EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA

TECNOLÓGICO NACIONAL DE MÉXICO

# Tecnológico Nacional de México

## Centro Nacional de Investigación y Desarrollo Tecnológico

# Tesis de Maestría

**Mapeo visual de interiores con robot móvil.**


presentada por
**Ing. Katia Rubit Benítez Castro**


como requisito para la obtención del grado de
**Maestra en Ciencias Computacionales**

Director de tesis
**Dr. José Ruiz Ascencio**

**Cuernavaca, Morelos, México. Enero de 2023**

cenidet
Centro Nacional de Investigación y Desarrollo Tecnológico

Centro Nacional de Investigación y Desarrollo Tecnológico
Departamento de Ciencias Computacionales

Cuernavaca, Mor., 23/septiembre/2022

OFICIO No. DCC/075/2022
**Asunto:** Aceptación de documento de tesis
CENIDET-AC-004-M14-OFICIO

**DR. CARLOS MANUEL ASTORGA ZARAGOZA**
**SUBDIRECTOR ACADÉMICO**
**PRESENTE**

Por este conducto, los integrantes de Comité Tutorial de la C. KATIA RUBIT BENITEZ CASTRO, con número de control M19CE054, de la Maestría en Ciencias de la Computación, le informamos que hemos revisado el trabajo de tesis de grado titulado **"MAPEO VISUAL DE INTERIORES CON ROBOT MÓVIL"** y hemos encontrado que se han atendido todas las observaciones que se le indicaron, por lo que hemos acordado aceptar el documento de tesis y le solicitamos la autorización de impresión definitiva.

**DR. JOSÉ RUIZ ASCENCIO**
**Director de tesis**

**DR. RAÚL PINTO ELÍAS**
**Revisor 1**

**DR. GERARDO REYES SALGADO**
**Revisor 2**

EDUCACIÓN | TECNOLÓGICO NACIONAL DE MÉXICO
22 SEP 2022
CENTRO NACIONAL DE INVESTIGACIÓN Y DESARROLLO TECNOLÓGICO
SERVICIOS ESCOLARES
RECIBIDO

C.c.p. Depto. Servicios Escolares.
Expediente / Estudiante
JGGS/ibm

Cuernavaca, Mor., 18/octubre/2022
No. De Oficio: SAC/153/2022
Asunto: Autorización de impresión de tesis

**KATIA RUBIT BENÍTEZ CASTRO**
**CANDIDATO(A) AL GRADO DE MAESTRO(A) EN CIENCIAS DE LA COMPUTACIÓN**
**P R E S E N T E**

Por este conducto, tengo el agrado de comunicarle que el Comité Tutorial asignado a su trabajo de tesis titulado **"MAPEO VISUAL DE INTERIORES CON ROBOT MÓVIL"**, ha informado a esta Subdirección Académica, que están de acuerdo con el trabajo presentado. Por lo anterior, se le autoriza a que proceda con la impresión definitiva de su trabajo de tesis.

Esperando que el logro del mismo sea acorde con sus aspiraciones profesionales, reciba un cordial saludo.

A T E N T A M E N T E
**Excelencia en Educación Tecnológica®**
"Educación Tecnológica al Servicio de México"

**DR. CARLOS MANUEL ASTORGA ZARAGOZA**
**SUBDIRECTOR ACADÉMICO**

EDUCACIÓN | TECNOLÓGICO NACIONAL DE MÉXICO
CENTRO NACIONAL DE INVESTIGACIÓN Y DESARROLLO TECNOLÓGICO
SUBDIRECCIÓN ACADÉMICA

C. c. p. Departamento de Ciencias Computacionales
Departamento de Servicios Escolares

CMAZ/CHG

EDUCACIÓN | TECNOLÓGICO NACIONAL DE MÉXICO
20 OCT 2022
CENTRO NACIONAL DE INVESTIGACIÓN Y DESARROLLO TECNOLÓGICO
SERVICIOS ESCOLARES
RECIBIDO
EBN

# Resumen

El paradigma dominante para dar a un robot móvil autonomía es la localización y mapeo simultáneo, conocido como SLAM, por sus siglas en inglés. En este trabajo se aumenta las capacidades de un sistema SLAM mediante un módulo con la capacidad de reconocer objetos y lugares de los que se dispone de imágenes. El sistema reporta sus hallazgos mediante anotaciones sobre el mapa producido en la exploración.

El método de reconocimiento es mediante un módulo de asociación, como el que es parte de un sistema SLAM, que actúa en paralelo con éste. Se analizan las mismas vistas tomadas por el robot, pero con detección, descripción y reconocimiento (o recuperación) propias. Se emplea el concepto de bolsa de palabras para describir toda una imagen, y el "vocabulario" empleado por el sistema reconocedor es propio, aunque similar al usado por el módulo de asociación de SLAM. La función principal del módulo de asociación es detectar ciclos en la trayectoria de exploración, por lo cual tiene que registrar cada vez más vistas de esa trayectoria, según pasa el tiempo. El módulo de reconocimiento trae descripciones mediante bolsas de palabras del conjunto fijo de objetos y lugares a reconocer. Por tanto, su tiempo de ejecución es pequeño y constante.

El sistema reconocedor es probado mediante tres recorridos: un recorrido exterior del conjunto KITTI, un recorrido interior del conjunto TUM y un recorrido interior del CENIDET. Como es un sistema nuevo, sólo se pueden establecer comparaciones cualitativas con unos pocos sistemas parecidos en objetivos.

El sistema funciona en tiempo real a 30 FPS, y entre las aportaciones está una modificación de ORB-SLAM2 para emplear la última versión de las librerías OpenCV, Pangolin y Eigen3.

# Abstract

The dominant paradigm for endowing a mobile robot with autonomy is simultaneous mapping and localization, known as SLAM. In this work, the capabilities of a SLAM system are increased through the ability to recognize objects and places from which images are available. The system thus augmented reports its findings through annotations on the map made in the exploration.

The recognition method is through an association module such as that which is part of a SLAM system, which acts in parallel. The same views taken by the robot are analyzed, but with its own detection, description and recognition (or recovery). The concept of a bag of words is used to describe a complete image, and the "vocabulary" used by the recognizer is its own, although similar to that used by the SLAM association module. The main function of the association module is to detect cycles in the robot's path, so it has to record more and more views of that path as time goes by. The recognition module brings descriptions through bags of words of the fixed set of objects and places to be recognized. Therefore, its execution time is small and constant.

The recognition system is tested by means of three routes: an outdoors route of the KITTI set, an indoors route of the TUM set and route of the CENIDET. As it is a new system, only qualitative comparisons of objectives can be made with a few similar systems.

The system works in real time at 30 FPS, and among the contributions is a modification of ORB-SLAM2 to use the latest version of the OpenCV, Pangolin and Eigen3 libraries.

# Dedicatoria

**A mi hija,** aunque aún no lo sepas eres y serás lo más importante en mi vida, hoy he dado un paso más para servir de ejemplo a la persona que más amo en este mundo.

**A mi esposo,** por estar conmigo en todo momento aun cuando el estudio y el trabajo ocuparon mi tiempo y esfuerzo.

**Y a mis padres** que también me apoyaron a cuidar de mi hija para que yo pudiera seguir estudiando.

# Agradecimiento

Agradezco al Consejo Nacional de Ciencia y Tecnología (CONACYT) y al Centro Nacional de Investigación y Desarrollo Tecnológico (CENIDET) por permitirme y apoyarme para realizar mis estudios de maestría.

Un agradecimiento a mi director el Dr. José Ruiz Ascencio, por su apoyo, consejos y tiempo para motivarme en la realización de este proyecto, también o agradezco a mis revisores el Dr. Raúl Pinto Elías, el Dr. Manuel Mejía Lavalle Q.E.P.D. y el Dr. Gerardo Reyes Salgado por sus enseñanzas, consejos y críticas. Así mismo doy las gracias a los profesores y personal del CENIDET que contribuyeron a mi formación como Maestro en Ciencias.

A mis compañeros y amigos que me apoyaron en todo momento y compartieron muchas de sus experiencias conmigo que me inspiraron en el desarrollo del proyecto.

# Índice

Capítulo 1 Introducción ..... 11
1.1 Planteamiento del Problema ..... 13
1.2 Objetivo general ..... 13
1.3 Organización de la tesis ..... 14
Capítulo 2 Estado del arte ..... 15
2.1 Antecedentes institucionales ..... 15
2.1.1 Navegación Visual en Trayectorias Cerradas ..... 15
2.1.2 Navegación, localización y mapeo de robots móviles para trayectorias pre especificadas por imágenes ..... 16
2.1.3 Odometría mediante visión artificial usando métodos directos ..... 16
2.1.4 Sistema de navegación inercial asistido por visión para robots móviles terrestres ..... 17
2.1.5 Evaluación de las técnicas SLAM disponibles en ROS ..... 17
2.1.6 Discusión ..... 19
2.2 Estado del Arte ..... 19
2.2.1 Discusión ..... 20
Capítulo 3 Metodología ..... 22
3.1 Detectores y descriptores de puntos característicos ..... 22
3.1.1 SURF ..... 22
3.1.2 ORB ..... 23
3.1.3 Tiempos ..... 24
3.1.4 Conclusiones de detectores y descriptores de puntos característicos ..... 24
3.2 Bag of Visual Words (BoVW) ..... 25
3.2.1 Creación del vocabulario ..... 26
3.2.2 Base de datos ..... 28
3.2.3 Medida de similitud ..... 29
3.3 Bancos de imágenes ..... 29
3.4 ORB-SLAM2 ..... 31
3.4.1 Instalación de ORB-SLAM2 ..... 33
3.4.2 Prueba de funcionamiento ..... 35

3.5 Módulo de reconocimiento propuesto ..... 36
3.5.1 Creación de BoVW ..... 38
Capítulo 4 Pruebas y resultados ..... 40
4.1 Prueba para definir parámetros de la base de datos. ..... 40
4.2 Pruebas para definir el valor de la variable del umbral (α): ..... 41
4.3 Prueba de análisis de parametrización ..... 43
4.4 Prueba del módulo de reconocimiento. ..... 46
4.5 Análisis de resultados ..... 50
4.6 Comparativa con estado del arte ..... 52
Capítulo 5 Conclusiones ..... 53
5.1 Objetivos específicos ..... 53
5.2 Alcances ..... 54
5.3 Aportaciones ..... 54
5.4 Conclusiones generales ..... 55
5.5 Trabajo a futuro ..... 55
5.6 Productos académicos adicionales ..... 56
REFERENCIAS ..... 58

# Índice de Figuras

Figura 1.- Esquema de referencia para el módulo propuesto. ..... 14
Figura 2.- Resultados del experimento de asociación [9]. ..... 15
Figura 3 .- Módulos del sistema: visual, inercial y fusión de datos [12]..... 17
Figura 4.- Detección de puntos destacados con SURF en C++ ..... 23
Figura 5.- Detección de puntos destacados con ORB en C++..... 24
Figura 6.- Ejemplo de BOVW[]..... 26
Figura 7.- a) Dibujado de los puntos característicos descritos en el plano. b) Primera aplicación de K-means [40] ..... 27
Figura 8.- Aplicación de iteraciones con K-means [38]. ..... 27
Figura 9.- Índice inverso [8]. ..... 28
Figura 10.- Índice directo [8]. ..... 29
Figura 11.- KITTI Odometría visual secuencia 02..... 30
Figura 12.- TUM secuencia freiburg3_xyz. ..... 30
Figura 13.- CENIDET. ..... 31
Figura 14.- Esquema de los módulos de ORB-SLAM2 [25]. ..... 32
Figura 15.- Prueba de ORB-SLAM2 con dataset KITTI secuencia 02 ..... 35
Figura 16.- Prueba de ORB-SLAM2 con dataset KITTI secuencia 02. ..... 35
Figura 17.- Diagrama de flujo del módulo realizado ..... 37
Figura 18.- Selección de la meta. ..... 38
Figura 19.- Características extraídas. ..... 39
Figura 20.- Creación del vocabulario. ..... 39
Figura 21.- Objetos metas para el dataset TUM. ..... 47
Figura 22.- Objetos metas para el dataset CENIDET. ..... 47
Figura 23.- Objetos metas para el dataset KITTI. ..... 47
Figura 24.- Muestra de la detección de objetos durante el recorrido de TUM. ..... 48
Figura 25.- Muestra de la detección de objetos durante el recorrido del CENIDET. ..... 49
Figura 26.- Muestra de la detección de objetos durante el recorrido de KITTI. ..... 49
Figura 27.- Imagen con poca textura..... 51
Figura 28.- Imágenes pequeñas..... 51

# Índice de Tablas

Tabla 1.- Comparación trabajos relacionados ..... 21
Tabla 2.- Tiempos y puntos detectados ORB y SURF ..... 24
Tabla 3.- Comparativa al crear base de datos con diferente número de características. ... 41
Tabla 4.- Evaluación de las detecciones utilizando una base de datos con 20 metas. ..... 45
Tabla 5.- Evaluación de las detecciones utilizando una base de datos con 10 metas. ..... 46
Tabla 6.- Comparativa bajo la métrica de precisión de los diferentes recorridos ..... 51
Tabla 7.- Mediana del tiempo en milisegundos por cuadro de ORB-SLAM2 ..... 52
Tabla 8.- Objetivos específicos logrados ..... 53
Tabla 9.- Alcances cumplidos ..... 54

# Índice de Ecuaciones

Gráfica 1.- Comparativa general del desempeño de los algoritmos en los distintos entornos[13]. ..... 18
Gráfica 2.- Valores obtenidos del reconocimiento con BoVW. ..... 42
Gráfica 3.- Detecciones de TUM utilizando una base de datos con 20 metas. ..... 43
Gráfica 4.- Deteccionesde CENIDET utilizando una base de datos con 20 metas. ..... 44
Gráfica 5.- Detecciones de TUM utilizando una base de datos con 10 metas. ..... 44
Gráfica 6.- Detecciones de CENIDET utilizando una base de datos con 10 metas. ..... 45
Gráfica 7.- Precisión del módulo propuesto. ..... 50

# Índice de Ecuaciones

Ecuación 1 ..... 27
Ecuación 2 ..... 28
Ecuación 3 ..... 29
Ecuación 4 ..... 37

# Acrónimos.

- **BA** Bundle Adjustment.
- **BoVW** Bag of Visual Words.
- **BoW** Bag of Words.
- **DoF** Degrees of Freedom.
- **EKF** Extended Kalman Filter.
- **GPS** Global Positioning System.
- **ICP** Iterative Closest Point.
- **IDF** Inverse Document frequency.
- **IMU** Inertial Measurement Unit.
- **INS** Inertial Navigation System.
- **ORB** Oriented FAST and Rotated BRIEF.
- **RGB** Red Green Blue
- **RGB-D** Red Green Blue Depth Sensor
- **SIFT** Scale-invariant feature transform.
- **SLAM** Simultaneous localization and mapping.
- **SURF** Speeded-Up Robust Features.
- **VSLAM** Visual SLAM

# Capítulo 1 Introducción

La navegación autónoma ha tomado gran importancia a través de los años, por su utilidad en áreas como la agricultura [1], exploración espacial [2], atención al cliente [3] y exploración en zonas de riesgo [4]. La navegación de los robots móviles es un reto en el campo de la robótica, y es aún más problemático su mapeo (representación gráfica del entorno en el que se encuentra el robot), el cual tiene probablemente más de tres décadas en el campo de investigación. En los primeros trabajos que se realizaron, al robot se le proporcionaba un mapa prefabricado. Bajo este enfoque, no fue posible crear un robot autónomo para realizar tareas útiles, como ayudar a los humanos en entornos hostiles como en la construcción, la minería, la gestión de residuos tóxicos o simplemente tareas de servicios cotidianos como barrer, desplazar objetos, entre otros. Para cumplir con estas tareas, el robot debería poder crear el mapa por sí mismo [5].

Para generar automáticamente el mapa, un robot autónomo necesita primero obtener información del ambiente que lo rodea, esto con ayuda de sensores. Se han desarrollado muchos algoritmos que permiten reaccionar al contenido de una escena registrada, uno de ellos es la detección de *puntos destacados*, que permiten mediante un rápido examen pixel por pixel de una imagen, detectar puntos que sobresalen por su alto contraste o por ser parte de un rasgo de alta curvatura. Una vez obtenidos los puntos destacados es necesario generar una *descripción* del punto, a partir de los píxeles circunvecinos que le aportan información. Las descripciones de los puntos destacados de la imagen A se pueden comparar con las descripciones de los puntos destacados de la imagen B, a encontrar descripciones iguales se le llama *correspondencia*.

