_θ). En cambio, el enfoque Gaussian Splatting (b) emplea una nube de gaussianas tridimensionales que almacenan de forma explícita su posición, orientación, tamaño, opacidad y color, contribuyendo directamente a la imagen final mediante una etapa de fusión diferenciable. Este tipo de representación se denomina explícita porque la información tridimensional está directamente disponible y manipulable sin necesidad de ser inferida por una red. Imagen adaptada de [7].	13
8.	Proceso de entrenamiento y renderizado de vistas en NeRF. Se muestran, respectivamente y en cuatro etapas: el muestreo espacial rayo por rayo, a partir del cual se obtienen los datos de posición de los puntos y la dirección del rayo desde la cámara; la predicción de color y densidad mediante perceptrones multicapa (redes neuronales); la integración del color por rayo mediante renderizado volumétrico para sintetizar imágenes; y, finalmente, la comparación de las vistas predichas con las imágenes originales, y el ajuste del modelo mediante una función de pérdida. Adaptado de [8].	15
9.	Reconstrucción tridimensional mediante técnicas de fotogrametría basadas en Campos de Radiancia Neuronal (<i>Neural Radiance Fields</i> , NeRF) de la Torre di Caprona en Pisa, Italia. Imagen adaptada de [9].	17

10.	Modelo Gaussian Splatting para evaluación de daños de un vehículo vandalizado. ¹	19
11.	Modelos de 3D Gaussian Splatting aplicados a la supervisión de obras de construcción. Esta técnica permite una reconstrucción eficiente y visualmente fidedigna de escenas complejas en entornos industriales. Imagen adaptada de [10].	19
12.	Ejemplo de síntesis de vistas con distintas apariencias en el mismo modelo 3D de la Fontana di Trevi, generado con Splatfacto-W. Ambas imágenes provienen de la misma reconstrucción geométrica, pero utilizan diferentes vectores latentes de apariencia para reflejar condiciones de iluminación contrastantes (día y noche). Esta capacidad de desacoplar geometría y apariencia permite generar vistas realistas en contextos variados a partir de un único conjunto de imágenes. Imagen tomada de [11].	27
13.	Vistas renderizadas mediante el método Nerfacto del <i>dataset poster</i> , el más sencillo e inmediato de los que ofrece Nerfstudio. Puede descargarse directamente desde la línea de comandos de dicho entorno con el comando <code>ns-download-data nerfstudio</code> . ²	29
14.	Vistas diversas circundantes a un objeto sencillo observado en tiempo real en el visor de Nerfstudio apenas segundos después de iniciar el entrenamiento del modelo Nerfacto y sin que este hubiera terminado. Como se ve en (a), durante el entrenamiento se puede observar la pose de las fotografías de partida en la GUI de Nerfstudio.	30
15.	Vistas de una curva cóncava a lo largo de la esquina de un espacio de interior durante el entrenamiento del modelo <i>Splatfacto-W</i> . Son visibles las poses de las fotografías utilizadas en el proceso, así como la baja calidad del resultado, consecuencia de una cobertura y solapamiento insuficientes. Esta etapa de la experimentación permitió concluir que el proceso de toma de imágenes tiene igual o mayor importancia que el propio algoritmo de reconstrucción para la generación de modelos mediante reconstrucción fotogramétrica mediante RFM.	31
16.	Esquema general del <i>pipeline</i> automatizado desarrollado para la reconstrucción 3D y localización visual a partir de datos del <i>rover</i> . Se han agrupado las tareas en tres subprocesos definidos por el tipo de datos con el que se trabaja y la tecnología de la que se depende mayoritariamente en cada fase. El flujo de trabajo automatizado completo se puede ejecutar directamente con el archivo de terminal <code>run_full_pipeline.sh</code>	38
17.	Comparativa entre el recorrido real y el estimado por COLMAP para la trayectoria 1. La reconstrucción se corresponde con el tramo inicial, mostrado en el extremo izquierdo del mapa completo. Se aprecian algunas inexactitudes en el cálculo de la pose en el extremo final de la trayectoria.	43
18.	Vista general del modelo generado de la trayectoria 1. Se puede ver cómo las inexactitudes en la reconstrucción dispersa del extremo final de la trayectoria afectan negativamente al resultado en dicha zona	43
19.	Comparaciones entre imágenes reales (izq.) y vistas renderizadas (dcha.) del modelo reconstruido, todas con orientación coincidente, seleccionadas para la evaluación de la trayectoria 1.	44

20.	Comparativa entre el recorrido real y el estimado por COLMAP para la trayectoria 2. La reconstrucción se corresponde con el tramo inicial, mostrado en el extremo izquierdo del mapa completo.	45
21.	Vista general del modelo generado de la trayectoria 2.	45
22.	Comparaciones entre imágenes reales (izq.) y vistas renderizadas (dcha.) del modelo reconstruido, todas con orientación coincidente, seleccionadas para la evaluación de la trayectoria 2.	46
23.	Comparativa entre el recorrido real y el estimado por COLMAP para la trayectoria 3. El extremo inferior e izquierdo del recorrido representado se corresponde con el inicio del itinerario del robot.	47
24.	Vista general del modelo generado de la trayectoria 3	47
25.	Comparaciones entre imágenes reales (izq.) y vistas renderizadas (dcha.) del modelo reconstruido, todas con orientación coincidente, seleccionadas para la evaluación de la trayectoria 3.	48
26.	Comparativa entre el recorrido real y el estimado por COLMAP para la trayectoria 4. El extremo izquierdo de las imágenes se corresponde con el tramo inicial del recorrido realizado por el robot.	49
27.	Vista general del modelo generado de la trayectoria 4	49
28.	Comparaciones entre imágenes reales (izq.) y vistas renderizadas (dcha.) del modelo reconstruido, todas con orientación coincidente, seleccionadas para la evaluación de la trayectoria 4.	50
29.	Comparativa entre el recorrido real y el estimado por COLMAP para la trayectoria 5.	51
30.	Vista general del modelo generado de la trayectoria 5.	51
31.	Comparaciones entre imágenes reales (izq.) y vistas renderizadas (dcha.) del modelo reconstruido, todas con orientación coincidente, seleccionadas para la evaluación de la trayectoria 5.	52

Índice de Tablas

1.	Comparación entre Visual SLAM, LiDAR SLAM y Radar SLAM	22
2.	Evaluación de Structure from Motion	23
3.	Comparación entre NeRF original, NeRF in the Wild y Nerfacto	24
4.	Comparación entre 3D Gaussian Splatting, Splatfacto y Splatfacto in the Wild	25
5.	Información general de las rutas de captura y muestreo de imágenes.	42
6.	Información temporal asociada a las rutas de captura.	42
7.	Resultados del método de preprocesamiento integrado de Nerfstudio.	53
8.	Resultados de preprocesamiento: emparejado secuencial.	53
9.	Resultados de preprocesamiento con método propio completo: emparejado secuencial e implementación de previos de localización de las cámaras.	54
10.	Evaluación del proceso de Structure from Motion para cada trayectoria. Va- lores promedio.	54
11.	Evaluación de la reconstrucción con Splatfacto in the Wild para cada trayec- toria. Se muestran valores promedio y desviación estándar.	54

1. Introducción

1.1. Contexto y motivación del trabajo

En los últimos años, la exploración espacial ha experimentado un impulso sin ningún precedente conocido desde la carrera espacial entre EEUU y la Unión Soviética a mediados del siglo XX. Pese a las grandes diferencias en la situación del sector espacial actual respecto al de aquel momento, en cierta manera diversos estados, agencias espaciales internacionales y empresas privadas compiten también hoy por alcanzar ventajas significativas que les aseguren una posición en sectores llamados a jugar un papel clave en la economía mundial durante las próximas décadas, tales como las telecomunicaciones, la defensa, y la futura explotación económica de recursos extraplanetarios.

Numerosos actores han logrado avances significativos en esta nueva era : pese a que Estados Unidos ha alcanzado importantes hitos como el *rover* Perseverance (Fig. 1) y pretende consolidar su liderazgo con programas como , cuyo propósito es traer de vuelta a la humanidad a la Luna tras más de medio siglo, su ambición no se detiene ahí: en la próxima década se esperan misiones como Gateway, una estación espacial orbital que pretende funcionar en el futuro como plataforma para misiones tripuladas a Marte y el espacio profundo; y *Mars Sample Return*, cuya meta es traer de vuelta a la Tierra muestras de la superficie marciana por primera vez en la historia.



Figura 1: *rover* Perseverance. Fuente: [1].

Tampoco China se queda atrás: tras convertirse en la primera administración en aterrizar en la cara oculta de la Luna con la misión Chang'e 4 (Fig. 2) y en traer muestras de dicho lugar con Chang'e 6, además de enviar el único *rover* no estadounidense, Zhurong, a la superficie marciana; el país asiático avanza en su objetivo de consolidarse como líder en materia espacial. En la actualidad, su proyecto más ambicioso es la *International Lunar Research Station* (ILRS), una base lunar permanente que pretende estar operativa para 2035 y que, de concretarse, marcaría un hito en la exploración y colonización del espacio profundo.

En paralelo, la Agencia Espacial Europea (*European Space Agency*, ESA) ocupa también un lugar destacado en este entorno. Además de tener un lugar destacado en el desarrollo de tecnologías espaciales y colaborar con las agencias internacionales en misiones como las ya citadas Artemis y Gateway, Europa ha redoblado sus esfuerzos para fortalecer su autonomía estratégica, con importantes proyectos esencialmente europeos como el desarrollo del cohete Ariane 6 (Fig 3) o la misión ExoMars, que planea enviar a Marte un rover diseñado para perforar hasta 2 metros bajo la superficie en busca de indicios de vida en el planeta rojo, así como importantes desarrollos técnicos como el módulo lunar Argonaut, o el sistema de navegación satelital Galileo.

Junto a estos actores, agencias espaciales como las de Rusia, Japón, India o Emiratos



Figura 2: Módulo de aterrizaje Chang'e-4 en la cara oculta de la Luna, captado por el *rover* Yutu-2. Fuente: [2].



Figura 3: Ariane 6 durante pruebas de infraestructura en el puerto espacial europeo, 2024. Fuente: [3].

Árabes Unidos han dado también un nuevo impulso a sus programas espaciales, dinamizando el ecosistema con ambiciosas misiones para extender el conocimiento científico y asegurar su independencia estratégica en materia espacial. En este contexto, la robótica espacial se está situando como una de las claves para el éxito de la mayoría de las misiones de exploración proyectadas debido a su carácter esencialmente imprescindible en las tareas de exploración, toma de muestras y mantenimiento de infraestructuras en órbita; así como sus enormes ventajas en materia logística, económica y de seguridad. Además, poseen muchas menos limitaciones que los astronautas humanos, pudiendo ser sometidos a las condiciones más extremas. Por estas razones, es un hecho ampliamente aceptado que los robots ejercerán de avanzada necesaria en todas las misiones proyectadas para el asentamiento permanente de seres humanos en la luna y Marte. No obstante, para lograrlo aún es necesario resolver

importantes limitaciones técnicas en relación a la autonomía, la resistencia frente a todo tipo de condiciones adversas, el consumo energético y los recursos computacionales que estos aparatos requieren para su funcionamiento.

Retos en la autonomía de robots espaciales

De entre los retos mencionados, uno de los desafíos más relevantes a los que se enfrentan ingenieros y equipos de investigación en robótica espacial es la necesidad de desarrollar sistemas capaces de actuar de manera autónoma debido al retardo en las telecomunicaciones entre la Tierra y los lugares en que se desarrollan las misiones espaciales, que varía entre unos pocos segundos y decenas de minutos en función de la distancia. Este hecho hace inviable la teleoperación simultánea, especialmente para tareas de exploración planetaria. Por ello, es necesario que los *rovers* tengan la capacidad de situarse de forma precisa en entornos desconocidos, identificar obstáculos, planificar rutas y desempeñar las funciones para las que han sido diseñados sin recibir instrucciones externas.

Para conseguir este fin, una de las limitaciones más significativas es la ausencia de sistema global de navegación por satélite (*Global Navigation Satellite System*, GNSS) en el espacio, dado que los satélites en que se basan estos sistemas orbitan la superficie terrestre y no están disponibles fuera de ella. Por tanto, para alcanzar el nivel requerido de autonomía, los *rovers* necesitan otros sistemas que les permitan conocer de forma precisa su posición en tiempo real. Actualmente, la solución predominante en este ámbito son los sistemas de localización y mapeo simultáneo (*Simultaneous Localization and Mapping*, SLAM), pero su aplicación en el espacio está limitada por las restricciones de capacidad computacional y el ruido que presentan las mediciones de los sensores.

En este sentido, una alternativa de extraordinario interés son las técnicas de reconstrucción tridimensional del entorno. Además de ser útiles para la resolución del problema de navegación en escenarios con escasez de texturas y rasgos distintivos frente a las técnicas basadas en nubes de puntos y a los enfoques geométricos tradicionales, permiten al robot obtener una información mucho más extensa sobre las características de su entorno, lo que puede suponer el punto de partida para el desarrollo de un amplio abanico de mejoras en términos de autonomía y capacidad para la toma de decisiones.

UMA Space Robotics Lab. BASEPROD

En respuesta a estos retos, el Laboratorio de Robótica Espacial de la Universidad de Málaga (*UMA Space Robotics Lab*, SRL) investiga y experimenta con técnicas orientadas a la optimización de la autonomía en exploración planetaria, con foco en el uso de algoritmos de inteligencia artificial para aplicaciones de localización y mapeo. Estos algoritmos buscan mejorar la capacidad de los robots para identificar obstáculos, planificar la trayectoria más adecuada en función de ellos y situarse en todo momento de forma precisa en entornos desconocidos.

Entre el 20 y el 23 de junio de 2023, el equipo del SRL llevó a cabo una misión de toma de datos en el paraje semidesértico de las Bárdenas Reales, en Navarra. Durante la campaña, se realizaron numerosas pruebas con el *Rover* Marciano para Pruebas de Autonomía (*Martian Rover for Testing Autonomy*, MaRTA) de la ESA, un vehículo terrestre diseñado para simular misiones de exploración planetaria al que se equipó con toda clase de sensores para comprobar la eficacia y utilidad de distintas formas de información sensorial para su uso en aplicaciones de navegación autónoma.

Fruto de esta campaña experimental, en 2024 se publicó *BASEPROD: The Bardenas Semi-Desert Planetary Rover Dataset* [12], un artículo científico que hizo públicos los datos recopilados durante las travesías del *rover*. BASEPROD incluye datos registrados de aproximadamente 1,7 km de trayectorias recorridas, que abarcan imágenes RGB, imágenes térmicas y de profundidad, mediciones de sensores inerciales, giroscopios, sensores de par en cada una de las ruedas y localización GNSS en tiempo real. La publicación de este conjunto de datos tiene como objetivo facilitar el desarrollo de técnicas avanzadas de guiado, navegación y control (*Guidance, Navigation and Control*, GNC) aplicables a futuros *rovers* espaciales.

1.2. Problema abordado

En la última década, la necesidad de sistemas con una autonomía cada vez mayor en materia de robótica espacial ha impulsado el desarrollo de sistemas de navegación que permitan a los *rovers* explorar entornos desconocidos sin intervención humana. Para ello, el mapeo de superficies en 3D puede resultar una herramienta de enorme utilidad para la exploración de entornos complejos, dotando al *rover* de contexto y de información mucho más detallada sobre su entorno que la que puede extraerse de los sistemas tradicionalmente utilizados para resolver el problema de localización. Esta necesidad se ve acentuada en escenarios en los que los sensores convencionales ofrecen información limitada o ambigua y no hay acceso a información satelital sobre la posición. En este contexto, una reconstrucción tridimensional precisa del entorno puede resultar una herramienta de gran ayuda para la localización, la detección de obstáculos y la planificación de trayectorias.

Sin embargo, la generación de modelos 3D de alta precisión depende significativamente de la cantidad y la calidad de los datos de partida, especialmente en entornos no estructurados y en condiciones cambiantes de iluminación, textura o relieve, donde a menudo es difícil obtener una representación suficientemente precisa de manera eficiente. En respuesta a este reto, en los últimos años han surgido nuevas técnicas basadas en inteligencia artificial que permiten generar reconstrucciones visuales detalladas a partir de conjuntos de datos visuales limitados, entre las que destacan los Campos de Radiancia Neuronal (*Neural Radiance Fields*, NeRF) y los métodos de representación mediante Gaussian Splatting. Ambas aproximaciones suponen un cambio de paradigma respecto a las técnicas clásicas de reconstrucción geométrica, permitiendo obtener modelos continuos y realistas a partir de información discreta y parcial. No obstante, su aplicación en contextos reales de robótica móvil aún presenta desafíos importantes en términos de integración, rendimiento computacional y automatización.

Por ello, este Trabajo Fin de Grado explorará las posibilidades que ofrecen dichos métodos en esta materia mediante la generación de reconstrucciones de un paraje natural semidesértico semejante a un entorno espacial. Asimismo, se ha desarrollado un software capaz de extraer la información directamente de los *rosbags* del *rover* y realizar de manera automática todas las operaciones necesarias para generar dichas reconstrucciones, proponiendo mejoras en el flujo de trabajo habitual de dos de las herramientas de código abierto más populares de la actualidad para reconstrucción de modelos 3D con inteligencia artificial, con el fin de adaptarlas a las necesidades de nuestro proyecto, reducir en la medida de lo posible el tiempo de procesado y aumentar la calidad de la reconstrucción.

1.3. Objetivos del trabajo

- Estudiar la viabilidad y aplicaciones de NeRF y Gaussian Splatting en el campo de la robótica espacial.
- Generar reconstrucciones 3D a partir de imágenes de conjuntos de datos de distintas procedencias, especialmente de las tomadas por el *rover* MARTA en el paraje semidesértico de las Bárdenas Reales de Navarra.
- Diseñar un algoritmo unificado capaz de extraer las imágenes y los datos de los sensores de los *rosbags* de *Robot Operating System 2* (ROS 2), exportarlos a un formato legible por herramientas de reconstrucción 3D y llamar al procesamiento y generación automáticos del modelo 3D.
- Reducir el tiempo total de generación de las reconstrucciones en un 50 % con respecto a los obtenidos con la configuración de serie.
- Comparar las prestaciones de las reconstrucciones mediante NeRF y Gaussian Splatting para distintos tipos de conjuntos de datos.
- Proponer mejoras para hacer el proceso viable en misiones reales.

1.4. Estructura del documento

El presente trabajo se organiza en seis capítulos principales que guían al lector desde la contextualización del problema hasta la validación de los resultados obtenidos, pasando por un análisis detallado del estado del arte y un desarrollo experimental basado en herramientas de reconstrucción 3D avanzadas.

El documento comienza con esta introducción, en la que se describe el contexto tecnológico y científico de la investigación, se motiva la necesidad de explorar nuevos métodos de localización para robots en entornos donde no es posible utilizar sistemas GNSS, y se delimitan los objetivos concretos que se persiguen. También se ofrece esta breve presentación de la estructura del trabajo para facilitar su comprensión global (ver Sección 1).

A continuación, se presenta una revisión de los enfoques más relevantes relacionados con la localización y reconstrucción tridimensional en robótica móvil (ver Sección 2). Esta revisión incluye desde sistemas tradicionales de localización basados en sensores inerciales o visuales, hasta métodos avanzados como Estructura desde el Movimiento (*Structure from Motion*, SfM) y modelos neuronales para la representación de escenas. Finalmente se presentan y analizan en profundidad los campos de radiancia como NeRF y Gaussian Splatting, que constituyen la base metodológica del presente trabajo.

El capítulo metodológico (Sección 3) constituye el núcleo técnico del documento. En él se detalla el proceso experimental seguido, comenzando con una etapa de evaluación preliminar de NeRF y Gaussian Splatting sobre conjuntos de datos controlados. Posteriormente, se describe el diseño y desarrollo de un software específico para la extracción de datos directamente desde los *rosbags* del *rover* y su conversión a un formato interpretable por las tecnologías en las que se basa este proyecto; así como la integración de estos datos en un pipeline automatizado de reconstrucción cuyas especificaciones y proceso de diseño también se detallan. A lo largo de esta sección, el lector podrá seguir el desarrollo cronológico del trabajo: se identifican las principales dificultades técnicas, se analizan sus causas, y se explican las soluciones implementadas para superarlas. Entre ellas destacan la mejora del proceso de mapeo mediante la implementación de estimaciones iniciales de la pose, la integración optimizada

de COLMAP y la estandarización del flujo mediante la creación de programas de terminal que automatizan el proceso y la dockerización del proyecto para poder implementarlo en cualquier ordenador sin problemas de compatibilidad mediante un contenedor.

El apartado de resultados (Sección 4) recoge los productos visuales y cuantitativos obtenidos a lo largo del proceso experimental. Se incluyen reconstrucciones tridimensionales generadas desde datos reales del *rover*, así como métricas de calidad, tiempo de procesamiento y robustez. Esta sección ofrece una validación objetiva del enfoque adoptado, comparando distintas configuraciones y condiciones de entrada.

En el capítulo de discusión (Sección 5) se realiza un análisis crítico de los resultados obtenidos, examinando tanto las ventajas como las limitaciones del método propuesto en comparación con enfoques tradicionales. Se estudia su viabilidad para su uso en entornos reales como la exploración espacial, así como su posible escalabilidad a otras plataformas robóticas.

Finalmente, la sección de conclusiones (Sección 6) resume las principales aportaciones del trabajo, reflexiona sobre las limitaciones observadas durante el desarrollo y plantea futuras líneas de investigación que podrían continuar esta línea de trabajo. A esto se suma un conjunto de anexos técnicos (Sección 7) que documentan en profundidad la configuración de las herramientas utilizadas y facilitan la replicación de los experimentos.

2. Estado del Arte

2.1. Sistemas tradicionales de localización para robots sin GNSS

En el contexto de la investigación en robótica espacial, el desarrollo de sistemas de localización autónoma en tiempo real supone una necesidad evidente y un reto de primer orden. La ausencia de los satélites que hacen posible la señal GNSS, la distancia con la tierra y el retardo que esta provoca en las comunicaciones y las características del terreno hacen necesario el desarrollo de sistemas alternativos para establecer la posición y trazar rutas de manera rápida, segura y precisa. Con este fin existen numerosas soluciones de dilatada trayectoria y contrastada efectividad, a las que en los últimos años se han ido sumando herramientas basadas en redes neuronales y sistemas de visión artificial. Conviene detenerse a repasar las más importantes:

Sistemas de medición inercial (IMU)

Este método se basa en la medición de la orientación, velocidad, aceleración y fuerzas actuantes sobre el robot, obteniendo la información posicional mediante la posterior integración numérica de dichos datos. Tiene la ventaja de ser un método inmediato y económico, pero generalmente la precisión es insuficiente y se va degradando progresivamente por la acumulación de errores no corregidos. Adicionalmente, en el ámbito de la robótica de exploración planetaria, la irregularidad del terreno de estudio resulta en vibraciones y otras interferencias que amplifican este efecto, por lo que usualmente no es viable su uso como referencia única de la posición y es necesaria su combinación con otros métodos.

