

Universidad Internacional de La Rioja (UNIR)

ESIT

Máster Universitario en Inteligencia Artificial

Segmentación semántica RGB-D para navegación en interiores

Trabajo Fin de Máster

Presentado por: Aldalur Corta, Mikel

Director/a: Cobos Guzman, Salvador

Ciudad: Bilbao

Fecha:

Resumen

En este apartado se introducirá un breve resumen en español del trabajo realizado (extensión máxima: 150 palabras). Este resumen debe incluir el objetivo o propósito de la investigación, la metodología, los resultados y las conclusiones.

Palabras Clave: Se deben incluir de 3 a 5 palabras claves en español

Abstract

En este apartado se introducirá un breve resumen en inglés del trabajo realizado (extensión máxima: 150 palabras). Este resumen debe incluir el objetivo o propósito de la investigación, la metodología, los resultados y las conclusiones.

Keywords: Se deben incluir de 3 a 5 palabras claves en inglés

Índice de contenidos

1. Introducción.....	1
1.1 Motivación	1
1.2 Planteamiento del trabajo	2
1.3 Estructura de la memoria	2
2. Contexto y estado del arte.....	3
2.1. SLAM mediante visión por computador.....	3
2.1.1. Cámara monocular	3
2.1.2. Cámara estéreo doble	5
2.1.3. Cámaras estereoscópicas	6
2.1.4. LiDAR.....	8
2.3. Detección y segmentación	8
2.3.1. Detección de objetos	8
2.3.2. Segmentación	11
2.3.2. Posición y trayectoria del ser humano	13
3. Objetivos y metodología de trabajo	15
3.1. Objetivo general.....	15
3.2. Objetivos específicos	15
3.3. Metodología del trabajo	16
4. Identificación de requisitos	18
5. Descripción de la herramienta software desarrollada	19
6. Evaluación.....	20
7. Conclusiones y trabajo futuro	21
7.1. Conclusiones	21
7.2. Líneas de trabajo futuro	21
8. Bibliografía	22
Anexos	26

Anexo I. Título del anexo I	26
Anexo II. Título del anexo II	26
Anexo. Artículo de investigación	26

Índice de tablas

Tabla 1: Planificación de esprints para el proyecto.....	17
--	----

Índice de figuras

Figura 1: Imagen utilizando el algoritmo ORB para detectar objetos mediante características (http://jematoscv.blogspot.com/2014/05/matching-features-with-orb-using-opencv.html).	4
Figura 2: Simulación ORB SLAM de desde diferentes prespectivas (https://image.slidesharecdn.com/orb-slamntu-170423052928/95/orb-slam-proposal-for-ntu-gpu-programming-course-2016-8-638.jpg?cb=1492932027).....	4
Figura 3: Sistema 3D creado por ORB-SLAM (https://image.slidesharecdn.com/orb-slamntu-170423052928/95/orb-slam-proposal-for-ntu-gpu-programming-course-2016-8-638.jpg?cb=1492932027).....	5
Figura 4: Principios de ajuste para cámaras estéreo (https://www.researchgate.net/figure/Principle-drawing-of-a-stereo-camera-setup-Objects-1-2-in-various-depth-ranges-are_fig5_303307354).....	5
Figura 5: Generando un mapa de cámaras estéreo (https://www.researchgate.net/figure/Stereo-visual-SLAM-system-overview-First-we-undistort-and-rectify-the-stereo-images_fig1_257523126).....	6
Figura 6: Tipo cámara estéreo ZED (https://www.stereolabs.com/zed-2/).	6
Figura 7: cámara Intel-RealSense-D435i (https://simplecore.intel.com/newsroom/wp-content/uploads/sites/11/2018/11/Intel-RealSense-D435i-1.jpg).....	7
Figura 8: Imagen de profundidad de la cámara estereoscópica (https://github.com/IntelRealSense/librealsense/issues/4271).	7
Figura 9: Creación de SLAM mediante cámaras estereoscópicas (Zhang et al., 2021).	7
Figura 10: Ejemplo de detector de cara con Haar Cascade (https://miro.medium.com/max/2330/1*ELoJu38cHKMb8e3_IVz_eA.png).	9
Figura 11: Comparando los algoritmos de detección de objetos en base al tiempo de respuesta(Bochkovskiy et al., 2020).	10
Figura 12: Evolución de Imagenet (http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture11.pdf).	10
Figura 13: Imagen exterior segmentada (https://www.researchgate.net/figure/Illustration-of-challenges-in-semantic-segmentation-a-Input-Image-b-Ground-Truth_fig1_332186435).	11
Figura 14: Mapa semántico para navegación (Deng et al., 2020).	11

Figura 15: Detección de objetos y segmentación de una imagen (https://blogs.unity3d.com/2021/04/19/supercharge-your-computer-vision-models-with-synthetic-datasets-built-by-unity/).	12
Figura 16: Estimación de posicionamiento de humanos en entorno segmentado (Seichter et al., 2020).	13
Figura 17: Segmentación de trayectorias después de procesado (Tamaki et al., 2019).	13
Figura 18: Ejemplo de la metodología Scrum (https://www.eclee.com/wp-content/uploads/2020/01/Scrum_Pic.png).	16

1. Introducción

La navegación autónoma es cada vez más popular entre los robots y los vehículos móviles. De esta manera estos robots o vehículos pueden moverse en base a los sensores de captación que integran, siendo necesario la efectividad y la correcta limitación de navegación. Las disciplinas de visión por computador y aprendizaje automático son las que realizan la investigación en el área de navegación autónoma y cada vez realizan avances más elaborados y profundos.

Estos avances han ayudado a que los robots sean cada vez más fiables en cuanto a evitar la colisión con los obstáculos del entorno, crear mapa de navegación, percepción de personas o la segmentación. Los procesos de analizar el entorno han ido cambiando y la seguridad de estos sistemas es esencial en los entornos donde se encuentran personas, animales, plantas o cualquier tipo de obstáculo. La navegación puede ser dentro de un entorno aprendido o un entorno nuevo según las necesidades, pero en todos los casos es necesario identificar o captar bien el entorno para luego poder elegir el camino adecuado. Como ejemplos podemos tener diferentes tipos de entornos como oficinas, escuelas, hospitales o cualquier tipo de edificio donde se quiera introducir un robot.

1.1 Motivación

Los robots autónomos son de una gran ayuda para realizar cualquier tarea y la navegación y la percepción son unos de los puntos en los que se están realizando las mejorando. Hoy en día, podemos ver cómo los robots nos limpian la casa, acompañan a personas, realizan búsquedas de elementos en diferentes entornos o realizan las tareas de inventario de almacenes. Estos robots tienen que ser seguros y evitar todos los obstáculos. Estos obstáculos pueden ser seres vivos y pueden estar en peligro si el robot no realiza bien su trabajo.

La segmentación es una de las áreas en progreso para este tipo de problemática y los últimos avances de la navegación autónoma se centran en la visión por computador para realizarla. La segmentación mediante la captación de imágenes del entorno es esencial para que un robot navegue autónomamente con la suficiente fiabilidad y seguridad para que no ocurra ningún imprevisto.

Es necesario entrenar a los modelos de visión por computador mediante el aprendizaje automático para que realicen una segmentación correcta de los elementos que se encuentran

en el entorno y los clasifique para que el robot tenga claro cuál es el sitio por donde puede circular. El robot tiene que navegar tanto en el interior como exterior y tiene que ser capaz de tomar decisiones correctas a la hora de realizar la navegación, aunque el entorno sea desconocido.

1.2 Planteamiento del trabajo

En este trabajo se intenta abordar el problema en los entornos de interior donde el robot tiene que elegir el camino por donde circular. Para ello, en vez de utilizar solo un sensor RGB para la segmentación, se utilizará un sensor RGB-D para tener una mejor segmentación donde se le añade la profundidad del entorno. A la cámara Intel RealSense D435i se le añadirá un sistema de aprendizaje profundo para detectar las trayectorias de los humanos para que realice una correcta segmentación.

Para la movilidad se necesita un robot que sea capaz de realizar movimientos en el entorno. Este robot móvil será el que se llama Jetbot, el robot móvil de software libre que utiliza como módulo de control el kit de desarrollo Jetson Nano. Al robot se le añadirá un sistema de navegación para que puede moverse de forma autónoma.

