



Sistema de videovigilancia basado en planificación aleatoria y detección de intrusos para áreas restringidas en entornos externos usando drones de ala rotativa.

Albuja Cadena, Marissa Carolina y Mosquera Romero, Cristian Fabricio

Departamento de Eléctrica, Electrónica y Telecomunicaciones

Carrera de Ingeniería en Electrónica, Automatización y Control

Trabajo de titulación, previo a la obtención del título de Ingeniero en Electrónica,
Automatización y Control

Dr. Aguilar Castillo, Wilbert Geovanny PhD.

19 de mayo 2023

15/5/23, 20:30

Tesis Albuja-Mosquera

Informe de originalidad

NOMBRE DEL CURSO

Tesis

NOMBRE DEL ALUMNO

CRISTIAN FABRICIO MOSQUERA ROMERO

**NOMBRE DEL ARCHIVO**

TESIS_ESCRITOFINAL

SE HA CREADO EL INFORME

17 may 2023

Resumen

Fragmentos marcados	5	0,7 %
Fragmentos citados o entrecuillados	1	0,1 %

Coincidencias de la Web

importacionesamexico.com.mx	1	0,2 %
ua.es	1	0,1 %
eluniverso.com	1	0,1 %
facebook.com	1	0,1 %
aunar.edu.co	1	0,1 %
gbif.es	1	0,1 %

1 de 6 fragmentos

<https://classroom.google.com/g/gp/NTQyOTEwNDI4NTcy/NjEwMzA4MjlyNzIx?hl=es&u=NjM3NTQwNTgyNjNa&f>

1/3



Departamento de Eléctrica, Electrónica y Telecomunicaciones

Carrera de Ingeniería en Electrónica, Automatización y Control

Certificación

Certifico que el trabajo de titulación: **"Sistema de videovigilancia basado en planificación aleatoria y detección de intrusos para áreas restringidas en entornos externos usando drones de ala rotativa"** fue realizado por los señores **Albuja Cadena, Marissa Carolina y Mosquera Romero, Cristian Fabricio;** el mismo que cumple con los requisitos legales, teóricos, científicos, técnicos y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, además fue revisado y analizado en su totalidad por la herramienta de prevención y/o verificación de similitud de contenidos; razón por la cual me permito acreditar y autorizar para que se lo sustente públicamente.

Sangolquí, 18 de mayo del 2023



WILBERT GEOVANNY
AGUILAR CASTILLO

Aguilar Castillo, Wilbert Geovanny

C.C. 0703841696



Departamento de Eléctrica, Electrónica y Telecomunicaciones

Carrera de Ingeniería en Electrónica, Automatización y Control

Responsabilidad de Autoría

Nosotros, **Albuja Cadena, Marissa Carolina** y **Mosquera Romero, Cristian Fabricio**, con cédulas de ciudadanía n°1716685951 y n°1725120123, declaramos que el contenido, ideas y criterios del trabajo de titulación: **Sistema de videovigilancia basado en planificación aleatoria y detección de intrusos para áreas restringidas en entornos externos usando drones de ala rotativa** es de nuestra autoría y responsabilidad, cumpliendo con los requisitos legales, teóricos, científicos, técnicos, y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, respetando los derechos intelectuales de terceros y referenciando las citas bibliográficas.

Sangolquí, 11 de mayo del 2023

Marissa Carolina Albuja Cadena

Albuja Cadena, Marissa Carolina

C.C.: 1716685951

Cristian Fabricio Mosquera

Mosquera Romero, Cristian Fabricio

C.C.: 1725120123



Departamento de Eléctrica, Electrónica y Telecomunicaciones

Carrera de Ingeniería en Electrónica, Automatización y Control

Autorización de Publicación

Nosotros **Albuja Cadena, Marissa Carolina** y **Mosquera Romero, Cristian Fabricio**, con cédulas de ciudadanía n°1716685951 y n°1725120123, autorizamos a la Universidad de las Fuerzas Armadas ESPE publicar el trabajo de titulación: **Sistema de videovigilancia basado en planificación aleatoria y detección de intrusos para áreas restringidas en entornos externos usando drones de ala rotativa** en el Repositorio Institucional, cuyo contenido, ideas y criterios son de nuestra responsabilidad.

Sangolquí, 11 de mayo del 2023

Marissa Carolina Albuja Cadena

Cadena Albuja, Marissa Carolina

C.C.: 1716685951

Cristian J. Mosquera

Mosquera Romero, Cristian Fabricio

C.C.: 1725120123

Dedicatoria

Quiero dedicar este trabajo a mis padres Danilo y María Cristina, quienes me han acompañado durante todas las etapas de mi vida, y con su consejo y sabiduría me han sabido guiar. Gracias a su amor incondicional he sabido encontrar una fortaleza para superar todos los obstáculos que he tenido a lo largo de formación académica. Su alegría y orgullo son el pago más valioso para mí.

También a mi hermana que con su compañía y amistad me ha levantado en los malos momentos, espero que junto a Dios y a mis padres puedas alcanzar todas tus metas. A mi abuelito que con su experiencia ha sido fuente de aprendizaje.

A Cristian, Josselyn y Luis por su gran apoyo a lo largo de esta etapa académica en los buenos y más aún en los malos momentos, su presencia siempre significará mucho para mí. Aprendí de su constancia y gran corazón.

Finalmente, de manera especial quiero dedicar este logro a mi mamá Luchita sus enseñanzas, amor incondicional y valores siempre estarán en mi corazón. Quererte fue fácil, pero olvidarte imposible.

Marissa Carolina Albuja Cadena

Dedicatoria

Quiero dedicar este trabajo a mis padres y hermanos, quienes han sido el apoyo fundamental en mi vida, ya que cada uno ha permitido transformar a la persona que está cumpliendo uno de sus sueños, a mi madre que con su amor me ha mostrado la nobleza en una persona, a mi padre que me ha enseñado la constancia y el trabajo arduo, mis hermanos por siempre estar a pesar de los roces. Gracias por la confianza y amor brindado, he podido encontrar la resiliencia necesaria para superar cada momento difícil, por eso todo el trabajo es para ustedes en honor a permanecer siempre junto a mí.

También, quiero dedicar este logro a todos mis amigos: CHB, Cazaticos, Cosa, Dani, Jhordy y Krly que han estado en lo largo del camino siendo apoyo en las risas, alegrías, tristezas, llantos, con quienes he compartido gratos momentos, por cada vivencia que se quedan grabadas en la memoria.

A Marissa, quien se convirtió en un apoyo fundamental para no decaer en los malos momentos y por estar en los buenos también, enseñándome la ternura de un corazón agradable a Dios.

A Beike quien apareció en un momento muy difícil en mi vida para rescatarme y enseñarme que no es necesario hablar para demostrar el amor verdadero. Finalmente, Dios por guiarme y ser la luz en mi camino.

Cristian Fabricio Mosquera Romero

Agradecimiento

Primero quiero agradecer a Dios porque me enseñó que con valentía y esfuerzo todo es posible, y porque su fidelidad sigue persiguiéndome. A mi papá porque me enseñó que ninguna prueba es sobrehumana porque tan fiel es Dios, que juntamente con la prueba ha mandado la salida. A mi mamá por ser esa salida y no dejar que desmaye en el camino, siempre estaremos juntas.

A toda mi familia por el apoyo incondicional, por el sacrificio que han hecho por darme siempre lo mejor. Al Dr. Aguilar por su paciencia, dirección y apoyo en este trabajo. Al Departamento de Eléctrica y Electrónica por ser fuente de aprendizaje y respaldo académico.

Marissa Carolina Albuja Cadena

Primero agradezco a Dios por convertirme y guiarme a la persona de hoy en día, mi familia y amigos con quienes comparto día a día mis vivencias. A todas las personas que hicieron posible la realización de este trabajo para culminar una etapa, sus contribuciones fueron necesarias para cumplir este objetivo.

Agradecimiento a nuestro tutor, el Dr. Wilbert Aguilar, cuya orientación, dedicación, conocimientos y perspectivas enriquecieron enormemente este proyecto.

Cristian Fabricio Mosquera Romero

Contenido

Resumen.....	15
Abstract.....	16
Capítulo I. Preliminares	17
Antecedentes.....	17
Justificación e Importancia.....	19
Alcance.....	22
Objetivos.....	23
Objetivo General	23
Objetivos Específicos	23
Descripción del proyecto.....	24
Capítulo II. Fundamentación Teórica.....	26
Vehículos aéreos no tripulados (UAV)	26
Navegación Gruesa	27
Sistema Mundial Geodésico WGS84.....	29
Dinámica del Movimiento	29
Navegación Fina.....	31
Control Servovisual	31
Detección y Seguimiento Visual	37
Capítulo III. Desarrollo del Sistema	48
Introducción	48

	10
Hardware del Sistema.....	48
Parrot Bebop 2 – UAV	49
Estación a Tierra	51
Software del Sistema	52
OpenCV	53
Entorno de Desarrollo: ROS	54
Sistema de Detección: YOLO.....	56
PyTorch.....	57
Comunicación del Sistema.....	59
Ejecución.....	60
Tópicos principales.....	60
Detección de Intrusos	62
Entrenamiento del Modelo.....	63
Datos de Entrenamiento.....	63
Etiquetado de personas.....	64
Entrenamiento	64
Resultados	65
Capítulo IV. Navegación Autónoma en Entornos Externos.....	67
Descripción del Entorno	67
Marcadores de referencia	68
Estimación del Modelo del micro-UAV.....	70

Modelo de la Planta para el Movimiento en Roll	71
Modelo de la Planta para el Movimiento en Pitch	73
Diseño del Controlador	75
Implementación de Controlador para Posicionamiento del micro-UAV y Acción de Aterrizaje	77
Planificación de Trayectoria	79
Navegación Autónoma por GPS	80
Trayectoria aleatoria de ruta GPS	82
Capítulo V. Pruebas y Resultados	86
Trayectoria de navegación por GPS	86
Detección de Intrusos (Navegación Fina)	89
Aterrizaje Automático (Navegación Fina)	92
Conclusiones y Recomendaciones	98
Conclusiones	98
Recomendaciones	99
Bibliografía	101

Índice de Tablas

Tabla 1	Cifras de homicidios por mes en Ecuador	19
Tabla 2	Incremento de muertes violentas.....	20
Tabla 3	Tipos de transformaciones con sus características principales.....	47
Tabla 4	Especificaciones del UAV – parrot bebop 2.....	49
Tabla 5	Especificaciones del computador portátil.....	51
Tabla 6	Especificaciones de la antena de WiFi	52
Tabla 7	Especificaciones del computador portátil para usar Ubuntu 20.04 LTS	52
Tabla 8	Tópicos publicados.....	61
Tabla 9	Tópicos suscritos.....	61
Tabla 10	RMSE de la trayectoria realizada por el micro-UAV.....	88
Tabla 11	Sensibilidad y especificidad en la detección de intrusos.....	91
Tabla 12	Datos obtenidos del aterrizaje automático del sistema	95
Tabla 13	Comparación del sistema de aterrizaje.....	96

Índice de Figuras

Figura 1	Diagrama de las etapas del proyecto.....	23
Figura 2	Aplicaciones de los micro-UAV	27
Figura 3	Descripción de los grados de libertad del cuadricóptero	30
Figura 4	Movimiento de motores.....	31
Figura 5	Sistema de referencias y fuerzas.....	34
Figura 6	Ángulos de Euler	34
Figura 7	Prueba de detección de esquinas en 12 puntos en un parche de imagen.....	40
Figura 8	Diagrama de flujo de coincidencia de imágenes basado en ORB	40
Figura 9	Comparación de descriptores un ciclo de detección-descripción.	42
Figura 10	Modificación en las imágenes por el cambio de dominio.....	43
Figura 11	Transformaciones geométricas 3D	44
Figura 12	Matriz de transformación	45
Figura 13	Transformación afín.....	45
Figura 14	Diagrama y componentes del proyecto	48
Figura 15	Dimensiones bebop 2	49
Figura 16	Visión por computadora	54
Figura 17	Elementos de la red de ROS	56
Figura 18	Sistema de Detección YOLO	57
Figura 19	Comparación de la velocidad de seis modelos de aprendizajes	58
Figura 20	Flujo de trabajo de PyTorch.....	59
Figura 21	Sistema de Detección YOLO	62
Figura 22	Imágenes base para la recolección de datos	63
Figura 23	Etiquetado de imágenes en makesense.ai.....	64
Figura 24	Configuración archivo “.yaml” de datos a entrenar y clase de datos	65