Simultaneous localization and mapping (SLAM)

Se conoce como SLAM a un conjunto de técnicas que, por diversos medios, construyen en tiempo real el mapa de un entorno y se sitúan sobre él. Aunque según el método utilizado se presentan variaciones, como norma general el proceso se basa en la toma cíclica de datos (imágenes, información tridimensional...), la selección y almacenamiento de características clave capaces de describir el entorno, el estudio de la variación de dichas características en términos absolutos y el cruce de dichos datos con el mapa ya generado a partir de ellas en iteraciones anteriores, optimizándolo de forma constante. Algunos de las variantes más extendidas [13] son:

SLAM visual: Para este método se toman imágenes sucesivas, bien monoculares, o bien imágenes RGB-D con información de profundidad. Se detectan puntos clave representativos del terreno, como esquinas, bordes o cambios de textura, y se estima la trayectoria de la cámara mediante el análisis de las variaciones de su posición en el plano 2D de las imágenes. El sistema más extendido es ORB-SLAM(Fig 4), que optimiza la extracción y emparejamiento de características mediante el uso de descriptores orb, generando un mapa 3D de forma paralela al seguimiento y realizando una corrección iterativa mediante cierre de bucles. [14]

Pese a su buen rendimiento en entornos estructurados con condiciones lumínicas adecuadas, el SLAM Visual presenta algunas limitaciones a tener en cuenta en otros supuestos. Este método presenta una dependencia significativa de la cantidad y calidad de los puntos clave detectados, por lo que su eficacia se ve considerablemente reducida en entornos homogéneos como desiertos o superficies arenosas en las que es difícil tomar suficientes referencias



Figura 4: Sistema ORB-SLAM2 sobre imágenes en el campus de la Universidad Austral. Crédito: Alejandro Silvestri – CC BY-SA 4.0. Fuente: [4].

visuales fiables. Además, también es sensible a cambios bruscos en la iluminación tales como los producidos por sombras, reflejos y perturbaciones similares, que pueden dificultar la detección y emparejamiento de puntos clave entre imágenes sucesivas, reduciendo la precisión general del método. Otras restricciones a tener en cuenta son el efecto de los objetos dinámicos, que generan puntos clave efímeros que aumentan la probabilidad de errores; y la sensibilidad a las condiciones ópticas de la cámara y la calidad de las imágenes tomadas. Alteraciones como el desenfoque de imágenes por el movimiento de la cámara, el tiempo de exposición mal ajustado o una resolución insuficiente también pueden afectar al correcto funcionamiento del método, por lo que es frecuente complementarlo con los datos de otros sensores como IMU o LiDAR, que aportan información adicional y mitigan el efecto de las limitaciones descritas.

A pesar de estas dificultades, el SLAM visual sigue siendo una herramienta fundamental en la robótica espacial y en otras aplicaciones donde el peso, el consumo energético y las limitaciones computacionales impiden el uso de sensores más avanzados. La posibilidad de generar mapas detallados con cámaras ligeras y de bajo coste, junto con el potencial de integración de técnicas avanzadas basadas en inteligencia artificial para mejorar la detección de puntos clave en condiciones adversas, sigue impulsando su desarrollo y su aplicación en proyectos de exploración planetaria, navegación autónoma y vehículos no tripulados.

Lidar SLAM: Los sensores LiDAR (Light Detection and Ranging) son dispositivos capaces de estimar la distancia hacia puntos y superficies emitiendo haces láser y midiendo el tiempo que tardan en regresar al ser reflejados. Esto permite generar nubes de puntos con información tridimensional del entorno.

El método LIDAR SLAM aprovecha dichos sensores para generar mapas 3D en tiempo real, que se actualizan periódicamente y son comparados con los generados en instantes anteriores para calcular el desplazamiento relativo. Este método es significativamente más preciso que las alternativas visuales, permitiendo hacer mejores reconstrucciones de entornos cuyas texturas no facilitan la identificación visual de puntos clave. No obstante, se ve afectado por superficies reflectantes que alteran el comportamiento de los haces de luz láser y el coste computacional se multiplica al trabajar con información tridimensional en tiempo real. (Fig 5)

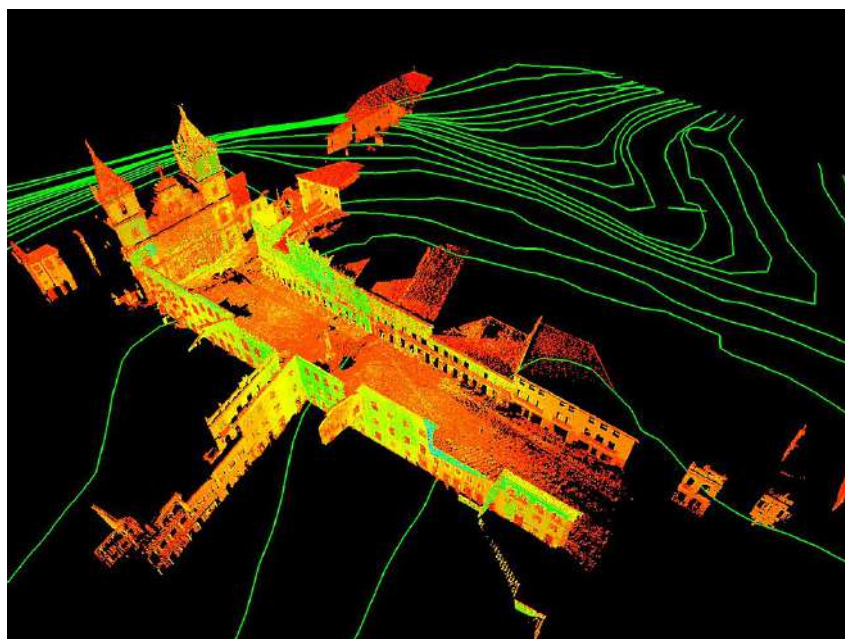


Figura 5: Captura de modelo 3D LiDAR del Pelourinho (Salvador de Bahía), generada por un sistema SLAM. Crédito: CyArk – CC BY-SA 3.0. Fuente: [5].

Radar-SLAM: Otro enfoque que ha despertado un gran interés recientemente es el empleo de señales de radar para aplicaciones de localización y mapeo simultáneo, en especial en entornos que presentan dificultades para el empleo de otros métodos.

Un radar es un dispositivo capaz de emitir una señal de radio y recibirla tras ser reflejada por los objetos de su entorno, pudiendo extraer información de este último a través del estudio de dicha señal, sus características y el tiempo de retorno. La principal ventaja de este método es su robustez frente a condiciones habituales en medios exteriores tales como precipitaciones, polvo en suspensión, destellos; además de ser superior en la detección de características en entornos poco estructurados. Asimismo, su consumo de energía y su precio son significativamente inferiores a los de otros sensores como los LiDAR. Sin embargo, ofrece una resolución inferior a otros métodos, y su precisión se ve considerablemente disminuida con la aparición de ruido o reflexiones indeseadas. Para superar estas limitaciones pueden utilizarse técnicas basadas en el filtrado de señales y algoritmos de aprendizaje automático, que mejoran la precisión del sistema y lo hacen especialmente útil para su uso en entornos adversos.

2.2. Structure from Motion (SfM):

Se denomina Estructura desde el Movimiento (también conocido por la terminología inglesa Structure from Motion), a la automatización de un conjunto de técnicas procedentes del campo de la fotogrametría destinadas a la reconstrucción dispersa (se obtienen puntos discretos) de objetos y entornos a partir de fotografías individuales tomadas desde distintas perspectivas o ángulos. Se fundamenta en la detección de características y la búsqueda de correlaciones entre las mismas en imágenes consecutivas para reconstruir la estructura 3D y estimar de manera simultánea la posición de las cámaras [15].

Suponiendo que se parte de una cámara ideal sin distorsiones (modelo pinhole), es posible relacionar un punto 3D

$$\mathbf{X} = (X, Y, Z, 1)^T$$

con su proyección

$$\mathbf{x} = (u, v, 1)^T$$

en el plano de la imagen mediante la ecuación:

$$\mathbf{x} = \mathbf{K} [\mathbf{R} \mid \mathbf{t}] \mathbf{X},$$

donde:

- $\mathbf{K}$ es la matriz intrínseca que incluye parámetros como las distancias focales f_x y f_y y el centro óptico (c_x, c_y) ,
- $[\mathbf{R} \mid \mathbf{t}]$ representa la matriz de pose extrínseca, con $\mathbf{R}$ siendo la matriz de rotación y $\mathbf{t}$ el vector de traslación.

El siguiente paso consiste en detectar características visuales (por ejemplo, mediante algoritmos como SIFT u ORB) en cada imagen y establecer correspondencias entre imágenes. Con estas correspondencias se pueden derivar las relaciones geométricas entre imágenes, expresadas mediante la geometría epipolar. En este contexto, la relación entre las proyecciones $\mathbf{x}_1$ y $\mathbf{x}_2$ de un mismo punto 3D en dos imágenes se describe mediante la ecuación:

$$\mathbf{x}_2^T \mathbf{F} \mathbf{x}_1 = 0,$$

donde $\mathbf{F}$ es la matriz fundamental. En cámaras calibradas, esta relación se puede expresar de forma más sencilla a través de la matriz esencial:

$$\mathbf{E} = [\mathbf{t}]_{\times} \mathbf{R},$$

que vincula directamente la rotación y traslación entre dos cámaras.

Una vez estimadas las poses relativas de las cámaras, se procede a la triangulación, que consiste en recuperar las coordenadas 3D de los puntos observados minimizando el error entre sus proyecciones y las imágenes reales (Fig 6). Este proceso se formula típicamente como:

$$\min_{\mathbf{X}} \sum_{i=1}^N \|\mathbf{x}_i - \pi(\mathbf{P}_i \mathbf{X})\|^2,$$

donde $\pi(\cdot)$ representa la función de proyección y $\mathbf{P}_i = \mathbf{K}_i [\mathbf{R}_i \mid \mathbf{t}_i]$ es la matriz de proyección de la i -ésima cámara.

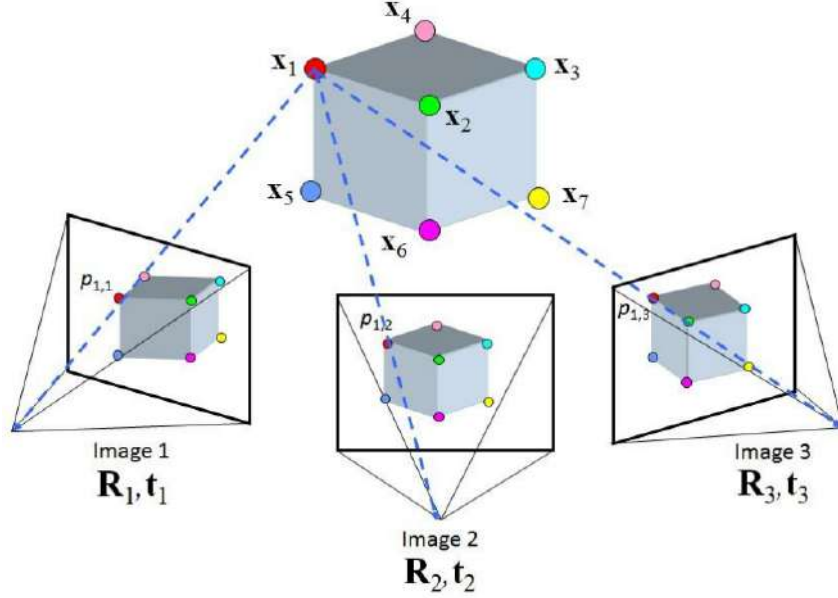


Figura 6: Esquema del método Structure from Motion (SfM). Partiendo de imágenes desde distintas perspectivas de un mismo objeto 3D, a partir de la posición sobre los planos imagen de determinadas características se estiman la localización de las cámaras y la estructura del cuerpo 3D. Adaptado de [6].

Finalmente, mediante el ajuste de haz o *Bundle Adjustment* (BA) se optimizan simultáneamente las poses de las cámaras y las coordenadas de los puntos 3D para minimizar el error total de reproyección. Matemáticamente, el problema se plantea de la siguiente manera:

$$\min_{\{\mathbf{X}_j, \mathbf{R}_i, \mathbf{t}_i\}} \sum_{i=1}^M \sum_{j=1}^N \|\mathbf{x}_{ij} - \pi(\mathbf{K}_i [\mathbf{R}_i | \mathbf{t}_i] \mathbf{X}_j)\|^2,$$

donde:

- $\mathbf{x}_{ij}$ es la proyección observada del punto $\mathbf{X}_j$ en la imagen de la cámara i ,
- M es el número de cámaras y N el número de puntos 3D.

A diferencia de SLAM, SfM no requiere que las imágenes sean tomadas secuencialmente siguiendo una trayectoria definida, sino que permite el uso de imágenes completamente aleatorias y distantes espacial y cronológicamente, siempre y cuando se refieran al mismo objeto de reconstrucción. Por contra, no es un proceso indicado para aplicaciones en tiempo real, y generalmente se utiliza para procesamiento offline de las imágenes de manera posterior a su captura. A pesar de esto, en este trabajo, se ha utilizado SfM para obtener la pose de las cámaras, así como información estructural auxiliar del entorno para realizar las reconstrucciones de manera más precisa. En la discusión (sección 5) se hace referencia a los inconvenientes que conlleva este hecho y se proponen alternativas para subsanarlos.

En la actualidad, la herramienta SfM más extendida tanto en el ámbito académico como en aplicaciones industriales es COLMAP. Es una herramienta de código abierto de carácter modular basada en SfM y *Multi-View Stereo* (MVS), reconocida por su robustez, precisión,

flexibilidad, y amplia documentación. Teniendo en cuenta las ventajas citadas y que el repositorio Nerfstudio con el que se ha realizado este trabajo lo implementa como opción preferente para el preprocesado de sus datos de entrada, se ha optado por esta herramienta para la primera etapa del proceso diseñado. En la subsección 3.1 se habla de esta herramienta con mayor detalle.

Aunque COLMAP suponga el estándar reconocido por la industria, cabe destacar otras herramientas que obtienen buenos resultados para casos de uso concretos. Entre estas herramientas destacan VisualSfM [16], OpenMVG [17] y MVE [18], que priorizan distintos enfoques, como la rapidez o la facilidad de uso. También merece una mención especial GLOMAP (*Global and Local Mapping*)[19], un sistema reciente que ha despertado interés por su enfoque híbrido, capaz de combinar la precisión global de las optimizaciones tradicionales de SfM con una refinación local eficiente que reduce el tiempo de procesamiento sin sacrificar la calidad de la reconstrucción. Esta dualidad hace que GLOMAP resulte especialmente útil en aplicaciones en tiempo real o en escenarios de grandes dimensiones, donde la escalabilidad y la eficiencia computacional resultan cruciales.

2.3. Métodos basados en campos de radiancia: Nerf y Gaussian Splatting

Frente a los anteriores métodos, que poseen una amplia y contrastada trayectoria en todo tipo de aplicaciones robóticas, en los últimos años ha ido incrementándose el interés por los métodos de generación de modelos tridimensionales a partir de campos de radiancia, que permiten una representación mucho más detallada y precisa del entorno a partir de imágenes tomadas desde diversos puntos de vista.

De manera general, un campo de radiancia puede definirse como una función 5D que modela la distribución de la luz en el espacio observado por una cámara a partir de su posición y su orientación. Aunque existen diversos desarrollos dentro de este ámbito, las tecnologías de campos de radiancia neuronal (NeRF por sus siglas en inglés) y proyecciones gaussianas (más conocida por el término inglés Gaussian Splatting) han cobrado especial relevancia por su eficiencia computacional en el modelado de entornos complejos frente a las representaciones explícitas de la geometría del terreno a partir de mallas, nubes de puntos, etc. En adelante, en el presente trabajo se referirá al conjunto de estos enfoques como métodos basados en campos de radiancia o por las siglas en inglés RFM (*Radiance Field-based Methods*).

Es importante destacar que, aunque ambos métodos (NeRF y Gaussian Splatting) pueden incluirse en cierto sentido en la misma categoría general de RFM, difieren fundamentalmente en su naturaleza y en el grado en que pueden considerarse métodos de campos de radiancia. NeRF es un modelo de campo de radiancia “puro” y, además, implícito, es decir, representa la escena directamente mediante funciones continuas que modelan la densidad y la radiancia en el espacio tridimensional a través del aprendizaje de una red neuronal. Por el contrario, Gaussian Splatting, aunque también proyecta propiedades de la luz para generar nuevas vistas de la escena, lo hace de manera explícita, ya que almacena puntos clave en el espacio 3D y los asocia con distribuciones gaussianas que se proyectan al plano 2D para crear una imagen renderizada. Esto significa que, aunque Gaussian Splatting utiliza conceptos propios de los campos de radiancia, su representación explícita y basada en puntos aproxima la escena de forma diferente, como se observa en la Figura 7. Esto influye de manera directa en aspectos como la eficiencia computacional, la generalización de vistas y el tiempo necesario para el entrenamiento y la inferencia, como veremos a continuación .

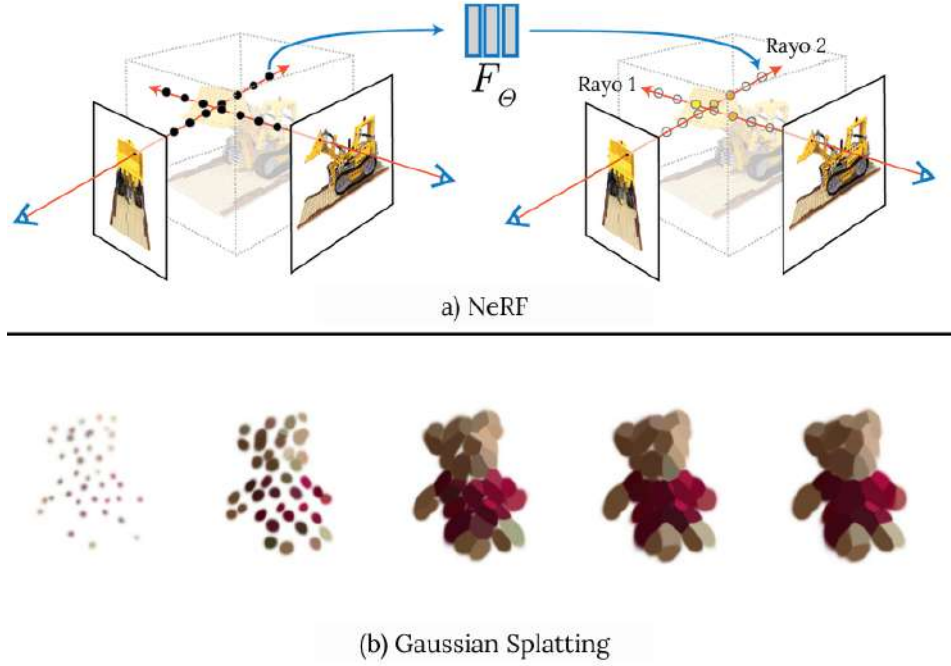


Figura 7: Comparativa conceptual entre NeRF y Gaussian Splatting. En el enfoque NeRF (a), se muestrean puntos a lo largo de un rayo y se evalúan mediante una red neuronal (MLP) para obtener color y densidad. Este tipo de representación se considera un modelo implícito, ya que la geometría y la apariencia de la escena no están directamente almacenadas como estructuras espaciales discretas, sino codificadas en los parámetros de la red neuronal (F_θ). En cambio, el enfoque Gaussian Splatting (b) emplea una nube de gaussianas tridimensionales que almacenan de forma explícita su posición, orientación, tamaño, opacidad y color, contribuyendo directamente a la imagen final mediante una etapa de fusión diferenciable. Este tipo de representación se denomina explícita porque la información tridimensional está directamente disponible y manipulable sin necesidad de ser inferida por una red. Imagen adaptada de [7].

Campos de radiancia neuronal (NeRF)

Los Campos Radiantes Neuronales (NeRF) son una técnica emergente en el ámbito de la reconstrucción 3D que utiliza redes neuronales profundas para modelar escenas tridimensionales a partir de imágenes 2D. Este enfoque no se limita a generar nubes de puntos o mapas de profundidad discretos; en su lugar, crea una representación continua de la escena que permite visualizarla desde cualquier ángulo, incluso desde puntos de vista que no están presentes en los datos originales. En esencia, la tecnología NeRF consiste en el entrenamiento de redes neuronales para establecer la relación entre los rayos de luz proyectados sobre la imagen, modelados por el color $c = (r, g, b)$ y la densidad volumétrica σ ; y la pose de la cámara, definida por la posición $x = (x, y, z)$ y la dirección $d = (\theta, \phi)$. Dicha relación se calcula usando la formulación integral:

$$C(r) = \int_{t_1}^{t_2} T(t) \sigma(r(t)) c(r(t), d) dt$$

donde dt representa la distancia diferencial atravesada por el rayo en cada paso de integración, $\sigma(r(t))$ y $c(r(t), d)$ la densidad volumétrica y el color del punto atravesado por el rayo en función de su posición y la dirección de este último; y $T(t)$ la transmitancia acumulada desde t_1 hasta t .

La fórmula anterior se computa dividiendo el rayo en N intervalos discretos:

$$C(r) = \sum_{i=1}^N \alpha_i T_i c_i, \quad T_i = \exp \left(- \sum_{j=1}^{i-1} \sigma_j \delta_j \right) \quad (1)$$

donde δ_i se refiere al intervalo entre muestras consecutivas t_i y t_{i+1} ; σ_i y c_i la densidad volumétrica y el color en el punto de muestreo. Por último, α_i describe la opacidad.

De manera general, el proceso seguido para la generación de NeRFs es el siguiente (Fig 8):

- Muestreo de rayos: partiendo de los parámetros extrínsecos (relativos a la posición y orientación de la cámara) e intrínsecos (que describen el sistema óptico por el que se transforman los puntos tridimensionales fotografiados en proyecciones 2D al tomar la imagen) el software proyecta un rayo desde cada píxel de la imagen hacia el centro de la cámara que representa la trayectoria de la luz. Posteriormente, se escogen puntos de muestreo distribuidos a lo largo de cada rayo y se almacenan sus coordenadas y la dirección del rayo. De esta manera, el modelo puede aprender cómo varían el color y la densidad en el espacio tridimensional a lo largo de la línea de visión de la cámara, lo que permite la representación de la escena desde vistas alternativas.
- Evaluación de los puntos muestreados: Cada punto se procesa para calcular el color del píxel como combinación RGB observado desde la dirección del rayo (radiancia) y la densidad volumétrica, parámetro que representa la proporción de la luz reflejada y absorbida por el punto y que da información sobre si pertenece a un espacio vacío o a un objeto consistente.
- Renderizado volumétrico: los valores generados por la red son integrados para el cálculo de la imagen renderizada desde una nueva perspectiva. Para ello, se calcula la contribución de cada uno de los puntos en la imagen asignándoles mayor o menor prioridad de manera proporcional a su cercanía a la pose de origen de la nueva vista y su densidad volumétrica.
- Optimización mediante el entrenamiento: Se escogen imágenes de prueba que no se aportan como información en el proceso de renderizado, se predicen vistas renderizadas para sus valores de posición y orientación y se compara el modelo con la imagen real (*ground truth*), calculando la diferencia y reajustando los pesos en sucesivas iteraciones para la optimización de la red neuronal mediante el método de descenso del gradiente, hasta que ésta converge y el modelo se corresponde con las imágenes reales para todos los datos de prueba.

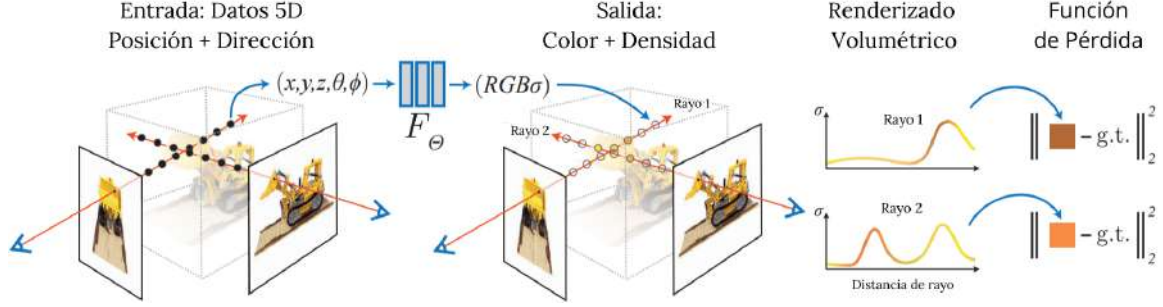


Figura 8: Proceso de entrenamiento y renderizado de vistas en NeRF. Se muestran, respectivamente y en cuatro etapas: el muestreo espacial rayo por rayo, a partir del cual se obtienen los datos de posición de los puntos y la dirección del rayo desde la cámara; la predicción de color y densidad mediante perceptrones multicapa (redes neuronales); la integración del color por rayo mediante renderizado volumétrico para sintetizar imágenes; y, finalmente, la comparación de las vistas predichas con las imágenes originales, y el ajuste del modelo mediante una función de pérdida. Adaptado de [8].