Todo el software irá dentro del framework ROS “Robot Operating System”, el software libre que facilita el desarrollo de los proyectos robóticos.

1.3 Estructura de la memoria

En el capítulo 2 se realiza el análisis y estado del arte de todo lo referente a los robots móviles y la percepción por computador. Se analizan diferentes publicaciones como sistemas de percepción de entorno para la localización y mapeo simultáneo (SLAM), Detección de objetos o segmentación semántica. Se parte de conceptos básicos para finalmente centrarse en los conceptos que se abordaran en el presente trabajo.

En el capítulo 3 se detallan los objetivos y la metodología a utilizar, mientras que en el capítulo 4 se identifican los requisitos para la elaboración del presente trabajo de la herramienta software.

Todo el desarrollo a detalle se lleva a cabo en el capítulo 5 y seguidamente se muestran los resultados obtenidos en el capítulo 6, las pruebas realizadas y sus valoraciones para ser mas exactos. Finalmente, en el capítulo 7 se muestran las conclusiones obtenidas.

2. Contexto y estado del arte

La navegación autónoma está avanzando a pasos agigantados, para ello son evidentes los coches autónomos, robots humanoides o cuadrúpedos que ayudan en muchas de las tareas que tienen que realizar los seres humanos. La interacción de los robots con la humanidad también está creciendo ya que cada vez es más seguro y viable, por el avance de la tecnología y la investigación en este ámbito. La utilización de los diferentes sistemas, con extensas características, es indispensable para establecer la seguridad de los humanos y del entorno. En este apartado se analizarán estos elementos y algoritmos de procesamiento de esta información.

2.1. SLAM mediante visión por computador

La visión por ordenador se encarga de capturar y procesar las imágenes. Existen diferentes algoritmos para procesar las imágenes y obtener una perspectiva de tres dimensiones. En este apartado se analizan diferentes formas de crear un mapa utilizando diferentes configuraciones de cámaras para que los robots autónomos tengan conocimiento previo del entorno.

2.1.1. Cámara monocular

Las cámaras monoculares, las que se han utilizado toda la vida para sacar las fotos, son las más utilizada en todos los sistemas por la simplicidad y el pequeño coste que suponen. Con estas cámaras y los algoritmos desarrollados hasta ahora, se pueden realizar muchas operaciones.

Para obtener la posición de un elemento o comparar los elementos de las imágenes se utilizan los algoritmos de extracción de características. Con la extracción de las características se sacan los descriptores que se compararan con la siguiente imagen. Uno de los algoritmos utilizados es rotated brief (ORB) (R. Wang et al., 2019), que viene después de varios algoritmos como SIFT o SURF para mejorar la eficiencia y el coste computacional.

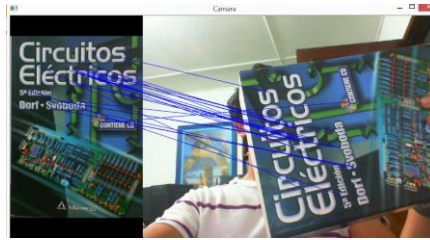


Figura 1: Imagen utilizando el algoritmo ORB para detectar objetos mediante características (<http://jematoscv.blogspot.com/2014/05/matching-features-with-orb-using-opencv.html>).

Este mismo algoritmo que se utiliza para la detección de objetos, realizar las imágenes panorámicas o para la mejora de imágenes en movimiento, también se ha utilizado en conjunto con una unidad de mediciones de inercia (IMU) para crear mapas de tres dimensiones (Mur-Artal et al., 2015). Con el sensor de inercias para detección de velocidad, orientación y aceleraciones se capta el movimiento realizado entre cada fotograma y realiza la estimación del entorno en 3D.

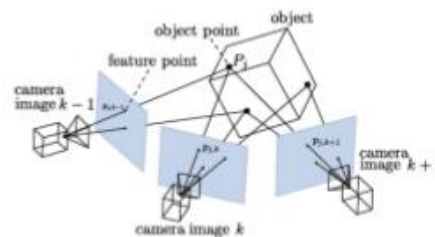


Figura 2: Simulación ORB SLAM de desde diferentes perspectivas (<https://image.slidesharecdn.com/orb-slamntu-170423052928/95/orb-slam-proposal-for-ntu-gpu-programming-course-2016-8-638.jpg?cb=1492932027>).

La diferencia de movimiento calculado por el sensor IMU y los puntos generados en cada fotograma se convierten en un sistema de varias fotos con diferentes perspectivas que puede ser generada a un sistema de tres dimensiones.

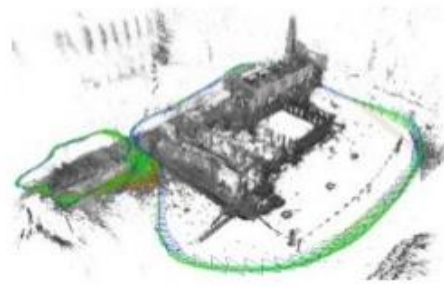


Figura 3: Sistema 3D creado por ORB-SLAM (<https://image.slidesharecdn.com/orb-slamntu-170423052928/95/orb-slam-proposal-for-ntu-gpu-programming-course-2016-8-638.jpg?cb=1492932027>).

2.1.2. Cámara estéreo doble

El sistema monocular puede funcionar bien para ciertas funcionalidades, pero para asegurarse de que las mediciones se hacen de una manera más exacta, siempre es mejor contar con dos cámaras.

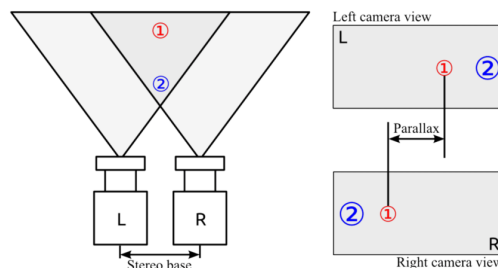


Figura 4: Principios de ajuste para cámaras estéreo (https://www.researchgate.net/figure/Principle-drawing-of-a-stereo-camera-setup-Objects-1-2-in-various-depth-ranges-are_fig5_303307354).

El sistema de funcionalidad es similar a la de cámara monocular, la diferencia es que estas cámaras, cada una de ellas, tienen diferentes ángulos de visión tal y como se puede ver en la Figura 4. Se ve claramente que la perspectiva de la profundidad se plasma en las dos imágenes que ayudan a realizar las medidas necesarias para obtener la visión estéreo (Zaarane et al., 2020). En este caso se puede ver que con las cámaras estéreo es posible medir las distancias para detectar obstáculos.

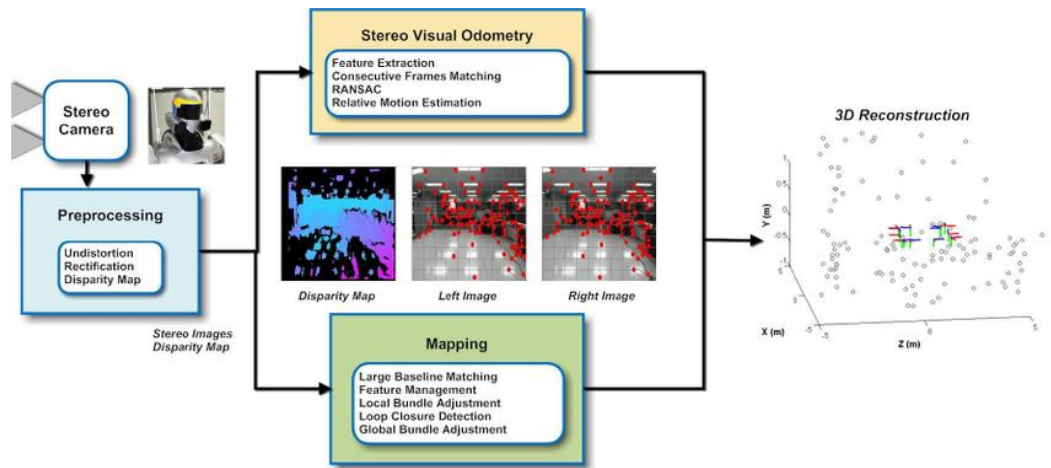


Figura 5: Generando un mapa de cámaras estéreo (https://www.researchgate.net/figure/Stereo-visual-SLAM-system-overview-First-we-undistort-and-rectify-the-stereo-images_fig1_257523126).

