# Evaluación del desempeño de un algoritmo de odometría visual en escenarios no estructurados

por

**Alejandra Márquez Cabrera**

Tesis sometida como requisito parcial para obtener el grado de

**MAESTRA EN CIENCIAS EN EL ÁREA DE CIENCIA Y TECNOLOGÍA DEL ESPACIO**

por el

**Instituto Nacional de Astrofísica, Óptica y Electrónica**

Junio, 2025
Santa María Tonantzintla, Puebla

Bajo la supervisión de

**Dr. Leopoldo Altamirano Robles, INAOE**

*A mis padres, Ceci y Pedro,*
*por su apoyo incondicional en cada decisión que he tomado*
*y por siempre brindarme el cariño*
*y las herramientas necesarias para salir adelante.*
*Gracias por estar en cada paso de mi camino.*

*A la memoria de mi hermano, Chuchis,*
*quien sigue presente en cada uno de mis logros y pensamientos.*
*Gracias por haber sido una inspiración en mi vida.*
*Esta tesis también es para ti.*

# Agradecimientos

Agradezco profundamente al INAOE, que me inspiró a tomar la decisión de realizar un posgrado en este instituto, ya que fue durante una visita en mi etapa de licenciatura, que conocí este espacio, su gente y su labor académica, lo cual sembró en mí la motivación para seguir este camino.

Expreso mi agradecimiento a la SECIHTI por el apoyo brindado a través de una beca, que me permitió desarrollar esta investigación.

Agradezco sinceramente al Dr. Leopoldo Altamirano Robles, por su guía, paciencia y compromiso a lo largo de este proceso. Gracias por compartir conmigo parte de su experiencia y conocimientos, adquiridos a lo largo de años de dedicación, los cuales enriquecieron esta tesis. Además, le agradezco por su apoyo para enfrentar y resolver los desafíos que surgieron en cada etapa de este proyecto.

Finalmente, agradezco a todas las personas que conocí durante mi estancia en el instituto, quienes con su compañía y apoyo hicieron más llevadero este proceso. Agradezco especialmente a Victor Romero, por brindarme su ayuda en cada desafío que enfrenté y por compartir conmigo sus experiencias, las cuales me sirvieron como guía en momentos clave. También extiendo mi gratitud a mi familia, por su apoyo incondicional a lo largo de mi vida, y a mi amiga Areli Aguilar, cuyo apoyo y palabras de aliento fueron un gran consuelo en los momentos más difíciles.

# Resumen

La odometría visual, VO por sus siglas en inglés, es una técnica, basada principalmente en imágenes, que estima el movimiento de un vehículo móvil mediante la información capturada por cámaras.

La estimación de la trayectoria del vehículo es un reto cuando la navegación se realiza de forma autónoma, y esto se dificulta más cuando el vehículo se desplaza sobre ambientes complejos como los llamados "no estructurados", semejantes a los que se encuentran en la superficie marciana, por lo que se requiere que los algoritmos sean diseñados específicamente para este tipo de ambientes.

En esta tesis se evalúa el algoritmo de odometría visual DF-VO comparando diferentes métricas en las trayectorias estimadas por el algoritmo, utilizando secuencias de imágenes con diversas características, desde escenarios dinámicos y estáticos hasta entornos estructurados y no estructurados. Ejemplos de estos ambientes se encuentran en los datasets KITTI, MADMAX y Rosario.

Los resultados muestran que los terrenos irregulares, con escenarios rocosos y vegetación afectan considerablemente a la precisión de las estimaciones, en comparación con escenarios urbanos y estructurados. Por lo que se valida que es necesario adaptar los algoritmos de odometría visual para que se reduzca el error que se genera en ambientes complejos y con condiciones específicas, con el fin de que sean capaces de implementarse en proyectos espaciales, donde es crucial conocer la ubicación exacta de los vehículos autónomos.

**Palabras clave:** Odometría visual, DF-VO, KITTI, MADMAX, Rosario, ambientes no estructurados, benchmarking.

# Abstract

Visual odometry (VO) is a technique, primarily based on images, that estimates the movement of a mobile vehicle using information captured by cameras.

Estimating the vehicle's trajectory is a challenge when navigation is carried out autonomously, and this becomes even more difficult when the vehicle moves through complex environments, such as so-called "unstructured environments", similar to those found on the Martian surface. Therefore, algorithms must be specifically designed for this type of environment.

This thesis evaluates the DF-VO visual odometry algorithm by comparing different metrics on the trajectories estimated by the algorithm, using image sequences with various characteristics, ranging from dynamic and static scenarios to structured and unstructured environments. Examples of these environments can be found in the KITTI, MADMAX, and Rosario datasets.

The results presented in Chapter 4 show that irregular terrains, characterized by rocky environments and vegetation, significantly affect the accuracy of the estimations when compared to urban and structured scenarios. This confirms the necessity to adapting visual odometry algorithms to minimize the error generated in complex environments and under specific conditions, in order to ensure their applicability in space exploration projects, where knowing the exact location of autonomous vehicles is crucial.

**Keywords:** Visual odometry, DF-VO, KITTI, MADMAX, Rosario, unstructured environments, benchmarking.

# Índice general

# Índice de figuras

# Índice de tablas

# Capítulo 1

# Introducción

La exploración planetaria ha permitido ampliar nuestro conocimiento sobre el universo, brindando información crucial sobre la formación y evolución de cuerpos celestes, como Marte. En este contexto, los rovers planetarios desempeñan un papel importante al capturar imágenes y muestras que permiten el análisis detallado de la superficie y las condiciones de estos entornos no estructurados.

Para que los rovers puedan desplazarse de manera autónoma y eficiente en entornos complejos, el uso de algoritmos de odometría visual es clave, ya que permiten predecir el movimiento del vehículo mediante el procesamiento de secuencias de imágenes capturadas por sus cámaras.

Sin embargo, la precisión de los algoritmos de odometría visual (VO, por sus siglas en inglés) depende de las características del entorno y de diversas fuentes de incertidumbre presentes en el proceso de predicción, por lo que esta falta de certeza puede afectar a la capacidad de los rovers en la toma de decisiones y en la ejecución de tareas, por ejemplo, en la navegación autónoma, la recolección de muestras o la generación de mapas 3D del entorno.

Debido a esto, el objetivo de esta tesis es evaluar el algoritmo de odometría visual DF-VO, utilizando datasets que muestren ambientes no estructurados, como terrenos irregulares, presencia de vegetación, zonas rocosas y condiciones topológicas complejas. Estas características permiten identificar en qué medida afectan las condiciones del entorno a la precisión de las estimaciones.

A través de este análisis se busca remarcar la importancia de diseñar algoritmos robustos y confiables, capaces de trabajar eficientemente en entornos desafiantes, como los que se tienen en la superficie marciana.

## 1.1 Antecedentes

La odometría visual se ha convertido en una técnica clave para la estimación del movimiento en sistemas de navegación autónoma, especialmente en entornos donde los métodos tradicionales, como el uso del GPS, no es viable. El ejemplo más pionero de su uso se dio con la misión Mars Exploration Rovers (MER, por sus siglas en inglés) de la NASA, con los rovers Spirit y Opportunity, que navegaron sobre la superficie marciana a partir del 2004.

El artículo "Two Years of Visual Odometry on the Mars Exploration Rovers"[1] documenta cómo la NASA enfrentó y superó importantes desafíos al implementar un sistema de VO en un entorno tan hostil y desconocido como Marte. Uno de los principales retos fue la incertidumbre inherente al terreno marciano, compuesto en su mayoría por pendientes y superficies arenosas, lo que provocaba deslizamientos significativos que no podían ser detectados únicamente con sensores inerciales o encoders en las ruedas.

Este tipo de errores acumulativos en la estimación de la posición ponía en riesgo tanto la seguridad del vehículo como la eficiencia de la misión.

Para afrontar esta situación, se desarrolló un sistema de VO capaz de estimar con precisión la pose del rover en seis grados de libertad (posición y orientación), mediante la comparación de pares de imágenes estéreo tomadas por las cámaras de navegación (NAVCAMs). El sistema

seleccionaba automáticamente características del terreno entre dos pares de imágenes y analizaba su desplazamiento para calcular el movimiento relativo del rover. Este enfoque, a pesar de estar computacionalmente limitado por el hardware a bordo, logró detectar desplazamientos milimétricos y compensar deslizamientos de hasta un 125 %, incluso en pendientes de 25°.

Posteriormente, la odometría visual evolucionó hasta convertirse en un componente esencial del sistema de navegación ya que gracias a su implementación, se logró aumentar la autonomía de los rovers, reducir riesgos de atascos y optimizar los tiempos de llegada a zonas de interés científico.

En definitiva, la implementación de VO en la misión MER de la NASA sentó las bases para el desarrollo de sistemas de percepción visual más avanzados, y demostró que, aún con recursos computacionales limitados y condiciones extremas, es posible el desarrollo de algoritmos de odometría visual robustos y confiables para la navegación en ambientes no estructurados.

## 1.2 Descripción del problema

En la actualidad, uno de los principales desafíos en el desarrollo de algoritmos de odometría visual (VO) es lograr una robustez suficiente que les permita operar de manera confiable en ambientes no estructurados. Esta necesidad se origina en el hecho de que, en contextos como la exploración planetaria, es crucial conocer con precisión la localización del rover en todo momento, ya que de ello depende la exactitud en la ubicación de muestras y registros visuales obtenidos durante la misión.

El diseño de algoritmos capaces de afrontar esta problemática implica una tarea compleja, en la que intervienen múltiples factores, como el correcto funcionamiento de los sensores y cámaras, la calidad y procesamiento de las imágenes, la cinemática del vehículo, su estimación de pose, así como las características visuales del entorno.

En este contexto, resulta pertinente identificar algoritmos de odometría visual que hayan demostrado un desempeño sólido en entornos estructurados y someterlos a evaluación en escenarios no estructurados, con el objetivo de detectar sus limitaciones y establecer áreas de oportunidad para mejorar la precisión de las estimaciones. Este análisis puede contribuir significativamente al desarrollo de soluciones más robustas y adaptables, adecuadas para su implementación en misiones espaciales y otros entornos complejos.

## 1.3 Objetivos

Dada la problemática descrita anteriormente, se proponen los siguientes objetivos para el desarrollo de la tesis.

### 1.3.1 Objetivo general

Evaluar, usando diferentes datasets, los errores asociados a las estimaciones de un algoritmo de odometría visual, y analizar la relación entre dichos errores y las características particulares de los entornos.

### 1.3.2 Objetivos específicos

- Identificar y seleccionar datasets adecuados, según las características visuales de los entornos que presentan.
- Seleccionar un algoritmo de odometría visual de reciente creación, cuya evaluación no contemple escenarios no estructurados y realizar las modificaciones necesarias para garantizar la compatibilidad y funcionalidad con diferentes datasets.
- Analizar, para cada dataset, las trayectorias estimadas por el algoritmo, así como los errores asociados a dichas estimaciones.
- Determinar la influencia de las características del entorno en la precisión de las estimaciones del algoritmo, identificando patrones o tendencias que afecten su robustez.
- Proponer modificaciones o ajustes en el algoritmo que permitan mejorar la precisión de las estimaciones al operar en ambientes complejos o no estructurados, a partir del análisis de los resultados obtenidos.

## 1.4 Metodología

Esta investigación tiene como propósito evaluar el desempeño de un algoritmo de odometría visual en entornos no estructurados, con el fin de analizar su capacidad de adaptación a condiciones diferentes a aquellas para las cuales fue originalmente diseñado.

