# EVALUACIÓN DE UN SISTEMA SLAM VISUAL CON SEGMENTACIÓN SEMÁNTICA PARA ENTORNOS DINÁMICOS

**Edison Torres Medina**

**Luis Alejandro Caldas Pinzón**

**UNIVERSIDAD DEL CAUCA**

**FACULTAD DE INGENIERÍA ELECTRÓNICA Y TELECOMUNICACIONES**

**DEPARTAMENTO DE ELECTRÓNICA, INSTRUMENTACIÓN Y CONTROL**

**INGENIERÍA EN AUTOMÁTICA INDUSTRIAL**

**POPAYÁN – CAUCA**

**2023**

# EVALUACIÓN DE UN SISTEMA SLAM VISUAL CON SEGMENTACIÓN SEMÁNTICA PARA ENTORNOS DINÁMICOS

**Monografía presentada como requisito parcial para optar por el título de ingeniero en Automática Industrial**

**Edison Torres Medina**

**Luis Alejandro Caldas Pinzón**

**Director: Elena Muñoz España**

**Codirector: Juan Fernando Flórez Marulanda**

**UNIVERSIDAD DEL CAUCA**

**FACULTAD DE INGENIERÍA ELECTRÓNICA Y TELECOMUNICACIONES**

**DEPARTAMENTO DE ELECTRÓNICA, INSTRUMENTACIÓN Y CONTROL**

**INGENIERÍA EN AUTOMÁTICA INDUSTRIAL**

**POPAYÁN – CAUCA**

**2023**

# Tabla de Contenido

# Índice de Tablas

# Índice de Diagramas

# Índice de Figuras

# Capítulo 1

# Introducción

## 1.1. Resumen de la propuesta

El proyecto propone evaluar el desempeño de un algoritmo SLAM visual de visión artificial para entornos dinámicos que permita la evaluación y reconstrucción de trayectorias con presencia de perturbaciones dinámicas.

El punto de interés en este proyecto es la evaluación de la integración de un sistema vSLAM que combine el algoritmo de seguimiento ORB-SLAM con el uso de filtros de movimiento, y la inclusión de una red de segmentación semántica, utilizando una base de datos pública denominada TUM RGB-D, específicamente la sección de videos en interiores con presencia dinámica, para así poder evaluar el desempeño de esta integración.

El desarrollo del algoritmo ha sido implementado en el lenguaje Matlab usando las herramientas de visión artificial pertenecientes al software. Se recurre al lenguaje Python para la evaluación de los datos estimados en Matlab contra los suministrados por la base de datos para la evaluación.

En el presente trabajo se implementó un algoritmo ORB-SLAM con filtros, ajustes de poses, ajustes de movimiento y finalmente una reproducción del algoritmo con la adición de una red neuronal para segmentación semántica MASK-RCNN con el fin de evaluar la incidencia de la segmentación semántica en la reconstrucción de trayectorias con escenas dinámicas; es necesario aclarar que el algoritmo base ORB-SLAM es tomado directamente de la plataforma virtual de Matlab, denominada mathworks, donde se implementan numerosos algoritmos de patente abierta para pruebas educativas, así como también se extrajo de esta misma manera el recurso correspondiente a la red de segmentación semántica pre-entrenada (MASK-RCNN) para la detección de clases como personas y vehículos; por tanto se identifica como objetivo principal de este trabajo la integración y adaptación de estas dos partes con la finalidad de evaluar los resultados de esta unión durante la reconstrucción de la trayectoria de un objeto situado en un entorno interior con la incidencia de objetos dinámicos. Para corroborar la efectividad de la integración del algoritmo propuesto, este se evalúa sobre seis secuencias de imágenes con diferentes estándares dinámicos comparando los resultados del sistema con y sin la incidencia del método de segmentación semántica; el algoritmo con segmentación presenta mejoras en la evaluación que contiene alta presencia dinámica, sin embargo en ocasiones no se hace suficiente, presentando fallas en el cierre de mapa para obtener los resultados, esto debido a que hay variaciones bruscas

en la exposición luminosa capturada o en la dinámica del sistema, que imposibilita la obtención de características necesarias para continuar la ejecución del sistema.

## 1.2. Estado del Arte

El estado del arte presenta una revisión bibliográfica de investigación con artículos recientes de las técnicas de SLAM visual que se vienen utilizando en entornos de interiores con objetos dinámicos; la información se extrajo mediante el método de revisión sistemática, utilizando las siguientes bases de datos: SciELO, Google Scholar e IEEE. Las palabras clave empleadas para la búsqueda de la información requerida son: *SLAM*, *visual SLAM*, *dynamic SLAM*, *semantic segmentation*, *SLAM in indoor environments*.

Los sistemas de visión artificial son muy utilizados en la actualidad, SLAM es una técnica que permite la reconstrucción de mapas en entornos desconocidos, esto buscando estimar la trayectoria de un robot en cuanto a orientación y translación [1]. Esta técnica tiene muy buenos resultados en la actualidad, siempre y cuando en la escena no se encuentren objetos dinámicos que obstruyan con la estimación de la trayectoria [2]. A consecuencia, el enfoque de esta técnica en investigación está ligado a resolver el problema de los entornos dinámicos.

Davison et al. es uno de los precursores de SLAM visual, en 2007 este propuso un sistema de SLAM monocular en tiempo real, fue conocido como Mono-SLAM [3]. Posterior a Mono-SLAM, Kleint et al. propone PTAM [4], el cual divide el seguimiento y mapeo en dos subprocesos, estos siendo puntos clave en la investigación de los sistemas SLAM. Para el año 2014 LSD-SLAM [5] propone aclarar la relación entre el gradiente de píxel y el método directo, esto en busca de estimar la pose de la cámara directamente de los valores de la intensidad en la imagen, buscando mejorar la consistencia de la reconstrucción a gran escala del entorno. Este mismo año Forster [6], [7], propone aplicar odometría visual monocular semidirecta y rápida, combinando así los puntos característicos con el flujo óptico de seguimiento, con el fin de buscar un sistema SLAM más robusto que el simple método de extracción de puntos característicos, presentando mejores resultados en la estimación de la posición de la cámara.

Con la llegada de la extracción de puntos característicos se desarrollan los algoritmos propuestos por Mur-Artal en todas sus versiones ORB-SLAM [8] en 2015 y ORB-SLAM2 [9] en 2017. Su composición está basada en 3 hilos de trabajo: seguimiento, mapeo local y cierre de bucle [10], [11]. Dichos hilos extraen los puntos característicos, optimizan estos puntos característicos y mejoran el gráfico de pose compuesto por los cuadros clave respectivamente.

Los sistemas ORB-SLAM por sí solos no solucionan el problema de los entornos dinámicos por eso se integra con diferentes técnicas para obtener mejores resultados. Para esto se hace necesario identificar los objetos en movimiento del resto de la escena [12] antes de realizar la estimación de pose en la reconstrucción. La técnica de más relevancia para la identificación de objetos dinámicos es conocida como segmentación semántica, la cual permite segmentar, clasificar y categorizar los objetos detectados que se encuentren en la escena. Sengupta et al [13], incorporan un mapa de octa-árbol en un campo aleatorio de Markov jerárquicamente robusto, de esta manera proporciona

segmentación semántica a cada píxel en el mapa. Vasudevan et al [14] aplican un clasificador bayesiano simple, con el fin de identificar distintas categorías de la escena, dando así una representación probabilística basado en objetos previamente conocidos.

Los resultados obtenidos a partir de la segmentación semántica mejoraron considerablemente con la llegada del aprendizaje profundo, a partir del cual se dio origen a nuevas técnicas de detección de objetos y filtrado [2], dentro de las cuales es importante mencionar la destacada participación de las redes neuronales convolucionales. Modelos destacados como Segnet [15], Dunet [16] y Resnet [17] aportan diferentes técnicas, por ejemplo, la estimación de movimiento entre una imagen u otra para identificar un objeto dinámico [15], [16], optimización de postura para la eliminación de puntos característicos [17] y el uso de restricciones geométricas para la estimación de la trayectoria [18].

Con estas técnicas los investigadores lograron demostrar avances en el uso de SLAM para la reconstrucción y estimación de mapas y trayectorias en entornos dinámicos, llegando al punto de utilizar dichas técnicas de reconstrucción y estimación en tiempo real, reduciendo el impacto de los objetos dinámicos en la escena [2], [19], y demostrando gran robustez en las técnicas empleadas, sin embargo, esto implica un gran uso de recursos computacionales para su ejecución.

A partir de las nuevas herramientas que surgen con la implementación del aprendizaje profundo, se combina el sistema ORB-SLAM con una red de segmentación semántica y flujo óptico [20], [21], buscando reducir la influencia de los objetos dinámicos en la estimación de la posición, dando como resultado un sistema SLAM semántico capaz de determinar si algunos puntos dentro de un objeto segmentado son dinámicos. Eliminando dichos puntos si el objeto es catalogado como dinámico. Sin embargo, los autores resaltan dos factores importantes a tener en cuenta; primero que el tipo de objetos que puede reconocer la red de segmentación semántica se encuentran restringidos y que los resultados obtenidos a partir de la segmentación no son ideales. También se presentan trabajos como [22] que presenta avances en tiempo real haciendo uso de los sensores y IMU de los Xiaomi MI 8 dando como resultado una aplicación denominada CIGRLogger con el objetivo de recopilar imágenes, mediciones de IMU y GPS, procesando dichas imágenes con SLAM visual para posicionamiento continuo.

Durante el transcurso del año 2021 se evidencia un algoritmo de SLAM visual para la problemática de la pandemia [23] aplicando en una escena de hospital al servicio del tratamiento del covid-19, buscando reducir el riesgo médico-paciente, mediante mapeo y reconstrucción para mejorar la eficiencia de la operación hospitalaria, utiliza un gráfico de conocimiento sobre las imágenes para conseguir una mejor relación entre las entidades y la información semántica, la segmentación se basa en algoritmo MASK R-CNN y lo combina con su método de gráfico de conocimiento, mejorando así la respuesta y eliminando los elementos en movimiento de la escena. Los autores especifican que se puede mejorar el sistema cambiando la segmentación ya que esta no tiene buena respuesta en los contornos de los objetos segmentados.

Por último, se observa los resultados de una red neuronal convolucional [16] que implementa un detector de objetos de disparo único SSD (*Single-Shot Detector)* basado en la invariancia de velocidad de cuadros adyacentes para detectar objetos dinámicos, también incorpora un algoritmo

de compensación de detección de pérdida el cual mejora notablemente la precisión en la detección de objetos.

En el último año las mejoras en cuanto a algoritmos de Visual SLAM se han presentado enfocadas en la robustez de la red implementada como lo hacen en [24][25] que implementan una red YOLOv5 que presenta mejoras en la robustez y en el procesamiento haciendo posible un mejor uso, pero eso no es todo, ya que estas redes suelen necesitar altos recursos computacionales se presentan algoritmos que agregan tecnologías MEC (*Mobile Edge Computing*) [26] la cual combinada con redes 5G ultradensas permite que las tareas computacionales complejas en los sistemas SLAM visuales se puedan descargar en servidores informáticos.

De la literatura revisada se concluye, que la precisión de la técnica de segmentación semántica aún no es suficiente para resolver completamente el problema de los sistemas SLAM visual en entornos dinámicos, por lo que se siguen proponiendo distintos métodos, para abordar este problema. En este proyecto se propone evaluar la integración de un algoritmo de invarianza en la velocidad entre imágenes adyacentes basado en [16], con una técnica de segmentación semántica, con el fin de medir la eficiencia mediante la estimación de la trayectoria en entornos dinámicos interiores en SLAM visual.

## 1.3. Objetivos

### 1.3.1 Objetivo General

Evaluar el desempeño de la integración de un método de segmentación semántica y un algoritmo de invarianza de la velocidad entre marcos adyacentes, en un sistema SLAM visual para entornos interiores con objetos dinámicos.

### 1.3.2 Objetivos Específicos

- Definir un sistema ORB-SLAM que realice la extracción de características propias del entorno capaz de integrarse con un método de segmentación semántica.
- Estimar los objetos dinámicos de una escena empleando un método de segmentación semántica y un algoritmo de invarianza de la velocidad.
- Comparar el desempeño de la integración propuesta contra un sistema SLAM visual, mediante la estimación de la trayectoria en entornos dinámicos usando el conjunto de datos TUM RGB-D.

## 1.4. Estructura del documento

Este trabajo se encuentra dividido en cinco capitulo (incluido el presente), los cuales se estructuran de acuerdo con una traza lógica que abarca desde la teoría base del trabajo hasta la implementación y validación del algoritmo propuesto, así como las conclusiones obtenidas de los resultados y posibles trabajos futuros. En el capítulo 2, se consolidan los aspectos claves de forma teórica de tal manera que el lector pueda alcanzar una comprensión detallada del algoritmo desarrollado a través de los capítulos siguientes. En el capítulo 3, se detalla la propuesta fundamental de este trabajo de grado consolidado sobre un algoritmo SLAM visual, así como la descripción y aplicación de los diferentes filtros que implementa el algoritmo y el proceso de integración del método de segmentación semántica, la forma en que este último se desarrolla y los parámetros que utiliza para la consecución de sus fines. Además, se detalla mediante diagramas de flujo la etapa precisa de SLAM sobre la que se implementa la segmentación semántica y la forma en que se lleva a cabo este proceso. En el capítulo 4, se exponen los resultados obtenidos y se ilustran visualmente las diferentes pruebas realizadas sobre seis tipos de secuencias de imágenes diferentes. En el capítulo 5, se plasman las conclusiones obtenidas una vez analizados los resultados y se proponen algunos trabajos futuros para el mejoramiento del desempeño del algoritmo desarrollado.

# Capítulo 2

# Marco Teórico

En este capítulo se plantean las bases teóricas acerca del algoritmo SLAM, sus variaciones más comunes, su estructura general y se describen algunos elementos implícitos en el procesamiento de las secuencias de imágenes a evaluar como por ejemplos filtros para mejorar el rendimiento así como los diferentes tipos de detectores y extractores de características; por otra parte se aborda y se caracteriza los diferentes métodos de procesamiento digital de imágenes orientados a la segmentación desde los más básicos aplicados sobre imágenes binarias, hasta concluir con la segmentación semántica que realiza etiquetado de pixeles para la categorización de las estructuras extraídas.

## 2.1. SLAM

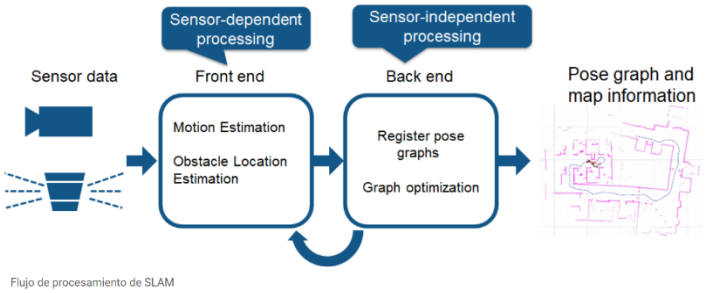
SLAM (*Simultaneous Localization And Mapping)* es un método que permite la localización y creación de mapas simultáneamente, esta técnica se emplea para obtener la estructura tridimensional y el movimiento de un sensor en un entorno desconocido; es muy utilizado para realizar tareas de planificación de rutas, ya que originalmente fue planteado como una propuesta para alcanzar el control autónomo de robots [27].

Con las mejoras tecnológicas y el aumento en la disponibilidad de sensores de bajo costo como cámaras, telémetros y localizadores de posicionamiento global (GPS, *global positioning system)*, SLAM se implementa en un número creciente de aplicaciones como modelos tridimensionales en línea, visualización basada en realidad aumentada (AR, *augmented reality*) [28], automóviles autónomos [29], control de flotas de navegación, entrega de paquetes por dron, entre otros [30].

Para entender mejor el funcionamiento de SLAM se debe dividir en dos partes que diferencian sus componentes tecnológicos como se muestra en la Figura 1, estas son el procesamiento de las señales que se obtienen mediante los sensores implementados al cual se le denomina procesamiento frontal y (*front end*) la segunda parte que se encarga de la optimización de gráficos de posición, incluyendo el procesamiento final (*back end*) que es independiente de los parámetros obtenidos por los sensores.

Al interior del procesamiento frontal ilustrado en la Figura 1, es posible discriminar dos métodos diferentes de SLAM, SLAM visual [31] y SLAM por detección y distancia con luz (LIDAR, *light detection and ranging*) [32]; la principal diferencia entre estos dos métodos radica en la cantidad de sensores asociados para la toma de datos adicionales.

Figura 1. Flujo de procesamiento de SLAM




*Fuente:* Adaptado de *SLAM (Localización y mapeo simultaneo)* [Diagrama], por mathworks, 2022, (https://la.mathworks.com/discovery/slam.html)

El LIDAR es una tecnología de teledetección que utiliza pulsos de luz para obtener mediciones de distancia (profundidad) a los objetos del entorno. Este método se caracteriza por el uso de un sensor láser o de distancia para obtener información respecto de la profundidad, la cual se calcula mediante el tiempo de retraso entre la emisión del pulso y su detección a través de la señal reflejada por el haz de luz.

Mediante el LIDAR se obtienen resultados de alta precisión que se ajustan bastante bien a la realidad, esto lo ha posicionado como una de las principales opciones de implementación en sistemas de alta precisión como la conducción autónoma de vehículos o los sistemas de orientación para vehículos aéreos no tripulados. La principal dificultad de su ejecución reside en un costo elevado de implementación, así como también presenta una alta demanda de recursos computacionales para su correcto procesamiento.

## 2.2. SLAM visual

Este método utiliza imágenes captadas mediante cámaras y otros sensores de imagen, (cámaras ojo de pez, esférica y objetivos de gran angular, estereoscópicas y multicámaras e incluso cámaras de profundidad) con la finalidad de reconstruir el mapa de un entorno desconocido además de estimar la trayectoria realizada por la cámara durante la captura de la secuencia de imágenes [31].

SLAM visual nace con el objetivo de aplicarse en la realidad aumentada, esta tecnología se basa en el volumen de información capaz de ser proporcionado por una cámara, siendo información útil para la detección de puntos de referencia (posiciones medidas previamente); se habla de SLAM monocular cuando toda su información recae en una única cámara como sensor [33], presentando complicaciones a la hora de obtener información de la profundidad de la escena. Este problema se resuelve detectando marcadores de realidad aumentada [34], estructuras a partir del movimiento (SFM, *structure from motion*), odometría visual y ajustes de paquetes [35], que detectan objetos en la escena para estimar la localización; también se le combinan otros sensores como las unidades

de medición inercial (IMU, *inertial measurement unit*) [36], que pueden medir directamente magnitudes físicas como la velocidad y orientación.

Se presentan dos categorías en el SLAM visual, una de ellas está compuesta por los métodos densos que utilizan la totalidad del brillo de la imagen para estimar la escena con algoritmos como Seguimiento Denso y Mapeo en Tiempo Real (DTAM, *dense tracking and mapping in real-time)* [37], SLAM monocular directo a gran escala (LSD-SLAM, *large-scale direct monocular SLAM)* [38], Odometria Dispersa Directa (DSO, *direct sparse odometry*) [39] y Odometría Visual Inercial Semidirecta (SVO, *semi-direct visual-inertial odometry*)[40].

Por otro lado, se encuentran los métodos dispersos que identifican correspondencias entre los puntos característicos de las imágenes y utilizan algoritmos como Seguimiento y Mapeo Paralelo (PTAM, *parallel traking and mapping*) [41] y SLAM con Orientación Rápida y Giro (ORB-SLAM, *oriented fast and rotated*)[42].

### 2.2.1. Detectores y Extractores de características para SLAM visual

Para la identificación de correspondencias entre puntos característicos es necesario realizar los procesos de detección y extracción de los mismos a través de un método disperso; los detectores de características como SURF (*speeded-up robust features*) y SIFT (*scale invariant feature transform*) [43] se consolidan como pioneros en el área de extracción de características para algoritmos de visión artificial, estos extractores utilizan una técnica llamada multi-resolución donde se transforma la imagen a coordenadas, la cual consiste en hacer una réplica de la imagen original de forma piramidal Gaussiana o piramidal Laplaciana y así obtener imágenes del mismo tamaño pero con el ancho de banda reducido, logrando un efecto de borrosidad que asegura que los puntos de interés sean invariantes en el escalado. Es importante mencionar que el extractor SURF es una actualización de SIFT, donde se optimiza el esquema de aplicación para obtener una mayor velocidad de procesamiento y mejores resultados trabajando sobre imágenes con invarianzas en el escalado.

### 2.2.2. Extractor ORB

Otro método disperso y hoy en día el más investigado y diversificado en cuanto a su aplicación sobre algoritmos SLAM es el extractor ORB [42], el cual presenta una mayor eficiencia respecto al consumo de recursos frente a sus predecesores haciendo de este extractor un ideal para aplicaciones en tiempo real. La popularización de ORB nace gracias a que inicialmente ofrecía una potencial ventaja alrededor de su uso de forma gratuita, dejando de lado el costo asociado de las patentes y permitiendo llegar a más usuarios; a pesar de que su velocidad de procesamiento lo convierte en un extractor ideal para las aplicaciones en tiempo real esto solo es posible sacrificando calidad en el descriptor empleado, el cual tan solo cuenta con treinta y dos bits de resolución, lo que se consolida en varios impedimentos como la presencia de dificultades para encontrar puntos

de interés en imágenes con espectro visible e infrarrojo cercano y la alta sensibilidad a las acciones de reescalado en las imágenes procesadas [44].

La creación del extractor ORB parte de la fusión del algoritmo de FAST (*Features from accelerated segment test*) y el algoritmo descriptor de características BRIEF (*Binary Robust Independent Elementary Features* ) [27][45]. El primero es una técnica que localiza los bordes y esquinas, recibiendo como parámetro el umbral de la diferencia de intensidad entre el píxel central y aquellos situados en un círculo alrededor del centro; debido a que no se cuenta con una medida cuantificable para determinar si un punto puede ser considerado una esquina, se producen altas respuestas a lo largo de los bordes y las esquinas en la imagen. Para solucionar esto se busca apoyo en la técnica de detección de esquinas de Harris la cual ordena los puntos de acuerdo con su relevancia y tan solo tiene en cuenta un número determinado de ellos que se seleccionan según el ponderado establecido [46]. Como dato adicional el método FAST no produce características multiescala, buscando suplir este problema se emplea un método piramidal (similar al aplicado en el extractor de características SURF) de donde se extraen las características filtradas previamente por la medida de Harris.

Por otro lado, el descriptor BRIEF es una descripción de la cadena de bits para la representación binaria de una imagen. Este se construye a partir de un conjunto de pruebas de intensidad binaria y se caracteriza por requerir muy poco cómputo, así como memoria de almacenamiento; al ser una descripción de la cadena de bits para la representación binaria de una imagen que compara la intensidad del píxel con una versión suavizada de esta misma, se presenta una gran variación si no se orienta a lo largo de la dirección del punto clave [47].

### 2.2.3. Filtros y ajustes de movimiento

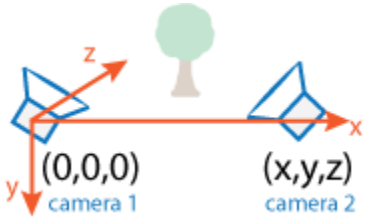
Durante la extracción de características asociadas mediante puntos de mapa a cada uno de los cuadros claves, es frecuente que dicha asociación no se realice de manera completamente acertada, es decir puede presentar pequeños desfases que crecen continuamente a lo largo del tiempo, puesto que el procesamiento de las imágenes es secuencial; de este desface conocido como error fotométrico se desprende un concepto más común en los sistemas SLAM conocido como error de reproyección, que nace a partir de la proyección en el espacio del error fotométrico. Con la finalidad de minimizar el error de reproyección de las posiciones relativas de la cámara [48], un algoritmo SLAM implementa uno o varios filtros de ajustes de movimiento; dichos filtros se encargan de tareas influyentes sobre la reconstrucción del mapa y estructuras 3D mediante el refinamiento de puntos tridimensionales y poses de cámara, optimización del gráfico de poses de acuerdo con sus restricciones de borde definidas y correspondencia de puntos sobre imágenes bidimensionales.

Debido a la naturaleza de las escenas empleadas para la implementación del sistema SLAM las cuales cuentan con objetos dinámicos influyentes para la reconstrucción de mapa, se puede implementar un filtro de estructura a partir del movimiento [49] el cual busca minimizar la repercusión de los objetos dinámicos sobre el resultado de la aplicación de SLAM.