Estos sistemas también se utilizan para generar mapas de entornos ya que tienen referencias más precisas de las que se obtienen con cámara monocular (Krombach et al., 2018). Se extraen las características del entorno y las distancias para que en cada fotograma que se detecte el movimiento se autogeneren los punto y distancias generando mapas de tres dimensiones.



Figura 6: Tipo cámara estéreo ZED (<https://www.stereolabs.com/zed-2/>).

Podemos visualizar en la Figura 6 el tipo de cámaras que se utilizan como sistema de captura estéreo. Puede ser que haya dos cámaras instaladas independientemente o que las cámaras estén dentro de un encapsulado como la que vemos en la imagen.

2.1.3. Cámaras estereoscópicas

Las cámaras estéreo no son las únicas que pueden ser más exactas. Las cámaras estereoscópicas son otro tipo de cámaras que pueden medir las distancias. Como ejemplo la cámara Intel RealSense D435i que tiene un sensor infrarrojo para la detección de profundidad con dos receptores diferentes, con similar función a la cámara estéreo, y la cámara RGB.



Figura 7: cámara Intel-RealSense-D435i (<https://simplecore.intel.com/newsroom/wp-content/uploads/sites/11/2018/11/Intel-RealSense-D435i-1.jpg>).

Estas cámaras son denominadas RGB-D, ya que miden la profundidad mediante los sensores infrarrojos. Se pueden utilizar cada uno de los sensores por separado o también traen incorporada la opción de utilizar todos los sensores en conjunto. Utilizando todos los sensores en conjunto se puede obtener la imagen de la cámara con su respectiva profundidad. Esto no es práctico para sacar una imagen de un solo fotograma, la imagen que genera suele ser una imagen con manchas negras por la detección de la profundidad, ver Figura 8.



Figura 8: Imagen de profundidad de la cámara estereoscópica (<https://github.com/IntelRealSense/librealsense/issues/4271>).

También lleva incorporado el sensor IMU lo que facilita el proceso de creación de mapas o entornos en tres dimensiones (Zhang et al., 2021). Existen diferentes algoritmos para la creación de mapas, Figura 9, que ayudan en la creación y entrenamiento de los robots para que se localicen e identifiquen su posición.

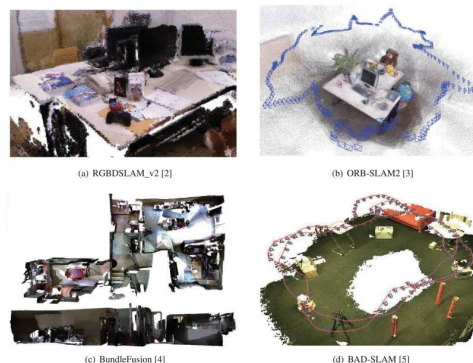


Figura 9: Creación de SLAM mediante cámaras estereoscópicas (Zhang et al., 2021).

2.1.4. LiDAR

Aparte de las cámaras, hay sistemas para la detección de entornos. Uno de ellos se denomina detección de luz y rango (LiDAR, siglas en ingles), que se puede ver hoy en día en los robots aspiradores. Estos sistemas de captación crean un mapa tipo rejilla en dos dimensiones siendo capaces de identificar la localización del robot (Chang et al., 2020).

Todos los elementos mencionados previamente se utilizan para un objetivo similar, aunque en realidad sean diferentes. Por ello se han realizado varios estudios identificando puntos de mejora donde se puede ver que si se juntan estos sistemas de captación, los sistemas funcionan de una manera más eficaz (Shin et al., 2020) (Mu et al., 2020).

2.3. Detección y segmentación

Los sistemas descritos en la sección anterior, las de captación, pueden ayudar a crear mapas y con ello identificar por donde circula el robot. Pero en realidad, lo interesante es que el robot identifique el lugar donde se encuentra y además detecte los objetos que tiene alrededor para que no se choque y tenga una perspectiva más clara para circular por el camino que le corresponda.

La detección y la segmentación de objetos son esas áreas que tratan que los robots autónomos identifiquen (Cao et al., 2019) y etiqueten (Mu et al., 2020) respectivamente los objetos de su entorno. Para ello es indispensable saber cómo se puede realizar un detector de objetos. Hoy en día, parece muy simple este concepto de detectar los objetos ya que lo podemos ver en distintas aplicaciones de nuestros móviles o en otros lugares, pero es interesante saber cómo funciona para poder crear o modificar modelos de redes neuronales convolucionales (CNN) que realizan este proceso. La segmentación de objetos se hace de una manera similar, pero en vez de detectar esos elementos e identificar su posición en la respectiva imagen, colorea la imagen de tal manera que se pueden diferenciar los objetos de forma semántica y de esta manera se crea un mapa de la misma imagen.

2.3.1. Detección de objetos

Como se menciona anteriormente, la detección de objetos es uno de los sistemas para que el robot identifique los elementos que le rodean. Es necesario realizar la ejecución estos

algoritmos en los sistemas locales, ya que no se prescinde de tiempo para ejecución de estos algoritmos (Xu & Wu, 2020).

Antes de nada, para crear estos modelos de detección de objetos es necesario saber como funcionan las CNN. Pero antes de estas redes tan complejas se empezó a hacer la detección de diferentes elementos mediante Haar Cascade Classifiers, que consta de filtros morfológicos con diferentes kernel para la detección de los elementos como puedes ser las caras humanas.

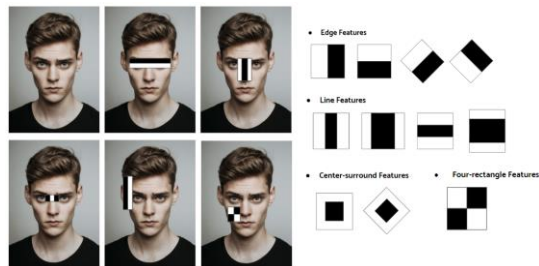


Figura 10: Ejemplo de detector de cara con Haar Cascade

(https://miro.medium.com/max/2330/1*ELoJu38cHKMb8e3_IVz_eA.png).

Estos sistemas pueden funcionar bien para ciertos elementos, pero hoy en día, hay diferentes modos de utilización de CNN que son más fiables. El proceso de detección de objetos se diferencia en tres fases que son la clasificación de imágenes (Chan et al., 2014), clasificación con localización del objeto y detección de diferentes objetos en una sola imagen (Ren et al., 2017).

Los clasificadores han evolucionado para mejorar la eficiencia y muestra de ello es la Figura 11, que ya en el año 2017 había mejorado mucho el tiempo de ejecución de detección de objetos. Hoy en día, ya se hace casi a tiempo real, muestra de ello es Yolov4 (Bochkovskiy et al., 2020) que realiza un buen desempeño teniendo en cuenta la velocidad o el tiempo de respuesta que tiene el algoritmo.

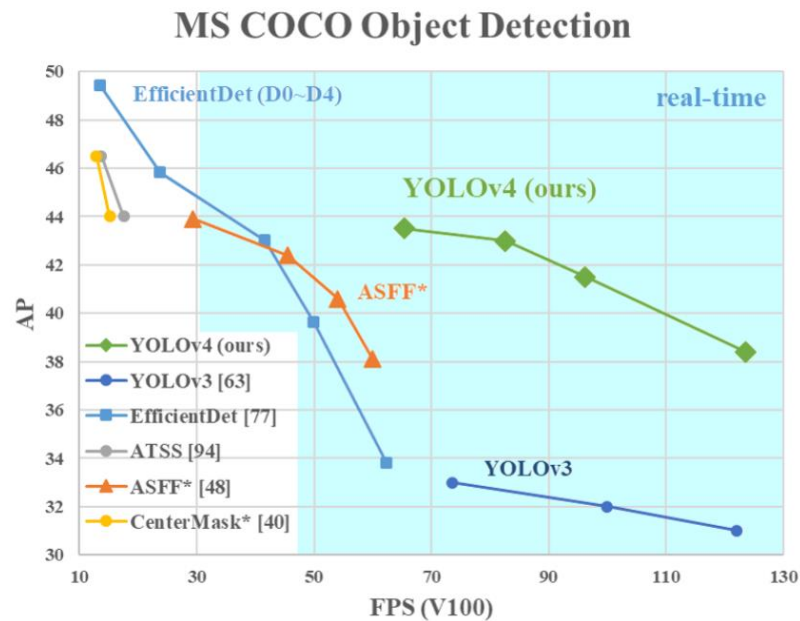


Figura 11: Comparando los algoritmos de detección de objetos en base al tiempo de respuesta(Bochkovskiy et al., 2020).