Figura 25	Configuración para entrenamiento del modelo.....	65
Figura 26	Resultados de entrenamiento del modelo para la detección de intrusos	66
Figura 27	Estadio de la Universidad de las Fuerzas Armadas, entorno utilizado	67
Figura 28	Zona de vuelo y aterrizaje del micro-UAV	68
Figura 29	MarcadorArUco.....	68
Figura 30	Diagrama de flujo para la detección de marcador	69
Figura 31	Entrada del sistema para estimación de modelo del movimiento en Roll	71
Figura 32	Salida del sistema para estimación de modelo del movimiento en Roll.....	71
Figura 33	Identificación de la planta en Roll.....	73
Figura 34	Entrada del sistema para estimación de modelo del movimiento en Pitch ...	73
Figura 35	Salida del sistema para estimación de modelo del movimiento en Pitch	74
Figura 36	Identificación de la planta en Pitch.....	75
Figura 37	Respuesta al escalón del controlador PID para Roll	76
Figura 38	Respuesta al escalón del controlador PID para Pitch.....	77
Figura 39	Error de posición del marcador respecto a la imagen del micro-UAV.....	78
Figura 40	Diagrama de flujo para el aterrizaje con respecto al marcador.....	79
Figura 41	Diagrama de flujo para la navegación autónoma mediante GPS	81
Figura 42	Rango de detección.....	82
Figura 43	Zona de vuelo	83
Figura 44	Diagrama de flujo para la trayectoria de vigilancia del micro-UAV	85
Figura 45	Trayectoria de navegación aleatoria por GPS.....	87
Figura 46	Sistema de detección de intrusos.	91
Figura 47	Tomas aterrizaje automático micro-UAV	93
Figura 48	Identificadores de las pruebas de aterrizaje automático del micro-UAV	94
Figura 49	Dispersión del error del aterrizaje automático del micro-UAV	97

Resumen

Este documento presenta el diseño e implementación de un sistema de videovigilancia basado en planificación aleatoria y detección de intrusos para áreas restringidas en entornos externos haciendo uso del micro-UAV Parrot Bebop 2 y la cámara incorporada que posee. Para la navegación autónoma, se dividió en dos partes, la navegación gruesa mediante un controlador P en función del cálculo progresivo del ángulo de rumbo y la distancia entre dos coordenadas GPS, a su vez para la planificación aleatoria se ha implementado un algoritmo que realiza las distintas rutas posibles ha desempeñar por el micro-UAV en función a las coordenadas establecidas del entorno externo. La navegación fina en la cual se realiza un control servo visual implementando un controlador PID en roll y pitch, fueron diseñados posterior a obtener el modelo matemático en la estimación del movimiento por visión, en esta segunda parte denominada navegación fina se hizo uso de un marcador (código ArUco) que será el objetivo en el que se realizará el aterrizaje autónomo. Para la detección de intrusos (personas), se utilizó el sistema YOLOv8 por la capacidad de tener una respuesta rápida y precisa al realizar dicha tarea. El rendimiento del sistema se evalúa en un escenario externo sin obstáculos, además de variar los factores externos para conseguir un desempeño robusto del sistema.

Palabras Clave: Planificación aleatoria, detección de intrusos, navegación autónoma y control servo visual.

Abstract

This document presents the design and implementation of a video monitoring system based on random planning and intrusion detection for restricted areas in outdoor environments using the Parrot Bebop 2 micro-UAV and its built-in camera. For autonomous navigation, it was divided into two parts, the coarse navigation using a P-controller based on the progressive calculation of the heading angle and the distance between two GPS coordinates, in turn for the random planning has been implemented an algorithm that performs the different possible routes to be performed by the micro-UAV based on the coordinates set in the external environment. The fine navigation in which a visual servo control is performed by implementing a PID controller in roll and pitch, were designed after obtaining the mathematical model in the estimation of the movement by vision, in this second part called fine navigation was made use of a marker (ArUco code) that will be the target on which the autonomous landing will be performed. For intruder detection (people), the YOLOv8 system was used for its ability to have a fast and accurate response when performing this task. The performance of the system is evaluated in an unobstructed external scenario, in addition to varying the external factors to achieve robust system performance.

Keywords: Random planning, intrusion detection, autonomous navigation and visual servo control.

Capítulo I. Preliminares

Antecedentes

Para el hombre una de las herramientas más importantes ha sido el desarrollo de algoritmos que permitan impulsar nuevos cambios para generar un mejor estilo de vida. Como por ejemplo la navegación autónoma que permite apoyar en diferentes labores de logística y hacer más eficiente la operación que se realice, otro ejemplo de gran ayuda es la inteligencia artificial que permite automatizar procesos, reducir el error humano, análisis de datos, entre otras.

La navegación autónoma es la capacidad de un objeto para trasladarse de un lugar a otro evitando los obstáculos que se presenten a lo largo de dicha trayectoria, en este caso el objeto será un vehículo no aéreo no tripulado, a lo largo del desarrollo de tecnología estos vehículos no tripulados (UAV) han sido capaces de realizar servicios a pueblos aislados, mediante sus cámaras termográficas dan a conocer el estado de un cultivo (Beltrán, 2017), inspección de alguna infraestructura mediante un control servo visual como por ejemplo la seguridad en complejos o centros comerciales, etc. (Montoya & Briones, 2019). El Centro de Investigación y Desarrollo de la Fuerza Aérea (CIDFAE); en colaboración con el Ministerio de Defensa y la Secretaría de Estado de Educación Científica y Tecnológica (SENESCYT), desarrollaron el proyecto Detección, Observación, Comunicación y Reconocimiento (DOCR), el proyecto incluye Vehículos Aéreos Autónomos No Tripulados (UAVs), Relés de Comunicación, EMCT Estaciones y optoelectrónica EO/IR El Charge con el propósito de permitir misiones de vigilancia, reconocimiento y transmisión de datos en tiempo real. (Montoya & Briones, 2019)

Pero también se puede clasificar a estos sistemas en: UAVs de ala fija y UAVs de ala rotatoria. (Cuerno Rejado, García Hernández, & Sánchez Carmona, 2016)

- UAVs de ala fija: La particularidad de este tipo de vehículo es el despegue, siempre horizontal. Otra característica peculiar es la posición del motor ya que puede variar de una plataforma a otra:
 - Hélice tractora, delante de la aeronave.
 - Hélice empujadora, extremo trasero de la aeronave.
 - Hélice tractora, bajo el ala.
- UAVs de ala rotatoria: Su principal característica es el tipo de vuelo, vuelan aprovechando el soporte generado por las alas rotatorias o por las palas del rotor, quienes son montadas generalmente en un solo mástil y giran en torno al rotor. El rotor es quien dará la clasificación en este caso: (Cuerno Rejado et al., 2016)
 - Tipo helicóptero: pueden despegar verticalmente, volar hacia adelante, atrás o laterales manteniendo su posición. Pueden usar un solo motor para todo el vuelo.
 - Tipo ciclogiro: Tienen mayor eficiencia y velocidad, avanzados en ruido y vibración.
 - Tipo autogiro: Usa un rotor sin motor, impulsado por autorrotación; mientras que para el empuje se lo realiza mediante una hélice con motor.
 - Tipo girodino: Incluyen uno o más motores para garantizar el empuje hacia adelante durante el vuelo de crucero. Para el despegue y el aterrizaje, solamente el motor principal.

Para planificar los caminos de estos vehículos se utilizan varios métodos (Salgado, Tierra, Sandoval, & Aguilar, 2017).

Al ser un algoritmo es importante contar con una plataforma, el sistema operativo robótico ROS es una plataforma de código abierto que cuenta con una colección de

herramientas y librerías. Su arquitectura distribuida permite convertir procesos complejos en simples.(Quigley, Gerkey, & Smart, Programming Robots with ROS, 2007)

Justificación e Importancia

Ecuador es el tercer país con mayor incidencia de robos y asaltos en los primeros cuatro meses de este año, según una encuesta realizada por CID *Gallup Consulting* entre el 10 y el 19 de mayo de 2022. Según datos de las dependencias gubernamentales, en todo el 2021 habrá 25.389 robos registrados en todo el país, mientras que de enero a mayo de 2022 habrá 12.548 robos registrados, una disminución de casi la mitad en cinco meses. Solo en mayo de este año hubo 2.562 incidentes a nivel nacional, de los cuales 829 ocurrieron en Guayaquil, frente a 585 en mayo de 2021. A nivel nacional, en cambio, hubo 2.496 homicidios dolosos en 2021 y 2.116 hasta junio de 2022.(Redacción, 2022)

Tabla 1

Cifras de homicidios por mes en Ecuador

MES	2022
ENERO	315
FEBRERO	311
MARZO	344
ABRIL	366
MAYO	426

JUNIO

354

Nota. Esta tabla muestra las cifras de homicidios a lo largo del año 2022. Tomado de Así están las cifras de la inseguridad en el primer semestre del 2022 comparadas con los tres años previos, por El Universo, 2022.

El ministerio del interior junto a la Policía Nacional registró 145 atentados en los 8 primeros meses de este año, alcanzando un récord en la historia del país. Cerca de la mitad de estos atentados han ocurrido en Guayaquil (72) (Oropeza, 2022).

Según la Policía Nacional, Ecuador alcanzó la tasa más alta de muertes violentas, teniendo 15.48 muertes violentas por cada 100.000 habitantes a tan solo cuatro meses de finalizar el año 2022 (Mella, 2022). A continuación, se muestra una tabla con las provincias con las tasas más altas de este tipo de muertes.

Tabla 2

Incremento de muertes violentas

PROVINCIAS	MUERTES EN 2021	MUERTES EN 2022	TASA
ESMERALDAS	83	322	48.79
GUAYAS	408	939	29.1
EL ORO	97	190	25.94
LOS RÍOS	94	239	25.34
SUCUMBÍOS	19	53	22

MANABÍ	144	294	18.23
SANTO DOMINGO	34	84	17.71

Nota. Esta tabla muestra el número de muertes en 2021 y 2022, Tomado de *Muertes Violentas* por Policía Nacional, 2022.

En el ámbito de seguridad es importante recalcar que a principios de 2007, la Agencia Nacional del Espacio Acuático (DIRNEA) planteó un proyecto de control y vigilancia para reducir el uso inapropiado y trasiego ilegal de combustible marino, bajo este supuesto se estableció la importancia de un sistema integrado de vigilancia marítima, uno de los componentes es un vehículo aéreo no tripulado sistemas *Air Vehicle* (UAV)(Ramírez & Aguilar, 2019); los esenciales para las operaciones de exploración aire-mar en áreas cercanas a las costas continentales; así, a partir de 2009, cuatro UAV de búsqueda y dos UAV Heron que están en servicio con la Armada apoyaron la lucha contra el contrabando marítimo, narcotráfico, piratería y crimen organizado. El sistema consta de una aeronave, estación de control, sistema de comunicación por enlace de datos y elementos de apoyo.(Montoya & Vásquez, 2019) (Sanchez & Aguilar, 2019)

El uso de UAV permite evitar riesgos contra nuevas amenazas como, tráfico ilícito, contrabando, inmigración ilegal realizando las siguientes acciones: vigilancia, reconocimiento, adquisición de objetivos, entre otras. (Douhet, 1921). Explorando otros posibles usos según (Jhon, 2021), se tienen:

- Lanzamiento automático de un UAV para rescatar un náufrago.
- Seguridad en bases, puertos y fondeaderos: patrullas avanzadas perimetrales con rutas de tránsito preestablecidas y sistema de inteligencia artificial para reconocimiento de posibles amenazas.