Para ello, se procederá a seleccionar un algoritmo de VO que haya demostrado un rendimiento favorable en ambientes estructurados, y que, al mismo tiempo, no haya sido específicamente desarrollado para operar en escenarios no estructurados. Paralelamente, se llevará a cabo la selección de secuencias de imágenes representativas de entornos complejos, cuyas características visuales y estructurales presenten desafíos relevantes para el proceso de estimación.

Una vez definido el algoritmo más adecuado, tras haber considerado criterios como su arquitectura y versatilidad para operar con distintos tipos de datos, se realizarán pruebas de estimación de trayectorias utilizando las secuencias previamente seleccionadas. Posteriormente, se procederá a la evaluación cuantitativa del error mediante el uso de la herramienta "evo", la cual permitirá comparar las trayectorias estimadas con el ground truth.

Los resultados obtenidos permitirán analizar la capacidad de adaptación del algoritmo ante condiciones distintas a las previstas en su diseño original, así como identificar posibles limitaciones o áreas de mejora para su aplicación en entornos no estructurados.

# Capítulo 2

# Marco teórico

El presente capítulo tiene como propósito establecer los fundamentos teóricos que sustentan el desarrollo de esta tesis. Se presentan los conceptos relacionados con la odometría visual y se describe, de forma general, el funcionamiento del algoritmo DF-VO, el cual es la base de esta investigación.

También se mencionan las características principales de los datasets KITTI, MADMAX y Rosario, y se abordan aspectos relevantes sobre la utilización de librerías de python que hacen el cálculo del error de las estimaciones, como la librería "evo".

## 2.1 Odometría visual

La odometría visual se define como el proceso de estimar el movimiento del robot (traslación y rotación respecto a un sistema de referencia) mediante la observación de una secuencia de imágenes de su entorno [2].

De forma general, el proceso que realiza la odometría visual inicia al adquirir secuencias de imágenes mediante una o dos cámaras. A continuación, se extraen características visuales relevantes de las imágenes, como puntos de interés, esquinas o bordes. Estas características se rastrean o emparejan entre imágenes sucesivas para establecer correspondencias temporales.

Con base en estas correspondencias, se triangulan puntos en el espacio 3D, y se calcula la transformación rígida (rotación y traslación del vehículo) que relaciona las posiciones sucesivas de la cámara.

Esta estimación se optimiza mediante métodos como mínimos cuadrados, y se refina con algoritmos robustos como RANSAC para eliminar correspondencias erróneas. Finalmente, las transformaciones se integran en el tiempo para construir la trayectoria global del sistema.

### 2.1.1 Visión monocular y estéreo

La configuración del sistema de odometría visual depende del número de cámaras utilizadas. Cuando se usa una sola cámara se trata de un sistema monocular y cuando se utilizan dos cámaras hablamos de un sistema estéreo.

Cada una de estas configuraciones presenta diferentes características:

- **Configuración monocular:** emplea una sola cámara y requiere múltiples imágenes temporales para reconstruir el entorno. Su principal limitación es la ambigüedad de escala, ya que no puede inferirse la distancia real entre la cámara y los puntos observados sin ayuda externa, por ejemplo, del GPS. A pesar de su bajo costo y simplicidad, es más propenso al error acumulativo en la estimación de trayectorias.
- **Configuración estéreo:** utiliza, simultáneamente, imágenes de dos cámaras con una distancia de separación conocida. Permite la triangulación directa de puntos 3D en cada momento, lo que proporciona información de escala y profundidad. Esto reduce significativamente el error acumulativo. Sin embargo, implica mayor complejidad en el hardware y la calibración.

## 2.2 DF-VO

DF-VO (Depth-Flow Visual Odometry) [3] es un sistema monocular de odometría visual que combina módulos tradicionales de estimación geométrica con predicciones aprendidas mediante redes neuronales para mejorar la robustez, la precisión de escala y la adaptabilidad en escenarios visuales desafiantes.

### 2.2.1 Arquitectura de DF-VO

El algoritmo DF-VO, recopilado de [12], se compone de la siguiente arquitectura:

1. **Aprendizaje profundo:** utiliza dos redes neuronales entrenadas de forma independiente:
   - **Flujo óptico:** se estiman los desplazamientos de píxeles hacia adelante y hacia atrás entre imágenes consecutivas. La consistencia entre ambos flujos se emplea como medida de confianza.
   - **Profundidad:** se predice a partir de una única imagen usando un enfoque auto-supervisado. El mapa de profundidad se utiliza para generar puntos 3D en la cámara de referencia.

   Estas predicciones densas permiten eliminar la dependencia de detectores de características locales y proporcionan correspondencias más completas en escenas con baja textura o poca iluminación.

2. **Selección de correspondencias:** una vez calculados los mapas de flujo y profundidad, se proyectan los puntos 3D obtenidos en la imagen objetivo, y se comparan con los flujos predichos.

Aquellas correspondencias que violan la consistencia bidireccional (por ejemplo, en regiones ocluidas o dinámicas) se descartan. Este filtrado mejora la robustez del sistema al restringir la estimación de la pose a regiones geométricamente confiables.

3. **Recuperación de escala:** para abordar el problema inherente de la ambigüedad de escala en sistemas monoculares, DF-VO incluye un mecanismo de alineación de escala iterativa. Dado un conjunto de puntos 3D triangulados a partir de la profundidad estimada y proyectados en el cuadro actual, se calcula un factor de escala óptimo que minimiza el error de reproyección. Este proceso ajusta las estimaciones de profundidad a lo largo del tiempo, manteniendo la consistencia de escala y reduciendo el "scale drift" típico en odometría visual monocular.

4. **Selección del modelo de estimación:** DF-VO incorpora dos métodos distintos para la estimación de la pose:

   - **E-tracker:** basado en la estimación directa desde la matriz esencial, cuando no se dispone de triangulación robusta.
   - **PnP-tracker:** basado en resolver el problema PnP (Perspective-n-Point) con puntos 3D a partir de la profundidad estimada.

   Ambos modelos se ejecutan en paralelo, y se selecciona el más adecuado para cada par de imágenes utilizando el valor del criterio GRIC (Geometric Robust Information Criterion), que considera la calidad de ajuste y la complejidad del modelo.

## 2.3 LiteFlowNet

LiteFlowNet [4] es una red neuronal convolucional diseñada para estimar el flujo óptico de manera eficiente y precisa, superando a métodos previos como FlowNet2 en precisión y eficiencia computacional, a pesar de contar con una arquitectura significativamente más compacta. LiteFlowNet es aproximadamente 30 veces más pequeña y 1.36 veces más rápida que FlowNet2, logrando un rendimiento superior en datasets como Sintel y KITTI.

### 2.3.1 Arquitectura de LiteFlowNet

LiteFlowNet puede encontrarse en [13], y se basa en un enfoque especializado en la extracción de características piramidales y la estimación de flujo óptico.

1. **Extracción piramidal de características:** se compone de dos subredes denominadas NetC y NetE. NetC extrae características de alto nivel de cualquier par de imágenes en la entrada, generando pirámides de múltiples escalas. NetE estima el flujo óptico de manera progresiva desde las resoluciones más bajas hasta las más altas, refinando las estimaciones en cada nivel de la pirámide. Esta separación permite una mayor flexibilidad y precisión en la estimación del flujo óptico.

2. **Deformación de características:** en cada nivel de la pirámide, las características de la segunda imagen se deforman hacia las de la primera utilizando la estimación de flujo previa. Este enfoque reduce la distancia en el espacio de características, mejorando la precisión de la estimación del flujo.

3. **Inferencia de flujo en cascada:** LiteFlowNet introduce un módulo de inferencia de flujo en cascada compuesto por dos submódulos:

   - **Coincidencia de descriptores:** realiza una coincidencia píxel a píxel de las características de alto nivel para obtener una estimación inicial del flujo.
   - **Refinamiento subpíxel:** refina la estimación inicial para alcanzar precisión subpíxel.

   Esta estructura en cascada permite correcciones tempranas y mejora la precisión general del flujo estimado.

4. **Regularización del flujo:** para abordar problemas como valores atípicos y bordes de flujo difusos, LiteFlowNet incorpora un módulo de regularización de flujo basado en una convolución local guiada por características. En este módulo no solo se utiliza un filtro distinto para cada posición del mapa de características, sino que el filtro se construye adaptativamente para parches de flujo individuales [4].

## 2.4 Monodepth2

Monodepth2 [5] es un sistema de estimación de profundidad monocular que utiliza aprendizaje auto-supervisado, es decir, no requiere datos de profundidad etiquetados para su entrenamiento. El modelo aprende a predecir mapas de profundidad a partir de secuencias de imágenes monoculares, aprovechando la información geométrica implícita en el movimiento de la cámara entre fotogramas consecutivos.

### 2.4.1 Arquitectura de monodepth2

La arquitectura de monodepth2 en la que se basaron en [14] se compone de dos redes principales:

- **Red de estimación de profundidad:** utiliza una arquitectura tipo encoder-decoder basada en U-Net. Esta red extrae características de las imágenes de entrada y genera mapas de disparidad a múltiples escalas.
- **Red de estimación de pose:** recibe como entrada un par de imágenes consecutivas y predice la transformación de pose relativa entre ellas, representada mediante rotación y traslación.

Una de las principales contribuciones de monodepth2 es la introducción de la pérdida de reproyección mínima. Esta técnica aborda el problema de las oclusiones al calcular la pérdida fotométrica entre la imagen objetivo y las imágenes fuente reproyectadas, seleccionando la pérdida mínima para cada píxel. Esto permite manejar eficazmente regiones ocluidas y mejora la robustez del entrenamiento.

Además, el modelo incorpora un mecanismo de auto-masking que identifica y excluye píxeles que no presentan cambios significativos entre imágenes consecutivas, como los de objetos estáticos o regiones sin textura. Esto se logra comparando el error de reconstrucción de la imagen original sin deformar con el error de la imagen reconstruida; si el primero es menor, el pixel se enmascara y no contribuye a la pérdida [5].

## 2.5 Secuencias de imágenes

Para evaluar el desempeño y la precisión de los algoritmos de VO, se emplean distintos datasets. Entre los más usados está KITTI; sin embargo, existen datasets que muestran escenarios no estructurados, como es el caso de MADMAX y Rosario, los cuales representan mayores desafíos y permiten evaluar la robustez de los algoritmos en condiciones más complejas.

### 2.5.1 Entornos urbanos

El dataset KITTI, presentado en el artículo "Vision meets Robotics: The KITTI Dataset"[6], es una referencia fundamental en la investigación de robótica móvil y conducción autónoma.

Este dataset fue recopilado utilizando una automóvil equipado con múltiples sensores, incluyendo cámaras estéreo de alta resolución (tanto en color como en escala de grises), un escáner láser 3D Velodyne y un sistema de navegación inercial GPS/IMU de alta precisión. La recopilación de datos abarca aproximadamente seis horas de escenarios de tráfico reales en Karlsruhe, Alemania.

En cuanto a los tipos de escenarios presentes en el dataset KITTI, se destacan cinco categorías principales: 'Road', 'City', 'Residential', 'Campus' y 'Person'. Estas categorías representan una diversidad de entornos reales, desde autopistas y zonas rurales hasta áreas urbanas y campus universitarios, proporcionando una amplia gama de condiciones para evaluar y desarrollar algoritmos de percepción y navegación en vehículos autónomos.