Aunque estos pasos son más o menos comunes a todos los métodos de nerf, existen numerosas variantes que permiten aumentar la eficacia del modelo original u optimizarlo para misiones y objetivos concretos. Las innovaciones presentadas por métodos como Mip-Nerf [20], que reduce el *aliasing* (artefactos visuales producidos por un muestreo insuficiente) en escenas con cambios de escala y vistas amplias; Instant-NGP [21], que maximiza la eficiencia computacional mediante el uso de hash tables multirresolución para las reconstrucciones; Meganerf [22], que permite generar reconstrucciones de gran tamaño y Nerf-W [23], que es capaz de trabajar con imágenes con diferentes condiciones de iluminación y elementos no deseados fundamentalmente mediante el uso de vectores latentes que codifican la apariencia.

Gaussian Splatting

El método de salpicaduras gaussianas o Gaussian Splatting es un método probabilístico que ha ganado atención en la comunidad científica por su eficiencia y flexibilidad. Aunque no se basa en redes neuronales, se parece al NeRF en el sentido de que ambos generan puntos en el espacio y estudian las propiedades de la luz reflejada sobre ellos aproximando la apariencia de la escena. Sin embargo, en el caso del Gaussian Splatting estos puntos se asocian a distribuciones Gaussianas 3D: funciones suaves y diferenciables que pueden proyectarse al espacio 2D como “manchas” o *splats* que pueden ser mezcladas con las otras que coincidan sobre el mismo punto [24]. Estas Gaussianas son matrices de covarianza 3D de la forma

$$G(x) = \alpha e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)} \quad (2)$$

Donde α representa el valor máximo, que puede modelar, por ejemplo, el color, x el vector de posición (x, y, z) en el espacio, μ el centro de la gaussiana en el que se alcanza el valor α y Σ la matriz de covarianza.

Al renderizar, estas gaussianas han de ser proyectadas sobre el plano 2D. La función gaussiana en 2D, a la que llamaremos $G'(x)$, se modela igual que su equivalente 3D pero proyectando los centros de cada punto μ mediante una transformación de cámara (p.ej.

Pinhole simple) y transformando la matriz de covarianza a coordenadas de cámara. Tradicionalmente:

$$\Sigma' = JW\Sigma W^T J^T \quad (3)$$

Reduciendo posteriormente Σ' a una matriz Σ_{2D} de 2×2 obviando la profundidad.

Una vez claros los fundamentos matemáticos, podemos detenernos en el proceso exacto por el que generamos la reconstrucción. En resumidas cuentas, el procedimiento iterativo por el que generamos un modelo 3D mediante Gaussian Splatting es similar al empleado para NeRF, aunque con diferencias significativas:

- **Muestreo de rayos:** De manera análoga a la descrita para la generación de NeRF, se proyectan rayos desde cada píxel de la imagen. A lo largo de cada rayo se obtienen puntos clave en el espacio 3D que modelan la geometría de la escena, habitualmente estimados mediante técnicas tradicionales como SfM (Structure from Motion).
- **Generación de gaussianas 3D:** Cada uno de estos puntos se convierte en una gaussiana tridimensional mediante la función $G(x)$, que asigna a cada punto parámetros como posición, orientación, tamaño, opacidad y color. Estas gaussianas constituyen una representación explícita y continua de la escena, que se ajustará progresivamente durante el entrenamiento.
- **Proyección de las gaussianas al plano de imagen:** Con cada iteración del entrenamiento, las gaussianas se proyectan al plano 2D mediante las transformaciones de cámara, generando “manchas” (*splats*) que contribuyen a la imagen final a través de una etapa de renderizado diferenciable.
- **Entrenamiento del modelo:** Se compara la imagen generada por la proyección de las gaussianas con las imágenes reales correspondientes a la misma pose de cámara. A partir del error de reconstrucción, se retropropaga la información para ajustar los parámetros de las gaussianas 3D. Así, en cada iteración, las proyecciones deberían parecerse más al *ground truth* (imagen real capturada desde la misma vista). Aunque se emplea descenso por gradiente, el método evita el uso de redes neuronales, lo que reduce significativamente las necesidades computacionales y el tiempo de procesamiento.

Al igual que sucede con los NeRF, existen varias alteraciones del método original de Gaussian Splatting que le otorgan distintas capacidades y lo adaptan a situaciones de uso variadas. Entre ellas, cabe destacar Splatfacto-W [11], que combina las capacidades de NeRF-W [23] para manejar alteraciones en la apariencia de las imágenes y objetos no deseados mediante vectores latentes y la eficiencia computacional intrínseca al Gaussian Splatting.

Aplicaciones Actuales de NeRF y Gaussian Splatting

Pese a tratarse de tecnologías recientes y en desarrollo, los RFM han supuesto un profundo avance frente a otras técnicas de generación de modelos 3D por su precisión y calidad visual. En la actualidad, estos métodos ya están siendo utilizados en diversos sectores que los demandan por sus prestaciones. Cabe destacar algunos de los más significativos en términos de implantación:

Fotogrametría Si hay un campo en el que las técnicas basadas en campos de radiancia neuronal y Gaussian Splatting han experimentado una implantación significativa es en el de la generación de modelos 3D de entornos naturales y obras arquitectónicas de diversa índole. Sus prestaciones y características particulares las han consolidado como opciones preferentes en sectores como la arquitectura, la ingeniería civil, la arqueología y la gestión del patrimonio cultural (Fig9), donde el nivel de detalle y la exactitud en la representación son esenciales para alcanzar distintos objetivos técnicos y científicos.



Figura 9: Reconstrucción tridimensional mediante técnicas de fotogrametría basadas en NeRF de la Torre di Caprona en Pisa, Italia. Imagen adaptada de [9].

En el campo de la arquitectura e ingeniería, se ha discutido las ventajas técnicas de estas tecnologías teniendo en mente procesos como la planificación urbana, la gestión inteligente de ciudades y la documentación de proyectos frente a métodos preexistentes [25]. La posibilidad de generar modelos tridimensionales a partir de fotografías permite crear reconstrucciones virtuales precisas de edificios y estructuras, lo que puede emplearse tanto para analizar el estado actual de una construcción como para simular su comportamiento y optimizar su rendimiento a lo largo del tiempo mediante la predicción de fallos estructurales, la programación de tareas de mantenimiento y la planificación de intervenciones futuras, reduciendo costos operativos en el proceso. Asimismo, resultan trascendentales sus aplicaciones para la creación de gemelos digitales de la infraestructura de las ciudades para posibles problemas relacionados con la densidad urbana, el tráfico y la accesibilidad.

En otras materias como la arqueología y la conservación del patrimonio, NeRF y Gaussian Splatting han mejorado significativamente la capacidad para preservar digitalmente monumentos, sitios arqueológicos y piezas valiosas que pueden encontrarse en riesgo de deterioro o desaparición debido al paso del tiempo, a la acción de las fuerzas de la naturaleza o eventos imprevistos de toda índole, con casos de uso de repercusión mundial como su contribución en las tareas de reconstrucción de la catedral de Notre Dame de París tras el incendio de 2019 [26]. Estudios independientes como [27] han puesto a prueba sus prestaciones frente a técnicas tradicionales de fotogrametría, mostrando sus ventajas al describir características materiales, en especial al trabajar con datos de entrada limitados. Son particularmente relevantes las aplicaciones de este enfoque en proyectos de restauración arquitectónica, en los que resulta crucial la documentación rigurosa de las condiciones previas a cualquier intervención. Los modelos 3D generados a partir de estas tecnologías no solo proporcionan

información precisa del estado actual de las estructuras, sino que también permiten realizar estudios comparativos a lo largo del tiempo para evaluar posibles cambios o daños estructurales. Además, estas representaciones pueden servir para la creación de réplicas virtuales accesibles al público, facilitando la divulgación y el acceso remoto a bienes culturales que, de otro modo, serían difíciles de visitar o estudiar.

Por otro lado, la combinación de estas técnicas con herramientas complementarias, como drones o sensores avanzados, incrementa aún más su potencial. Desarrollos como FlyNeRF [28] destacan por la integración conjunta de ambas tecnologías para tareas de mapeo y generación de modelos 3D. Esta sinergia permite realizar análisis topográficos y modelados de sitios a gran escala, facilitando así una mejor comprensión del entorno. Esto resulta particularmente valioso en la planificación de operaciones con sistemas robóticos semiautónomos como las que se describen en el presente trabajo.

Realidad Aumentada (AR) y Realidad Virtual (VR) Aunque es una de sus aplicaciones más inmediatas, el impacto de NeRF y Gaussian Splatting no se limita al ámbito de la fotogrametría. Actualmente, son relevantes las contribuciones de ambos métodos en el desarrollo de aplicaciones de realidad aumentada (AR) y realidad virtual (VR), mejorando la calidad y realismo de los modelos y favoreciendo la inmersividad de la experiencia. En particular, destaca el impacto en el sector del entretenimiento, en el que los diseñadores de videojuegos y experiencias de ocio virtual se ven beneficiados con la mejora de los gráficos.

Además, los RFM también se están viendo utilizados en educación y simulación, con usos potenciales como el desarrollo de simulaciones quirúrgicas ultrarrealistas que permitan a los estudiantes de medicina practicar técnicas complejas sin poner en riesgo la vida de pacientes reales. [29] De manera análoga, la implementación de estos métodos en simuladores de vuelo o de conducción resulta muy ventajosa y facilita que la experiencia pueda ser lo más parecida posible a la realidad

Industria del Seguro y Evaluación de Daños Otro sector en el que estas herramientas están empezando a demostrar un alto potencial es el de los seguros. La generación de modelos 3D sin necesidad de equipos sofisticados y con un grado de detalle cada vez mayor permite valoraciones mucho más precisas, objetivas y sin que sea necesario el desplazamiento recurrente al lugar del siniestro, reduciendo significativamente el tiempo de resolución de las reclamaciones. Además, el peritaje queda mejor registrado y la eficiencia se ve mejorada al disminuir la probabilidad de errores humanos y aumentar la especificidad e imparcialidad de los peritajes. Un buen ejemplo puede verse en la Figura 10

Construcción y Minería Como en los casos anteriores, también en el ámbito de la construcción se están utilizando las técnicas de modelado 3D para monitorear el progreso de importantes obras. Al ser capaz NeRF y Gaussian Splatting de producir modelos tan detallados, se puede formar una imagen del proceso de construcción y compararlo con el diseño original, lo que permite corregir errores o desviaciones en cuanto aparecen. Gracias a ello, la toma de decisiones se flexibiliza y disminuye el riesgo de problemas complejos debido a un diseño incorrecto. En la Figura 11 se muestran algunos ejemplos.

³Modelo 3D interactivo disponible en <https://sketchfab.com/models/abc123>. Autor: Michal Gula <https://x.com/michalgula/status/1729806941461913923>



Figura 10: Modelo Gaussian Splatting para evaluación de daños de un vehículo vandalizado.³



Figura 11: Modelos de 3D Gaussian Splatting aplicados a la supervisión de obras de construcción. Esta técnica permite una reconstrucción eficiente y visualmente fidedigna de escenas complejas en entornos industriales. Imagen adaptada de [10].

En relación con la minería, estas técnicas están siendo utilizadas para supervisar y controlar el terreno ocupado por la empresa, así como para administrar recursos. Al analizar los modelos tridimensionales resultantes, la compañía puede estudiar la topografía de un área explotable, evaluar la viabilidad y condiciones en el emplazamiento en que se plantean realizar las excavaciones y planificar futuras operaciones, mejorando la eficiencia del proceso y aumentando la seguridad de los operarios.

Conducción Autónoma Para terminar, otro campo que se está empezando a beneficiar de las ventajas aportadas por estas técnicas es el de la conducción autónoma. Dado que los vehículos autónomos dependen en gran medida de su capacidad para percibir y comprender el entorno que los rodea, las técnicas de modelado 3D pueden resultar de gran utilidad para la mejora de sus capacidades al aumentar significativamente el alcance de esta percepción. Al generar mapas tridimensionales detallados en tiempo real, los vehículos pueden detectar obstáculos con mayor precisión, calcular distancias de manera más fiable y anticipar mejor

los movimientos de otros vehículos o peatones. Esto contribuye a la mejora de la eficiencia y seguridad de la navegación, lo que es clave para el futuro de la movilidad autónoma. Además, cabe destacar su utilidad en la generación de modelos tridimensionales de entornos reales de conducción [30] para el entrenamiento de las inteligencias artificiales que hacen posible la conducción autónoma).

Ventajas e inconvenientes de las metodologías

Los RFM han supuesto un salto adelante con respecto a métodos anteriores de representación por su versatilidad y elevado grado de detalle. Además, presentan diversas ventajas que los hacen atractivos para tareas de reconstrucción en general y en especial en el contexto de la robótica espacial de exploración planetaria. Algunas de las más relevantes son:

- Alta fidelidad en la reconstrucción de escenas: Ambos métodos permiten generar representaciones extremadamente realistas del entorno. Esta característica, que de por sí resulta una evidente ventaja, resulta fundamental en el contexto de la exploración planetaria, donde la caracterización e identificación del terreno resulta en numerosas ocasiones compleja por la uniformidad y falta de características definitorias de muchos entornos, dificultando las tareas de localización absoluta por falta de referencias fiables. También resulta muy atractiva para la generación de simulaciones de alta precisión.
- Representaciones continuas y compactas: a diferencia de las mallas tradicionales y las nubes de puntos, los RFM generan una representación continua del espacio, lo que permite almacenar modelos más detallados con un consumo inferior de memoria (aunque este todavía es significativo en el caso de Gaussian Splatting). Dado que las necesidades computacionales a bordo suponen una de las principales limitaciones del espacio, esta ventaja puede resultar especialmente trascendental.
- Renderización en tiempo real (Gaussian Splatting): La inferior carga computacional de Gaussian Splatting hace factible la generación de vistas en tiempo real, permitiendo su uso para aplicaciones donde es crucial la actualización rápida de la percepción del entorno.
- Capacidad de modelado en condiciones de iluminación variables. Algunas variantes de RFM, como NeRF-W y Splatfacto-W (que supone el método de referencia de este trabajo), son robustas ante condiciones de iluminación variables, lo cual es importante en entornos espaciales donde la luz puede cambiar drásticamente debido a la rotación planetaria o sombras proyectadas por montañas y cráteres.
- Compatibilidad con sensores no ópticos: Si bien estos métodos se desarrollaron para imágenes RGB, investigaciones recientes han explorado su aplicación con sensores de profundidad y LiDAR, lo que puede ser útil en la navegación autónoma de *rovers* en superficies planetarias donde las imágenes RGB pueden ser insuficientes.

No obstante, todavía presentan importantes carencias que han de ser solventadas antes de ser utilizadas de manera funcional en el contexto de la localización robótica entre las que destacan las siguientes:

- Coste computacional: Las representaciones más detalladas o de superficies más extensas en el método NeRF original pueden llegar a tomar días, lo que hace inviable su uso en tiempo real. Aunque es necesaria una mayor investigación y desarrollo en este sentido, en los últimos años se han visto progresos esperanzadores con métodos de

nueva generación más eficientes y que consiguen tiempos de entrenamiento bastante inferiores, como los modelos generalizables preentrenados.

- Calidad necesaria para los datos de entrada: La red neuronal se nutre de imágenes y datos precisos sobre sus parámetros de cámara, lo que resulta contradictorio en un contexto en el que lo que se pretende obtener es precisamente algunos de dichos parámetros, por lo que es necesario un preprocesamiento que hace que por el momento NeRF sea más útil como parte de un sistema multicapa, para aumentar la precisión de la localización o convertir una localización relativa en una absoluta que además aporte información del entorno.
- Dificultad para procesar escenas complejas: los modelos más consolidados de NeRF presentan dificultades en la representación de entornos extensos como los que se presuponen en tareas de exploración. Métodos en desarrollo como NeRF-W o las técnicas de superposición de NeRF más pequeños se encaminan a resolver este problema.
- Dependencia de la calidad de las imágenes, de las trayectorias seguidas en su toma y de las condiciones de luminosidad: dado que las redes NeRF modelan la acción de la luz sobre las superficies desde distintas perspectivas, para un mejor funcionamiento del algoritmo es deseable que la iluminación se mantenga estable en la medida de lo posible y que la trayectoria seguida al tomar imágenes cubra un rango amplio de perspectivas del mismo punto observado desde distintas poses. Además, a la hora de preestimar la pose es habitual utilizar técnicas Structure from Motion (SfM) que requieren de un solapamiento suficiente entre imágenes sucesivas. Estos requisitos pueden ser difíciles de cumplir en algunos casos, ya que se espera que el *rover* siga trayectorias rectas y funcione de forma completamente autónoma.
- Falta de información tridimensional directa: A diferencia de otros métodos tradicionales, las representaciones generadas por NeRF no generan estructuras 3D estándar como mallas o nubes de puntos, sino que suponen una representación continua del espacio basada en la generación de vistas completamente nuevas de la escena desde coordenadas posicionales y rotacionales aportadas por el usuario. Esto supone un problema para ciertos métodos de localización, ya para cruzar los modelos 3D con renderizaciones tridimensionales estándar es necesario un procesamiento adicional que aumenta el tiempo de procesamiento y los requisitos computacionales y no siempre es capaz de representar de manera fidedigna las superficies y texturas 3D.

2.4. Comparativa entre MRFs y métodos tradicionales de localización y mapeo

A continuación se presentan una serie de tablas comparativas que resumen las principales características de algunos de los métodos de mapeo más relevantes abordados en este trabajo. La comparación abarca desde técnicas tradicionales de SLAM hasta las tres tecnologías que se han implementado en el proyecto: SfM, NeRF y Gaussian Splatting.

Aunque todos estos métodos comparten la capacidad de representar entornos tridimensionales a partir de un conjunto limitado de imágenes, difieren significativamente en su funcionamiento, naturaleza y ámbitos de aplicación. El objetivo de estas comparativas es proporcionar una visión clara de sus ventajas, limitaciones y requisitos, con el fin de facilitar la selección del enfoque más adecuado según el contexto de uso.

Tabla 1

Parámetro	Visual SLAM	LiDAR SLAM	Radar SLAM
Tiempo de procesado	Casi real optimizado	Variable	Tiempo real
Precisión de reconstrucción	Media-Alta	Alta	Media
Coste computacional	Medio-Alto	Alto	Medio
Uso de memoria	Moderado	Alto	Medio
Compatibilidad con trayectorias regulares	Buena	Excelente	Alta
Dependencia del solapamiento de imágenes	Alta	Baja	Nula
Calidad de imagen necesaria	Media-Alta	No aplica	No depende
Robustez ante iluminación	Sensible	Robusto	Muy robusto
Sensores adicionales	Cámaras	LiDAR	Radar + IMU
Representación 3D estándar	Sí	Sí	Sí
Tamaño mínimo de dataset	Bajo	Medio	Bajo
Entrenamiento requerido	No	No	No
Capacidad de actualización incremental	Alta	Alta	Media
Fusión multisensorial	Media	Alta	Alta
Dependencia de calibración previa	Alta	Media	Baja
Requisitos de ancho de banda	Bajo	Medio	Bajo

Tabla 1: Comparación entre Visual SLAM, LiDAR SLAM y Radar SLAM

Los métodos tradicionales de SLAM suponen el estándar en la mayoría de misiones de robótica de exploración por su combinación de precisión, coste computacional y robustez. Visual SLAM [31] es posiblemente la solución más equilibrada y ampliamente adoptada para tareas generales, ya que requiere una infraestructura accesible y barata (cámaras fotográficas). En general, su calidad y precisión resultan más que adecuadas para entornos bien iluminados y trayectorias regulares, aunque se ve limitado ante condiciones de baja visibilidad e iluminación irregular. LiDAR SLAM [32], por su parte, proporciona alta precisión, mayor compatibilidad con trayectorias complejas y una robustez muy superior ante condiciones visuales difíciles; no obstante, los sensores de los que depende esta tecnología son generalmente costosos, y su implementación supone un incremento significativo de la demanda computacional y energética. Radar SLAM [33] resulta un término medio entre la eficiencia y bajo costo del primero y la robustez y riqueza de información del segundo, sobresaliendo en escenarios con escasa iluminación o condiciones ambientales adversas, como misiones planetarias, túneles o exploración subterránea. Su baja dependencia de calibración previa, alta robustez y capacidad de operación en tiempo real lo convierten en la opción más fiable cuando la visibilidad óptica no está garantizada.

Tabla 2

Parámetro	SfM (Structure from Motion)
Tiempo de procesado	Lento
Precisión de reconstrucción	Alta
Coste computacional	Alto
Uso de memoria	Alto
Compatibilidad con trayectorias regulares	Baja
Dependencia del solapamiento de imágenes	Muy alta
Calidad de imagen necesaria	Alta
Robustez ante iluminación	Sensible
Sensores adicionales	Imágenes RGB
Representación 3D estándar	Sí
Tamaño mínimo de dataset	Medio
Entrenamiento requerido	No
Capacidad de actualización incremental	Baja
Fusión multisensorial	Baja
Dependencia de calibración previa	Alta
Requisitos de ancho de banda	Bajo

Tabla 2: Evaluación de Structure from Motion

Estructura desde el Movimiento (SfM) sigue siendo un método sólido para la reconstrucción 3D cuando se dispone de un entorno controlado, imágenes de alta calidad y tiempo de procesamiento suficiente. Aunque tiene similitudes con Visual SLAM en su planteamiento, se diferencia principalmente de él en su enfoque global frente al incremental del primero: Visual SLAM compara cíclicamente y en tiempo real la última imagen tomada con el modelo reconstruido hasta ese instante, mientras que SfM requiere la totalidad de las imágenes desde el principio, llevando a cabo un proceso más complejo y preciso. Por ello, es capaz de generar reconstrucciones de gran exactitud, pero presenta limitaciones significativas en términos de eficiencia y escalabilidad, siendo uno de los procesos que requieren de mayor tiempo para realizar su tarea. Pese a ello, es posible paliar en gran medida este efecto mediante la adopción de un enfoque secuencial en el emparejamiento de características y una correcta configuración del proceso, como se observa en las tablas de la sección 4

Su dependencia del solapamiento elevado entre imágenes, la sensibilidad a la iluminación y la necesidad de una calibración precisa lo hacen poco adecuado para entornos dinámicos o no estructurados. En consecuencia, SfM se recomienda principalmente para aplicaciones en arqueología, cartografía o documentación arquitectónica, donde los datos pueden capturarse cuidadosamente y no se requieren resultados en tiempo real.

Tabla 3

Parámetro	NeRF (original)	NeRF in the Wild	Nerfacto
Tiempo de procesado	Lento	Lento	Mejorado
Precisión de reconstrucción	Muy alta	Alta	Alta
Coste computacional	Muy alto	Muy alto	Alto
Uso de memoria	Alto	Alto	Moderado
Compatibilidad con trayectorias regulares	Baja	Baja	Media
Dependencia del solapamiento de imágenes	Muy alta	Alta	Alta
Calidad de imagen necesaria	Alta	Alta	Media-Alta
Robustez ante iluminación	Sensible	Robusto	Robusto
Sensores adicionales	Imágenes RGB cali- bradas	Imágenes sin calibrar	Imágenes calibradas
Representación 3D estándar	No directa- mente	No directa- mente	No directa- mente
Tamaño mínimo de dataset	Alto	Alto	Medio
Entrenamiento requerido	Sí, intensivo	Sí, intensivo	Sí, optimi- zado
Capacidad de actualización incremental	Baja	Baja	Media
Fusión multisensorial	Baja	Media	Media
Dependencia de calibración previa	Alta	Baja	Media
Requisitos de ancho de banda	Alto	Alto	Medio

Tabla 3: Comparación entre NeRF original, NeRF in the Wild y Nerfacto

Los modelos basados en NeRF representan un avance significativo al permitir reconstrucciones fotorrealistas desde puntos de vista no capturados originalmente, gracias a su representación continua del espacio. Sin embargo, el modelo original presenta un elevado coste computacional, altos requerimientos de memoria y un tiempo de entrenamiento prolongado, lo que dificulta su uso en sistemas embarcados o aplicaciones en tiempo real. NeRF in the Wild mejora la robustez frente a condiciones no controladas, como iluminación cambiante y datos no calibrados, pero mantiene una alta demanda de recursos. Por su parte, Nerfacto introduce optimizaciones sustanciales que reducen los requisitos computacionales y de memoria, haciendo más viable su uso práctico sin sacrificar significativamente la calidad de reconstrucción. En resumen, NeRF es ideal para lograr la máxima fidelidad visual cuando se cuenta con suficiente potencia computacional; Nerfacto ofrece un equilibrio entre eficiencia y calidad para entornos con recursos limitados; y NeRF in the Wild es el más robusto ante datos imperfectos, aunque con menor eficiencia y mayores requerimientos que Nerfacto.