Alcance

En el presente proyecto se plantean 5 etapas que tienen como propósito desarrollar e implementar un Sistema de videovigilancia basado en planificación aleatoria y detección de intrusos para áreas restringidas en entornos externos usando drones de ala rotativa.

En la primera etapa del proyecto se realizará la búsqueda y análisis del estado del arte, de las distintas tecnologías que se implementarán para la comunicación, navegación autónoma, planificación de trayectoria y detección de intrusos.

La segunda etapa consiste en la instalación del sistema operativo el cual estará basado en ROS (*RoboticOperativeSystem*), y se realizará la comunicación la cual se establecer en el protocolo WIFI que permitirá interactuar la base terrestre con el UAV (Vehículo aéreo no tripulado), que permitan realizar las acciones de control correspondientes a despegar, aterrizar, navegación, activación de la cámara.

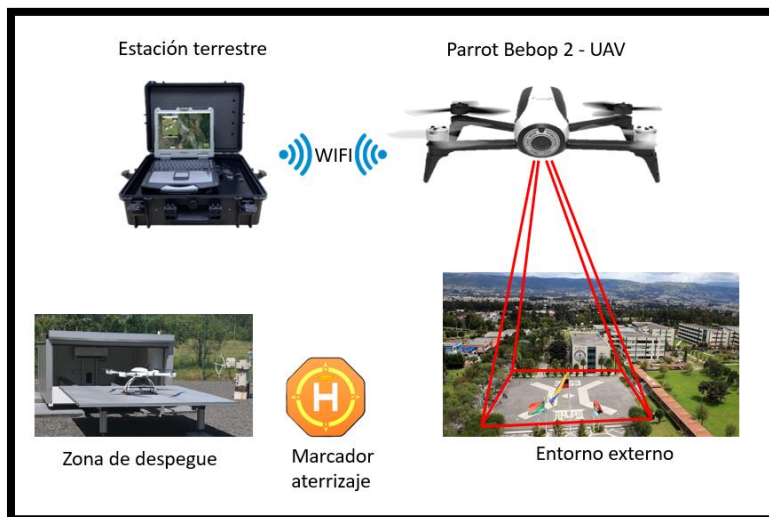
La tercera etapa consiste en la navegación autónoma del UAV la cual está conformada por dos partes la navegación gruesa que se realizará mediante GPS, y la navegación fina que se lo hará mediante control servo visual.

La cuarta etapa consiste en desarrollar un sistema de planificación de trayectoria para un área establecida en un entorno externo, el cual realizará o establecerá la trayectoria de manera aleatoria, haciendo uso de algoritmos existentes.

La quinta etapa consiste en la detección de intrusos lo cual se lo realizará por medio de visión por computadora haciendo uso de la cámara monocular que se encuentra integrada en el UAV Parrot Bebop 2.

Figura 1

Diagrama de las etapas del proyecto



Nota. La figura muestra las etapas de este proyecto, desde la zona de despegue hasta el entorno que será vigilado.

Objetivos

Objetivo General

Desarrollar un sistema de videovigilancia basado en planificación aleatoria y detección de intrusos para áreas restringidas en entornos externos usando drones de ala rotativa.

Objetivos Específicos

- Desarrollar un análisis del estado del arte, tecnologías implementadas y técnicas para la navegación autónoma de vehículos aéreos no tripulados en entornos externos.
- Realizar la comunicación entre la estación terrestre y el vehículo aéreo no tripulado que permita la navegación de este.
- Diseñar un sistema de planificación que permita determinar la trayectoria de navegación de vehículos aéreos no tripulados.

- Desarrollar un sistema para la detección de intrusos que permita supervisar y brindar seguridad en el área determinada.
- Implementar el diseño del sistema en un vehículo aéreo no tripulado para determinar los resultados y mejoras a futuro.

Descripción del proyecto

- a) Estudio del estado del arte:** Realizar la investigación de conceptos teóricos y técnicos actualizados referentes al proyecto a desarrollarse; vehículos aéreos no tripulados, sistema de comunicación, planificación de trayectoria y control.
- b) Establecer comunicación entre comunicación terrestre y vehículo aéreo no tripulado:** Se realiza la comunicación para poder desarrollar el algoritmo que permita realizar las acciones correspondientes del vehículo aéreo no tripulado.
- c) Diseño del control servo visual para la navegación autónoma de vehículo aéreo no tripulado:** Se realiza el diseño del control servo visual que realizará la navegación autónoma del vehículo aéreo no tripulado, todo esto mediante ROS.
- d) Diseño de sistema de planificación para la trayectoria de vuelo basado en la percepción de puntos de interés:** Se realiza el diseño del sistema de planificación de la trayectoria aleatoria en base a los algoritmos teniendo en cuenta los puntos de interés.
- e) Diseño de sistemas de intrusos:** Se realiza el diseño del sistema de intrusos mediante visión por computadora haciendo uso de la cámara monocular que se encuentra integrada en el UAV Parrot Bebop 2.
- f) Pruebas y Resultados:** En esta etapa se realizarán varios experimentos, mediante los cuales se pretende analizar el funcionamiento de todo el sistema con diferentes usuarios, además, se realizará pruebas para comparar el sistema desarrollado frente a los sistemas tradicionales para controlar un miro-UAV con

el fin de medir la velocidad de respuesta del sistema, así como la precisión y el grado de utilidad.

g) Conclusiones y Recomendaciones

En base a las actividades realizadas se redactan las conclusiones y recomendaciones del presente proyecto, además, se mencionan algunos de los futuros trabajos de investigación dentro de esta área.

Capítulo II. Fundamentación Teórica

En este capítulo se presentan conceptos básicos que son importantes revisar para el desarrollo del sistema de videovigilancia basado en planificación aleatoria y detección de intrusos para áreas restringidas en entornos externos usando drones de ala rotativa. Este capítulo se divide en dos secciones, la primera es la navegación gruesa en donde se trata el sistema de navegación y la dinámica de movimiento, y en la segunda parte se encuentra la navegación fina donde se tiene el control servo visual, detección y seguimiento visual.

Vehículos aéreos no tripulados (UAV)

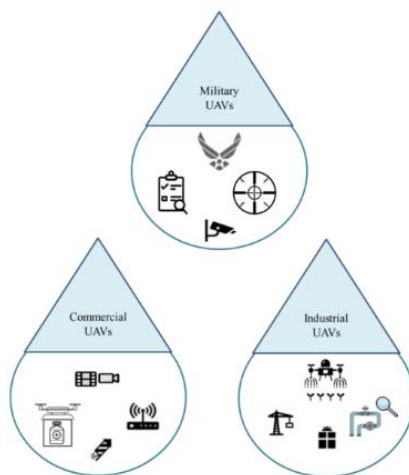
Un vehículo aéreo no tripulado (UAV) es un sistema aéreo autónomo capaz de realizar una misión determinada sin necesidad de tener una tripulación a bordo (Orbea, Moposita, Aguilar, Paredes, León, & Jara-Olmedo, 2017), aunque esto no quiere decir que no tenga un piloto u operador de misión que lo controla desde tierra (Sánchez, 2017). Este tipo de vehículo complejo contiene varios subsistemas como por ejemplo plataformas aéreas y terrestres, satélites y comunicaciones entre el vehículo aéreo no tripulado y lo mencionado anteriormente (Cuerno, García, Sanchez, Carrio, & Sanchez, 2016). Su historia remonta a la primera guerra mundial cuando en Europa se probaron varios diseños sin personas a bordo los cuales se consideraron los primeros UAVs, en 1920 el ejército británico implementó una aeronave de tipo monoplano controlado por radio capaz de transportar únicamente cargamento militar; durante la segunda guerra mundial, Alemania desarrolló un misil crucero que tenía equipado un motor pulsorreactor, sistema barométrico y un anemómetro. El primer UAV fue desarrollado por la compañía RADIOPLANE, de tipo ala rotatoria, su objetivo fue atacar submarinos enemigos que estaban lejos del alcance de otras armas (Martinez, 2020). Finalmente, a partir de los años 70 y 80 del siglo XX, el mundo sería testigo de diferentes diseños que eran capaces de realizar operaciones de vigilancia y reconocimiento, tanto de

corto y largo alcance como también a gran altitud (Cuerno Rejado, García Hernández, & Sánchez Carmona, 2016).

Cabe recalcar que los micro-UAV o comúnmente llamados drones se han convertido en una clave para las redes de tipo 5G, ya sea como estación base o como sistema móvil de retransmisión, la Figura 2 muestra varias aplicaciones de los micro-UAV como una tecnología en comunicación y detección, han despertado intereses en grandes multinacionales como Walmart, DHL, Google y Amazon por su aumento de las compras en línea (Hassnain, Hamood, Khan, Amjad, & Zywolek, 2022).

Figura 2

Aplicaciones de los micro-UAV



Nota. La figura muestra varias aplicaciones comerciales, militares e industriales. Tomado de *A Comprehensive Review of Micro UAV Charging Techniques*, por Hassnain, S., 2022, *Micromachines*, XIII(977), 1-30.

Navegación Gruesa

Este proceso es capaz de estimar la orientación y posición de un objeto con respecto a una referencia, esto es posible gracias a la información que proporcionan sensores, cámara,

GPS, etc. (Merizalde, Aguilar, & Calderón, 2020) o IMU(Segarra, Caballeros, Aguilar, Samà, & Rodríguez-Martín, Orientation Estimation Using Filter-Based Inertial Data Fusion for Posture Recognition, 2019)), cámara(Calderón, Aguilar, & Merizalde, 2020), etc. Existe varios métodos para realizar este proceso, pero todo depende del tipo de información que se use, por ejemplo, puntos de referencia(Tituaña, Amaguaña, Calderón, Merizalde, & Aguilar, 2020), número de cámaras(Manosalvas, Guillén, Collaguazo, & Aguilar, 2018),localización y mapeo simultáneo(Rodríguez, Álvarez, Sandoval, Quisaguano, Limaico, & Aguilar, 2017), etc.

En esta tesis se revisarán dos técnicas de estimación, odometría visual (Galarza, Pérez, Serrano, Tapia, & Aguilar, 2018) y obtención de la posición relativa del UAV con respecto a un objeto (realimentación con lazos de control)(Zurita, Aguilar, & Enríquez, 2019).

El GPS, también conocido como sistema de navegación por satélite (SNS), se considera una de las mejores formas de facilitar una posición 3D para vehículos terrestres no tripulados, vehículos submarinos autónomos y drones. Los sistemas de navegación basados en GPS tienen las ventajas de cobertura global, precisión y bajo costo. Debido a su capacidad para recibir señales GPS en cualquier parte de la Tierra (Antichán, Morán, & Núñez, 2009). Los receptores GPS están ampliamente disponibles y pueden proporcionar una precisión de hasta varios metros bajo cielo abierto. Sin embargo, el GPS tiene la desventaja de ser vulnerable a las interferencias y depender de las señales de los satélites. Generalmente, los dispositivos GPS se utilizan para la navegación en exteriores, mientras que los sensores basados en la visión se utilizan en interiores o en sistemas GPS (Muhammad , Muhammad, & Sangman, 2023).

El UAV utilizará el GPS como orientación espacial porque permite acceder a los datos de posicionamiento global que serán gestionados desde el ordenador (Merizalde, Aguilar, & Calderón, 2020).

Sistema Mundial Geodésico WGS84

La tendencia actual del uso de GPS trae a la par el uso constante de sistemas de referencia geocéntricos asociados a elipsoides globales como es el caso del sistema WGS84 (Leiva, 2008).