### 2.5.2 Ambientes análogos marcianos

El dataset MADMAX (Morocco-Acquired Dataset of Mars-Analogue eXploration) [7], es un recurso relevante para la investigación en navegación visual en entornos análogos a Marte. Este dataset fue recopilado en el desierto de Marruecos, específicamente en ocho sitios que simulan condiciones marcianas, como Gara Medouar, Kess Kess y Maadid. En total, se capturaron 36 trayectorias, abarcando una distancia en conjunto de 9.2 km, donde la trayectoria más extensa es de 1.5 km.

Los escenarios capturados presentan una amplia variedad de terrenos desafiantes, incluyendo áreas planas con piedras, formaciones rocosas, valles con terreno accidentado y zonas con baja textura visual.

El sistema de adquisición de datos, llamado SUPER (Sensor Unit for Planetary Exploration Rovers), integró múltiples sensores, como cámaras estéreo monocromáticas, una cámara a color, cámaras omnidireccionales en configuración estéreo vertical y una unidad de medición inercial (IMU). Además, se empleó un sistema diferencial GNSS para obtener datos de referencia con 5 grados de libertad, proporcionando información precisa de posición.

### 2.5.3 Escenarios agrícolas

El dataset Rosario, presentado en [8], fue diseñado específicamente para capturar las condiciones reales y desafiantes de los campos de agricultura.

Los datos fueron recolectados en campos de cultivo ubicados en Rosario, Argentina. Los escenarios en los que se llevó a cabo la captura presentan características típicas de estos entornos: filas de

cultivo homogéneas y repetitivas; terrenos irregulares, que introducen perturbaciones mecánicas en el movimiento del robot; y condiciones de iluminación variables, incluyendo zonas de sombra y sobreexposición por la luz solar directa, lo cual complica la percepción visual y la fotometría.

El dataset incluye un total de 6 secuencias diferentes, cada una correspondiente a trayectorias reales en el campo y fueron recopiladas utilizando un robot desmalezador equipado con múltiples sensores, incluyendo una cámara estéreo ZED, una IMU, odometría de ruedas y un sistema GPS-RTK.

## 2.6 Métricas de evaluación

El paquete **evo** es una herramienta de código abierto ampliamente utilizada en robótica y visión por computadora para el análisis de trayectorias. Se encuentra disponible en el repositorio de GitHub [15] y permite la comparación entre trayectorias estimadas por algoritmos de odometría visual y trayectorias de referencia (ground truth).

Entre las métricas que calcula evo se encuentra el Absolute Pose Error (APE). Esta métrica cuantifica el error absoluto entre las posiciones estimadas por el algoritmo y la trayectoria de referencia, evaluando directamente la precisión global del sistema.

Además, el paquete evo realiza una transformación de alineamiento mediante el método de Umeyama, el cual estima una transformación de similitud, incluyendo rotación, traslación y escala, entre dos conjuntos de puntos en tres dimensiones. Este proceso permite minimizar el error entre las trayectorias, alineando la estimación con la referencia [16].

# Capítulo 3

# Trabajo relacionado

En escenarios no estructurados, los vehículos atraviesan terrenos irregulares llenos de polvo u obstáculos que provocan un aumento en la incertidumbre. Además, el desenfoque por movimiento, la vibración o iluminación variable pueden afectar directamente a la calidad de las imágenes, afectando, en consecuencia, en la estimación del movimiento.

En este capítulo se presenta una revisión del desempeño que han tenido diferentes algoritmos de odometría visual al utilizar este tipo de escenarios. Los resultados se basan en dos artículos en específico: "Evaluation of Visual SLAM Algorithms in Unstructured Planetary-like and Agricultural Environments"[9] y "Experimental Evaluation of Visual-Inertial Odometry Systems for Arable Farming"[11]. Ambos incluyen evaluaciones con el dataset Rosario, pero el primero también analiza el desempeño con el dataset MADMAX.

Esta revisión permite remarcar la problemática que aborda esta tesis y justificar la necesidad de investigar el impacto de dichas fuentes de incertidumbre, así como plantear estrategias para diseñar algoritmos de VO más robustos, capaces de entregar estimaciones precisas con escenarios no estructurados.

En esta misma línea, el trabajo de Romero-Bautista et al. [10] propone un sistema de odometría visual monocular basado en aprendizaje profundo, diseñado específicamente para ambientes agrícolas no estructurados, en los que los métodos tradicionales suelen fallar por la falta de referencias visuales confiables.

El objetivo principal de este estudio es desarrollar un sistema end-to-end sin requerimientos de sensores adicionales ni procesos de calibración complejos. Sus resultados muestran mejoras significativas en la estimación de trayectoria bajo condiciones visuales adversas, lo que refuerza la importancia de continuar investigando y diseñando algoritmos adaptados a este tipo de entornos.

## 3.1 Evaluación de desempeño con MADMAX

s En el artículo de Romero-Bautista et al. [9], se presenta una evaluación comparativa de algoritmos de SLAM visual en entornos no estructurados, que incluyen algoritmos como ORB-SLAM3, DSO, DXSLAM, DROID-SLAM y MonoViT.

La evaluación, que se realizó en 6 secuencias (A0 a A5), mostró una diferencia entre los métodos tradicionales e híbridos contra los basados en aprendizaje profundo:

- **ORB-SLAM3** en su configuración estéreo demostró ser el más robusto, con un APE promedio de 2.42 m.
- **DSO** no logró inicializar en la secuencia A0, y en las demás secuencias perdió la trayectoria después de algunas imágenes.
- **DXSLAM** falló en inicializar en todas las secuencias.
- **DROID-SLAM** y **MonoViT** completaron el 100 % de las trayectorias, aunque esto no significa un buen rendimiento, ya que la estimación puede ser errónea.

El artículo destaca que los principales desafíos identificados incluyeron textura compleja, cambios de iluminación y movimientos drásticos debido a perturbaciones del terreno.

## 3.2 Rosario en algoritmos de odometría visual

Se utilizaron todas las secuencias del dataset Rosario en ambos artículos para evaluar la robustez de algoritmos VSLAM y VIO frente a los desafíos típicos de zonas agrícolas, como baja textura, repetitividad visual, movimiento de vegetación y vibraciones provocadas por el terreno irregular.

En el artículo de Romero-Bautista et al. [9], se evaluaron los mismos algoritmos mencionados anteriormente. Los resultados fueron:

- **ORB-SLAM3** en su configuración estéreo nuevamente demostró ser el de mejor desempeño completando todas sus trayectorias, aunque con un APE promedio de 9.79 m.
- **MonoViT** completó todas las trayectorias, pero con un APE promedio de 44.89 m y mostró un mejor desempeño cuando las trayectorias eran en línea recta.
- **DROID-SLAM** mostró el peor resultado, ya que aunque completó todas las trayectorias, todas fueron erróneas, con APE mayor a 100.
- **DSO** y **DXSLAM** tuvieron problemas en la estimación del eje **X** y perdieron el seguimiento.

El trabajo de Cremona et al. [11] incorporó una lista más amplia de algoritmos, la cual incluyó Basalt, FLVIS, Kimera-VIO, OKVIS, ORB-SLAM3 en su configuración monocular, REBVO, ROVIO, R-VIO, SVO-2.0, S-MSCKF y VINS-Fusion.

Se realizaron múltiples experimentos que permitieron identificar que S-MSCKF y ORB-SLAM3 tuvieron el mejor desempeño contra los demás. Destacan ROVIO, Basalt y FLVIS por su robustez, mientras que VINS-Fusion falla ocasionalmente en la secuencia 1, al igual que OKVIS y SVO-2.0 en la secuencia 5. El resto de los métodos fallan en al menos una secuencia.

Ambos estudios concluyen que, aunque algunos algoritmos logran desempeños aceptables, ninguno alcanza la robustez necesaria para aplicaciones en entornos de agricultura sin GNSS. Se hace énfasis en mejorar los módulos de detección de características y estimación de movimiento, especialmente en escenas con cielo dominante y con patrones repetitivos.

# Capítulo 4

# Evaluación en ambientes no estructurados

En este capítulo se describe la metodología para la evaluación del algoritmo DF-VO en escenarios no estructurados. Se abordan los criterios para la selección de secuencias específicas; características del algoritmo de odometría visual, su ejecución y el entrenamiento de monodepth2; así como consideraciones adicionales sobre la evaluación del desempeño.

## 4.1 Selección de datasets

Para la evaluación del algoritmo propuesto se utilizaron los datasets **KITTI**, **MADMAX** y **Rosario**, cada uno con propiedades particulares en cuanto a formato, resolución y características visuales. La elección de secuencias de cada dataset responde a la necesidad de evaluar el algoritmo bajo desafíos visuales debidos a ambientes no estructurados.

### 4.1.1 KITTI

Este dataset contiene imágenes estéreo en formato **.png** con una resolución de 1241 $\times$ 376 píxeles. Para la evalución se usó el mismo formato y resolución, y se eligieron las secuencias 3, 7 y 10, mostradas en la Figura 4.1, debido a que presentan zonas urbanas con vegetación y

giros suaves; entornos urbanos con autos estacionados y en movimiento, y cambios frecuentes de dirección; y carreteras con vistas llenas de árboles y con menor presencia de edificios, respectivamente.

(a) (b) (c)

Figura 4.1: Escenas de las secuencias (a) 3, (b) 7 y (c) 10 del dataset KITTI.

### 4.1.2 MADMAX

Las imágenes de MADMAX fueron preprocesadas para tener una resolución de 516 × 386 píxeles y formato **.jpg**. Se seleccionaron las secuencias A0, D2 y F2, mostradas en la Figura 4.2, ya que muestran terrenos arenosos con presencia de colinas y rocas; zonas llenas de rocas pequeñas y grava; y terreno irregular compuesto de grava y arena, respectivamente.

(a) (b) (c)

Figura 4.2: Escenas de las secuencias (a) A0, (b) D2 y (c) F2 del dataset MADMAX.

### 4.1.3 Rosario

Las imágenes de las secuencias seleccionadas también fueron preprocesadas, con el fin de tener una resolución diferente de las imágenes anteriores, por lo que se definió una resolución de $672 \times 376$ píxeles con formato **.jpg**. Las secuencias seleccionadas fueron 1-3, mostradas en la Figura 4.3 ya que las tres muestran entornos agrícolas con líneas de surco, terreno irregular y escenas repetitivas.

(a) (b) (c)

Figura 4.3: Escenas de las secuencias (a) 1, (b) 2 y (c) 3 del dataset Rosario.

## 4.2 Selección del algoritmo

El algoritmo DF-VO representa un enfoque híbrido y altamente versátil para resolver el problema de la estimación de movimiento en secuencias de imágenes monoculares. Su diseño se destaca por integrar técnicas clásicas de estimación geométrica con modelos de aprendizaje profundo, proponiendo así una solución robusta y con alta precisión para tareas de odometría visual, incluso en entornos desafiantes.

Una característica significativa, por la cuál se eligió el algoritmo, es su estructura modular. DF-VO separa los módulos de aprendizaje

profundo de los módulos de geometría clásica, permitiendo actualizar o reemplazar fácilmente los componentes según las necesidades del sistema o las características del entorno. Esta modularidad facilita su adaptación a distintos datasets o escenarios, como entornos urbanos, naturales o incluso planetarios.

Otra de las características más distintivas de DF-VO es su capacidad para estimar el movimiento de la cámara a partir de la combinación de la estimación del flujo óptico y la profundidad de la escena. En lugar de confiar únicamente en puntos característicos o en emparejamiento directo de imágenes, el algoritmo utiliza mapas de flujo y profundidad generados por redes neuronales profundas (como LiteFlowNet para el flujo óptico y monodepth2 para la estimación de profundidad monocular), lo que le permite mejorar la precisión de la estimación de pose.