Tabla 4

Parámetro	3D Gaussian Splatting	Splatfacto	Splatfacto in the Wild
Tiempo de procesado	Rápido	Muy rápido	Rápido
Precisión de reconstrucción	Alta	Alta	Media-Alta
Coste computacional	Alto	Medio	Medio
Uso de memoria	Bajo	Bajo	Moderado
Compatibilidad con trayectorias regulares	Media	Alta	Alta
Dependencia del solapamiento de imágenes	Media	Media	Alta
Calidad de imagen necesaria	Media	Media	Alta
Robustez ante iluminación	Más robusto que NeRF	Robusto	Robusto
Sensores adicionales	Imágenes RGB	Imágenes RGB	Imágenes RGB no calibradas
Representación 3D estándar	Parcialmente	Parcialmente	Parcialmente
Tamaño mínimo de dataset	Medio	Medio-Bajo	Bajo
Entrenamiento requerido	Sí, rápido	Parcial	Parcial
Capacidad de actualización incremental	Media	Alta	Alta
Fusión multisensorial	Baja	Media	Media
Dependencia de calibración previa	Media	Baja	Baja
Requisitos de ancho de banda	Medio	Bajo	Bajo

Tabla 4: Comparación entre 3D Gaussian Splatting, Splatfacto y Splatfacto in the Wild

Las variantes de Gaussian Splatting ofrecen una alternativa eficiente frente a NeRF, al lograr tiempos de procesado mucho más rápidos y una menor carga computacional, sin comprometer de forma significativa la calidad de reconstrucción visual. El modelo original ya presenta un buen equilibrio entre fidelidad y eficiencia, pero variantes como Splatfacto y Splatfacto in the Wild mejoran aún más aspectos clave como la capacidad de actualización incremental, la robustez frente a cambios en la iluminación y la dependencia de calibración. Estas características las hacen especialmente adecuadas para aplicaciones en entornos no estructurados, misiones de exploración autónoma o sistemas con recursos limitados. En particular, Splatfacto destaca por su bajo uso de memoria y su buena adaptabilidad a trayectorias complejas, mientras que Splatfacto in the Wild extiende esta adaptabilidad a escenarios reales con imágenes no calibradas.

3. Metodología

3.1. Herramientas y tecnologías utilizadas

Nerfstudio

Nerfstudio [34] es un repositorio de código abierto diseñado para facilitar el uso, desarrollo y experimentación con RFMs. Su objetivo es proporcionar al usuario un entorno unificado y modular que automatiza la mayoría de tareas asociadas a la generación de modelos 3D con NeRF y Gaussian Splatting, facilitando el procesamiento de datos y su uso para el entrenamiento de los modelos de manera sencilla y eficiente.

La biblioteca incluye los principales modelos de ambos métodos, así como varios conjuntos de datos para poder trabajar con ellos. Asimismo, también implementa COLMAP y otras herramientas que permiten importar conjuntos de fotografías sin información adicional, obtener la pose de las cámaras e información dispersa 3D del entorno y generar archivos con los que interactuar y entrenar modelos. También incluye una GUI capaz de visualizar y desplazarse por algunas de las representaciones en tiempo real y de software de renderizado para generar videos y grabaciones exactas de las reconstrucciones realizadas.

En la actualidad su código y todos los modelos que incluye son accesibles de manera gratuita en Github [35], estando sometido a un proceso de mejora continua gracias a su comunidad.⁴

En el presente trabajo, el repositorio ha sido modificado para hacerlo capaz de extraer los datos directamente de los *rosbags* de los *rovers* y adaptarlos a formatos compatibles. Además, se ha sustituido la versión de COLMAP implementada por la última versión, cuyo código también ha sufrido modificaciones para su adaptación a las características de los conjuntos de datos escogidos; y se ha utilizado para la generación de los modelos 3D, su visualización y la generación de vistas y renders.

Método RFM escogido: Splatfacto-W Tras llevar a cabo numerosas pruebas que se describen en detalle a lo largo de esta sección, se ha optado por el uso del método Splatfacto in the Wild (O Splatfacto-W) [11] para llevar a cabo las reconstrucciones 3D.

La principal característica de este método es su especialización en datos tomados “*in the wild*” es decir, en entornos no supervisados, con condiciones luminosas cambiantes y objetos en movimiento, que generalmente son de difícil procesamiento aplicando otros sistemas más tradicionales. Frente al 3D Gaussian Splatting original, que asocia a las gaussianas 3D que definen cada punto del modelo una posición, tamaño, opacidad y color fijos, Splatfacto-W añade por cada Gaussiana una red Neuronal reducida que predice el color de cada punto en función de ciertos parámetros, permitiendo generar reconstrucciones más expresivas y gestionar mejor las variaciones fotométricas en las imágenes de partida sin sacrificar significativamente el tiempo de renderizado del modelo.

Otra novedad de gran interés es la adaptación a Gaussian Splatting del concepto de vectores latentes utilizado en NeRF-W. Estos vectores contienen información numérica abstracta sobre cómo se ve una imagen: su iluminación, color dominante, ambiente, etc. Al ser independientes a la Gaussiana, permiten aislar las condiciones de apariencia del modelo sin alterar su estructura tridimensional. Gracias a ello, el método ofrece la posibilidad de representar un mismo volumen 3D adaptado a las condiciones lumínicas y de color de cualquiera

⁴<https://github.com/nerfstudio-project/nerfstudio>

de las fotografías a partir de las que ha sido generado (Fig 12).



Figura 12: Ejemplo de síntesis de vistas con distintas apariencias en el mismo modelo 3D de la Fontana di Trevi, generado con Splatfacto-W. Ambas imágenes provienen de la misma reconstrucción geométrica, pero utilizan diferentes vectores latentes de apariencia para reflejar condiciones de iluminación contrastantes (día y noche). Esta capacidad de desacoplar geometría y apariencia permite generar vistas realistas en contextos variados a partir de un único conjunto de imágenes. Imagen tomada de [11].

Como inconvenientes, cabe destacar que estos vectores no contienen información estructurada y no son calculados directamente, sino que se aprenden mediante un proceso iterativo durante el entrenamiento, por lo que no es posible identificar con qué característica se corresponde cada elemento del vector para crear escenarios con características visuales personalizadas, ya que con cada entrenamiento variará el parámetro asociado a cada una de las variables. Para paliar esta problemática, una posibilidad es interpolar entre los parámetros de dos fotografías utilizadas en el mismo entrenamiento para buscar condiciones intermedias a las que presentan cada una de ellas.

Gracias a estas particularidades, Splatfacto-W logra un incremento medio de 5.3 dB en PSNR con respecto al 3D Gaussian Splatting original, y alcanza velocidades de entrenamiento hasta 150 veces superiores a los métodos basados en NeRF, por lo que se presenta como la mejor alternativa para la generación de modelos a partir de imágenes tomadas en entornos no controlados, combinando la velocidad y eficacia del primero con la flexibilidad en la representación de los segundos.

COLMAP

COLMAP⁵ es un software de código abierto para la caracterización y reconstrucción de entornos 3D basado en técnicas de Structure from Motion (SfM). Partiendo de un conjunto de imágenes de un mismo objeto o espacio tomadas desde diferentes perspectivas, el software identifica las características definitorias de cada una, las empareja estableciendo correspondencia entre imágenes contiguas, triangulándolas para su caracterización como puntos 3D y la generación de un modelo disperso de la escena. A continuación, se sigue un proceso de optimización global y refinado de las poses mediante ajuste de haz. Finalmente, se generan y exportan archivos que contienen el modelo generado, existiendo varias posibilidades en este sentido.

Además, el proceso permite la importación de modelos externos y la inyección de coordenadas GNSS para su uso como información inicial y refinamiento, acelerando la triangulación de puntos y el ajuste de haces y mejorando la precisión de los datos de los sensores. En este trabajo se ha investigado sobre las posibilidades de esta funcionalidad para acercarse en la

⁵<https://colmap.github.io/>

medida de lo posible a la capacidad de procesamiento en tiempo real.

Cabe destacar que Nerfstudio implementa una versión de COLMAP que puede utilizarse para obtener las poses de las cámaras necesarias para el entrenamiento de NeRFs y generar nubes de puntos que mejoran significativamente la velocidad y rendimiento del entrenamiento de Gaussian Splatting. También este proyecto se sirve de él en su algoritmo, aunque se han aplicado modificaciones para la automatización de ciertos procesos en el flujo conjunto con Nerfstudio y ROS 2 y se ha adaptado el método de emparejado a las características de los datos inyectados (generalmente de trayectorias regulares, con imágenes consecutivas y bien solapadas) para su optimización en términos de eficiencia y velocidad. Debido a esto, finalmente ha sido necesario implementar una versión independiente.

ROS 2 y formatos *rosvbag*

ROS 2 es la segunda generación del *Robot Operating System* (ROS), un conjunto de librerías y bibliotecas diseñadas para facilitar el desarrollo de aplicaciones robóticas. Supone un entorno de desarrollo de código abierto que permite el acceso y comunicación en tiempo real de los sensores, actuadores, computadoras y demás partes del robot[36]. También permite el almacenamiento de la información recopilada por el robot en archivos a los que denomina *rosvbags*. En este proyecto su uso es fundamental para operar el *rover* sobre el terreno, capturando y almacenando datos a través de sus sensores en tiempo real y organizándolos de forma estructurada en sus *rosvbags*. También ha sido implementado en el flujo de trabajo de reconstrucción como primera etapa del proceso, siendo necesaria su instalación para la extracción, preparación y administración de dichos datos. La versión del sistema operativo implementada para este trabajo es **Humble**.⁶

3.2. Proceso de experimentación con NeRF y Gaussian Splatting

Primeros Pasos: fase previa de la investigación

En la etapa inicial de investigación de este trabajo, se experimentó con los RFM, implementando diferentes configuraciones y métodos de los mismos para comprobar sus posibilidades y limitaciones. En primera instancia, se procedió a probar los modelos de NeRF y Gaussian Splatting por defecto desarrollados por la comunidad de Nerfstudio, Nerfacto y Splatfacto respectivamente. Estos métodos se caracterizan por su eficiencia computacional, teniendo como principal objetivo la reducción del tiempo de entrenamiento sin sacrificar excesivamente la calidad de la reconstrucción. En esencia, ambos modelos seleccionan los avances y mejoras de funcionamiento más robustas entre los métodos realizados por equipos de investigación de todo el mundo y los unifican en un solo método sin añadir aportaciones propias, suponiendo en cierto modo los métodos resultantes un estándar para las reconstrucciones de uso general.

En esta primera aproximación se optó igualmente por un *dataset* de serie cuyo funcionamiento estaba contrastado para ambos RFMs: una reconstrucción de tamaño mediano de una habitación con mobiliario variopinto que supuso un buen punto de partida (Fig. 13).

⁶<https://docs.ros.org/en/humble/index.html>

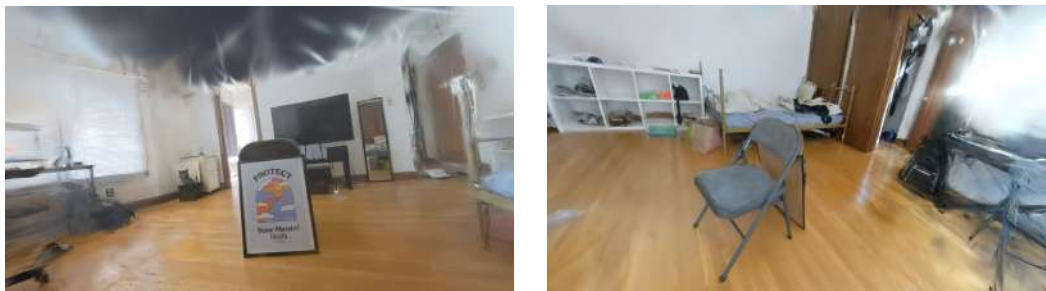


Figura 13: Vistas renderizadas mediante el método Nerfacto del *dataset poster*, el más sencillo e inmediato de los que ofrece Nerfstudio. Puede descargarse directamente desde la línea de comandos de dicho entorno con el comando `ns-download-data nerfstudio`.⁷

Los resultados, como era de esperar para métodos y datos contrastados, fueron satisfactorios, permitiendo al autor familiarizarse con la tecnología y las herramientas utilizadas y observar algunas diferencias entre ambos métodos.

La primera conclusión obtenida es en relación a la velocidad del entrenamiento, pudiendo comprobarse cómo, pese a haber utilizado una de las versiones más rápidas existentes en la actualidad (Splatfacto), el modelo NeRF tardaba un tiempo significativamente mayor que el de Gaussian Splatting en realizar la reconstrucción. En términos de calidad, también se puede apreciar cómo el nerf presenta una mayor resolución, pero, en todo caso, Splatfacto presenta unos resultados más que satisfactorios para los fines de este trabajo, siendo la reconstrucción fidedigna y aportando la información necesaria del entorno. Este hecho, sumado a la información geométrica explícita generada durante el entrenamiento, hace que en este punto de la investigación empiecen a considerarse los modelos de Gaussian Splatting como preferentes para la tarea asignada, aunque se siguen poniendo a prueba las prestaciones de ambos.

Generación e importación de *datasets* propios

La siguiente fase del proyecto consistió en la generación e importación de conjuntos de datos sencillos propios con el objetivo de evaluar cómo se comportaban los modelos frente a conjuntos de imágenes de menor calidad. Esta decisión se tomó en previsión de las difíciles condiciones de toma de imágenes que presenta un *rover* en una misión de exploración planetaria, donde las cámaras pueden enfrentarse a limitaciones severas en cuanto a resolución, estabilidad y condiciones de iluminación. Además, este paso también tenía como finalidad estudiar el comportamiento y la robustez de los RFM ante distintos tipos de trayectorias y disposiciones de cámara, algo fundamental si se pretende aplicar estos métodos a escenarios reales fuera de laboratorio.

Para la generación de estos archivos se utilizó la cámara integrada en un teléfono Xiaomi Redmi Note 11, un dispositivo de gama media cuyas prestaciones fotográficas son representativas de lo que podría esperarse de cámaras embarcadas en plataformas móviles ligeras, especialmente en misiones de bajo coste. Este modelo incorpora una cámara cuádruple trasera, siendo el sensor principal un Samsung S5KJN1 de 50 MP, con apertura $f/1.8$, tamaño de píxel de $0.64\ \mu\text{m}$, y una distancia focal equivalente de 26 mm. La cámara cuenta con enfoque automático PDAF (*Phase Detection Auto Focus*) y soporta resoluciones de hasta

⁷https://docs.nerf.studio/reference/cli/ns_download_data.html

8160 x 6144 píxeles. Aunque se trata de un sensor relativamente potente en términos de resolución, su rendimiento en condiciones de baja iluminación es limitado, lo que lo convierte en un candidato adecuado para poner a prueba la robustez de los métodos de reconstrucción bajo condiciones adversas. La cámara también permite grabación de vídeo en 1080p a 30/60 fps, lo cual resultó útil para capturar trayectorias continuas de cámara y generar conjuntos de imágenes por muestreo de fotogramas.

Se realizaron múltiples sesiones de toma de imágenes y grabaciones de vídeo de distintos objetos y lugares, incluyendo tanto escenas interiores como exteriores, bajo distintas condiciones de iluminación y desde diferentes distancias y ángulos. Esto permitió no solo analizar cómo afectan las condiciones físicas de captura a la calidad de reconstrucción, sino también experimentar con diversas trayectorias (laterales, frontales, circulares, irregulares) para valorar el impacto de la geometría del recorrido en la coherencia espacial de las reconstrucciones generadas. Las fases específicas de este proceso, junto con las observaciones realizadas en cada una, se detallan a continuación :

Objetos sencillos. Para empezar, se procedió con la toma de imágenes de objetos sencillos como botellas desde numerosas perspectivas, procurando rodear toda su silueta y capturar vistas desde todas las alturas posibles. Como era de esperar, a pesar de proceder de un conjunto de apenas 40 fotografías, el proceso en este caso fue exitoso por la alta cobertura de vistas, las condiciones lumínicas controladas y favorables y la simplicidad de la representación (Fig 14).



Figura 14: Vistas diversas circundantes a un objeto sencillo observado en tiempo real en el visor de Nerfstudio apenas segundos después de iniciar el entrenamiento del modelo Nerfacto y sin que este hubiera terminado. Como se ve en (a), durante el entrenamiento se puede observar la pose de las fotografías de partida en la GUI de Nerfstudio.

Entornos de interior: escenas circulares cóncavas y convexas. El siguiente hito en la investigación fue la reconstrucción de espacios de interior, aumentando la superficie de captura. Con este fin, se tomaron fotografías del laboratorio de robótica espacial. capturadas primero realizando un desplazamiento lateral paralelo a la pared y posteriormente tomando trayectorias circulares cóncavas y convexas. En esta etapa, se comprobó cómo las reconstrucciones de trayectorias cóncavas presentaban calidades inferiores. Posteriormente se determinó que esto era debido al menor número de puntos comunes entre imágenes sucesivas, lo que hacía necesario que el solapamiento fuera mayor. En este punto de la investigación se tomó conciencia de la importancia del adecuado solapamiento y cobertura de vistas diversas entre las imágenes de los *datasets*, suponiendo una de las principales limitaciones a lo largo de todo el proceso (Fig 15).



Figura 15: Vistas de una curva cóncava a lo largo de la esquina de un espacio de interior durante el entrenamiento del modelo *Splatfacto-W*. Son visibles las poses de las fotografías utilizadas en el proceso, así como la baja calidad del resultado, consecuencia de una cobertura y solapamiento insuficientes. Esta etapa de la experimentación permitió concluir que el proceso de toma de imágenes tiene igual o mayor importancia que el propio algoritmo de reconstrucción para la generación de modelos mediante reconstrucción fotogramétrica mediante RFM.

Entornos de exterior. A continuación, una vez comprobadas las capacidades del entorno utilizado para la reconstrucción de escenas complejas, se ha realizado trabajo de campo en los alrededores de la Escuela de Ingenierías Industriales de la Universidad de Málaga como el LAENTIEC⁸. Se han realizado tomas de trayectorias frontales, laterales y circulares, con resultados diversos. De esta fase se toman las siguientes conclusiones:

En relación a la calidad de la reconstrucción en función de las trayectorias seguidas durante la toma de imágenes, las más convenientes resultan ser aquellas de tipo circular convexo, enfocando el centro de la circunferencia. Este tipo de trayectorias permite mayor solapamiento y una mejor cobertura de la escena que ningún otro desplazamiento. Sin embargo, se trata de un patrón que no resulta operativo para misiones de robótica espacial de exploración, donde se espera que el *rover* cubra la mayor superficie posible en un tiempo razonable. Por tanto, este tipo de desplazamiento queda descartado salvo para la reconstrucción tridimensional de objetos de especial interés.

Por otro lado, las trayectorias laterales ofrecen resultados comparativamente superiores. Aun así, con frecuencia es necesario realizar múltiples pasadas a diferentes alturas y distancias del foco para aportar suficiente información tridimensional y lograr una cobertura adecuada de la escena. No obstante, este tipo de trayectorias pueden ser valiosas como información complementaria, especialmente considerando que son perfectamente compatibles con un desplazamiento regular como el que sigue el *rover* en sus misiones.

En contraste, los desplazamientos frontales presentan una dificultad añadida: la proyección 2D de sus características se modifica mucho más visiblemente entre tomas que en las trayectorias alternativas, en las que el desplazamiento del sensor es correlativo al de la imagen. En los desplazamientos frontales, las variaciones de profundidad alteran de manera significativa la distribución de las características sobre la proyección 2D. Por este motivo, este formato requiere el mayor solapamiento de todas las alternativas existentes para asegurar una reconstrucción aceptable.

Cabe destacar también que los modelos base utilizados hasta el momento, concretamente

⁸<https://www.uma.es/robotics-and-mechatronics/cms/menu/robotica-y-mecatronica/area-de-experimentacion/>

nerfacto y splatfacto, empiezan a mostrar carencias importantes, derivadas de su especialización en entornos limitados y controlados. A partir de este punto, se decide sustituir estos métodos por NeRF in the Wild y Splatfacto in the Wild, modelos específicamente diseñados para escenas de gran escala al aire libre y condiciones ambientales difíciles de controlar.

Durante esta fase también se pone de manifiesto la importancia de la calidad de las imágenes tomadas. En sucesivas jornadas de trabajo de campo se ha podido comprobar que la probabilidad de obtener una reconstrucción exitosa es directamente proporcional a la nitidez y la iluminación de las imágenes, especialmente en la fase previa de extracción de poses mediante SfM. Este hecho se explica en gran medida por la relación directa entre la calidad y consistencia de las imágenes y la fiabilidad de las características extraídas de ellas, necesarias para una correcta estimación de la pose. Como es de esperar, también en la fase de entrenamiento del modelo 3D influyen de forma significativa las condiciones de iluminación, aunque se intenta paliar este efecto mediante la elección de métodos especialmente robustos ante este tipo de alteraciones.

Primeras pruebas con imágenes de BASEPROD Finalmente, se han utilizado las fotografías de una de las travesías que conforman el conjunto de datos BASEPROD, cuya reconstrucción constituye el núcleo de este proyecto. A partir de este punto, Gaussian Splatting se vuelve una alternativa cada vez mejor frente a Nerf, cuyos tiempos de proceso son progresivamente mayores debido al creciente número de fotografías necesarias. Sorprende la elevada calidad de la reconstrucción y la magnitud de la superficie reconstruida para el poco tiempo de procesamiento que requiere el Gaussian Splatting, debiéndose ello en gran parte al notable solapamiento entre imágenes sucesivas. Pese a ello, es necesario tener en cuenta que el preprocesamiento para obtener las características y los puntos 3D es muy elevado, suponiendo el principal cuello de botella que hace imposible la implementación en tiempo real para esta disposición.

Dado este último dato y teniendo en cuenta que el solapamiento entre imágenes supera con holgura el necesario para realizar la reconstrucción de manera correcta, se decide disminuir la frecuencia de las tomas.

Se sabe que la velocidad del *rover* MaRTA es de ca. 2 a 10 cm/s, por lo que con el periodo estándar de un segundo entre tomas sucesivas se tiene un espacio aproximado de 5 cm entre imágenes sucesivas. Se decide realizar pruebas en la siguiente fase de la investigación aumentando las distancias entre imágenes y estimando la frecuencia adecuada para reducir el tiempo de computación manteniendo un adecuado solapamiento entre las imágenes.

Tras realizar estas comprobaciones, se confirma la operatividad del sistema para reconstruir el entorno capturado por el robot, asimilable a un entorno extraterrestre, y se decide diseñar un programa capaz de extraer el resto de fotografías y datos, realizar las adaptaciones necesarias e implementarlas en las reconstrucciones.

3.3. Diseño del software para extracción y manipulación de datos de los *rosbags*

Pese a tener disponibles las imágenes de algunas de las travesías realizadas por el *rover*, la mayoría se encuentran contenidas en los *rosbags*, por lo que se ha desarrollado un software para la interacción con los nodos (procesos que ejecutan una función específica dentro del sistema del robot) y *topics* (canales de comunicación mediante los cuales los nodos publican y reciben mensajes) contenidos en ellos y la posterior extracción de los datos y mensajes relevantes para este proyecto. Este software está compuesto por varios *scripts* que permiten

explorar el contenido de los *rosbags*, extraer información de sensores, sincronizar los datos con las imágenes y almacenarlos en un formato que pueda ser utilizado por las herramientas de reconstrucción 3D.