El sistema mundial de coordenadas geográficas WGS84 es un sistema que tiene la capacidad de localizar cualquier punto de la Tierra (sin necesidad de referencias) por medio de tres unidades dadas. Se estima un error de cálculo menor a 2 cm. por lo que es en la que se basa el Sistema de Posicionamiento Global (GPS) (Guamba, 2010). Los parámetros de este sistema son:

- Origen: Centro de masa de la Tierra
- Eje X: intersección del meridiano origen definido en 1984 y el plano del Ecuador.
- Eje Y: eje perpendicular a X y Z, que coinciden en el origen.
- Eje Z: dirección del polo de referencia del IERS.

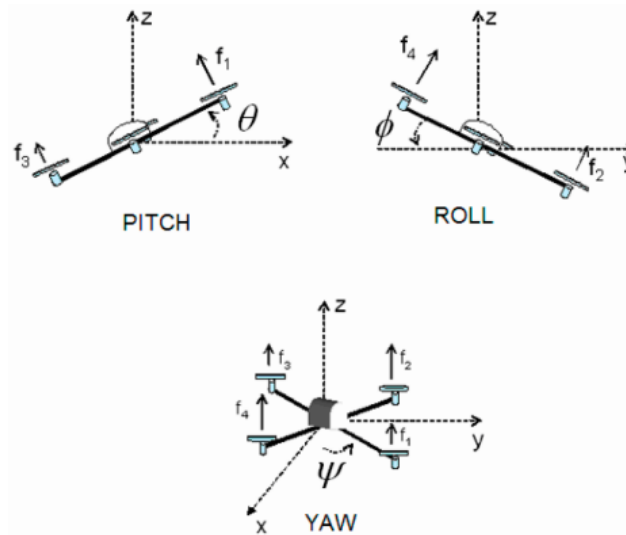
Dinámica del Movimiento

Este tipo de vehículos aéreos se manejan en un espacio tridimensional por lo que se tienen tres ángulos de orientación y tres coordenadas de piso con (seis grados) (Rodríguez, 2016). Para este tipo de estudio se usan los ángulos de Tait-Bryan mostrados en la Figura 3, detalladas a continuación:

- Cabeceo (ϕ , roll): Este lleva a cabo la inclinación del eje del UAV.
- Alabeo (pitch, θ): Rotación respecto al morro-cola del UAV.
- Guiñada (yaw, γ): Rotación intrínseca alrededor del eje vertical perpendicular del UAV.

Figura 3

Descripción de los grados de libertad del cuadricóptero

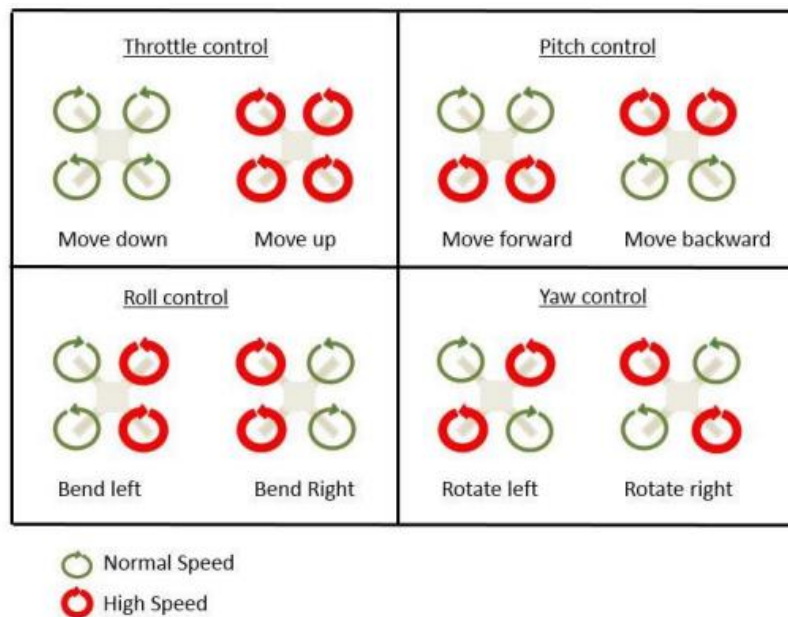


Nota. Ángulos de rotación: Pitch, Roll y Yaw. Tomado de *Desarrollo del control visual de un UAV para el seguimiento de líneas de cultivo*, por Rodríguez, J., 2016, Universidad Politécnica de Madrid.

Partiendo de la definición de dichos ángulos, se enfocará el movimiento de las hélices en disposición cruz para que no exista ningún choque (posición opuesta), haciendo que giren al mismo lado, teniendo así dos hélices girando en sentido horario y las dos restantes girando en sentido anti horario. En la Figura 4 se observa un resumen de los movimientos del UAV según la potencia suministrada a los motores, esta potencia es directamente proporcional al movimiento concreto del UAV y la altura que este logre, así si se suministra potencia en partes iguales a los motores se tendrá el movimiento ascendente de los motores (Rodríguez, 2016).

Figura 4

Movimiento de motores



Nota. Resumen de movimientos de los motores de un micro-UAV. *Tomado de Desarrollo de Sistema de Aterrizaje Autónomo Preciso para Multicóptero* por López, S., 2017, Universidad Politécnica de Cartagena

Navegación Fina

Dentro de este apartado se revisará la navegación del micro-UAV mediante un controlador PID que permitirá tener más presión en el aterrizaje.

Control Servovisual

El propósito principal del control servo visual (Aguilar, Casaliglla, & Pólit, Obstacle Avoidance Based-Visual Navigation for Micro Aerial Vehicles, 2017) es controlar el dron utilizando los datos obtenidos en el espacio de la imagen (Zhang, Yu, Jia, & Wang, 2017). Existen dos categorías, basado en imágenes y basado en pose. En el primero se puede

trabajar directamente con los datos de la imagen, mientras que en el segundo implica una estimación de poses (Yang, Xiao, Fu, & Chaofan, 2017).

El control de servo visual implica el uso de referencias de imágenes obtenidas a través de elementos fotográficos, como cámaras de video, para mover los servos de la manera deseada en función de la información obtenida (Caiza, Aguilar, Albán, & Fernández, 2020). Este tipo de control se utiliza para plataformas estabilizadas por giroscopio (*gimbals*), por ejemplo, para rastrear un objetivo determinado (Soria, Roberti, Carelli, & Sebastian, 2008).

La unificación de estas plataformas es bastante común en el desarrollo de UAV (vehículos aéreos no tripulados), ya que pueden compensar el movimiento del vehículo para proporcionar una imagen estable (Murillo Álvarez, 2021).

Modelo Servovisual.

En esta sección se muestra el modelo dinámico de un cuadricóptero, se debe considerar que la entrada es la fuerza de empuje de los 4 rotores y los pares aplicados al mismo (Guarnizo, 2019). Para lo cual se considera:

1. La estructura del vehículo es indeformable.
2. Los efectos aerodinámicos de la estructura son despreciables.
3. El cuadricóptero es totalmente simétrico respecto a sus ejes.
4. El vehículo no realiza movimientos bruscos.
5. La dinámica rotacional en lazo cerrado es más rápida que la dinámica traslacional.
6. La fuerza de empuje que generan los rotores es alcanzada instantáneamente.

El método de Newton-Euler será la base para el desarrollo del modelo dinámico, el cual describe la traslación y rotación de un cuerpo sólido en el espacio, dando como

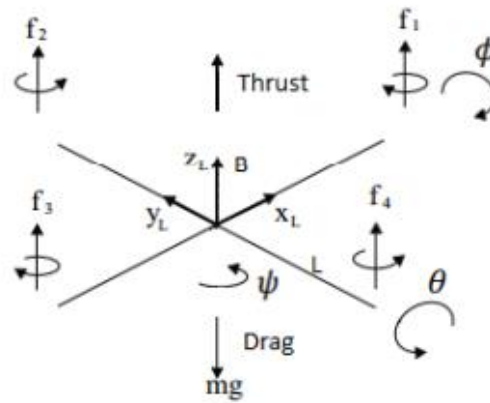
resultado que hacia el punto de referencia fijo hay a seis grados de libertad, tres grados para traslación, (Arriba-Abajo, Adelante-Atrás, Derecha-Izquierda) y tres grados para rotación, (Yaw ψ , Pitch θ , Roll ϕ). Una vez teniendo en cuenta estas consideraciones se efectuarán las soluciones a las ecuaciones finales del modelo mediante el análisis matemático (Cardoso, Fernández, Marrero-Osorio, & Guardado, 2017).

Luego de haber definido el sistema de coordenadas, se debe considerar que el sistema está enlazado al cuerpo del cuadrimotor y su origen se encuentra en el centro de masas del cuerpo, el sistema de coordenadas está dado por $B=\{X, Y, Z\}$. Para la rotación se utilizarán los ángulos de Euler, quienes describen en el espacio tridimensional, la rotación general (Jaramillo, 1998).

Con el objetivo de cambiar de un sistema de coordenadas móvil a un sistema fijo, se debe obtener una matriz que es proporcionado por cada componente del sistema fijo, girando sobre el propio eje de cada componente, obteniendo así las componentes primas (X', Y', Z') y finalmente llegar al arreglo matricial en este caso la matriz [R] Matriz de Rotación Tridimensional o Matriz de Rotación de Cosenos directores. A continuación, en la Figura 5 y Figura 6 se describe los ejes de coordenadas junto a las fuerzas aplicadas que ejercen los motores y los ángulos (ψ, θ, ϕ) que interactúan cuando se realiza la rotación de los ejes (X, Y, Z) y las componentes primas (X', Y', Z') respectivamente (Villanueva, 2018).

Figura 5

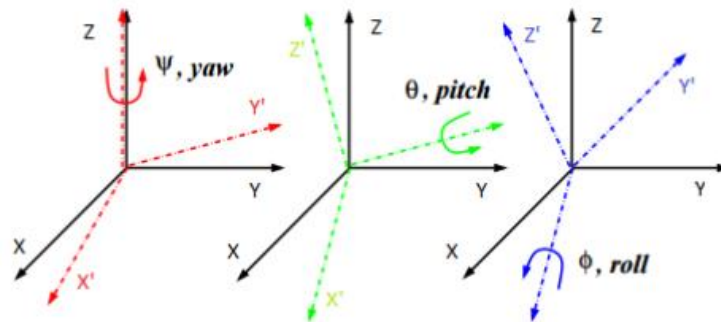
Sistema de referencias y fuerzas



Nota. La figura muestra el sistema de referencias, y el giro de los cuatro motores del micro-UAV. Tomado de *Herramienta para la Simulación Integrada de Subsistemas en un Equipo Quadrotor*, por García, R., 2015, Universidad de Sevilla.

Figura 6

Ángulos de Euler



Nota. La figura muestra la rotación de los ángulos de Yaw, Pitch y Roll. Tomado de *A Biomechanical Assessment of Direct and Inertial Head Loading in Rugby Union*, por Tierney, G., 2018, University of Dublin.

Matrices de Rotación.

De acuerdo con la Figura 3, a continuación, la rotación de X, Y, Z (Miranda, 2016).

Rotando el eje Z, se tiene el ángulo Yaw, con lo que se tendrán componentes primas en X y Y.

$$x' = a_1x + a_2y \quad (1)$$

$$y' = -b_1x + b_2y \quad (2)$$

En ángulos se tiene (1) y (2).

$$x' = \cos\psi \sin\psi \quad (3)$$

$$y' = -\sin\psi + \cos\psi \quad (4)$$

Entonces (3) y (4) en matrices es.

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\psi & \sin\psi \\ -\sin\psi & \cos\psi \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \quad (5)$$

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (6)$$

El mismo proceso anterior se cumplirá para el eje Y, por lo tanto, las matrices son

$$\begin{bmatrix} x' \\ z' \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} x \\ z \end{bmatrix} \quad (7)$$

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} \cos \theta & 0 & -\sin \theta \\ 0 & 1 & 0 \\ \sin \theta & 0 & \cos \theta \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (8)$$

Por último, el eje Z.

$$\begin{bmatrix} y' \\ z' \end{bmatrix} = \begin{bmatrix} \cos \phi & \sin \phi \\ -\sin \phi & \cos \phi \end{bmatrix} \begin{bmatrix} y \\ z \end{bmatrix} \quad (9)$$

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & \sin \phi \\ 0 & -\sin \phi & \cos \theta \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (10)$$

Una vez obtenidas las tres matrices, se proceden a multiplicar en (10) para obtener la Matriz de Rotación Tridimensional.