## 4.3 Experimento 1: Evaluación de DF-VO en ambientes no estructurados.

Originalmente, el algoritmo DF-VO utiliza el estimador de profundidad monodepth2, entrenado con imágenes del dataset KITTI. Incluso la evaluación presentada por los autores de DF-VO se llevó a cabo utilizando secuencias de dicho dataset. Por esta razón, el primer experimento consistió en transformar los datasets MADMAX y Rosario al formato de KITTI, con el objetivo de evaluar el desempeño del algoritmo en escenarios no estructurados.

Como parte del proceso de integración, las imágenes y archivos de los datasets de MADMAX y Rosario se añadieron a las carpetas correspondientes dentro de la estructura del repositorio de DF-VO.

Además, para cada dataset se modificaron tanto las rutas de acceso a los archivos como la resolución de las imágenes, de acuerdo con sus características particulares, garantizando la compatibilidad con el algoritmo.

Los resultados presentados en la Tabla 4.1 evidencian que, al utilizar secuencias del dataset KITTI, el algoritmo mantiene un desempeño favorable. En contraste, al aplicarlo a las secuencias de MADMAX y Rosario (Tablas 4.2 y 4.3, respectivamente), si bien DF-VO logra completar las trayectorias, las estimaciones contienen errores significativos.

Una observación adicional relevante es que, a pesar de que DF-VO no fue diseñado para operar en entornos no estructurados, su rendimiento con el dataset MADMAX es superior al obtenido con Rosario. Esto se refleja en una mayor coherencia entre las estimaciones por eje, así como en un menor valor del APE en comparación con el obtenido con Rosario. Esta observación motivó la realización de un segundo experimento.

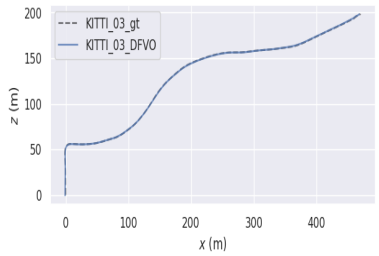
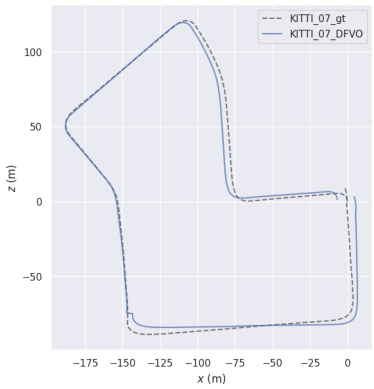
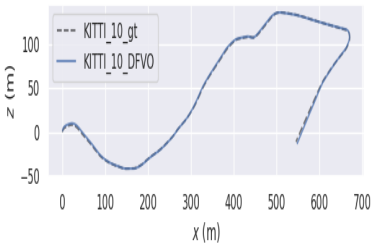
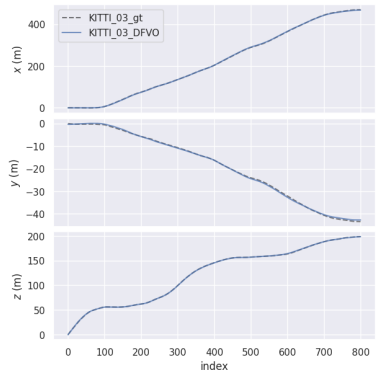
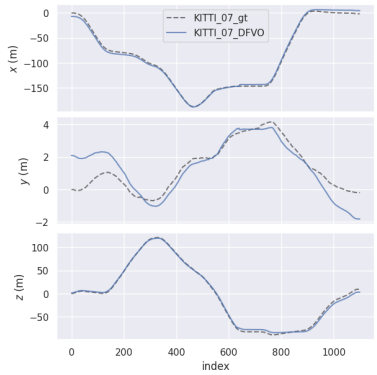
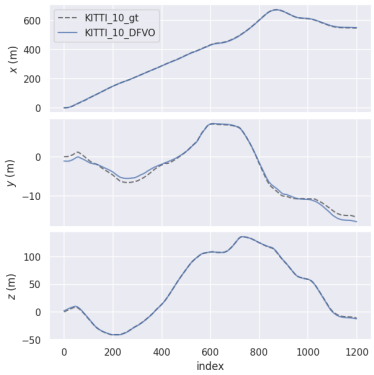
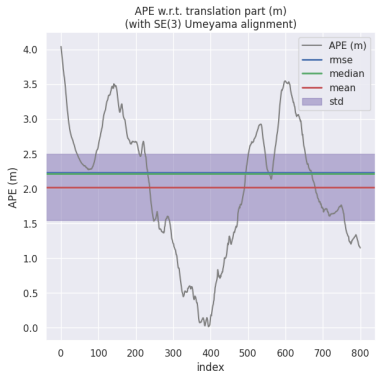
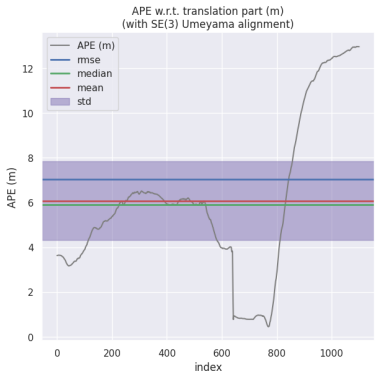
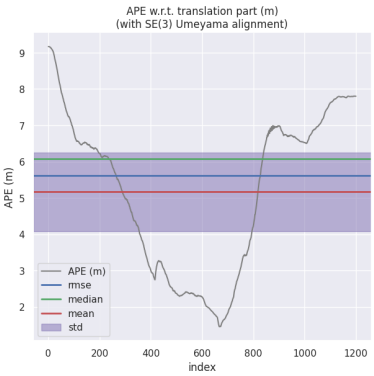
| KITTI | | |
|---|---|---|
| **Secuencia 03** | **Secuencia 07** | **Secuencia 10** |
|  |  |  |
| Figura 4.4: E1 - Estimación de trayectoria de la secuencia 3 de KITTI. | Figura 4.5: E1 - Estimación de trayectoria de la secuencia 7 de KITTI. | Figura 4.6: E1 - Estimación de trayectoria de la secuencia 10 de KITTI. |
|  |  |  |
| Figura 4.7: E1 - Estimación por coordenadas de la secuencia 3 de KITTI. | Figura 4.8: E1 - Estimación por coordenadas de la secuencia 7 de KITTI. | Figura 4.9: E1 - Estimación por coordenadas de la secuencia 10 de KITTI. |
|  |  |  |
| Figura 4.10: E1 - APE de la secuencia 3 de KITTI. | Figura 4.11: E1 - APE de la secuencia 7 de KITTI. | Figura 4.12: E1 - APE de la secuencia 10 de KITTI. |

Tabla 4.1: Dataset KITTI y entrenamiento de monodepth2 con dataset KITTI.

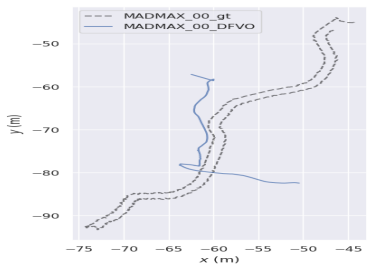
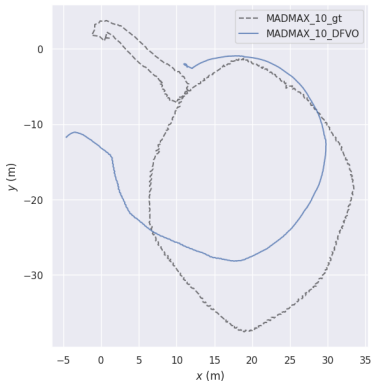
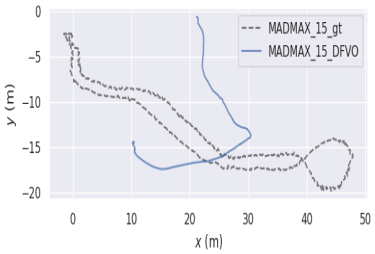
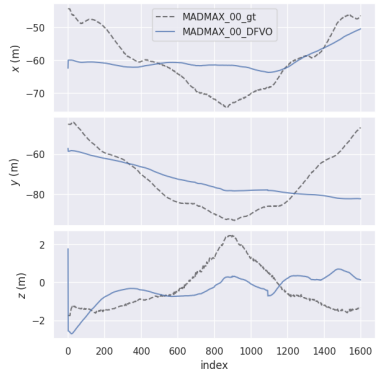
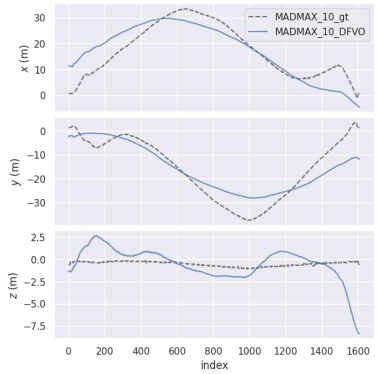
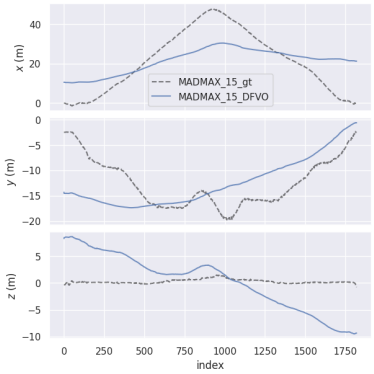
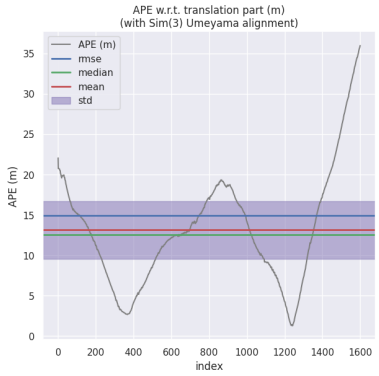
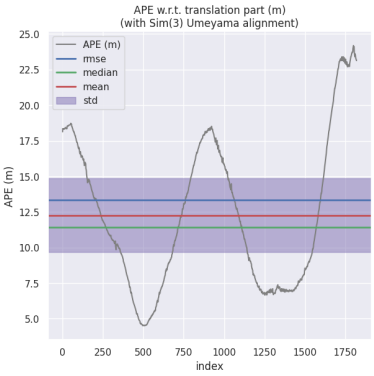
| MADMAX | | |
|---|---|---|
| **Secuencia A0** | **Secuencia D2** | **Secuencia F2** |
|  |  |  |
| Figura 4.13: E1 - Estimación de trayectoria de la secuencia A0 de MADMAX. | Figura 4.14: E1 - Estimación de trayectoria de la secuencia D2 de MADMAX. | Figura 4.15: E1 - Estimación de trayectoria de la secuencia F2 de MADMAX. |
|  |  |  |
| Figura 4.16: E1 - Estimación por coordenadas de la secuencia A0 de MADMAX. | Figura 4.17: E1 - Estimación por coordenadas de la secuencia D2 de MADMAX. | Figura 4.18: E1 - Estimación por coordenadas de la secuencia F2 de MADMAX. |
|  |  |  |
| Figura 4.19: E1 - APE de la secuencia A0 de MADMAX. | Figura 4.20: E1 - APE de la secuencia D2 de MADMAX. | Figura 4.21: E1 - APE de la secuencia F2 de MADMAX. |

Tabla 4.2: Dataset MADMAX y entrenamiento de monodepth2 con dataset KITTI.