Estudio de la estructura interna del *rosbag*

El primer paso necesario ha sido la comprobación de los mensajes disponibles en el *rosbag* y su disposición para poder seleccionar cuáles de ellos se deseaban copiar para su exportación, para lo que se ha diseñado un sencillo *script* denominado “`inspect_rosbag.py`”.

Este *script* emplea la librería **rosbag2_py** para listar los *topics* registrados dentro del *rosbag*, junto con información sobre su tipo de mensaje y número de instancias almacenadas. Gracias a esto, se ha podido analizar la estructura interna de los archivos, identificando qué *topics* contienen imágenes, datos de navegación (GNSS), orientación (IMU) y parámetros de la cámara. Este análisis preliminar ha permitido definir la estrategia óptima para la extracción de datos.

Extracción de datos del *rover*

El siguiente paso fue el diseño de un *script* llamado “`extract_data.py`” que se encarga de extraer y sincronizar datos de un archivo *rosbag* de ROS 2, combinando imágenes, datos GNSS (posición) y datos de orientación (giroscopio), junto con la información intrínseca de la cámara. Utiliza la librería **rcipy** para inicializar un nodo y **rosbag2_py** para leer secuencialmente los mensajes almacenados en el bag.

Dentro de la clase **DataExtractor** se configuran búferes para almacenar temporalmente los datos GNSS y de orientación, y se definen funciones auxiliares para convertir las marcas temporales o *timestamp* en diccionarios con segundos y milisegundos, transformar los arreglos (*arrays*) en listas y buscar en los búferes el dato cuya marca sea la más cercana al instante de cada imagen. Esto es clave para sincronizar la información de distintos sensores con las imágenes, que se convierten a formato OpenCV mediante **CvBridge** y se guardan en un directorio de salida.

Durante la ejecución, el *script* identifica y procesa mensajes de cuatro *topics* específicos: imágenes, GNSS, orientación y datos de la cámara. Cada imagen se guarda como un archivo PNG (aunque en una etapa posterior del desarrollo se ha sustituido este formato por el JPEG para poder integrar la información GNSS como metadatos) y se establece la asociación con los datos GNSS y de orientación más cercanos en el tiempo; además, se puede aplicar un *offset* en el ángulo de guiñada a la orientación para corregir el desfase respecto al norte producido por el giroscopio, que asigna su referencia en la dirección al momento de ser encendido. Finalmente, toda la información (nombre de la imagen, marca temporal, posición, orientación ajustada y parámetros de la cámara) se guarda en un archivo JSON, facilitando su posterior uso y procesamiento para poder ser utilizada por Nerfstudio.

3.4. Reconstrucciones a partir de datos de los *rosbags*

A continuación, se ha procedido a la realización de modelos de algunas travesías seleccionadas. Para ello se ha escogido Splatfacto-W como método por su independencia de las condiciones de luminosidad, su tolerancia a las grandes reconstrucciones y su velocidad. Se ha medido el tiempo de reconstrucción de los modelos, el número de imágenes empleadas y

comprobado su precisión. Algunas imágenes de las reconstrucciones, así como los resultados numéricos se pueden encontrar en la sección 4

Se comprueba cómo para la mayoría de los *rosbags*, el procesado previo supone un cuello de botella, aumentando significativamente cuando no impidiendo el proceso. Se observa cómo en las reconstrucciones el algoritmo `ns-process-data` no es capaz de hallar las poses, por lo que al no disponer de datos de entrada no es posible la reconstrucción. Por ello, se plantean dos alternativas: o bien evitar el algoritmo de SfM y suministrar directamente los datos a partir de los sensores del *rover* o bien implementar mejoras en el preprocesamiento en aras de una mayor velocidad y robustez.

3.5. Valoración de la viabilidad de la realización de reconstrucciones sin SfM previo: programación de software adaptador de formato

En todo el proceso, se constata que, pese a la velocidad y eficacia mostrada por el modelo de Gaussian Splatting seleccionado, requiere de las poses de las cámaras y, casi obligatoriamente, de información tridimensional dispersa previa para su correcto funcionamiento. En el caso que nos ocupa, se obtienen estos requisitos mediante el software de SfM COLMAP, que utiliza un tiempo inasumible en tareas de tiempo real y poco operativo para otro tipo de misiones. Por ello, se plantea la posibilidad de utilizar los datos de posición y orientación de los sensores del *rover* como alimentación del modelo.

El objetivo de este planteamiento es agilizar el procesamiento evitando el cálculo de poses mediante SfM y, en su lugar, aprovechar directamente los datos de los sensores del *rover*. Esto haría que las reconstrucciones fueran mucho más rápidas y, en teoría, facilitaría su uso en robots autónomos que exploran entornos desconocidos. Sin embargo, esta alternativa no está exenta de desafíos, y es importante evaluar sus limitaciones antes de adoptarla.

Fundamentalmente, el enfoque presenta dos inconvenientes: por un lado, el GNSS no es una tecnología disponible en las misiones espaciales reales debido a la ausencia de los satélites de los que depende. Además, presenta unos márgenes de error elevados, de igual manera que los datos de la mayoría de los sensores. Pese a ello, se comprobará si es operativo, pudiendo ser útil tanto en misiones y experimentos terrestres como como aproximación a otros sistemas.

Desarrollo del software adaptador de formato

La información a utilizar está contenida en los archivos `.txt` generados con el código de extracción de datos de ROS que se ha diseñado para este proyecto. Cada uno de estos archivos almacena información sobre la imagen capturada, su instante asociado, la posición GNSS del *rover* en ese instante y la orientación determinada por la unidad inercial. Sin embargo, estos datos deben transformarse en un formato compatible con Nerfstudio para ser procesados adecuadamente.

Para su implementación en el entorno de Nerfstudio, se ha desarrollado un *script* que genera a partir de estos archivos un archivo `“transforms.json”` con el formato que la plataforma utiliza (`convert_outputs_to_transforms.py`). Este JSON incluye las siguientes secciones clave:

- **camera_frames**: una lista de todas las imágenes junto con sus posiciones y orientaciones en coordenadas homogéneas.

- **intrinsics:** parámetros de la cámara, como la distancia focal y el tamaño del sensor, extraídos de los metadatos.
- **frames:** cada imagen es asociada con su transformación en el espacio 3D, obtenida directamente de los datos de los sensores del *rover*.

El *script* ha sido diseñado para permitir la interpolación de datos en caso de discrepancias en las marcas temporales exactas y para corregir valores atípicos en los datos GNSS mediante un filtrado basado en umbrales de variación de posición y orientación.

Tras utilizar el archivo `transforms.json`, se comprueba que la reconstrucción es fallida, siendo similar al *ground truth* únicamente en puntos y orientaciones muy cercanas a las de las imágenes reales y llenándose de *splats* borrosos en el resto del espacio tridimensional. Ello se debe a que sin la información 3D real aportada por los puntos de muestreo obtenidos durante el preprocesamiento, Nerfstudio los distribuye de forma aleatoria a lo largo del espacio sin referencias, intentando después refinarlos y desplazarlos a sus posiciones reales. La convergencia en este caso es mucho más difícil y de menor calidad.

Constatado este hecho, se decide adaptar el preprocesamiento mediante SfM con el objetivo de seguir utilizándolo, pero optimizándolo previamente en términos de velocidad, precisión y robustez.

3.6. Mejoras en el proceso de Structure from Motion

Una vez descartada la posibilidad de realizar las reconstrucciones tridimensionales mediante RFM sin procesamiento previo de los datos de entrada mediante ajuste de haz en COLMAP, se decidió optimizar el flujo de trabajo de este último software con el objetivo de adaptarlo a las características concretas de los datos obtenidos en misiones de exploración planetaria, los cuales en general se capturan de forma periódica y secuencial a lo largo de la trayectoria seguida por el robot. Esta optimización se llevó a cabo en dos etapas diferenciadas: primero, se ajustaron los métodos de emparejamiento (*feature matching*) y los parámetros de COLMAP para mejorar el rendimiento y, posteriormente, se implementaron las poses GNSS como *priors* (pose inicial conocida, en este trabajo también denominados “previos”) en las imágenes, lo que permitió refinar la precisión y la robustez de la reconstrucción tridimensional.

Ajuste del método de emparejamiento (*Feature Matching*)

En la primera etapa de la optimización, se centró la atención en el proceso de emparejamiento de características visuales, una parte fundamental del mecanismo de SfM. Este proceso consiste en identificar puntos en común entre imágenes sucesivas para calcular la pose de las cámaras y, en última instancia, reconstruir la estructura tridimensional del entorno observado. COLMAP ofrece distintos métodos para llevar a cabo este emparejamiento, cada uno con sus ventajas y desventajas.

El primero de estos métodos es el *exhaustive matching* o emparejamiento exhaustivo, descrito en trabajos como [37] que compara cada imagen con todas las demás del conjunto de datos. Este enfoque es robusto y garantiza un alto grado de solapamiento entre imágenes, lo que puede mejorar la precisión de la reconstrucción. Sin embargo, su complejidad computacional es prohibitiva cuando se trabaja con grandes conjuntos de imágenes, ya que el tiempo de procesamiento crece cuadráticamente con el número de imágenes procesadas. Otro

método disponible es el *vocab tree matching* o emparejamiento mediante árboles de vocabulario [38], que utiliza estructuras jerárquicas conocidas como *vocabulary trees* para agrupar características visuales similares y acelerar el proceso de emparejamiento. Este enfoque, que supone el método por defecto en la versión de COLMAP implementada en Nerfstudio, es especialmente eficiente en situaciones donde se dispone de grandes volúmenes de imágenes desordenadas o tomadas en momentos no consecutivos. No obstante, su rendimiento puede verse afectado en casos en los que las imágenes siguen una secuencia temporal y espacial continua, como ocurre en nuestro caso de aplicación y de manera general en aplicaciones de robótica móvil.

El tercer enfoque, y el que finalmente se seleccionó para este proyecto en esta etapa de desarrollo, es el *sequential matching* o emparejamiento secuencial. Este método parte del supuesto de que las imágenes fueron capturadas en una secuencia ordenada y continua, lo que es habitual en misiones de exploración planetaria, donde un *rover* o robot móvil sigue trayectorias definidas. Al emparejar cada imagen únicamente con un número limitado de imágenes adyacentes en la secuencia, este enfoque reduce drásticamente el número de comparaciones necesarias y mantiene un alto grado de precisión siempre que la secuencia de imágenes sea consistente. Presenta como principal inconveniente su dependencia del solapamiento de las imágenes, lo que en ocasiones puede resultar conflictivo para los datos procedentes de ROS 2 debido a los frecuentes saltos que pueden presentar por causas variadas como frecuencia de muestreo inconsistente, pérdida de mensajes por colas pequeñas o calidad de servicio inadecuado y desincronización de relojes entre nodos. Pese a ello, tras evaluar estas opciones y considerando las características del *dataset* BASEPROD, se optó por el emparejamiento secuencial debido a su eficiencia computacional y a su adecuación a las trayectorias regulares del *rover* en las Bárdenas Reales. Además, se ajustaron parámetros específicos en COLMAP, como el número de imágenes adyacentes a comparar y los umbrales de detección de características, con el objetivo de maximizar el rendimiento en función de los datos disponibles.

Implementación de poses GNSS como previos

Tras la optimización inicial del método de emparejamiento, se procedió a implementar las poses GNSS como *priors* para mejorar aún más la precisión y robustez de la reconstrucción. Este cambio se justificó por varias razones. En primer lugar, el método de *sequential matching*, si bien es eficiente, puede fallar en trayectorias con discontinuidades o interrupciones, algo que ocurre ocasionalmente en misiones reales debido a la pérdida de imágenes o a cambios bruscos en la dirección del *rover*. En segundo lugar, las imágenes registradas en BASEPROD contenían información temporal precisa que podía ser utilizada para emparejarlas con datos GNSS registrados en paralelo. Esto permitió calcular la posición aproximada de cada cámara en el espacio y utilizar dicha información como previos en el proceso de reconstrucción.

El proceso de implementación de las poses como información de partida se desarrolló en varios pasos. El primer paso consistió en convertir las imágenes del formato PNG original al formato JPG, ya que este último facilita la inserción de metadatos EXIF, donde se almacenaría la información de posición GNSS. Para llevar a cabo esta conversión, se desarrolló un *script* específico que procesó todas las imágenes del conjunto de datos y las transformó a JPG, conservando la calidad original.

Una vez convertidas las imágenes, el siguiente paso fue emparejar cada imagen con los datos GNSS correspondientes mediante las marcas de tiempo (*timestamps*). Para ello, se desarrolló un segundo *script* que importaba los datos de posición y los añadía como metada-

tos EXIF a las imágenes correspondientes. Este emparejamiento temporal aseguró que cada imagen recibiera la posición GNSS más cercana en el tiempo, lo que mejoró la coherencia espacial del modelo resultante.

Finalmente, en lugar de realizar la reconstrucción desde cero, se empleó el mapeador con poses iniciales conocidas o **Pose Prior Mapper** de COLMAP, que ajusta las poses iniciales estimadas a partir de los datos GNSS y refina posteriormente la reconstrucción mediante ajuste por haces. Este enfoque permitió reducir los errores acumulativos en la estimación de poses y mejorar la precisión global del modelo tridimensional.

3.7. Optimización del flujo de trabajo

Automatización del proceso mediante *scripts* de terminal

Tras definir, diseñar y configurar cada parte del proceso, se ha procedido a su automatización mediante *scripts* de terminal, que ejecutan los comandos de forma consistente y autónoma y agrupa partes extensas del proceso en comandos individuales. Con esto se consigue reducir la intervención manual del operador y estandarizar los parámetros y pasos del algoritmo, aumentando la reproducibilidad, escalabilidad y eficiencia del proceso.

La estructura general del proceso automatizado, que se muestra esquematizado en 16 para una mejor comprensión, va definida por los siguientes *scripts*:

- **run_extract_data.sh:** Su finalidad es configurar y regular la extracción de los datos de los *rosbags* mediante el *script* `extract_data.py` y su preparación para ser procesados y utilizados para la generación de modelos nerf y Gaussian Splatting. Para ello, primero se seleccionan las rutas de los *rosbags* y del directorio del proyecto a crear. A continuación se da la opción de explorar dicho *rosvbag* para comprobar los *topics* relevantes mediante una llamada al *script* propio `inspect_rosvbag.py`. Posteriormente, se ajustan parámetros diversos como los nombres de los *topics* relevantes en el *rosvbag*, el máximo de imágenes cuya extracción se permite, el *offset* añadido a la orientación y los instantes de entrada y salida. Finalmente, se crean las carpetas en las que se distribuirá de forma ordenada toda la información, se extraen los datos solicitados, se genera un archivo .json con los datos más importantes mediante una llamada al *script* de python con la configuración seleccionada y se importa la información del GNSS como metadatos EXIF en cada imagen para su uso como previos.
- **run_colmap_process.sh:** como su nombre indica, este *script* coordina la ejecución de todo el preprocesamiento de las imágenes de forma externa a Nerfstudio mediante COLMAP. Este modo de hacerlo permite trabajar con una versión superior de COLMAP y añade funcionalidades clave como la implementación como poses iniciales conocidas de la información de los sensores de posición.

Su funcionamiento es el siguiente: tras seleccionar el método de emparejamiento de las imágenes y la ruta de la carpeta de proyecto (la misma en la que extrajimos los datos de los *rosbags*, se crean las carpetas y los archivos auxiliares requeridos por COLMAP (como el archivo database necesario para el almacenamiento de características e información diversa). Después, se extraen las características visuales de las imágenes y los metadatos de posición GNSS en caso de existir, se realiza el emparejamiento con el método seleccionado (este algoritmo permite los enfoques exhaustivo y secuencial por ser los utilizados en el presente trabajo, pero no se descarta la adición de otros como el emparejamiento espacial o el uso de vocab trees); y se realiza el mapeado del

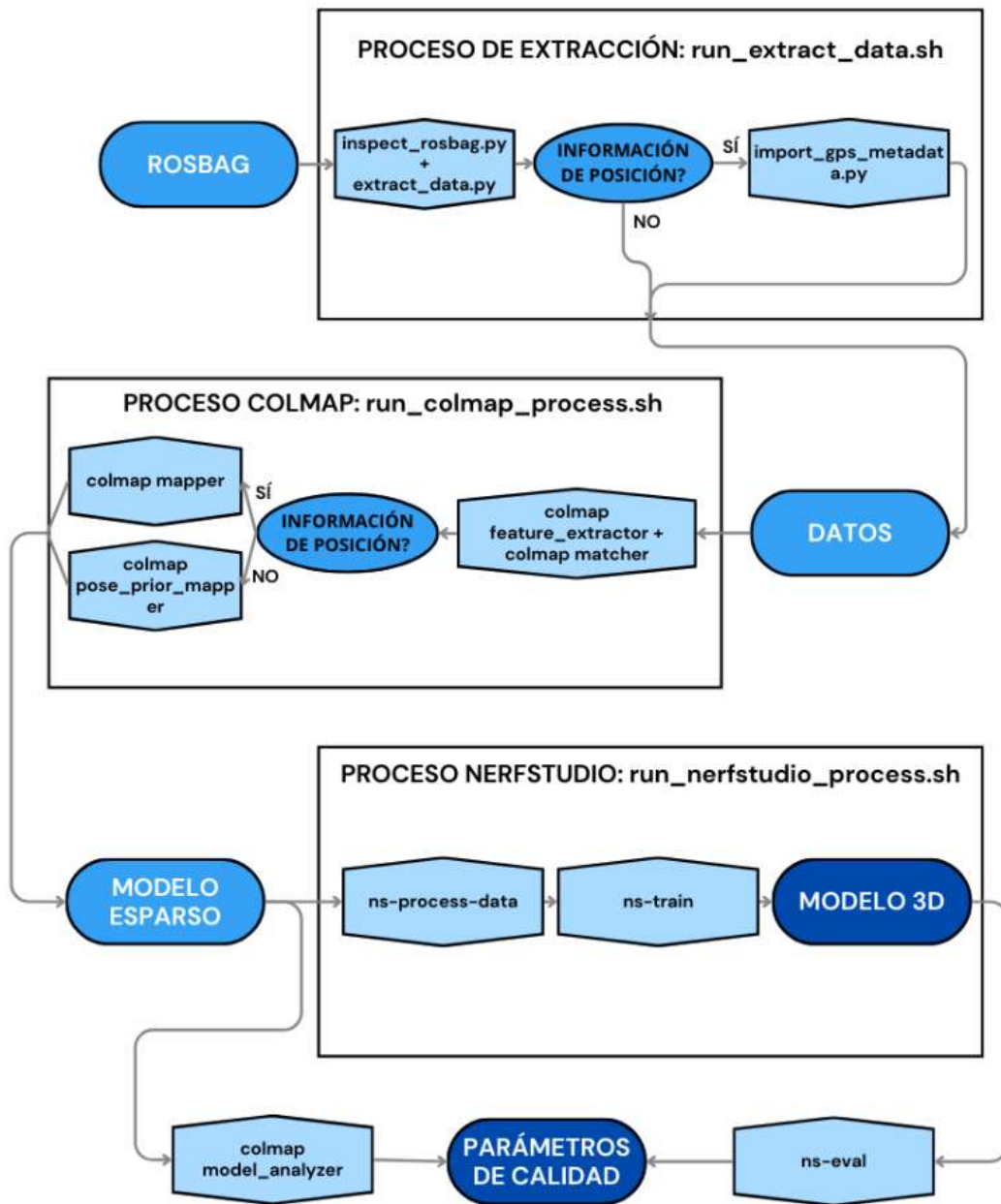


Figura 16: Esquema general del *pipeline* automatizado desarrollado para la reconstrucción 3D y localización visual a partir de datos del *rover*. Se han agrupado las tareas en tres subprocessos definidos por el tipo de datos con el que se trabaja y la tecnología de la que se depende mayoritariamente en cada fase. El flujo de trabajo automatizado completo se puede ejecutar directamente con el archivo de terminal `run_full_pipeline.sh`.

terreno mediante `Mapper` o el `Pose Prior Mapper` de COLMAP dependiendo de si existen o no metadatos de coordenadas GNSS en las imágenes. Todos estos pasos se realizan tras configurar cada una de las herramientas de COLMAP utilizadas para el uso optimizado de dichas funcionalidades. Por último, se analizan los parámetros de calidad del modelo y se visualiza el mapa en la gui para compararlo con las imágenes de partida y ver si se corresponde con el resultado esperable.

- **`run_nerfstudio_process.sh`**: este *script* constituye la última etapa del proceso y se encarga de procesar los datos generados por COLMAP y extraídos de los *rosbags* para su uso con Nerfstudio. Su objetivo principal es automatizar la generación de reconstrucciones visuales realistas mediante modelos tipo NeRF y Gaussian Splatting, basándose en las poses calculadas previamente y evitando el uso de herramientas redundantes o menos optimizadas.

Para ello, el *script* toma como entrada las rutas a la carpeta de imágenes, al directorio de salida y a la carpeta que contiene los modelos dispersos de COLMAP. Si dicha reconstrucción ya existe, se solicita al usuario que introduzca el número del modelo que desea emplear, se verifica su validez (comprobando que contiene los binarios requeridos como `cameras.bin`) y se invoca el comando `ns-process-data` con la opción `-skip-colmap`, que omite la reconstrucción automática y reutiliza el modelo disperso externo ya calculado, acelerando el flujo de trabajo y reduciendo errores.

Si no se encuentra una reconstrucción válida, el *script* recurre al algoritmo interno de Nerfstudio para ejecutar su propio proceso de reconstrucción desde cero. En ambos casos, los datos se almacenan en una carpeta de salida estructurada y lista para ser usada en el entrenamiento de modelos.

Una vez finalizada la preparación de los datos, el *script* ejecuta el entrenamiento del modelo Splatfacto-W, que ofrece una reconstrucción densa basada en Gaussian Splatting. Finalmente, se lanza una evaluación automática de la reconstrucción generada para comprobar su calidad, guardar métricas relevantes y verificar visualmente la fidelidad del modelo respecto al entorno original capturado.

- **`run_full_pipeline.sh`**: Este *script* actúa como coordinador del proceso completo, agrupando todas las fases del algoritmo de reconstrucción 3D a partir de los datos de los *rosbags* —inspección de *rosbags*, extracción de datos, reconstrucción con COLMAP y entrenamiento con Nerfstudio— en una única ejecución automatizada y guiada. Su principal objetivo es maximizar la reproducibilidad y reducir al mínimo la intervención manual, permitiendo llevar a cabo el flujo de trabajo completo con un solo comando.

El funcionamiento se estructura en tres bloques principales. En primer lugar, se solicitan al usuario los parámetros esenciales como la ruta al archivo del *rosbag*, la ubicación de los directorios de datos crudos y procesados, el número máximo de imágenes a extraer, el periodo entre capturas y el método de emparejamiento. Estos valores pueden también definirse mediante argumentos en línea, permitiendo su uso desatendido o en entornos automatizados. A continuación, se ejecuta el *script* `run_extract_data.sh`, encargado de extraer y preparar los datos. En caso de fallo, se notifica al usuario y se le da la opción de continuar o abortar el procedimiento, aumentando la robustez del sistema.

Si se decide continuar, el siguiente bloque ejecuta el *script* `run_colmap_process.sh`, que gestiona la reconstrucción mediante COLMAP, utilizando los parámetros de emparejamiento seleccionados y aplicando, si existen, poses iniciales conocidas extraídas del sensor GNSS. Nuevamente, se realiza un control de errores y se informa del estado del proceso. Finalmente, si el usuario así lo desea, se activa la última fase: la conversión

de los datos al formato Nerfstudio y el entrenamiento de un modelo Gaussian Splatting mediante el *script* `run_nerfstudio_process.sh`, que automatiza tanto la conversión como el entrenamiento y la evaluación del modelo resultante.

De esta manera se hace posible la ejecución del proceso completo de principio a fin, pudiendo detenerse o evitar cualquiera de las etapas, facilitando su integración en procesos más complejos o en sistemas autónomos de reconstrucción de entornos mediante modelos 3D.