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} \cos \psi & \sin \psi & 0 \\ -\sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \theta & 0 & -\sin \theta \\ 0 & 1 & 0 \\ \sin \theta & 0 & \cos \theta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & \sin \phi \\ 0 & -\sin \phi & \cos \theta \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (11)$$

Algebraicamente se obtiene la matriz [R] que tiene la ventaja de simplificar los 3 ángulos (Roll, Pitch y Yaw).

$$A' = \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} \quad (12)$$

Pero la matriz que se necesita es la de rotación con respecto al sistema de coordenadas inicial (x, y, z) debido a que sobre este sistema se realizan las rotaciones(Chasnov, 2012).

Entonces, se debe realizar el cambio de sistema. Por lo que se realiza el siguiente cambio en (13).

$$A = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (13)$$

Entonces en (14) se obtiene.

$$A' = R \cdot A \quad (14)$$

Luego de aplicar algunas reglas matriciales se tiene finalmente en (15) la matriz A.

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = R^{-1} \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} \quad (15)$$

Detección y Seguimiento Visual

Este proceso es capaz de estimar la orientación y posición de un objeto respecto a una referencia, esto gracias a la unión de información que generan IMU, visión, GPS, entre otros sensores. Esta técnica es capaz de aportar a varias aplicaciones como interacción hombre – máquina, navegación de vehículos(Limaico, y otros, 2018), localización (García L. , 2016) y mapeo (Asimbaya, Albán, Aguilar, & Fernández, 2020). En esta tesis se revisarán dos tipos de

estimación, odometría visual, que es la integración y rotación de fotogramas (secuencia de imágenes) para estimar el movimiento 3D; la segunda es capaz de estimar la posición relativa del UAV (Grijalva & Aguilar, 2019) con respecto a un objeto mediante la información visual como una retroalimentación de bucles de control (Martínez Luna, 2013).

Detección de Puntos.

La detección de características es el proceso de calcular la abstracción de la información de la imagen y tomar una decisión en cada punto de la imagen para analizar si hay una característica de la imagen del tipo dado existente en ese punto. A continuación, algunas técnicas utilizadas (Karami, Prasad, & Shehata, 2017) (Salcedo, Aguilar, Cobeña, Pardo, & Zahira, 2018).

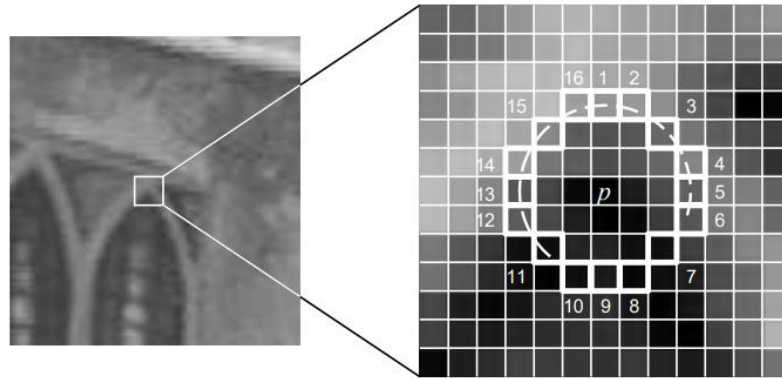
- **SIFT:** este tipo de algoritmo presenta 4 pasos básicos, utilizando la diferencia gaussiana se debe estimar un extremo del espacio de escala (Rublee, Rabaud, Konolige, & Bradski, 2011), después la localización de puntos clave, descartando los puntos de bajo contraste, a continuación, se asigna la orientación en función del gradiente local de la imagen y, finalmente se obtiene un generador de descriptores para que cada punto clave tenga su descriptor basado en la magnitud del gradiente y la orientación de la imagen (Karami, Prasad, & Shehata, 2017).
- **SURF:** a diferencia de *SIFT*, *SURF* utiliza convolución de cuadrado, la ventaja es que es más rápida y se usa la imagen íntegra, el detector es de tipo BLOB que se basa en una matriz hessiana (Angulo & Aguilar, 2013), que es la que se usa para realizar la detección de puntos de interés, para la descripción de características se selecciona una región alrededor del punto clave y se divide en subregiones para componer el descriptor *SURF* (Benavides, 2016). Gracias al signo de su Laplaciano se diferencian las manchas brillantes de las oscuras,

pero si la imagen es un fondo oscuro se evalúa el contraste, lo que permite un emparejamiento más rápido (Karami, Prasad, & Shehata, 2017).

- **ORB:** es el algoritmo en el que está basada esta tesis, básicamente determina los puntos clave RÁPIDOS (características de la prueba de segmento acelerado, (Mur-Artal, Montiel, & Tardos, 2015)) y crea un descriptor *BRIEF* (características esenciales independientes robustas binarias) (Mikolajczyk & Schmid, 2005).
- **FAST:** es un método de detección de esquinas que le permite obtener puntos de objetos que luego se usan para rastrear y mapear objetos, para cada píxel se compara rápidamente el brillo de ese píxel con los 16 píxeles que forman un círculo alrededor de la esquina, como se puede observar en la Figura 7(Zhang, Zheng, & Fu, 2020). Los píxeles del círculo se clasifican en tres clases (más claros, más oscuros o similares), y si hay más de 8 píxeles más oscuros que p, se selecciona el punto clave. Los puntos clave o puntos de interés de una imagen son los puntos más relevantes de la imagen que no cambian a pesar de algunas transformaciones (Calonder M. , Lepetit, Strecha, & Fua, 2010). Después de identificar los puntos clave, el ORB asigna una orientación a cada punto clave en función de los cambios en los niveles de intensidad alrededor del punto clave(Lowe, 2004). *BRIEF* se crea seleccionando algorítmicamente todos los puntos clave que se encuentran en la imagen con *FAST* y convertirlos en vectores de características binarias, definiendo así el objeto(Cabero & Barboza, 2019).

Figura 7

Prueba de detección de esquinas en 12 puntos en un parche de imagen

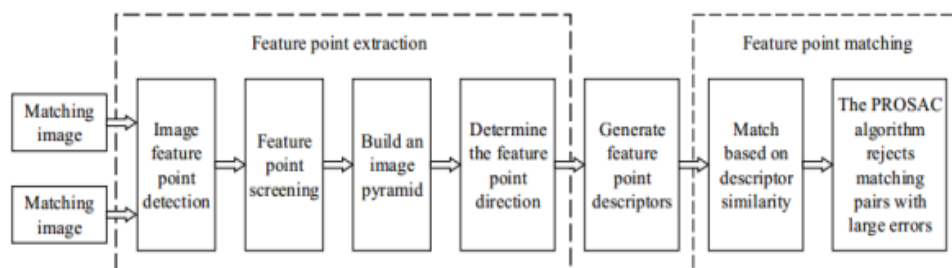


Nota. Esta imagen describe el funcionamiento del algoritmo *FAST*. Tomado de *Machine Learning for High-Speed Corner Detection*, por Rosten, E., & Drummond, T., 2006, Universidad de Cambridge.

El algoritmo de coincidencia de imágenes ORB generalmente se divide en tres pasos: extracción de puntos característicos, generar descriptores de puntos característicos y coincidencias de puntos característicos (Shu & Xiao, 2018) (Kümmerle, Grisetti, Strasdat, Konolige, & Burgard, 2011). La Figura 8 muestra el diagrama de flujo específico.

Figura 8

Diagrama de flujo de coincidencia de imágenes basado en ORB



Nota. La figura muestra el flujo que sigue el algoritmo ORB para encontrar coincidencias en los puntos. Tomado de *Overview of Image Matching Based on ORB Algorithm*, por Luo, C., Yang, W., & Huang, P., 2019, *Journal of Physics*(1237).

El algoritmo ORB utiliza el algoritmo *FAST* mejorado (características de la prueba de segmento acelerado) para detectar puntos característicos(Awaludin & Yasin, 2020). La idea es que si un píxel es significativamente diferente de los píxeles vecinos entonces es más probable que sea un punto de esquina(Luo, Yang, & Huang, 2019).

Descripción de puntos.

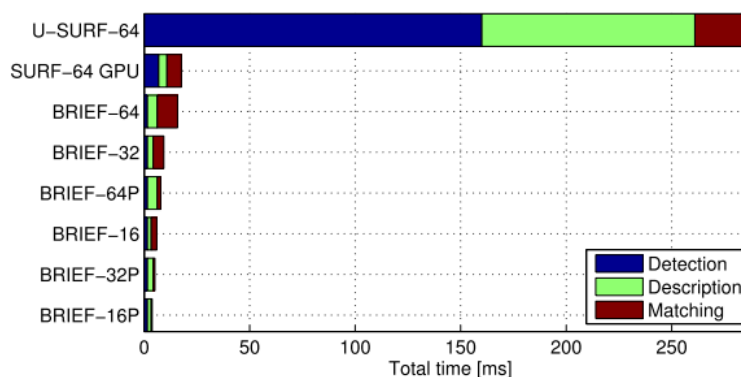
Para la descripción de puntos los más conocidos son de tipo flotantes y binarios. En el caso de detectores flotantes se tienen *SIFT*, *SURF* y *MSER* (Carmichael, Laganière, & Bose, 2010). Mientras que para descriptores binarios los más populares son *BRIEF*, *BRISK* y *FREAK* (Karami, Prasad, & Shehata, 2017). La Figura 9 muestra una comparación entre ellos.

Los descriptores binarios se están volviendo cada vez más populares como un medio para comparar puntos característicos muy rápidamente y requieren cantidades de memoria comparativamente pequeñas (Rojas, Aguilar, Álvarez, & Grijalva, 2019). Solo se necesitan 256 bits, o también 128 bits, para obtener excelentes resultados de emparejamiento (Calonder M. , Lepetit, Ozuysal, Trzcinski, Strecha, & Fua, 2012). Por lo tanto, *BRIEF* resulta ser más eficiente para calcular y almacenar en memoria. Esta característica de eficiencia permite que *BRIEF* supere fácilmente a otros descriptores como *SURF* y *U-SURF* en términos de velocidad(Wang, Zhai, Xu, & Cheng, 2017). Es un resultado importante desde un punto de vista práctico. Esto significa que el rendimiento de coincidencia en tiempo real puede lograrse incluso en dispositivos con una potencia computacional muy limitada(Qin, Xu, & Chen, 2014). También es importante destacar desde un punto de vista teórico pasar de la euclidiana a la distancia

de Hamming para fines de comparación (Calonder M., Lepetit, Ozuysal, Trzcinski, Strecha, & Fua, 2012).

Figura 9

Comparación de descriptores un ciclo de detección-descripción.



Nota. Esta figura muestra la comparación entre los descriptores binarios más importantes. Tomado de BRIEF: Computing a Local Binary Descriptor Very Fast, por Calonder, M., Lepetit, V., Ozuysal, M., Trzcinski, T., Strecha, C., & Fua, P, 2012, IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE.

Emparejamiento de Puntos de Interés.

Después de determinar la información de escala y rotación de los puntos característicos de la imagen, es necesario determinar la similitud entre los descriptores de puntos característicos. La forma más fácil de emparejar es *Brute-ForceMatcher*. El método mide la distancia entre cada punto característico para proceder a elegir el más cercano como el punto de coincidencia (Bay, Ess, Tuytelaars, & Van Gool, 2008). Para el descriptor BRIEF, la distancia de Hamming se utiliza para medir el número de caracteres diferentes entre dos caracteres de igual longitud. El algoritmo ORB utiliza FLANN (*Fast Library for Approximate*, Vecinos más cercanos) para hacer coincidir y establecer múltiples árboles KD aleatorios (Raheem & Al-Assadi, 2022). Este método es

computacionalmente más eficiente que el *Brute-ForceMatcher*. Además, el algoritmo se usa generalmente para eliminar algunos pares coincidentes con grandes errores de coincidencia, lo que mejora aún más la precisión de la coincidencia (Calonder M., Lepetit, Ozuysal, Trzcinski, Strecha, & Fua, 2012).