Para determinar la posición y orientación del robot sin ningún conocimiento previo del entorno, es necesario utilizar el proceso de la *odometría*, mediante el análisis de una secuencia de imágenes adquiridas. La odometría nos dará la forma y tamaño de la trayectoria del robot. Otro punto importante es la localización del robot, es decir, determinar dónde se encuentra con respecto a un plano. Como se desconoce la ubicación del robot, su *localización* se obtiene en relación con los puntos de interés. Entonces nos

enfrentamos al problema de que la odometría y la localización del robot autónomo deben resolverse al mismo tiempo. De estos problemas se originó la *Localización y Mapeo Simultáneo* (SLAM), del cual se desarrollaron diferentes métodos con el uso de diferentes sensores. Estos pueden ser sonares, mecánicos, láser, inerciales, etc. Cuando se agregan sensores de visión, a esta combinación se le llama *Visual SLAM* (VSLAM) [6]. Las técnicas de SLAM crean un mapa en un ambiente desconocido y se localiza dentro de ese mapa en tiempo real [7]. La realización de este proceso involucra tres tareas importantes: la localización del robot, la construcción del mapa del entorno y la navegación del robot en el entorno.

En el sistema SLAM se encuentra un proceso importante llamado *cierre de ciclos* que ayuda a detectar si el robot está en un área previamente visitada y corregir el error acumulado durante la trayectoria. En el sistema SLAM se encuentra un proceso importante llamado cierre de ciclos que ayuda a detectar si el robot está en un área previamente visitada y corregir el error acumulado durante la trayectoria. Es una técnica que utiliza un vocabulario visual para convertir una imagen en un vector numérico, lo que permite gestionar grandes conjuntos de imágenes [8].

En la navegación autónoma se requiere generar mapas 3D o 2D, para que el robot sea capaz de localizar y construir una trayectoria, pero estos procesos pueden generar un alto costo computacional por lo cual esto afectaría también al tiempo en que se procesa toda la información. El SLAM ha sido un problema intensamente investigado y abordado por muchas metodologías novedosas, ya que para realizar un mapa preciso se requiere una localización precisa y viceversa. Por otro lado, el reconocimiento de lugares en la navegación autónoma ha sido enfocado principalmente en el cierre de ciclos, para detectar si el robot móvil ha regresado a un área previamente visitada y corregir el error acumulado durante la trayectoria.

## 1.1 Planteamiento del Problema

Dado a que los robots autónomos son útiles en tareas de exploración, es muy valioso poder utilizar su capacidad de reconocimiento de objetos y lugares para estos fines. Sin embargo, los algoritmos de asociación requieren consultar frecuentemente con la base de datos para compararla con la escena actual, por lo cual se requiere utilizar algoritmos de asociación que no afecten el rendimiento del sistema SLAM. Existen pocos trabajos donde los sistemas SLAM realicen búsqueda de objetos y/u lugares como principal enfoque. Por esto, el desarrollo de esta tesis propone desarrollar un módulo de reconocimiento que pueda ser implementado en sistemas SLAM, donde logre ser de ayuda para búsqueda en ambientes hostiles como en zonas de riesgo, la construcción, la minería, etc. Para realizar esto, el método de las BoVW es una de las mejores opciones debido a que la idea general es representar una imagen como un conjunto de características las cuales consisten en puntos clave y descriptores. Por lo que no es necesario utilizar una gran cantidad de imágenes para crear el vocabulario, además el tiempo que necesita para realizar la asociación es muy bajo, por lo que puede ofrecer muy buenos resultados.

## 1.2 Objetivo general

Se puede enunciar como objetivo general el implementar un algoritmo explorador que use visión como sistema sensorial, que sea capaz de reconocer lugares u objetos, mostrando su ubicación en un mapa.

El módulo de reconocimiento propuesto en este trabajo de tesis no modificó en gran medida al sistema SLAM que se implementó, como se muestra en la Figura 1, el procesamiento del módulo se realiza fuera del sistema por lo que solo se utilizó el cuadro obtenido de la cámara para realizar la consulta y después devolver el nombre de la meta encontrada.

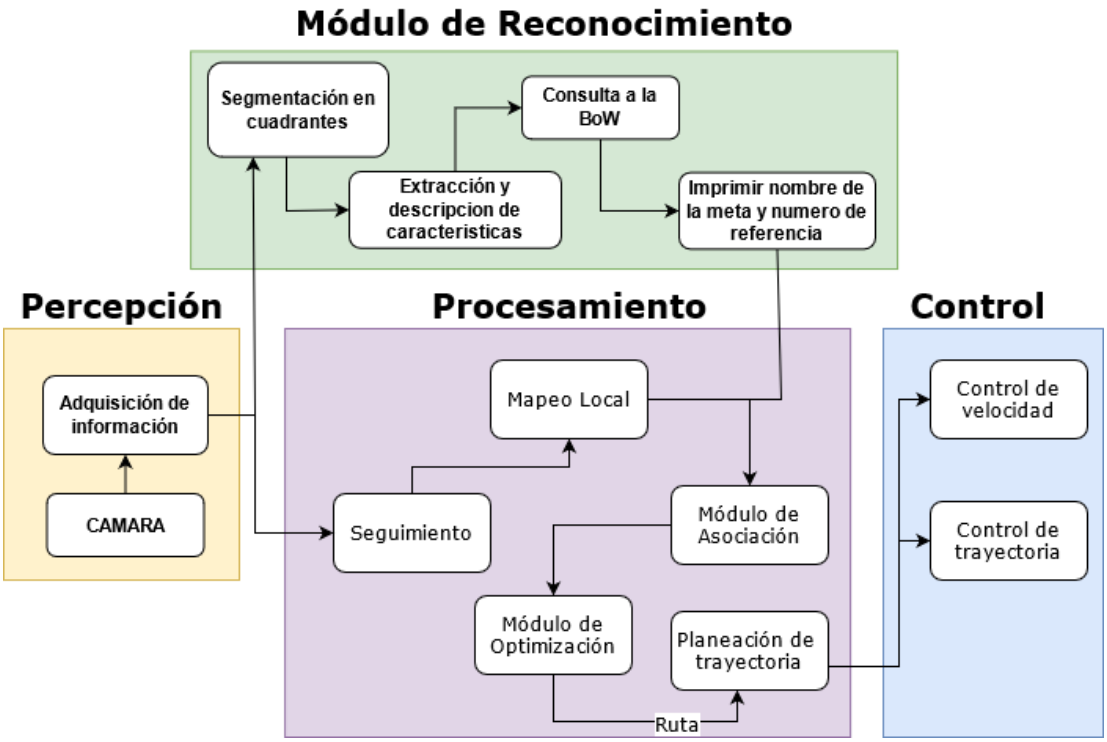



*Figura 1.- Esquema de referencia para el módulo propuesto.*

El sistema se ejecutó en tiempo real en función de la velocidad de captura de las cámaras con las que se cuentan y la velocidad de desplazamiento del robot, además no se trabaja con un entorno dinámico, por lo que los objetos en las escenas son estáticos. La velocidad de procesamiento se limitó por las características del CPU (i7 8750H, 16gb de memoria RAM, tarjeta de video NVIDIA GTX 1050 Ti).

## 1.3 Organización de la tesis

Con el fin de orientar al lector acerca del contenido de la presente tesis, se enumeran y describen brevemente las partes constitutivas: el Capítulo 2 contiene el estudio del estado del arte; el Capítulo 3 contiene la metodología de solución, detalla la implementación del módulo de reconocimiento propuesto; el Capítulo 4 contiene distintos experimentos, el módulo propuesto implementado en ORB-SLAM2 y los resultados. Finalmente, las conclusiones obtenidas, los objetivos y las aportaciones son mencionados en el Capítulo 5.

# Capítulo 2 Estado del arte

## 2.1 Antecedentes institucionales

En el Centro Nacional de Investigación y Desarrollo Tecnológico (**CENIDET**) se han desarrollado algunos proyectos de investigación relacionados con el tema de este proyecto de tesis, a continuación, se encuentra una breve descripción.

### 2.1.1 Navegación Visual en Trayectorias Cerradas

Trabajo realizado por MC. Karla Margarita Peñaloza Membrila[9], presentó un sistema de navegación visual para generar instrucciones de control para un robot móvil en la navegación autónoma de una determinada trayectoria cerrada y sin obstáculos. El sistema se centró en la detección, descripción y correspondencia de puntos destacados obteniendo la ruta por imágenes usando un algoritmo de correspondencia basado en el seguimiento de puntos destacados. Esto fue realizado usando el descriptor ORB en la librería Opencv. Para la asociación de la escena utilizó un árbol de búsqueda tipo *kdtree*. En sus resultados demostró que las pruebas (Figura 2) realizadas proponen al detectór-descriptor ORB como el mejor algoritmo para detectar y describir puntos destacados de manera rápida y precisa, haciendo al algoritmo *kd-tree* como el mejor algoritmo de asociación de imágenes.

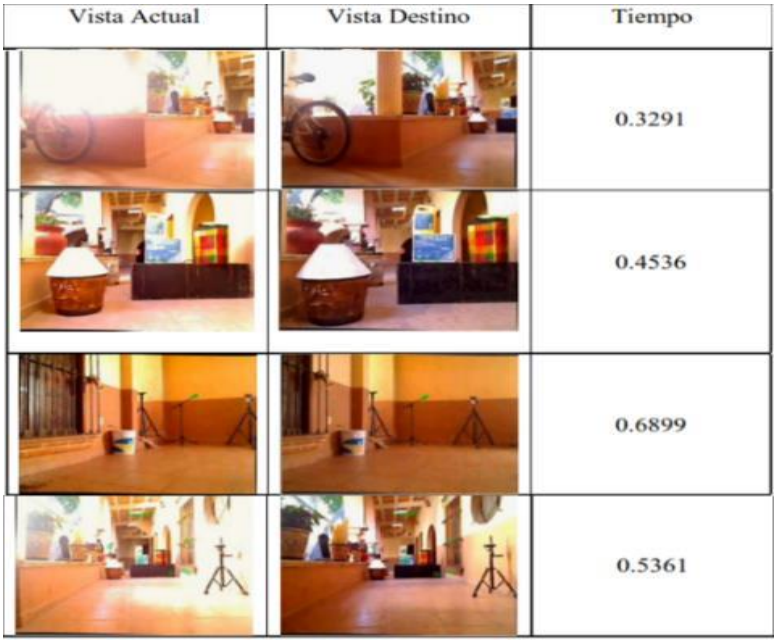



*Figura 2.- Resultados del experimento de asociación* [9].

### 2.1.2 Navegación, localización y mapeo de robots móviles para trayectorias pre especificadas por imágenes

El MC. Andrés Vergara Bahena[10], presentó un sistema de visión artificial que permite a un robot móvil realizar un recorrido autónomo sobre una trayectoria definida por una secuencia de imágenes extraídas de un video capturado lo que le permite generar un mapa de trayectorias con la librería *LIBVISO2*. Utilizó la librería *OpenRobotino* para el robot FESTO y *OpenCV* para procesar las imágenes. Mejoró el algoritmo para la estimación de la orientación empleado en [9]debido que en las pruebas usando dicha metodología mostró que en cada iteración los ángulos eran inconsistentes debido a que la orientación variaba demasiado al grado de estimar ángulos en sentido contrario. En los resultados determinó que para mejorar la precisión en la estimación de la orientación es mejor tomar los puntos que se encuentran al centro de cada conjunto en lugar de los extremos y a partir de ellos trazar las líneas rectas hacia el centro. Utilizó tres métodos de selección de vistas claves y desarrolló uno basado en el cálculo de la homografía entre dos vistas, fueron comparadas y optó por el método de cálculo de la homografía ya que mostró buenos resultados al probar los conjuntos generados con este método.

### 2.1.3 Odometría mediante visión artificial usando métodos directos

El proyecto fue desarrollado por MC. Charles Fernando Velázquez Dodge [11], quien presentó una investigación de los métodos directos y semidirectos para la odometría visual para desarrollar un algoritmo basado en estos métodos, que funcionara en tiempo real. Se utilizó una IMU para reducir el costo computacional en conjunto con transformaciones simples, pero estas aún tenían un costo elevado y se dedujo que no es posible trabajar en tiempo real a la perfección. Con la ayuda de este algoritmo se obtuvo información de localización de un robot móvil y mapas de sus trayectorias recorridas. Una desventaja que se podría considerar es que el algoritmo desarrollado tiene la cámara siempre viendo para el piso lo que hace imposible detectar obstáculos frente del robot, una mejora que propone el autor sería que la cámara detecte rotaciones junto con la IMU para mejorar la precisión.

### 2.1.4 Sistema de navegación inercial asistido por visión para robots móviles terrestres

El proyecto realizado por MC. Larisa Navarrete Olmedo[12], presentó un sistema de navegación inercial asistido por visión para robots móviles, que reduce el error con la asistencia de datos visuales. El sistema fija las velocidades lineales y angulares en cero mientras la visión detecta que el prototipo se encuentra en reposo y ejecuta las ecuaciones de navegación mientras el prototipo se encuentra en movimiento. En la Figura 3 se muestran los 3 módulos con los que cuenta el sistema.

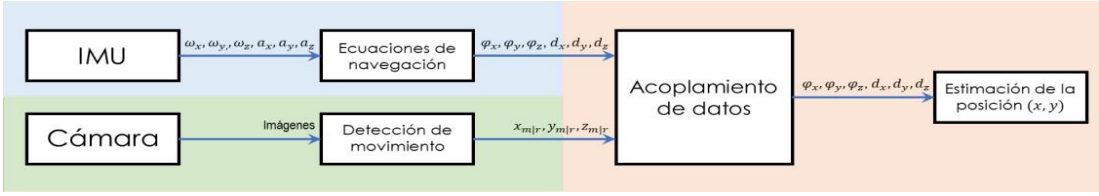


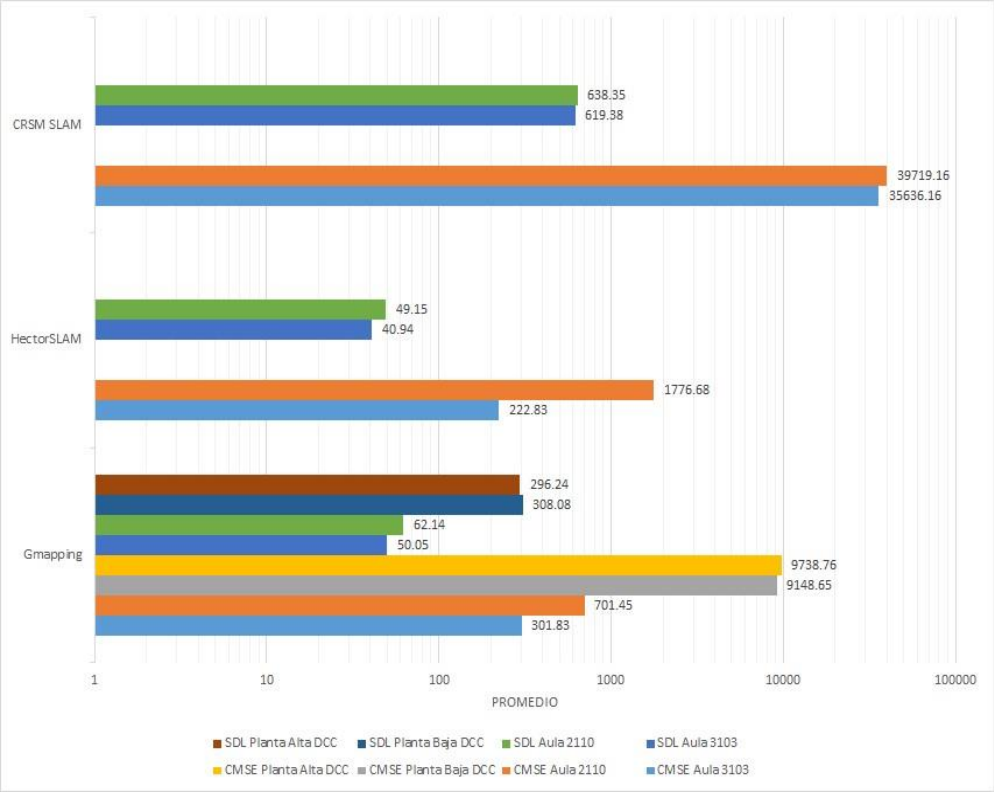

*Figura 3 .- Módulos del sistema: visual, inercial y fusión de datos* [12].