La aplicación de este filtro tiene como objetivo, en primera instancia, la reconstrucción de un objeto tridimensional a partir de un conjunto de imágenes en dos dimensiones. Este procedimiento se puede implementar de distintas maneras teniendo en cuenta el número de cámaras y los sensores presentes durante la obtención de la información.

Para el caso particular de ORB-SLAM monocular, sólo se cuenta con una cámara sobre la cual se implementa una toma de dos vistas; debido a esto la estructura tridimensional y la trayectoria de la cámara solo se puede recuperar a escala. Si se quisiera obtener la escala real se hace necesario la implementación de un equipo que brinde más información sobre la escena como por ejemplo, un odómetro y para el caso simple donde se cuenta con una cámara en movimiento el sistema asume que la cámara uno es el origen del sistema de coordenadas y su eje óptico se encuentra a lo largo del eje z como se muestra en la Figura 2.

Figura 2. Estructura de movimiento de dos vistas




Fuente: Adaptado de *Structure from motion Overview* [Figura], por Mathworks, 2022, (https://la.mathworks.com/help/vision/ug/structure-from-motion.html)

Haciendo uso de las vistas proporcionadas por la cámara en movimiento ORB-SLAM implementa una función capaz de detectar puntos coincidentes entre ellas como se muestra en la Figura 3, esto lo hace a partir de las características propias extraídas mediante el uso de la información brindada por los píxeles circundantes y el algoritmo de Harris que se encarga de la detección de esquinas [46]. La función también considera el origen tridimensional del sistema de coordenadas a partir de la posición original de la cámara y establece un método para hallar la posición previa al desplazamiento; dicha posición se halla y se expresa en términos del sistema de coordenadas anteriormente normalizado.

Como se mencionó, la información obtenida por este método es idónea en todos sus parámetros a excepción de la escala, por esta razón la distancia entre las cámaras se fija en un valor unitario; es necesario recordar que, aunque solo se establece el uso de una cámara, para efectos prácticos cada una de las vistas previas a un desplazamiento es considerado como una nueva cámara.

Figura 3. Correspondencias de puntos coincidentes entre dos marcos

*Fuente:* Adaptado de *TUM School of Computation, Information and Technology*[Fotografía], por Computer Vision Group,2022.

Este enfoque de estructura a partir de movimiento puede ser llevado a un nivel más alto de complejidad donde se tienen en cuenta múltiples vistas, caso que resulta más aterrizado a las aplicaciones del mundo práctico, al igual que en el modelo anteriormente mencionado es posible hallar la pose de la cámara 3 en relación con la cámara 2 y así sucesivamente para determinar las poses de todas las cámaras consideradas en la escena. De igual manera las poses relativas de todas las cámaras deben transformarse en un sistema de coordenadas común y por lo general todas las poses de las cámaras se calculan en relación con la cámara 1.

Cada estimación de pose de una cámara de una vista a la siguiente contiene errores que pueden surgir de la localización imprecisa de puntos en las imágenes, coincidencias ruidosas o calibraciones imprecisas de las cámaras; estos errores se acumulan a medida que aumenta el número de vistas percibidas generando un efecto conocido como *drift* o desviación. La aplicación de un filtro para optimizar las posiciones de la cámara y refinar la posición de los puntos tridimensionales [50] se aplica con el fin de reducir la desviación percibida en las trayectorias reconstruidas [51].

### 2.2.4. Estructura general del algoritmo SLAM visual

Al ser un método visual, la información recae en la adquisición de puntos clave mediante un extractor de características, el cual se ejecuta desde el inicio de su estructura, ayudando a definir un sistema de coordenadas iniciales y estimando la pose inicial de la cámara además de la reconstrucción tridimensional del entorno desconocido (inicialización del mapa). Una vez inicializado el mapa se realiza el seguimiento y mapeo simultáneo, este es fundamental en el SLAM ya que se ejecuta constantemente en el sistema; mediante este procedimiento se busca estimar la pose de la cámara utilizando las coincidencias de características extraídas previamente,

además dichas coincidencias derivan en la creación de puntos al interior del mapa. Tanto la estimación de la pose como la creación de puntos del mapa tridimensional se expanden a medida que la cámara detecta nuevas regiones desconocidas que hacen parte de un entorno general, este procedimiento es continuo mientras se continúe encontrando características coincidentes.

Para la última etapa, la cual se encarga de establecer el cierre de mapa, en los sistemas SLAM visual basados en métodos dispersos de identificación de características, se emplea el uso de una Bolsa de Objetos de palabras Visuales (*Bag of features),* este es un método usado para representar las características de las imágenes, es decir un algoritmo de extracción, generación y representación de características; basado en el conjunto de imágenes se extrae las características haciendo uso de un descriptor, posteriormente se genera el libro de códigos que representa cada una de estas características extraídas para que puedan ser etiquetadas y representadas mediante la creación de un vocabulario visual, construyendo un vector de características necesario para la reproducción de la imagen de prueba o consulta. En la etapa de cierre de mapa, este vector de características representado a partir del conjunto de imágenes se emplea como consulta para comparar cada imagen procesada por el sistema, buscando coincidencias semejantes las cuales sean influyentes para iniciar un cierre de bucle.

## 2.3. Procesamiento digital de imágenes

El procesamiento digital de imágenes (PDI) ha adquirido en los últimos años una creciente presencia en los entornos comunes a cualquier persona a través de las diferentes tecnologías que se encuentra diariamente a su alrededor; de este modo es muy común ver la incidencia de dichos procesos digitales en los diferentes campos de la ciencia como por ejemplo en la medicina, donde el procesamiento de imágenes es una herramienta singular para la detección y el diagnóstico de patologías o en el campo de la astronomía donde toda la información obtenida del espacio exterior es enviada de forma digital y debe analizarse para extraer los datos de mayor relevancia captada por las diferentes sondas espaciales y satélites que orbitan el planeta tierra[52].

El PDI debe en gran medida su expansión, prácticamente a todos los campos de la ciencia, gracias a la popularización de las computadoras personales que cada vez son más asequibles al mercado y ha permitido la exploración y experimentación de esta tendencia sobre diversos escenarios, dando como resultado una amplia gama de aplicaciones prácticas.

Los procedimientos que se engloban dentro del amplio concepto del PDI se basan en su totalidad en técnicas matemáticas y variaciones especiales de las mismas como por ejemplo el álgebra lineal, las regresiones aritméticas, el cálculo derivativo e integral además de técnicas estadísticas y probabilísticas; lo que proporciona resultados bien sustentados algorítmicamente, consecuentes, y lo más importante es que son parametrizados, característica que permite ajustar los procedimientos de procesamiento para conseguir los diferentes resultados deseados. De aquí que el PDI cuente con diferentes métodos y técnicas [53] para la consecución de los principales objetivos que son por lo general la extracción de características o el reconocimiento de figuras, la reconstrucción o restauración de imágenes y la interpretación física [54].

En los últimos años la ciencia de las tecnologías ha avanzado a gran velocidad, esto ha repercutido en eventos tales como que las computadoras aumentan su rendimiento desencadenando con esto la mejora en el manejo de datos y la popularización de muchos procesos computacionales. El tratamiento digital de imágenes es uno de estos, el cual ha sido empleado en gran medida sobre sistemas de visión artificial con la finalidad de obtener información basada en los datos extraídos de las imágenes, llevando toda la información a un plano matemático que proporciona la capacidad de modelar desde sistemas sencillos [55] hasta algunos muy complejos [56].

Según IBM [57] la visión artificial se define en tres aspectos muy importantes:

- La visión como un proceso computacional: Pretende dotar a los ordenadores de la capacidad de simular la visión humana.
- La descripción de datos a obtener depende del observador: Su objetivo es proporcionar la descripción y reconocimiento de la imagen debido a que cada descriptor o proceso de caracterización no posee la misma relevancia respecto a diferentes imágenes.
- La reducción de información: Este aspecto es muy importante ya que al reducir la información obtenida de la imagen se optimizan los procesos de interés desarrollados a partir de dichos datos.

Antes de comprender el funcionamiento de la visión artificial y sus diferentes aplicaciones es necesario contar con la información previa acerca de que es una imagen para un ordenador y como se representa, de igual modo algunas de las formas en que se lleva a cabo el tratamiento de datos.

Una imagen es una representación visual de un objeto iluminado por una fuente radiante, las que se perciben en las actividades visuales cotidianas provienen normalmente de la luz reflejada por los objetos. La naturaleza básica de una imagen está caracterizada por dos componentes: La cantidad de luz incidente que procede de la fuente de la escena contemplada y la cantidad de luz reflejada por los objetos de la escena. Dichas componentes reciben el nombre de iluminación y reflectancia, a partir de estos elementos es posible obtener la información total de la imagen mediante un proceso de combinación como el producto de ambas [58].

En el proceso de formación de la imagen intervienen elementos básicos como el objeto, la fuente radiante y el sistema de formación de la imagen que consiste básicamente en un sistema óptico, un sensor y un digitalizador. La imagen digital se representa por una matriz como la que se observa en la Figura 4, donde cada elemento o píxel, representa la intensidad en un punto específico en la imagen.

Figura 4. Representación digital de una imagen en una matriz NxM

$$f = \begin{bmatrix} f(1,1) & f(1,2) & \cdots & f(1,M) \\ f(2,1) & f(2,2) & \cdots & f(2,M) \\ \vdots & \vdots & \cdots & \vdots \\ f(N,1) & f(N,2) & \cdots & f(N,M) \end{bmatrix}$$

*Fuente:* Adaptado de *Técnica y algoritmos básicos de visión artificial* (p.12), por Ana Gonzales Marcos, 2006.

Durante la formación de imágenes y el tratamiento digital de estas se presentan algunos fenómenos como la cuantificación; este efecto viene dado por la imposibilidad de tener un rango infinito de valores de medida de intensidad de brillo en los píxeles. Las imágenes deben pasar por un proceso de escalado de grises, donde la imagen inicial se encuentra en formato de composición de colores (RGB, *red, green, blue)* y es transformada a una imagen en escala de grises, pero la intensidad de esos grises viene dada por la tecnología implementada ya sea de 10 bits, o la más general 8 bits, esto equivale a que se cuenta tan solo con 256 niveles de grises para codificar la intensidad lumínica [59]. Por ejemplo, en la Figura 5 se puede apreciar el impacto de la cuantificación a medida que se comprime el rango de números finitos o pesos que se le asigna a cada uno de los pixeles, pasando de la imagen del lado izquierdo, la cual cuenta con un rango relativamente amplio de matices grises, a la del lado derecho que a simple vista se encuentra solamente a dos tonos: blanco y negro.

Figura 5. Efecto de la cuantificación en la transformación de una imagen a blanco y negro

Fuente: Adaptado de Técnica y algoritmos básicos de visión artificial (p.12), por Ana Gonzales Marcos, 2006.

La comprensión de las bases teóricas al respecto de la formación de imágenes es fundamental en el entendimiento hacia las aplicaciones de la visión artificial y sus diferentes componentes; retomando el eje central del campo de conocimiento a tratar, se puede decir al respecto que la visión artificial tiene una gran acogida en la sociedad actual, haciéndose presente en un sin número de aplicaciones cotidianas como por ejemplo la conducción autónoma, la detección de productos defectuosos en las líneas de manufactura, entre otras muchas más.

Los diferentes algoritmos que se albergan al interior de la visión artificial se centran en la detección y extracción de características a partir de una o varias imágenes.

Teniendo como base la descripción obtenida en [60] es posible dividir el procesamiento de imágenes en las siguientes técnicas principales:

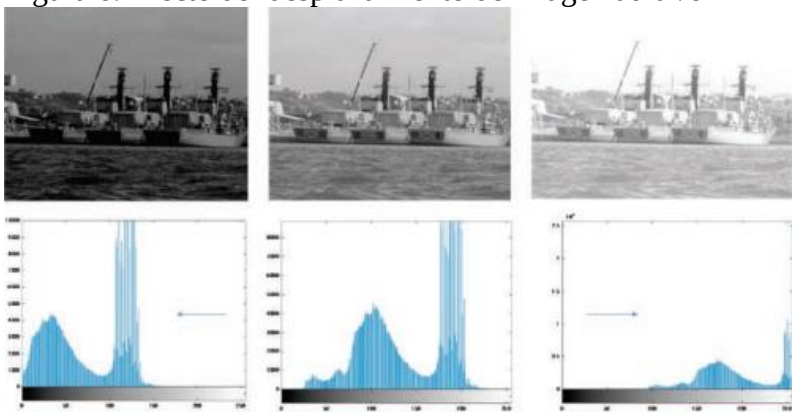
- Adquisición

- Realce y mejora
- Segmentación
- Extracción de características

Es importante tener en cuenta que el PDI varia las metodologías empleadas de acuerdo al tipo de imagen que se desea procesar, es decir, existen imágenes representadas en escalas de grises donde cada uno de sus pixeles estará compuesto solo por los colores blanco y negro y cada uno de ellos varia su intensidad para dar forma a la imagen en conjunto; por otra parte están las imágenes representadas a color o Rojo, Verde, Azul (RGB, *red-green-blue*) las cuales ilustran las imágenes en una amplia gama de colores para cada uno de sus pixeles, colores producto de la combinación de la intensidad de los diferentes colores primarios. De acuerdo con [61] dentro de las técnicas de procesamiento para las imágenes en escala de grises se encuentran algunas como:

- Desplazamiento aditivo de la imagen: Durante la aplicación de esta técnica se define un desplazamiento sobre el histograma de la imagen representada en escala de grises, este desplazamiento afecta la imagen de forma tal que se logra apreciar en la Figura 6, donde se muestra el efecto del desplazamiento del histograma sobre la luminosidad de la imagen, durante el desplazamiento del histograma a la izquierda la imagen se oscurece y en el desplazamiento a la derecha la imagen toma mayor luminosidad, esta técnica es importante para la visualización de detalles que no son perceptibles a simple vista con la escasa luminosidad que originalmente se capturan algunas imágenes.

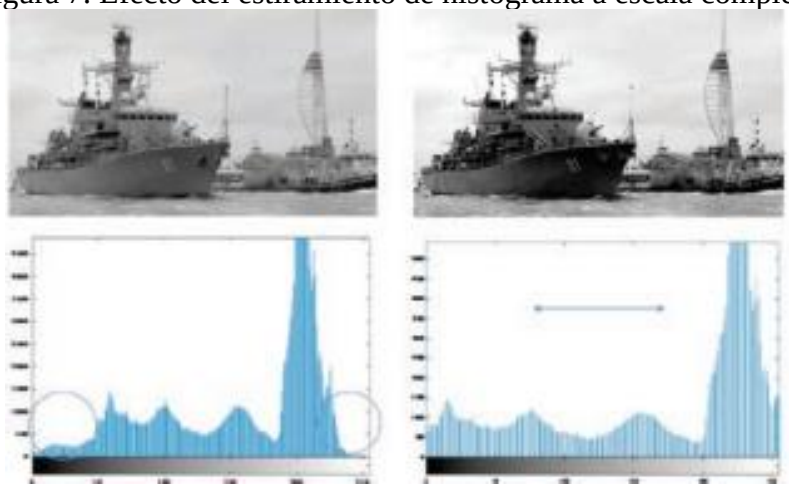
Figura 6. Efecto del desplazamiento de imagen aditivo



*Fuente:* Adaptado de *Revista de Marina N° 969* (p. 69), por Andrés Catalán Urzúa, 2019.

- Estiramiento de histograma a escala completa: Esta técnica consiste en expandir el histograma de la imagen a todo el rango disponible de la escala de grises, proporcionando un mejor contraste como se observa en la Figura 7.

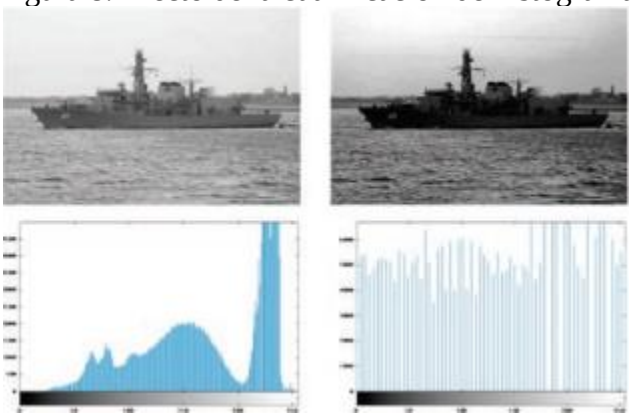
Figura 7. Efecto del estiramiento de histograma a escala completa



*Fuente:* Adaptado de *Revista de Marina N° 969* (p. 70), por Andrés Catalán Urzúa, 2019.

- Ecualización de histograma: Esta operación de punto no lineal es una variación de la técnica de estiramiento de histograma, donde no solo se expande longitudinalmente el histograma a través de todo el rango disponible para escala de grises, sino que además lo reparte proporcionalmente dentro del mencionado rango.

Figura 8. Efecto de la cualificación de histograma

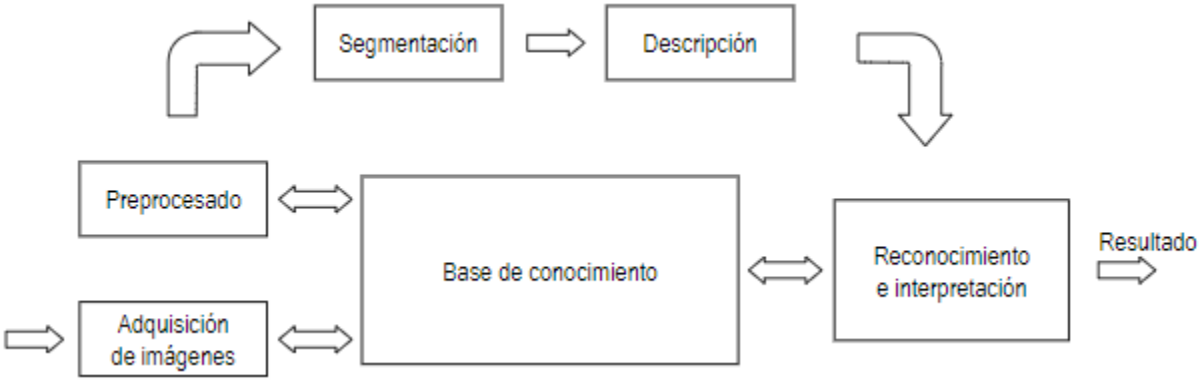


*Fuente:* Adaptado de *Revista de Marina N° 969* (p. 70), por Andrés Catalán Urzúa, 2019.

A partir de las técnicas de procesamiento para imágenes en escala de grises se desglosan innumerables desarrollos de técnicas adaptadas y aplicadas a las imágenes RGB, puesto que éstas proporcionaron una base sólida dentro del PDI. Las técnicas de procesamiento de imágenes aplicadas hoy día son más robustas y presentan una mayor complejidad, (al punto de que han constituido por sí mismas un campo del conocimiento independiente como lo es la visión artificial, la cual representa una de las bases fundamentales de la automatización). Dentro de las técnicas para el PDI se encuentra un grupo de ellas dedicadas a lo que se conoce como segmentación de imágenes, la cual consiste en discriminar objetos o regiones de una imagen del resto de sus componentes con el fin de obtener información puntual de una zona capturada dentro de la misma,

la segmentación constituye una parte fundamental del proceso básico del PDI [62], proceso que se muestra en la Figura 9.

Figura 9. Etapas del procesamiento digital de imágenes




*Fuente:* Adaptado de *Revista de investigación de sistemas e informática Vol. 6*(p. 10), por Nora Palomino e Ulises Román Concha, 2009.

En la Figura 9 se muestran las etapas necesarias que se deben seguir para realizar el procesamiento de imágenes. El proceso se inicia con la etapa de adquisición de imágenes, en donde se requiere de un sensor de imágenes (una cámara por lo general) cuyas señales producidas deben ser digitalizadas. Por ejemplo, se utilizan la luz para la fotografía, rayos X para la radiografía, ultrasonido para la ecografía, entre otros [63]. La naturaleza del sensor dependerá del tipo de aplicación que se quiera estudiar.

La primera etapa es el preprocesamiento, que se realiza con el fin de detectar y eliminar las fallas que puedan existir en la imagen para mejorarla. Las técnicas más utilizadas en esta etapa son:

- Mejora del contraste
- Eliminación del ruido
- Restauración.

En la siguiente etapa llamada segmentación, la imagen se divide en sus partes constituyentes u objetos con el fin de separar las partes necesarias de procesamiento del resto de la imagen que no interesan de acuerdo con la aplicación que se quiera dar. Las técnicas básicas en esta etapa son aquellas orientadas a:

- El píxel
- Los bordes
- Las regiones.

Sin embargo, las técnicas no son excluyentes, sino que se combinan de acuerdo con tipo de aplicación. Durante la etapa de descripción o extracción de características se realiza la extracción

de características que presente información cuantitativa de interés o que sean fundamentales para diferenciar una clase de objetos de otra.

Luego, la etapa de reconocimiento es el proceso que asigna una etiqueta a un objeto basándose en la información proporcionada por sus descriptores. La interpretación implica asignar significado a un conjunto de objetos reconocidos.

Finalmente, la etapa base de conocimiento, es aquella donde se va almacenar el dominio del problema para guiar la operación de cada módulo de procesamiento, también controla la interacción entre dichos módulos.

## 2.4. Segmentación de imágenes

La segmentación se basa en subdividir una imagen en formas, contornos, texturas, curvas o demás patrones que resulten de ayuda para el alcance de un objetivo preciso, establecido previamente; estos objetivos suelen caracterizarse por determinar la profundidad con la que se debe realizar la segmentación sobre la imagen, es decir que de acuerdo con el nivel de detalle demandado para la extracción de características, la segmentación puede establecerse más fina o minuciosa. El proceso de segmentación sobre una imagen se lleva a cabo siguiendo algunos lineamientos comunes en la mayoría de los casos, estos determinan la lógica con la que se discriminan los diferentes patrones al interior de la imagen como lo describe [62].

Los algoritmos de segmentación de imágenes monocromáticas generalmente se basan en una de las dos propiedades básicas de los valores del nivel de gris: discontinuidad y similaridad [58]. En la discontinuidad el método consiste en dividir una imagen basándose en los cambios bruscos del nivel de gris. Los temas más importantes en la discontinuidad son:

- Detección de puntos aislados
- Detección de líneas
- Detección de bordes de una imagen.

En la similaridad, se presenta la regularidad en los valores del nivel de gris, los principales métodos están basados en:

- Umbralización
- Crecimiento de región
- División y fusión de regiones

### 2.4.1. Algoritmos de segmentación basados en la propiedad de discontinuidad

- Detección de puntos aislados: Un punto aislado de una imagen presenta una intensidad en su tonalidad que difiere en gran medida de sus 8 píxeles circundantes en su entorno de 3x3, una máscara Laplaciana para la detección de un píxel aislado podría ser la que se muestra en la Tabla 1.

Tabla 1. Máscara Laplaciana para la detección de un punto aislado

| | | |
|---|---|---|
| -1 | -1 | -1 |
| -1 | 8 | -1 |
| -1 | -1 | -1 |

*Fuente:* propia

Es posible afirmar que un píxel es un punto aislado, si el resultado de la aplicación de la máscara sobre el píxel en valor absoluto es mayor que el valor de umbralización y se establece que se encuentra en una región de intensidad uniforme si el resultado de aplicación de la máscara es cero [64].

- Detección de líneas: El interés de detectar líneas en determinada dirección sobre una imagen [65] demanda una mayor complejidad que hacerlo sobre simples píxeles; este procedimiento es posible mediante la aplicación de una máscara específica que causa una reacción extrema sobre los píxeles que componen un tipo particular de línea, ya sean líneas horizontales como se muestra en la Tabla 2, verticales (Tabla 3) o diagonales (Tabla 4 y Tabla 5).