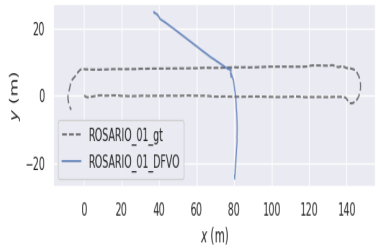
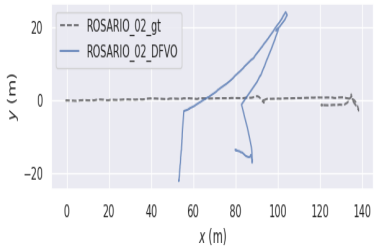
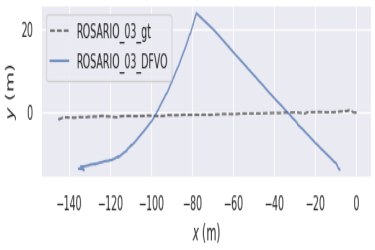
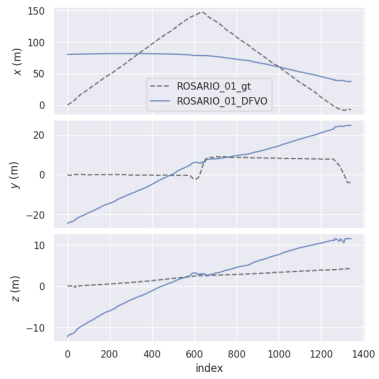
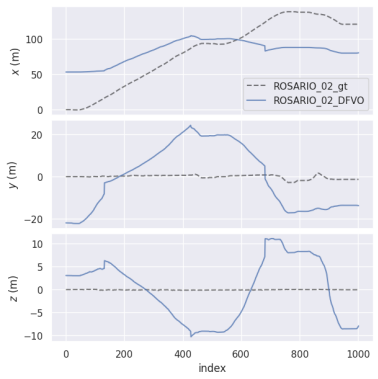
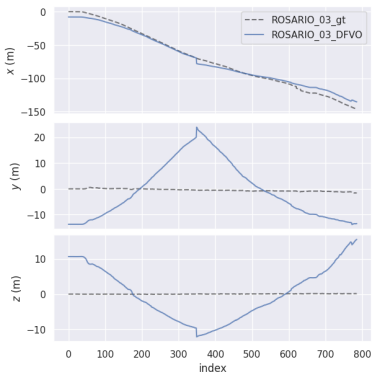
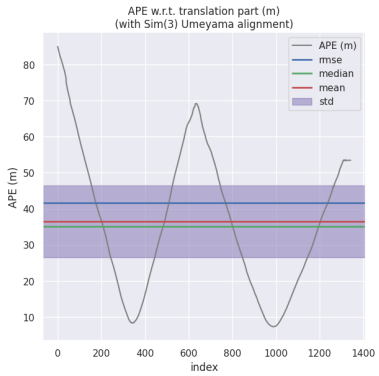
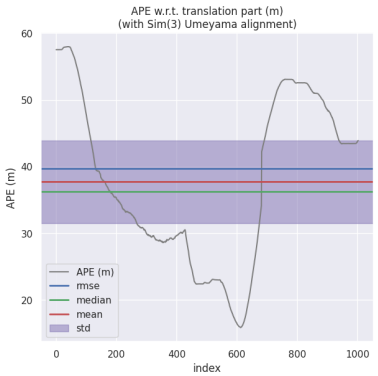
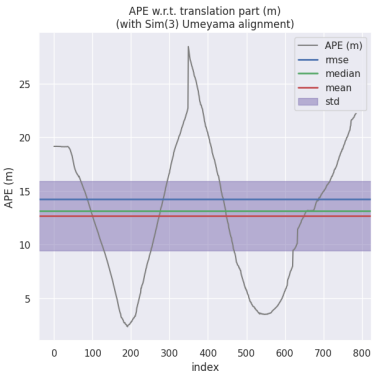
| ROSARIO | | |
|---|---|---|
| **Secuencia 01** | **Secuencia 02** | **Secuencia 03** |
|  |  |  |
| Figura 4.22: E1 - Estimación de trayectoria de la secuencia 1 de Rosario. | Figura 4.23: E1 - Estimación de trayectoria de la secuencia 2 de Rosario. | Figura 4.24: E1 - Estimación de trayectoria de la secuencia 3 de Rosario. |
|  |  |  |
| Figura 4.25: E1 - Estimación por coordenadas de la secuencia 1 de Rosario. | Figura 4.26: E1 - Estimación por coordenadas de la secuencia 2 de Rosario. | Figura 4.27: E1 - Estimación por coordenadas de la secuencia 3 de Rosario. |
|  |  |  |
| Figura 4.28: E1 - APE de la secuencia 1 de Rosario. | Figura 4.29: E1 - APE de la secuencia 2 de Rosario. | Figura 4.30: E1 - APE de la secuencia 3 de Rosario. |

Tabla 4.3: Dataset Rosario y entrenamiento de monodepth2 con dataset KITTI.

## 4.4 Experimento 2: Reentrenamiento de monodepth2 con Rosario

El trabajo de Romero-Bautista et al. [9] y [10] motivó a la realización del segundo experimento, el cual está orientado al reentrenamiento del modelo monodepth2 utilizando imágenes del dataset Rosario.

Para esto, las imágenes fueron convertidas del formato TUM al formato KITTI, y se redimensionaron a una resolución de $320 \times 160$ píxeles, siguiendo la recomendación de los autores de monodepth2 para evitar la susceptibilidad a la escala durante el entrenamiento. A partir de esto, el entrenamiento se llevó a cabo utilizando una GPU NVIDIA RTX 2070 con una taza de aprendizaje de $10^{-5}$, durante un total de 100 épocas, lo cual requirió un tiempo aproximado de 16 horas.

En cuanto a los resultados obtenidos, en la Tabla 4.4 se observa que en el caso del dataset KITTI no se presentan cambios significativos: a pesar de que los valores del APE se incrementaron en cada secuencia, las estimaciones de trayectorias se conservan semejantes al ground truth.

Sin embargo, con MADMAX (Tabla 4.5) el APE se incrementa en comparación con el primer experimento, y las estimaciones de trayectoria siguen mostrando inconsistencias considerables.

En la Tabla 4.6 se presenta el caso del dataset Rosario, donde se aprecia una reducción del APE, comparado con el primer experimento, lo cual indica una mejora en el desempeño del algoritmo. Aunque las trayectorias estimadas aún presentan desviaciones respecto al ground truth, se observa una mayor fidelidad en la forma general del recorrido, lo que sugiere un mejor ajuste a las características particulares del entorno representado en Rosario.

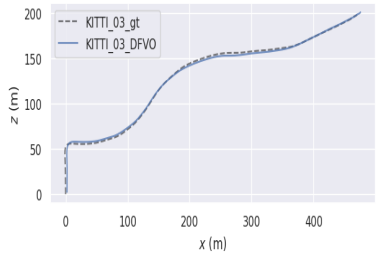
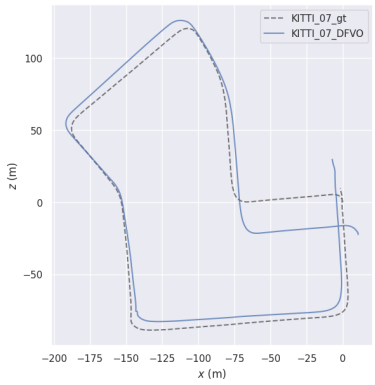
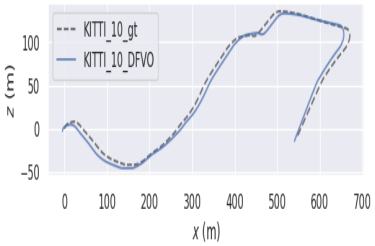
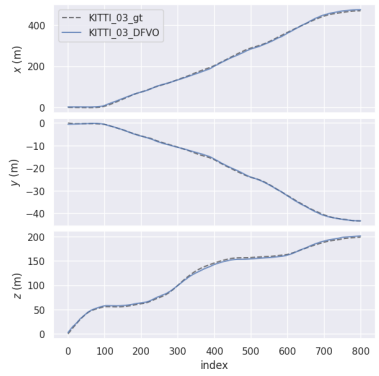
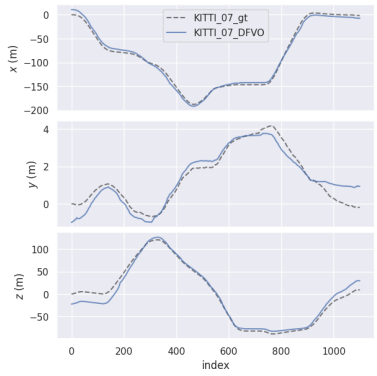
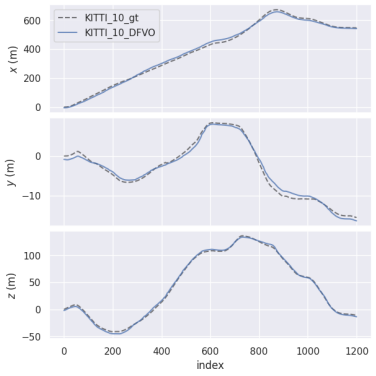
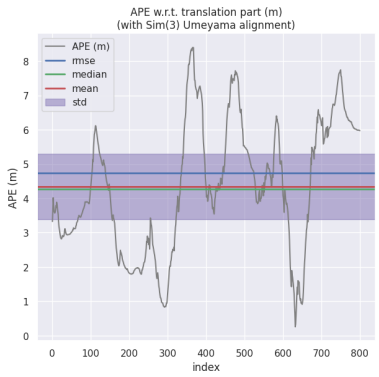
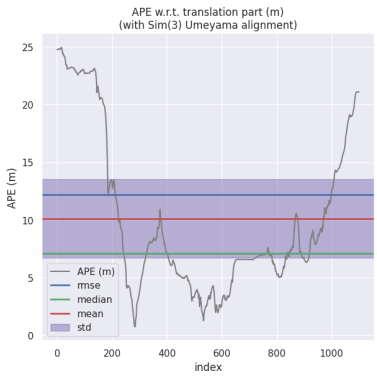
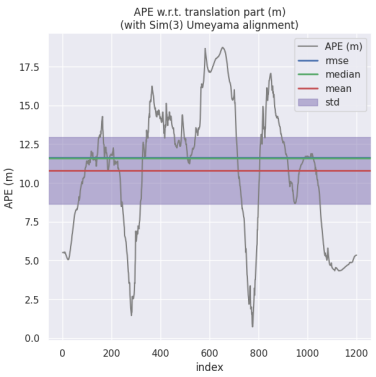
| KITTI | | |
|---|---|---|
| **Secuencia 03** | **Secuencia 07** | **Secuencia 10** |
|  |  |  |
| Figura 4.31: E2 - Estimación de la trayectoria de la secuencia 3 de KITTI. | Figura 4.32: E2 - Estimación de la trayectoria de la secuencia 7 de KITTI. | Figura 4.33: E2 - Estimación de la trayectoria de la secuencia 10 de KITTI. |
|  |  |  |
| Figura 4.34: E2 - Estimación por coordenadas de la secuencia 3 de KITTI. | Figura 4.35: E2 - Estimación por coordenadas de la secuencia 7 de KITTI. | Figura 4.36: E2 - Estimación por coordenadas de la secuencia 10 de KITTI. |
|  |  |  |
| Figura 4.37: E2 - APE de la secuencia 3 de KITTI. | Figura 4.38: E2 - APE de la secuencia 7 de KITTI. | Figura 4.39: E2 - APE de la secuencia 10 de KITTI. |

Tabla 4.4: Dataset KITTI y entrenamiento de monodepth2 con dataset Rosario.