Uno de los principales motivos de las mejoras de estos sistemas son las competiciones de Imagenet, donde las empresas mas punteras en el sector de la inteligencia artificial (IA) toman parte. Podemos ver en la Figura 12 como han evolucionado durante años los resultados de estas redes neuronales.

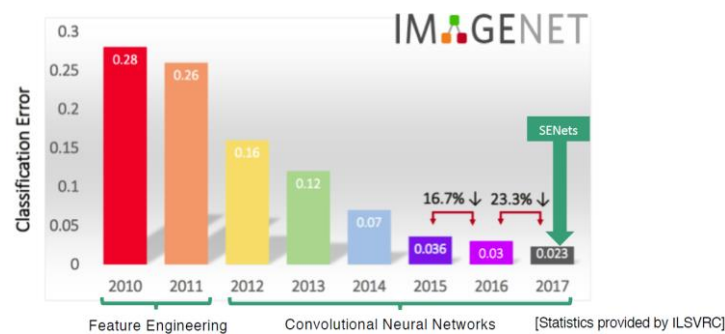


Figura 12: Evolución de Imagenet

(http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture11.pdf).

La detección de objetos en una imagen se puede realizar tanto en dos dimensiones como en tres, utilizando para ello los elementos descritos en el apartado de SLAM mediante visión por computador. La detección volumétrica de los elementos es realmente interesante para los robots autónomos, de esta manera pueden realizar las mediciones necesarias.

2.3.2. Segmentación

En adición a todo lo anterior, la segmentación es uno de los métodos más utilizados últimamente en los sistemas de navegación. Se puede decir que es una extensión a la detección de objetos, para que los sistemas computacionales entiendan mejor todo lo que están visualizando (Wolf et al., 2014).

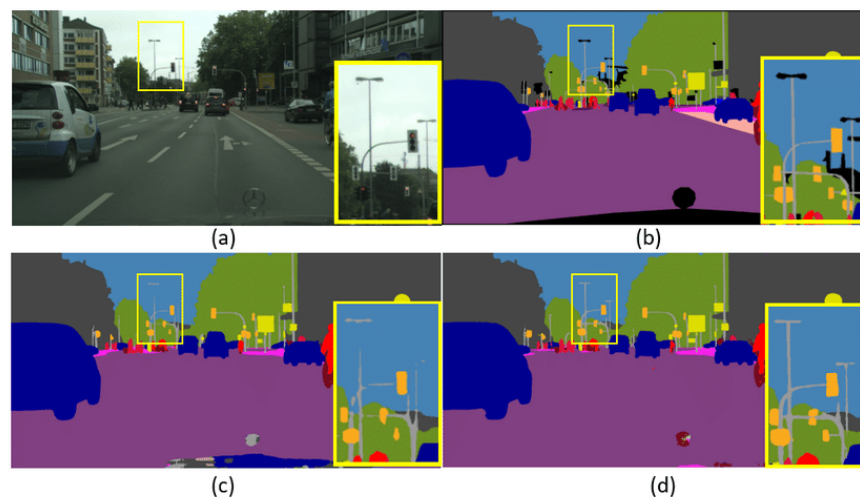


Figura 13: Imagen exterior segmentada (https://www.researchgate.net/figure/Illustration-of-challenges-in-semantic-segmentation-a-Input-Image-b-Ground-Truth_fig1_332186435).

La segmentación puede ser realizada por diferentes tipos de algoritmos con los distintos sistemas vistos anteriormente. Se pueden ver sistemas monoculares (Y. Wang et al., 2020), sistemas que utilizan profundidad (Seichter et al., 2020), mapas creadas en base a la segmentación (Deng et al., 2020), para la navegación interior (Teso-Fz-Betoño et al., 2020) o para navegación exterior añadiendo detección de objetos (Feng et al., 2021).

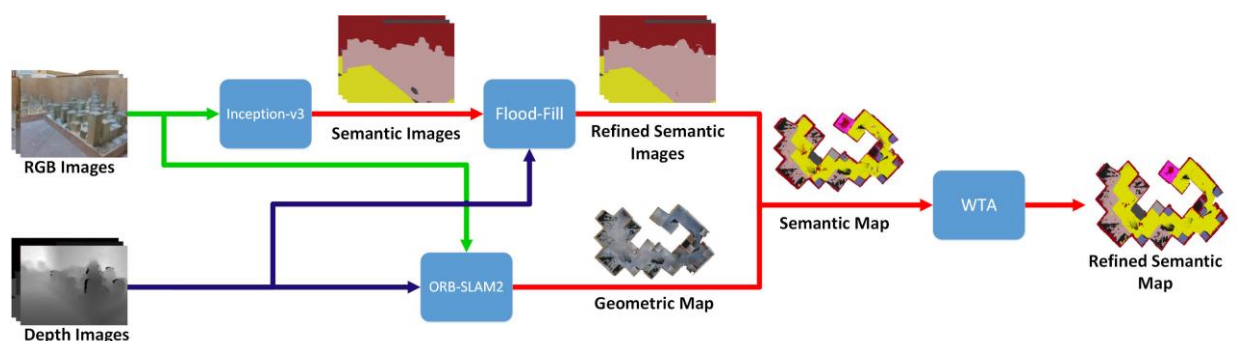


Figura 14: Mapa semántico para navegación (Deng et al., 2020).

Uno de los grandes problemas para los robots es el tiempo de ejecución de estos modelos (Li et al., 2020). Otra cosa tener en cuenta es que las personas no son elementos fijos en estos entornos, por lo que es necesario borrarlos para realizar mapas segmentados (Ai et al., 2020). También es posible realizar el mapa segmentado mediante un sistema monocular que capta el entorno y lo segmenta (Miyamoto et al., 2020).

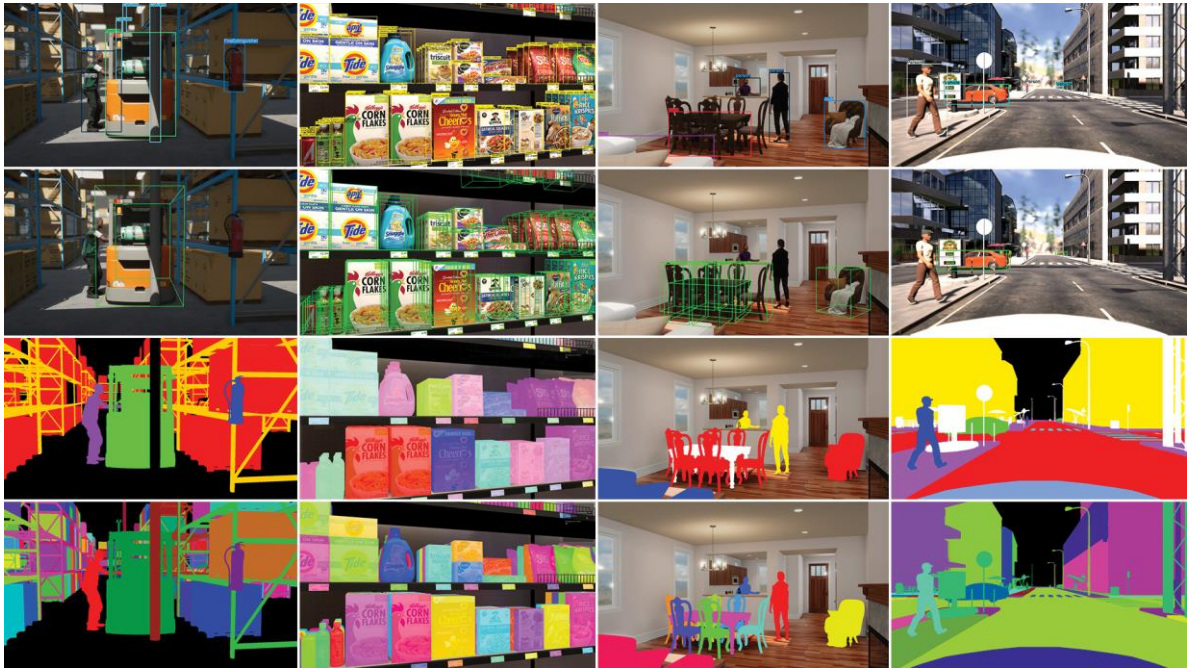


Figura 15: Detección de objetos y segmentación de una imagen

(<https://blogs.unity3d.com/2021/04/19/supercharge-your-computer-vision-models-with-synthetic-datasets-built-by-unity/>).