Sus aportaciones fueron el desarrollo de una heurística que permite controlar las integraciones por componentes ($\boldsymbol{x}$, $\boldsymbol{y}$) del sistema de navegación inercial (INS) mediante la detección del movimiento utilizando la información del flujo óptico. Construyó un prototipo integrado que consta principalmente de una IMU y una cámara, el cual es de tamaño pequeño y tiene un peso menor a los 0.50 kg, implementó una mejora al método de detección del estado del vehículo, dicho método utiliza toda la imagen para determinar si el vehículo se encontraba en reposo o movimiento. La mejora consistió en utilizar solamente 5/9 de la imagen y poder detectar las direcciones (adelante, atrás, derecha e izquierda) del movimiento sobre los ejes $\boldsymbol{x}$ y $\boldsymbol{y}$ además de la detección del reposo. Y también desarrolló un sistema de navegación que mediante el acoplamiento de datos inerciales con visuales reduce el error acumulado en las mediciones del INS.

### 2.1.5 Evaluación de las técnicas SLAM disponibles en ROS

El proyecto "***Evaluación de las técnicas SLAM disponibles en ROS***" fue realizado por MC. Diana Elizabeth López Borreguero[13] presentó un estudio comparativo de las metodologías que dispone ROS sobre localización, mapeo y navegación. Los algoritmos que se utilizaron son Gmapping, HectorSLAM y CRSM SLAM, los cuales permiten realizar la localización del robot y el mapeo del entorno creando mapas de ocupación de celdas en 2D

con el uso de sensores láser, mientras que para la navegación se utilizó el paquete *Frontier Exploration* cual se basa en el concepto de fronteras, que son las regiones que se encuentran en el límite del espacio abierto y del espacio inexplorado y la tele operación. Realizó pruebas utilizando los algoritmos Gmapping, HectorSLAM y CRSM por medio de un robot con configuración diferencial y un sensor RGB-D Kinect. Las pruebas anteriores, demostraron que es posible generar mapas de entornos en 2D de tres formas distintas: a) simulación de los entornos, b) tiempo real y c) a partir de un archivo de datos dando como resultado que Gmapping fue el algoritmo con el mejor desempeño, ya que pudo generar mapas aproximados de todos los entornos y mostró su capacidad de recuperar la pose del robot en los entornos en la mayoría de las veces, lo cual no pudieron realizar HectorSLAM y CRSM SLAM (Gráfica 1).




*Gráfica 1.- Comparativa general del desempeño de los algoritmos en los distintos entornos*[13].

### 2.1.6 Discusión

Después de haber realizado un estudio sobre los trabajos realizados dentro del CENIDET, podemos notar que existen trabajos que establecen asociación entre descripciones de puntos de la escena y descripciones de puntos de imágenes previamente registradas, en los antecedentes no es necesario buscar el lugar en la base de datos, pues ya se sabe cuál es la siguiente escena. Por otro lado, en el CENIDET no se ha realizado un trabajo que realice localización de objetos y lugares con anotación dentro de un mapa en tiempo real dentro de un sistema SLAM.

## 2.2 Estado del Arte

Los humanos tenemos la capacidad de reconocer múltiples objetos en una imagen o escenario, aún si el objeto tiene diferencias de rotación, tamaño, escala o posición. Aunque podemos realizar esta tarea sin esfuerzo, en la visión artificial esta tarea es un desafío, debido a que se necesita superar o minimizar varios problemas como la iluminación, escala, textura o características [14].

En la navegación autónoma, el reconocimiento se ha utilizado en sistemas SLAM enfocado principalmente en el cierre de ciclos, esto para detectar si el robot móvil ha regresado a un área previamente visitada y corregir el error acumulado durante la trayectoria [15]–[17]. Uno de los métodos más utilizados en la cerradura de ciclos es la bolsa de palabras visuales (BoVW) [18]. Esta técnica usa una discretización del espacio de descripción empleado por algún método de detección de puntos destacados que aparezcan en una imagen de la escena. Esta discretización de celdas irregulares que llamamos palabras visuales hace que el espacio de descripción tenga dimensiones manejables, y no ocupe espacio para descripciones que no aparecen en la escena, esto permite gestionar grandes conjuntos de imágenes. Estas características pueden ser extraídas utilizando descriptores como SIFT[19], [20], SURF [21] u ORB [22]. Un método de BoVW fue presentado por D. Gálvez-López y J. D. Tardós llamado DBoW2, en el cual se utilizó por primera vez las bolsas de palabras binarias obtenidas de los descriptores BRIEF[23] y el detector de características FAST[24], reduciendo el tiempo de las extracciones de características. Este método se ha utilizado en

varios sistemas SLAM para el cierre de ciclos [7], [16], [25],[26] quienes reportan buenos resultados, ya que dicha librería no necesita un alto costo computacional al ser implementada.

En algunos sistemas SLAM se desarrollaron sistemas de reconocimiento como “SLAM with Object Discovery, Modeling and Mapping” [27]y “Detect-SLAM: Making Object Detection and SLAM Mutually Beneficial“[28].Dichos trabajos utilizan los objetos descubiertos como puntos de referencia para ayudar a localizar el robot y en la detección de cierre de ciclos mediante la reconstrucción de los objetos. Otros trabajos también realizaron detección de objetos pero a diferencia de los anteriores, los objetos detectados fueron colocados en el mapa generado por el sistema SLAM como Pillai y Leonard[29] presentaron un sistema SLAM-AWARE utilizando visión monocular enfocado al reconocimiento de objetos basado en ORB-SLAM[7], utilizando mapas semi-densos. Para el reconocimiento utilizaron los objetos de diferentes vistas en un entorno 3D reconstruido similar a LSD-SLAM[30] y utilizaron el método de las BoVW, llamado BoVW + FLAIR para la detección y etiquetado de los objetos. El BoVW+FLAIR les permitió escalar el sistema a una gran cantidad de categorías de objetos con un tiempo de ejecución casi constante. Otro trabajo llamado “Integrating SLAM and object detection for service robot tasks” [31] realizó detección de objetos utilizando histogramas de color de coocurrencia para resolver problemas con fondo complejo, iluminación variable y oclusión de objetos.

### 2.2.1 Discusión

En la literatura se encontraron trabajos que realizan detección de objetos o lugares, algunos realizando un trabajo similar al propuesto, pero con diferentes métodos o propósitos como mejorar la precisión de mapa o reconstrucción del objeto en el entorno. De los trabajos encontrados en la literatura los más relevantes para esta tesis se muestran en la Tabla 1, comparando los métodos utilizados.

*Tabla 1.- Comparación trabajos relacionados*

| Trabajo | Detector | Descriptor | Método de Asociación |
|---|---|---|---|
| **Real-Time Loop Detection with Bags of Binary Words**[32] | FAST | BRIEF | BoVW |
| **Monocular SLAM supported object recognition**[29] | SIFT | PCA+SIFT | BoVW+FLAIR |
| **ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras** [25] | ORB | ORB | BoVW |
| **Integrating SLAM and Object Detection for Service Robot Tasks**[31] | | histogramas | Receptive Field Cooccurrence Histograms (RFCH) |

# Capítulo 3 Metodología

En esta sección se presenta la metodología que se utilizó para la solución del problema.

## 3.1 Detectores y descriptores de puntos característicos

Existen muchos algoritmos para detectar y describir puntos característicos en una imagen, esto para facilitar el uso de la información en una imagen, en la literatura señala que los descriptores más usados para las BoVW son: ORB, SIFT y SURF. Pero para trabajar en tiempo real, los más recomendados son ORB y SURF debido a su rapidez por lo que en esta sección analizaremos cual es el más factible para este proyecto de tesis.

### 3.1.1 SURF

El algoritmo Speeded-Up Robust Features (SURF) fue propuesto por Tuytelaars(Bay, Tuytelaars, y Gool 2006) como una alternativa eficiente de SIFT, es un detectór y un descriptor de los puntos de interés de una imagen, donde se transforma la imagen en coordenadas, utilizando una técnica llamada multi-resolución. Consiste en hacer una réplica de la imagen original de forma Piramidal Gaussiana o Piramidal Laplaciana, y obtener imágenes del mismo tamaño, pero con el ancho de banda reducido. Esta técnica asegura que los puntos de interés son invariantes en el escalado. Un ejemplo del funcionamiento de SURF utilizando las librerías de OpenCV implementado con en el lenguaje de programación C++ como se puede observar en la Figura 4.

*Figura 4.- Detección de puntos destacados con SURF en C++*

### 3.1.2 ORB

Este algoritmo presenta una alternativa eficiente a SIFT o SURF presentado en “ORB: An efficient alternative to SIFT or SURF”. ORB presenta una combinación entre el algoritmo FAST para detectar puntos característicos y BRIEFF para la descripción, en ambos algoritmos se realizaron modificaciones para mejorar el rendimiento. FAST encuentra los puntos clave, luego aplica la medida de la esquina de Harris para encontrar los mejores N puntos entre ellos. Por otro lado, el descriptor BRIEFF funciona mal con la rotación, Por lo tanto, lo que ORB hace es “dirigir” según la orientación de los puntos clave. En la Figura 5 se puede observar una implementación de SURF utilizando las librerías de OpenCV implementadas con en el lenguaje de programación C++.

*Figura 5.- Detección de puntos destacados con ORB en C++*

### 3.1.3 Tiempos

Se realizaron las pruebas utilizando la librería OpenCV y ejecutados en el sistema operativo Windows, utilizando los tiempos y la cantidad de puntos característicos detectados como se muestra en la tabla 2.

*Tabla 2.- Tiempos y puntos detectados ORB y SURF*

| Detector y descriptor | Puntos característicos detectados | Tiempos en segundos |
|---|---|---|
| **ORB** | 500 | 0.034 |
| **SURF** | 1800 | 0.411 |

### 3.1.4 Conclusiones de detectores y descriptores de puntos característicos

Analizando los resultados, SURF fue el que detectó más puntos destacados con 1800, mientras que ORB detectó 500 puntos usando la misma imagen, pero se encontró una diferencia notable en el tiempo de ejecución ya que ORB fue el más rápido a comparación de SURF. Tomando en cuenta el propósito de este tema de tesis, SURF no es viable para

aplicaciones en tiempo real. Lo recomendado en la literatura para trabajar en tiempo real es ORB, a pesar de que extrae una menor cantidad de características a diferencia de SIFT o SURF, se enfoca en los puntos más característicos de la imagen como esquinas, textura o variaciones en contraste por los que se obtiene información importante de la imagen. Además, ORB es la mejor opción para realizar la asociación debido a su rapidez y bajo costo computacional. [7], [16], [25], [33]

## 3.2 Bag of Visual Words (BoVW)

Las bolsas de palabra visuales o Bag of Visual Words (BoVW) consisten en representar una imagen como un vector numérico cuantificando las características extraídas de la imagen (Figura 6), lo que permite administrar grandes conjuntos de imágenes [34]. Esta técnica implica una etapa que consiste en agrupar las descripciones de los puntos destacados de la imagen en un número fijo N de grupos. Esto se hace con un conjunto suficientemente rico de imágenes de entrenamiento, que pueden ser independientes de las imágenes de destino. Según Sirishaa et al. [35] el proceso de BOVW se divide en los siguientes pasos:

1. Se detectan automáticamente los puntos característicos de las imágenes.
2. Los descriptores locales se calculan sobre los puntos característicos detectados.
3. Cuantifican los descriptores locales en palabras visuales para formar el vocabulario visual.
4. Se encuentran las instancias / ocurrencias en las imágenes de entrada de cada palabra visual en el vocabulario para la creación de una bolsa de palabras visuales, que se presenta como un vector o un histograma de frecuencias de palabras visuales.

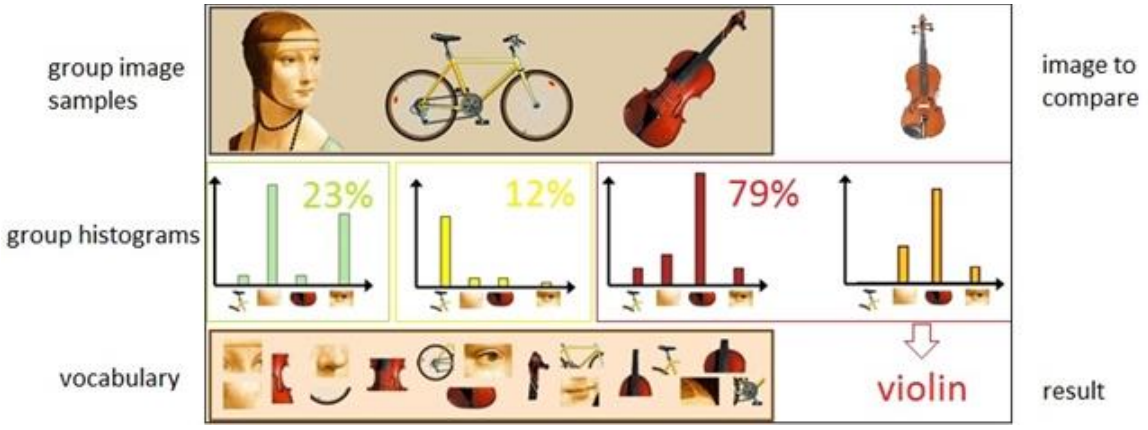



*Figura 6.*- Ejemplo de BOVW*[]*

### 3.2.1 Creación del vocabulario

La construcción del vocabulario se realiza mediante agrupación (clustering) esto se puede realizar con ayuda del algoritmo K-means [36], que consiste en encontrar grupos de puntos tal que se minimice la varianza intra-grupo, es decir, minimizar la suma de las distancias al cuadrado de cada punto al centro más cercano a él; o usar variaciones de este como el Approximate K-means [37], se aproxima a K-means utilizando 8 árboles k-d aleatorios, Hierarchical K-means (HKM)[38], el cual utiliza el vector resultante de las extracciones de características de todo el conjunto de entrenamiento y aplica el algoritmo K-means recursivamente y el algoritmo K-means++ [39], que aborda la deficiencia en la precisión encontrada mediante la especificación de un procedimiento para inicializar los centros de los conjuntos antes de proceder con las K-means iteraciones de optimización estándar.

El procedimiento para construir el vocabulario empieza desde que se obtiene la información de los descriptores. Imaginemos que los descriptores nos entregan un vector de 2 dimensiones (x, y) como se muestra en la Figura 7(a), donde representamos estos puntos, para después aplicar el algoritmo K-means y dividirlo en k regiones, con sus respectivos centroides Figura 7(b).

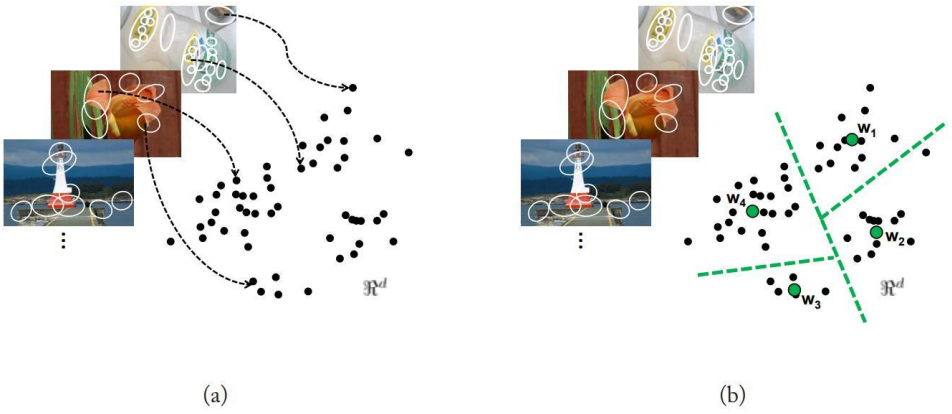



*Figura 7.- a) Dibujado de los puntos característicos descritos en el plano. b) Primera aplicación de K-means* [40] .

Este proceso se repite hasta L veces, por lo cual seguirá dividiéndose Figura 8. Una vez que termina la iteración los centroides que se encuentran en el último nivel se asignan como las palabras visuales.

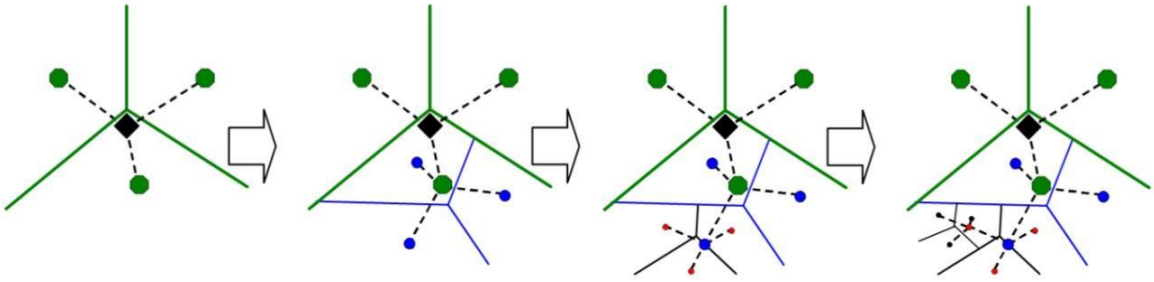


*Figura 8.- Aplicación de iteraciones con K-means* [38].