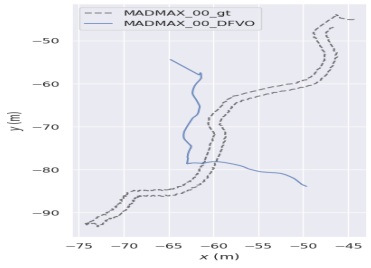
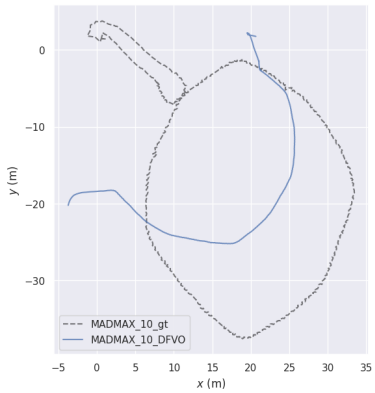
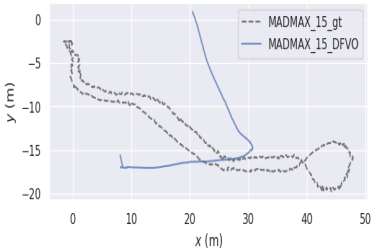
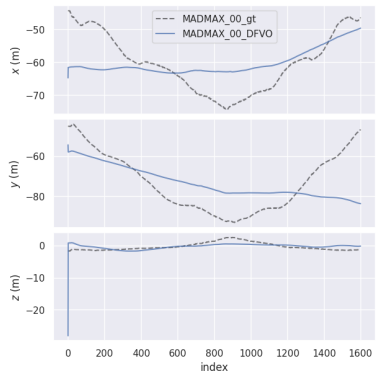
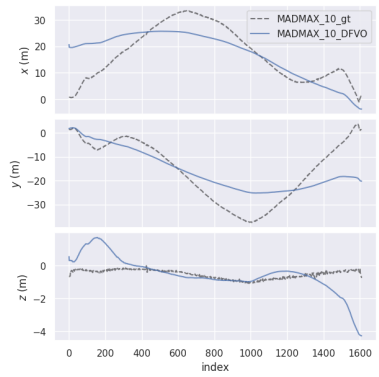
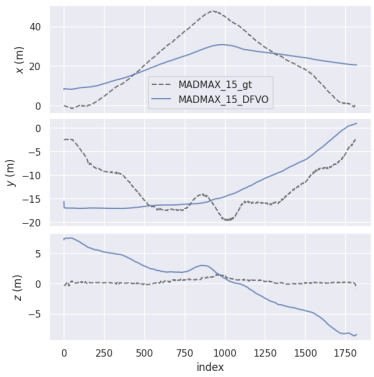
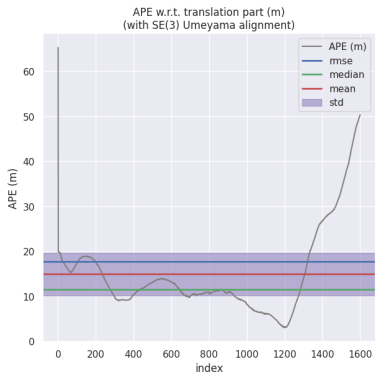
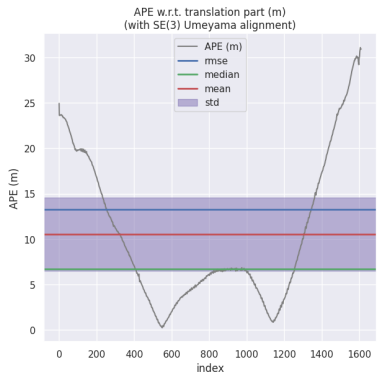
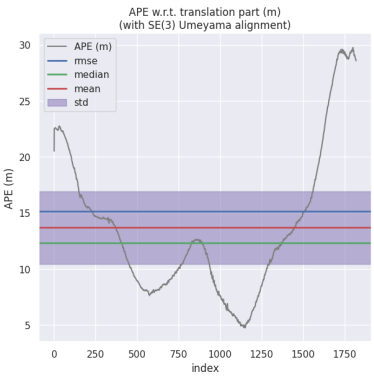
| MADMAX | | |
|---|---|---|
| **Secuencia A0** | **Secuencia D2** | **Secuencia F2** |
|  |  |  |
| Figura 4.40: E2 - Estimación de la trayectoria de la secuencia A0 de MADMAX. | Figura 4.41: E2 - Estimación de la trayectoria de la secuencia D2 de MADMAX. | Figura 4.42: E2 - Estimación de la trayectoria de la secuencia F2 de MADMAX. |
|  |  |  |
| Figura 4.43: E2 - Estimación por coordenadas de la secuencia A0 de MADMAX. | Figura 4.44: E2 - Estimación por coordenadas de la secuencia D2 de MADMAX. | Figura 4.45: E2 - Estimación por coordenadas de la secuencia F2 de MADMAX. |
|  |  |  |
| Figura 4.46: E2 - APE de la secuencia A0 de MADMAX. | Figura 4.47: E2 - APE de la secuencia D2 de MADMAX. | Figura 4.48: E2 - APE de la secuencia F2 de MADMAX. |

Tabla 4.5: Dataset MADMAX y entrenamiento de monodepth2 con dataset Rosario.

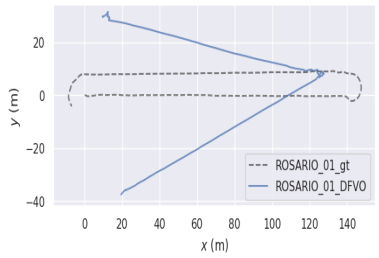
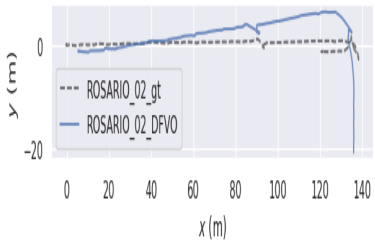
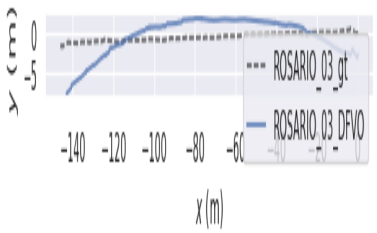
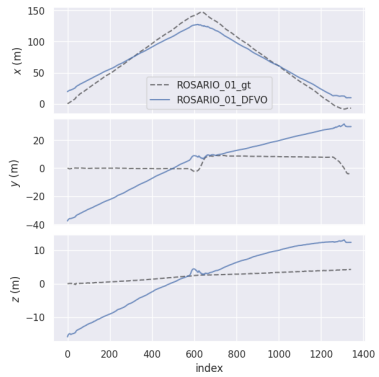
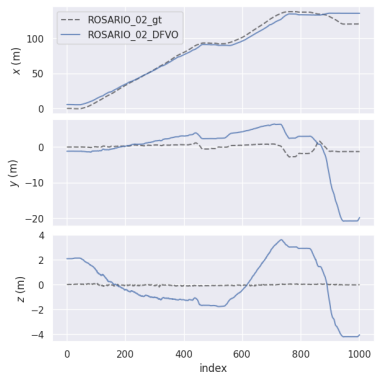
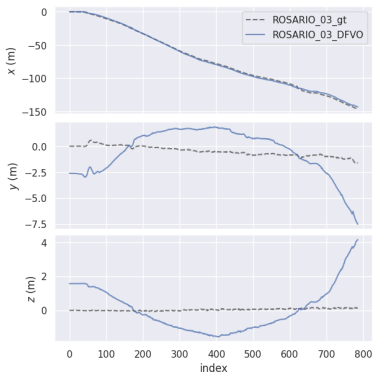
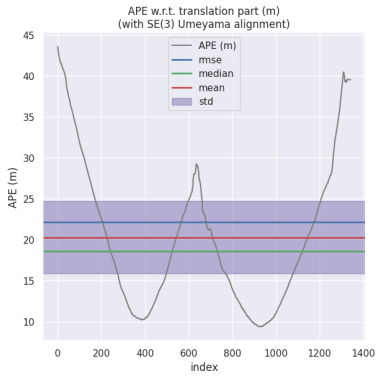
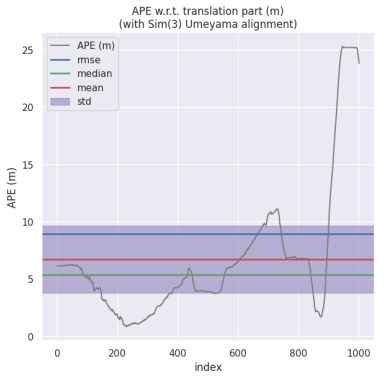
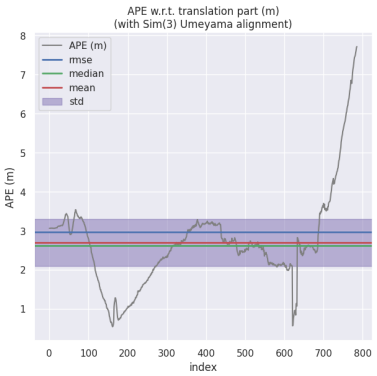
| ROSARIO | | |
|---|---|---|
| **Secuencia 01** | **Secuencia 02** | **Secuencia 03** |
|  |  |  |
| Figura 4.49: E2 - Estimación de la trayectoria de la secuencia 1 de Rosario. | Figura 4.50: E2 - Estimación de la trayectoria de la secuencia 2 de Rosario. | Figura 4.51: E2 - Estimación de la trayectoria de la secuencia 3 de Rosario. |
|  |  |  |
| Figura 4.52: E2 - Estimación por coordenadas de la secuencia 1 de Rosario. | Figura 4.53: E2 - Estimación de la trayectoria de la secuencia 2 de Rosario. | Figura 4.54: E2 - Estimación de la trayectoria de la secuencia 3 de Rosario. |
|  |  |  |
| Figura 4.55: E2 - APE de la secuencia 1 de Rosario. | Figura 4.56: E2 - APE de la secuencia 2 de Rosario. | Figura 4.57: E2 - APE de la secuencia 3 de Rosario. |

Tabla 4.6: Dataset Rosario y entrenamiento de monodepth2 con dataset Rosario.

# Capítulo 5

# Discusión de resultados

Los resultados obtenidos en ambos experimentos permiten identificar la influencia que tiene el entrenamiento de monodepth2 con el dataset Rosario en el desempeño general del algoritmo DF-VO. En el primer experimento, donde se utilizó el modelo entrenado con KITTI, el algoritmo mostró un rendimiento óptimo únicamente sobre secuencias de dicho dataset, mientras que en los entornos no estructurados de MADMAX y Rosario las estimaciones de trayectoria fueron claramente erróneas, aunque las trayectorias fueron completadas.

Tras reentrenar monodepth2 con imágenes del dataset Rosario, se notó una mejora en la adaptación del algoritmo a este entorno en específico, expresada en una reducción del error y una mayor fidelidad en la forma de la trayectoria estimada. Sin embargo, esta mejora no se extendió a MADMAX, ya que el error aumentó, y el desempeño con KITTI se mantuvo estable.

Estos resultados sugieren que el entrenamiento del estimador de profundidad juega un papel crucial en la capacidad de adaptación de DF-VO a distintos ambientes. Si bien, el reentrenamiento con datos de Rosario mejora el ajuste en ese ambiente en particular, lo hace a costa de la degradación del desempeño en otros. Por tanto, para mejorar la robustez del algoritmo en entornos no estructurados, es necesario considerar esquemas de entrenamiento más diversos o específicos para cada escenario.