Tabla 2. Máscara Laplaciana para la detección de líneas horizontales

| | | |
|---|---|---|
| -1 | -1 | -1 |
| 2 | 2 | 2 |
| -1 | -1 | -1 |

Fuente: propia

Tabla 3. Máscara Laplaciana para la detección de líneas verticales

| | | |
|---|---|---|
| -1 | 2 | -1 |
| -1 | 2 | -1 |
| -1 | 2 | -1 |

Fuente: propia

Tabla 4. Máscara Laplaciana para la detección de líneas diagonales de -45°

| | | |
|---|---|---|
| -1 | -1 | 2 |
| -1 | 2 | -1 |
| 2 | -1 | -1 |

Fuente: propia

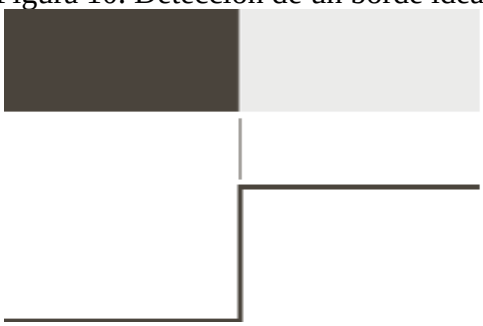
Tabla 5. Máscara Laplaciana para la detección de líneas diagonales de +45°

| | | |
|---|---|---|
| 2 | -1 | -1 |
| -1 | 2 | -1 |
| -1 | -1 | 2 |

Fuente: propia

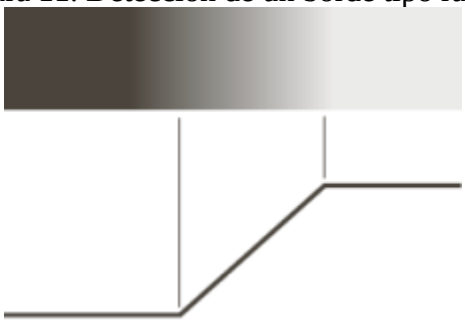
- Detección de bordes: Es la aproximación más común en la detección de discontinuidades en la intensidad de la imagen [66]; para llevar a cabo este procedimiento es necesario valerse de herramientas como la primera derivada (también conocida como el gradiente) y la segunda derivada. El gradiente es cero en zonas donde la intensidad de la imagen es constante y los valores son proporcionales al cambio en la intensidad de los píxeles donde los valores de estos son variables, esto se conoce como detección de un borde ideal y se ilustra en la Figura 10; por otro lado es posible obtener algo más cercano a la realidad como se ilustra en la Figura 11 donde se representa la detección de un borde tipo rampa, aquí el gradiente es cero es zonas donde la intensidad de la imagen es constante pero los valores no son proporcionales uniformemente al cambio de la intensidad de los pixeles donde los valores de estos son variables. Además, una de las características del gradiente es que siempre apunta en la máxima dirección de cambio que se presenta en la imagen. La segunda derivada en el procesamiento de imágenes generalmente se computa usando el Laplaciano en dos dimensiones, es necesario tener en cuenta que esta herramienta resulta muy sensible al ruido. En general, para la detección de un borde se deben seguir dos pasos elementales: el primero es hallar los lugares de la imagen donde la primera derivada es mayor que un umbral y el segundo es encontrar los puntos donde la segunda derivada presenta un cruce por cero.

Figura 10. Detección de un borde ideal



Fuente: propia.

Figura 11. Detección de un borde tipo rampa

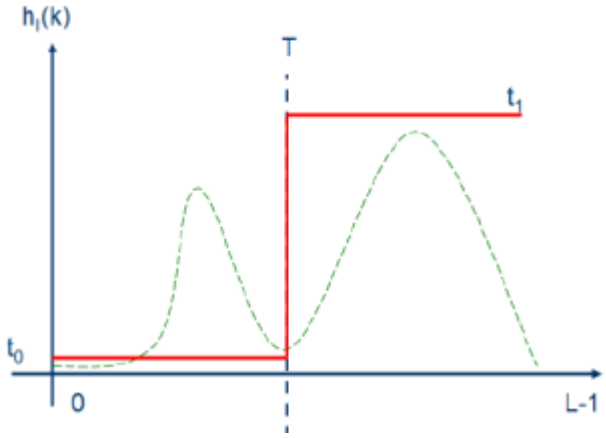


*Fuente:* propia.

## 2.4.2. Algoritmos de segmentación basados en la propiedad de similitud

- Umbralización: La umbralización es una técnica de segmentación simple y eficiente que permite separar los pixeles de una imagen en escala de grises en dos categorías a partir de un valor umbral de intensidad. En la Figura 12 se ilustra un ejemplo de histograma bimoidal el cual es de gran utilidad en el proceso de umbralización de una imagen, en esta grafica el valor definido por 'T' representa el valor de intensidad umbral, y t0 y t1 son los dos posibles niveles de gris de los pixeles de la imagen umbralizada [67].

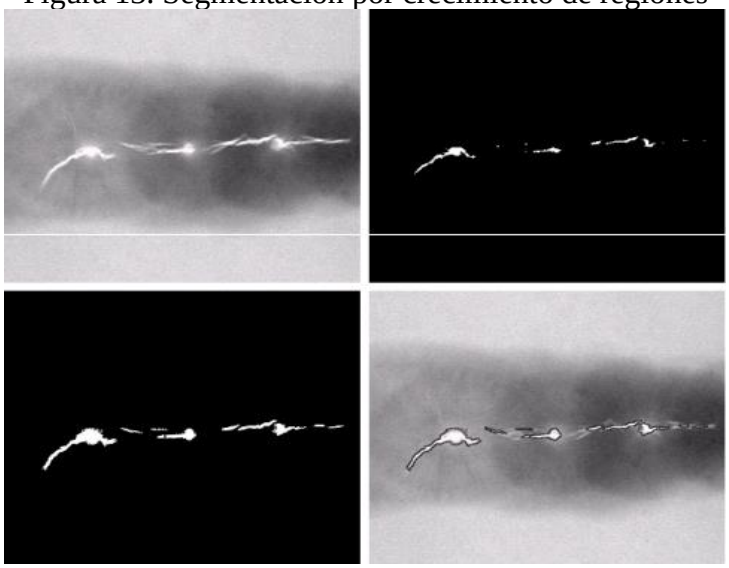
Figura 12. Histograma bimodal




Fuente: Adaptado de Técnicas de umbralización para el procesamiento digital de imágenes de GEMFoils(p.355), por Nayid Triana, 2016.

- Crecimiento de región: Este procedimiento agrupa los píxeles de una imagen en regiones de acuerdo con un criterio fijo preestablecido [68]; normalmente se inicia el proceso a partir de puntos semilla con los que se busca forma una determinada región de segmentación en la imagen añadiendo píxeles colindantes que cumplan con el criterio de selección establecido como se muestra en la Figura 13 donde se analiza el comportamiento de la segmentación en 4 tipos de pruebas diferentes variando los puntos semillas utilizados en cada una de ellas. Para llevar a cabo esta técnica se debe tener en cuenta la información sobre adyacencia y conectividad de la imagen.

Figura 13. Segmentación por crecimiento de regiones



*Fuente:* propia

- División y fusión de regiones: Se subdivide la imagen inicialmente en un conjunto de regiones disjuntas, dentro de las cuales, se volverá a realizar una subdivisión o bien una fusión entre ellas, de acuerdo con la validación de las condiciones preestablecidas [62]. Las estructuras de uso más común para la subdivisión es el árbol cuaternario, los pasos a seguir son:
    - Definir una prueba de homogeneidad
    - Subdividir la imagen conformando cuatro cuadrantes disjuntos
    - Calcular la media de homogeneidad para cada cuadrante
    - Fusionar dos regiones si la condición de homogeneidad es verificada para la unión de las mismas
    - Si una región no valida la condición, se realiza nuevamente el proceso de división sobre esa región específica y se repite el proceso hasta que todas las regiones validen la prueba de homogeneidad.

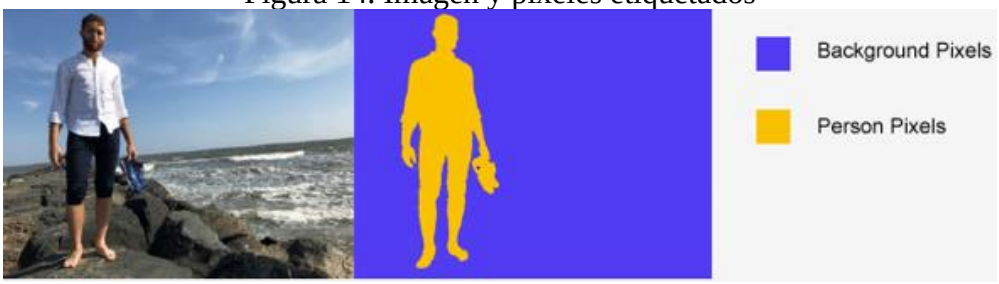
Con la evolución de los métodos y las técnicas generales de procesamiento de datos aparece una nueva rama del conocimiento que se interpola a prácticamente todos los campos de la ciencia; se hace referencia al aprendizaje de máquina (*machine learning*) [69] y aprendizaje profundo (*deep learning*) [70], desde luego el procesamiento de imágenes digitales no es ajeno a la incorporación del concepto en sus procedimientos.

## 2.5. Segmentación semántica

Como resultado de la integración del aprendizaje profundo con el procesamiento de imágenes digitales aparece un nuevo concepto conocido como segmentación semántica [71], la incorporación del aprendizaje profundo revoluciona la forma en que se realiza la segmentación de imágenes porque proporciona al procesamiento una herramienta creada basándose en imitar la forma en que piensan los seres humanos, la red neuronal artificial [72] reforma sustancialmente la estructura de los algoritmos empleados hasta ese momento para dicho fin. Para establecer con claridad el concepto, la segmentación semántica es un algoritmo de aprendizaje profundo que asocia una categoría a cada píxel dentro de una imagen, en pocas palabras realiza una clasificación píxel a píxel discriminando las formas y contornos para lo cual haya sido entrenada la red neuronal incorporada.

El ejemplo más básico es la segmentación de una imagen en tan solo dos categorías de píxeles, caso común cuando se desea destacar una sola forma del resto de la imagen, como se muestra en la Figura 14 donde la zona de mayor interés corresponde a los píxeles de la figura que conforman a la persona y se discriminan los demás en una categoría diferente, en este caso sería el fondo de la imagen.

Figura 14. Imagen y pixeles etiquetados




*Fuente:* Adaptado de *Segmentación Semántica* [Fotografía], por mathworks, 2022, (https://la.mathworks.com/solutions/image-video-processing/semantic-segmentation.html?s_tid=srchtitle_semantic_9)

No obstante, la segmentación semántica no se ve acotada a tan solo dos categorías de clasificación, sino que es posible aumentar el número de estas indefinidamente hasta alcanzar el grado de separación deseado en la imagen; para un ejemplo particular es posible aumentar las categorías de segmentación de la Figura 14 para así obtener los píxeles discriminadamente correspondientes a persona, cielo, mar y fondo [73]. Desde luego la cantidad de categorías y la exactitud con la que se extraen los píxeles asociados a cada uno dependerá de la red neuronal y su entrenamiento [74], así como también se debe tener en cuenta que un mayor número de categorías demanda un mayor esfuerzo de procesamiento digital.

La segmentación semántica proporciona una alternativa frente a los algoritmos de detección de objetos, puesto que esta técnica permite que el objeto a segmentar abarque diferentes áreas de la imagen a nivel de píxel, además de ser capaz de extraer de forma precisa formas irregulares a diferencia de la extracción de objetos, donde el elemento de interés debe contenerse al interior de una caja rectangular.

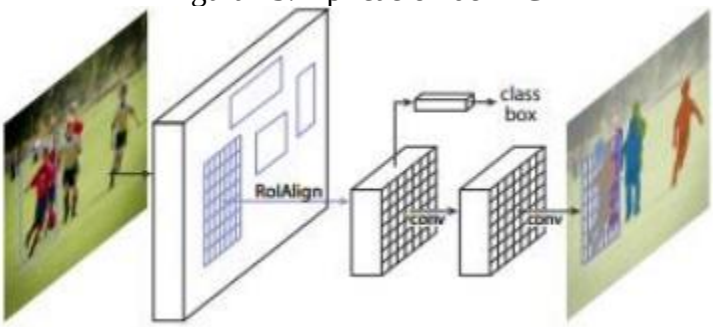
La incorporación de la segmentación semántica presenta diversas forma de interactuar con los algoritmos que se puedan emplear en el procesamiento de imágenes digitales, o con los algoritmos que emplean estos procedimientos con el fin de extraer información que será utilizada con otro propósito más complejo y estructurado; como por ejemplo, los algoritmos de visión artificial los cuales normalmente pueden ser aplicados sobre entornos estáticos o sobre entornos dinámicos, siendo el segundo caso una situación problema, que requiere de un algoritmo de segmentación semántica para lograr un establecimiento seguro y preciso debido a que los objetos dinámicos en el entorno deben ser tratados de una forma especial, teniendo en cuenta que su ocupamiento de píxeles será variable imagen tras imagen en la secuencia que constituye el vídeo base. El papel que desempeña el algoritmo de segmentación semántica es fundamental porque se hace necesario discriminar los objetos dinámicos en la escena para poder realizar un procesamiento posterior una vez detectados; sin el aislamiento preciso y oportuno de estos objetos resulta imposible el tratamiento inclusivo en el algoritmo de mayor complejidad sobre el que se implementa la segmentación semántica.

Según [75], los modelos obtenidos partiendo de redes neuronales convoluciones para segmentación semántica pueden clasificarse en dos categorías que se explican en las secciones a continuación.

## 2.5.1. Segmentación semántica basada en la región

Comparte el principio de "segmentación que usa reconocimiento" [76], en donde el primer paso es extraer regiones libres de la imagen y luego se eligen conjuntos que son función de la base de la región discriminada. Este tipo de redes convolucionales se basan en combinar dos procesos, el primero es un proceso de cómputo donde se recogen características de un gran número de regiones y el segundo es aquel donde se clasifican e identifican cada una de las segmentaciones de regiones obtenidas. Durante el reconocimiento inicial es común hacer primero el reconocimiento del fondo; un ejemplo del caso descrito anteriormente se muestra en la Figura 15, se trata de una R-CNN que ilustra este proceso según los pasos descritos anteriormente [75].

Figura 15. Aplicación de R-CNN




*Fuente: Adaptado de* Métodos de aprendizaje profundo para la segmentación semántica de personas *(p. 12), por Eduardo Coleto, 2020.*

## 2.5.2. Segmentación semántica basada en redes completamente convolucionales

Las redes convolucionales completamente conectadas FCNN, es una extensión de la metodología de las redes CNN [77]. Debido a que estas redes presentan un tamaño fijo de procesamiento de datos y que además solo presentan capaz de convolución y reducción, se hace permisible la entrada de tamaños arbitrarios complementarios de imagen, lo que da como resultado de la segmentación una máscara binaria del mismo tamaño que la entrada. Por lo tanto, uno de los principales problemas que presenta este enfoque, es que con el objetivo de reducir los tiempos de entrenamiento se opta por realizarlo con tamaños de imágenes pequeños, debido a esto se pueden obtener resultados poco precisos puesto que es necesario escalar las imágenes a tamaño real [78]. Es por esta razón que se presentan alternativas como Deeplab-CRF [79] que ofrecen alternativas durante el procedimiento del reescalado la cual consiste en hacer uso de una convolución para escalar la resolución de las características de la solución llegando a obtener la resolución deseada u original aplicando una interpolación bilineal [80] de los resultados escalados mediante la convolución.

Dentro de este tipo de segmentaciones existen diferentes mecanismos según la aplicación final y los objetivos deseados mediante la misma; de manera destacada se encuentran las técnicas que contienen un nivel bajo de supervisión. Durante el desarrollo de este tipo de técnicas se pretende segmentar una imagen a partir de anotaciones a nivel de ROI o incluso imágenes etiquetadas. Teniendo en cuenta que normalmente tienen una menor precisión, en algunos casos como el de la Figura 16 se selecciona de forma cuidadosa el conjunto de datos con el objetivo de mejorar la eficiencia de la red, alcanzando en algunos casos una precisión de hasta un 95%.

Figura 16. Segmentación semántica poco supervisada

*Fuente: Adaptado de* Métodos de aprendizaje profundo para la segmentación semántica de personas *(p. 12), por Eduardo Coleto, 2020.*

En este proyecto se hará uso del primer tipo de segmentación semántica, ya que es la metodología más adecuada para los propósitos de esta aplicación. Además es necesario destacar, se hará en una de sus formas más básicas, ya que la complejidad que puede presentar desarrollar de una manera adecuada [81] supera notablemente al alcance esperado del proyecto y no se cuenta con los recursos computacionales necesarios para su implementación.

# Capítulo 3

# Integración de un Algoritmo SLAM Visual con un Método de Segmentación Semántica Basado en el crecimiento de la Región

---

A través de este capítulo se describe en detalle los algoritmos utilizados para la ejecución del sistema SLAM visual, el cual está basado en un ejemplo proporcionado por el soporte del software Matlab; se expone su estructura y todos los métodos asociados a las diferentes etapas durante su ejecución, además por otro, también se muestra la composición del método de segmentación semántica basado en la región que se integra al sistema SLAM visual con el fin de fortalecer la aplicación de este en entornos dinámicos interiores. Como complemento al método de segmentación semántica se integran dos filtros, uno de ellos para la eliminación de puntos atípicos y un algoritmo de marcos adyacentes para la invarianza de la velocidad.

## 3.1. Estructura del algoritmo ORB-SLAM visual Implementado

Las partes que componen el ejemplo del algoritmo ORB-SLAM visual extraído de Matlab son:

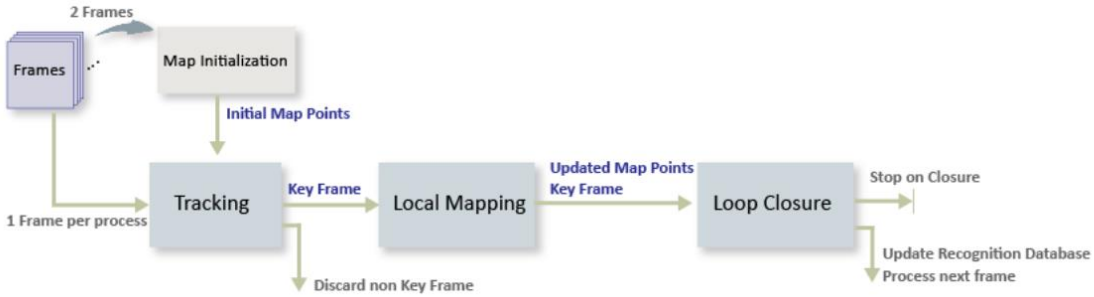
- Inicialización de mapa
- Seguimiento
- Mapa local
- Cierre de bucle

La ejecución del sistema de ORB-SLAM se plantea como se muestra en la Figura 17. Considerando que el sistema ORB-SLAM está basado en características, es pertinente mencionar que la información es obtenida únicamente de las imágenes proporcionadas por la base de datos, la información extraída se realiza mediante el método ORB para obtención de características, el cual presenta extracción y etiquetado de estas, siendo una función indispensable desde el inicio de la ejecución del sistema. Además, con la finalidad de minimizar los errores de reproyección de las posiciones relativas de la cámara, el algoritmo implementa uno o varios filtros de ajustes de movimiento; dichos filtros se encargan de tareas influyentes sobre la reconstrucción del mapa y estructuras 3D mediante el refinamiento de puntos tridimensionales y poses de cámara,

optimización del gráfico de poses de acuerdo con sus restricciones de borde definidas y correspondencia de puntos sobre imágenes bidimensionales.

Debido a la naturaleza de las escenas utilizadas para la implementación del sistema ORB-SLAM las cuales cuentan con objetos dinámicos influyentes para la reconstrucción de mapa, se implementa un filtro de estructura a partir del movimiento basado en la triangulación de marcos adyacentes reduciendo la invarianza de la velocidad en los cuadros, el cual busca disminuir la repercusión de los objetos dinámicos sobre el resultado de la aplicación de SLAM.

Figura 17. Flujograma básico de ORB-SLAM

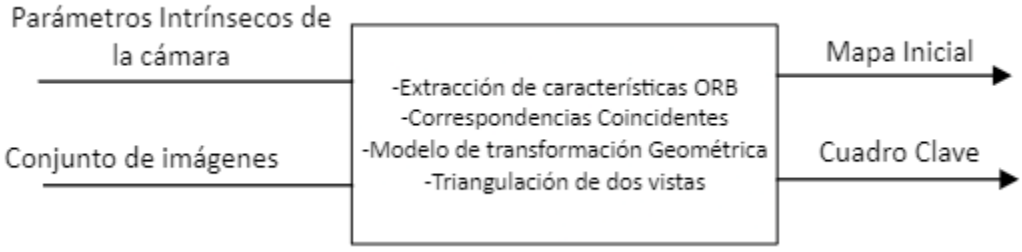



*Fuente:* Adaptado de *Monocular visual simultaneous localization and maping* [Diagrama], por mathworks, 2022, (https://la.mathworks.com/help/vision/ug/monocular-visual-simultaneous-localization-and-mapping.html)

## 3.1.1. Inicialización de mapa

En el Diagrama 1, se observa el proceso mediante el cual se realiza la inicialización del mapa, este se ejecuta una sola vez en todo el algoritmo y busca encontrar las coordenadas de partida que sitúan el punto inicial de referencia en el mapa, también se encarga de proyectar los puntos del mundo 3D a 2D. Para la ejecución de esta etapa del algoritmo, es necesario contar con la secuencia de imágenes que se desea procesar además de los parámetros intrínsecos de la cámara, que como su nombre lo menciona son inherentes a cada equipo con el que se capture las secuencias de imágenes.

Diagrama 1. Inicialización de mapa




*Fuente:* Propia

Como se mencionó anteriormente la función de extracción de características es la primera en ejecutarse, con esta se detecta y extrae los puntos característicos de los cuadros, almacenando vectores de características y sus ubicaciones correspondientes en una imagen binaria (); posteriormente haciendo uso de la función de características coincidentes se busca el siguiente cuadro clave que contenga el mayor número de semejanzas asociadas al primero, este parámetro se establece mediante el emparejamiento de los puntos característicos detectados al interior de cada cuadro.

La estimación de la pose relativa inicial de la cámara se calcula a partir de un modelo de transformación geométrica; el modelo geométrico es seleccionado entre la homografía o la estimación de matriz fundamental el cual se ejecuta a partir de los datos de los cuadros coincidentes y se selecciona uno de ellos teniendo en cuenta aquel que menor error de reproyección arroje como resultado. Este se encarga de estimar la rotación y traslación relativa, componentes que constituyen la pose relativa inicial.

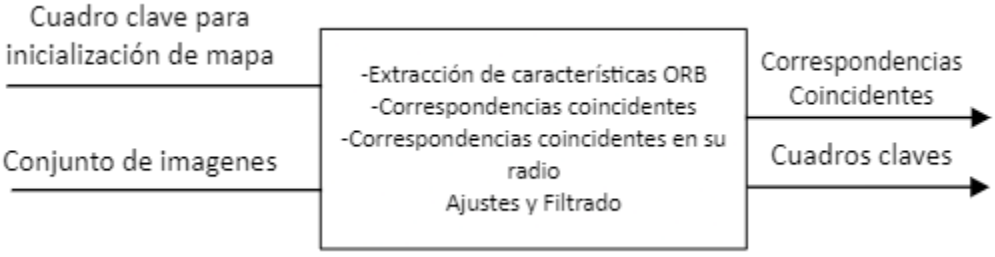
Adicionalmente se ejecutan algoritmos de ajuste y filtrado que buscando minimizar sus errores de reproyección en la toma de puntos tridimensionales y sus poses relativas.

## 3.1.2. Seguimiento

Durante la ejecución del algoritmo ORB-SLAM la etapa de seguimiento es la más iterativa; este ciclo se encarga de tomar cada cuadro de la base de datos con el fin de determinar si es apto para clasificarse como un cuadro clave o si este se descarta; los cuadros catalogados como claves serán utilizados por el algoritmo para estimar la trayectoria y reconstrucción del mapa.

En el Diagrama 2, se representa la etapa de seguimiento, así como las funciones involucradas en ella; la correcta ejecución de esta etapa repercute en gran medida sobre el resultado final del sistema, puesto que para cada nuevo cuadro se busca estimar la pose de la cámara mediante la coincidencia entre cuadros claves (la coincidencia debe existir entre el cuadro actual inmerso en la ejecución y el último cuadro clave seleccionado).

Diagrama 2. Etapa de seguimiento




Fuente: Propia

Cada cuadro que ingresa a la etapa de seguimiento se utiliza para determinar cuándo se debe insertar un nuevo cuadro clave. El flujo de trabajo al interior de la etapa de seguimiento inicia con un procedimiento idéntico al utilizado para establecer la inicialización de mapa, utilizando la función de características coincidente y ubicando nuevos cuadros claves a lo largo de toda la secuencia de imágenes para determinar la pose actual de la cámara a partir de las correspondencias coincidentes entre los cuadros claves. Adicionalmente es importante mencionar que durante toda la ejecución de esta etapa también se aplica, cuadro por cuadro, el filtro por estructura de movimiento basado en la triangulación de marcos adyacentes reduciendo la invarianza de la velocidad en los cuadros y minimizando los errores de reproyección y la repercusión de los objetos dinámicos en la escena.