Estos algoritmos de segmentación pueden ser entrenados de diferentes maneras, supervisado y no supervisado. La diferencia entre ellos es que el no supervisado (Jin et al., 2020), como bien dice el nombre, no se entrena o no realiza el entrenamiento con los resultados, en cambio el supervisado se entrena con el resultado para que puede obtener el mejor resultado posible.

2.3.2. Posición y trayectoria del ser humano

En todos los casos vistos anteriormente, es indispensable saber el estado del ser humano, donde se encuentra y cual es su pose (Cui et al., 2020). Esto nos ayuda en evitar las colisiones o mantener distancias de seguridad por si el ser humano se mueve.

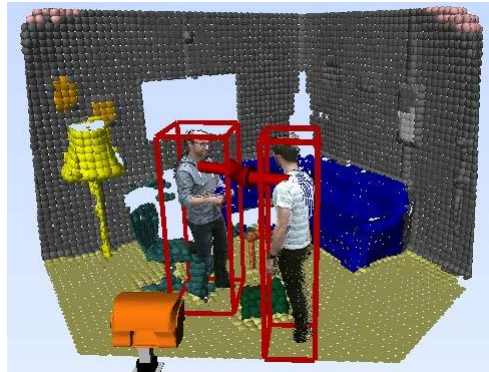


Figura 16: Estimación de posicionamiento de humanos en entorno segmentado (Seichter et al., 2020).

Para la eficiencia de un robot autónomo es de gran ayuda saber por donde andarán los humanos (Tamaki et al., 2019) o también predecir por donde pueden ir para evitar esa trayectoria y elegir la más rápida posible (Liu et al., 2015), ayudando con la gesticulación y mirada del mismo (Moors et al., 2015).

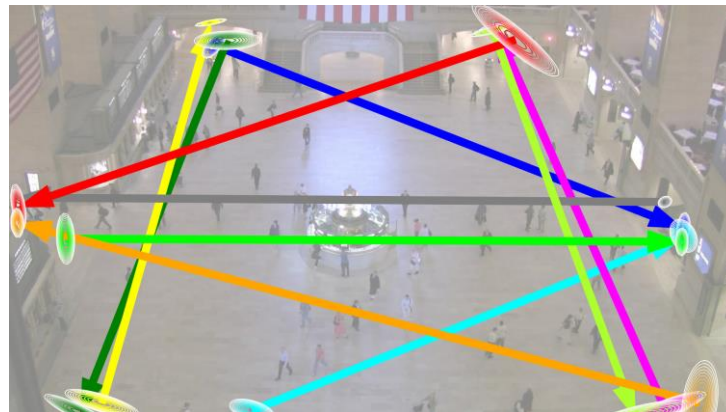


Figura 17: Segmentación de trayectorias después de procesado (Tamaki et al., 2019).

Para poder realizar todo lo mencionado con un buen funcionamiento se necesitan entender bien los entornos y entrenar los algoritmos de una manera eficaz a como se hace para cada lugar donde se encontrará el robot autónomo. Para ello es necesario tener datasets de

distintos lugares como supermercados (Lewandowski et al., 2020), hospitales (H. M. Gross et al., 2017) o domésticos (H.-M. Gross et al., 2019).

3. Objetivos y metodología de trabajo

Tras el análisis de los sistemas de captación y procesamiento para mejora y seguridad de los robots autónomos, es necesario definir qué es lo que se quiere hacer, como se pretende hacer y de que manera se va a hacer para que se pueda realizar una valoración del proceso.

3.1. Objetivo general

El objetivo general es realizar la segmentación del entorno interior para que el robot tenga una visión más precisa al navegar y predecir las trayectorias de los humanos para proteger la integridad de ellos y aumentar la eficiencia a la hora de navegar utilizando para ello las técnicas de aprendizaje profundo.

3.2. Objetivos específicos

En este trabajo se van a realizar las siguientes tareas que ayudaran a describir mejor las especificaciones técnicas.

- Realizar análisis del estado del arte para sistemas de segmentación para navegación autónoma de robots y creación de SLAM
- Detección de personas y su estado de postura para asegurarnos de que no haya colisión entre robot y humano.
- Detectar movimiento en el sistema y predecir su trayectoria para poder esquivar y prevenir el paro del robot.
- Realizar o elegir el algoritmo de segmentación semántica que se ajuste a las necesidades.
- Elegir el algoritmo de navegación que se ajuste a las necesidades del sistema y probar si es la correcta.
- Juntar los objetivos anteriores y valorar si el sistema de predicción de trayectorias es eficiente.
- Implementar el resultado del proceso en la tarjeta de Nvidia Jetson Nano para poder probarlo con Jetbot, el robot creado para ello.
- Realizar la implementación en ROS.

3.3. Metodología del trabajo

Para poder llevar a cabo los objetivos descritos se establecerá una cierta estrategia. Esta estrategia o metodología será la denominada Scrum, que consiste en objetivo a corto plazo o sprints que por si se cambian los pequeños objetivos u ocurre algún cambio en el desarrollo. Se puede ver el ejemplo en la Figura 18. Esta metodología se utiliza mucho en los desarrollos software, que normalmente suelen ser complejos y pueden producirse cambios en cualquier momento durante el transcurso del proyecto.



Figura 18: Ejemplo de la metodología Scrum (https://www.eclee.com/wp-content/uploads/2020/01/Scrum_Pic.png).

Para poder empezar a plantear esta metodología es necesario tener un proyecto y realizar una planificación o objetivos previos. Esta planificación no va a ser la definitiva, pero ayudará a tener un objetivo fijo y tener idea de qué trata el proyecto, es decir, tener una visión clara del proyecto en general. Para poder realizar este planteamiento es necesario realizar el estudio del proyecto, analizar la información que existe y mirar si tienes recursos necesarios para poder llevar a cabo el proyecto.

Teniendo la información analizada y los recursos a disposición, se realizará el planteamiento general, que a su vez contendrá los pequeños sprints o objetivos que se marcarán para el cumplimiento del proyecto. Hay que tener claro de que cada sprint no está estrictamente definido, por lo que se realizará este proceso de Scrum cada vez que trabajemos en el proyecto, ya sea para repasar lo hecho hasta ese punto o realizar algún cambio que se ha detectado.

Cada esprint consistirá en distintas tareas que habrá que llevarlas a cabo en un tiempo determinado, siendo este tiempo flexible. Una vez realizada la tarea, es conveniente que sea subida a una plataforma de desarrollo colaborativo para que los integrantes puedan analizar lo que se ha hecho. En este caso al ser un solo integrante no es necesario subir, pero es interesante para llevar un control de versiones.

Puede ser que cada esprint tenga sus apartados o subtareas a realizar. Por ello en cada sprint o subtask, según donde se encuentre se analizará lo que se ha realizado, como se ha realizado y si existen otras maneras de abordar el problema para poder buscar mejoras para del proyecto.

En este caso los objetivos serán los sprints generales que se han definido y estos tendrán una fecha específica para cumplir el plazo de entrega. Como se comenta anteriormente se evaluará y valorará el trabajo realizado y respecto a esa evaluación se realizarán las modificaciones necesarias para que el proyecto avance.

Para una mejor gestión de este sistema se realizará el concepto de Kanban para que en todo momento se sepa el estado del proyecto. En Kanban se encontrarán los sprints que contendrán las subtareas y en todo momento se podrá saber el estado de cada sprint.