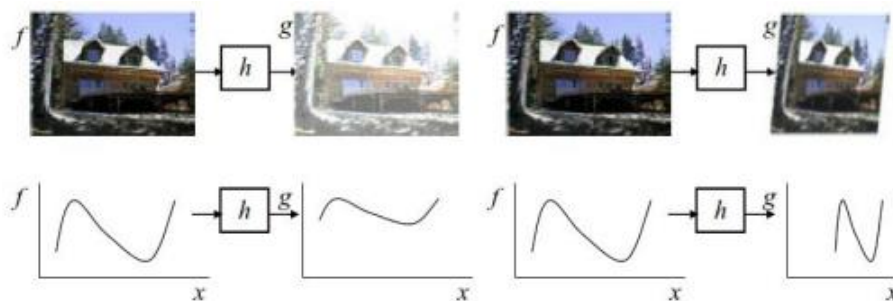
Estimación de Transformación Geométrica.

Este proceso es capaz de estimar el movimiento que permite relacionar el *frame* actual con el *frame* consigna. En 2013, Aguilar & Angulo mencionan que este proceso consta de dos partes: detección, descripción y comparación de puntos de interés, y estimación de parámetros de movimiento a partir de una correspondencia. Y una transformación es una función capaz de pasar a otro dominio, matemáticamente la ecuación que modela es:

$$g(x) = f(h(x)) \quad (16)$$

Figura 10

Modificación en las imágenes por el cambio de dominio

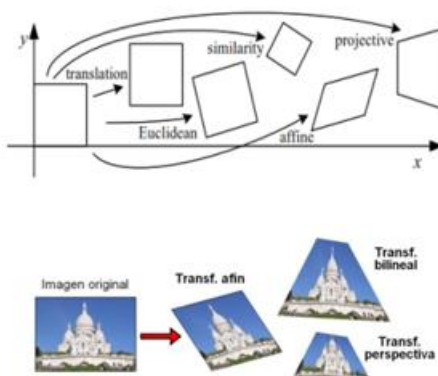


Nota. Se muestran las transformaciones de acuerdo con la ecuación (1). Tomado de *Computer Vision: Algorithms and Applications*, por Szeliski, R., 2010, Springer.

Una vez que las características de las imágenes han sido extraídas, la siguiente etapa es coincidir las características en diferentes imágenes. Uno de los componentes importantes de esta coincidencia es comprobar si el conjunto de características es congruente geoméricamente. Es decir, si los desplazamientos de las características se pueden describir mediante una transformación geométrica (Szeliski, 2010). En la Figura 11 se muestran las diferencias entre los diferentes tipos de transformaciones.

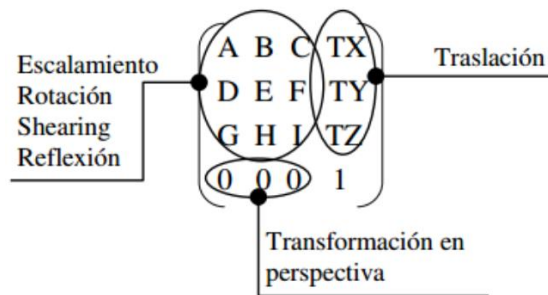
Figura 11

Transformaciones geométricas 3D



Nota. Ejemplos de transformaciones. Tomado de *Computer Vision: Algorithms and Applications*, por Szeliski, R., 2010, Springer.

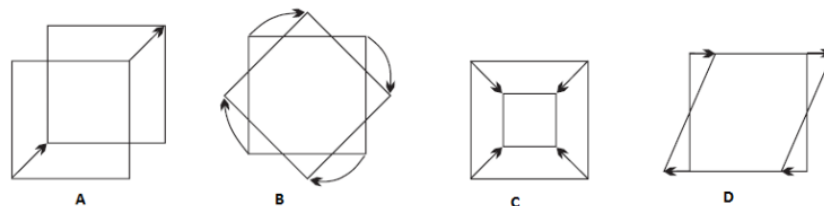
También, según Villamarín en 2015, se puede definir como operaciones matriciales sobre puntos en donde cada uno representa una matriz constituida por las coordenadas (x, y, z) . La mayoría de las transformaciones se obtienen por multiplicación de matrices, pero también se obtienen por suma de matrices como por ejemplo la transformación de tipo traslacional. En la Figura 12 se puede observar la matriz de transformación en coordenadas homogéneas de acuerdo con los tipos de operaciones.

Figura 12*Matriz de transformación*

Nota. El bloque 3x3 produce una transformación lineal, el bloque 1x3 una transformación de la perspectiva y finalmente el bloque 3x1 produce una traslación. El elemento 1x1 es un factor de escala. Tomado de *Estado del Arte, Herramientas y Aplicaciones para Transformaciones geométricas 3D*, por Villamarin, D., 2015, Congreso de Ciencia y Tecnología ESPE.

Transformación Afín.

Es una transformación lineal que conserva el paralelismo y rectitud, al igual que una proporción entre las rectas (Palomino, 2017). Esta incluye las transformadas elementales como traslación, rotación, escalado y deformación como se puede observar en la Figura 13. Las modificaciones que realiza esta transformación es el tamaño, en ciertos caso rotación y cambio de escala uniforme (Villamarin, 2015).

Figura 13*Transformación afín*

Nota. Transformadas elementales (A) traslación, (B) rotación, (C) escalado y (D) deformación. Tomado de *Estado del Arte, Herramientas y Aplicaciones para Transformaciones geométricas 3D*, por Villamarin, D., 2015, Congreso de Ciencia y Tecnología ESPE.

Por lo tanto, una transformación afín es un caso general de las transformaciones euclidianas. Por ende, en las transformaciones afines, los cuadrados pueden convertirse en rectángulos o paralelogramos (Hartley & Zisserman, 2011). La representación matricial, presenta seis elementos, por lo tanto, se tiene seis grados de libertad, la transformación se puede calcular a partir de la correspondencia de tres puntos, se presenta la siguiente ecuación:

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} \begin{bmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

Transformación de Similitud.

También es conocida como rotación escalada, esta transformación pertenece al plano \mathbb{R}^2 que conserva las distancias euclidianas, y una escala isotrópica. Puede ser expresada como $x' = sRx + t$ donde s es un factor arbitrario, de forma matricial es:

$$x' = [sR \ t]x = \begin{bmatrix} s \cos\theta & -s \sin\theta \\ s \sin\theta & s \cos\theta \end{bmatrix} x + \begin{bmatrix} t_x \\ t_y \end{bmatrix} = \begin{bmatrix} a & -b \\ b & a \end{bmatrix} \begin{bmatrix} t_x \\ t_y \end{bmatrix} \quad (17)$$

Transformación de Perspectiva.

Esta transformación lineal general, también conocida como homografía que trabaja con coordenadas homogéneas. Tiene la particularidad de que las líneas rectas permanecen rectas (Szeliski, 2010). Donde H representa una matriz homogénea de tres filas y tres columnas su expresión es:

$$\tilde{x}' = \tilde{H}\tilde{x} \quad (18)$$

A continuación, se muestra la Tabla 3 resumen, que contiene la matriz, los grados de libertad y la característica que conserva.

Tabla 3

Tipos de transformaciones con sus características principales

TRANSFORMACIÓN	MATRIZ	GRADOS LIBERTAD	CONSERVACIÓN
TRASLACIÓN	$\begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \end{bmatrix}$	2	Orientación
EUCLIDIANA	$\begin{bmatrix} c_\theta & -s_\theta & t_x \\ s_\theta & c_\theta & t_y \end{bmatrix}$	3	Tamaño
SIMILITUD	$\begin{bmatrix} 1 + a & -b & t_x \\ b & 1 + a & t_y \end{bmatrix}$	4	Ángulos
AFÍN	$\begin{bmatrix} 1 + a_{00} & a_{01} & t_x \\ a_{10} & 1 + a_{11} & t_y \end{bmatrix}$	6	Paralelismo
PROYECTIVA	$\begin{bmatrix} 1 + h_{00} & h_{01} & h_{02} \\ h_{10} & 1 + h_{11} & h_{12} \\ h_{20} & h_{21} & 1 \end{bmatrix}$	8	Líneas rectas

Nota. Resumen de las diferentes transformaciones. Tomado de *Computer Vision: Algorithms and Applications*, por Szeliski, R., 2010, Springer.

Capítulo III. Desarrollo del Sistema

Introducción

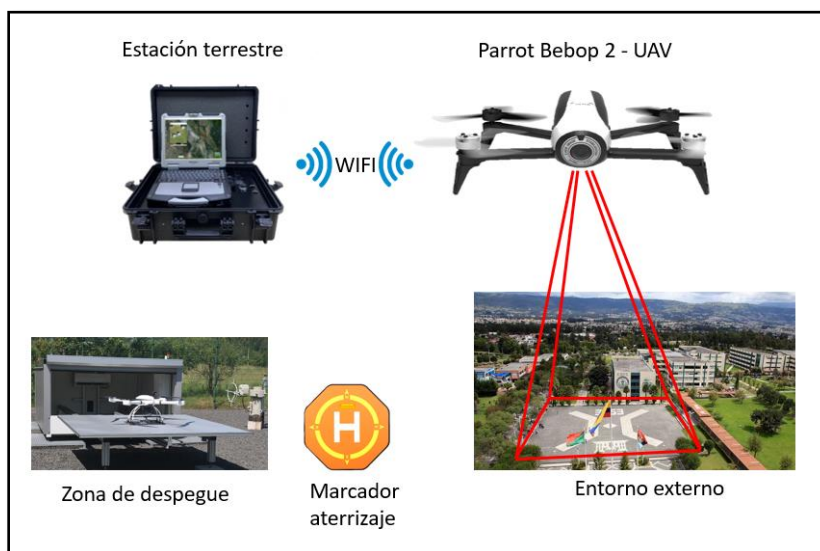
Este capítulo hará referencia a los componentes necesarios para el desarrollo de este proyecto, la comunicación necesaria para la obtención de imágenes. Así como también la librería “*bebop_autonomy*” que es la encargada del movimiento básico del micro UAV.

Hardware del Sistema

Para este proyecto es necesario una estación terrestre, la cual recibirá las imágenes captadas por el micro UAV (ParrotBebop 2) mediante el protocolo WiFi y realizar el control en tierra. También se cuenta con una zona de despegue y aterrizaje, donde se encuentra el marcador de aterrizaje (código ArUco) que posteriormente se especificará.

Figura 14

Diagrama y componentes del proyecto



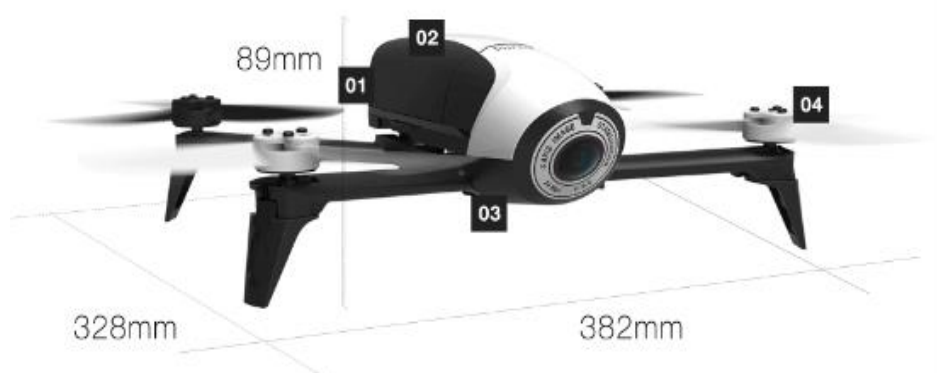
Nota. Diagrama del proyecto, y los componentes en hardware que serán necesarios para su funcionamiento.