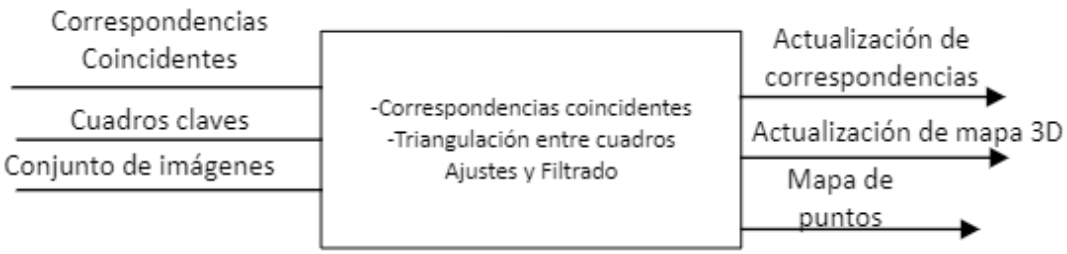
### 3.1.3. Mapeo Local

Paralelamente al seguimiento se ejecuta el mapeo local, proceso que se describe gráficamente en el Diagrama 3 y que se desarrolla para cada cuadro clave identificado previamente, buscando agregar los puntos de mapa correspondientes, observados por los cuadros claves, a los atributos para la construcción de la nube de puntos seleccionados.

Para la creación de nuevos puntos de mapa el algoritmo utiliza la función de triangulación entre dos cuadros, encontrando las correspondencias entre los puntos característicos extraídos en el cuadro actual y los que se tienen del último cuadro clave, ya que se pudo generar algún tipo de inconsistencia al comparar correspondencia de características coincidentes, el algoritmo busca coincidencias entre puntos no coincidentes de los anteriores cuadros, realizando refinamiento de pose tanto al cuadro en el que se está trabajando como a los previamente conectados y todos sus puntos de mapa observados.

Además de esto emplea un filtro para objetos dinámicos el cual se ejecuta cada vez que se agrega un nuevo cuadro clave; su función es hacer que el mapa de puntos contenga la menor cantidad de valores atípicos, debido a esta razón antes de agregar un nuevo punto de mapa este se discrimina a través del filtro, el cual elimina el punto de no ser observado al menos en 3 cuadros claves previos al actual, consiguiendo de esta manera eliminar los puntos que posiblemente representen movimientos bruscos durante la reproducción de la escena.

Diagrama 3. Etapa de mapeo local




Fuente: Propia

### 3.1.4. Cierre de bucle

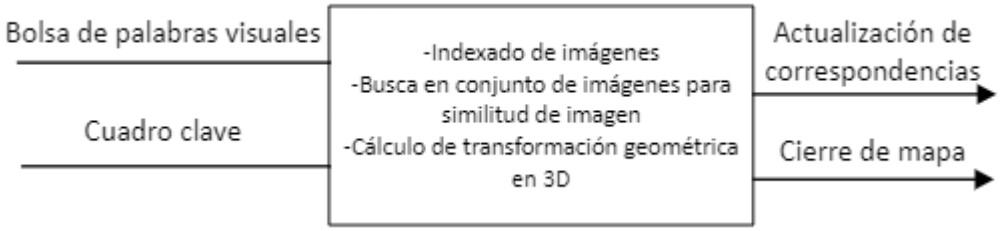
Debido a que se está trabajando sobre un algoritmo de SLAM visual basado en características, se hace necesario emplear el uso de la función de bolsa de objetos de palabras visuales cuyo objetivo consiste en identificar cierres de bucle a través de la identificación de ubicaciones visitadas previamente (reconocimiento de lugares).

Esta función se crea previo al ciclo de seguimiento utilizando métodos de indexado de imágenes con la información proporcionada por la base de datos, para posteriormente ser utilizada durante el cierre de bucle.

Como se ilustra en el Diagrama 4, la detección de un posible cierre de bucle se realiza teniendo en cuenta el cuadro clave actualmente procesado por el mapa local para determinar si este es un candidato de bucle, estos candidatos de bucle se identifican al consultar a la bolsa de objetos de palabras visuales y encontrar similitudes con el cuadro clave actual. Para saber si un cuadro clave es candidato a bucle este no debe estar conectado al último cuadro clave y tres de sus cuadros claves vecinos ser candidatos a bucle.

Al encontrar un candidato de bucle válido, se calcula la transformación geométrica de pares de puntos coincidentes entre el cuadro candidato a cierre de bucle y el cuadro clave actual para obtener la pose relativa, luego se agregan las conexiones de bucle a la pose relativa y se actualiza el mapa de puntos y sus puntos claves.

Diagrama 4. Etapa de cierre de bucle

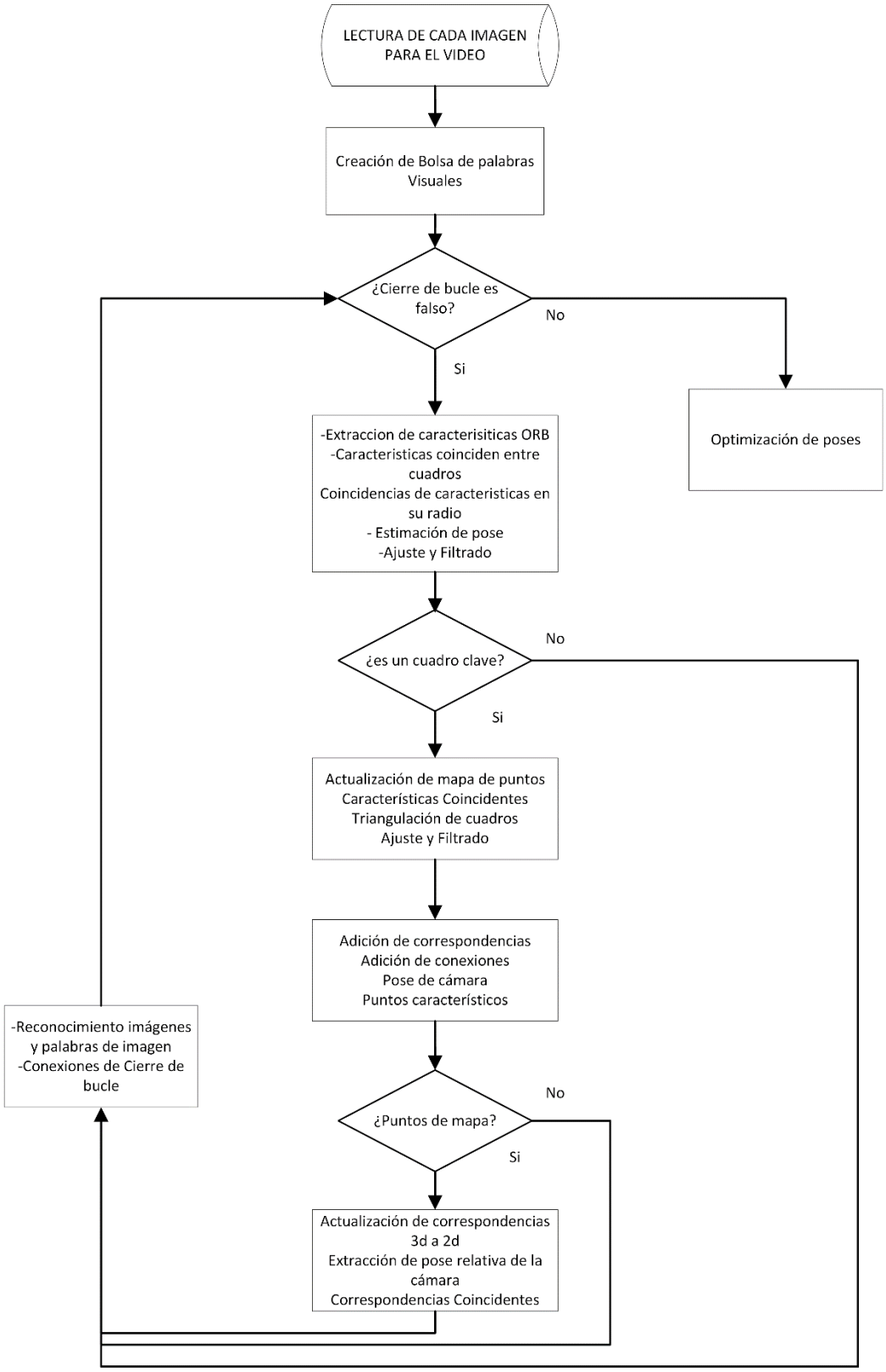



Fuente: Propia

Una vez detectado el cierre de bucle se procede a cerrar el mapa y graficar la trayectoria estimada, para esto se realiza una optimización en el gráfico de poses de similitudes; está optimización elimina las conexiones con pocos puntos coincidentes en el gráfico de covisibilidad, corrigiendo las desviaciones de la pose de la cámara, posteriormente actualiza las ubicaciones tridimensionales de los puntos del mapa con las poses optimizadas.

Para concluir de manera general el procesamiento completo que realiza el algoritmo sobre cada secuencia de imágenes, se presenta el Diagrama 5 donde se ilustra cada una de las etapas concatenadas lógicamente al interior de las cuales se describen las actividades puntuales desarrolladas de manera secuencial.

Diagrama 5. Estructura descriptiva del proceso interno para el algoritmo ORB-SLAM

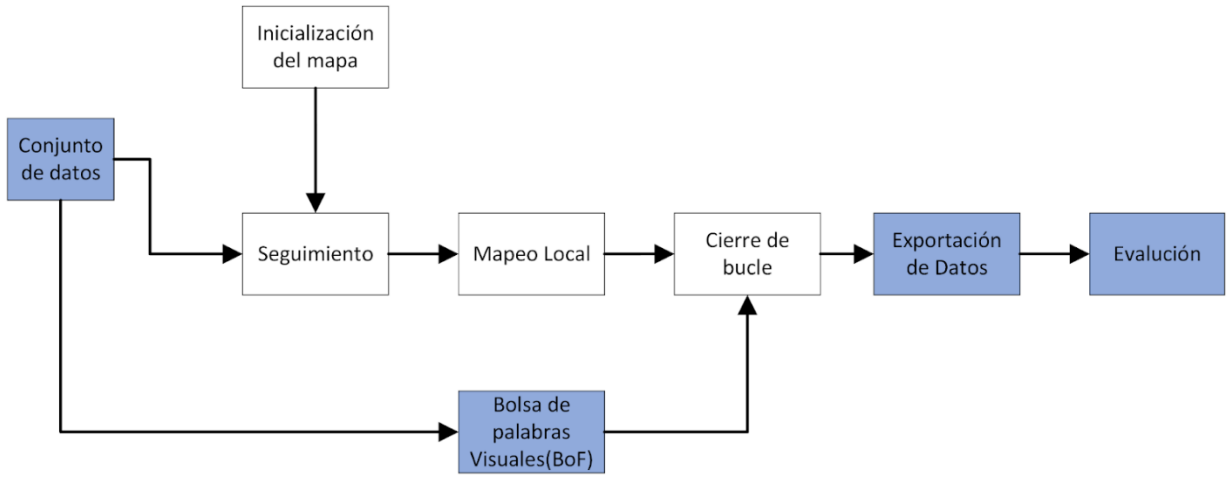



Fuente: Propia

### 3.1.5 Adaptación del Algoritmo ORB-SLAM para los objetivos propuestos

El algoritmo base prediseñado tomado de Matlab, se modifica y se adapta con la finalidad de acondicionarlo a los objetivos propuestos en este trabajo. La primera de las modificaciones se realiza en la etapa de ingreso del conjunto de datos al motor del algoritmo pasando de un ingreso mediante descarga vía internet y almacenamiento en memoria volátil durante la ejecución del algoritmo, a un cargue de datos local mediante la obtención de los archivos de la base de datos TUM RGB-D y almacenamiento permanente; el segundo cambio significativo radica en la implementación de una bolsa de palabras visuales construida con base en el conjunto de datos correspondiente a cada una de las secuencias de imágenes procesadas con la finalidad de optimizar el proceso de cierre de bucle; un tercer cambio radica en la etapa de obtención final de datos para evaluación de los resultados, puesto que mediante Matlab la evaluación presentó dificultades para escalar las trayectorias adecuadamente y obtener una comparación objetiva, los datos de salida correspondiente a la estimación y optimización de la trayectoria reconstruida por el algoritmo, dichos datos debieron ser exportados y adecuados mediante un archivo de extensión de texto para su posterior evaluación en el lenguaje de programación Python. En el Diagrama 6 se ilustra los cambios realizados sobre la base original del algoritmo ORB SLAM tomado de Matlab, donde los cuadros en color blanco representan las etapas del algoritmo que no fueron modificadas y los cuadros en color azul representan los cambios de adaptación que se implementaron, descritos anteriormente.

*Diagrama 6 Adaptación del algoritmo OBR-SLAM implementado*




Fuente: propia

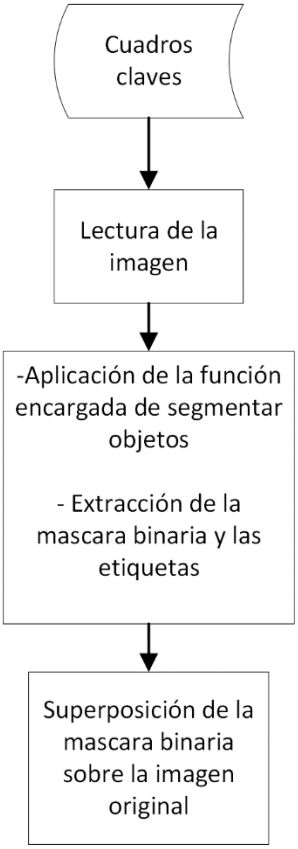
## 3.2. Obtención del modelo para la segmentación semántica

Como se ha determinado anteriormente, el objetivo es la evaluación de un sistema SLAM Visual con segmentación semántica en entorno dinámicos, por lo tanto, es de suma importancia ahondar en el proceso de obtención del modelo mediante el cual se realiza la segmentación semántica, así como también el conjunto de datos sobre los cuales se desea aplicar dicho modelo.

Para este caso, se utiliza una red neuronal convolucional basada en regiones de máscara multiclase (R-CNN, *regions whit CNN features*) la cual segmenta instancias individuales de personas y automóviles, esta red constituye una parte de un conjunto de redes pre-entrenadas y disponibles para uso académico en la página oficial de soporte del software Matlab.

El paso siguiente es realizar la lectura de la imagen que se pretende segmentar, para este caso se utiliza una imagen de prueba ilustrada en la Figura 18; el procesamiento digital que conlleva la imagen una vez realizada su importación al software se describe secuencialmente en el Diagrama 7.

Diagrama 7. Proceso que realiza la segmentación semántica




Fuente*:* propia

Inicialmente se realiza la lectura digital de la imagen extraída del conjunto de cuadros claves para realizar la extracción de características producto de la segmentación de objetos sobre la imagen, es decir que se obtiene la máscara binaria que describe todos los objetos dinámicos segmentados en la escena así como las etiquetas asignadas para cada uno de los pixeles, la puntuación de cada etiqueta y si se desea visualizar también las ubicaciones de cada objeto segmentado delimitado como cajas de texto. Es importante mencionar que la función encargada de la extracción de dichas características hace parte del conjunto de funciones desarrolladas por Matlab y asociadas a la caja de herramientas denominada ‘computer vision toolbox’.

Figura 18. Imagen de prueba seleccionada para segmentación semántica

*Fuente:* Adaptado de *TUM School of Computation, Information and Technology*[Fotografía], por Computer Vision Group,2022.

La función principal encargada de la segmentación de objetos recibe como parámetros esenciales la red previamente creada sobre el conjunto de datos almacenados en la variable local, la imagen (o directorio si se trata de una secuencia) a procesar y el umbral de segmentación. Como datos adicionales al respecto de esta función es relevante mencionar que inicialmente, antes de realizar el procesamiento para la extracción de características, estructura dos pasos de vital importancia para el éxito del propósito del algoritmo que se constituye en la precisión de la segmentación, los cuales son:

- Centrar en cero las imágenes utilizando la media del conjunto de datos COCO: este procedimiento es de vital importancia para evitar afectaciones durante el proceso de segmentación, debido a que normalmente las etiquetas de pixeles en una imagen no se encuentran equilibradas en cuanto al volumen de pixeles que componen cada una de

ellas (es por esto que se deben equilibrar mediante una ponderación teniendo como referencia un valor promedio, del conjunto de datos COCO para este caso) de no realizar este ponderado la segmentación podría verse afectada al encontrarse sesgada hacia las clases mas dominantes, es decir con mayor número de pixeles etiquetados al interior de ellas.

- Cambiar el tamaño de la imagen al tamaño de entrada de la red, manteniendo la relación de aspecto: debido a que la red realiza una seria de escalados durante el procesamiento de cada imagen, realizar la validación sobre el tamaño en pixeles de cada una de ellas es importante para tener un buen punto de partida libre de fallos por disparidad en la correlación de pixeles entra una etapa de escalado y otra.

El resultado de la aplicación del algoritmo de segmentación semántica sobre la imagen de prueba se puede visualizar gráficamente en la Figura 19, lo que se aprecia es el resultado de la superposición entre la máscara binaria, arrojada por la función principal del algoritmo de segmentación semántica, y la imagen original que se utilizó para su procesamiento. La precisión asociada entre la región sombreada en azul y el contorno de las dos personas (las cuales fueron identificadas por el algoritmo como objetos dinámicos) varía en función de la red neuronal utilizada para la segmentación, el volumen de datos que esta reciba para el entrenamiento en reconocimiento de formas, contornos y patrones, así como también de los recursos computacionales con los que se cuenta para su correcta implementación.

Figura 19. Imagen segmentada con superposición de máscara binaria

*Fuente:* Adaptado de *TUM School of Computation, Information and Technology*[Fotografía], por Computer Vision Group,2022.

Por último es importante recalcar la contribución de la segmentación semántica a la consecución del objetivo que se busca mediante la aplicación del algoritmo SLAM; para ahondar en este proceso se debe tener en cuenta que la reconstrucción de la trayectoria y el mapeo local que realiza el algoritmo SLAM se basa principalmente en los puntos característicos obtenidos de los cuadros claves, es justamente sobre dichos puntos donde la segmentación toma protagonismo procesando pixel a pixel cada una de los cuadros claves y filtrando los puntos característicos propios de cada uno de ellos, para de esta manera cotejar si un punto se encuentra al interior de la región segmentada y eliminarlo para que no presente disturbios durante la reconstrucción de la trayectoria y el mapeo local, o por el contrario si dicho punto no se encuentra al interior de la región segmentada y es válido para ser utilizado durante el mapeo.

## 3.3. Integración de las partes y consolidación del algoritmo final

Partiendo de la premisa de que el sistema SLAM visual en entornos dinámicos no presenta el mejor desempeño respecto a la reconstrucción de la trayectoria y el mapa local, se integra un método de segmentación semántica basado en el crecimiento de la región, el cual se incorpora a la estructura general de SLAM como se muestra en el Diagrama 10.

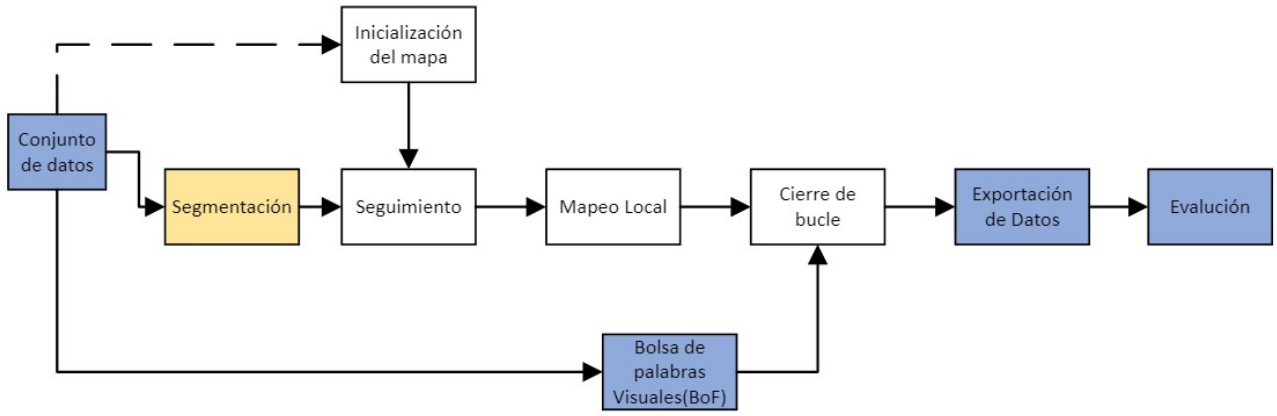
Anteriormente se expuso el resultado de aplicar el algoritmo de segmentación semántica basado en regiones sobre una imagen de prueba, lo cual arrojó un resultado favorable en pro de la consecución del objetivo final del presente trabajo, por esta razón, es prioridad llevar a cabo la integración de este algoritmo sobre el sistema de localización y mapeo simultáneos con la finalidad de mejorar el rendimiento de SLAM sobre entornos dinámicos, además de poder establecer una comparación cuantitativa acerca de las mejoras y diferencias en el rendimiento de este sistema una vez implementado el algoritmo de segmentación.

### 3.3.1. Optimización de la ejecución durante la integración de los algoritmos

En busca de que la integración de la segmentación semántica repercuta en la menor cantidad posible sobre la extensión de los tiempos de ejecución por secuencia de imágenes y el consumo de los recursos computacionales, se hace necesario establecer el punto optimo de integración del método de segmentación semántica sobre el algoritmo general de ORB-SLAM; para ello se realizaron dos evaluaciones que presentaron marcadas diferencias en cuanto a los parámetros de rendimiento de ejecución asociados a cada una de ellas y se seleccionó para realizar las respectivas pruebas y experimentos, aquella estructura de integración que arrojo menor tiempo de ejecución sin deteriorar o disminuir el impacto del método de segmentación semántica sobre la reconstrucción de la trayectoria.

En el Diagrama 8 se ilustra el primer intento de integración de los dos algoritmos, para este caso la segmentación semántica es aplicada en primera instancia a todo el conjunto de imágenes que conforman la secuencia previamente suministrada al algoritmo, para que posterior a ello se inicie la etapa de seguimiento con la información correspondiente a la máscara binaria que suministra la segmentación semántica para cada una de las imágenes. Este intento de integración presento un gran problema sobre los tiempos de ejecución del algoritmo general, puesto que el impacto de aplicar la segmentación semántica a toda la secuencia de imágenes (mil imágenes en promedio por cada secuencia) tuvo una gran repercusión sobre el consumo de los recursos computacionales, logrando un tiempo de ejecución aproximado de 600 minutos, lo cual es un tiempo demasiado largo para una secuencia de tan solo 30 segundos de duración.

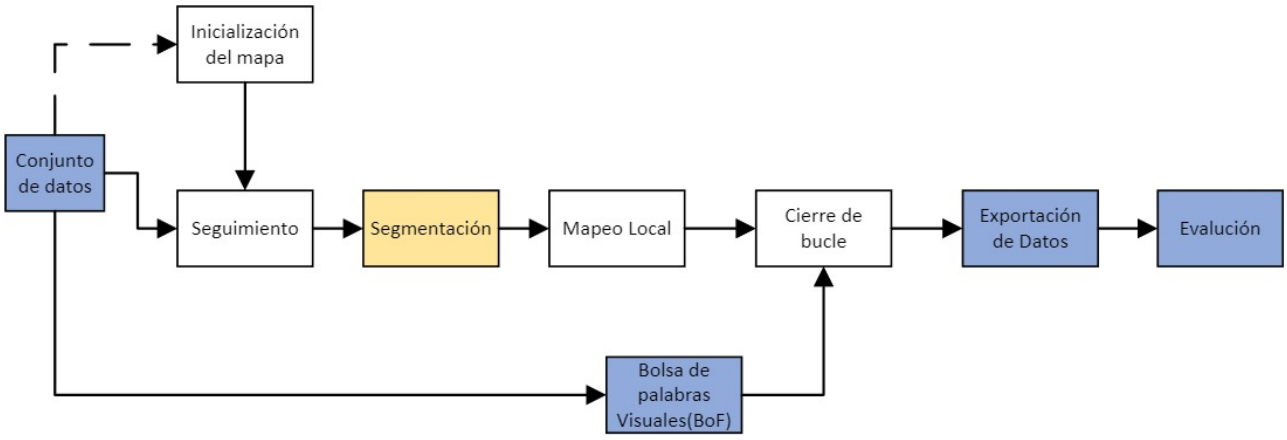
Diagrama 8 integración del método de segmentación semántica




Fuente. Propia

Con la finalidad de optimizar la integración del proceso de segmentación semántica al sistema, este se incorpora sobre el algoritmo ORB-SLAM secuencialmente como un subproceso de la etapa de seguimiento como se muestra en el Diagrama 9, una vez seleccionados los cuadros claves, acción que disminuye el costo computacional y las iteraciones innecesarias de segmentación sobre cuadros que posteriormente no van a ser tenidos en cuenta para la localización y reconstrucción de mapa. Durante la etapa de seguimiento se descartan aproximadamente el 90% de las imágenes correspondientes a cada secuencia y se conserva tan solo el 10% aproximadamente para la reconstrucción de la trayectoria estimada, esta cantidad de imágenes corresponde a aquellas que es sistema catalogo como cuadros claves.