Podemos ver la Tabla 1 donde se definen los sprints del proyecto.

Tabla 1: Planificación de sprints para el proyecto

ESPRINT	OBJETIVO	FECHA ESTIMADA
1	Análisis o estado del arte	6/05/2021
2	Realizar o elegir algoritmo de segmentación	11/05/2021
3	Realizar la detección de personas y sus trayectorias	16/05/2021
4	Elegir el algoritmo de navegación	18/05/2021
5	Juntar los algoritmos para implementarlos en Jetbot	20/05/2021
6	Implementación de los algoritmos en el Sistema ROS	25/05/2021
7	Pruebas para comprobar y calificar el funcionamiento	30/05/2021

4. Identificación de requisitos

En este capítulo se debe indicar el trabajo previo realizado para guiar el desarrollo del software. Esto debería incluir la identificación adecuada del problema a tratar, así como del contexto habitual de uso o funcionamiento de la aplicación. Idealmente, la identificación de requisitos se debería hacer contando con expertos en la materia a tratar.

5. Descripción de la herramienta software desarrollada

En el caso de **desarrollos de software**, deberían aportarse detalles del proceso de desarrollo, incluyendo las fases e hitos del proceso. También deben presentarse diagramas explicativos de la arquitectura o funcionamiento, así como capturas de pantalla que permitan al lector entender el funcionamiento del programa.

6. Evaluación

La evaluación debería cubrir por lo menos una mínima evaluación de la usabilidad de la herramienta, así como de su aplicabilidad para resolver el problema propuesto. Estas evaluaciones suelen realizarse con usuarios expertos.

7. Conclusiones y trabajo futuro

7.1. Conclusiones

Este último capítulo (en ocasiones, dos capítulos complementarios) es habitual en todos los tipos de trabajos y presenta el resumen final de tu trabajo y debe servir para informar del alcance y relevancia de tu aportación.

Suele estructurarse empezando con un resumen del problema tratado, de cómo se ha abordado y de por qué la solución sería válida.

Es recomendable que incluya también un resumen de las contribuciones del trabajo, en el que relaciones las contribuciones y los resultados obtenidos con los objetivos que habías planteado para el trabajo, discutiendo hasta qué punto has conseguido resolver los objetivos planteados.

7.2. Líneas de trabajo futuro

Finalmente, se suele dedicar una última sección a hablar de líneas de trabajo futuro que podrían aportar valor añadido al TFM realizado. La sección debería señalar las perspectivas de futuro que abre el trabajo desarrollado para el campo de estudio definido. En el fondo, debes justificar de qué modo puede emplearse la aportación que has desarrollado y en qué campos.

8. Bibliografía

- Ai, Y., Rui, T., Lu, M., Fu, L., Liu, S., & Wang, S. (2020). DDL-SLAM: A robust RGB-d SLAM in dynamic environments combined with deep learning. *IEEE Access*, 8, 162335–162342. <https://doi.org/10.1109/ACCESS.2020.2991441>
- Bochkovskiy, A., Wang, C.-Y., & Liao, H.-Y. M. (2020). YOLOv4: Optimal Speed and Accuracy of Object Detection. *ArXiv*. <http://arxiv.org/abs/2004.10934>
- Cao, C., Wang, B., Zhang, W., Zeng, X., Yan, X., Feng, Z., Liu, Y., & Wu, Z. (2019). An Improved Faster R-CNN for Small Object Detection. *IEEE Access*, 7, 106838–106846. <https://doi.org/10.1109/ACCESS.2019.2932731>
- Chan, T.-H., Jia, K., Gao, S., Lu, J., Zeng, Z., & Ma, Y. (2014). PCANet: A Simple Deep Learning Baseline for Image Classification? *IEEE Transactions on Image Processing*, 24(12), 5017–5032. <https://doi.org/10.1109/TIP.2015.2475625>
- Chang, L., Niu, X., & Liu, T. (2020). Gnss/imu/odo/lidar-slam integrated navigation system using imu/odo pre-integration. *Sensors (Switzerland)*, 20(17), 1–18. <https://doi.org/10.3390/s20174702>
- Cui, X., Lu, C., & Wang, J. (2020). 3D Semantic Map Construction Using Improved ORB-SLAM2 for Mobile Robot in Edge Computing Environment. *IEEE Access*, 8, 67179–67191. <https://doi.org/10.1109/ACCESS.2020.2983488>
- Deng, W., Huang, K., Chen, X., Zhou, Z., Shi, C., Guo, R., & Zhang, H. (2020). Semantic RGB-D SLAM for Rescue Robot Navigation. *IEEE Access*, 8, 221320–221329. <https://doi.org/10.1109/ACCESS.2020.3031867>
- Feng, D., Haase-Schutz, C., Rosenbaum, L., Hertlein, H., Glaser, C., Timm, F., Wiesbeck, W., & Dietmayer, K. (2021). Deep Multi-Modal Object Detection and Semantic Segmentation for Autonomous Driving: Datasets, Methods, and Challenges. *IEEE Transactions on Intelligent Transportation Systems*, 22(3), 1341–1360. <https://doi.org/10.1109/TITS.2020.2972974>
- Gross, H.-M., Scheidig, A., Muller, S., Schutz, B., Fricke, C., & Meyer, S. (2019). Living with a Mobile Companion Robot in your Own Apartment - Final Implementation and Results of a 20-Weeks Field Study with 20 Seniors*. *2019 International Conference on Robotics and Automation (ICRA)*, 2019-May, 2253–2259. <https://doi.org/10.1109/ICRA.2019.8793693>

- Gross, H. M., Meyer, S., Scheidig, A., Eisenbach, M., Mueller, S., Trinh, T. Q., Wengelfeld, T., Bley, A., Martin, C., & Fricke, C. (2017). Mobile robot companion for walking training of stroke patients in clinical post-stroke rehabilitation. *Proceedings - IEEE International Conference on Robotics and Automation*, 1028–1035. <https://doi.org/10.1109/ICRA.2017.7989124>
- Jin, S., Chen, L., Sun, R., & McLoone, S. (2020). A novel vSLAM framework with unsupervised semantic segmentation based on adversarial transfer learning. *Applied Soft Computing Journal*, 90, 106153. <https://doi.org/10.1016/j.asoc.2020.106153>
- Krombach, N., Droeschel, D., Houben, S., & Behnke, S. (2018). Feature-based visual odometry prior for real-time semi-dense stereo SLAM. *Robotics and Autonomous Systems*, 109, 38–58. <https://doi.org/10.1016/j.robot.2018.08.002>
- Lewandowski, B., Wengelfeld, T., Muller, S., Jenny, M., Glende, S., Schroter, C., Bley, A., & Gross, H.-M. (2020). Socially Compliant Human-Robot Interaction for Autonomous Scanning Tasks in Supermarket Environments. *2020 29th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, 363–370. <https://doi.org/10.1109/RO-MAN47096.2020.9223568>
- Li, F., Li, W., Chen, W., Xu, W., Huang, L., Li, D., Cai, S., Yang, M., Xiong, X., & Liu, Y. (2020). A Mobile Robot Visual SLAM System with Enhanced Semantics Segmentation. *IEEE Access*, 8, 25442–25458. <https://doi.org/10.1109/ACCESS.2020.2970238>
- Liu, W., Chan, A. B., Lau, R. W. H., & Manochaiee, D. (2015). Leveraging long-term predictions and online learning in agent-based multiple person tracking. In *IEEE Transactions on Circuits and Systems for Video Technology* (Vol. 25, Issue 3). <https://doi.org/10.1109/TCSVT.2014.2344511>
- Miyamoto, R., Adachi, M., Ishida, H., Watanabe, T., Matsutani, K., Komatsuzaki, H., Sakata, S., Yokota, R., & Kobayashi, S. (2020). Visual navigation based on semantic segmentation using only a monocular camera as an external sensor. In *Journal of Robotics and Mechatronics* (Vol. 32, Issue 6). <https://doi.org/10.20965/jrm.2020.p1137>
- Moors, P., Germeys, F., Pomianowska, I., & Verfaillie, K. (2015). Perceiving where another person is looking: The integration of head and body information in estimating another person's gaze. *Frontiers in Psychology*, 6(JUN). <https://doi.org/10.3389/fpsyg.2015.00909>
- Mu, L., Yao, P., Zheng, Y., Chen, K., Wang, F., & Qi, N. (2020). Research on SLAM algorithm