La cantidad de palabras visuales se puede calcular a través de la ecuación 1 [38]

$$\frac{K^{L+1}-K}{K-1} \approx K^{L}$$

*Ecuación 1*

Gálvez-López y Tardos [32] proponen una ramificación $(K)$ de 10 y una profundidad $(L)$ de 6 generando un total aproximado de $10^6$= 1000000 palabras visuales. Para la asignación de peso a cada palabra visual sería usar la forma "inverse document frequency o idf" que se representa con la ecuación 2.

$$w_i = log\left(\frac{N}{n_i}\right)$$

*Ecuación 2*

En donde $N$ representa la cantidad de imágenes de entrenamiento y $n_i$ el número de veces que aparece la palabra $w_i$ en dichas imágenes.

### 3.2.2 Base de datos

Para realizar la detección es necesario que, al momento de capturar una nueva imagen, sea posible obtener todas las imágenes que compartan un grado de similitud para esto se construye una base de datos. Para generar esta base de datos las palabras visuales son guardadas bajo índice invertido, el cual consiste en que cada palabra visual alberga la relación de la imagen o imágenes en que fue encontrada, esto con la finalidad que al realizar consultas tenga una mayor facilidad al encontrar coincidencias y reducir los tiempos de estas. Además, en la Figura 9 se puede observar un árbol de vocabulario de K= 3 y L=2, formando 9 palabras visuales en sus hojas.

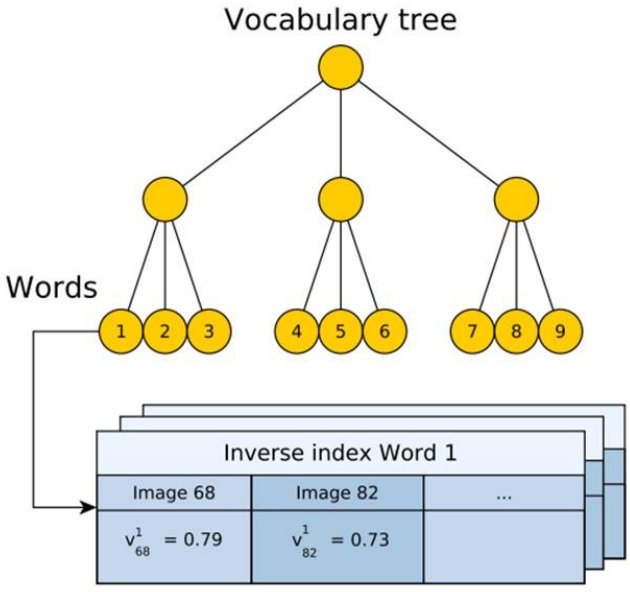



*Figura 9.- Índice inverso [8].*

Por otro lado, el índice directo guarda la relación de todas las palabras que fueron encontradas en cada imagen, como se muestra en la Figura 10.

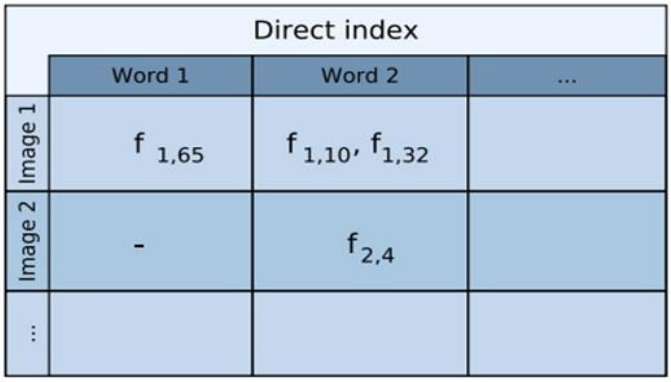



*Figura 10.- Índice directo [8].*

### 3.2.3 Medida de similitud

Esta medida se utiliza para obtener el vector de BoVW asociado a cada nueva imagen recibida. Este vector es comparado contra el vector asociado a otra imagen presente en la base de datos, que comparta alguna palabra en común. Para saber qué tanta similitud hay entre dos vectores de BoVW, el artículo "Scalable Recognition with a Vocabulary Tree" [38] hace mención que la norma L1 ( Ecuación 3) da mejores resultados.

$$S(V1, V2) = \frac{V1}{|V1|} - \frac{V2}{|V2|}$$

*Ecuación 3*

En donde, V1 y V2 son vectores de BoVW. Los que tengan una menor diferencia, son considerados posibles candidatos a las metas buscadas.

## 3.3 Bancos de imágenes

Para realizar la asociación de los objetos y/o lugares que se desean buscar se necesita realizar un árbol de vocabulario con imágenes de las metas desde diferentes vistas. Uno de los bancos de imágenes más citados en la literatura es KITTI [41], el cual cuenta con diferentes secuencias de recorridos en exteriores de zonas residenciales y carreteras,

además ORB-SLAM2 también lo utilizó para realizar pruebas del sistema SLAM. Se escogió la secuencia número 02 de “odometry_dataset”, la cual consta de 4,661 imágenes en escala de grises como se muestra en la Figura 11.

*Figura 11.- KITTI Odometría visual secuencia 02.*

Para las pruebas en interiores se utilizó el banco de imágenes de TUM[42], la secuencia “Freiburg3” la cual consta de 2,585 imágenes a color (Figura 12), Freiburg3 muestra una escena de oficina con dos escritorios separados por una mampara, grabados desde una cámara portada manualmente por una persona que camina alrededor del conjunto.

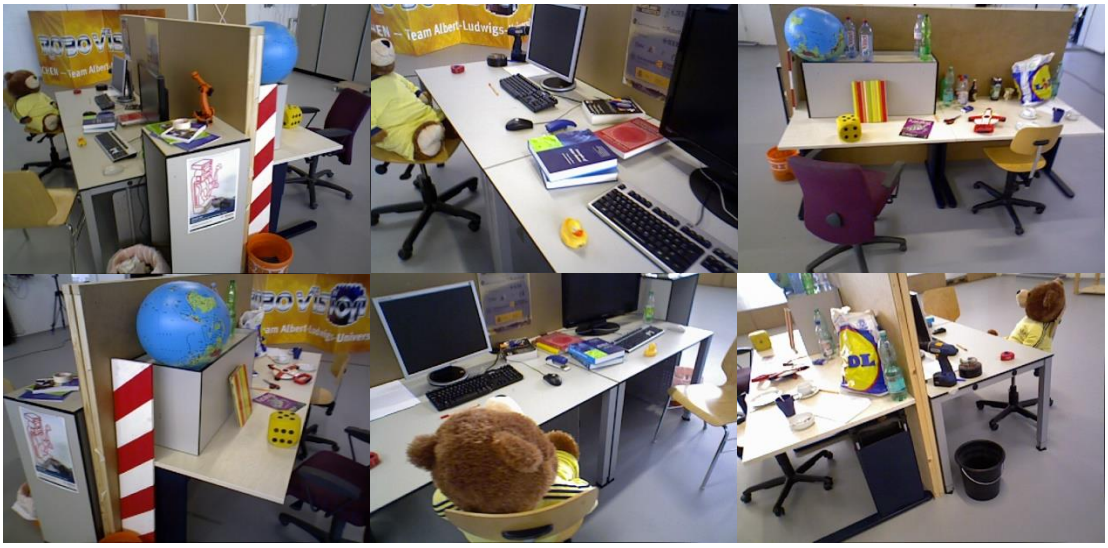


*Figura 12.- TUM secuencia freiburg3_xyz.*

Por otra parte, era necesario hacer unas pruebas en el CENIDET, por lo cual se realizó un recorrido grabando en la planta baja de la unidad Académica #3 de TecNM/CENIDET. Se

comenzó desde la entrada del edificio y se dio una vuelta al pasillo que se encuentra entre el laboratorio y el aula 3001, donde se colocaron diversos objetos estratégicamente, grabando desde una cámara de celular portada manualmente por una persona, Figura 13.

*Figura 13.- CENIDET.*

## 3.4 ORB-SLAM2

Es un sistema SLAM para cámaras monoculares, estéreo y RGB-D realizado por R. Mur-Artal y J. D. Tardós[25], basado en su trabajo anterior ORB-SLAM. El sistema cuenta con 3 procesos paralelos (Figura 14):

1. *El seguimiento* que ayuda a la localización de la cámara con cada fotograma, encontrando que las características coinciden con el mapa local y minimizando el error de reproyección aplicando el *bundle adjustment (BA),*
2. *el mapeo local* para gestionar el mapa local y optimizarlo realizando BA local, y
3. *la cerradura de ciclos* para detectar bucles grandes y corregir el error acumulado realizando una optimización de la pose.

El sistema cuenta con un cuarto hilo para realizar BA completo después de la optimización de gráfico de pose.

El sistema también cuenta con un módulo de reconocimiento de lugares basado en BoW para la reubicación en caso de falla de seguimiento o para el cierre de bucles. El módulo de cerradura de ciclos o *asociación* usa una base de datos (BD) de descripciones de las escenas

que va explorando. Las descripciones se codifican como bolsas de palabras visuales o BoVW. La función de la BD es tener una memoria compacta de todo lo visitado por el robot. Esta BD va creciendo según avanza la exploración, por lo cual el ciclo de ejecución va creciendo con el tiempo.

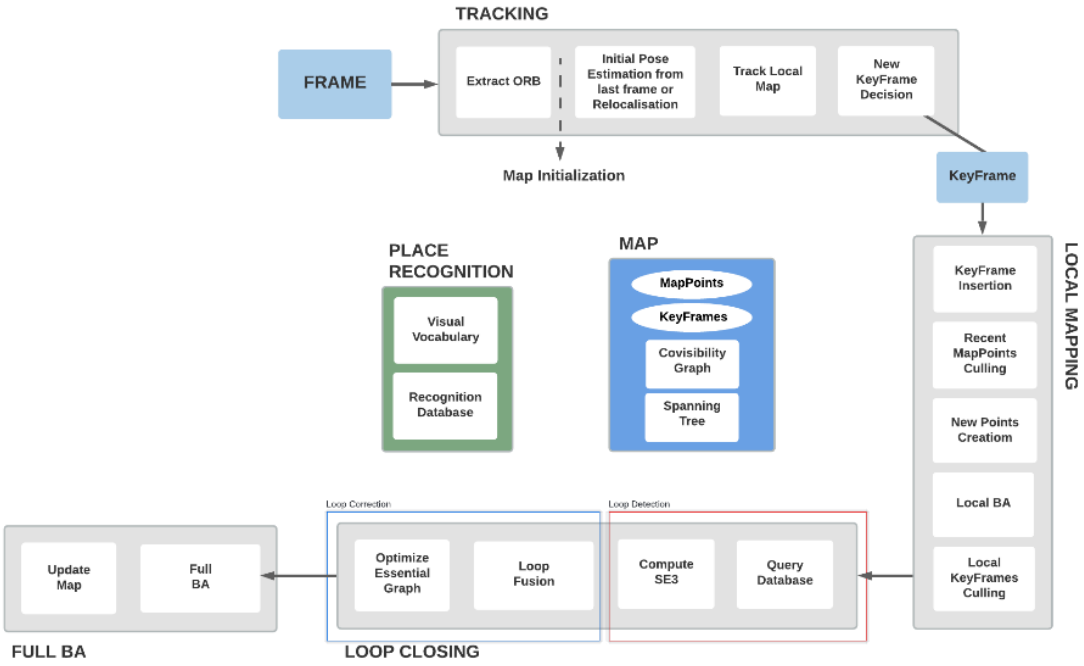



*Figura 14.- Esquema de los módulos de ORB-SLAM2* [25].

En su comparación con cuatro sistemas SLAM demostraron que ORB-SLAM2 logra en la mayoría de los casos la máxima precisión. Usando la base de datos KITTI, el banco de pruebas de odometría visual. ORB-SLAM2 actualmente se considera la mejor solución SLAM, en los resultados con cámara RGB-D demostraron que, si deseaban una localización más precisa de la cámara, el BA funciona mejor que los métodos directos o el *punto iterativo más cercano* (ICP) y tiene la ventaja de ser menos costoso computacionalmente aun en tiempo real. Debido a que este sistema SLAM genera un mapa y muestra la localización durante la trayectoria, es ideal para el módulo de reconocimiento propuesto. Tiene la ventaja de ser código libre, por lo tanto, se decidió descargarlo y realizar algunas pruebas de funcionamiento del sistema SLAM.

### 3.4.1 Instalación de ORB-SLAM2

La instalación de ORB-SLAM2 se realizó en el sistema LINUX, en este caso se utilizó UBUNTU 18.04 LTS. Para poder utilizar ORB-SLAM2 es necesario instalar las siguientes herramientas.

**CMAKE**

Se instala CMAKE para compilar el programa.

```
1 sudo apt-get install cmake
```

Instalar gcc, g ++.

```
1 sudo apt-get install g++
2 sudo apt-get install gcc
```

**Pangolín**

Instalar dependencias.

```
1 sudo apt-get install libpython2.7-dev
2 sudo apt-get install libglew-dev
3 sudo apt-get install libboost-dev libboost-thread-dev
  libboost-filesystem-dev
```

Descarga el proyecto de Github a local.

```
1 git clone
  https://github.com/stevenlovegrove/Pangolin.git
```

Para compilar e instalar Pangolin se utilizan los siguientes comandos

```
1 cd Pangolin
2 mkdir build
```

```
cd build
cmake ..
make -j
sudo make install
```

**OpenCV**

Para instalar OpenCV primero se descarga el repositorio.

```
git clone https://github.com/opencv/opencv/tree/4.2.0
```

Compile OpenCV (compile varios subprocesos de acuerdo con la situación de su propia computadora, algunos se bloquearán).

```
cd opencv-4.2.0
mkdir build
cd build
cmake -D CMAKE_BUILD_TYPE=Release -D CMAKE_INSTALL_PREFIX=/usr/local ..
sudo make -j4
```

**Instalar OpenCV**

```
sudo make install
```

**Instalar Eigen3**

```
sudo apt-get install libeigen3-dev
```

**Instalar ORB-SLAM2**

Al igual que el paquete de instalación, también puede optar por clonar el almacén

```
git clone https://github.com/raulmur/ORB_SLAM2.git ORB_SLAM2
```

Compilar ORB-SLAM2

```
cd ORB_SLAM2
chmod +x build.sh
./build.sh
```

### 3.4.2 Prueba de funcionamiento

A continuación, se muestran las imágenes (Figuras 15 y 16) de la ejecución de ORB-SLAM2 utilizando el dataset de KITTI.


*Figura 15.- Prueba de ORB-SLAM2 con dataset KITTI secuencia 02*

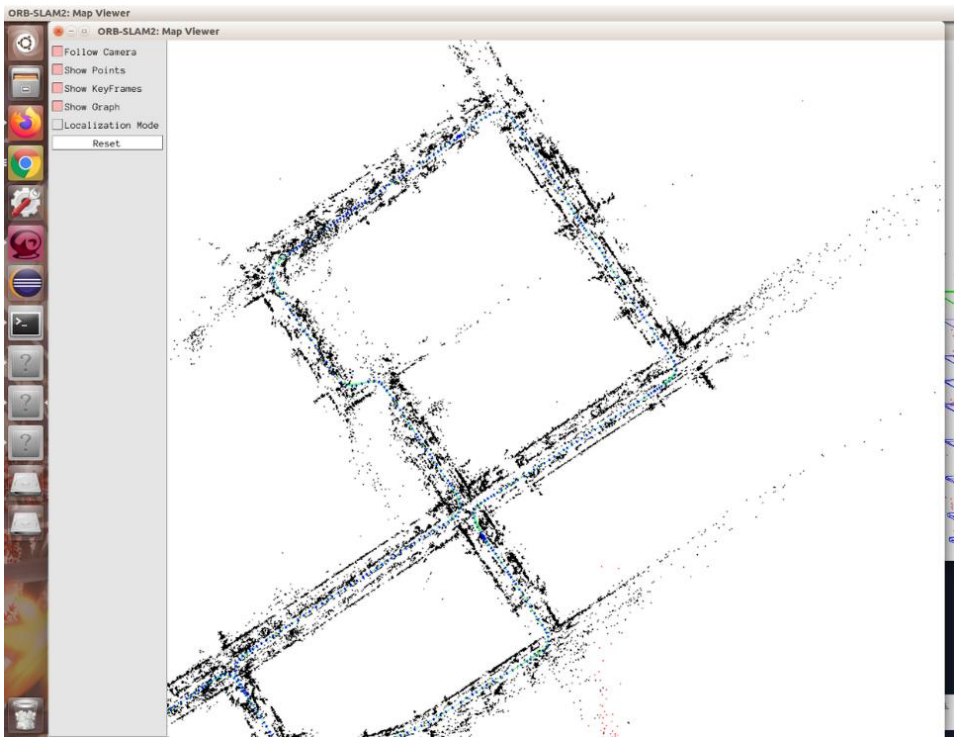



*Figura 16.- Prueba de ORB-SLAM2 con dataset KITTI secuencia 02.*

## 3.5 Módulo de reconocimiento propuesto

El módulo de reconocimiento usa una base de datos de descripciones de los objetos y lugares que se desean reconocer. Cada objeto y lugar se registra desde varios puntos de vista dado que no se sabe la orientación en la que será detectado por el robot. A diferencia de la BD de asociación, la base de datos de reconocimiento no crece, por lo cual el módulo de reconocimiento se tarda siempre lo mismo en ejecutar. La funcionalidad de este módulo consiste en reconocer objetos previamente definidos y localizarlos en el mapa generado en este caso por ORB-SLAM2. El módulo se realizó utilizando el descriptor ORB para la extracción y descripción de características, debido a que es invariante a rotación y escala; además es rápido para la extracción y de bajo costo computacional por lo que permite el funcionamiento en tiempo real. El método de BoVW que se utilizó fue DBoW2 para el reconocimiento.