Finalmente, para complementar el análisis, en la Tabla 5.1, la Tabla 5.2 y la Tabla 5.3, se presentan visualizaciones de las trayectorias estimadas en los datasets KITTI, MADMAX y Rosario, respectivamente, correspondientes a ambos experimentos.

En lás imágenes, los resultados del primer experimento se muestran en color azul, mientras que los del segundo experimento, tras el reentrenamiento de monodepth2 con imágenes de Rosario, se indican en color verde.

Estas representaciones permiten observar de forma cualitativa los errores cometidos por el algoritmo en cada caso. En particular, es notorio el incremento del APE en las secuencias de KITTI cuando el modelo ha sido entrenado con imágenes del dataset Rosario, lo que refuerza la hipótesis de que la red de profundidad pierde capacidad de generalización fuera del dominio de entrenamiento. Por otro lado, en las secuencias correspondientes al dataset Rosario, se observa una disminución considerable del APE en comparación con el primer experimento, así como una mayor coherencia en la trayectoria estimada, evidenciando una mejor adaptación del algoritmo a las condiciones del entorno para el cual fue ajustado el modelo de profundidad.

Estas visualizaciones contribuyen a una interpretación más intuitiva del impacto que tiene el entorno de entrenamiento sobre la precisión del sistema DF-VO, y refuerzan la necesidad de considerar la compatibilidad entre los datos de entrenamiento y el entorno de aplicación al momento de implementar sistemas de odometría visual basados en aprendizaje profundo, como es el caso de DF-VO.

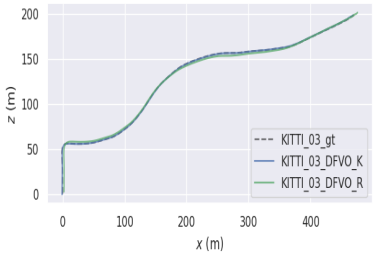
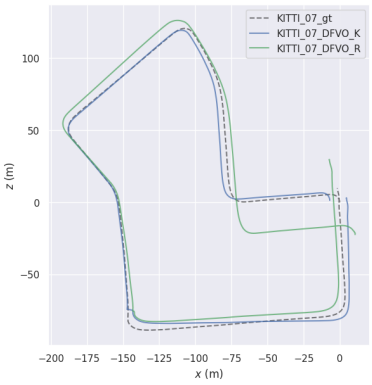
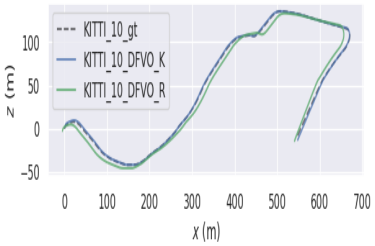
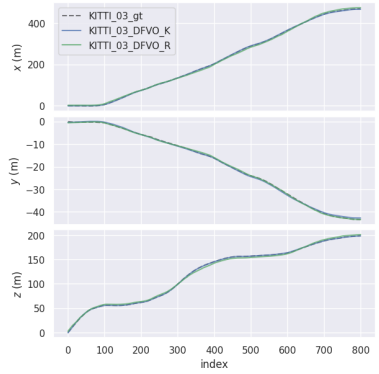
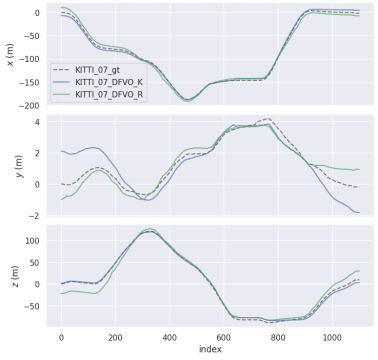
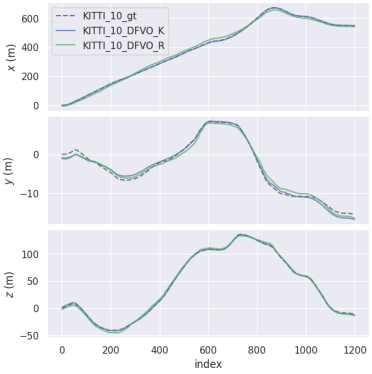
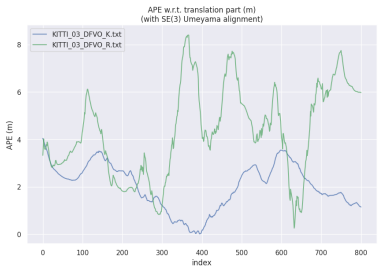
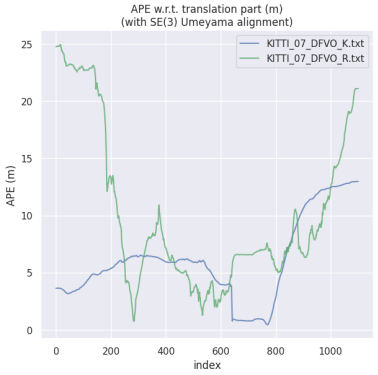
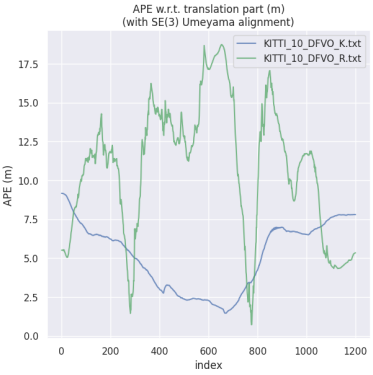
| KITTI | | |
|---|---|---|
| **Secuencia 03** | **Secuencia 07** | **Secuencia 10** |
|  |  |  |
| Figura 5.1: Trayectorias de la secuencia 3 de KITTI. | Figura 5.2: Trayectorias de la secuencia 7 de KITTI. | Figura 5.3: Trayectorias de la secuencia 10 de KITTI. |
|  |  |  |
| Figura 5.4: Coordenadas de la secuencia 3 de KITTI. | Figura 5.5: Coordenadas de la secuencia 7 de KITTI. | Figura 5.6: Coordenadas de la secuencia 10 de KITTI. |
|  |  |  |
| Figura 5.7: APE de la secuencia 3 de KITTI. | Figura 5.8: APE de la secuencia 7 de KITTI. | Figura 5.9: APE de la secuencia 10 de KITTI. |

Tabla 5.1: Comparación de resultados de los experimentos 1 y 2 usando el dataset de KITTI.

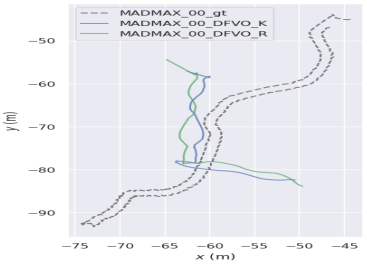
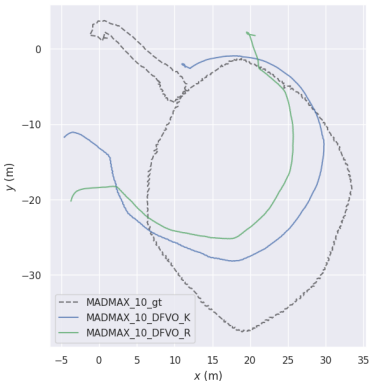
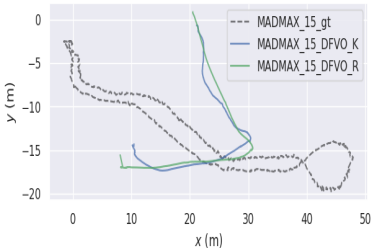
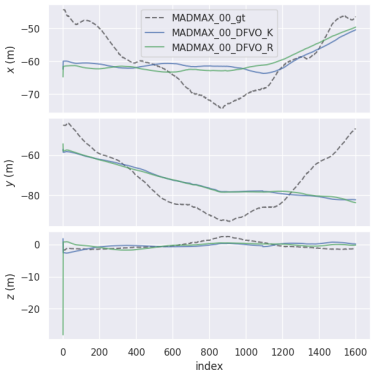
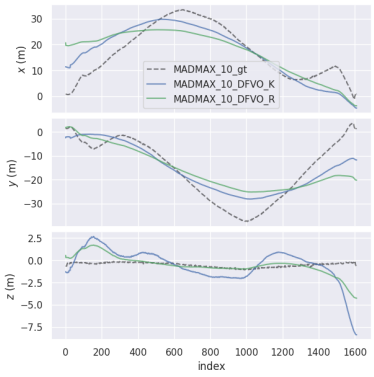
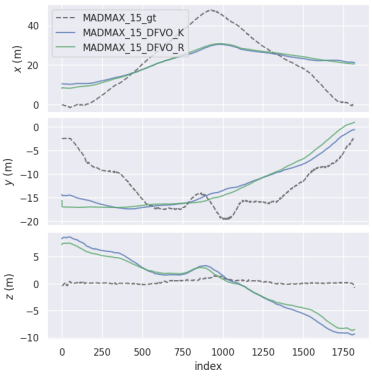
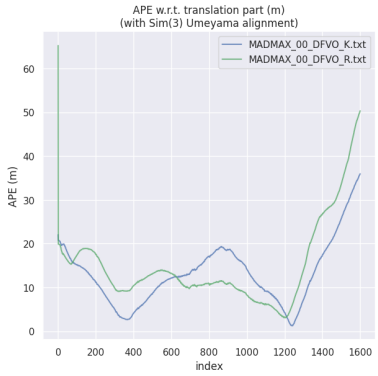
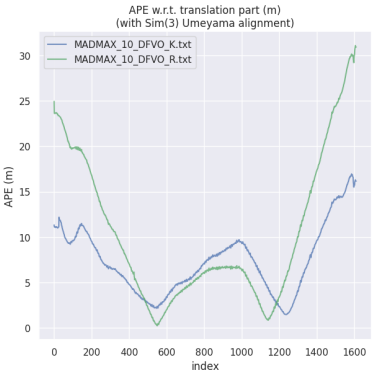
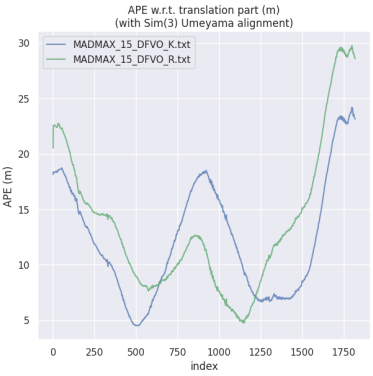
| MADMAX | | |
|---|---|---|
| **Secuencia A0** | **Secuencia D2** | **Secuencia F2** |
|  |  |  |
| Figura 5.10: Trayectorias de la secuencia A0 de MADMAX. | Figura 5.11: Trayectorias de la secuencia D2 de MADMAX. | Figura 5.12: Trayectorias de la secuencia F2 de MADMAX. |
|  |  |  |
| Figura 5.13: Coordenadas de la secuencia A0 de MADMAX. | Figura 5.14: Coordenadas de la secuencia D2 de MADMAX. | Figura 5.15: Coordenadas de la secuencia F2 de MADMAX. |
|  |  |  |
| Figura 5.16: APE de la secuencia A0 de MADMAX. | Figura 5.17: APE de la secuencia D2 de MADMAX. | Figura 5.18: APE de la secuencia F2 de MADMAX. |

Tabla 5.2: Comparación de resultados de los experimentos 1 y 2 usando el dataset de MADMAX.