Dockerización y ejecución del proceso desde servidor

Con el fin de mejorar aún más la portabilidad y reproducibilidad del método, se ha construido una imagen docker que genera un entorno preparado para poder ejecutar el código desde cualquier dispositivo, permitiendo su uso en una amplia variedad de sistemas operativos. El contenedor resultante contiene dos entornos virtuales en los que se instalan de forma aislada respectivamente Nerfstudio y COLMAP, así como una instalación de ROS2 en la base del contenedor, además de todas las librerías complementarias requeridas por los distintos *scripts*, conteniendo así todas las herramientas necesarias para ejecutar el proceso de forma satisfactoria.

Aprovechando esta herramienta, se ha implementado el entorno de trabajo completo en los servidores del Instituto Universitario de Investigación en Ingeniería Mecatrónica y Sistemas Ciberfísicos de la Universidad de Málaga (IMECH) [39]. Estos servidores destacan por sus 4 GPU NVIDIA A6000, tarjetas gráficas de alto rendimiento especializadas en tareas como inteligencia artificial, simulaciones 3D o procesamiento de vídeo, que son de las más potentes que existen en el ámbito profesional; por lo que permiten un uso mucho más extensivo en términos de potencia computacional y memoria de la GPU. Esta herramienta permite realizar reconstrucciones más veloces, precisas y, sobre todo, partiendo de conjuntos de datos mucho mayores que los admisibles por un ordenador de sobremesa convencional. Además, el servidor presenta una memoria masiva y de gran velocidad en el acceso a datos.

El almacenamiento interno del servidor se basa en discos NVMe de última generación, con velocidades de escritura y lectura superiores a los 4 GB/s y 6.9 GB/s respectivamente, lo que garantiza una gestión extremadamente ágil de los datos durante la ejecución del proceso. Este almacenamiento se organiza mediante el sistema de volúmenes lógicos LVM, lo que proporciona flexibilidad para combinar múltiples dispositivos físicos como un solo espacio de almacenamiento lógico. Concretamente, el volumen lógico que da soporte al directorio `/home` está formado por una partición de 2 TB y un disco completo adicional de 3.5 TB, ampliables en el futuro si fuera necesario.

A su vez, se cuenta con un sistema NAS de alta capacidad, con un volumen total de 114 TB, montado en `/home/NAS`, que alberga los directorios de trabajo de los usuarios. Este dispositivo está conectado mediante una interfaz de red de 10 Gbps, lo que proporciona una velocidad media de escritura de aproximadamente 450 MB/s. A pesar de que la conexión teóricamente permitiría mayores tasas de transferencia, el límite está impuesto por el propio rendimiento del NAS. La arquitectura descrita en el anexo 7.1 permite una alta disponibilidad de los datos y una gestión eficiente incluso en flujos de trabajo intensivos.

Gracias a todo ello ha sido posible completar la reconstrucción de trayectorias en las que, para el ordenador de sobremesa con que se ha desarrollado la mayoría del proyecto, el entrenamiento se veía interrumpido por falta de memoria VRAM en la GPU; permitiendo observar las reconstrucciones en tiempo real durante el entrenamiento pero sin posibilidad de extraer y almacenar modelos definitivos.

Finalmente, se han añadido al algoritmo unificado sendos programas de terminal que coordinan la correcta configuración y ejecución automática del contenedor tanto en ordenadores de sobremesa como para servidores externos como los descritos, quedando con este paso completamente definido y estandarizado el proceso.

4. Resultados

4.1. Reconstrucciones 3D realizadas

Como parte central de este trabajo, se han reconstruido cinco modelos a partir de porciones de algunas de las travesías realizadas por el *rover* MaRTA del SRL en las Bárdenas Reales de Navarra. Se han seleccionado teniendo en cuenta su diversidad en las características visuales del terreno, tipo de trayectoria seguida y número de imágenes, entre otros; intentando que el conjunto sea lo más heterogéneo posible a pesar de su reducida extensión. El conjunto de datos completo del que se han extraído está disponible en [12]. En las siguientes tablas se encuentran identificados los tramos escogidos, así como el número de imágenes, intervalo de muestreo en número de imágenes entre datos implementados consecutivos, los instantes de entrada y salida, y el *offset* o margen que se aplica a la rotación respecto al eje vertical de forma que esté referenciada al norte. Estos parámetros son necesarios como datos de entrada en el software de extracción de datos para la correcta replicación del experimento:

Nombre	Archivo	Nº de imágenes	Intervalo
Trayectoria 1	2023-07-22_14-18-23	500	2
Trayectoria 2	2023-07-21_12-38-15	300	2
Trayectoria 3	2023-07-23_13-05-11	600	2
Trayectoria 4	2023-07-23_12-52-39	385	2
Trayectoria 5	2023-07-21_17-34-18	350	2

Tabla 5: Información general de las rutas de captura y muestreo de imágenes.

Nombre	Marca entrada	Marca salida	Offset (rad)
Trayectoria 1	1690028327	1690028967	2,165
Trayectoria 2	1689935910	1689936320	-1,8
Trayectoria 3	1690110334	1690111147	-1,1949
Trayectoria 4	1690109585	1690110247	-1,25
Trayectoria 5	1689953661	1689954222	1,82

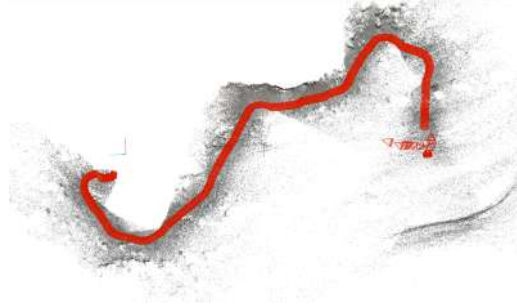
Tabla 6: Información temporal asociada a las rutas de captura.

A continuación se presenta una vista general de cada una de las reconstrucciones obtenidas, junto con una serie de comparativas que permiten evaluar de forma visual la calidad del resultado. Se incluyen, por un lado, imágenes que contrastan el desplazamiento real sobre el plano con el estimado por COLMAP, permitiendo valorar el ajuste espacial del recorrido; y por otro, pares de imágenes que enfrentan algunas capturas reales con las correspondientes vistas renderizadas desde la misma posición y orientación, facilitando así una comparación directa entre la escena real y la reconstrucción generada.

TRAYECTORIA 1



Itinerario real registrado por los sensores del rover.



Trayectoria estimada por COLMAP a partir del emparejado y la reconstrucción SfM.

Figura 17: Comparativa entre el recorrido real y el estimado por COLMAP para la trayectoria 1. La reconstrucción se corresponde con el tramo inicial, mostrado en el extremo izquierdo del mapa completo. Se aprecian algunas inexactitudes en el cálculo de la pose en el extremo final de la trayectoria.



Figura 18: Vista general del modelo generado de la trayectoria 1. Se puede ver cómo las inexactitudes en la reconstrucción dispersa del extremo final de la trayectoria afectan negativamente al resultado en dicha zona

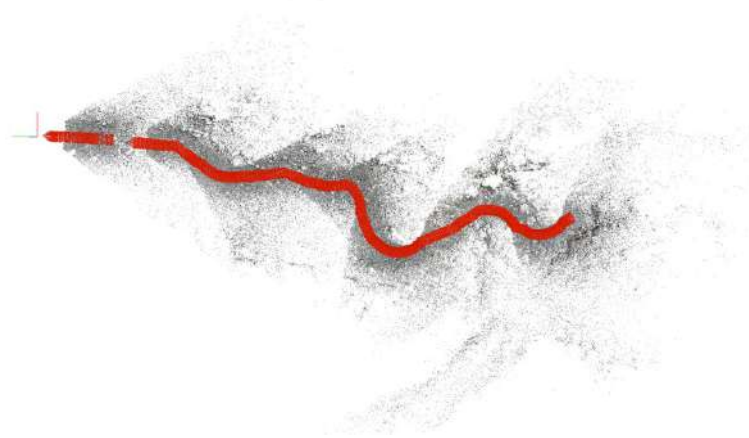


Figura 19: Comparaciones entre imágenes reales (izq.) y vistas renderizadas (dcha.) del modelo reconstruido, todas con orientación coincidente, seleccionadas para la evaluación de la trayectoria 1.

TRAYECTORIA 2



Itinerario real registrado por los sensores del rover.



Trayectoria estimada por COLMAP a partir del emparejado y la reconstrucción SfM.

Figura 20: Comparativa entre el recorrido real y el estimado por COLMAP para la trayectoria 2. La reconstrucción se corresponde con el tramo inicial, mostrado en el extremo izquierdo del mapa completo.



Figura 21: Vista general del modelo generado de la trayectoria 2.

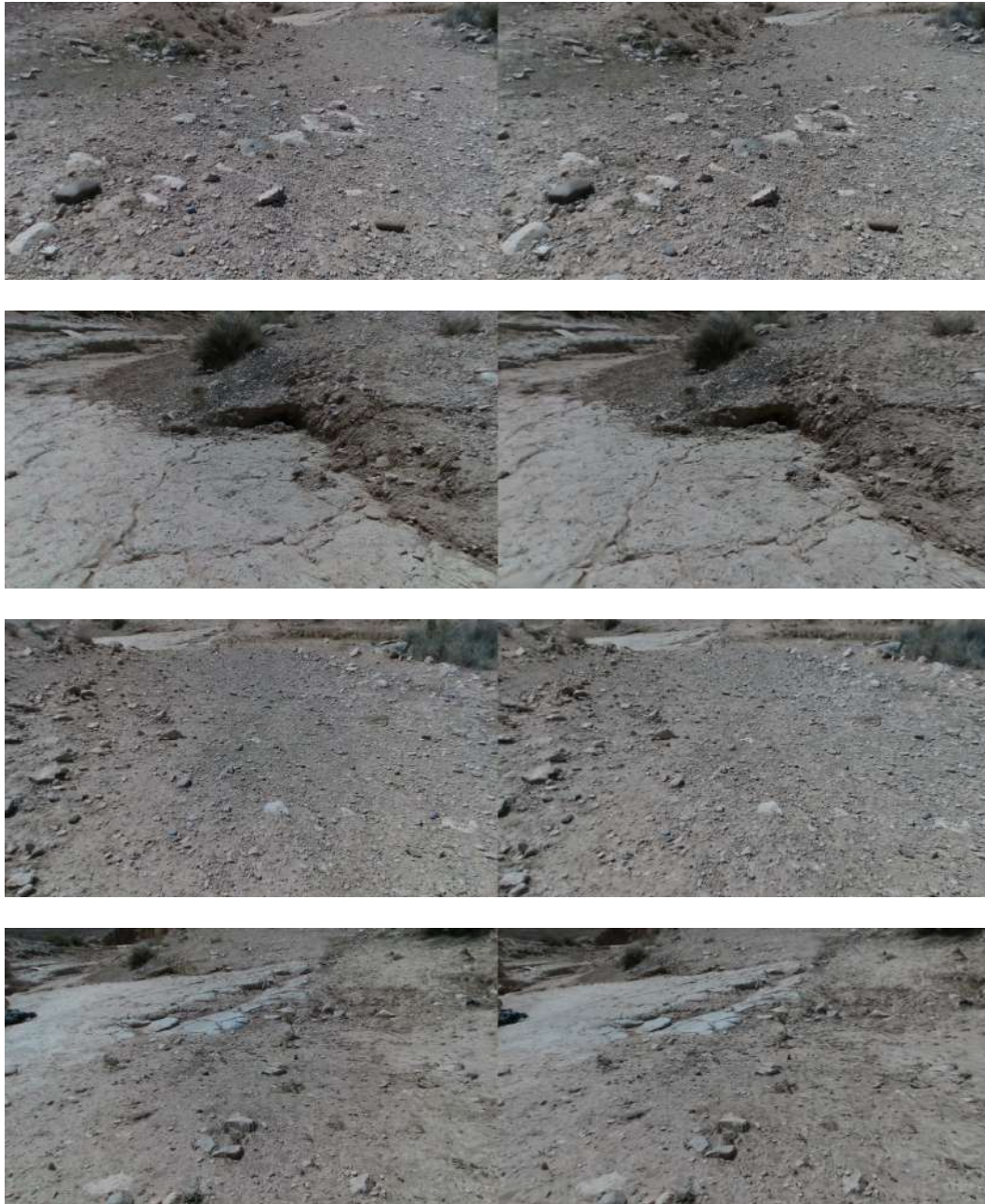
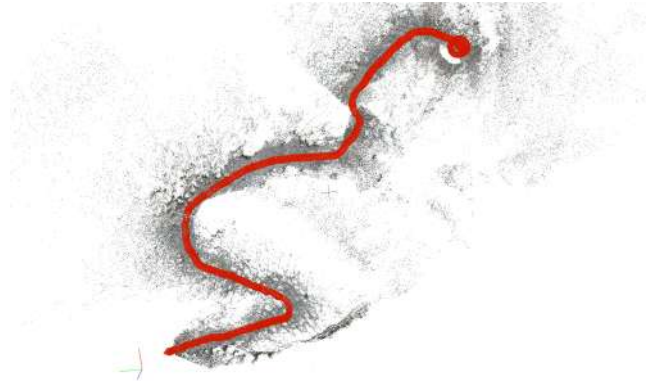


Figura 22: Comparaciones entre imágenes reales (izq.) y vistas renderizadas (dcha.) del modelo reconstruido, todas con orientación coincidente, seleccionadas para la evaluación de la trayectoria 2.

TRAYECTORIA 3



Itinerario real registrado por los sensores del rover.



Trayectoria estimada por COLMAP a partir del emparejado y la reconstrucción SfM.

Figura 23: Comparativa entre el recorrido real y el estimado por COLMAP para la trayectoria 3. El extremo inferior e izquierdo del recorrido representado se corresponde con el inicio del itinerario del robot.



Figura 24: Vista general del modelo generado de la trayectoria 3

Esta trayectoria es un buen ejemplo de la dependencia del relieve del terreno y la cobertura del espacio reconstruido en la generación de reconstrucciones. Tal y como se aprecia en la 24, se representa con gran fidelidad la pared vertical de la primera curva (lado derecho de la imagen), mientras que no existe una cobertura suficiente de la llanura a la izquierda de la trayectoria y se producen zonas borrosas e indeterminadas.

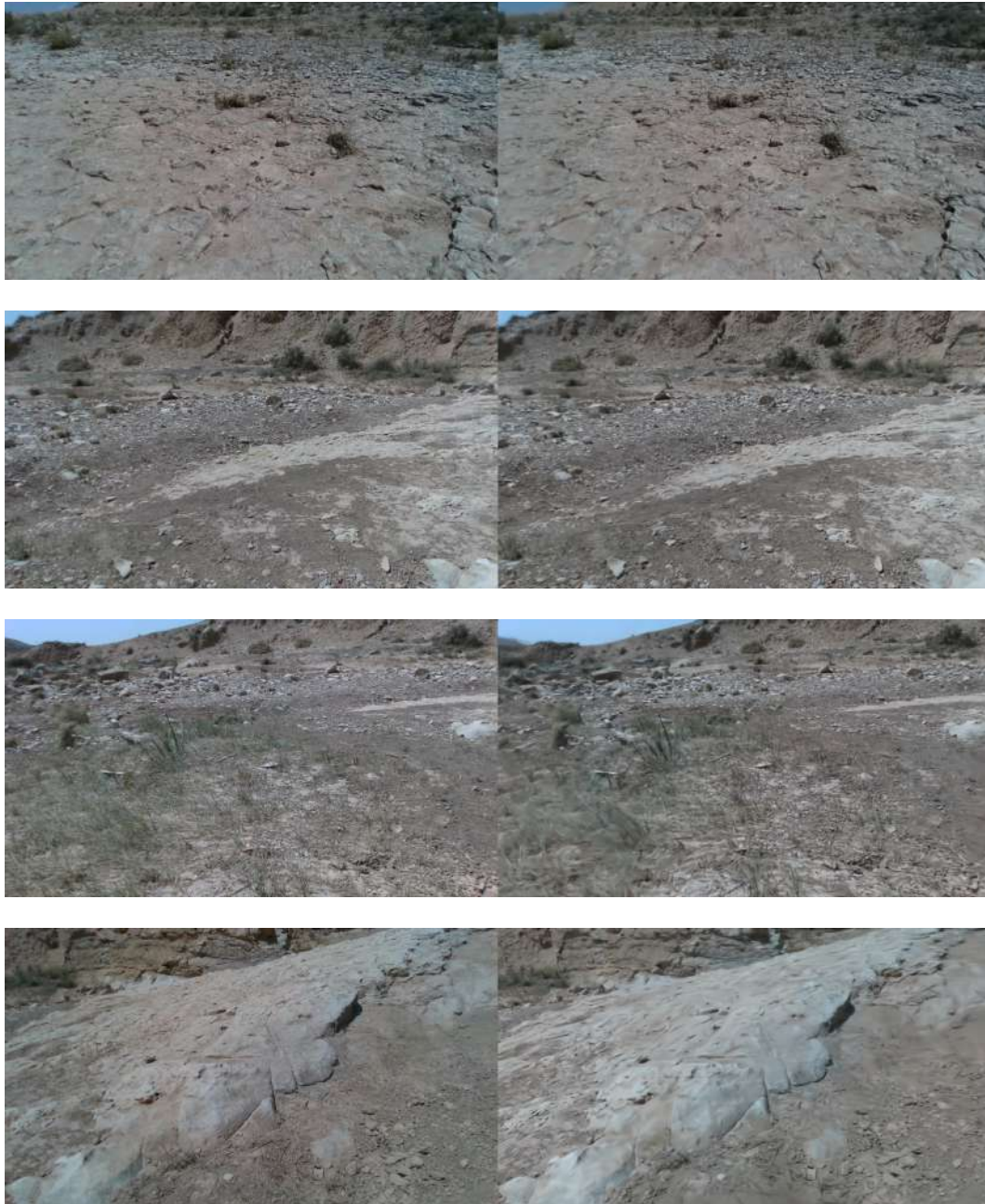
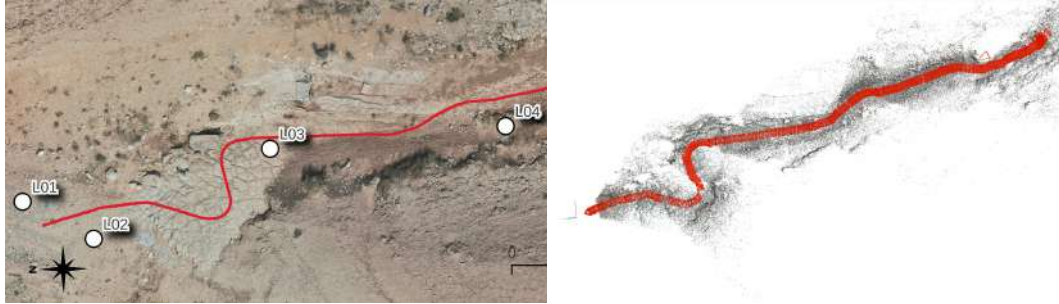


Figura 25: Comparaciones entre imágenes reales (izq.) y vistas renderizadas (dcha.) del modelo reconstruido, todas con orientación coincidente, seleccionadas para la evaluación de la trayectoria 3.

TRAYECTORIA 4



Itinerario real registrado por los sensores del rover.

Trayectoria estimada por COLMAP a partir del emparejado y la reconstrucción SfM.

Figura 26: Comparativa entre el recorrido real y el estimado por COLMAP para la trayectoria 4. El extremo izquierdo de las imágenes se corresponde con el tramo inicial del recorrido realizado por el robot.

La trayectoria 4 ha resultado ser la de peor calidad y menor precisión, probablemente debido a la presencia de algunas cámaras mal situadas en el tramo final. Pese a haber intentado corregir este hecho, la escasa calidad visual de las imágenes implicadas no ha hecho posible la correcta localización de su pose.



Figura 27: Vista general del modelo generado de la trayectoria 4

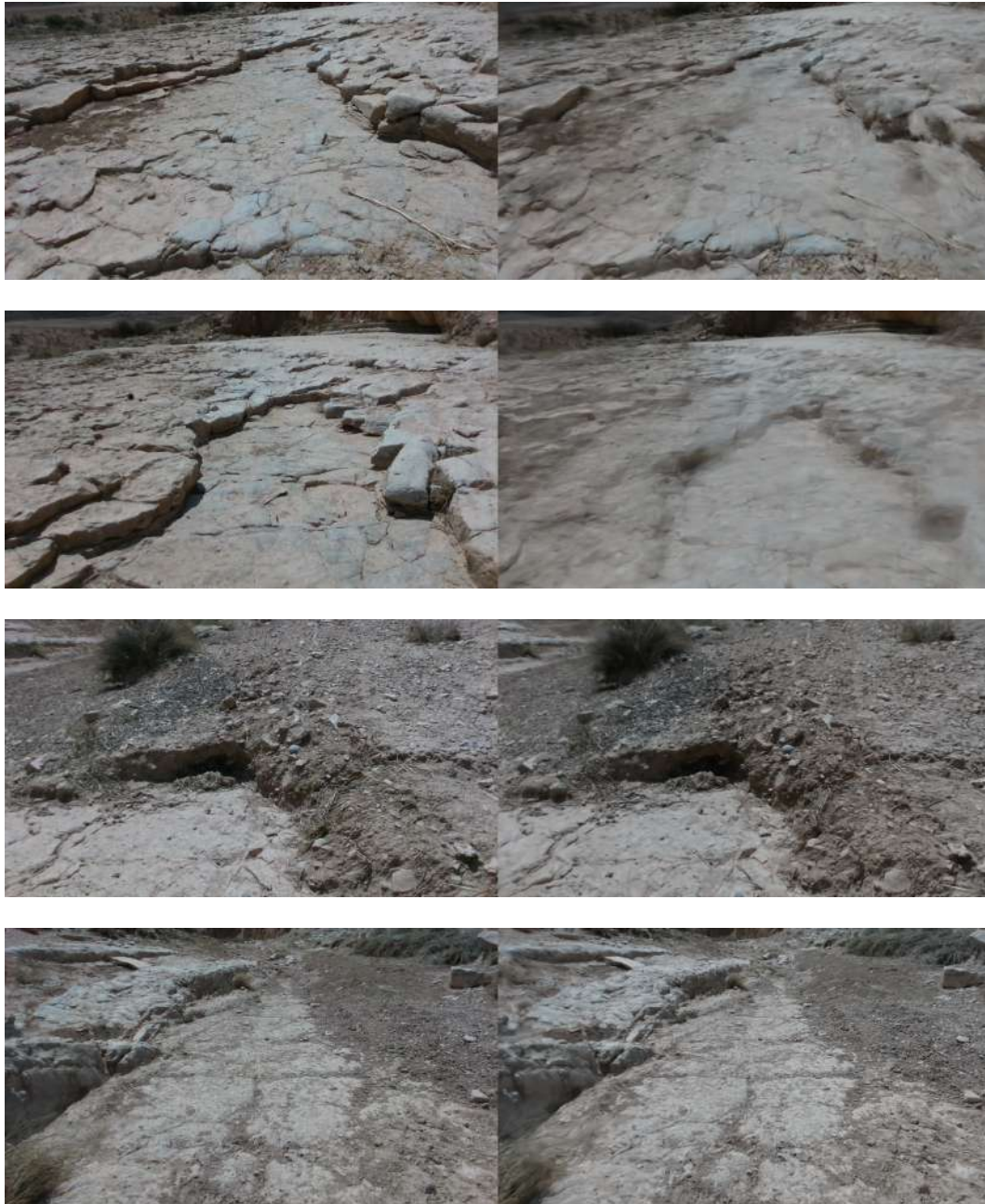


Figura 28: Comparaciones entre imágenes reales (izq.) y vistas renderizadas (dcha.) del modelo reconstruido, todas con orientación coincidente, seleccionadas para la evaluación de la trayectoria 4.