El proceso de reconocimiento se describe en la Figura 17.

1. Lo primero consiste en cargar la base de datos, previamente creada con ayuda del ejemplo que se encuentra en el repositorio de la librería DBOW2 para identificar los objetos. Al momento de crear la base de datos se crea un archivo de texto que contiene el etiquetado de los objetos y lugares.
2. El siguiente paso es la obtención de la imagen de entrada y su segmentación en cuatro cuadrantes para distribuir 100 puntos/cuadrante dando 400 puntos en total, debido a que al utilizar ORB para la extracción y descripción de características es necesario que los puntos destacados se encuentran distribuidos a través de toda la escena. De otra forma, las características pueden quedar agrupadas en un objeto que no es el objetivo, y pocas o ninguna en un objetivo, si lo hubiera.
3. Los descriptores obtenidos en los cuatro cuadrantes se fusionan para enviar a la librería de Gálvez-López y Tardós para buscar coincidencias de la escena en la base de datos de la BoVW. Si hay una o más coincidencias la librería regresa el identificador y el $Score$ del candidato con más coincidencia.

4. Si se encuentra algún candidato su puntaje ($Score$) es comparado con un umbral como se muestra en la ecuación (3), el cual es establecido para minimizar los falsos positivos.

$$obj = \begin{Bmatrix} 1, si\ Score \geq umbral \\ 0, en\ otro\ caso \end{Bmatrix}$$

*Ecuación 4*

Dónde $Score$ es el puntaje del candidato, que se compara con el valor del umbral establecido para ver si es una "meta".

5. Finalmente, cuando el puntaje del candidato encontrado es mayor al umbral, el nombre del objeto o lugar y el número de la imagen con la que coincide es impreso en el mapa generado por el sistema ORB-SLAM2, en la localización actual que se encuentra el robot.

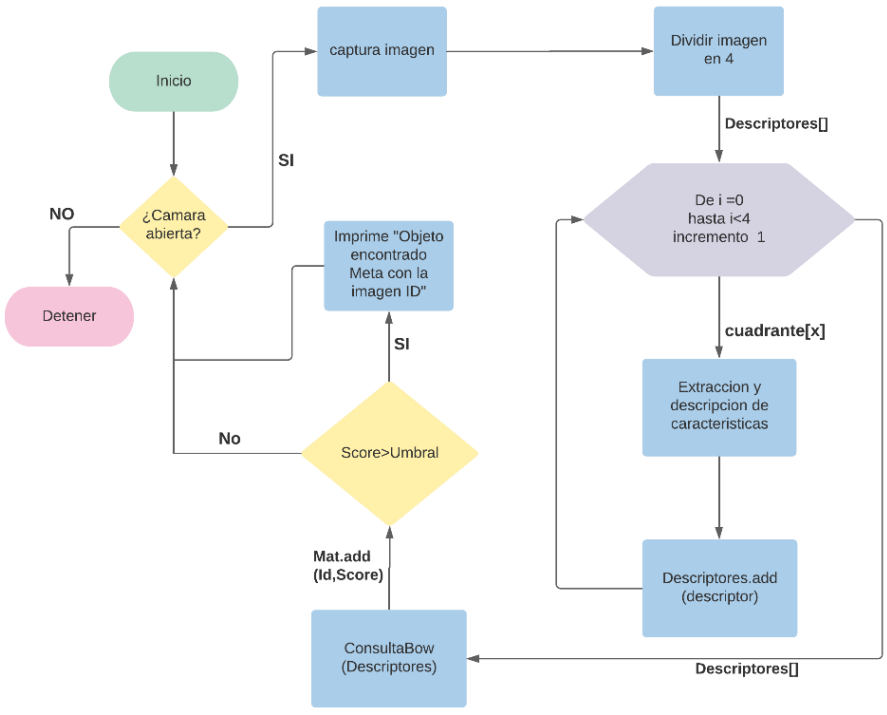



*Figura 17.- Diagrama de flujo del módulo realizado*

### 3.5.1 Creación de BoVW

Para crear la base de datos de las “metas” se utilizó la librería de Gálvez-López y Tardós llamada DBoW2 [43] debido a que en la literatura fue muy recomendada [40], para esto fue necesario realizar las siguientes modificaciones al algoritmo:

En cada imagen se selecciona la “meta” a buscar como una región de interés como se muestra en la Figura 18.

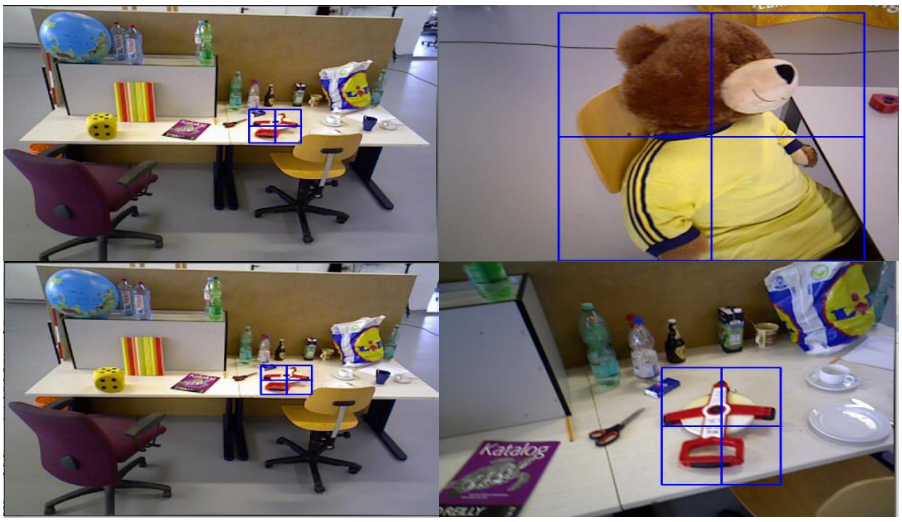



*Figura 18.- Selección de la meta.*

Una vez teniendo las “metas”, se extraen y describen las características (Figura 19), mientras se crea un archivo de etiquetado que guarda la referencia de las “metas” con cada una de las imágenes utilizadas, después crea el vocabulario con la metodología explicada en la sección 3.2.

*Figura 19.- Características extraídas.*

Cuando se termina de crear el vocabulario, se crea la base de datos y se genera un archivo en formato YML. Después se muestra el número de entradas, la cantidad de grupos($K$), el nivel de profundidad ($L$) y la cantidad de palabras visuales obtenidas. Finalmente, se hace una consulta rápida para cada una de las imágenes obteniendo sus 4 mejores candidatos (Figura 20).

```
Database information:
Database: Entries = 100, Using direct index = no. Vocabulary: k = 10, L = 5, Weighting = tf-idf, Scoring = L1-norm, Number of words = 7499
Querying the database:
Searching for Image 0. 4 results:
<EntryId: 0, Score: 1>
<EntryId: 1, Score: 0.0411456>
<EntryId: 2, Score: 0.0379383>
<EntryId: 3, Score: 0.035784>
Searching for Image 1. 4 results:
<EntryId: 1, Score: 1>
<EntryId: 0, Score: 0.0411456>
<EntryId: 2, Score: 0.0288317>
<EntryId: 3, Score: 0.0254485>
Searching for Image 2. 4 results:
<EntryId: 2, Score: 1>
<EntryId: 3, Score: 0.0661508>
<EntryId: 4, Score: 0.0653609>
<EntryId: 0, Score: 0.0379383>
Searching for Image 3. 4 results:
<EntryId: 3, Score: 1>
<EntryId: 4, Score: 0.076415>
<EntryId: 2, Score: 0.0661508>
<EntryId: 0, Score: 0.035784>
Searching for Image 4. 4 results:
<EntryId: 4, Score: 1>
<EntryId: 3, Score: 0.076415>
<EntryId: 2, Score: 0.0653609>
<EntryId: 0, Score: 0.0286079>
Searching for Image 5. 4 results:
<EntryId: 5, Score: 1>
<EntryId: 2, Score: 0.0377128>
<EntryId: 8, Score: 0.0238791>
<EntryId: 21, Score: 0.0206893>
Searching for Image 6. 4 results:
<EntryId: 6, Score: 1>
<EntryId: 3, Score: 0.0265935>
<EntryId: 49, Score: 0.01956>
<EntryId: 93, Score: 0.0185464>
```

*Figura 20.- Creación del vocabulario.*

# Capítulo 4 Pruebas y resultados

En este capítulo se presentan las experimentaciones más importantes realizadas para la BoVW. En las pruebas que se realizaron es necesario definir las variables que se utilizaron para analizar los resultados.

- **Base de datos (DB)**: se refiere a que banco de imágenes se utilizó KITTI, TUM o el propio.
- **Cantidad de características (f)**: Es la cantidad de características extraídas en cada imagen.
- **Número de palabras visuales (Nw):** Es la cantidad de palabras visuales presentes en el árbol de vocabulario, en esta tesis es un valor fijo de K=10 y L=5 dando un total de aproximadamente 100,000 palabras visuales.
- **Similitud (*S*):** es la puntuación resultante que se obtiene de dos imágenes con respecto a sus vectores de BoVW. Esta puntuación puede estar entre 0 y 1.0, donde 1.0 es totalmente parecido y 0 es nada parecido.
- **Alfa (α):** esta variable es el valor dado al umbral que considera si fue encontrada alguna meta.

## 4.1 Prueba para definir parámetros de la base de datos.

Esta prueba se realizó con la finalidad de conocer cuantas palabras visuales podemos obtener con diferente número de características. El método para crear el árbol de vocabulario se definió con **K=10**, que representa la cantidad de ramificaciones y **L=5** que representa la cantidad de niveles en el vocabulario. Tomando en cuenta estos datos, se debería encontrar 100,000 palabras, pero no siempre se consiguen todas ellas, debido a la poca dimensionalidad que se pueden presentar en las descripciones.

*Tabla 3.- Comparativa al crear base de datos con diferente número de características.*

| DB | f | Nw | Tiempo (s) |
|---|---|---|---|
| TUM | 100 | 6,131 | 8.497 |
| | 250 | 11,896 | 13.255 |
| | 500 | 17,097 | 22.507 |
| KITTI | 100 | 9,691 | 14.566 |
| | 250 | 22,230 | 35.659 |
| | 500 | 38,540 | 91,462 |
| CENIDET | 100 | 7,499 | 8.775 |
| | 250 | 15,876 | 20.532 |
| | 500 | 25,316 | 43.649 |

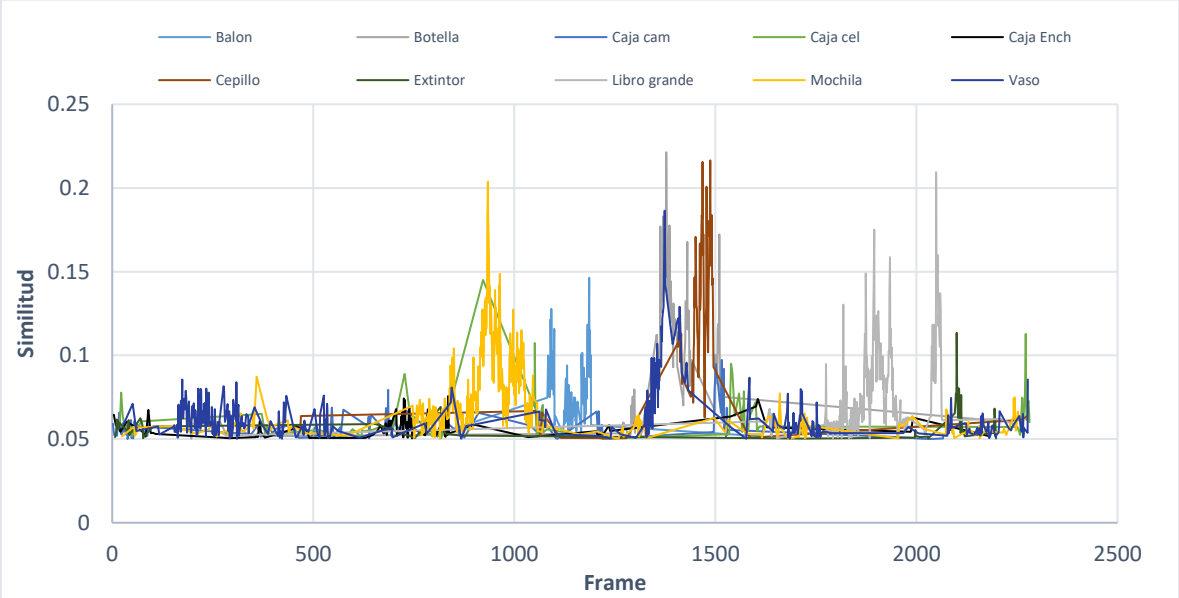
Como se muestra en la tabla 3, al crear una base de datos con mayor cantidad de características extraídas, la cantidad de palabras visuales obtenidas aumenta. Por otro lado, el tiempo para crear la base de datos puede aumentar significativamente cuando se extraen muchas características o se utiliza una gran cantidad de imágenes. Se optó por crear una base de datos extrayendo solo 100 características, aunque son pocas características pueden ser muy buenas características y una ventaja seria que el tiempo para crear y cargar la base de datos sería muy bajo.

## 4.2 Pruebas para definir el valor de la variable del umbral (α):

Lo que se busca en estas pruebas es el valor de alfa (α) que ayude a encontrar las imágenes en las que hay alguna coincidencia con alguna meta sin provocar muchos falsos positivos. Las pruebas se realizaron con la secuencia del banco de imágenes del CENIDET, la cual está conformada con 2280 imágenes donde se colocaron objetos estratégicamente. Estos objetos fueron las metas configuradas con los siguientes parámetros:

- Número de características= 100.
- Cantidad de metas=10.
- Imágenes en total=100.

Para analizar el reconocimiento de objetos y elegir el valor de "**α**" la prueba se realizó con diferentes valores, para obtener información del funcionamiento del módulo propuesto al detectar las metas, se propuso mostrar una gráfica de los valores obtenidos cuando el valor del umbral (α) es 0.05 y analizar los resultados (Gráfica 2).