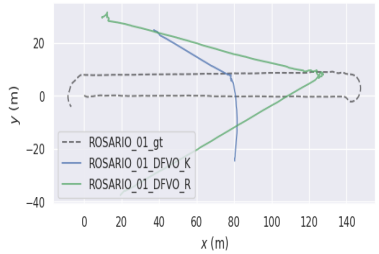
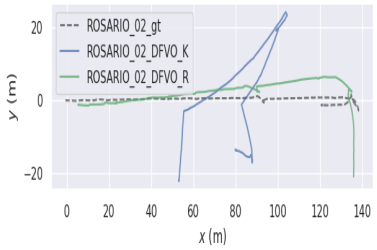
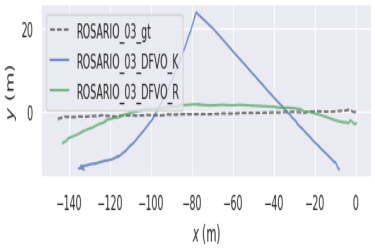
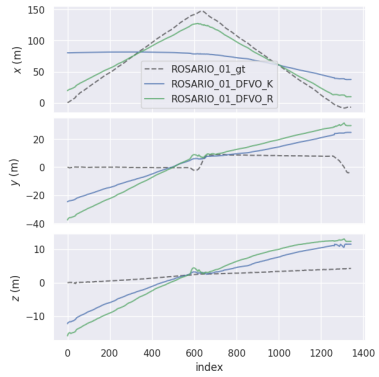
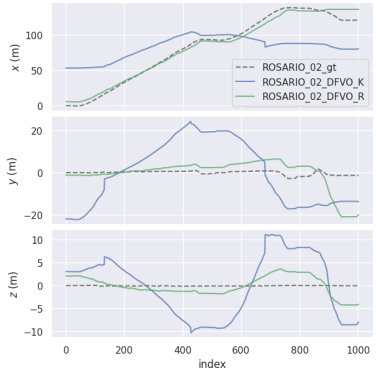
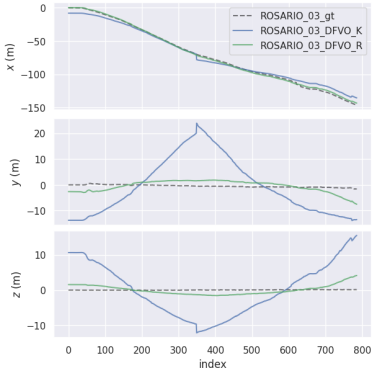
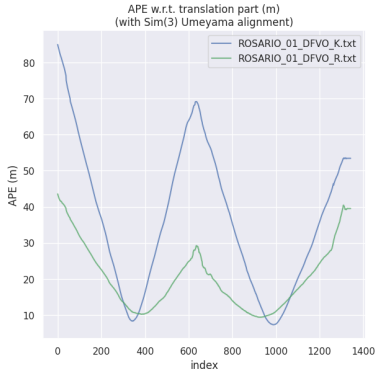
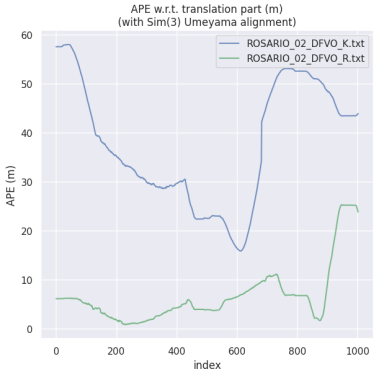
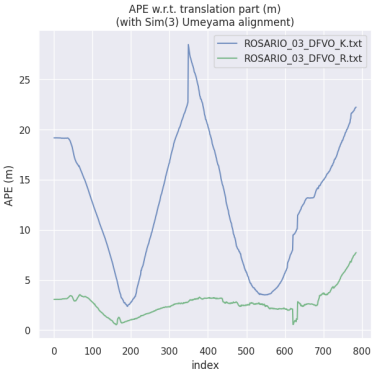
| ROSARIO | | |
|---|---|---|
| **Secuencia 01** | **Secuencia 02** | **Secuencia 03** |
|  |  |  |
| Figura 5.19: Trayectorias de la secuencia 1 de Rosario. | Figura 5.20: Trayectorias de la secuencia 2 de Rosario. | Figura 5.21: Trayectorias de la secuencia 3 de Rosario. |
|  |  |  |
| Figura 5.22: Coordenadas de la secuencia 1 de Rosario. | Figura 5.23: Coordenadas de la secuencia 2 de Rosario. | Figura 5.24: Coordenadas de la secuencia 3 de Rosario. |
|  |  |  |
| Figura 5.25: APE de la secuencia 1 de Rosario. | Figura 5.26: APE de la secuencia 2 de Rosario. | Figura 5.27: APE de la secuencia 3 de Rosario. |

Tabla 5.3: Comparación de resultados de los experimentos 1 y 2 usando el dataset de Rosario.

# Capítulo 6

# Comentarios finales

En este capítulo se presentan las conclusiones a las que se llegaron después de los experimentos realizados, así como las contribuciones de la tesis y una lista de adaptaciones al algoritmo que pueden servir de objetivos para trabajos a futuro.

## 6.1 Conclusiones

El presente trabajo se encargó de evaluar el desempeño del algoritmo de odometría visual DF-VO utilizando secuencias de imágenes de ambientes no estructurados, como los datasets de KITTI, MADMAX y Rosario, con el fin de identificar cómo las características de los entornos influyen en la precisión de las estimaciones de las trayectorias.

Los experimentos mostrados en el Capítulo 4 evidencian que la precisión del algoritmo se ve fuertemente afectada en escenarios no estructurados, como caminos rocosos y desérticos, campos de cultivo o ambientes poco detallados. Por el contrario, en entornos urbanos bien definidos, el algoritmo mostró un desempeño más preciso.

Esta observación permitió identificar que los cambios en la iluminación, similitud entre imágenes consecutivas, falta de puntos característicos, fallos en la captura de las imágenes y el entrenamiento de las redes comprometen la precisión de las predicciones.

A pesar de que en el experimento 2 se logró una reducción en el error para las estimaciones con Rosario, las estimaciones continúan presentando desviaciones significativas, especialmente con MADMAX. Este comportamiento sugiere que hacer modificaciones internas en DF-VO, como el reentrenamiento de monodepth2 con datos específicos, pueden afectar significativamente la calidad de las estimaciones.

## 6.2 Contribuciones

La principal contribución de esta tesis fue la evaluación del algoritmo DF-VO en escenarios no estructurados, como los representados en los datasets de MADMAX y Rosario. Esta evaluación constituye una contribución significativa, dado que DF-VO fue, originalmente, diseñado y optimizado para trabajar con entornos estructurados.

En segundo lugar, se logró una mejora en el desempeño del algoritmo mediante el reentrenamiento del estimador de profundidad utilizado por DF-VO, lo que permitió adaptarlo a las características de los ambientes no estructurados, específicamente a los entornos agrícolas del dataset Rosario. Esta modificación demuestra la importancia del diseño y adaptación de los algoritmos de VO de acuerdo a las características de los escenarios de interés.

Finalmente, a partir del análisis de los resultados obtenidos, se identificaron las modificaciones que podrían hacer que el algoritmo mejore su desempeño. Estas áreas de oportunidad se enlistan en la siguiente sección a modo que sirvan de guía para la continuación de esta línea de investigación.

## 6.3 Trabajo a futuro

Dentro de los posibles trabajos a futuro se encuentran:

- Identificar más algoritmos de VO, de reciente creación y que no hayan sido diseñados para operar con escenarios no estructurados, y realizar una evaluación similar a la presentada en esta tesis, con el fin de aumentar el número de algoritmos candidatos para ser usados en ambientes como los que se encuentran en planetas como Marte.
- Considerar usar datasets distintos que muestren escenarios no estructurados, similares a los de la misión ROBEX, que se enfocó en capturar escenarios análogos lunares en el volcán Etna, en Italia.
- Actualizar la versión de LiteFlowNet que utiliza DF-VO, dado que actualmente existe una versión 3 que podría mejorar el desempeño del algoritmo.
- Realizar el entrenamiento de monodepth2 con imágenes del dataset de MADMAX, o en su defecto, cambiar el estimador de profundidad por uno más robusto, que esté optimizado para trabajar con entornos no estructurados.

# Referencias


[1] M. Maimone, Y. Cheng, y L. Matthies, "Two years of Visual Odometry on the Mars Exploration Rovers", J. Field Robot., vol. 24, núm. 3, pp. 169–186, 2007.

[2] K. Yousif, A. Bab-Hadiashar, y R. Hoseinnezhad, "An overview to visual odometry and visual SLAM: Applications to mobile robotics", Intell. Ind. Syst., vol. 1, núm. 4, pp. 289–311, 2015.

[3] H. Zhan, C. S. Weerasekera, J.-W. Bian, R. Garg, y I. Reid, "DF-VO: What Should Be Learnt for Visual Odometry?", arXiv [cs.CV], 2021.

[4] T.-W. Hui, X. Tang, y C. C. Loy, "LiteFlowNet: A lightweight convolutional neural network for optical flow estimation", en 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018.

[5] C. Godard, O. M. Aodha, M. Firman, y G. Brostow, "Digging into self-supervised monocular depth estimation", en 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 2019.

[6] A. Geiger, P. Lenz, C. Stiller, y R. Urtasun, "Vision meets Robotics: The KITTI Dataset", Int. J. Rob. Res., vol. 32, núm. 11, pp. 1231–1237, 2013.

[7] L. Meyer et al., "The MADMAX data set for visual-inertial rover navigation on Mars", J. Field Robot., vol. 38, núm. 6, pp. 833–853, 2021.

[8] T. Pire, M. Mujica, J. Civera, y E. Kofman, "The Rosario dataset: Multisensor data for localization and mapping in agricultural environments", Int. J. Rob. Res., vol. 38, núm. 6, pp. 633–641, 2019.

[9] V. Romero-Bautista, L. Altamirano-Robles, R. Díaz-Hernández, S. Zapotecas-Martínez, y N. Sanchez-Medel, "Evaluation of visual SLAM algorithms in unstructured planetary-like and agricultural environments", Pattern Recognit. Lett., vol. 186, pp. 106–112, 2024.

[10] V. Romero-Bautista, L. Altamirano-Robles, y R. Díaz-Hernández, "Towards end-to-end visual odometry for unstructured agricultural environments", en Lecture Notes in Computer Science, Cham: Springer Nature Switzerland, pp. 245–256, 2025.

[11] J. Cremona, R. Comelli, y T. Pire, "Experimental evaluation of Visual-Inertial Odometry systems for arable farming", J. Field Robot., vol. 39, núm. 7, pp. 1121–1135, 2022.

[12] H. Zhan, DF-VO: Depth and Flow for Visual Odometry. Recuperado de: https://github.com/Huangying-Zhan/DF-VO.

[13] J. T. W. Hui, LiteFlowNet: LiteFlowNet: A Lightweight Convolutional Neural Network for Optical Flow Estimation, CVPR 2018 (Spotlight paper, 6.6 %). Recuperado de: https://github.com/twhui/LiteFlowNet.

[14] monodepth2: [ICCV 2019] Monocular depth estimation from a single image. Recuperado de: https://github.com/nianticlabs/monodepth2.

[15] M. Grupp, evo: Python package for the evaluation of odometry and SLAM. Recuperado de: https://github.com/MichaelGrupp/evo.

[16] S. Umeyama, "Least-squares estimation of transformation parameters between two point patterns", IEEE Trans. Pattern Anal. Mach. Intell., vol. 13, núm. 4, pp. 376–380, 1991