Diagrama 9 Optimización de la integración del método de segmentación semántica



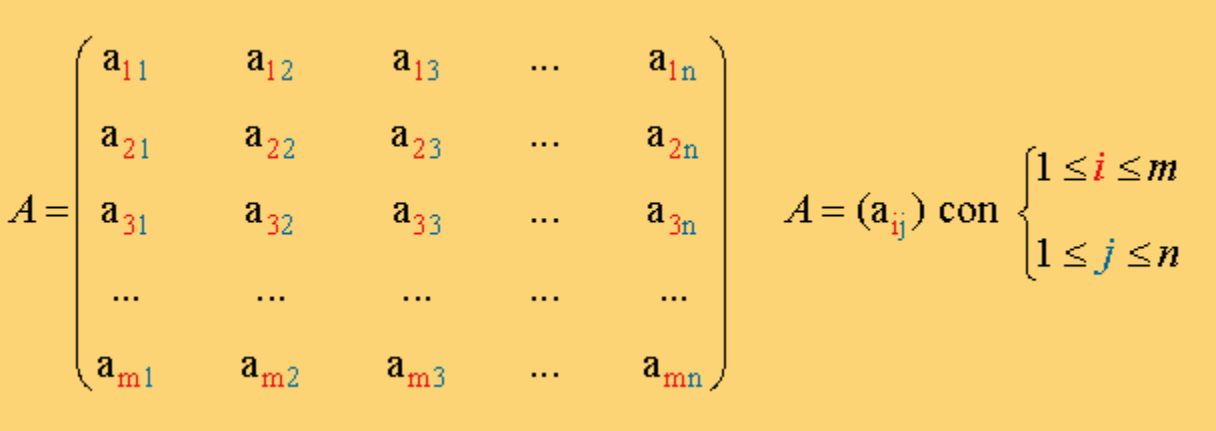

Fuente: Propia

## 3.3.2. Eliminación de puntos característicos segmentados

El desarrollo de este proceso parte del hecho de que ya se cuenta con una máscara binaria asociada a un cuadro en particular, máscara binaria proporcionada por la función principal de segmentación de objetos producto de la segmentación de cada cuadro clave, esta máscara se utiliza con la finalidad de evitar que determinados puntos característicos al interior de los objetos dinámicos sean tenidos en cuenta para la localización y reconstrucción de mapa.

Dado que la máscara binaria corresponde a una matriz tridimensional que obedece a la profundidad donde se encuentren los objetos segmentados, es necesario realizar una conversión donde se elimine la dimensión correspondiente a dicha profundidad sin incurrir en pérdidas de información, obteniendo como resultado final una máscara binaria que comparta las dimensiones de las imágenes procesadas desde el banco de datos. Para esto se debe superponer cada una de las capas de la matriz correspondiente a la máscara dando prioridad a los píxeles que contienen la información de los objetos segmentados, los cuales se encuentran representados por un valor unitario.

Figura 20. Supresión de dimensión en matriz



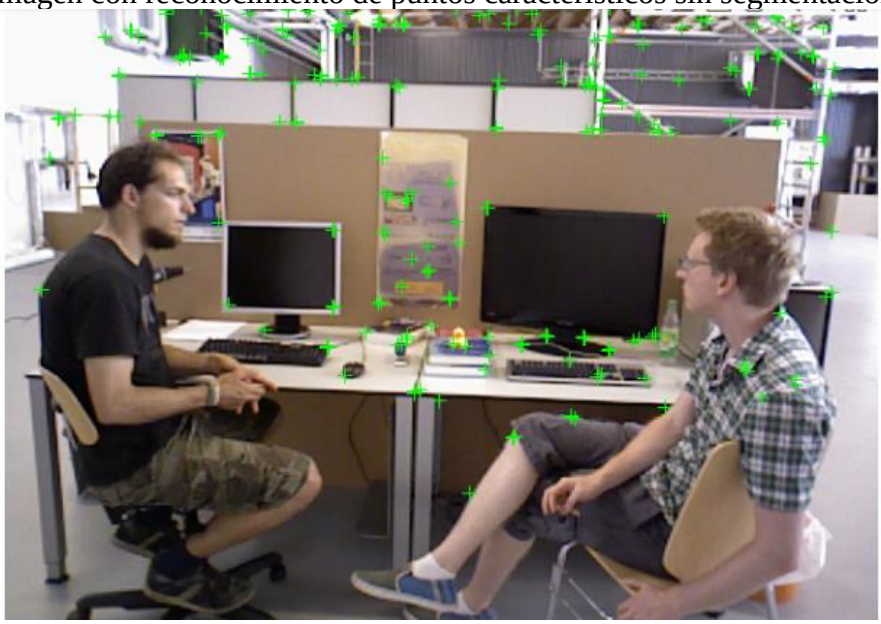

Fuente: propia

Una vez definida la máscara binaria en una matriz bidimensional con las mismas dimensiones de las imágenes procesadas, se procede a la eliminación de puntos característicos contenidos al interior de los objetos dinámicos, puesto que estos no serán tenidos en cuenta para las etapas posteriores del sistema. Inicialmente se obtienen las coordenadas de cada uno de los puntos característicos extraídos de la imagen en cuestión para posteriormente cotejar los pares de coordenadas sobre la máscara binaria correspondiente a la imagen, en caso de que el punto característico evaluado se corresponda con un valor unitario en la máscara, se asume que se encuentra en la región segmentada y la variable contenedora de los puntos característicos se actualiza, eliminando todos los puntos proyectados sobre un objeto dinámico.

Terminada la integración de la segmentación semántica al sistema, se realiza una validación práctica del efecto propagado sobre la toma y eliminación de puntos característicos. Para la validación se realizan dos pruebas sobre el mismo conjunto de datos, el resultado de la primera de ellas se muestra en la Figura 21, donde se evidencia todos los puntos característicos extraídos por el sistema, pero este no cuenta aún con la segmentación incorporada, por lo tanto, se visualiza el conjunto de puntos característicos extraídos sin ningún tipo de filtrado. Para efectos visuales es oportuno tener en cuenta que la segmentación se realiza solo sobre objetos dinámicos, los cuales en las escenas a procesar se ven representados en su totalidad por personas que difieren únicamente en su disposición física al interior de la secuencia de imágenes.

El resultado obtenido de la segunda prueba desarrollada sobre el mismo conjunto de datos se ilustra en la Figura 22, donde es posible evidenciar en color celeste la superposición de la máscara binaria - obtenida del algoritmo de segmentación - sobre los objetos dinámicos, además de ello notar que los puntos característicos visualizados en la imagen (anterior) ya no se proyectan en su mayoría cuando se encuentran sobre un objeto segmentado.

Figura 21. Imagen con reconocimiento de puntos característicos sin segmentación semántica



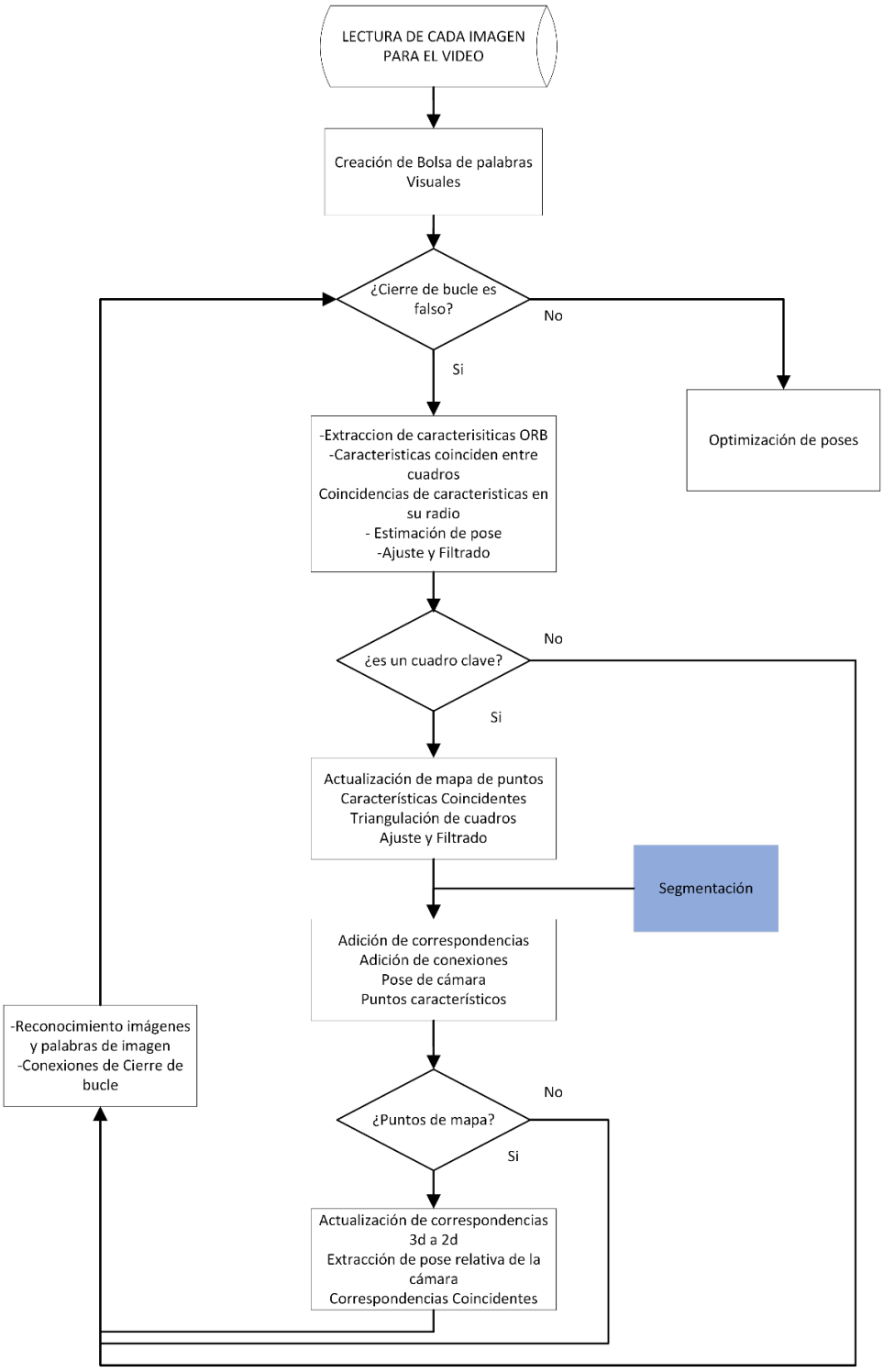
*Fuente:* Adaptado de *TUM School of Computation, Information and Technology*[Fotografía], por Computer Vision Group,2022.

Figura 22. Resultado de la prueba de segmentación para la eliminación de puntos característicos

*Fuente:* Adaptado de *TUM School of Computation, Information and Technology*[Fotografía], por Computer Vision Group,2022.

Es importante destacar que la eliminación de los puntos característicos sobre los objetos segmentados se hace más compleja para el sistema cuando dichos puntos se encuentran cerca al borde del objeto, por esta razón es posible que algunos de estos no sean eliminados en su totalidad bajo estas condiciones específicas de localización en la imagen.

Diagrama 10. Integración de la segmentación semántica a la estructura de SLAM visual




Fuente: Propia

# Capítulo 4

# Pruebas y Resultados

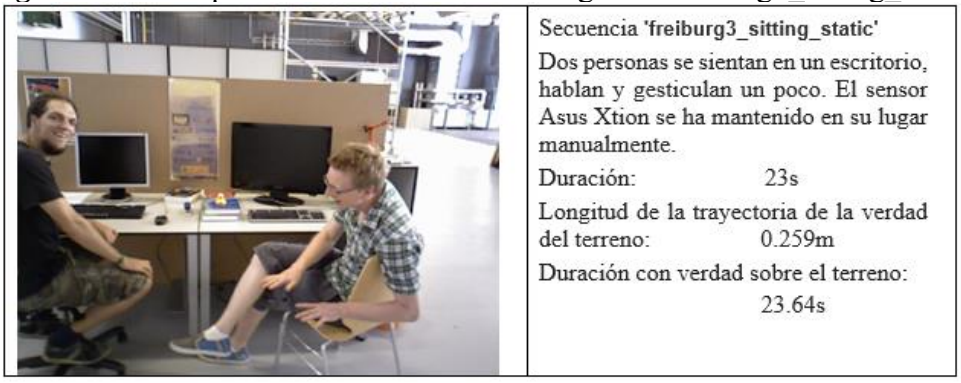
En este capítulo se exponen los resultados experimentales producto del procesamiento de seis secuencias de imágenes diferentes bajo distintas condiciones algorítmicas; en primera instancia se realizaron pruebas de rendimiento con un algoritmo SLAM visual y posteriormente se comparó el desempeño de este frente al mismo algoritmo SLAM visual adicionando la integración de un método de segmentación semántica.

## 4.1. Base de datos: TUM

La base de datos TUM es una base de datos proporcionada por la universidad técnica de Múnich la cual cuenta con un grupo de investigación enfocado en la visión por computadora, esta proporciona distintos conjuntos de datos para la evaluación de sistemas de visión artificial, además cuenta con un apartado específico para la evaluación de sistemas SLAM visuales con la intervención de objetos dinámicos en entornos interiores, por esta razón dicho apartado fue el seleccionado para las pruebas y evaluaciones del algoritmo propuesto. El conjunto de datos contiene las imágenes a color y de profundidad de distintas secuencias de imágenes tomadas mediante una cámara Xtion monocular, los datos de la base de datos fueron registrados a 30Hz de velocidad y una resolución de 640 por 480, también la base de datos proporciona en archivos de texto la trayectoria real tomada por la cámara y los criterios óptimos para su evaluación

Las seis secuencias que se presentan y describen a continuación - ver Figura 23 a la Figura 28 - están destinadas para la evaluación de algoritmos de odometría visual y SLAM visual para objetos dinámicos que se mueven rápidamente en gran parte de la escena visible.

Figura 23. Descripción de la secuencia de imágenes ‘freiburg3_sitting_static’

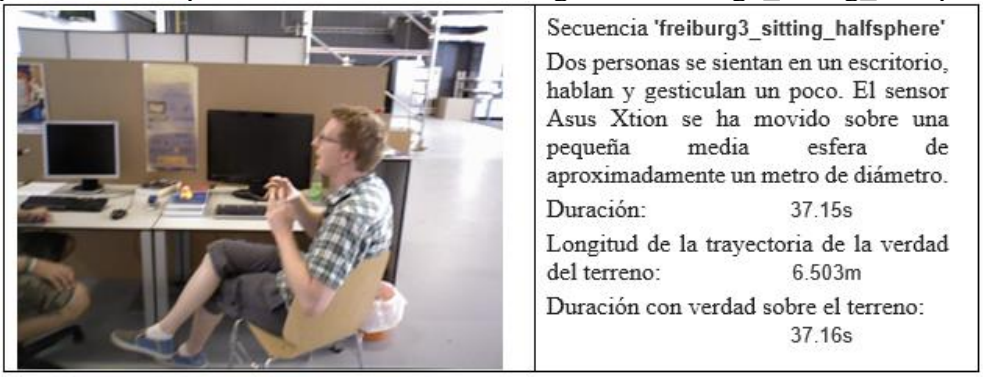


*Fuente:* Adaptado de *TUM School of Computation, Information and Technology*[Conjunto de datos], por Computer Vision Group,2022.

Figura 24. Descripción de la secuencia de imágenes ‘freiburg3_sitting_xyz’


*Fuente:* Adaptado de *TUM School of Computation, Information and Technology*[Conjunto de datos], por Computer Vision Group,2022.

Figura 25. Descripción de la secuencia de imágenes ‘freiburg3_sitting_halfsphere’

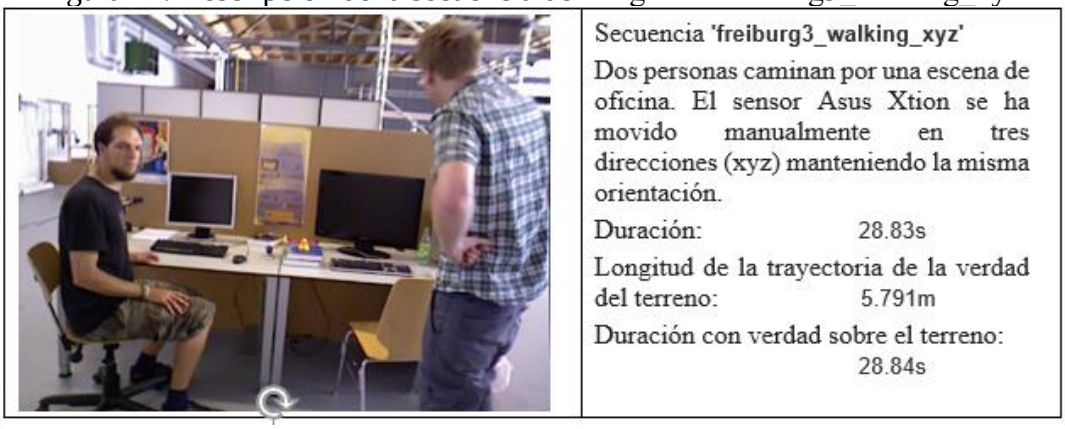



*Fuente:* Adaptado de *TUM School of Computation, Information and Technology*[Conjunto de datos], por Computer Vision Group,2022.

Figura 26. Descripción de la secuencia de imágenes ‘freiburg3_walking_static’

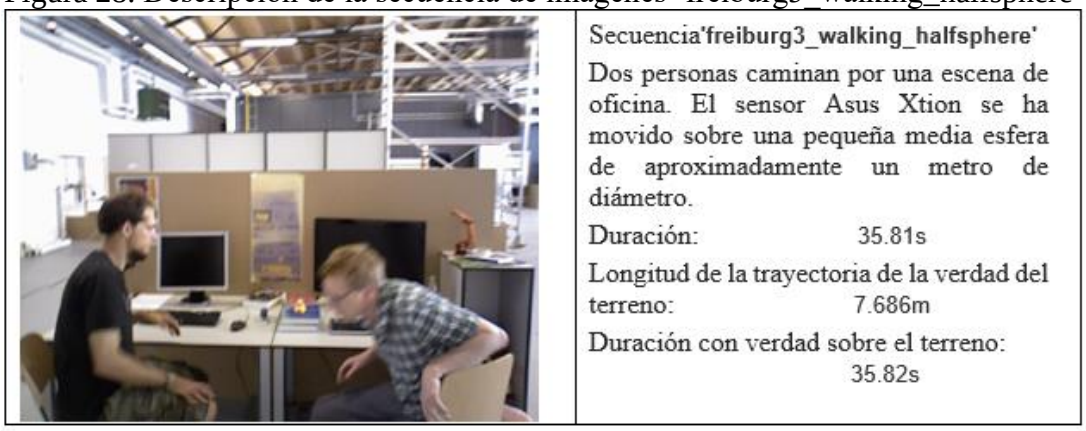
*Fuente:* Adaptado de *TUM School of Computation, Information and Technology*[Conjunto de datos], por Computer Vision Group,2022.

Figura 27. Descripción de la secuencia de imágenes ‘freiburg3_walking_xyz’




*Fuente:* Adaptado de *TUM School of Computation, Information and Technology*[Conjunto de datos], por Computer Vision Group,2022.

Figura 28. Descripción de la secuencia de imágenes ‘freiburg3_walking_halfsphere’




*Fuente:* Adaptado de *TUM School of Computation, Information and Technology*[Conjunto de datos], por Computer Vision Group,2022.

Para la evaluación es importante explicar el formato en que se proporcionan los archivos por la base de datos, siendo la trayectoria de la verdad (Ground truth) un archivo de texto que contiene la traslación y orientación de la cámara en un marco de coordenadas fijas, cada archivo de trayectoria de verdad viene dado por el siguiente formato:

- Cada fila en el archivo de texto es una pose tomada por la cámara.
- El formato en cada fila es **'timestamp tx ty tz qx qy qz qw'.**
- **'timestamp'** se refiere a la estampa de tiempo en la que está situada la fila, se utiliza para emparejar con las trayectorias estimadas por el algoritmo.
- **'tx ty tz'** Se refiere a la posición del centro óptico de la cámara con respecto al origen definido por el sistema de coordenadas.
- **'qx qy qz qw' Se** refiere al cuaternión unitario de orientación con respecto al eje óptico de la cámara con respecto al origen definido por el sistema de coordenadas.

## 4.2. Índices de desempeño

En esta sección se describen los índices de desempeño bajo los cuales se estandarizan los resultados obtenidos, dando lugar a una evaluación objetiva que permita establecer las diferencias relevantes y de interés para el análisis final del proyecto y el establecimiento de conclusiones asertivas, para esto se evalúa mediante diferencias métricas de error entre las trayectorias.

### 4.2.1. Error de trayectoria absoluta (ATE, absolute trajectory error)

El error de trayectoria absoluta es muy adecuado para medir el rendimiento de los algoritmos SLAM visuales, debido a que mide directamente la diferencia entre las poses de la trayectoria de la verdad y la estimada, para esto como se dijo anteriormente se asocian las poses mediante su estampa de tiempo alineando sus trayectorias, para finalmente calcular la diferencia entre cada par de poses y generar la media/mediana y desviación estándar de estas diferencias.

### 4.2.2. Parametrización de la evaluación del sistema

Dado que la evaluación en Matlab no es capaz de proporcionar la escala comparativa a la trayectoria de la verdad del terreno como se muestra en la Figura 29, se toman los datos de trayectoria estimada obtenidos de la ejecución en Matlab y se almacenan en archivos de texto, para posteriormente evaluarlos mediante un algoritmo en lenguaje Python creado específicamente para la evaluación de sistemas SLAM; este algoritmo toma las estampas de tiempo y compara las poses de la cámara punto a punto midiendo su error absoluto de trayectoria y proporcionando la mejor escala en la evaluación, graficando ambas trayectorias y sus errores como se muestra en la Figura 30.