- of mobile robot based on the fusion of 2D LiDAR and depth camera. *IEEE Access*, 8, 157628–157642. <https://doi.org/10.1109/ACCESS.2020.3019659>
- Mur-Artal, R., Montiel, J. M. M., & Tardos, J. D. (2015). ORB-SLAM: a Versatile and Accurate Monocular SLAM System. *IEEE Transactions on Robotics*, 31(5), 1147–1163. <https://doi.org/10.1109/TRO.2015.2463671>
- Ren, S., He, K., Girshick, R., & Sun, J. (2017). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In *IEEE Transactions on Pattern Analysis and Machine Intelligence* (Vol. 39, Issue 6). <https://doi.org/10.1109/TPAMI.2016.2577031>
- Seichter, D., Köhler, M., Lewandowski, B., Wengefeld, T., & Gross, H.-M. (2020). Efficient RGB-D Semantic Segmentation for Indoor Scene Analysis. *ArXiv*. <http://arxiv.org/abs/2011.06961>
- Shin, Y.-S., Yeong, ., Park, S., & Kim, A. (2020). DVL-SLAM: sparse depth enhanced direct visual-LiDAR SLAM. *Autonomous Robots*, 44, 115–130. <https://doi.org/10.1007/s10514-019-09881-0>
- Tamaki, T., Ogawa, D., Raytchev, B., & Kaneda, K. (2019). Semantic segmentation of trajectories with improved agent models for pedestrian behavior analysis. *Advanced Robotics*, 33(3–4), 153–168. <https://doi.org/10.1080/01691864.2018.1554508>
- Teso-Fz-Betoño, D., Zulueta, E., Sánchez-Chica, A., Fernandez-Gamiz, U., & Saenz-Aguirre, A. (2020). Semantic segmentation to develop an indoor navigation system for an autonomous mobile robot. *Mathematics*, 8(5). <https://doi.org/10.3390/MATH8050855>
- Wang, R., Zhang, W., Shi, Y., Wang, X., & Cao, W. (2019). GA-ORB: A New Efficient Feature Extraction Algorithm for Multispectral Images Based on Geometric Algebra. *IEEE Access*, 7, 71235–71244. <https://doi.org/10.1109/ACCESS.2019.2918813>
- Wang, Y., Chen, Q., Chen, S., & Wu, J. (2020). Multi-Scale Convolutional Features Network for Semantic Segmentation in Indoor Scenes. *IEEE Access*, 8, 89575–89583. <https://doi.org/10.1109/ACCESS.2020.2993570>
- Wolf, D., Bajones, M., Prankl, J., & Vincze, M. (2014). *Find my mug: Efficient object search with a mobile robot using semantic segmentation*. <http://arxiv.org/abs/1404.5765>
- Xu, D., & Wu, Y. (2020). Improved YOLO-V3 with densenet for multi-scale remote sensing target detection. *Sensors (Switzerland)*, 20(15), 1–24. <https://doi.org/10.3390/s20154276>

- Zaarane, A., Slimani, I., Al Okaishi, W., Atouf, I., & Hamdoun, A. (2020). Distance measurement system for autonomous vehicles using stereo camera. *Array*, 5, 100016. <https://doi.org/10.1016/j.array.2020.100016>
- Zhang, S., Zheng, L., & Tao, W. (2021). Survey and Evaluation of RGB-D SLAM. *IEEE Access*, 9, 21367–21387. <https://doi.org/10.1109/ACCESS.2021.3053188>

Anexos

Cuestionarios, encuestas, resultados de pilotos, documentos adicionales, capturas de pantalla, etc.

Además, al final de la memoria y como un anexo obligatorio deberá incluirse un artículo de investigación que resuma el trabajo realizado y los principales resultados obtenidos.

Anexo I. Título del anexo I

Anexo I.

Anexo II. Título del anexo II

Anexo II.

Anexo. Artículo de investigación

Al final de la memoria y como un anexo obligatorio deberá incluirse un artículo de investigación que resuma el trabajo realizado y los principales resultados obtenidos. Este artículo deberá seguir la plantilla proporcionada a continuación y tendrá una extensión de entre 6 y 8 páginas. El artículo se podrá desarrollar en español o en inglés (para ello utilizar la plantilla adecuada).

Título

Nombre y Apellidos del Estudiante

Universidad Internacional de la Rioja, Logroño (España)

Fecha



RESUMEN

Breve resumen del trabajo realizado (extensión máxima: 150 palabras). Este resumen debe incluir el objetivo o propósito de la investigación, la metodología, los resultados y las conclusiones.

PALABRAS CLAVE

Tres a cinco palabras clave ordenadas alfabéticamente y separadas por comas.

I. INTRODUCCIÓN

INTRODUCCIÓN en la que debes resumir de forma esquemática pero suficientemente clara lo esencial de cada una de las partes del trabajo.

La lectura de esta introducción ha de dar una primera idea clara de lo que se pretendía, las conclusiones a las que se ha llegado y del procedimiento seguido.

II. ESTADO DEL ARTE

Estudio a fondo el dominio de aplicación, citando numerosas referencias.

Debe aportar un buen resumen del conocimiento que ya existe en el campo de los problemas habituales identificados.

Numerar las citas de forma consecutiva entre corchetes [1].

III. OBJETIVOS Y METODOLOGÍA

Objetivo general, objetivos específicos y metodología de trabajo aplicada.

IV. CONTRIBUCIÓN

Desarrollar la descripción de tu contribución.

XX
XX

XX
XX
XX

XX
XX

XX
XX

V. EVALUACIÓN Y RESULTADOS

Descripción de la evaluación y los resultados obtenidos (Tipo 2. Desarrollo de Software).

XX
XX

Evaluación 1

XX
XX

XX
XX

En la Figura 1...

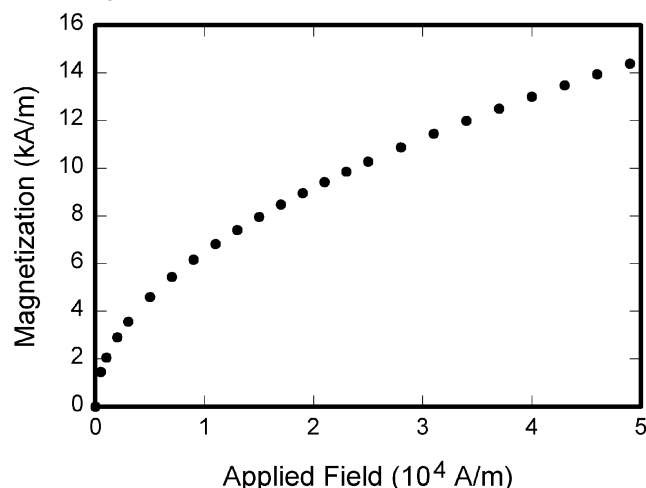


Fig. 1. Magnetization as a function of applied field. Note that “Fig.” is abbreviated. There is a period after the figure number, followed by two spaces. It is good practice to explain the significance of the figure in the caption.

En la Tabla I ...

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX.

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

Evaluación 2

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXX

VI. DISCUSIÓN

Tras la presentación objetiva de los resultados, querrás aportar una discusión de los mismos.

VII. CONCLUSIONES

Resumen de las contribuciones del trabajo, en el que relaciones las contribuciones y los resultados obtenidos con los objetivos que habías planteado para el trabajo, discutiendo hasta qué punto has conseguido resolver los objetivos planteados.

Finalmente, hablar de líneas de trabajo futuro que podrían aportar valor añadido al TFM realizado. La sección debería señalar las perspectivas de futuro que abre el trabajo desarrollado para el campo de estudio definido. En el fondo, debes justificar de qué modo puede emplearse la aportación que has desarrollado y en qué campos.

APÉNDICES

Apéndices, en caso de ser necesario.