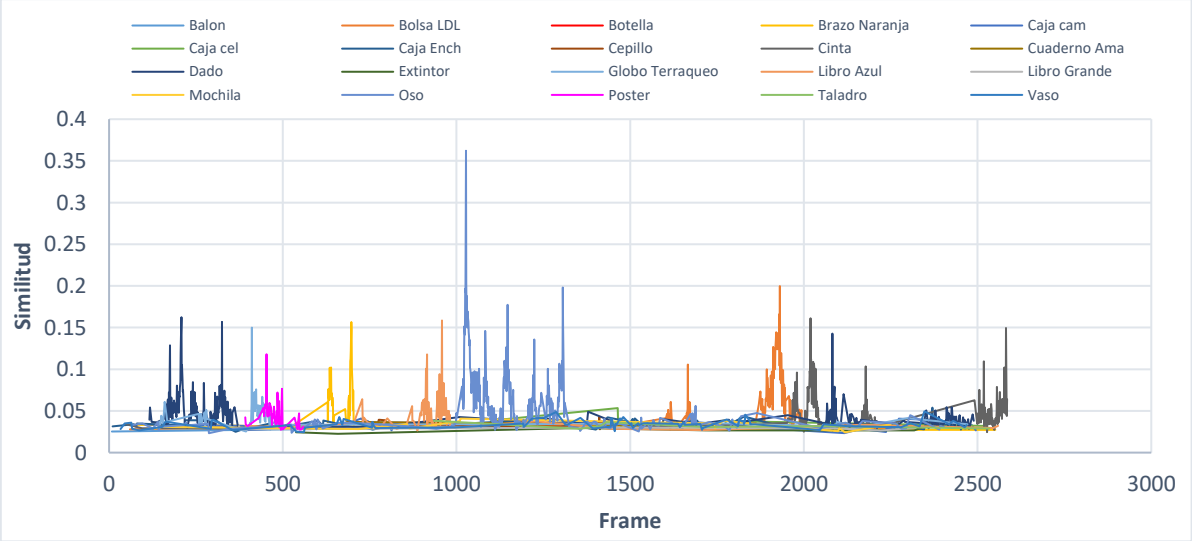

*Gráfica 2.- Valores obtenidos del reconocimiento con BoVW.*

Los resultados obtenidos en esta prueba muestran que el valor asignado al umbral es muy bajo, por lo que la posibilidad de presentar falsos positivos es alta. Por otro lado, utilizando esta gráfica podemos encontrar un valor adecuado para el umbral, se puede observar que las detecciones entre 0.09 y 0.2 muestra la menor cantidad de falsos positivos. Por esta razón, valor de umbral debería estar entre este rango. Por último, tomar en cuenta que si el valor del umbral (**α**) aumenta, entonces el módulo no permite que se detecten tantas metas lo cual permitiría descartar falsos positivos, pero si es demasiado elevado también descartaría verdaderos positivos. En este caso para minimizar los falsos positivos el valor del umbral puede asignarse en 0.09 el cual no descartará verdaderos positivos y descartará la máxima cantidad de falsos posibles.

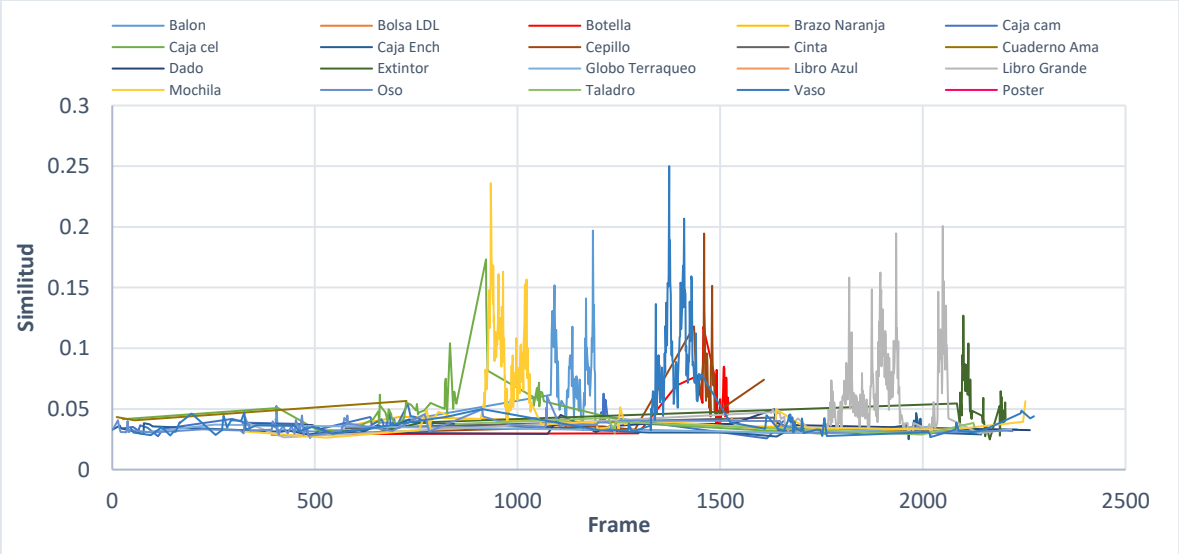
## 4.3 Prueba de análisis de parametrización

La prueba de parametrización consiste en realizar detecciones de objetos con 16 bases de datos con diferentes parámetros (cantidad de característica, cantidad de metas y diferente nivel de profundidad del vocabulario) con el propósito de analizar la robustez del módulo de reconocimiento. Posteriormente se probaron las bases utilizando los recorridos de TUM y CENIDET, para registrar las detecciones de cada cuadro y analizar los datos en las siguientes gráficas.

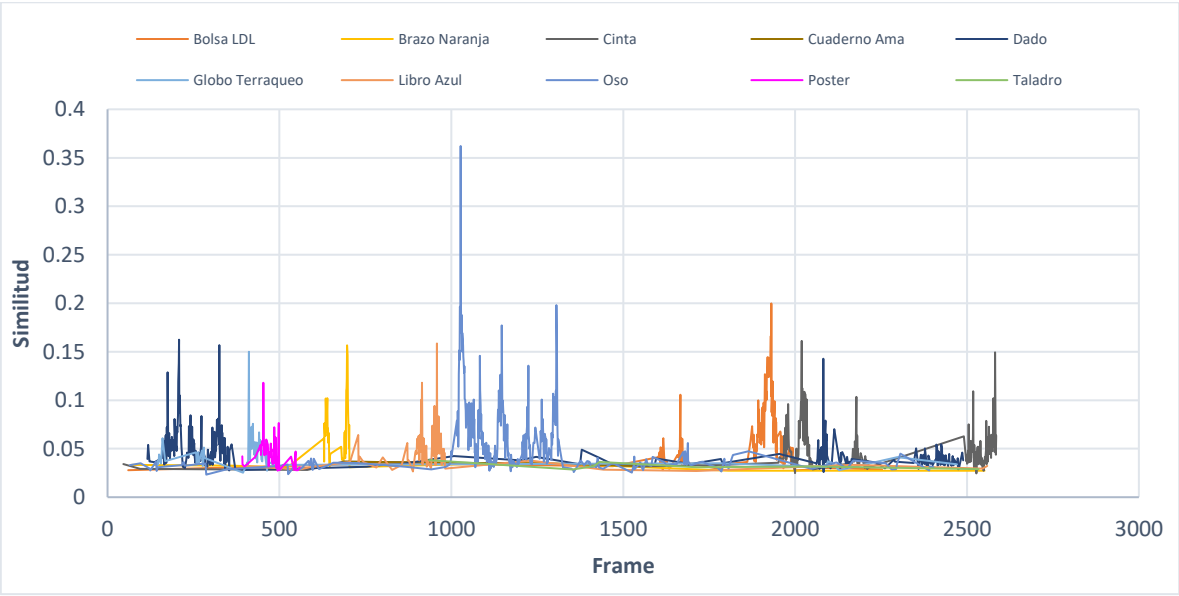
Los datos obtenidos en las pruebas con los recorridos de TUM y CENIDET mostraron que, al usar las bases de datos con diferentes parámetros, no se obtuvo mucha diferencia al detectar las metas. En la gráfica 3 y 4 se observa una muestra de los datos que se obtuvieron de una base de datos definida con 20 metas. Las gráficas 5 y 6 contienen datos de una base definida con 10 metas.



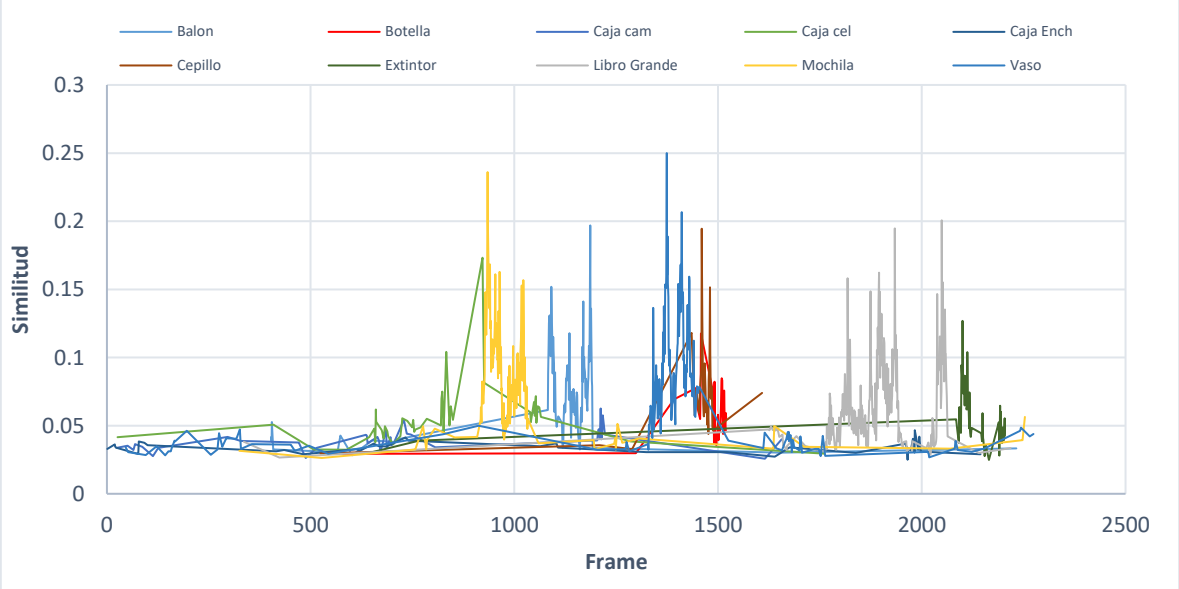

*Gráfica 3.- Detecciones en el recorrido de TUM utilizando una base de datos con 20 metas.*




*Gráfica 4.- Detecciones en el recorrido de CENIDET utilizando una base de datos con 20 metas.*




*Gráfica 5.- Detecciones en el recorrido de TUM utilizando una base de datos con 10 metas.*




*Gráfica 6.- Detecciones en el recorrido de CENIDET utilizando una base de datos con 10 metas.*

Analizando estas graficas se puede notar que las detecciones en bases de datos con diferente cantidad de metas son similares, la única diferencia es la cantidad de objetos con la que fue creada la base de datos. En las tablas 4 y 5 se muestran los resultados de las pruebas con las 16 bases de datos con diferencias en sus parámetros, evaluados bajo las métricas de exhaustividad (*recall)* y exactitud (*accuracy*).

*Tabla 4.- Evaluación de las detecciones utilizando una base de datos con 20 metas.*

| Recorrido | f | K | L | Cant. metas a encontrar | Cant. metas en la BD | Recall (%) | Accuracy (%) |
|---|---|---|---|---|---|---|---|
| TUM | 100 | 10 | 6 | 10 | 20 | 80 | 90 |
| | 200 | | | | | 80 | 90 |
| | 300 | | | | | 80 | 90 |
| | 400 | | | | | 70 | 85 |
| CENIDET | 100 | 10 | 6 | 10 | 20 | 80 | 90 |
| | 200 | | | | | 80 | 90 |
| | 300 | | | | | 70 | 85 |
| | 400 | | | | | 70 | 85 |

*Tabla 5.- Evaluación de las detecciones utilizando una base de datos con 10 metas.*

| Recorrido | f | K | L | Cant. metas a encontrar | Cant. metas en la BD | Recall (%) | Accuracy (%) |
|---|---|---|---|---|---|---|---|
| TUM | 100 | 10 | 5 | 10 | 10 | 80 | 80 |
| | 200 | | | | | 80 | 80 |
| | 300 | | | | | 80 | 80 |
| | 400 | | | | | 70 | 70 |
| CENIDET | 100 | 10 | 5 | 10 | 10 | 80 | 80 |
| | 200 | | | | | 80 | 80 |
| | 300 | | | | | 70 | 70 |
| | 400 | | | | | 70 | 70 |

Con base en lo demostrado anteriormente se puede comprobar que el módulo de reconocimiento es capaz de obtener buenos resultados ante variaciones en las bases de datos como cantidad de objetos y extracción de características debido a su robustez.

## 4.4 Prueba del módulo de reconocimiento.

El principal objetivo en la experimentación es evaluar la precisión y rendimiento del sistema con el módulo de reconocimiento. Para realizar los experimentos se utilizó una PC con AMD Ryzen 5 3600x y una RAM de 32Gb. Los experimentos se realizaron utilizando el dataset TUM [42] (exterior), dataset KITTI [41] (interior) y un dataset propio (interior), el cual se creó haciendo un recorrido en el pasillo la unidad Académica #3 de TecNM/CENIDET. Para la detección se creó una base de datos para cada dataset utilizando 10 objetos con 10 imágenes de diferentes vistas cada uno. En las siguientes imágenes se observan una pequeña cantidad de objetos como ejemplo de metas en cada recorrido (Figuras 21, 22 y 23).

*Figura 22.- Objetos metas para el dataset TUM.*

*Figura 21.- Objetos metas para el dataset CENIDET.*

*Figura 23.- Objetos metas para el dataset KITTI.*

Los parámetros definidos fueron 100 características para interiores y exteriores, para crear el dataset se definió una ramificación $(K) = 10$ y una profundidad $(L) = 5$ para obtener palabras visuales más finas, aunque es posible que no se obtengan las 100,000 palabras visuales por la cantidad de características definidas como se explicó en la prueba para definir los parámetros de la base de datos.

La prueba consiste en realizar 3 corridas a cada recorrido buscando distinta cantidad de metas. La primera corrida busca 6 metas, la segunda 8 y la tercera 10 en cada banco de imágenes. Finalmente, se analizan los resultados bajo la métrica de precisión.

Para la interpretación de los resultados del reconocimiento y mapeo de objetos debe tomarse en cuenta que en los recorridos se pueden apreciar varios objetos en una misma escena, y un mismo objeto puede detectarse desde dos puntos del recorrido mientras el objeto se encuentre en la escena. En las Figuras 24, 25 y 26 se muestran los resultados de cada uno de los recorridos en el cual se aprecia la utilidad de insertar las etiquetas de los objetos encontrados directamente sobre el mapa producido por el ORB-SLAM2. Adicionalmente se decidió agregar a la etiqueta el número de la imagen con la que se relaciona, para obtener más información de qué imagen es muy detectable, cuál se confunde con otras, o no es detectable.