Parrot Bebop 2 – UAV

El cuadricóptero francés *parrot* Bebop 2 cuenta con un sistema de cuatro rotores, el alcance de su señal es de hasta 300 metros, una cámara de 14 megapíxeles, y graba con un video en formato de 1080p a 30 imágenes por segundo con estabilizador de imagen. Sus mayores ventajas son su ligereza, su cámara tipo ojo de pez y la capacidad de hacer fotos a 14MP muy estables (Giernacki, Kozierski, Michalski, Retinger, Madonski, & Campoy, 2020). En la Tabla 4 se muestran las especificaciones del UAV, bebop 2.

Figura 15

Dimensiones bebop 2



Nota: Junto con las dimensiones del UAV *parrot* Bebop 2, se muestran algunos indicadores: (1) Piloto trasero de LED para una mejor visibilidad, (2) batería con mayor autonomía dando 25 minutos de tiempo de vuelo, (3) Estructura resistente y flexible, (4) Hélices flexibles. Tomado de BEBOP 2 por Kote, P., 2016.

Tabla 4

Especificaciones del UAV – parrot bebop 2

Especificaciones Técnicas Parrot Bebop 2 - UAV	
Dimensiones	38.2 x 32.8 x 8.9 cm
Conectividad	Antena Wi-Fi: MIMO a frecuencias

	2.4 y 5 GHz
Velocidad	Horizontal: 16 m/s – Despegue 6 m/s
Cámara	<ul style="list-style-type: none"> • CMOS 14 Mpx • Estabilización de video digital • Video con resolución de 1920 x 1080p a 30 cuadros por Segundo • Resolución de foto de 4096 x 3072 • Memoria de almacenamiento interno de 8GB • Tipo ojo de pez
Rango de control	300m <i>Parrot Skycontroller 2: hasta 2km</i>
Batería	Sin interrupción 25 minutos Capacidad: 2700 mAh
Procesador	Procesador <i>dual-core</i> con GPU <i>quad-core</i>
Peso	500 g
Motores	El motor también es conocido por su velocidad, capaz de alcanzar una altitud de 100 metros en solo 18 segundos después del despegue. Sin escobillas

Nota. Especificaciones técnicas del UAV *parrot bebop 2*. Tomado de BEBOP 2 por Kote, P., 2016.

Estación a Tierra

Para este proyecto se ha usado un computador portátil, cuyas especificaciones están detalladas en la Tabla 5. Pero cabe recalcar que estas especificaciones han sido suficientes para el procesamiento de imágenes en tiempo real (Basantes, y otros, 2018), comunicación vía wifi con el UAV (Jara-Olmedo A. , Medina-Pazmiño, Tozer, Aguilar, & Pardo, 2018), y en general para el coste computacional de tipo alto que requiere este proyecto. Cabe recalcar que al usar una antena de WiFi 6 se tendrá una cobertura de 500 a 800 metros, mucho más amplia con una transmisión rápida (Oughton , Lehr, Katsaros, Selinis, Bublely, & Kusuma, 2021), y es compatible con la banda de frecuencia de 2.4G (Jara-Olmedo A. , Medina-Pazmiño, Mesías, Araujo-Villaroel, Aguilar, & Pardo, 2018), en la Tabla 6 se muestran más detalles.

Tabla 5

Especificaciones del computador portátil

Modelo	MSI Katana F66/ MSI GF75 Thin
Procesador	Intel Core i7-11800H a 2.30 GHz/ Core i7-10750H a 2.60 GHz
Memoria RAM	16Gb
Sistema operativo	64bits
Tarjeta de video	Nvidia GEFORCE RTX 3060

Nota. Especificaciones de los computadores portátiles usados en este proyecto. Tomado de Katana GF75, por MSI, 2021 (MSI, 2020).

Tabla 6

Especificaciones de la antena de WiFi

Bandas	2,4, 5 GHz (160 MHz)
Máxima velocidad	2,4 Gbps
Conectividad	WiFi 6 (802.11ax)
Red	MU-MIMO dual band

Nota. Especificaciones técnicas de la antena WiFi que se usará en la estación a tierra. Tomado de *WiFi 6 AX201*, por INTEL, 2021.

Software del Sistema

De acuerdo con criterios impartidos por antiguos trabajos relacionados a este tema como por ejemplo (Merizalde, Aguilar, & Calderón, 2020), el sistema operativo utilizado fue Ubuntu 20.04 LTS, por su estabilidad y compatibilidad con las bibliotecas de procesamiento de imagen de OpenCV y con otras herramientas necesarias para el desarrollo de este proyecto. Los requisitos mínimos para esta versión de Ubuntu son los especificados en la Tabla 7.

Tabla 7

Especificaciones del computador portátil para usar Ubuntu 20.04 LTS

Procesador	Dos núcleos de 2GHz o más
Memoria RAM	2 Gb
Espacio libre	25 Gb
Acceso a internet	Si

Nota. Especificaciones de los computadores portátiles para usar Ubuntu

OpenCV

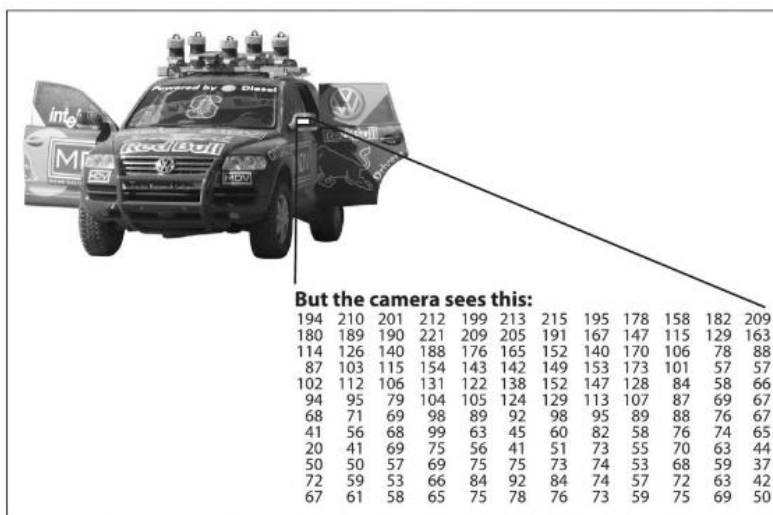
OpenCV es una biblioteca de visión artificial por computadora diseñada con un enfoque diferente y robusto para aplicaciones en tiempo real. OpenCV está hecho en C y tiene la capacidad de aprovechar procesadores de tipo multinúcleo. Uno de los objetivos de OpenCV es facilitar una infraestructura de visión artificial fácil de utilizar creando aplicaciones de visión robustas. La biblioteca OpenCV posee más de 500 funciones que cubren muchos dominios de visión (Cowan, Imanberdiyev, Fu, Dong, & Kayacan, 2016). Dado que la visión por computadora y el aprendizaje automático suelen trabajar en conjunto, OpenCV también incluye una completa biblioteca general de aprendizaje automático (MLL). MLL es útil como herramientas de visión que están en el origen de la misión de OpenCV, pero es lo suficiente para usarse en cualquier tipo problema de aprendizaje automático (Gary & Kaehler, 2008).

Visión por Computadora.

Es la transformación de los datos de una cámara en una nueva representación como se puede observar en la Figura 16. La entrada de datos puede incluir información contextual como "la cámara de un UAV". Un ejemplo podría ser convertir una imagen de color a una imagen de grises o eliminar el movimiento de cámara que se produce cuando se está realizando una secuencia de imágenes. OpenCV está en la capacidad de proporcionar herramientas básicas que faciliten la resolución de problemas respecto a visión por computadora. A menudo, la biblioteca de alto nivel será suficiente para resolver complejos problemas de visión artificial (Gary & Kaehler, 2008).

Figura 16

Visión por computadora



Nota. La Figura muestra lo que la computadora ve, un conjunto de datos, solo una cuadrícula de números. Tomado de *Learning OpenCV: Computer Vision with the OpenCV Library*, por Gary, B., & Kaehler, A., 2008, Oreilly.

Entorno de Desarrollo: ROS

El sistema operativo de robot (ROS) es un conjunto de bibliotecas de software y herramientas utilizadas para el desarrollo de software robótico a gran escala, ROS es conocido por un diseño distribuido y modular (Antenucci, Mazzaro, Fiorilla, Messina, Massa, & Matta, 2021). Es una plataforma totalmente abierta, esto es especialmente cierto cuando hardware y muchos niveles de software se diseñan y se depuran en paralelo. ROS transmite datos entre módulos utilizando comunicaciones entre procesos, y no requiere que los módulos se enlacen en el mismo ejecutable. (Quigley, Gerkey, Conley, Faust, & Foote, 2010)

ROS Noetic es uno de los lanzamientos recientes de ROS, esta versión fue lanzada el 23 de mayo de 2020. Está diseñado para Ubuntu 20.04 versión que se utilizó en este trabajo.

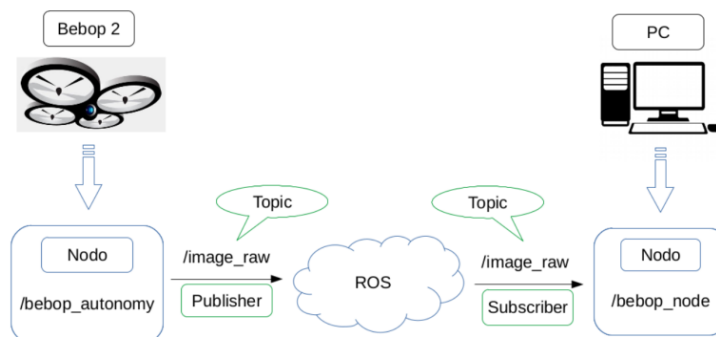
Conceptos Básicos.

Los conceptos que se utilizaron a lo largo de este proyecto han sido nodos, tópicos, servicios, paquetes, etc. A continuación, se detallan los conceptos(ROS, 2014).

- **Nodos:** Un sistema de control de robot generalmente es el conjunto de nodos, estos nodos son procesos que son los encargados de hacer los cálculos. Para declarar un nodo ROS, se usa una biblioteca de cliente ROS, en este proyecto se usó rospy pero también se puede usar roscpp.
- **Maestro (*Master*):** Permite el intercambio de mensajes o llamar servicios una vez estos se encuentren.
- **Mensajes:** Es la forma en la que se comunican los nodos, es una estructura de datos. Se pueden admitir como enteros, flotantes, booleanos, matrices, estructuras anidadas, etc.
- **Servicios:** Es la asistencia que ofrece un nodo a un cliente, pero para que este cliente acceda a este servicio se debe solicitar dicho servicio. Este paradigma de comunicación es de tipo publicación- suscripción, que es el más flexible.
- **Tópicos:** Como se mencionó anteriormente, la comunicación es de tipo publicación y suscripción. Por lo tanto, los nodos son capaces de enviar mensajes a través de la publicación en tópicos, por ende, también sirve para dar una identificación al mensaje.

Figura 17

Elementos de la red de ROS



Nota. Gráfico de los elementos detallados anteriormente. Tomado de *Desarrollo de aplicaciones basadas en visión con entornos ROS para el Drone*, por Valero, C., 2017, Universidad de Alcalá.