REFERENCIAS

- [1] G. O. Young, "Synthetic structure of industrial plastics (Book style with paper title and editor)," in *Plastics*, 2nd ed. vol. 3, J. Peters, Ed. New York: McGraw-Hill, 1964, pp. 15–64.
- [2] W.-K. Chen, *Linear Networks and Systems* (Book style). Belmont, CA: Wadsworth, 1993, pp. 123–135.

TABLA I

UNITS FOR MAGNETIC PROPERTIES

Symbol	QUANTITY	Conversion from Gaussian and CGS EMU to SI ^a
Φ	magnetic flux	1 Mx → 10 ⁻⁸ Wb = 10 ⁻⁸ V·s
B	magnetic flux density, magnetic induction	1 G → 10 ⁻⁴ T = 10 ⁻⁴ Wb/m ²
H	magnetic field strength	1 Oe → 10 ³ /(4π) A/m
m	magnetic moment	1 erg/G = 1 emu → 10 ⁻³ A·m ² = 10 ⁻³ J/T
M	magnetization	1 erg/(G·cm ³) = 1 emu/cm ³ → 10 ³ A/m
4πM	magnetization	1 G → 10 ³ /(4π) A/m
σ	specific magnetization	1 erg/(G·g) = 1 emu/g → 1 A·m ² /kg
j	magnetic dipole moment	1 erg/G = 1 emu → 4π × 10 ⁻¹⁰ Wb·m
J	magnetic polarization	1 erg/(G·cm ³) = 1 emu/cm ³ → 4π × 10 ⁻⁴ T
χ, κ	susceptibility	1 → 4π
χ _p	mass susceptibility	1 cm ³ /g → 4π × 10 ⁻³ m ³ /kg
μ	permeability	1 → 4π × 10 ⁻⁷ H/m = 4π × 10 ⁻⁷ Wb/(A·m)
μ _r	relative permeability	μ → μ _r
w, W	energy density	1 erg/cm ³ → 10 ⁻¹ J/m ³
N, D	demagnetizing factor	1 → 1/(4π)

Vertical lines are optional in tables. Statements that serve as captions for the entire table do not need footnote letters.

^aGaussian units are the same as cgs emu for magnetostatics; Mx = maxwell, G = gauss, Oe = oersted; Wb = weber, V = volt, s = second, T = tesla, m = meter, A = ampere, J = joule, kg = kilogram, H = henry.

Title

Name and Surname of the Student

Universidad Internacional de la Rioja, Logroño (España)

Date



ABSTRACT

Breve resumen del trabajo realizado (extensión máxima: 150 palabras). Este resumen debe incluir el objetivo o propósito de la investigación, la metodología, los resultados y las conclusiones.

KEYWORDS

Tres a cinco palabras clave ordenadas alfabéticamente y separadas por comas.

I. INTRODUCTION

INTRODUCCIÓN en la que debes resumir de forma esquemática pero suficientemente clara lo esencial de cada una de las partes del trabajo.

La lectura de esta introducción ha de dar una primera idea clara de lo que se pretendía, las conclusiones a las que se ha llegado y del procedimiento seguido.

II. STATE OF THE ART

Estudio a fondo el dominio de aplicación, citando numerosas referencias.

Debe aportar un buen resumen del conocimiento que ya existe en el campo de los problemas habituales identificados.

Numerar las citas de forma consecutiva entre corchetes [1].

III. OBJECTIVES AND METHODOLOGY

Objetivo general, objetivos específicos y metodología de trabajo aplicada.

IV. CONTRIBUTION

Desarrollar la descripción de tu contribución.

XX
XX.

XX
XX
XX.

XX
XX.

XX
XX.

V. EVALUATION AND RESULTS

Descripción de la evaluación y los resultados obtenidos (Tipo 2. Desarrollo de Software).

XX
XX.

Evaluación 1

XX
XX.

XX
XX.

En la Figura 1...

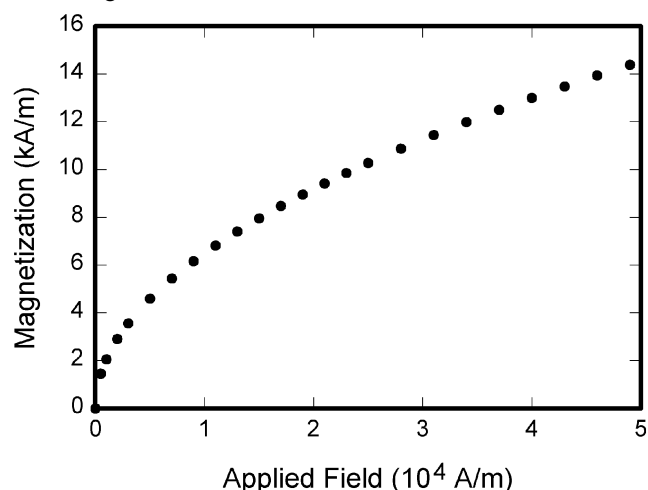


Fig. 1. Magnetization as a function of applied field. Note that "Fig." is abbreviated. There is a period after the figure number, followed by two spaces. It is good practice to explain the significance of the figure in the caption.

En la Tabla I ...

XX
XX
XX.

XX
XX
XX

Evaluación 2

XX
XX
XX
XXXXXXXXXXXXXXXXXXXX

VI. DISCUSSION

Tras la presentación objetiva de los resultados, querrás aportar una discusión de los mismos.

VII. CONCLUSION

Resumen de las contribuciones del trabajo, en el que relaciones las contribuciones y los resultados obtenidos con los objetivos que habías planteado para el trabajo, discutiendo hasta qué punto has conseguido resolver los objetivos planteados.

Finalmente, hablar de líneas de trabajo futuro que podrían aportar valor añadido al TFM realizado. La sección debería señalar las perspectivas de futuro que abre el trabajo desarrollado para el campo de estudio definido. En el fondo, debes justificar de qué modo puede emplearse la aportación que has desarrollado y en qué campos.

APPENDIX

Apéndices, en caso de ser necesario.

REFERENCES

- [3] G. O. Young, "Synthetic structure of industrial plastics (Book style with paper title and editor)," in *Plastics*, 2nd ed. vol. 3, J. Peters, Ed. New York: McGraw-Hill, 1964, pp. 15–64.
- [4] W.-K. Chen, *Linear Networks and Systems* (Book style). Belmont, CA: Wadsworth, 1993, pp. 123–135.

TABLA I

UNITS FOR MAGNETIC PROPERTIES

Symbol	QUANTITY	Conversion from Gaussian and CGS EMU to SI ^a
Φ	magnetic flux	1 Mx → 10 ⁻⁸ Wb = 10 ⁻⁸ V·s
B	magnetic flux density, magnetic induction	1 G → 10 ⁻⁴ T = 10 ⁻⁴ Wb/m ²
H	magnetic field strength	1 Oe → 10 ³ /(4π) A/m
m	magnetic moment	1 erg/G = 1 emu → 10 ⁻³ A·m ² = 10 ⁻³ J/T
M	magnetization	1 erg/(G·cm ³) = 1 emu/cm ³ → 10 ³ A/m
4πM	magnetization	1 G → 10 ³ /(4π) A/m
σ	specific magnetization	1 erg/(G·g) = 1 emu/g → 1 A·m ² /kg
j	magnetic dipole moment	1 erg/G = 1 emu → 4π × 10 ⁻¹⁰ Wb·m
J	magnetic polarization	1 erg/(G·cm ³) = 1 emu/cm ³ → 4π × 10 ⁻⁴ T
χ, κ	susceptibility	1 → 4π
χ _ρ	mass susceptibility	1 cm ³ /g → 4π × 10 ⁻³ m ³ /kg
μ	permeability	1 → 4π × 10 ⁻⁷ H/m = 4π × 10 ⁻⁷ Wb/(A·m)
μ _r	relative permeability	μ → μ _r
w, W	energy density	1 erg/cm ³ → 10 ⁻¹ J/m ³
N, D	demagnetizing factor	1 → 1/(4π)

Vertical lines are optional in tables. Statements that serve as captions for the entire table do not need footnote letters.

^aGaussian units are the same as cgs emu for magnetostatics; Mx = maxwell, G = gauss, Oe = oersted; Wb = weber, V = volt, s = second, T = tesla, m = meter, A = ampere, J = joule, kg = kilogram, H = henry.