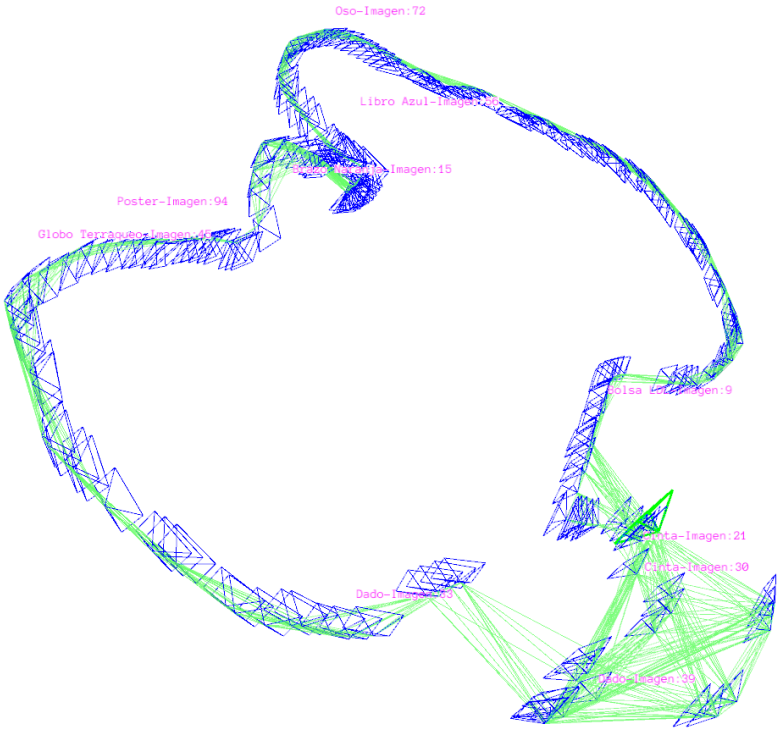



*Figura 24.- Muestra de la detección de objetos durante el recorrido de TUM.*

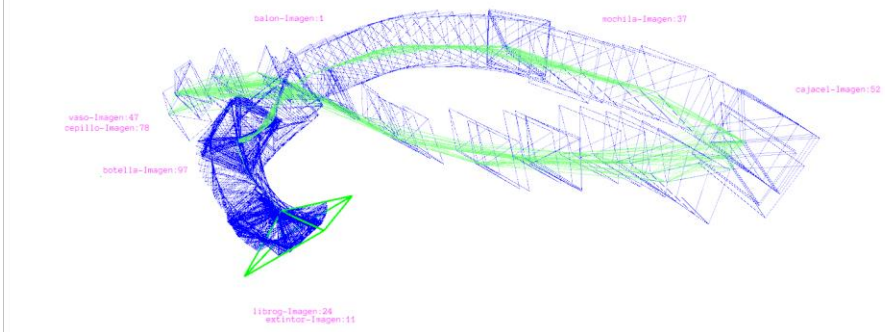



*Figura 25.- Muestra de la detección de objetos durante el recorrido del CENIDET.*

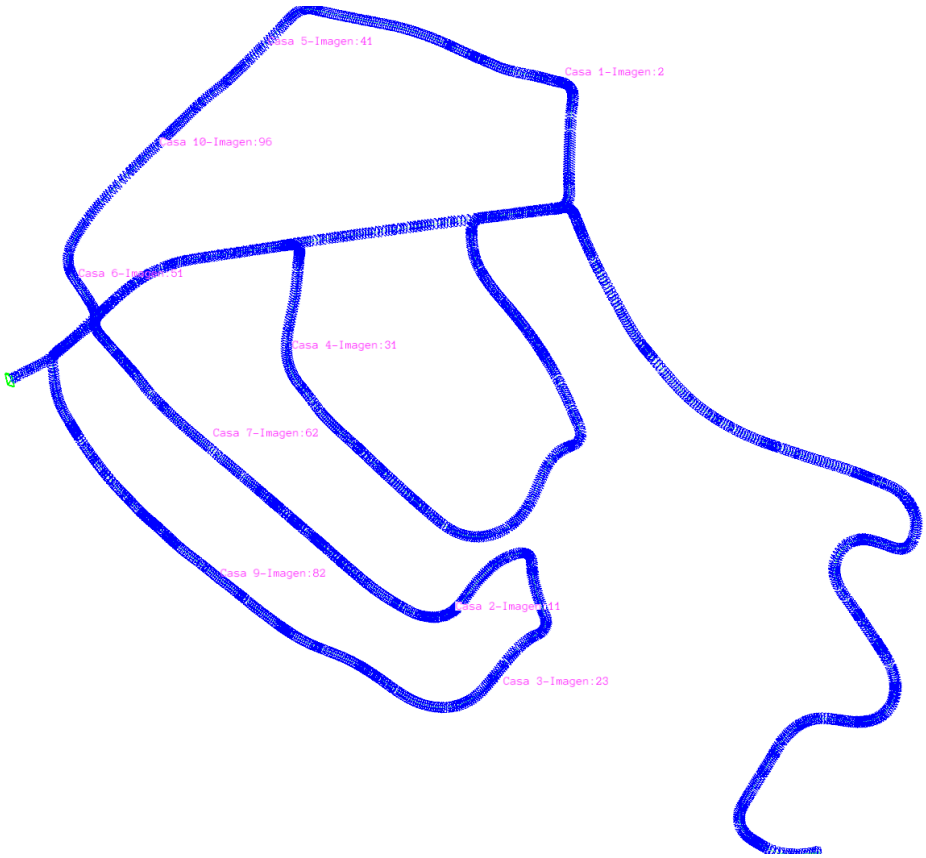
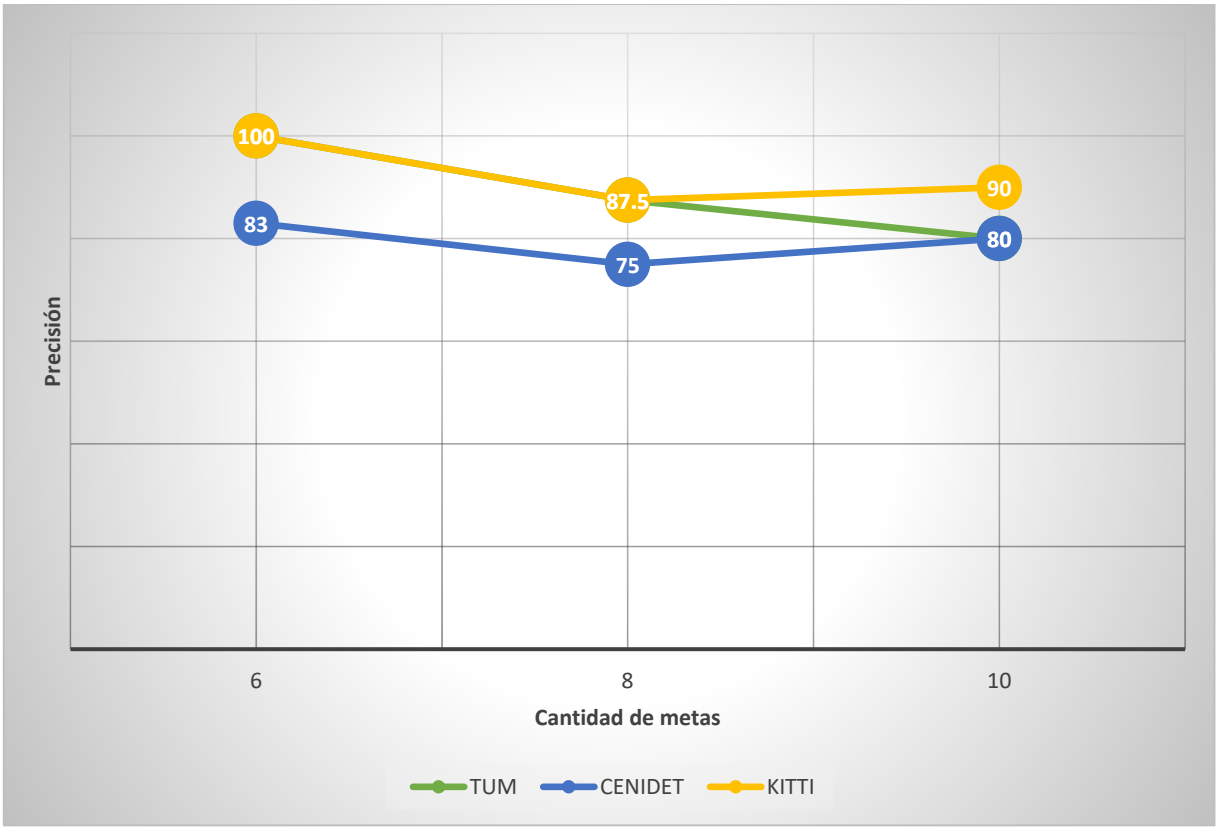



*Figura 26.- Muestra de la detección de objetos durante el recorrido de KITTI.*

Como se mostró en las imágenes anteriores el sistema SLAM se encarga de mapeo y la localización, mientras el módulo de reconocimiento propuesto trata de encontrar coincidencias con las metas para marcar su ubicación en el mapa tomando la localización del vehículo o robot como referencia para la ubicación de la meta encontrada.

## 4.5 Análisis de resultados

Una vez obtenido los resultados de las detecciones, se registró la precisión de reconocimiento que se obtuvo en cada uno de los recorridos. La prueba realizada para el módulo de reconocimiento consistió en realizar 3 recorridos en cada banco de imágenes para detectar 6, 8 y 10 metas.




*Gráfica 7.- Precisión del módulo propuesto.*

En la Gráfica 3 se puede observar que la precisión del módulo de reconocimiento se encuentra entre el 75% al 100%, esto se debe a la cantidad de información extraída de las metas. En la Tabla 4 se muestra cada uno de los porcentajes que se obtuvieron de cada recorrido buscando 6, 8 y 10 metas, en el caso de los porcentajes más bajos cabe la posibilidad que se descartaran verdaderos positivos por el valor de umbral asignado el cual fue **α=0.1.**

*Tabla 6.- Comparativa bajo la métrica de precisión de los diferentes recorridos*

| Cantidad de metas a encontrar | Precisión (%) | | |
|---|---|---|---|
| | INTERIOR | | EXTERIOR |
| | TUM | CENIDET | KITTI |
| **6** | **100** | 83 | **100** |
| **8** | **87.5** | 75 | **87.5** |
| **10** | 80 | 80 | **90** |

Otras causas que puede ocasionar que una meta no se detecte es que tenga poca textura o sea monótono, lo que ocasiona detectar pocos o ningún punto destacado (Figura 27).

*Figura 27.- Imagen con poca textura*

Por otro lado, las imágenes muy pequeñas también son un problema al crear el vocabulario, debido que al seleccionar el objeto a buscar si este se encuentra lejos la imagen resultante para la extracción de características queda muy pequeña (menos de 100 x 100px) por lo que no se le detectan características (Figura 28), una solución para esto es que al momento de seleccionar el objeto se abarque más espacio en la imagen para que este supere el tamaño de 100 x 100px.

*Figura 28.- Imágenes pequeñas.*

En el caso de las imágenes que no muestran características esto genera que, al crear el vocabulario y la base de datos, la imagen sea omitida como una meta, por lo que esto se debe considerar al momento de seleccionar las imágenes que contengan el objeto a buscar, seleccionando imágenes donde el objeto a buscar esté a una distancia adecuada, no más de un metro en interiores. Respecto al rendimiento del módulo, podemos citar su promedio de tiempo de ejecución, mostrado en la Tabla 5.

*Tabla 7.- Mediana del tiempo en milisegundos por cuadro de ORB-SLAM2*

| **ORB-SLAM2** | **Tiempo (ms)** | | |
|---|---|---|---|
| | **INTERIOR** | | **EXTERIOR** |
| | TUM | CENIDET | KITTI |
| **Sin el módulo** | ≈22.4749 ms | ≈26.951 ms | ≈28.5201ms |
| **Con el módulo** | ≈22.9266 ms | ≈27.3714ms | ≈29.3175 ms |

La comparativa de la tabla 5, muestra que el módulo de reconocimiento propuesto aumentó el tiempo de procesamiento por cuadro del ORB-SLAM2 en menos del 3%, lo cual no afecta en gran medida su rendimiento en tiempo real.

## 4.6 Comparativa con estado del arte

En el estado del arte se encontraron dos trabajos similares a este proyecto de tesis. Dichos proyectos utilizan el reconocimiento solo para la detección de objetos como un proceso de soporte en el sistema SLAM. Pillai [29] utiliza la detección de objetos para reconstrucción 3D utilizando un mapa semidenso, el cual mejora la precisión del mapeo. Otro trabajo realizado por P. Jensfelt, S. Ekvall, D. Kragic, y D. Aarno,[31] realizó detección de objetos utilizando histogramas de color de coocurrencia para resolver problemas con fondo complejo, iluminación variable y oclusión de objetos. Ambos trabajos detectan objetos de una manera similar al este trabajo, pero desarrollados con la finalidad de mejorar la precisión del mapeo y la cerradura de ciclos. El modulo propuesto a diferencia de los trabajos anteriores está enfocado en realizar detección tanto de objetos como lugares para ser implementados en situaciones reales, como exploración de zonas riesgosas búsqueda y rescate en el cual se necesita una rápida velocidad de procesamiento.

# Capítulo 5 Conclusiones

En este capítulo final se muestran los objetivos planteados y como fueron cumplidos, los alcances que se lograron y las aportaciones que se consiguieron en esta tesis, finalmente las conclusiones generales y productos académicos.

## 5.1 Objetivos específicos

Se cumplió con los objetivos planteados originalmente de manera exitosa, para ello se realizaron las actividades de la Tabla 6.

*Tabla 8.- Objetivos específicos logrados*

| Objetivo | Comentarios |
|---|---|
| Seleccionar e implementar el algoritmo para el reconocimiento de lugares. | De acuerdo a lo revisado a la literatura el modelo de las BoVW fue el más recomendado por su bajo costo computacional y su rapidez. |
| Desarrollar el sistema en tiempo real usando visión binocular fue cambiado a visión monocular. | Varios trabajos mencionan que, para realizar la asociación en la cerradura de ciclos, no es necesaria la visión binocular debido a que solo se utiliza una de las imágenes obtenidas generalmente la imagen del lado izquierdo. |
| Definir los criterios para evaluar el algoritmo propuesto. | Los criterios para evaluar el módulo de reconocimiento fueron el tiempo de procesamiento por *frame* y la precisión de reconocimiento. |
| Seleccionar e implementar los algoritmos de SLAM más atractivos con respecto a su eficiencia, actualidad, robustez y precisión. | El sistema ORB-SLAM2 es uno de los sistemas más utilizados en la literatura debido a su robustez también gracias a su rápido procesamiento, este sistema puede ser utilizado en tiempo real y tiene buena precisión en el mapeo. |

## 5.2 Alcances

Los alcances propuestos y su cumplimiento en este proyecto se muestran en la Tabla 7.

*Tabla 9.- Alcances cumplidos*

| Alcances | Comentarios |
|---|---|
| Mapear toda una planta de la unidad Académica #3 de TecNM/CENIDET. | Utilizando el recorrido que se obtuvo del CENIDET se logró mapear exitosamente con ORB-SLAM2, pero se encontró el inconveniente que en ciertas ocasiones las características no eran suficientes y se perdía el seguimiento. Esto fue resuelto cambiando el número de características que detecta ORB-SLAM2. |
| Reconocer y ubicar 10 lugares u objetos sobre el mapa. | Se realizaron pruebas en interiores y exteriores para detectar las metas definidas, con 3 recorridos diferentes que se probaban para detectar 6, 8 y 10 objetos o lugares. |
| El sistema se ejecutará en tiempo real. | El sistema ORB-SLAM2 logró realizar la búsqueda de las metas utilizando el módulo propuesto en tiempo real sin errores ni retrasos y una de las ventajas es que el rendimiento aumentó menos del 3%. |

## 5.3 Aportaciones

Las aportaciones que se obtuvieron con la realización de este proyecto fueron:

1. Se implementó un módulo de reconocimiento que permite reportar sus hallazgos en el mapa.
2. Se logró la detección de las metas en tiempo real manteniendo los 30 FPS en un recorrido de más de 4 mil *frames* con una resolución de 1241 x 376 pixeles.
3. Se realizó una modificación a ORB-SLAM2 para emplear la última actualización de las librerías OpenCV, Pangolín, Eigen3.
4. Este módulo es de bajo costo computacional por lo que no afecta al sistema SLAM.
5. El módulo de reconocimiento propuesto puede ser implementado en otro sistema SLAM, ya que este módulo es desacoplado al sistema, es decir que su proceso se realiza por separado al sistema.

## 5.4 Conclusiones generales

Se reportó un módulo de reconocimiento y localización de objetos y lugares para un sistema SLAM. A la vez que el sistema SLAM hace su exploración, mapeo y localización, el módulo de reconocimiento va analizando la escena, detectando y describiendo puntos destacados. Estos puntos son descritos y dichas descripciones se convierten en palabras visuales las cuales se utilizan para buscar coincidencias en una base de datos de metas. El módulo tiene la arquitectura de un módulo de asociación, con la diferencia de que su base de datos va precargada de metas, y es invariante en el tiempo. El módulo reporta sus hallazgos mediante la anotación del identificador del objeto o lugar encontrado sobre el mapa del sistema SLAM y no afecta en gran medida al rendimiento del sistema SLAM en tiempo real. Las metas que fueron mejor detectadas por lo regular son objetos de buen tamaño y que contienen mucha textura de la cual se puede extraer mucha información. Debido a esto, los objetos que se tomen en cuenta para buscar no deben de ser objetos con poca textura, muy pequeños o que tengan reflexiones ya que esto puede afectar al reconocimiento haciendo que aumenten los falsos positivos u obtener puntuaciones muy bajas que generan falsos negativos.

## 5.5 Trabajo a futuro

El concepto del módulo reconocedor ha mostrado su factibilidad, y tiene un gran potencial para ser mejorado con ajustes de parámetros de sus partes: número de puntos, umbrales, configuración del vocabulario, y también con conversiones a otros detectores-descriptores de puntos.

Su implementación dentro de un sistema binocular pudiera mejorar su precisión, al eliminar la ambigüedad de escala y permitir la restricción de tamaño y distancia en los objetos usados como objetivos.