TRAYECTORIA 5

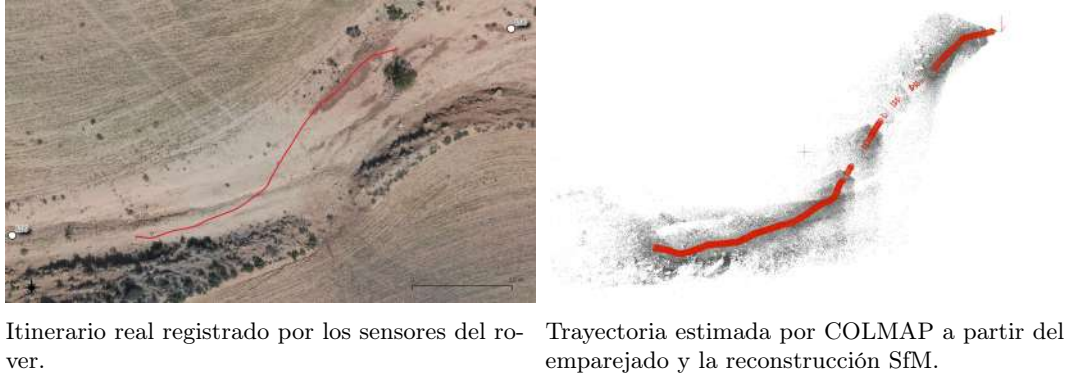


Figura 29: Comparativa entre el recorrido real y el estimado por COLMAP para la trayectoria 5.

En esta reconstrucción se puede comprobar con claridad la relación entre el solapamiento de las imágenes y la calidad de la reconstrucción. En las zonas con mejor cobertura el modelo es casi indistinguible de la realidad, mientras que se aprecian pequeños borrones en las zonas con huecos, debidos no al proceso de reconstrucción sino a discontinuidades en el proceso de captura de imágenes del *rover*. Este fenómeno es frecuente al trabajar con ROS 2, y supone un obstáculo relevante para el objeto de este trabajo que debe ser afrontado en revisiones posteriores.

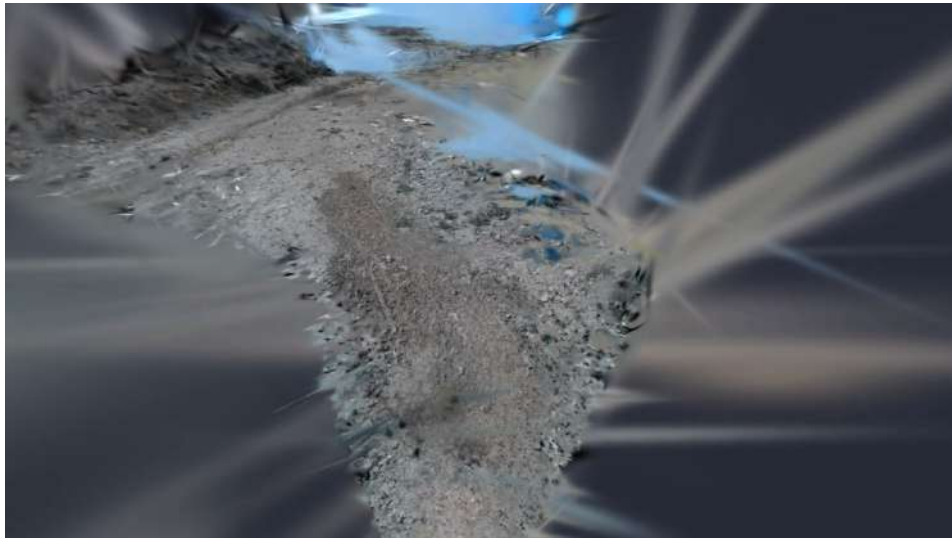


Figura 30: Vista general del modelo generado de la trayectoria 5.



Figura 31: Comparaciones entre imágenes reales (izq.) y vistas renderizadas (dcha.) del modelo reconstruido, todas con orientación coincidente, seleccionadas para la evaluación de la trayectoria 5.

4.2. Análisis de tiempo y precisión en las reconstrucciones

A continuación se aportan los siguientes parámetros evaluados para cada trayectoria en la fase de reconstrucción mediante SfM y renderizado:

- Tiempos de reconstrucción para el método de preprocesamiento por defecto de Nerfstudio.
- Tiempos de reconstrucción para nuestra implementación con emparejado secuencial.
- Tiempos de reconstrucción para el método completo con emparejado secuencial y previos de localización.
- Evaluación del proceso SfM de COLMAP: error de reproyección medio, número promedio de observaciones y la longitud media de pista *Track Length*, entendida como número promedio de imágenes en las que un punto 3D es visible y correctamente rastreado.
- Evaluación promedio de la reconstrucción generada con Splatfacto in the Wild.
- Desviación estándar de la reconstrucción generada con Splatfacto in the Wild.

En el caso del proceso de reconstrucción en Nerfstudio, no se incluye el tiempo total de reconstrucción ya que este depende directamente del número máximo de iteraciones definido en la configuración del experimento, el cual se ha mantenido en sus valores por defecto. No obstante, se ha optado por esta aproximación porque, aunque no se disponga de los datos finales al no haberse completado el entrenamiento, es posible visualizar la reconstrucción con una calidad más que aceptable tras apenas unos segundos desde el inicio, permitiendo así una visualización en tiempo real mientras el modelo continúa entrenándose en segundo plano. Del mismo modo, los parámetros de evaluación cuantitativa permiten valorar la calidad de la reconstrucción geométrica y del renderizado asociado en cada caso.

Nombre	Tiempo Matching (s)	Tiempo Mapeo (min)	Converge
TRAYECTORIA 1	6,18	68,1	SÍ
TRAYECTORIA 2	5,32	50,3	NO
TRAYECTORIA 3	13,8	143,85	SÍ
TRAYECTORIA 4	6,53	96,57	NO
TRAYECTORIA 5	6,25	73,6	NO

Tabla 7: Resultados del método de preprocesamiento integrado de Nerfstudio.

Nombre	Tiempo Matching (s)	Tiempo Mapeo (min)	Converge
TRAYECTORIA 1	0,447	56,37	SÍ
TRAYECTORIA 2	0,353	25,00	SÍ
TRAYECTORIA 3	1,2	54,564	SÍ
TRAYECTORIA 4	0,366	51,197	NO
TRAYECTORIA 5	0,297	40,88	SÍ

Tabla 8: Resultados de preprocesamiento: emparejado secuencial.

Nombre	Tiempo Matching (s)	Tiempo Mapeo (min)	Converge
TRAYECTORIA 1	0,441	26,858	SÍ
TRAYECTORIA 2	0,354	19,354	SÍ
TRAYECTORIA 3	1,2	51,00	SÍ
TRAYECTORIA 4	0,36	22,995	SÍ
TRAYECTORIA 5	0,319	25,725	SÍ

Tabla 9: Resultados de preprocesamiento con método propio completo: emparejado secuencial e implementación de previos de localización de las cámaras.

Nombre	Error de Reproy.	Nº Observaciones	Track Length
TRAYECTORIA 1	0,552	6437,783	7,070
TRAYECTORIA 2	0,548	6287,216	6,978
TRAYECTORIA 3	0,541	5594,433	7,433
TRAYECTORIA 4	0,520	6141,989	6,905
TRAYECTORIA 5	0,476	5310,430	7,918

Tabla 10: Evaluación del proceso de Structure from Motion para cada trayectoria. Valores promedio.

Nombre	PSNR (dB)	SSIM	LPIPS	PSNR std	SSIM std	LPIPS std
TRAY. 1	28,2	0,806	0,231	3,000	0,116	0,119
TRAY. 2	27,114	0,8027	0,204	1,807	0,0482	0,041
TRAY. 3	28,205	0,805	0,231	3,003	0,116	0,119
TRAY. 4	26,068	0,783	0,264	3,530	0,107	0,121
TRAY. 5	29,33	0,827	0,224	3,904	0,133	0,101

Tabla 11: Evaluación de la reconstrucción con Splatfacto in the Wild para cada trayectoria. Se muestran valores promedio y desviación estándar.

5. Discusión

5.1. Evaluación de los resultados obtenidos

Al interpretar los resultados obtenidos, es importante tener en cuenta algunas circunstancias que han condicionado tanto la calidad de los datos como los resultados de los procesos de reconstrucción y renderizado (subsección 4.2). Cabe destacar que una vez aplicadas todas las mejoras y adaptaciones desarrolladas en este trabajo, se han reconstruido con éxito todas las escenas con notables resultados, como puede comprobarse en las tablas de parámetros de calidad. Pese a ello, es necesario recordar que las imágenes y el resto de información sensorial han sido capturadas por un *rover* en movimiento en un experimento independiente, sin ser la fotogrametría su finalidad original. Debido a ello, las condiciones de iluminación y resolución son variables, y las trayectorias no están optimizadas para maximizar los ángulos de visión y la correcta cobertura de vistas. Además, han sido extraídas directamente desde un sistema ROS2, lo que implica cámaras sujetas a vibración e imágenes de calidad variable. Esto, que supone un aliciente por replicar de manera aproximada las condiciones de una misión real, conlleva también una serie de desventajas que repercuten en la interpretación de los resultados cualitativos de los datos y que han sido tenidos en cuenta en la presente sección. El análisis que sigue, por tanto, se enfoca en valorar los siguientes resultados numéricos de los experimentos: la velocidad del proceso de preprocesamiento en COLMAP en cada una de las fases de desarrollo del algoritmo, la precisión de dicho proceso para el cálculo de poses de las cámaras y generación de una nube de puntos del espacio a modelar y finalmente la calidad visual de la reconstrucción generada mediante RFM; todo ello con el fin de valorar la eficacia alcanzada, la robustez y estabilidad del proceso, el posible impacto de este trabajo, sus principales aportaciones y algunas posibles vías de mejora para su optimización y aumento de prestaciones.

Evaluación de la velocidad y robustez del procedimiento de preprocesamiento en COLMAP

En los experimentos realizados, se han adoptado progresivamente 3 enfoques (tablas 7, 8 y 9) para el preprocesamiento y generación de los archivos necesarios para el entrenamiento de cualquier modelo en Nerfstudio, cada vez más adecuados a la tipología y características de los datos utilizados. En primer lugar, se utilizó directamente el algoritmo implementado en Nerfstudio a través del comando `ns-process-data`. Aunque esto es modificable, por defecto se realiza el emparejamiento mediante vocab trees por ser el método más flexible y apto para una mayor variedad de datos de partida. Sin embargo, este método no es el más adecuado para trayectorias regulares como las descritas por el *rover* en su proceso de captura de imágenes fotográficas, donde el hecho de conocer de antemano que las fotografías que comparten características están consecutivas hace mucho más eficiente realizar un emparejamiento secuencial, en el que únicamente se empareja cada imagen con las más próximas en la carpeta y por consiguiente en el espacio tridimensional.

Este cambio de estrategia supuso una mejora sustancial tanto en los tiempos de emparejamiento como en la convergencia del proceso de mapeo. El método original mostró en múltiples casos una ineficiencia significativa, con tiempos de ejecución elevados y fallos de convergencia, especialmente en aquellas trayectorias más cortas o con menor diversidad visual. En promedio, el tiempo de emparejamiento se redujo un 93,1 % al pasar del enfoque estándar al secuencial (de 7,62 s a 0,526 s), mientras que el tiempo de mapeo bajó un 46,3 % (de 86,08 min a 46,8 min). Además, este segundo enfoque permitió lograr una reconstrucción

exitosa en todos los casos salvo uno, donde el volumen de imágenes y su solape era escaso.

La inclusión de información de poses previas procedente del sistema de navegación (mediante el uso del mapeador con poses iniciales conocidas **Pose Prior Mapper** de COLMAP) refuerza aún más esta tendencia, aumentando la robustez e incrementando la probabilidad de que la reconstrucción converja de manera satisfactoria. Este enfoque permitió una reducción adicional del 37,2 % en el tiempo del proceso de reconstrucción incremental respecto al emparejamiento secuencial simple (de 46,8 min a 29,4 min de media), manteniendo la tasa de éxito completa en la reconstrucción de todas las trayectorias. Con esta configuración, no solo se obtiene la mayor tasa de éxito en cuanto a convergencia de las reconstrucciones, sino también los tiempos de procesamiento más reducidos, lo que pone en evidencia la importancia de aprovechar todo el conocimiento disponible sobre la adquisición de datos cuando se trabaja con imágenes del mundo real.

Es importante tener en cuenta que estos experimentos tienen un componente de incertidumbre común en los procesos de entrenamiento, ya que estos dependen, entre otros, del número de hilos utilizados, la semilla o seed del entrenamiento y el par de fotografías escogidas inicialmente. Por ello, experimentos sucesivos no siempre darán los mismos resultados y no se puede esperar una replicación exacta de los mismos con la configuración actual. Este hecho puede corregirse en gran medida seleccionando manualmente parámetros como los mencionados, pero a efectos del presente trabajo se ha descartado la idea en favor de una mayor autonomía y automatización del proceso.

En conjunto, los resultados indican que el procedimiento basado en emparejamiento secuencial combinado con poses previas es el más eficaz y estable para reconstrucciones a partir de trayectorias conocidas y densas, como las generadas por el *rover*. Además, la robustez que aporta este enfoque lo convierte en el más recomendable para futuros trabajos con datos similares, especialmente cuando se dispone de información de navegación sincronizada.

Evaluación de la precisión y calidad geométrica de las reconstrucciones en COLMAP

Con respecto a la precisión de este proceso, implementado durante el preprocesado de las imágenes e imprescindible para la generación de los renderizados de Nerfstudio, se han escogido los siguientes parámetros para su evaluación, por encontrarse entre los más representativos y citados en otros textos de referencia: el error de reproyección medio, el número de observaciones por imagen y la longitud media de los tracks. Cada uno de estos indicadores permite valorar desde una perspectiva complementaria la calidad de la reconstrucción geométrica generada a partir de las imágenes de entrada y la estimación de las poses de cámara. Los resultados pueden consultarse en la tabla 10.

Error de reproyección medio: este valor, expresado en píxeles, representa la diferencia media entre la proyección calculada de un punto 3D y su posición observada real en la imagen. En general, valores superiores a 1 píxel se consideran deficientes; entre 0,8 y 1 se consideran aceptables en entornos no controlados; valores inferiores a 0,6 son buenos, y por debajo de 0,5 se consideran excelentes. En este caso, todos los caminos analizados se encuentran por debajo del umbral de 0,6, lo que indica una buena consistencia geométrica general. El mejor resultado se obtiene en la trayectoria 5, con un valor de 0,430865, que representa una precisión excelente. Le sigue la trayectoria 4 con 0,520875, también en el rango alto de calidad. La trayectoria 1 presenta el valor más alto (0,552), pero aún dentro de lo considerado como bueno. La homogeneidad de estos valores sugiere una ejecución robusta y controlada del ajuste de haces en todos los casos.

Número medio de observaciones por imagen: este parámetro mide cuántos puntos 3D han sido observados, de media, en cada imagen. A grandes rasgos, se considera deficiente cuando cae por debajo de 3000, aceptable entre 3000 y 5000, bueno entre 5000 y 6000, y excelente cuando supera los 6000. En los experimentos, cuatro de las cinco trayectorias superan el umbral de 6000 observaciones, siendo por tanto valoradas como excelentes en este aspecto. La trayectoria encabeza esta métrica con 6437,783 observaciones por imagen, seguido de la 2 y 4. La única excepción es la trayectoria 3, con 5594,433, que, si bien algo menor, sigue en el rango de calidad buena. Estos valores reflejan un buen solape visual y una densidad de puntos adecuados para garantizar redundancia geométrica durante la triangulación.

Longitud media de los tracks: este valor representa cuántas imágenes observan simultáneamente, de media, cada punto 3D. Cuanto mayor sea, mejor será la cobertura de cada punto y su precisión espacial. Se considera deficiente por debajo de 4, aceptable entre 4 y 6, buena entre 6 y 7, y excelente por encima de 7. En este caso, los resultados muestran una excelente cobertura para la trayectoria 5 (8,508), que destaca notablemente por encima del resto, lo que sugiere una gran coherencia entre imágenes consecutivas debido a la poca curvatura de la trayectoria a pesar de existir espacios notables entre algunas de las imágenes. Le sigue la 3 con 7,43, también en el rango excelente. Las trayectorias 1 y 2 se mantienen en torno al umbral de 7, mientras que la 4 muestra el valor más bajo con 6,90, aún considerado bueno. Este parámetro, en combinación con el número de observaciones, refuerza la idea de que las trayectorias analizadas ofrecen una continuidad visual y solapamiento espacial adecuados para una reconstrucción sólida.

En conjunto, los tres parámetros apuntan a una calidad geométrica notable en todas las reconstrucciones, siendo especialmente destacables los resultados obtenidos en la reconstrucción de la trayectoria 5, que lidera dos de las tres métricas con valores excepcionalmente buenos, lo que también se refleja en su estabilidad durante el renderizado final. La coherencia en los resultados y la baja variabilidad entre trayectorias permiten concluir que el desarrollo actual no solo ha alcanzado un alto grado de precisión en la estimación de las poses, sino que además es robusto frente a diferencias leves en las condiciones de captura. Esto refuerza su aplicabilidad práctica en contextos reales, donde los datos pueden ser imperfectos o difíciles de replicar.

Evaluación de la calidad visual del modelo renderizado por Nerfstudio

En cuanto al modelo renderizado por Nerfstudio en sí, que es el objeto principal de este trabajo, se ha optado por tres métricas ampliamente aceptadas en la literatura en general y en particular en las publicaciones originales de cada uno de los métodos utilizados para este proyecto: PSNR (Peak Signal-to-Noise Ratio), SSIM (Structural Similarity Index) y LPIPS (Learned Perceptual Image Patch Similarity). Estas métricas permiten comparar cuantitativamente las imágenes generadas con las imágenes reales de referencia, capturando diferentes aspectos de la fidelidad visual. La tabla 11 recoge los valores medios obtenidos en cinco trayectorias distintas, así como sus respectivas desviaciones estándar:

- **PSNR (dB):** Esta métrica evalúa la similitud pixel a pixel entre la vista renderizada y la imagen original correspondiente a la misma pose en términos de error cuadrático medio. Sin embargo, hay que tener en cuenta que su valor depende en gran medida de una correcta alineación entre ambas imágenes, por lo que cualquier pequeño desfase en la estimación de la pose o errores en la calibración pueden afectar significativamente al valor de este parámetro aunque la reconstrucción sea visualmente coherente. Es por ello que tiene que ser complementada con otros indicadores como SSIM o LPIPS. En la

práctica, valores por encima de 30 dB se consideran excelentes, entre 28 y 30 buenos, entre 25 y 28 aceptables, y por debajo de 25 deficientes. En los experimentos realizados, la trayectoria 5 destaca con 29,33 dB, situándose en el rango alto de calidad buena. Le siguen las trayectorias 1 y 3 con 28,2 dB, en el límite superior de lo aceptable. La 2 (27,10 dB) se sitúa en el umbral inferior de esa categoría, y la 4 (26,07 dB) obtiene el valor más bajo, aunque todavía dentro del rango aceptable.

- **SSIM:** Esta métrica evalúa la similitud estructural entre imágenes, considerando luminancia, contraste y estructura local. En este caso, de manera general se aceptan valores superiores a 0,9 como excelentes, entre 0,8 y 0,9 son buenos, de 0,7 a 0,8 aceptables, y por debajo de 0,7 se consideran deficientes. Todas las trayectorias presentan valores superiores a 0,78, lo que sugiere una preservación aceptable de la estructura visual. Una vez más, la trayectoria 5 (0,827) es la que mejor rendimiento ofrece, seguida de cerca por la 1 y la 3. La 4, con 0,782, obtiene el valor más bajo, pero aún dentro del rango aceptable. SSIM es más robusta que PSNR frente a pequeñas transformaciones geométricas y resulta útil para capturar degradaciones percibidas como borrosidad o pérdida de textura, aunque también puede verse favorecida si el modelo tiende a producir imágenes borrosas pero estructuralmente coherentes.
- **LPIPS:** Esta métrica perceptual, basada en activaciones de redes neuronales convolucionales entrenadas para aproximar la percepción humana, ofrece la medida más aproximada a la calidad visual entendida de forma subjetiva. Valores por debajo de 0,15 se consideran excelentes, entre 0,15 y 0,25 buenos, entre 0,25 y 0,35 aceptables, y por encima de 0,35 deficientes. En lo que a este criterio se refiere, la trayectoria 2 presenta el mejor resultado (0,204), seguido de cerca por la 5, con 0,224. En cambio, la 4, con 0,263, se acerca al límite de lo aceptable. Esta métrica tiende a penalizar imágenes visualmente incoherentes o con artefactos perceptibles, y puede ser más sensible que PSNR o SSIM a errores de textura o inconsistencias cromáticas.
- **Desviaciones estándar:** Las desviaciones estándar asociadas a cada métrica informan sobre la estabilidad de la calidad visual a lo largo de cada secuencia. En general, los valores son moderadamente bajos, indicando una consistencia razonable. La segunda trayectoria presenta la mayor homogeneidad, mientras que la quinta, a pesar de sus buenos valores absolutos, muestra las desviaciones más elevadas, lo que sugiere la coexistencia de zonas de alta calidad con otras más afectadas por fallos de reconstrucción.

Es importante contextualizar los resultados obtenidos considerando las características particulares del conjunto de datos utilizado. En este caso, cada una de las trayectorias que se evalúan procede de un único recorrido más o menos regular realizado con un dispositivo móvil, bajo condiciones no controladas, sin estabilización óptica, con iluminación natural cambiante, y presencia de desenfoques por movimiento. Además, las fotografías utilizadas en la evaluación están incluidas en las mismas trayectorias utilizadas para el entrenamiento del modelo de Nerfstudio. Esta condición puede introducir un sesgo en las métricas cuantitativas obtenidas, inflando artificialmente los valores al evaluar la capacidad de interpolación más que la de generalización del modelo. Es decir, los parámetros obtenidos, si bien notablemente positivos, se refieren mayoritariamente a las vistas generadas desde puntos cercanos a la trayectoria y con una orientación similar. En consecuencia, aunque las métricas pueden indicar una alta calidad visual, es necesario interpretarlas con cautela, ya que no necesariamente reflejan el rendimiento del sistema ante nuevas trayectorias no vistas o en escenarios no observados durante el entrenamiento. En tareas reales de reconstrucción 3D, especialmente aquellas orientadas a navegación autónoma o documentación del patrimonio, esta capacidad

de generalizar a nuevas vistas tiene gran importancia. Por tanto, se sugiere la posibilidad de complementar estas métricas con evaluaciones sobre puntos completamente independientes, o incluso imágenes tomadas en distintos momentos del día o desde dispositivos diferentes, para obtener una estimación más realista del rendimiento en entornos no controlados, para lo que será necesario cambiar del enfoque secuencial de vuelta al uso de vocab trees o a un enfoque directamente exhaustivo

Más allá de las métricas objetivas, se ha realizado una inspección visual detallada de las imágenes generadas. En general, las escenas presentan una reconstrucción geométrica razonablemente precisa, con buena continuidad en superficies planas y alineación de estructuras verticales, aunque se observan errores notables en elementos delgados o con texturas poco informativas. En cuanto a los colores, la paleta cromática se mantiene bastante coherente con las imágenes originales, con algunas diferencias en la iluminación, que se muestra algo suavizada.

Las texturas resultan considerablemente detalladas en superficies con alta frecuencia espacial y abundancia de características, como lugares con abundancia de rocas de gran o medio tamaño, pero tienden a difuminarse en zonas con pocos datos o mala cobertura visual. Las regiones vegetadas muestran un comportamiento errático, con mezclas de texturas y variaciones en el volumen según la perspectiva de origen. También los suelos rocosos más lisos presentan errores, probablemente por la homogeneidad, las propiedades reflectantes y la falta de referencias fiables. También se han identificado artefactos frecuentes en zonas cercanas a los límites del campo de visión, en los que es significativa la presencia de "fantasmas". Asimismo, algunos bordes presentan aliasing o halos debido a inconsistencias en la reconstrucción.