Figura 29. Reconstrucción de la escala y evaluación en Matlab

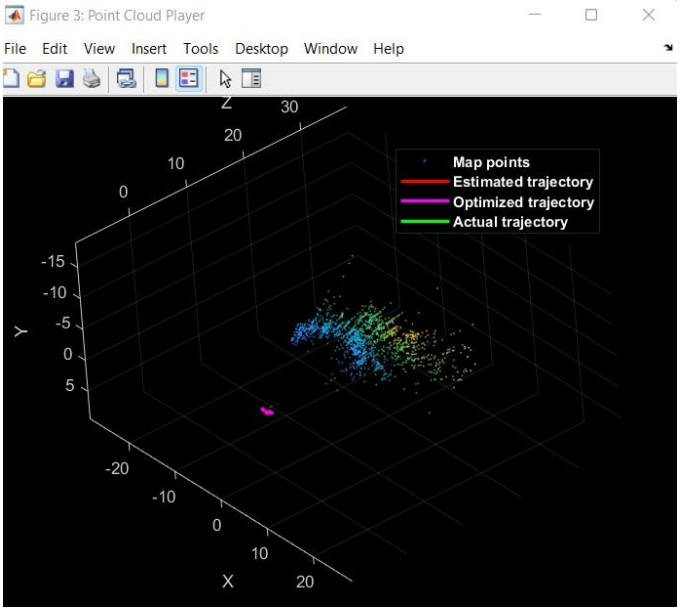
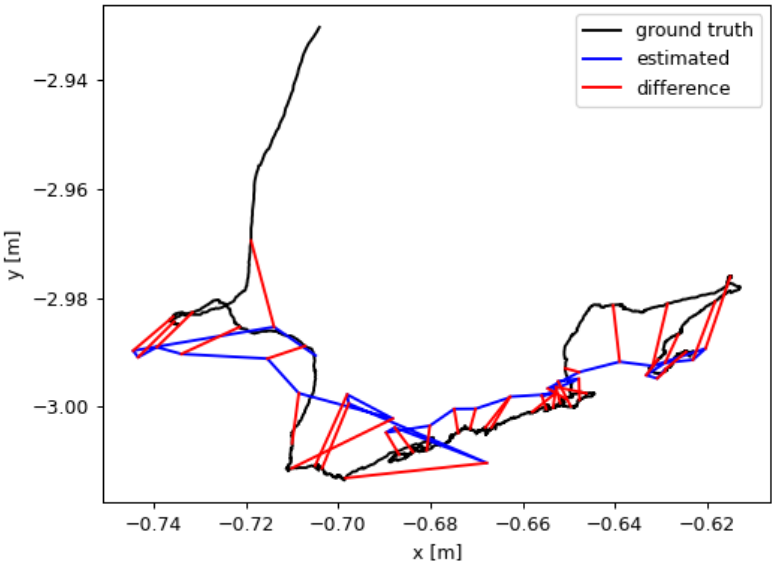



Figura 30. Reconstrucción de la trayectoria y evaluación en PYTHON




# 4.3. Resultados

En esta sección se exponen los resultados obtenidos tras la comparación de las diferentes secuencias de imágenes procesadas; la metodología implementada para la obtención de dichos resultados consiste inicialmente en realizar el procesamiento previo a la implementación del algoritmo de segmentación semántica con la finalidad de parametrizar el rendimiento del algoritmo SLAM puro sobre el seguimiento (total o parcial) de la trayectoria y su respectiva reconstrucción

de mapa. Posteriormente se procesan todas las secuencias de imágenes nuevamente con el objetivo de parametrizar los resultados tras la implementación del algoritmo de segmentación semántica y finalmente plantear una comparación entre los pares de resultados obtenidos por cada una de las secuencias de imágenes a través de un método teórico y/o grafico que permita evidenciar fácilmente las variaciones en el rendimiento de cada prueba realizada.

Debido a que el proceso de evaluación de cada secuencia es similar entre sí y resulta muy repetitivo explicar a fondo cada una de las pruebas realizadas, se profundiza de manera detallada en la evaluación de la secuencia ‘freiburg3_sitting_xyz' y los demás resultados obtenidos solo se expondrán, obviando el proceso mediante el cual se obtienen.

### 4.3.1. Evaluación mediante algoritmo SLAM sin segmentación semántica

Inicialmente el algoritmo toma la secuencia de imágenes consolidando un directorio interno con ellas, una vez son puestas a disposición del procesamiento el primer paso dentro del sistema es realizar la inicialización del mapa, este proceso se lleva a cabo tomando el primer cuadro que se encuentra en el directorio de imágenes y cotejando las coincidencias (detectadas y extraídas mediante ORB) con los demás hasta encontrar el que comparta un grado de similitud mayor o igual a ochenta puntos característicos, valor en el cual se fija el umbral. Para este caso el par de cuadros con los que se realiza la inicialización de mapa son el 1 y el 48, los cuales se muestran en la Figura 31 proyectando líneas de puntos coincidentes.

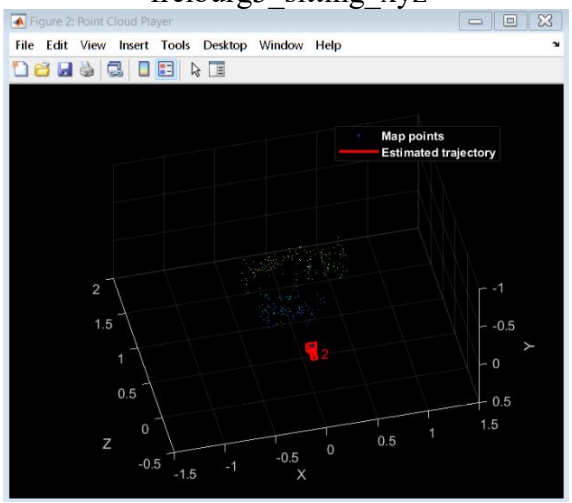
El siguiente paso es seleccionar el modelo de transformación geométrica mediante el cual establecer la inicialización del mapa, bien sea mediante homografía o haciendo uso de la matriz fundamental. El modelo seleccionado es aquel que da como resultado un error de reproyección menor, este se utiliza para estimar la rotación relativa y la traslación entre los dos cuadros mediante el método de triangulación; como resultado de este proceso inicial el algoritmo logra localizar la posición de origen de la cámara en el mundo tridimensional como se muestra en la Figura 32. Es pertinente mencionar que debido a que las imágenes son capturadas por una cámara monocular y no proporciona la información correspondiente a la profundidad de los objetos, la traducción relativa solo se puede recuperar hasta un factor de escala específico.

Figura 31. Correspondencia de puntos característicos

*Nota.* Adaptado de *TUM School of Computation, Information and Technology*[Fotografía], por Computer Vision Group,2022.

Los dos marcos utilizados para la inicialización de mapa, así como los puntos característicos obtenidos de ellos deben almacenarse para la posterior reconstrucción de la trayectoria. El lugar de almacenamiento seleccionado se utiliza nuevamente tras la selección de los cuadros claves; esta información es necesaria para obtener las posiciones de la cámara tanto relativa como absoluta, información base para la reconstrucción completa de la trayectoria realizada por la cámara.

Figura 32. Nube de puntos y localización del origen de la cámara para la secuencia de imágenes ‘freiburg3_sitting_xyz'




*Nota.* Adaptado de *Monocular Visual Simultaneous Localization and Mapping*[Imagen], por mathworks, 2022, (https://la.mathworks.com/help/vision/ug/monocular-visual-simultaneous-localization-and-mapping.html)

Posterior a la obtención y refinamiento de la posición inicial de la cámara se inicia con la etapa de seguimiento, mediante la cual se estima la reconstrucción de la trayectoria total que realiza la cámara haciendo uso de un bucle iterativo de obtención de características a través de la selección de cuadros claves parametrizados, estas iteraciones solo finalizan con la detección de un cierre de mapa. El paso inicial es determinar los cuadros claves, para ello cada uno recibe el siguiente procesamiento:

- Cada cuadro nuevo obtiene la extracción de características ORB las cuales se comparan con el último cuadro clave, el cual posee puntos de mapa tridimensional correspondientes conocidos.
- Se realiza el cálculo de la pose de la cámara
- Con la pose de la cámara obtenida, se proyectan los puntos del mapa observados en el último cuadro calve sobre el cuadro actual y se buscan las correspondencias de características.
- A través de una correspondencia de puntos tridimensionales a bidimensional en el cuadro actual se refina la pose de la cámara realizando un ajuste de solo movimiento.
- Con el fin de obtener más correspondencias de características se proyectan los puntos del mapa local en el marco actual y nuevamente se refina la pose de la cámara.
- El último paso es determinar si el cuadro actual es un cuadro clave; en caso de que lo sea se continua con el proceso de asignación local, en caso contrario se inicia el seguimiento para el siguiente cuadro.

Durante esta etapa puede ocurrir que el seguimiento no halle un cuadro clave dentro del rango máximo de tolerancia establecido para ello, por lo tanto, se pierda la continuidad del proceso y no se logre la reconstrucción de la trayectoria. Esto puede suceder porque el paquete de imágenes que se está intentado procesar no sea una secuencia o bien presente cambios demasiado bruscos a través de cada cuadro que dificulten la obtención de características coincidentes entre ellos. Una forma de garantizar la obtención de la trayectoria para cualquier paquete de imágenes, aunque presenten sendas discontinuidades, es establecer un valor máximo de tolerancia que determine el cuadro actual como nuevo cuadro clave independientemente de las correspondencias de características o eliminar la información de mayor disparidad entre cuadros, como por ejemplo haciendo uso de un sistema de segmentación.

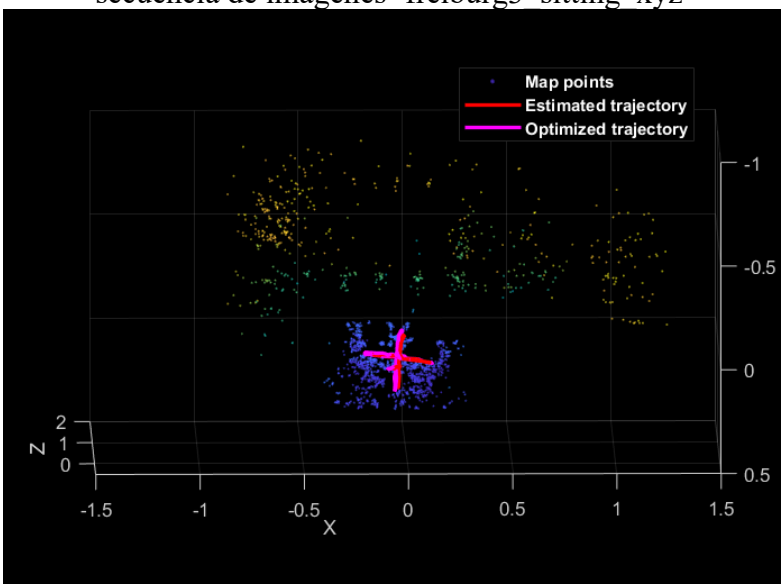
El mapeo local es la siguiente etapa durante la aplicación del algoritmo de reconstrucción de mapa y este inicia una vez se ha detectado un cuadro clave; dicho cuadro debe ser agregado a un directorio y los atributos del mapa observados por el nuevo cuadro clave son actualizados. Durante esta etapa se aplica un filtro de movimiento encargado de eliminar puntos característicos atípicos, discriminados en caso de que su aparición se presente tan solo en 1 o 2 cuadros claves. La aparición

de estos puntos se interpreta como un cambio brusco en la secuencia de imágenes, cambio que puede estar representado por un objeto dinámico en la escena. A continuación, se hallan más puntos de mapa mediante la triangulación de los puntos característicos del ORB en el cuadro clave actual y sus cuadros claves conectados. Posterior a ello se finaliza esta etapa realizando el refinamiento de los cuadros claves locales y los puntos de mapa, también se realiza la actualización de la dirección y la profundidad de las vistas.

La secuencia para cada cuadro durante del mapeo local finaliza con el cierre de bucle, este toma el cuadro actual procesado e intenta detectar que se cumplen los requisitos necesarios para considerar un posible cierre de bucle; dichos objetivos corresponden a que el cuadro clave actual no tenga conexión con el ultimo cuadro clave y tres de sus cuadros claves vecinos sean candidatos de bucle (un cuadro clave es identificado como candidato de bucle consultando imágenes en la base de datos que son visualmente similares al cuadro clave actual). Una vez se encuentra un candidato de bucle valido se calcula la pose relativa entre el cuadro clave candidato de bucle y el cuadro clave actual, esta pose relativa representa una transformación de similitud tridimensional.

Esta secuencia finaliza con la detección de un cierre de mapa, momento en que se determina que la trayectoria ha sido reconstruida a cabalidad y está lista para un previo procesamiento comparativo con los datos reales conocidos. Dicho procesamiento está determinado por la optimización del gráfico de pose de similitud para corregir el error acumulado de la reproyección de errores de cada vista de las poses de la cámara. Una vez realizado el proceso de optimización se actualiza las ubicaciones tridimensionales de los puntos del mapa utilizando las poses optimizadas y las escalas asociadas que en conjunto constituyen la trayectoria reconstruida como se muestra en la figura Figura 33.

Figura 33. Nube de puntos característicos y trayectoria de poses de cámara reconstruida para la secuencia de imágenes ‘freiburg3_sitting_xyz'

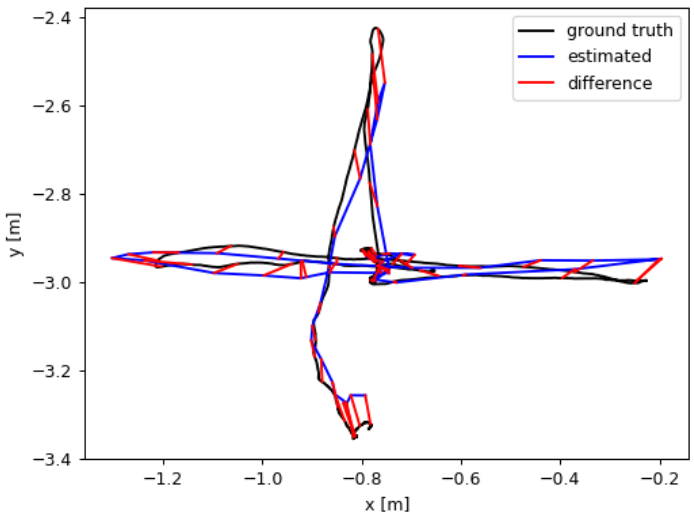



*Nota.* Fuente propia

Para evaluar estas trayectorias se deben comparar con la trayectoria de la verdad que proporciona la base de datos; para esto se debe exportar un archivo de texto que contenga los datos tanto de traslación y de rotación, concatenados con el vector de tiempos para cada cuadro clave que tuvo en cuenta el sistema. Se debe transformar los puntos y poses de cámara que nos entrega el algoritmo en formato *Rigid3d* a sus matrices de traslación y rotación, obteniendo así una matriz de ocho columnas.

Una vez obtenido el archivo de texto de la trayectoria estimada es necesario evaluar el desempeño haciendo uso de los algoritmos proporcionados por la base de datos para este fin. Los parámetros necesarios de entrada para su ejecución son dos y ambos se encuentran definidos por trayectorias expresadas en archivos de texto, el primero de ellos representa la trayectoria real obtenida directamente de la base de datos y el segundo constituye la trayectoria estimada obtenida mediante el sistema SLAM aplicado en Matlab sobre la secuencia de imágenes procesadas. Por último, en la Figura 34 se presenta el resultado grafico de la comparativa entre las dos trayectorias con la finalidad de visualizar las diferencias entre la trayectoria estimada y la real. Es pertinente mencionar que la utilización de este algoritmo de comparación implementado en el lenguaje de programación python se hizo necesario debido a que la secuencia de imágenes procesada es captada mediante una cámara monocular de dos vistas, la cual no proporciona información sobre la profundidad de los objetos, datos necesarios para la reconstrucción completa de la escala sobre la trayectoria, o dicho de otro modo al realizar la comparación de las trayectorias en el software Matlab, los errores medidos se incrementan porque la escala estimada de la trayectoria reconstruida no es una representación idónea de la escala real. El algoritmo de comparación implementado en python soluciona este problema mediante la asociación de las dos trayectorias y la evaluación de diferentes valores para la escala, hasta obtener aquella que arroje el menor error absoluto de trayectoria.

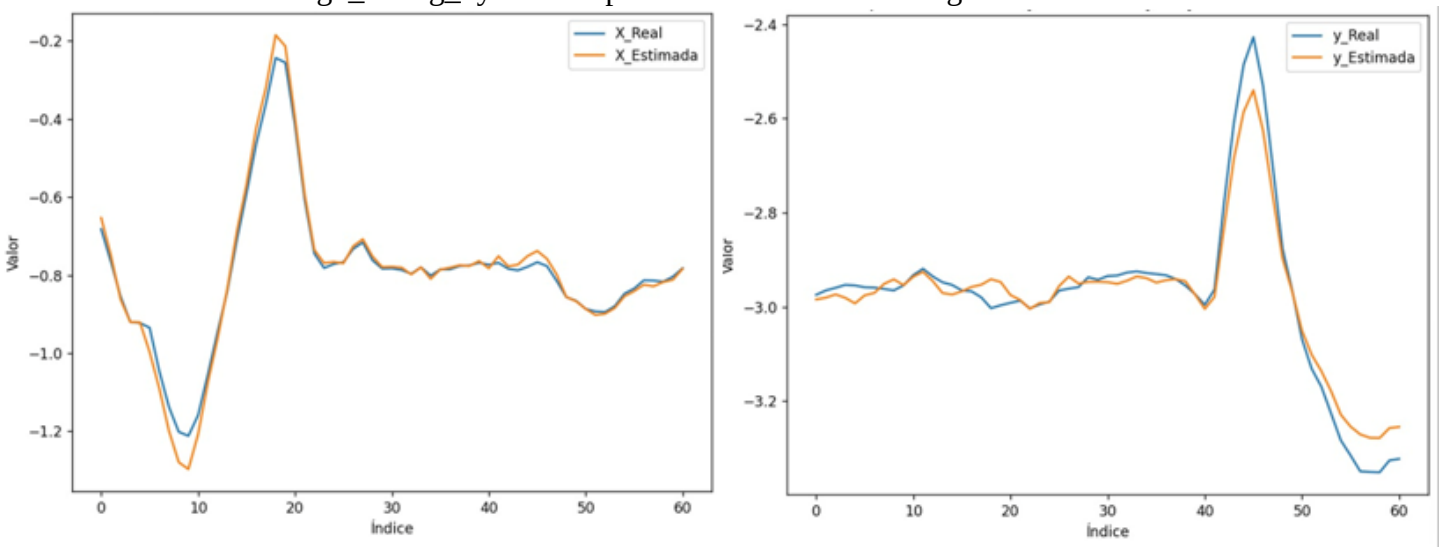
Figura 34. Comparación punto a punto de la trayectoria estimada y la trayectoria real para la secuencia de imágenes ‘freiburg3_sitting_xyz'




*Nota.* Fuente propia

Mediante la  Figura 35 se pretende ilustrar la asociación de cuadros claves utilizados para la reconstrucción de la trayectoria en la secuencia de imágenes ‘freiburg3_sitting_xyz’, frente al valor obtenido para cada uno de ellos tanto en el eje x como en el eje y por separado, con el motivo de evidenciar gráficamente el error de trayectoria absoluta en cada uno de los ejes *x* y *y* respectivamente, teniendo en cuenta que la trayectoria para este caso ha sido reconstruida sin la implementación del método de segmentación semántica.

Figura 35 Comparación independiente por eje x y y del error de trayectoria absoluta para la secuencia freiburg3_sitting_xyz sin la aplicación del método de segmentación semántica




Fuente: Propia

Tabla 6. Métricas de la comparativa entre la trayectoria estimada y la trayectoria real para la secuencia de imágenes ‘freiburg3_sitting_xyz'

| Métricas de los resultados | |
|---|---|
| Parámetro | Valor |
| Escala | 3,182761 |
| Numero de poses comparadas (Pares) | 60 |
| Error de trayectoria absoluto (Error cuadratico medio) | 0,051816m |
| Error de trayectoria absoluto (Media) | 0,042850m |
| Error de trayectoria absoluto (Mediana) | 0,033805m |
| Error de trayectoria absoluto (Desviación estándar) | 0,029134m |
| Error de trayectoria absoluto (mínimo) | 0,002746m |
| Error de trayectoria absoluto (máximo) | 0,125682m |

Nota. Fuente propia

### 4.3.1.1 Resultados de todas las secuencias evaluadas mediante el algoritmo SLAM visual

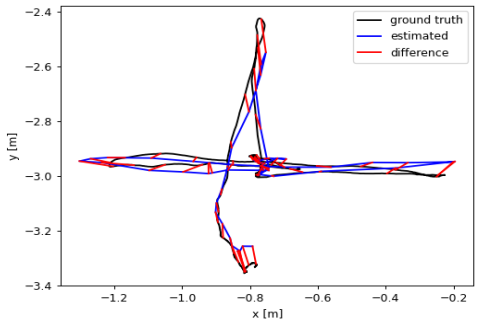
En este apartado se exponen los resultados gráficos mediante la Figura 36 de las evaluaciones para cada una de las secuencias de prueba sin la incidencia del algoritmo de segmentación semántica. Es importante mencionar que todas las trayectorias expuestas en la Figura 36 no fueron reconstruidas a cabalidad, puesto que en algunas de ellas el algoritmo fue incapaz de completar el seguimiento a través de toda la secuencia de imágenes; dichas secuencias son:

- Freiburg3_sitting_halsphere
- Freiburg3_walking_halsphere
- Freiburg3_walking_static

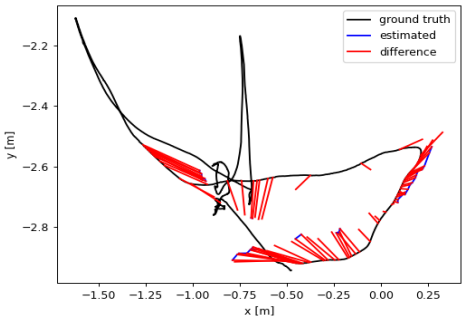
- Freiburg3_walking_xyz

Figura 36. Trayectorias estimadas mediante el uso del algoritmo SLAM visual sin la implementación del método de segmentación semántica

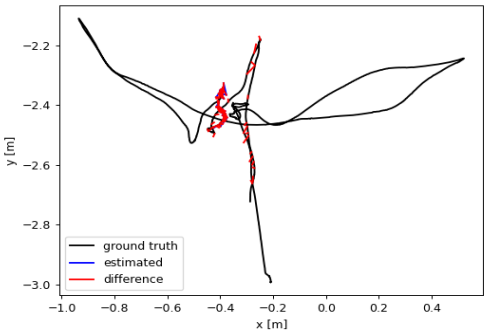
‘freiburg3_sitting_xyz’



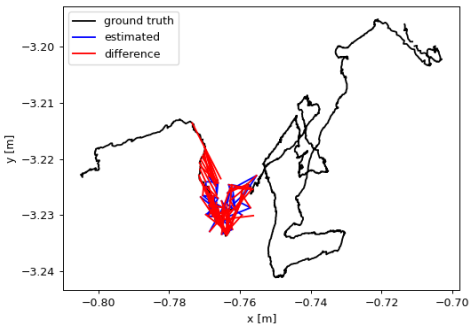

'freiburg3_walking_halfsphere'



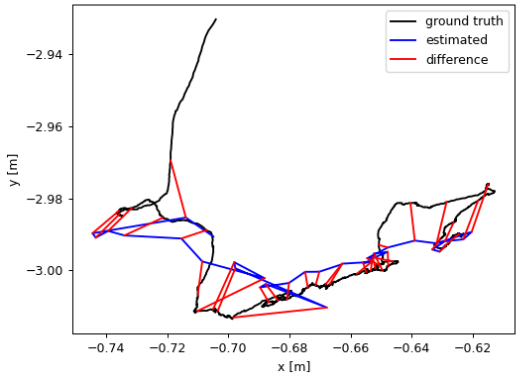

'freiburg3_sitting_halfsphere'



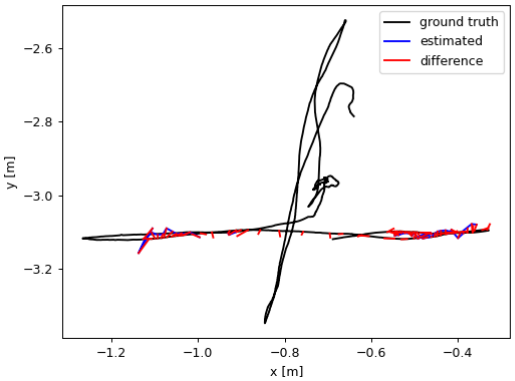

'freiburg3_walking_static'




'freiburg3_sitting_static’




'freiburg3_walking_xyz'

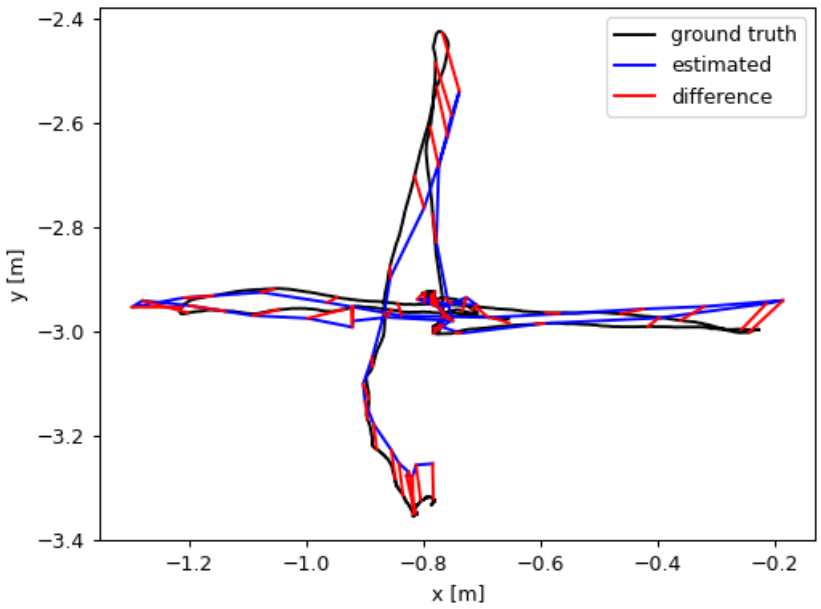


## 4.3.2. Evaluación mediante algoritmo SLAM con segmentación semántica

Esta sección evaluativa corresponde en gran similitud a la expuesta anteriormente en cuanto refiere a la aplicación del sistema SLAM, la principal diferencia radica en la integración de la segmentación semántica en la etapa de seguimiento la cual reduce el número de puntos característicos porque elimina aquellos que se encuentren al interior de un objeto dinámico. El cotejamiento de la máscara binaria obtenida para cada cuadro clave procesado, se encarga de brindar la información pertinente respecto de los objetos dinámicos segmentados en la escena. Posterior a la aplicación de la segmentación semántica sobre el sistema se puede evidencia en la Figura 38 la reducción de puntos característicos al interior de los objetos dinámicos segmentados en las escenas.