## 5.6 Productos académicos adicionales

**Anexo A: Artículo en la 3ª Jornada de Ciencia y Tecnología Aplicada.**

Jornada de Ciencia y Tecnología Aplicada
Vol. 2, Núm. 2, julio-diciembre 2019. ISSN en trámite

### Controversias en Torno a la Inteligencia Artificial y sus Verdaderos Retos

**Alejandro Aranda, José A.Arizmendi, Katia R.Benítez, José A.Corona, Roberto Munguia, Rances O.Sánchez, Manuel Mejía Lavalle**

*Tecnológico Nacional de México / CENIDET*
*Cuernavaca, Morelos, México;*
*e-mail:{alejandro.aranda19ca, jose.arizmendi19ca, katia.benitez19ca, jose.corona19ca, roberto.munguia19ca, rances.sanchez19ca, mlavalle}@cenidet.edu.mx*

**Resumen:** Hace 63 años que en el *Dartmouth College* (USA) se acuñó el término *Artificial Intelligence* (Inteligencia Artificial). Desde entonces se ha discutido largamente sobre si las máquinas pueden o podrán pensar algún día. En este artículo se relatan los argumentos, a favor o en contra, más citados en torno al tema y se presenta un análisis sobre los verdaderos retos que enfrenta la Inteligencia Artificial en nuestros días, basado en los logros alcanzados por la Inteligencia Artificial. Este análisis se realiza para enfatizar las áreas de trabajo que se prevé serán cruciales en el futuro a corto y mediano plazo para el desarrollo y crecimiento de la Inteligencia Artificial, lo cual a su vez sirve para que los jóvenes investigadores conozcan las áreas de oportunidad para trabajar en nuevos proyectos de investigación de utilidad regional y nacional.

*Palabras clave:* Inteligencia Artificial, Controversias, Logros, Retos, Futuro de la Inteligencia Artificial.

## Anexo B: Artículo en la 7ª Jornada de Ciencia y Tecnología Aplicada.

Jornada de Ciencia y Tecnología Aplicada
Vol. 4, Núm. 2, Julio - Diciembre 2021.

ISSN en trámite

# Módulo de Reconocimiento de Objetos para Robot Móvil con Sistema SLAM.

**Katia R. Benítez C.*, José Ruiz A.****
*Tecnológico Nacional de México / CENIDET*
*Cuernavaca, Morelos, México (e-mail:*
*{ katia.benitez19ca* & josera** }@cenidet.edu.mx)*
*Departamento de Ciencias Computacionales*

Resumen: Se reporta un módulo de reconocimiento y localización de objetos y lugares (las "metas") para un sistema SLAM. El módulo tiene la arquitectura de un módulo de asociación, con la diferencia de que su base de datos va previamente cargada con la descripción de las metas, y es invariante en el tiempo. A la vez que el sistema SLAM hace su exploración, mapeo y localización, el módulo de reconocimiento va analizando la escena, describiéndola mediante bolsas de palabras. Si la bolsa de palabras se encuentra en su base de metas, el módulo reporta su hallazgo mediante la anotación del identificador del objeto o lugar encontrado sobre el mapa del sistema SLAM.

*Keywords:* Detección de objetos, SLAM, Bag of Words.

## 1. INTRODUCCIÓN

Los humanos tenemos la capacidad de reconocer múltiples objetos en una imagen o escenario, aún si el objeto tiene diferencias de rotación, tamaño, escala o posición. Aunque podemos realizar esta tarea sin esfuerzo, en la visión artificial esta tarea es un desafío, debido a que se necesita superar o minimizar varios problemas como la iluminación, la escala, la textura o características(Siegwart & Nourbakhsh, 2004).

La navegación autónoma ha tomado gran importancia a través de los años, por su gran utilidad en áreas como la agricultura (Pou et al., 2010), exploración espacial (Vepa, 2019), atención al cliente (Trulls et al., 2011) y exploración en zonas de riesgo (Calisi et al., 2005). En los primeros trabajos que se realizaron, al robot se le proporcionaba un mapa prefabricado. Bajo este enfoque, no fue posible crear un robot autónomo para realizar tareas útiles, como ayudar a los humanos en entornos hostiles como, la construcción, la minería, la gestión de residuos tóxicos o simplemente tareas de servicio cotidianos como barrer, desplazar objetos, etc. Para cumplir con estas tareas, el robot debería poder crear el mapa por sí mismo, debido a que en la navegación autónoma se requiere generar mapas 3D y localizarse dentro del mismo para crear trayectorias en un entorno desconocido. Para esto se desarrolló la Localización y Mapeo Simultáneo (SLAM)(Bailey & Durrant-Whyte, 2006; Durrant-Whyte & Bailey, 2006), el cual crea un mapa en un ambiente desconocido y se localiza dentro de ese mapa en tiempo real.

El reconocimiento también se ha utilizado en sistemas SLAM, enfocado principalmente al cierre de ciclos, esto para detectar si el robot móvil ha regresado a un área previamente visitada y corregir el error acumulado durante la trayectoria(Liu et al., 2018; Mahon et al., 2008; Se et al., 2005). Uno de los métodos más utilizados en la cerradura de ciclos es la bolsa de palabras visuales (Bag of Visual Words o BoVW) (Csurka et al., 2004). Esta técnica usa una discretización del espacio de descripción empleado por algún método de detección de puntos destacados que aparezcan en una imagen de la escena. Esta discretización de celdas irregulares que llamamos palabras visuales hace que el espacio de descripción tenga dimensiones manejables, y no ocupe espacio para descripciones que no aparecen en la escena, esto permite gestionar grandes conjuntos de imágenes. Para describir las características se puede emplear casi cualquier descriptor, por ejemplo, ORB (Rublee et al., 2011), SIFT (Lowe, 1999, 2004) o SURF (Bay et al., 2006).

Un método de BoVW fue presentado por Gálvez-Lopez y Tardós llamado DBoW2 (Gálvez-López & Tardós, 2012) donde se utilizó por primera vez las bolsas de palabras binarias obtenidas de los descriptores BRIEF (Calonder et al., 2010) y el detector de características FAST (Hast et al., 2015), reduciendo el tiempo de las extracciones de características.

Dentro de los sistemas SLAM se desarrollaron sistemas de reconocimiento como "SLAM with Object Discovery, Modeling and Mapping" (Choudhary et al., 2014) que utiliza los objetos descubiertos como puntos de referencia para ayudar a localizar el robot y para detectar cierres de bucles en mapas más grandes.

Otros trabajos también realizaron detección de objetos y los coloca en el mapa generado por el sistema SLAM.

Pillai y Leonard presentaron un sistema SLAM-AWARE basado en ORB-SLAM (Mur-Artal et al., 2015), enfocado al reconocimiento de objetos utilizando BoVW para la clasificación (Pillai & Leonard, 2015). Otro trabajo realizo detección de objetos para robots de servicios (Jensfelt et al.,

# REFERENCIAS


[1] C. Pou *et al.*, "a Review of Autonomous Navigation Systems in Agricultural Environments" , *J. Virol.* , vol. 6374, no. March , 2010.

[2] R. Vepa, "Introduction to Autonomous Space Vehicles and Robotics" , in *Dynamics and Control of Autonomous Space Vehicles and Robotics*, 2019. doi: 10.1017/9781108525404.002.

[3] E. Trulls *et al.*, "Autonomous navigation for mobile service robots in urban pedestrian environments" , *J. F. Robot.* , vol. 28, no. 3 , 2011, doi: 10.1002/rob.20386.

[4] D. Calisi, A. Farinelli, L. Locchi, and D. Nardi, "Autonomous navigation and exploration in a rescue environment" , in *Proceedings of the 2005 IEEE International Workshop on Safety, Security and Rescue Robotics* , 2005 , vol. 2005 , pp. 54–59. doi: 10.1109/SSRR.2005.1501268.

[5] M. A. Moreno-Armendáriz and H. Calvo, "Visual SLAM and Obstacle Avoidance in Real Time for Mobile Robots Navigation" , *Proc. - 2014 IEEE Int. Conf. Mechatronics, Electron. Automot. Eng. ICMEAE 2014* , pp. 44–49 , 2015, doi: 10.1109/ICMEAE.2014.12.

[6] F. Bertolli and P. Fiorini, "Visual Slam - Mobile robot localization with environment mapping" , *IFAC Proc. Vol.* , vol. 8, no. PART 1 , pp. 286–291 , 2006, doi: 10.3182/20060906-3-IT-2910.00049.

[7] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM: A Versatile and Accurate Monocular SLAM System" , *IEEE Trans. Robot.* , vol. 31, no. 5 , pp. 1147–1163 , 2015, doi: 10.1109/TRO.2015.2463671.

[8] D. Gálvez-López and J. D. Tardós, "Bags of binary words for fast place recognition in image sequences" , *IEEE Trans. Robot.* , vol. 28, no. 5 , pp. 1188–1197 , 2012, doi: 10.1109/TRO.2012.2197158.

[9] K. M. M. Peñaloza, "Navegacion Visual en Trayectorias Cerradas" ,Centro Nacional de Investigación y Desarrollo Tecnológico ,2014.

[10] A. B. Vergara, "Navegación, Localización y Mapeo de Robots Móviles para Trayectorias Pre-especificadas por Imágenes" ,2015.

[11] C. F. D. Velázquez, "Odometría mediante visión artificial usando métodos directos" ,Centro Nacional de Investigación y Desarrollo Tecnológico ,2019.

[12] L. O. Navarrete, "Sistema de Navegación Inercial Asistido por Visión para Robots Móviles Terrestres" ,Centro Nacional de Investigación y Desarrollo Tecnológico Tesis ,2019.

[13] D. E. B. López, "Evaluación de las técnicas SLAM disponibles en ROS" ,Centro Nacional de Investigación y Desarrollo Tecnológico ,2017.

[14] R. Siegwart, I. R. Nourbakhsh, and D. Scaramuzza, *Introduction to Autonomous Mobile Robots* , Second Edi., vol. 2, no. 1. MIT Press , 2004.

[15] S. Se, D. G. Lowe, and J. J. Little, "Vision-based global localization and mapping for mobile robots" , *IEEE Trans. Robot.* , vol. 21, no. 3 , pp. 364–375 , 2005, doi: 10.1109/TRO.2004.839228.

[16] Y. Liu, D. Yang, J. Li, Y. Gu, J. Pi, and X. Zhang, "Stereo Visual-Inertial SLAM with Points and Lines" , *IEEE Access* , vol. 6 , pp. 69381–69392 , 2018, doi: 10.1109/ACCESS.2018.2880689.

[17] I. Mahon, S. B. Williams, O. Pizarro, and M. Johnson-Roberson, "Efficient view-based SLAM using visual loop closures" , *IEEE Trans. Robot.* , vol. 24, no. 5 , pp. 1002–1014 , 2008, doi: 10.1109/TRO.2008.2004888.

[18] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray, "Visual categorization with bags of keypoints (Cited by: 1590)" , *Earth* , vol. 1, no. May , 2004.

[19] D. G. Lowe, "Object Recognition from Local Scale-Invariant Features" , *Proc. Seventh IEEE Int. Conf. Comput. Vis.* , p. 8 , 1999, doi: 10.1109/ICCV.1999.790410.

[20] D. G. Lowe, "Distinctive image features from scale-invariant keypoints" , *Int. J. Comput. Vis.* , vol. 60, no. 2 , 2004, doi: 10.1023/B:VISI.0000029664.99615.94.

[21] H. Bay, T. Tuytelaars, and L. Van Gool, "SURF: Speeded Up Robust Features" , *Comput. Vision–ECCV 2006* , pp. 404–417 , 2006, [Online]. Available: http://link.springer.com/chapter/10.1007/11744023_32

[22] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF" , *Proc. IEEE Int. Conf. Comput. Vis.* , pp. 2564–2571 , 2011, doi: 10.1109/ICCV.2011.6126544.

[23] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "BRIEF: Binary robust independent elementary features" , in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* , 2010 , vol. 6314 LNCS, no. PART 4. doi: 10.1007/978-3-642-15561-1_56.

[24] A. Hast, V. A. Sablina, G. Kylberg, and I. M. Sintorn, "A simple and efficient feature descriptor for fast matching" , in *23rd International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision, WSCG 2015 - Full Papers Proceedings* , 2015 , pp. 135–142.

[25] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras" , *IEEE Trans. Robot.* , vol. 33, no. 5 , pp. 1255–1262 , 2017, doi: 10.1109/TRO.2017.2705103.

[26] T. J. Lee, C. H. Kim, and D. I. D. Cho, "A Monocular Vision Sensor-Based Efficient SLAM Method for Indoor Service Robots" , *IEEE Trans. Ind. Electron.* , vol. 66, no. 1 , pp. 318–328 , 2019, doi: 10.1109/TIE.2018.2826471.

[27] S. Choudhary, A. J. B. Trevor, H. I. Christensen, and F. Dellaert, "SLAM with object discovery, modeling and mapping" , in *IEEE International Conference on Intelligent Robots and Systems* , Oct. 2014 , pp. 1018–1025. doi: 10.1109/IROS.2014.6942683.

[28] F. Zhong, S. Wang, Z. Zhang, C. Chen, and Y. Wang, "Detect-SLAM: Making Object Detection and SLAM Mutually Beneficial" , in *Proceedings - 2018 IEEE Winter Conference on Applications of Computer Vision, WACV 2018* , 2018 , vol. 2018-January. doi: 10.1109/WACV.2018.00115.

[29] S. Pillai and J. J. Leonard, "Monocular SLAM supported object recognition" , in *Robotics: Science and Systems* , 2015 , vol. 11. doi: 10.15607/RSS.2015.XI.034.

[30] J. Engel, T. Schöps, and D. Cremers, "LSD-SLAM: Large-Scale Direct monocular SLAM" , in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* , 2014 , vol. 8690 LNCS, no. PART 2. doi: 10.1007/978-3-319-10605-2_54.

[31] P. Jensfelt, S. Ekvall, D. Kragic, and D. Aarno, "Integrating SLAM and object detection for service robot tasks" , *Cambridge Univ. Press* , 2007, [Online]. Available: http://kiosk.nada.kth.se/~patric/publications/workshop2005.pdf

[32] D. Gálvez-López and J. D. Tardós, "Real-time loop detection with bags of binary words" , *IEEE Int. Conf. Intell. Robot. Syst.* , pp. 51–58 , 2011, doi: 10.1109/IROS.2011.6048525.

[33] C. Grana, D. Borghesani, M. Manfredi, and R. Cucchiara, "A fast approach for integrating ORB descriptors in the bag of words model" , *Multimed. Content Mob. Devices* , vol. 8667 , p. 866709 , 2013, doi: 10.1117/12.2008460.

[34] C. Cadena, D. Gálvez-López, J. D. Tardós, and J. Neira, "Robust place recognition with stereo sequences" , *IEEE Trans. Robot.* , vol. 28, no. 4 , pp. 871–885 , 2012, doi: 10.1109/TRO.2012.2189497.

[35] B. Sirisha, B. Sandhya, C. S. Paidimarry, and A. S. C. Sastry, "Bag-of-Spatial Words(BoSW)Framework for Predicting SAR Image Registration in Real Time Applications" , in *Procedia Computer Science* , 2017 , vol. 115 , pp. 431–439. doi: 10.1016/j.procs.2017.09.102.

[36] J. MacQUEEN, "SOME METHODS FOR CLASSIFICATION AND ANALYSIS OF MULTIVARIATE OBSERVATIONS" , *Comput. Chem.* , vol. 4 , pp. 257–272 , 1967, doi: 10.1007/s11665-016-2173-6.

[37] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Object retrieval with large vocabularies and fast spatial matching" , *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.* , 2007, doi: 10.1109/CVPR.2007.383172.

[38] D. Nistér and H. Stewénius, "Scalable Recognition with a Vocabulary Tree" , *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.* , vol. 2 , pp. 2161–2168 , 2006, doi: 10.1109/CVPR.2006.264.

[39] D. Arthur and S. Vassilvitskii, “K-means++: The advantages of careful seeding” , in *Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms* , 2007 , vol. 07-09-January-2007.

[40] B. Williams, M. Cummins, J. Neira, P. Newman, I. Reid, and J. Tardós, “A comparison of loop closing techniques in monocular SLAM” , *Rob. Auton. Syst.* , vol. 57, no. 12 , 2009, doi: 10.1016/j.robot.2009.06.010.

[41] A. Geiger, P. Lenz, and R. Urtasun, “Are we ready for Autonomous Driving ? The KITTI Vision Benchmark Suite” , *IEEE Conf. Comput. Vis. Pattern Recognit.* , 2012, doi: 10.1109/CVPR.2012.6248074.

[42] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, “A benchmark for the evaluation of RGB-D SLAM systems” , in *IEEE International Conference on Intelligent Robots and Systems* , 2012 , pp. 573–580. doi: 10.1109/IROS.2012.6385773.

[43] D. Galvez-Lopez and J. D. Tardos, “DBoW2” , *Github* , 2012. https://github.com/dorian3d/DBoW2

[44] T. Nicosevici and R. Garcia, “Automatic visual bag-of-words for online robot navigation and mapping” , *IEEE Trans. Robot.* , vol. 28, no. 4 , pp. 886–898 , 2012, doi: 10.1109/TRO.2012.2192013.