Aunque no se ha evaluado cuantitativamente, de forma general se aprecia una buena estabilidad visual entre fotogramas consecutivos renderizados desde trayectorias virtuales, lo cual es fundamental en aplicaciones de navegación o realidad aumentada. No obstante, esta coherencia se ve reducida en los puntos que presentan solapamiento reducido y "huecos", y en etapas intermedias de la experimentación también han sido vistas en renderizados de espacios previamente mal reconstruidos por COLMAP.

En conjunto, los resultados muestran que, pese a las limitaciones del conjunto de datos y la posible sobrestimación de las métricas, el modelo ofrece una reconstrucción visual de calidad aceptable a buena en la mayoría de las trayectorias, siendo especialmente destacable la quinta por su calidad absoluta, y la segunda por su homogeneidad. Estos resultados respaldan la viabilidad del sistema para entornos reales, aunque es imprescindible continuar evaluando su robustez y capacidad de generalización en futuros trabajos.

5.2. Impacto potencial

Aunque su implantación real en aplicaciones de robótica espacial es todavía anecdótica, NeRF y Gaussian Splatting podrían mejorar de forma significativa la autonomía y precisión de los robots en misiones en las que la intervención humana está limitada. Frente a otros métodos de percepción y caracterización del entorno más extendidos, ambos sistemas presentan importantes ventajas en términos de costes, eficiencia computacional y precisión, estando llamados a complementarlos o sustituirlos en un futuro cercano. En particular, este trabajo ha sido inicialmente concebido para su uso potencial en los siguientes campos, diferenciados en función de si existen o no restricciones que obliguen a su uso en tiempo real:

Uso en entornos con restricción de tiempo real

Mapeo del entorno y sistemas de localización en misiones de exploración: es la aplicación más directa y evidente. Ambos métodos facilitan la generación de modelos exactos que aporten información valiosa sobre colores, texturas, relieves y posibles obstáculos obteniendo importantes ventajas en la planificación autónoma de trayectorias en misiones de exploración espacial. En este sentido, Gaussian Splatting parece una opción muy interesante por la información volumétrica real que aporta sobre el terreno, pudiendo cruzarse con mapas georreferenciados y otros modelos menos precisos generados anteriormente para aumentar su detalle y localizar el *rover* con un alto grado de precisión.

Aplicaciones en robótica espacial sin requerimientos de tiempo real

Generación de modelos para simulación y experimentación con gemelos digitales: la alta precisión de los modelos los hace idóneos para la experimentación de todo tipo de sistemas de navegación autónoma en una réplica exacta del terreno sobre el que se pretende operar, disminuyendo radicalmente el coste derivado de la realización de misiones y pruebas sobre el terreno y facilitando el desarrollo de mejoras y actualizaciones de manera segura y permanente.

Inspección, seguimiento y manipulación de infraestructuras: tanto NeRF como Gaussian Splatting facilitan el seguimiento del estado de infraestructuras en lugares inaccesibles por técnicos humanos, siendo viable la generación periódica y comparación automática de modelos para detectar averías y desperfectos, así como aportando información para generar instrucciones precisas para operaciones de robótica de mantenimiento.

5.3. Propuestas para investigaciones futuras

Pese a que el proceso modificado ha cumplido con éxito con su misión de reconstruir áreas extensas de parajes similares a los que podrían hallarse en Marte, presenta algunas carencias en términos de velocidad, robustez y sensibilidad a los datos de entrada. Por ello, en este apartado se proponen las siguientes mejoras para proyectos e investigaciones posteriores:

- Uso de cámaras 360: dado que como norma generar las trayectorias son bastante regulares, es particularmente difícil para los nerf generar vistas alternas desde las direcciones hacia las que no se orienta ninguna pose. Por ello, se propone el uso de cámaras 360, cuyas imágenes son soportadas soporta directamente por Nerfstudio y aportan una información muy superior en términos de variedad de vistas. Pese a ello, hacerlo implica renunciar al emparejado secuencial de imágenes y tener un número de imágenes muy superior, por lo que es necesario valorar la conveniencia de sacrificar velocidad en aras de una calidad y robustez superior.
- Mejoras en el frecuencia de captura de imágenes del *rover*: Uno de los mayores desafíos enfrentados durante el desarrollo del experimento ha sido los huecos entre fotos consecutivas en tramos puntuales de varios de los *datasets* utilizados, pudiendo afirmarse que el correcto solapamiento de las imágenes es uno de los factores más determinantes para una adecuada reconstrucción tridimensional mediante RFM. Por ello, es aconsejable mejorar la consistencia de la frecuencia entre imágenes consecutivas. Una mejora

más avanzada sería la creación de un algoritmo de detección de discontinuidades, generando reconstrucciones del tramo anterior y posterior a la discontinuidad por separado y procediendo a la superposición de modelos contiguos.

- Combinación de Gaussian Splatting con métodos suplementarios de mapeo y localización en tiempo real: en la actualidad, el principal obstáculo para el uso de RFM en misiones de exploración planetaria es la necesidad de contar con las poses y en el caso Gaussian Splatting sobre los puntos 3D como datos de partida para una correcta reconstrucción. Aunque en el presente trabajo se ha optado por un algoritmo de SfM para realizarlo, este sistema hace prácticamente imposible el desarrollo de sistemas capaces de generar los modelos en tiempo real como sería deseable para aplicaciones de localización. Por ello, se sugiere experimentar con la obtención de las poses mediante métodos de uso extendido como LIDAR SLAM, aprovechando los MRF para la generación de modelos que aportan una precisión y cantidad de información muy superior.
- Diseño de sistema de filtrado de secciones de reconstrucción fiable: desarrollo de sistema que filtre el ruido y elimine los *splats* borrosos o que no aporten información de calidad, manteniendo únicamente en la reconstrucción los fragmentos bien reconstruidos que se puedan observar con nitidez.

6. Conclusión

El presente trabajo fin de grado (TFG) ha explorado las posibilidades del empleo de técnicas basadas en Gaussian Splatting y Campos de radiancia neuronal (NeRF) en aplicaciones relacionadas con la investigación en robótica espacial. Estas técnicas, que suponen un avance significativo frente a otras más ampliamente extendidas como las basadas en estructura a partir del movimiento (SfM) y Localización y Mapeo Simultáneo (SLAM), han demostrado ser superiores en términos de calidad de la reconstrucción, información visual aportada y eficiencia computacional. Para llegar a esta conclusión, se ha desarrollado con éxito un flujo de trabajo automatizado a partir de herramientas existentes como COLMAP y los métodos existentes que las adapta y optimiza para su uso con datos procedentes de *rovers* provistos del sistema operativo ROS 2. Asimismo, se ha comprobado la funcionalidad tanto de las versiones por defecto de las herramientas mencionadas como del código propio para la generación de reconstrucciones reales procedentes de *datasets* de prueba, de imágenes tomadas de la cámara de un smartphone y de datos tomados sobre el terreno por el laboratorio de robótica espacial de la universidad de Málaga en misiones de investigación sobre robótica espacial en entornos asimilables a la superficie marciana. En todas las fases y configuraciones, las herramientas han demostrado su operatividad cumpliendo con los objetivos planteados.

Posteriormente, se han tomado medidas de forma metódica y sistemática del tiempo requerido por cada método para cada una de las fases necesarias para generar modelos 3D de distinta índole, y se han comprobado las ventajas en términos de velocidad, robustez y precisión del algoritmo diseñado sobre la configuración estándar existente.

A pesar de estos hechos, se han identificado las carencias de los métodos para el desarrollo de aplicaciones en tiempo real con la configuración planteada, fruto esencialmente del elevado tiempo de preprocesamiento que se requiere para obtener las imprescindibles características extrínsecas (fundamentalmente la pose) de las imágenes y la deseable información dispersa del entorno a reconstruir. Este inconveniente dificulta significativamente su

uso para aplicaciones de localización, por lo que se estima necesario diseñar un método de preprocesamiento diferente que obtenga dichos parámetros de manera directa a partir de los sensores del *rover* sin depender de la etapa inicial de SfM.

Para otro tipo de aplicaciones con acceso a la comunicación con estaciones de control y menores limitaciones en términos computacionales y de tiempo, las herramientas implementadas se presentan sin necesidad de grandes cambios como una alternativa sólida para la resolución de problemas que requieren sistemas de mapeo y reconstrucción 3D.

En conjunto, el trabajo ha demostrado la viabilidad y el potencial de las tecnologías avanzadas de reconstrucción mediante RFM, Y el programa implementado supone un estándar y un punto de partida bien documentado por el presente documento para futuras investigaciones y proyectos de investigación relacionadas con la materia. Con este fin, se ha publicado y se encuentra disponible en un repositorio de Github: ⁹. Asimismo, en el mismo repositorio puede accederse a vídeos que muestran algunas de las reconstrucciones realizadas.

7. Anexos

7.1. Configuración técnica de las herramientas utilizadas

La configuración técnica del equipo utilizado para llevar a cabo las reconstrucciones 3D y los experimentos de optimización en este trabajo es la siguiente:

- **Sistema Operativo:** Ubuntu 20.04.6 LTS
 - **Arquitectura:** x86-64
 - **Kernel:** Linux 5.15.0-134-generic
- **CPU (Procesador):**
 - **Modelo:** 12th Gen Intel® Core™ i7-12700F
 - **Frecuencia base y máxima:** 2.10 GHz (base) y 4.90 GHz (máxima)
 - **Cores y Threads:** 12 núcleos físicos y 20 threads
 - **Caché:**
 - L1d: 288 KiB
 - L1i: 192 KiB
 - L2: 7,5 MiB
- **GPU (Tarjeta Gráfica):**
 - **Modelo:** NVIDIA GeForce RTX 3070
 - **Versión del driver:** 535.183.01
 - **Versión de CUDA:** 12.2
 - **Memoria VRAM:** 8 GiB
- **Memoria RAM:** 62 GiB

⁹<https://github.com/spaceuma/SpaceRFM>

- Memoria disponible durante los procesos de ejecución: 56 GiB (con 5.6 GiB en uso promedio)
- **Memoria SWAP:** 2.0 GiB
- **Almacenamiento principal (SSD):**
 - **Capacidad total:** 862 GiB
 - Espacio utilizado: 700 GiB
 - Espacio disponible: 119 GiB
- **Controladores y buses relevantes:**
 - **NVMe Controller:** Phison Electronics Corporation E16 PCIe4 NVMe Controller
 - **Red:** Ethernet Realtek RTL8111/8168/8411 PCI Express Gigabit Ethernet Controller y controlador de red inalámbrica MEDIATEK
- **Utilidades adicionales utilizadas para la ejecución y gestión de recursos:**
Herramientas snap: Configuraciones adicionales con múltiples paquetes snap utilizados, incluyendo `core`, `core18`, `core20`, y herramientas como `CMake`, `VLC`, `Notepad++`, entre otros, esenciales para el desarrollo y pruebas.

Características del servidor de alto rendimiento (IMECH)

- **Sistema Operativo:** Ubuntu Server 22.04.4 LTS
 - **Kernel:** Linux 6.5.0-25-generic
 - **Arquitectura:** x86-64
- **CPU (Procesador):**
 - **Modelo:** Intel® Xeon® Gold 6342
 - **Frecuencia base y máxima:** 2.80 GHz (base) y 3.50 GHz (máxima)
 - **Cores y Threads:** 24 núcleos físicos y 48 threads
 - **Caché:**
 - L1: 1.5 MiB
 - L2: 24 MiB
 - L3: 36 MiB
- **GPU (Tarjetas Gráficas):**
 - **Modelo:** 4× NVIDIA RTX A6000
 - **Memoria VRAM por GPU:** 48 GiB GDDR6 ECC
 - **Versión del driver:** 550.54.14
 - **Versión de CUDA:** 12.3
- **Memoria RAM:** 512 GiB DDR4 ECC
 - **Memoria SWAP:** 64 GiB
- **Almacenamiento NVMe (interno):**

- **Discos:** 2× Samsung MZQL23T8HCLS-00A07 NVMe de 3.5 TB
 - **Velocidad de escritura:** >4.0 GB/s
 - **Velocidad de lectura:** 6.9 GB/s
 - **Estructura de particionado:**
 - nvme0n1p1: 1.1 GiB (EFI)
 - nvme0n1p2: 5 GiB (/boot)
 - nvme0n1p3: 1.5 TiB (/)
 - nvme0n1p4: 2.0 TiB (usada en volumen lógico /home)
 - nvme1n1: 3.5 TiB (completamente dedicada a /home)
 - **Gestión de volúmenes:** LVM — volumen lógico `Homes` sobre VG `VG-Datos`, totalizando 5.5 TiB.
- **NAS externo montado en /home/NAS:**
- **Capacidad total:** 114 TiB
 - **Velocidad de escritura:** 450 MiB/s (vía conexión de 10 Gbps)
 - **Velocidad de conexión alternativa (no usada):** 1 Gbps (115 MiB/s)
- **Red y conectividad:**
- **Adaptador de red principal:** Tarjeta de red 10 GbE instalada en el NAS
 - **Conexión activa:** Ethernet 10 Gbps
 - **Montaje remoto del NAS:** vía NFS sobre red local
- **Entornos y herramientas adicionales:**
- **Contenedor Docker:** imagen personalizada con ROS2, COLMAP y Nerfstudio en entornos virtuales separados
 - **Gestión automatizada:** *scripts* de terminal personalizados para ejecución en entorno de servidor

Bibliografía

- [1] NASA/JPL-Caltech. (2021) Perseverance. [Fotografía] Dominio público. [Online]. Available: <https://mars.nasa.gov/mars2020/>
- [2] CSNA, S. Zhang, and K. M. Gill. (2021) The chang'e-4 lander imaged by the yutu-2 rover on the lunar far side. [Fotografía] ChangE-4 - PCAM. Publicado el 31 de mayo de 2021. [Online]. Available: <https://flickr.com/photos/53460575@N03/51217408955>
- [3] E. S. A. (ESA). (2024) Ariane 6 - test removal of mobile gantry at europe's spaceport in french guiana. [Fotografía] ESA, 22 abril 2024. Licencia CC BY-SA 3.0 IGO. [Online]. Available: https://www.esa.int/ESA_Multimedia/Images/2024/04/Ariane_6_media_kit_cover
- [4] A. Silvestri. (2017) Sistema orb-slam2 modificado sobre imágenes en el campus de la universidad austral, argentina. [Captura de pantalla] Publicado bajo licencia CC BY-SA 4.0 Internacional. Fecha: 9 abril 2017. [Online]. Available: https://commons.wikimedia.org/wiki/File:ORB-SLAM2_Alejandro_Silvestri_UAustral.jpg
- [5] CyArk. (s.f.) Modelo 3d lidar del pelourinho (salvador de bahía, brasil). [Captura de modelo LiDAR] Licencia CC BY-SA 3.0 Unported. Imagen publicada por CyArk. [Online]. Available: <https://www.cyark.org>
- [6] S. Lazebnik. (2016) Computer vision - structure from motion. Material de clase, Universidad de Illinois. Imagen adaptada con fines educativos. [Online]. Available: <https://courses.engr.illinois.edu/cs543/fa2016/>
- [7] A. Dalal, D. Hagen, K. G. Robbersmyr, and K. M. Knausgård, "Gaussian splatting: 3d reconstruction and novel view synthesis: A review," *IEEE Access*, vol. 12, p. 96797–96820, 2024. [Online]. Available: <http://dx.doi.org/10.1109/ACCESS.2024.3408318>
- [8] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. [Online]. Available: <https://arxiv.org/abs/2003.08934>
- [9] V. Croce, D. Billi, G. Caroti, A. Piemonte, L. De Luca, and P. Véron, "Comparative assessment of neural radiance fields and photogrammetry in digital heritage: Impact of varying image conditions on 3d reconstruction," *Remote Sensing*, vol. 16, no. 2, 2024. [Online]. Available: <https://www.mdpi.com/2072-4292/16/2/301>
- [10] S. Rüter, K. Waldow, N. Bartels, and A. Fuhrmann, "3d gaussian splatting for construction sites," 11 2024.

- [11] C. Xu, J. Kerr, and A. Kanazawa, “Splatfacto-w: A nerfstudio implementation of gaussian splatting for unconstrained photo collections,” *arXiv preprint arXiv:2407.12345*, 2024.
- [12] L. Gerdes, T. Wiese, R. C. Arquillo, L. Bielenberg, M. Azkarate, H. Leblond, F. Wilting, J. O. Cortés, A. Bernal, S. Palanco, and C. P. del Pulgar, “BASEPROD: The Bardenas semi-desert planetary rover dataset,” *Scientific Data*, vol. 11, no. 1, p. 1054, Sep. 2024.
- [13] A. Yarovoi and Y. K. Cho, “Review of simultaneous localization and mapping (SLAM) for construction robotics applications,” *Automation in Construction*, vol. 162, p. 105344, 2024. [Online]. Available: <https://doi.org/10.1016/j.autcon.2024.105344>
- [14] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, “Orb-slam: A versatile and accurate monocular slam system,” *IEEE Transactions on Robotics*, vol. 31, no. 5, p. 1147–1163, Oct. 2015. [Online]. Available: <http://dx.doi.org/10.1109/TRO.2015.2463671>
- [15] O. Özyeşil, V. Voroninski, R. Basri, and A. Singer, “A survey of structure from motion,” *Acta Numerica*, vol. 26, pp. 305–364, 2017.
- [16] K. Bartoš, K. Pukanska, and J. Sabová, “Overview of available open-source photogrammetric software, its use and analysis,” *Int. J. Innov. Educ. Res.*, vol. 2, pp. 62–70, 04 2014.
- [17] P. Moulon, P. Monasse, R. Perrot, and R. Marlet, “Openmvg: Open multiple view geometry,” in *Proceedings of the International Workshop on Reproducible Research in Pattern Recognition (RRPR)*. Springer, 2016, pp. 60–74.
- [18] S. Fuhrmann, F. Langguth, N. Moehrle, M. Waechter, and M. Goesele, “Mve—an image-based reconstruction environment,” *Computers Graphics*, vol. 53, pp. 44–53, 2015, 40 years of Computer Graphics in Darmstadt. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S009784931500148X>
- [19] L. Pan, D. Baráth, M. Pollefeys, and J. L. Schönberger, “Global structure-from-motion revisited,” in *European Conference on Computer Vision (ECCV)*, 2024. [Online]. Available: <https://doi.org/10.48550/arXiv.2407.20219>
- [20] J. T. Barron, B. Mildenhall, M. Tancik, P. Hedman, R. Martin-Brualla, and P. P. Srinivasan, “Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields,” 2021. [Online]. Available: <https://arxiv.org/abs/2103.13415>
- [21] T. Müller, A. Evans, C. Schied, and A. Keller, “Instant neural graphics primitives with a multiresolution hash encoding,” *ACM Transactions on Graphics*, vol. 41, no. 4, p. 1–15, Jul. 2022. [Online]. Available: <http://dx.doi.org/10.1145/3528223.3530127>
- [22] H. Turki, D. Ramanan, and M. Satyanarayanan, “Mega-nerf: Scalable construction of large-scale nerfs for virtual fly-throughs,” 2022. [Online]. Available: <https://arxiv.org/abs/2112.10703>
- [23] R. Martin-Brualla, N. Radwan, M. S. M. Sajjadi, J. T. Barron, A. Dosovitskiy, and D. Duckworth, “Nerf in the wild: Neural radiance fields for unconstrained photo collections,” 2021. [Online]. Available: <https://arxiv.org/abs/2008.02268>
- [24] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, “3d gaussian splatting for real-time radiance field rendering,” *ACM Transactions on Graphics*, vol. 42, no. 4, pp. 1–13, 2023.

- [25] Z. Cai, J. Yang, T. Wang, H. Huang, and Y. Guo, “3d reconstruction of buildings based on 3d gaussian splatting,” in *Proceedings of the 8th International Conference on Smart Data and Smart Cities (SDSC)*, vol. XLVIII-4/W10-2024. Athens, Greece: International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, June 4–7 2024, pp. 1–10, school of Geomatics and Urban Spatial Informatics, Beijing University of Civil Engineering and Architecture, Beijing, China. [Online]. Available: <https://www.int-arch-photogramm-remote-sens-spatial-inf-sci.net/>
- [26] F. Comte, A. Pamart, K. Réby, and L. De Luca, “Strategies and experiments for massive 3d digitalization of the remains after the notre dame de paris’ fire,” *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XLVIII-2/W4-2024, pp. 127–134, 2024. [Online]. Available: <https://doi.org/10.5194/isprs-archives-XLVIII-2-W4-2024-127-2024>
- [27] V. Croce, D. Billi, G. Caroti, A. Piemonte, L. D. Luca, and P. Véron, “Comparative assessment of neural radiance fields and photogrammetry in digital heritage: Impact of varying image conditions on 3d reconstruction,” *Remote Sensing*, vol. 16, no. 2, p. 301, Jan. 2024. [Online]. Available: <https://doi.org/10.3390/rs16020301>
- [28] M. Dronova, V. Cheremnykh, A. Kotcov, A. Fedoseev, and D. Tsetserukou, “Fly-nerf: Nerf-based aerial mapping for high-quality 3d scene reconstruction,” Moscow, Russia, email: (maria.dronova, vladislav.cheremnykh, alexey.kotcov, aleksey.fedoseev, d.tsetserukou)@skoltech.ru.
- [29] B. G. A. Gerats, J. M. Wolterink, and I. A. M. J. Broeders, “NeRF-OR: neural radiance fields for operating room scene reconstruction from sparse-view RGB-D videos,” *International Journal of Computer Assisted Radiology and Surgery*, vol. 20, pp. 147–156, 2025. [Online]. Available: <https://doi.org/10.1007/s11548-024-03261-5>
- [30] M. Khan, H. Fazlali, D. Sharma, T. Cao, D. Bai, Y. Ren, and B. Liu, “Autosplat: Constrained gaussian splatting for autonomous driving scene reconstruction,” 2024. [Online]. Available: <https://arxiv.org/abs/2407.02598>
- [31] I. Abaspor Kazerouni, L. Fitzgerald, G. Dooly, and D. Toal, “A survey of state-of-the-art on visual slam,” *Expert Systems with Applications*, vol. 205, p. 117734, 2022. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0957417422010156>
- [32] B. Garigipati, N. Strokina, and R. Ghabcheloo, “Evaluation and comparison of eight popular lidar and visual slam algorithms,” 2022. [Online]. Available: <https://arxiv.org/abs/2208.02063>
- [33] Z. Hong, Y. Petillot, A. Wallace, and S. Wang, “Radar slam: A robust slam system for all weather conditions,” 2021. [Online]. Available: <https://arxiv.org/abs/2104.05347>
- [34] M. Tancik, E. Weber, E. Ng, R. Li, B. Yi, T. Wang, A. Kristoffersen, J. Austin, K. Salahi, A. Ahuja, D. Mcallister, J. Kerr, and A. Kanazawa, “Nerfstudio: A modular framework for neural radiance field development,” in *Special Interest Group on Computer Graphics and Interactive Techniques Conference Proceedings*, ser. SIGGRAPH ’23. ACM, Jul. 2023, p. 1–12. [Online]. Available: <http://dx.doi.org/10.1145/3588432.3591516>
- [35] nerfstudio-project, “nerfstudio,” <https://github.com/nerfstudio-project/nerfstudio>, 2025, accessed: 2025-05-20.

- [36] S. Macenski, T. Foote, B. Gerkey, C. Lalancette, and W. Woodall, “Robot operating system 2: Design, architecture, and uses in the wild,” *Science Robotics*, vol. 7, no. 66, May 2022. [Online]. Available: <http://dx.doi.org/10.1126/scirobotics.abm6074>
- [37] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004. [Online]. Available: <https://doi.org/10.1023/B:VISI.0000029664.99615.94>
- [38] D. Nistér and H. Stewénus, “Scalable recognition with a vocabulary tree,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006, pp. 2161–2168. [Online]. Available: <http://www.vis.uky.edu/~dnister/research/vocab/vocab.pdf>
- [39] IMECH, “Especificaciones técnicas del servidor de alto rendimiento,” Comunicación personal / documentación interna, 2025, información proporcionada directamente por el Instituto Universitario de Investigación en Ingeniería Mecatrónica y Sistemas Ciberfísicos (IMECH).