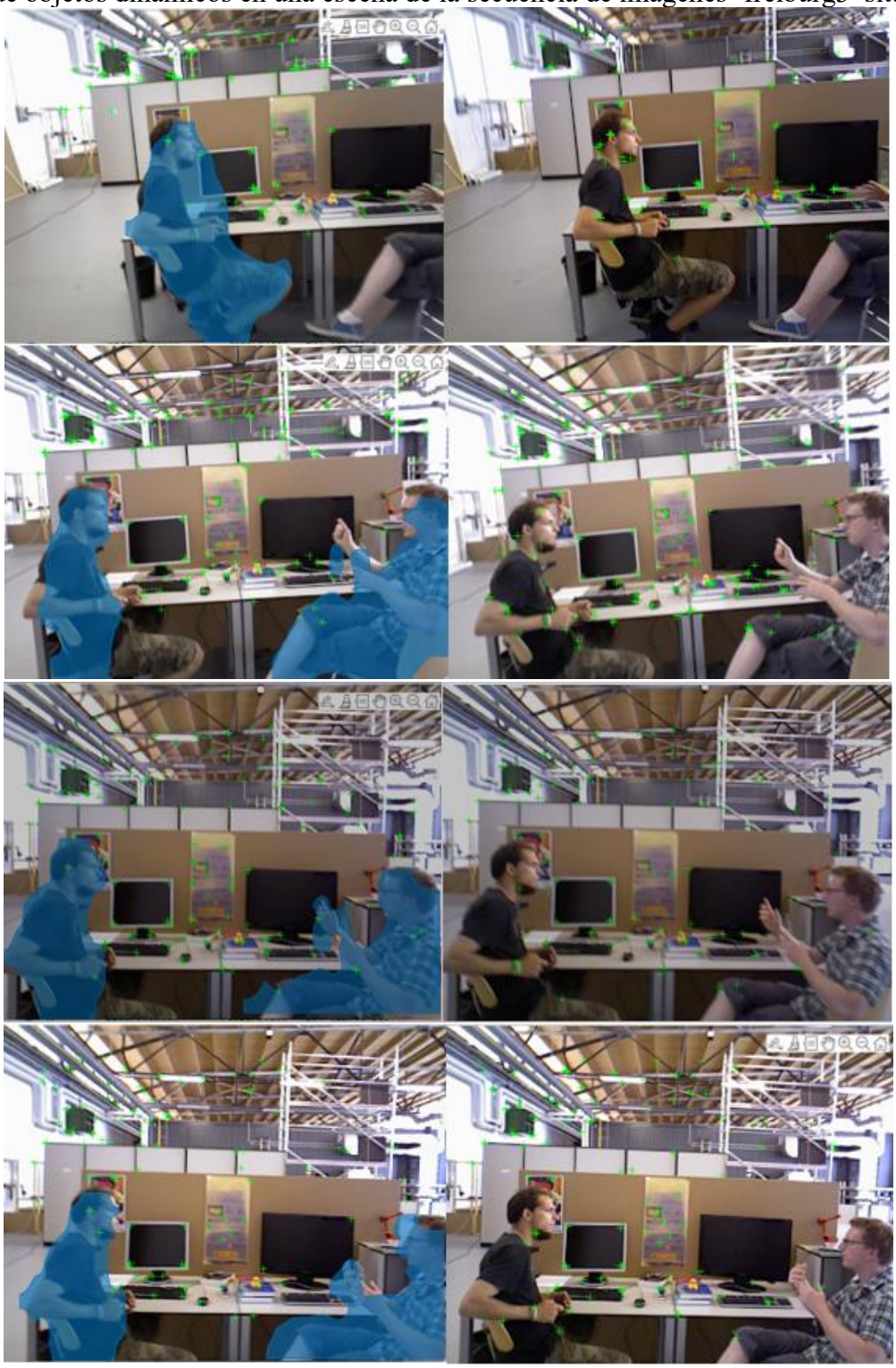
La comparativa de las trayectorias obtenidas una vez aplicada la segmentación semántica en el algoritmo de SLAM sobre el mismo conjunto de imágenes se ilustra en la Figura 37, los resultados obtenidos presentan cambios poco perceptibles gráficamente debido a que estos son proporcionales a la influencia de los objetos dinámicos presentes en la escena. En la Tabla 7 se exponen los resultados estadísticos obtenidos los cuales presentan diferencias del orden de los centímetros en casi todos los parámetros respecto a los resultados obtenidos del algoritmo SLAM sin influencia de la segmentación.

Figura 37. Comparación punto a punto de la trayectoria estimada con segmentación semántica y la trayectoria real para la secuencia de imágenes ‘freiburg3_sitting_xyz'




Nota. Fuente propia

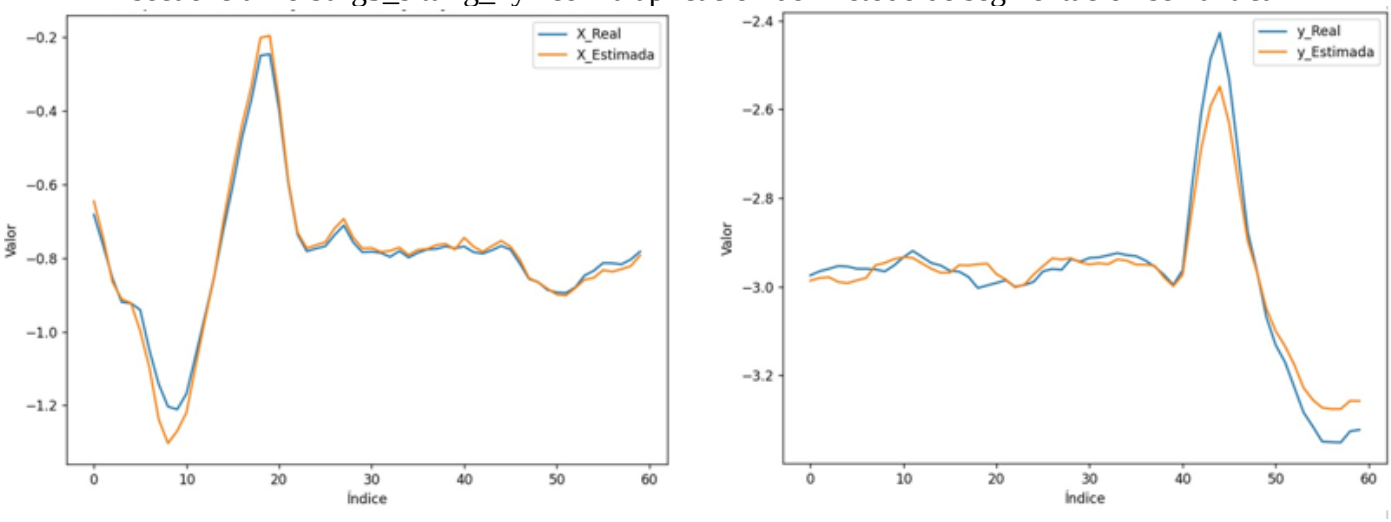
Figura 38. Efecto de la segmentación semántica sobre la reducción de puntos característicos al interior de objetos dinámicos en una escena de la secuencia de imágenes ‘freiburg3_sitting_xyz'



*Fuente:* Adaptado de *TUM School of Computation, Information and Technology*[Fotografía], por Computer Vision Group,2022.

Mediante la  Figura 39 se pretende ilustrar la asociación de cuadros claves utilizados para la reconstrucción de la trayectoria en la secuencia de imágenes ‘freiburg3_sitting_xyz’, frente al valor obtenido para cada uno de estos cuadros claves, tanto en el eje x como en el eje y discriminadamente con el objetivo de evidenciar gráficamente el error de trayectoria absoluta en cada uno de los ejes *x* y *y* respectivamente, teniendo en cuenta que la trayectoria para este caso ha sido reconstruida con la incidencia de la implementación del método de segmentación semántica.

Figura 39 Comparación independiente por eje x y y del error de trayectoria absoluta para la secuencia freiburg3_sitting_xyz con la aplicación del método de segmentación semántica




Fuente: Propia

Tabla 7. Métricas de la comparativa entre la trayectoria estimada con segmentación semántica y la trayectoria real para la secuencia de imágenes ‘freiburg3_sitting_xyz'

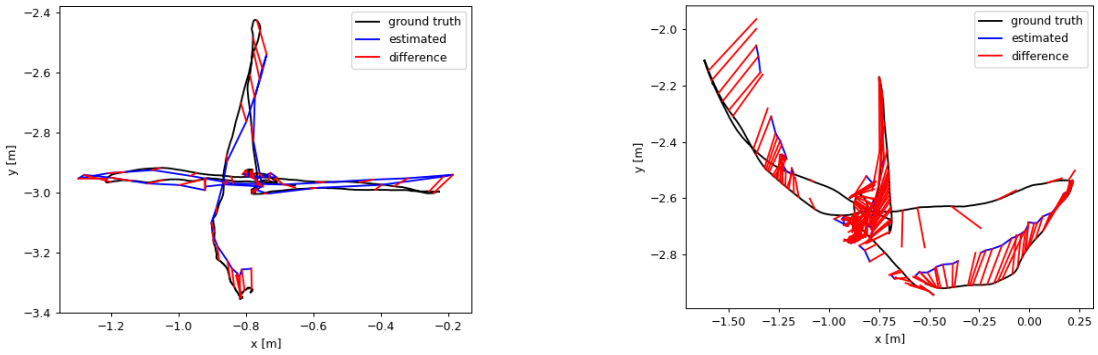
| Métricas de los resultados | |
|---|---|
| Parámetro | Valor |
| Escala | 3,285933 |
| Numero de poses comparadas (Pares) | 61 |
| Error de trayectoria absoluto (Error cuadratico medio) | 0,049411m |
| Error de trayectoria absoluto (Media) | 0,041023m |
| Error de trayectoria absoluto (Mediana) | 0,031554m |
| Error de trayectoria absoluto (Desviación estándar) | 0,027541m |
| Error de trayectoria absoluto (mínimo) | 0,002875m |
| Error de trayectoria absoluto (máximo) | 0,117015m |

## 4.3.2.1. Resultados de todas las secuencias evaluadas mediante el algoritmo SLAM visual con la integración del método de segmentación semántica

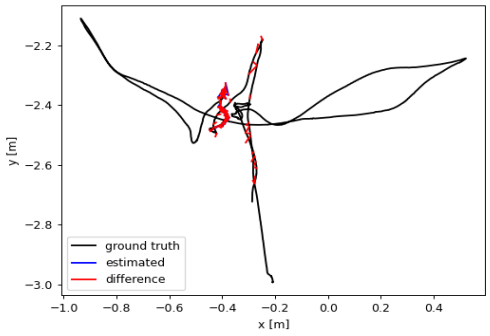
En este apartado se exponen los resultados gráficos mediante la Figura 40 de las evaluaciones para cada una de las secuencias de prueba una vez incorporado el método de segmentación semántica.

Figura 40. Trayectorias estimadas mediante el uso del algoritmo SLAM visual con la implementación del método de segmentación semántica

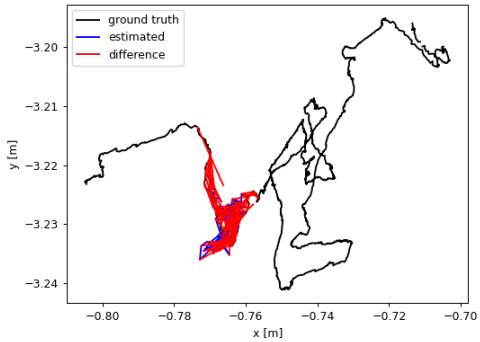
‘freiburg3_sitting_xyz’ 'freiburg3_walking_halfsphere'



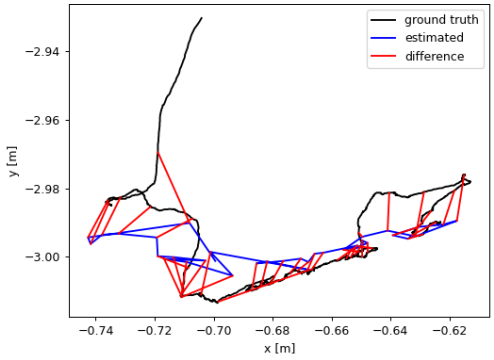
'freiburg3_sitting_halfsphere'



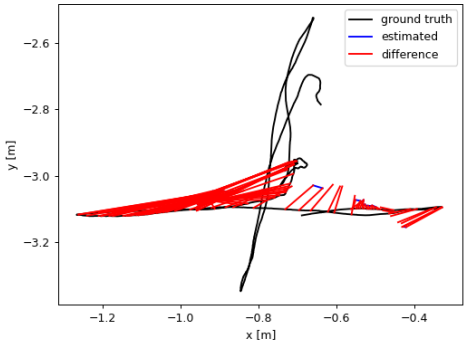

'freiburg3_walking_static'




'freiburg3_sitting_static'




'freiburg3_walking_xyz'

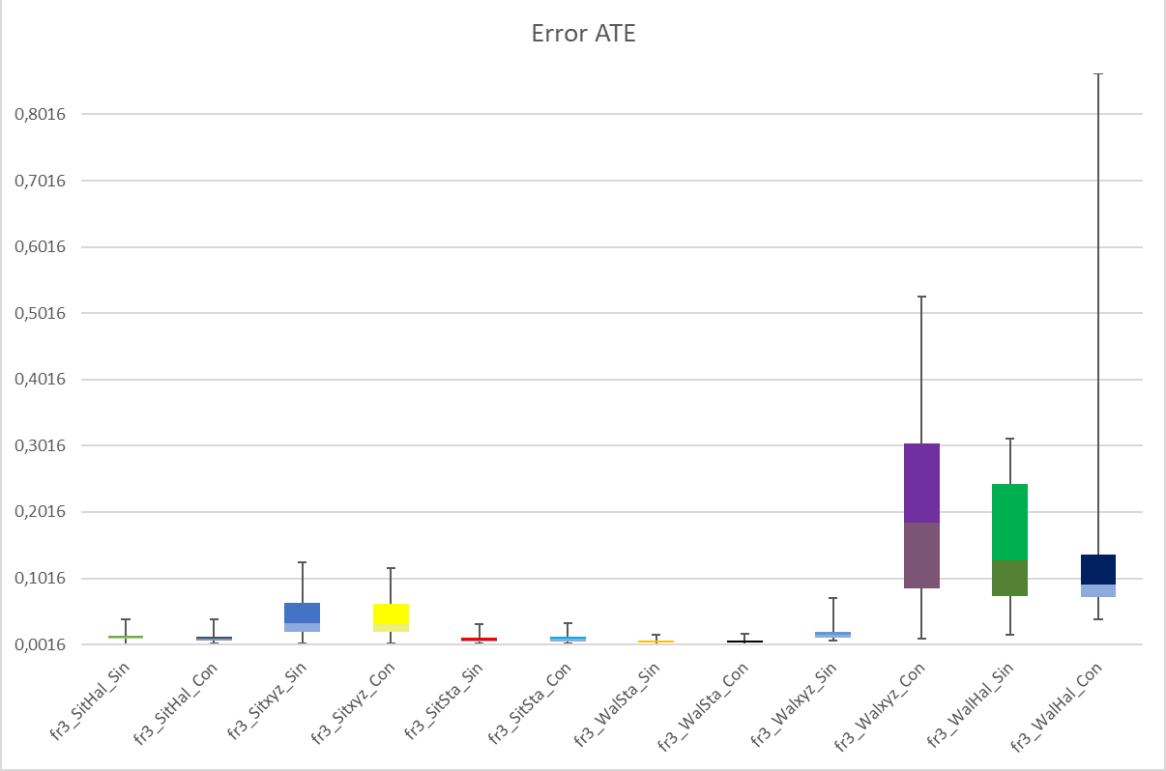



Se hace de gran interés mencionar, que a diferencia de los resultados obtenidos con el algoritmo SLAM visual puro expuestos en la Figura 36, para el caso particular de la secuencia Freiburg3_walking_halsphere esta vez mediante el procesamiento que incorpora el método de segmentación semántica, la trayectoria asociada a esta secuencia de imágenes se pudo reconstruir por completo siendo este ejemplo aquel que evidencia mayor impacto en el cambio del algoritmo encargado del procesamiento.

## 4.3.3. Comparación de los resultados

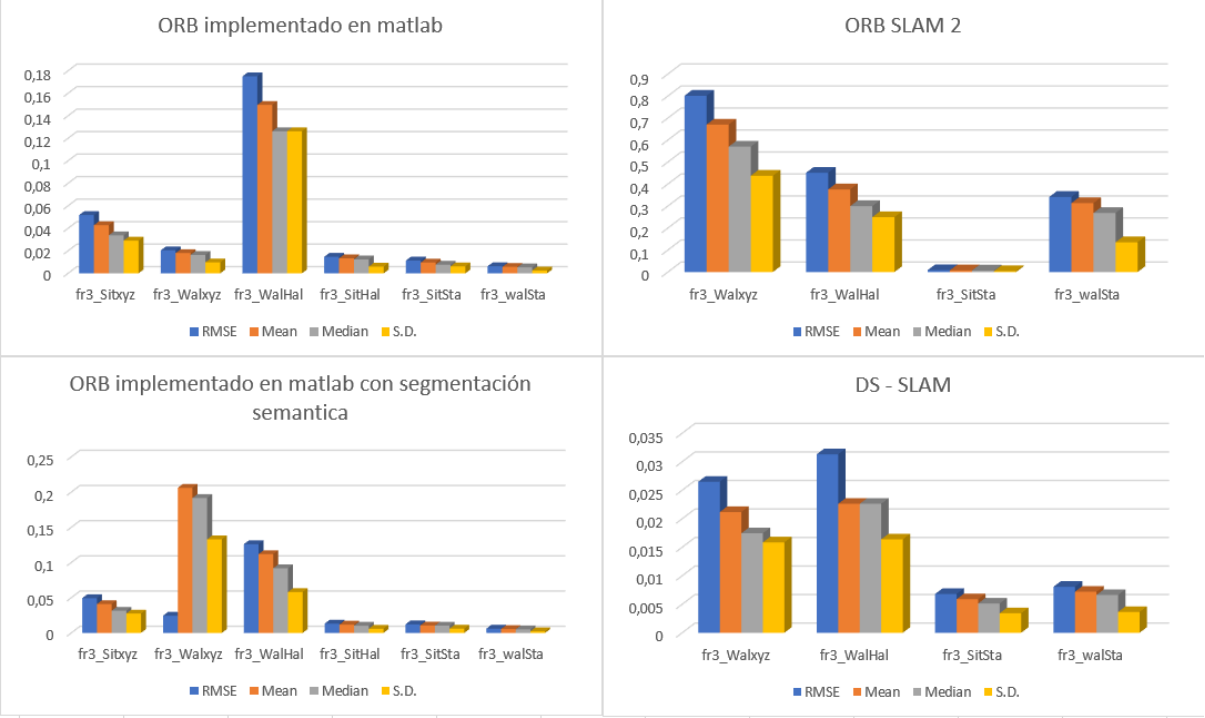
Una vez obtenidos todos los resultados para cada una de las secuencias sobre la evaluación en los dos algoritmos propuestos, toda la información se expone en la Figura 41 mediante un diagrama de cajas y bigotes donde se evidencian los errores de trayectoria absoluta.

Figura 41. Diagrama de cajas y bigotes para el error ATE




De la Figura 41 resulta pertinente interpretar que las secuencias que presentan menos movimiento sobre los ejes de la cámara o donde el entorno tiende a ser mayormente estático presentan un error ATE significativamente más bajo, como se evidencia en la secuencia Freiburg_sitting_static, que aquellas secuencias donde la cámara bascula y realiza movimientos sobre sus ejes o donde el entorno presenta mayor dinamismo como en la secuencia Freiburg3_walking_xyz. En el caso particular de la secuencia Freiburg3_sitting_xyz la cual presenta movimientos de la cámara a lo largo de sus ejes cartesianos, el sistema tiende a presentar un mayor ATE en promedio sin embargo este parámetro se ve atenuado por el efecto de la segmentación semántica durante el procesamiento con dicho algoritmo; En la secuencia Freiburg3_walking_halsphere se evidencia mayormente el impacto de incluir la segmentación semántica sobre el algoritmo SLAM visual porque tan solo en este caso se posibilita la reconstrucción total de la trayectoria estimada en contra parte al algoritmo SLAM puro que resulta incapaz de realizar la reconstrucción de la trayectoria por completo. Por otra parte, con la finalidad de contrastar los resultados obtenidos con otros algoritmos aplicados a las mismas secuencias de imágenes, se presenta en la Figura 42 un diagrama de barras donde se expone el desempeño obtenido por cada sistema.

Figura 42. Diagramas de barras para el rendimiento de diferentes algoritmos sobre las secuencias empleadas




De la Figura 42 es posible evidenciar que los algoritmos que implementan Segmentación semántica como lo son el DS-SLAM y el algoritmo de ORB implementado en este proyecto presenta mejoras con respecto a los algoritmos ORB-SLAM sin la incidencia de los métodos de segmentación, siendo el DS-SLAM mejor que el algoritmo propuesto, dado que la red empleada por este algoritmo presenta un mayor soporte de recursos computacionales y se ejecuta en condiciones idóneas para que su desarrollo no presente inconvenientes como por ejemplo tiempos de ejecución demasiado extendidos. Esto representa la mayor imposibilidad de hacerle frente con la red propuesta por el algoritmo empleado en este proyecto; sin embargo a pesar de que los recursos computacionales son limitados, la implementación de la segmentación semántica propuesta presenta mejoras con respecto a los algoritmos de ORB-SLAM logrando un tiempo de ejecución aceptable teniendo en cuenta que la segmentación se ejecuta para cada cuadro de las secuencias procesadas; también se puede concluir que las mejoras sobre el ATE son mucho más evidentes en las secuencias que presentan mayor influencia dinámica ya que el algoritmo ORB-SLAM es capaz de presentar buenos resultados en entornos estáticos.

A continuación se exponen mediante la Tabla 8 los valores precisos sobre los que se materializan los diagramas expuestos en la Figura 42.

Tabla 8. Resultados de error de trayectoria absoluta

| Secuencias | | Freiburg3 sittign_xyz | Freiburg3 walking_xyz | Freiburg3 Walking halfsphere | Freiburg3_Sitting_halfsphere | Freiburg3_Sitting Static | Freiburg3_walking Static |
|---|---|---|---|---|---|---|---|
| ORB implementado en Matlab | RMSE | 0,051816 | 0,020515 | 0,175148 | 0,014735 | 0,011459 | 0,006342 |
| | Mean | 0,04285 | 0,018041 | 0,149714 | 0,013453 | 0,009639 | 0,005808 |
| | Median | 0,033805 | 0,016428 | 0,126317 | 0,012395 | 0,007686 | 0,005355 |
| | S.D. | 0,029134 | 0,009765 | 0,126317 | 0,006012 | 0,006198 | 0,002547 |
| ORB implementado en Matlab con segmentación semántica | RMSE | 0,049411 | 0,0245243 | 0,126106 | 0,013294 | 0,012242 | 0,006075 |
| | Mean | 0,041023 | 0,206083 | 0,111937 | 0,011872 | 0,010578 | 0,005583 |
| | Median | 0,031554 | 0,191644 | 0,091862 | 0,010332 | 0,010339 | 0,005179 |
| | S.D. | 0,027541 | 0,132944 | 0,058077 | 0,005982 | 0,006162 | 0,002396 |
| ORB SLAM 2 | RMSE | - | 0,801255 | 0,4512938 | - | 0,008661 | 0,341436 |
| | Mean | - | 0,670056 | 0,3756166 | - | 0,007618 | 0,313265 |
| | Median | - | 0,569857 | 0,2997844 | - | 0,006817 | 0,268446 |
| | S.D. | - | 0,437586 | 0,2496192 | - | 0,004117 | 0,1354 |
| DS - SLAM | RMSE | - | 0,026494 | 0,0313056 | - | 0,00681 | 0,008074 |
| | Mean | - | 0,021179 | 0,0226248 | - | 0,005888 | 0,007201 |
| | Median | - | 0,017452 | 0,0226248 | - | 0,005132 | 0,006629 |
| | S.D. | - | 0,015863 | 0,0163846 | - | 0,003422 | 0,003649 |

# Capítulo 5

# Conclusiones y trabajos futuros

## 5.1. Conclusiones

- Se Concluyó que la integración de un algoritmo de segmentación semántica sobre un sistema SLAM visual como mecanismo de reducción del impacto de los objetos dinámicos presentes en la secuencia de imágenes procesada, resulta eficiente y es capaz de disminuir significativamente el error obtenido del cotejamiento de la trayectoria reconstruida contra la trayectoria real; aunque para garantizar un impacto positivo producto de esta integración es necesario que la escena cuente con objetos dinámicos, sino por el contrario la segmentación semántica puede constituirse en una pérdida de información innecesaria por la detección de puntos dinámicos ineficientes en la escena.
- Se determinó que mediante el uso de un filtro para puntos característicos atípicos se obtienen mejoras significativas frente a las métricas de evaluación establecidas, ya que al eliminar puntos que no se repitan al menos en tres cuadros claves, complementa significativamente la segmentación para los casos en que esta no es del todo eficiente.
- Se estableció que la segmentación semántica, en algunos casos, representa una condición necesaria más no suficiente para la obtención de la trayectoria que se desea reconstruir, puesto que esta suprime la información dinámica de la escena pero no establece un nuevo método de adquisición de características, por lo que es común en escenas que cuentan con un área dinámica muy grande que el sistema SLAM visual se quede sin la información mínima necesaria para terminar a cabalidad la reconstrucción de la trayectoria y el mapa local.

## 5.2. Trabajos futuros

Con el fin de mejorar los resultados obtenidos y expuestos mediante el presente documento, se plantean algunos conceptos e ideas prestar a tomar en cuenta para el desarrollo de trabajos futuros y complementarios.

- Construcción de una red neuronal más robusta capas de brindar mayor precisión en la segmentación de los objetos dinámicos, con la finalidad de perder la menor cantidad de información útil posible durante la eliminación de puntos aparentemente dinámicos, mal clasificados por imprecisión durante el proceso de la definición del área de segmentación.

- Diseño e incorporación de un algoritmo extractor de características, al ciclo general de SLAM visual, posterior a la aplicación de la segmentación semántica y la correspondiente eliminación de puntos dinámicos, con la finalidad de garantizar la información necesaria durante la ejecución del sistema general para la reconstrucción total de la trayectoria realizada por la cámara y el mapeo local

# ANEXOS

# Adaptación del código de programación

Durante el desarrollo de el ítem “Anexos” se especifican los aportes y adaptaciones realizadas sobre el código original del algoritmo ORB-SLAM extraídos de mathworks, para el buen desarrollo, ejecución y evaluación del algoritmo propuesto.

En la Figura A 1 se ilustra el código con el cual se agregan los conjuntos de imágenes para ser procesados por el algoritmo, este lo realiza sin hacer una descarga en línea, fijando el lugar en donde se descargo el conjunto de imágenes previamente dentro del equipo local.

Figura A 1 Carga del conjunto de imágenes al sistema

```
%imageFolder   = [dataFolder,'rgbd_dataset_freiburg3_long_office_household/rgb/'];
%imds          = imageDatastore(imageFolder);
dataFolder      = fullfile('rgbd_dataset_freiburg3_sitting_xyz', filesep);
%options         = weboptions('Timeout', Inf);
%tgzFileName     = [dataFolder, 'fr3_office.tgz'];

imageFolder   = [dataFolder,'rgb'];
imds          = imageDatastore(imageFolder);
% Inspect the first image
currFrameIdx = 1;
currI = readimage(imds, currFrameIdx);
himage = imshow(currI);
```

Fuente: Propia

La creación de la bolsa de palabras visuales se hace en base a las imágenes utilizadas en cada prueba del algoritmo haciendo uso de la función que se ilustra en la Figura A 2.

Figura A 2 Creación de la bolsa de palabras visuales

```
bag = bagOfFeatures(imds,CustomExtractor=@helperORBFeatureExtractorFunction, TreeProperties=[3, 10], StrongestFeatures=1);
```

Fuente: Propia

Se carga la red pre-entrenada “maskrcnn_object_person_car_v2.mat” para la segmentación de los objetos dinámicos como se muestra en la

Figura A 3 Red pre-entrenada maskrcnn

```
%segmenetacion
detector = load(fullfile(dataFolder,"maskrcnn_object_person_car_v2.mat"));
net = detector.net;
```

Fuente: propia

Una vez empieza la etapa de seguimiento se debe agregar las mascaras binarias de la segmentación con el mismo tamaño que tiene el conjunto de imágenes como se muestra en la

Figura A 4 Creación de matrices unitarias

```
masksLvl1= masks(1:480,1:640,1);
masksLvl2= masks(1:480,1:640,2);
```

Fuente: propia

En el seguimiento se realiza el cotejo de la máscara binaria con las imágenes seleccionadas como imágenes claves durante la etapa de seguimiento con el objetivo de eliminar los puntos característicos que se encuentran al interior de una región dinámica como se muestra en Figura A 5.

Figura A 5 Segmentación de objetos dinámicos y filtrado de puntos característicos

```
%
    [MaskSize] = size(masks);
if numel(MaskSize) == 3 && length(masks) ~= 0
masksLvl1= masks(1:480,1:640,1);
masksLvl2= masks(1:480,1:640,2);
for Px = 1:480
    for Py = 1:640
        if masksLvl1(Px,Py) || masksLvl2(Px,Py) == 1
            masksFrame(Px,Py) = 1;
        else
            masksFrame(Px,Py) = 0;
        end
    end
end
else
    masksFrame = masks;
end
 if length(masksFrame) ~= 0
for Pnt = 1:numel(featureIdx)
    [maskC] = round(currPoints.Location(featureIdx(Pnt),1:2),0);
    Cx = maskC(1);
    Cy = maskC(2);

    if masksFrame(Cy,Cx) == 1
        featureIdx(Pnt)= [0];
        mapPointsIdx(Pnt)=[0];
    end
end
    end
featureIdx(find(featureIdx==0)) = [];
mapPointsIdx(find(mapPointsIdx==0)) = [];
%
```

Fuente: Propia

Una vez termina todo el proceso los datos encontrados en Matlab serán guardados en un archivo de extensión de texto (.txt) bajo el siguiente código (ver Figura A 6) para su posterior evaluación con la trayectoria real proporcionada por la base de datos.

Figura A 6 Exportación de datos para evaluación

```
%Datos Exportados
  TransEstimada    = vertcat(optimizedPoses.AbsolutePose.Translation);
  % RotEstimada = vertcat(optimizedPoses.AbsolutePose.Translation);
  for i=1:currKeyFrameId
      ratoacion(i,:) = rotm2quat(optimizedPoses.AbsolutePose(i,:).Rotation);

  end
Rot = vertcat(ratoacion);
ThRotTrasEsti = horzcat(TransEstimada,Rot);
writematrix(ThRotTrasEsti,"TrayectoriaEstimadaConSittingxty.txt",'Delimiter','tab')
```

```
for i=1:currKeyFrameId
      frame = addedFramesIdx(i,:)
      Valores(i,:) = Times(frame,:)

end
VerdadVerdadera = horzcat(Valores,ThRotTrasEsti);
writematrix(VerdadVerdadera,"VerdadwalkingxyzCon1.txt",'Delimiter','tab')
```

# Referencias


[1] J. Cremona, L. Uzal, and T. Pire, "Wganvo: Monocular visual odometry based on ge-nerative adversarial networks," arXiv preprint arXiv:2007.13704, 2020.

[2] J. Li, X. Zhang, J. Li, Y. Liu, and J. Wang, "Building and optimization of 3d semantic map based on lidar and camera fusion," Neurocomputing, vol. 409, pp. 394–407, 2020.

[3] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "Monoslam: Real-time single camera slam," IEEE transactions on pattern analysis and machine intelligence, vol. 29, no. 6, pp. 1052–1067, 2007.

[4] G. Klein and D. Murray, "Parallel tracking and mapping for small ar workspaces," in 2007 6th IEEE and ACM international symposium on mixed and augmented reality. IEEE, 2007, pp. 225–234.

[5] J. Engel, T. Schöps, and D. Cremers, "Lsd-slam: Large-scale direct monocular slam," in
European conference on computer vision. Springer, 2014, pp. 834–849.

[6] C. Forster, M. Pizzoli, and D. Scaramuzza, "Svo: Fast semi-direct monocular visual odometry," in 2014 IEEE international conference on robotics and automation (ICRA). IEEE, 2014, pp. 15–22.

[7] C. Forster, Z. Zhang, M. Gassner, M. Werlberger, and D. Scaramuzza, "Svo: Semidi- rect visual odometry for monocular and multicamera systems," IEEE Transactions on Robotics, vol. 33, no. 2, pp. 249–265, 2016.

[8] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "Orb-slam: a versatile and accuratemonocular slam system," IEEE transactions on robotics, vol. 31, no. 5, pp. 1147–1163, 2015.

[9] R. Mur Artal and J. D. Tardos, "Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras," IEEE transactions on robotics, vol. 33, no. 5, pp. 1255–1262,2017.

[10] X. Lu, H. Wang, S. Tang, H. Huang, and C. Li, "Dm-slam: Monocular slam in dynamic environments," Applied Sciences, vol. 10, no. 12, p. 4252, 2020.

[11] Y. Liu and J. Miura, "Rds-slam: real-time dynamic slam using semantic segmentation methods," IEEE Access, vol. 9, pp. 23 772–23 785, 2021.

[12] G. Tian, L. Liu, J. Ri, Y. Liu, and Y. Sun, "Objectfusion: An object detection and segmentation framework with rgb-d slam and convolutional neural networks," Neuro-computing, vol. 345, pp. 3–14, 2019.

[13] S. Sengupta and P. Sturgess, “Semantic octree: Unifying recognition, reconstruction and representation via an octree constrained higher order mrf,” in 2015 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2015, pp. 1874–1879.

[14] S. Vasudevan and R. Siegwart, “Bayesian space conceptualization and place classification for semantic maps in mobile robotics,” Robotics and Autonomous Systems, vol. 56, no. 6,pp. 522–537, 2008.

[15] C. Yu, Z. Liu, X.-J. Liu, F. Xie, Y. Yang, Q. Wei, and Q. Fei, “Ds-slam: A semantic visual slam towards dynamic environments,” in 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2018, pp. 1168–1174.

[16] L. Xiao, J. Wang, X. Qiu, Z. Rong, and X. Zou, “Dynamic-slam: Semantic monocu- lar visual localization and mapping based on deep learning in dynamic environment,” Robotics and Autonomous Systems, vol. 117, pp. 1–16, 2019.

[17] S. Jin, L. Chen, R. Sun, and S. McLoone, “A novel vslam framework with unsupervised semantic segmentation based on adversarial transfer learning,” Applied Soft Computing,vol. 90, p. 106153, 2020.

[18] L. Cui and C. Ma, “Sdf-slam: Semantic depth filter slam for dynamic environments,”
IEEE Access, vol. 8, pp. 95 301–95 311, 2020.

[19] F. Zhong, S. Wang, Z. Zhang, and Y. Wang, “Detect-slam: Making object detection and slam mutually beneficial,” in 2018 IEEE Winter Conference on Applications of Computer Vision (WACV). IEEE, 2018, pp. 1001–1010.

[20] S. Han and Z. Xi, “Dynamic scene semantics slam based on semantic segmentation,” IEEE Access, vol. 8, pp. 43 563–43 570, 2020.

[21] C. Yu, Z. Liu, X.-J. Liu, F. Xie, Y. Yang, Q. Wei, and Q. Fei, “Ds-slam: A semantic visual slam towards dynamic environments,” in 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2018, pp. 1168–1174.

[22] Z. Niu, X. Zhao, J. Sun, L. Tao, and B. Zhu, “A continuous positioning algorithm basedon rtk and vi-slam with smartphones,” IEEE Access, vol. 8, pp. 185 638–185 650, 2020.

[23] Fang, B., Mei, G., Yuan, X., Wang, L., Wang, Z. y Wang, J. (2021). SLAM visual para la navegación de robots en instalaciones sanitarias. Reconocimiento de patrones, 113, 107822.

[24] Su, P., Luo, S., & Huang, X. (2022). Real-Time Dynamic SLAM Algorithm Based on Deep Learning. IEEE Access, 10, 87754-87766.

[25] Zhang, X., Zhang, R., & Wang, X. (2022). Visual SLAM Mapping Based on YOLOv5 in Dynamic Scenes. Applied Sciences, 12(22), 11548.

[26] Peng, J., Hou, Y., Xu, H., & Li, T. (2022). Dynamic visual SLAM and MEC technologies for B5G: a comprehensive review. EURASIP Journal on Wireless Communications and Networking, 2022(1), 1-23.

[27] Taketomi, T., Uchiyama, H. e Ikeda, S. (2017). Algoritmos SLAM visuales: una encuesta de 2010 a 2016. IPSJ Transactions on Computer Vision and Applications , 9 (1), 1-11.

[28] Montaner Cardoso, C. (2021). VSLAM para robótica móvil en la industria 4.0: Implementación de sistemas de navegación SLAM en escenarios interiores.

[29] Cerdá Koehler, A. J. (2012). SLAM: Simultaneous Localization and Mapping en la Carolo-cup (Doctoral dissertation, Universitat Politècnica de València).

[30] Ali, N. M. E. E., & MARTINEZ MONTIEL, J. O. S. É. (2018). Visual monocular SLAM for minimally invasive surgery and its application to augmented reality (Doctoral dissertation, Université de Strasbourg).

[31] Yousif, K., Bab-Hadiashar, A. y Hoseinnezhad, R. (2015). Una visión general de la odometría visual y SLAM visual: aplicaciones a la robótica móvil. Sistemas industriales inteligentes, 1 (4), 289-311.

[32] Xu, X., Zhang, L., Yang, J., Cao, C., Wang, W., Ran, Y., ... y Luo, M. (2022). Una revisión de los sistemas de golpe de fusión multisensor basados en 3D LIDAR. Detección remota, 14 (12), 2835.

[33] Rodríguez Plaza, Ó. T. (2016). Localización y mapeo visual monocular para robot móvil terrestre aplicado a la inspección ultrasónica aeronáutica (Doctoral dissertation, Industriales).

[34] Nava, M. R. Z., González, C. F. M., Galicia, H. A., & Flores, J. M. P. (2013). Marcadores para la realidad aumentada para fines educativos. ReCIBE. Revista electrónica de Computación, Informática, Biomédica y Electrónica, (3).

[35] Mohanty, V., Agrawal, S., Datta, S., Ghosh, A., Sharma, VD y Chakravarty, D. (2016). Deepvo: un enfoque de aprendizaje profundo para la odometría visual monocular. preimpresión de arXiv arXiv:1611.06069.

[36] Qureshi, U. y Golnaraghi, F. (2017). Un algoritmo para la calibración en campo de una IMU MEMS. Revista de sensores IEEE, 17 (22), 7479-7486.

[37] Newcombe, RA, Lovegrove, SJ y Davison, AJ (2011, noviembre). DTAM: seguimiento y mapeo densos en tiempo real. En 2011 conferencia internacional sobre visión artificial (pp. 2320-2327). IEEE.

[38] Engel, J., Schöps, T. y Cremers, D. (septiembre de 2014). LSD-SLAM: SLAM monocular directo a gran escala. En European conference on computer vision (pp. 834-849). Springer, Cham.

[39] Wang, R., Schworer, M. y Cremers, D. (2017). Stereo DSO: odometría visual dispersa directa a gran escala con cámaras estéreo. En Actas de la Conferencia Internacional IEEE sobre Visión por Computador (págs. 3903-3911).

[40] Forster, C., Pizzoli, M. y Scaramuzza, D. (2014, mayo). SVO: Odometría visual

monocular semidirecta rápida. En la conferencia internacional IEEE de 2014 sobre robótica y automatización (ICRA) (págs. 15-22). IEEE.

[41] Pire, T., Fischer, T., Castro, G., De Cristóforis, P., Civera, J., & Berlles, JJ (2017). S-PTAM: seguimiento y mapeo paralelo estéreo. Robótica y Sistemas Autónomos, 93 , 27-42.

[42] Mur-Artal, R., Montiel, JMM y Tardos, JD (2015). ORB-SLAM: un sistema SLAM monocular versátil y preciso. Transacciones IEEE sobre robótica, 31 (5), 1147-1163.

[43] Mistry, D. y Banerjee, A. (2017). Comparación de enfoques de detección y coincidencia de características: SIFT y SURF. GRD Journals-Global Research and Development Journal for Engineering , 2 (4), 7-13.

[44] Ballesta, M., Gil, A., Reinoso, O., & Ubeda, D. (2010). Análisis de detectores y descriptores de características visuales en slam en entornos interiores y exteriores. Revista Iberoamericana de Automática e Informática Industrial RIAI, 7(2), 68-80.

[45] Santos, D., Dallos, L., & Gaona-García, P. A. (2020). Algoritmos de rastreo de movimiento utilizando técnicas de inteligencia artificial y machine learning. Información tecnológica, 31(3), 23-38.

[46] Tyagi, D. ALGORTIMO DE HARRIS PARA DETECCIÓN DE ESQUINAS.

[47] Calonder, M., Lepetit, V., Ozuysal, M., Trzcinski, T., Strecha, C. y Fua, P. (2011). BREVE: Cálculo de un descriptor binario local muy rápido. Transacciones IEEE sobre análisis de patrones e inteligencia artificial, 34 (7), 1281-1298.

[48] Capua, FR, Sansoni, S. y Moreyra, ML (noviembre de 2018). Análisis comparativo de algoritmos visual-SLAM aplicados a entornos frutales. En 2018 Congreso Argentino de Control Automático (AADECA) (pp. 1-6). IEEE.

[49] Structure From Motion From Multiple Views [Articulo], por mathworks, 2022, (https://la.mathworks.com/help/vision/ug/structure-from-motion-from-multiple-views.html)

[50] Estévez, C. A. V. (2012). Fusión de SLAM monocular con sensores inerciales para la estimación de la orientación aplicando filtro Kalman.

[51] Fabresse, F. R., Caballero, F., Maza, I., & Ollero, A. Localización de vehıculos aéreos basada en 3D RO-SLAM con inicialización no retardada empleando mezcla de Gaussianas.

[52] Cuevas, E., Zaldívar, D., & Pérez, M. (2016). Procesamiento digital de imágenes con MATLAB & Simulink. Ra-Ma.

[53] Gómez Trejos, D. V., & Guerrero Guzmán, A. (2016). Estudio y análisis de técnicas para procesamiento digital de imágenes.

[54] Nieto Flores, A. C. (2018). Evolución del procesamiento digital de imágenes.

[55] Muñoz Manso, R. (2014). Sistema de visión artificial para la detección y lectura de matrículas.

[56] Nogué, A. L. B. E. R. T., & Antiga, J. O. R. D. I. (2012). Aplicación práctica de la visión artificial en el control de procesos industriales. Gobierno De España (Ministerio De Educación).

[57] Computer-vision [Articulo], por IBM, 2022, (https://www.ibm.com/co-es/topics/computer-vision)

[58] Valencia Díaz, E. (2007). Procesado de imagen digital en color: Adquisición, Análisis Colorimétrico y Realce. Universitat Politècnica de Catalunya.

[59] Marcos, A. G., de Pisón Ascacíbar, F. J. M., Elías, F. A., Limas, M. C., Meré, J. B. O., & González, E. P. V. (2006). Técnicas y algoritmos básicos de visión artificial. Técnicas y Algoritmos Básicos de Visión Artificial.

[60] Gómez Trejos, D. V., & Guerrero Guzmán, A. (2016). Estudio y análisis de técnicas para procesamiento digital de imágenes.

[61] Urzúa, A. C. (2019). Técnicas de procesamiento digital de imágenes. Revista de Marina Nº, 969, 68-71.

[62] Palomino, N. L. S., & Concha, U. N. R. (2009). Técnicas de segmentación en procesamiento digital de imágenes. Revista de investigación de Sistemas e Informática, 6(2), 9-16.

[63] Atienza Vanacloig, V. L. (2011). Digitalización de imágenes con ayuda del histograma.

[64] Monferrán, D., Gambini, M. J., & Frery Orgambide, A. C. (2017, August). Evaluación del error en la detección de puntos de borde en imágenes SAR polarimétricas. In XIX Workshop de Investigadores en Ciencias de la Computación (WICC 2017, ITBA, Buenos Aires).

[65] DIAZ, L. E. N., & ARCEO, L. E. C. (2018). Algoritmo rápido de la transformada de Hough para detección de líneas rectas en una imagen.

[66] Suárez, P., & Villavicencio, M. Detección de Contornos utilizando el Algoritmo Canny en Imágenes Cross-Espectrales Fusionadas (Canny Edge Detection in Cross-Spectral Fused Images). Comité Editorial, 16.

[67] Triana, N., Jaramillo, A. E., Gutiérrez, R. M., & Rodríguez, C. A. (2016). Técnicas de umbralización para el procesamiento digital de imágenes de GEM-Foils. Scientia et technica, 21(4), 352-359.

[68] Castrillo Leal, Y. (2017). Evaluación de técnicas de segmentación de imágenes de ultrasonido por crecimiento de regiones.

[69] Deng, L. y Yu, D. (2014). Aprendizaje profundo: métodos y aplicaciones. Fundamentos y tendencias® en el procesamiento de señales, 7 (3–4), 197-387.

[70] LeCun, Y., Bengio, Y. y Hinton, G. (2015). Aprendizaje profundo. naturaleza, 521 (7553), 436-444.

[71] Alonso Ruiz, Í., Murillo Arnal, A. C., & Cambra Linés, A. B. Segmentación semántica con modelos de deep learning y etiquetados no densos.

[72] Izaurieta, F., & Saavedra, C. (2000). Redes neuronales artificiales. Departamento de Física, Universidad de Concepción Chile.

[73] Segmentacion semántica [Articulo], por mathworks, 2022, (https://la.mathworks.com/solutions/image-video-processing/semantic-segmentation.html#:~:text=%C2%BFQu%C3%A9%20es%20la%20segmentaci%C3%B3n%20sem%C3%A1ntica,p%C3%ADxeles%20que%20conforman%20distintas%20categor%C3%ADas.)

[74] Larranaga, P., Inza, I., & Moujahid, A. (1997). Tema 8. redes neuronales. Redes Neuronales, U. del P. Vasco, 12, 17.

[75] Temprano Coleto, E. (2020). Métodos de aprendizaje profundo para la segmentación semántica de personas.

[76] Castro, C. B. P. (2010). Cómputo evolutivo como enfoque en la descripción del contenido de la imagen aplicado a la segmentación y el reconocimiento de objetos (Doctoral dissertation, tesis de maestría).

[77] López-Saca, F. (2019). Clasificación de imágenes usando redes neuronales convolucionales (Master's thesis, Universidad Autónoma Metropolitana (México). Unidad Azcapotzalco. Coordinación de Servicios de Información.).

[78] Rodríguez González, J., & Ugalde Saborio, E. (2021). Impacto de la estandarización y escalado: factor para predicción de costos en proyectos a través de una red neuronal artificial. Ingeniare. Revista chilena de ingeniería, 29(2), 265-275.

[79] Chen, LC, Papandreou, G., Kokkinos, I., Murphy, K. y Yuille, AL (2017). Deeplab: Segmentación semántica de imágenes con redes convolucionales profundas, convolución atrosa y crfs totalmente conectados. Transacciones IEEE sobre análisis de patrones e inteligencia artificial, 40 (4), 834-848.

[80] Porres de la Haza, M. J. (2015). Ejemplo de aplicación del remuestreo mediante interpolación bilineal.

[81] Wang, F., Jiang, M., Qian, C., Yang, S., Li, C., Zhang, H., ... y Tang, X. (2017). Red de atención residual para clasificación de imágenes. En Actas de la conferencia IEEE sobre visión artificial y reconocimiento de patrones (págs. 3156-3164).