Sistema de Detección: YOLO

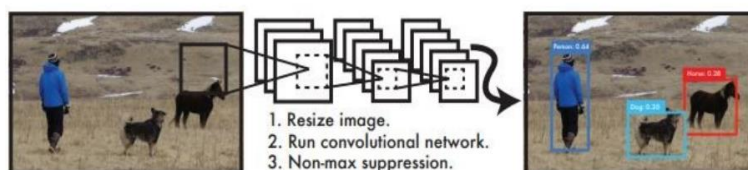
La detección de objetos es una tarea importante en muchos campos como medicina, navegación de robots (Caballeros, Segarra, Castro, & Aguilar, 2018), conducción automática (Sandoval, Limaico, Villegas-Pico, Asimbaya, & Aguilar, 2019), realidad aumentada (Collaguazo, Tituaña, Aguilar, & Amaguaña, 2018), etc. Existen muchos métodos tradicionales como redes neuronales, pirámide espacial, Redes totalmente convolucionales (Quisaguano, Rodríguez, Alvarez, Limaico, S, & Aguilar, 2018) basadas en regiones, pero YOLO (*You Only Look Once*) muestra mayores ventajas (Fang, Wang, & Peiming, 2020). YOLO es uno de los métodos más rápidos de detección con buen rendimiento en tiempo real y alta precisión, y se ha ido mejorando desde que se propuso (Munin, y otros, 2023).

YOLO es entrenado con imágenes completas y optimiza directamente el rendimiento de detección, como se puede observar en la Figura 18. Este modelo tiene varias ventajas sobre otros modelos tradicionales para detección de objetos (Kumar, Brunet, & Elias, 2023). Primero,

YOLO es muy rápido porque la detección se enmarca como un problema de regresión, no requiere canalizaciones complejas y se ejecuta a 45 fotogramas. La versión rápida ejecuta a más de 150 fps. Esto quiere decir que el procesamiento de video en tiempo real se puede lograr en menos de 25 milisegundos de latencia. Además, YOLO alcanza más del doble de precisión promedio de otros sistemas en tiempo real (Delgado , 2019).

Figura 18

Sistema de Detección YOLO



Nota. Como se puede observar el procesamiento con YOLO es simple y directo. Tomado de *Desarrollo de un sistema de detección y predicción de la pose 3D de objetos en la escena mediante técnicas de deep learning*, por Delgado, A., 2019, Universidad de Alicante.

PyTorch

PyTorch es una biblioteca de tensores optimizada para el aprendizaje automático de código abierto mediante GPU y CPU, basado en Torch. A continuación, se describen algunas características de esta biblioteca:

Estabilidad: Mantenimiento de compatibilidad con versiones anteriores. Evitar que existan vacíos en la documentación, es decir, descartar algunas funciones. Por lo tanto, en PyTorch se mantendrán a largo plazo las funciones y, por lo general, no debería haber grandes limitaciones de rendimiento.

Beta: estas características están etiquetadas como Beta porque la API puede cambiar según los comentarios de los usuarios, con el objetivo de mejorar el rendimiento y cobertura entre operadores.

Existen varios modelos, pero en comparación con los marcos de aprendizaje convencionales (Falconi, Perez, Aguilar, & Conci, 2020) (Conci, Falconi, Perez, & Aguilar, 2020) (Falconí, Pérez, & Aguilar, 2019) Chainer, CNTK, MXNet, PaddlePaddel y TensorFlow, PyTorch tiene una puntuación del 17 por ciento del marco más rápido en los seis puntos de referencia (Hecate, 2019).

Figura 19

Comparación de la velocidad de seis modelos de aprendizajes

Framework	<i>Throughput (higher is better)</i>					
	AlexNet	VGG-19	ResNet-50	MobileNet	GNMTv2	NCF
Chainer	778 ± 15	N/A	219 ± 1	N/A	N/A	N/A
CNTK	845 ± 8	84 ± 3	210 ± 1	N/A	N/A	N/A
MXNet	1554 ± 22	113 ± 1	218 ± 2	444 ± 2	N/A	N/A
PaddlePaddle	933 ± 123	112 ± 2	192 ± 4	557 ± 24	N/A	N/A
TensorFlow	1422 ± 27	66 ± 2	200 ± 1	216 ± 15	9631 ± 1.3%	4.8e6 ± 2.9%
PyTorch	1547 ± 316	119 ± 1	212 ± 2	463 ± 17	15512 ± 4.8%	5.4e6 ± 3.4%

Nota. En esta imagen se muestra la velocidad de entrenamiento del modelo de aprendizaje para los seis modelos comparados. Tomado de *Synced*, por Hecate, H., 2019.

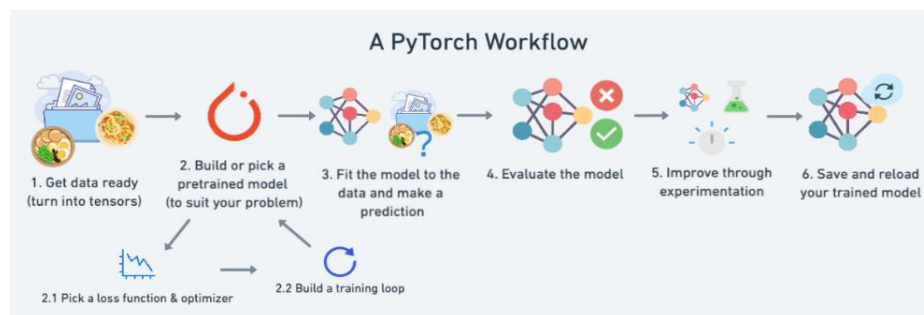
Flujo de Trabajo.

El flujo de trabajo de Pytorch se describe en la Figura 20. En primer lugar, se preparan los datos, en el caso de este trabajo son imágenes de personas (para la detección) pero en general se pueden utilizar cualquier tipo de dato. Para construir el modelo, es necesario aprender patrones en los datos (optimizador y un ciclo de entrenamiento). Para el entrenamiento del modelo (ajuste), en este paso el modelo debe encontrar patrones en los datos. Luego se debe evaluar al modelo, comparando sus

resultados con los datos reales. Para finalmente cargar el modelo al uso que se requiera (Anónimo, 2022).

Figura 20

Flujo de trabajo de PyTorch



Nota. Etapas del flujo de trabajo de PyTorch. Tomado de *Zero to Mastery Learn PyTorch for Deep Learning*, por Anónimo, 2022.

Red Neuronal Convolutacional (CNN) en *PyTorch*.

Las CNN son conocidas por su capacidad para descubrir patrones en datos visuales. Dado que estamos tratando con datos visuales, el uso de un modelo CNN mejora la línea de base. Una CNN es una versión regularizada de un perceptrón multicapa (Cleary, Yoo, Samuel, George, Sun, & Israel, 2020). Un perceptrón multicapa generalmente se refiere a una red completamente conectada, donde cada neurona en una capa está conectada a todas las neuronas en la siguiente capa. La conectividad de estas redes permite el sobre ajuste de datos (Anónimo, 2022).

Comunicación del Sistema

Para empezar la comunicación entre la estación en tierra (ordenador) y el micro UAV es necesario hacerlo mediante una red WiFi que es proporcionada por el micro UAV (Pardo, Aguilar, & Toulkeridis, 2017).

Es importante saber que esta comunicación se realiza mediante un *driver* de ROS normalmente usado para drones bebop de tipo 1 y 2. Este controlador oficial de Parrot se basa en ARDroneSDK3.

Ejecución

Para la ejecución se debe utilizar el archivo "*bebop_node.launch*", para acceder se debe usar el comando "cd" junto a la ruta: `$ roslaunch bebop_driver bebop_node.launch`. Una vez ejecutado el comando es posible conocer y manejar las distintas variables del micro-UAV. Es indispensable que la instalación corresponda al micro- UAV utilizado, ya sea "bebop 1" o "bebop 2".

Tópicos principales.

Los tópicos de la Tabla 8 se usarán para el control de todos los movimientos del micro UAV y los de la cámara, estos tópicos se encontrarán en la biblioteca Rospy cuando se escribe en lenguaje Python. Mientras los tópicos de la Tabla 9 son suscritos, usados para adquirir las imágenes de la cámara y saber el estado de la batería de igual forma que los anteriores estos se encuentran en la biblioteca Rospy.

Tabla 8*Tópicos publicados*

TÓPICO	MENSAJE	ACCIÓN
/BEBOP/LAND	std_msgs.msg.Empty	Despegue
/BEBOP/TAKEOFF	std_msgs.msg.Empty	Aterrizaje
/BEBOP/CMD_VEL	geometry_msgs.msg.Twist	Movimientos lineales en x, y, z.
/BEBOP/CAMERA_CONTROL	geometry_msgs.msg.Twist	Movimiento virtual de la cámara

Nota. Lista de tópicos publicados para realizar las acciones.

Tabla 9*Tópicos suscritos*

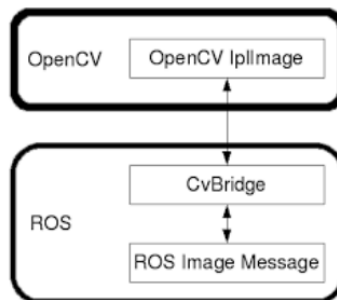
TÓPICO	MENSAJE	ACCIÓN
/bebop/states/common /commonstate /batterystatechanged	commoncommonStatebatteryStateChange	Porcentaje de batería
/bebop/image_raw	sensor_msgs.msg.Image	Obtención de imágenes

Nota. Lista de tópicos suscritos.

El proceso que se observa en la Figura 21, El mensaje se publica en el tópic "bebop/image_raw", luego se llama a la función "callback()", que permite la ejecución basándose en codificar el mensaje de ROS a un objeto OpenCV, mediante la biblioteca CvBridge, disponible en ROS y finalmente se almacena en el parámetro "Image".

Figura 21

Sistema de Detección YOLO



Nota. Flujo de YOLO. Tomado de *Seguimiento y búsqueda de objetivos en entornos complejos usando micro vehículos aéreos con cámaras monoculares para aplicaciones militares*, por Chauca, B., 2020, Departamento de Eléctrica, Electrónica y Telecomunicaciones.

Detección de Intrusos

Otro de los propósitos de este proyecto es implementar una herramienta para la detección de personas (Luna M. A., y otros, 2017) en las imágenes obtenidas mediante la cámara incorporada del UAV bebop, es decir, que sea capaz de identificar intrusos en el entorno propuesto (Tinizaray, Aguilar, & Lucio, 2022), así como también comunicar la existencia y localización de estos.

Por lo tanto, es necesario el uso de las siguientes herramientas en sus respectivas versiones:

- Python 3.8
- OPenCV 4.5
- Driver Nvidia 525
- PyTorch
- YOLO v8

Entrenamiento del Modelo

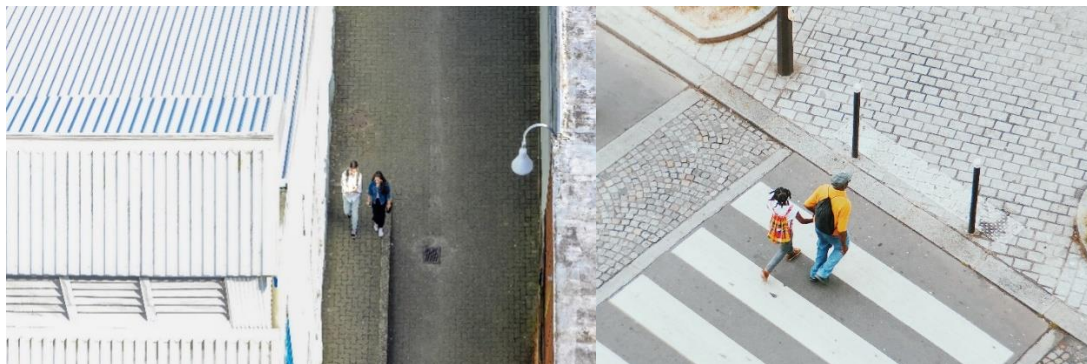
En este apartado se explicará el proceso de entrenamiento de la red neuronal (Quisaguano, Alvarez, Pardo, Zahira, & Aguilar, 2018) para lo cual se debe llevar a cabo las siguientes etapas: la adquisición de las imágenes, etiquetado de los objetos a detectar, en este caso (Luna M. A., y otros, 2017), y el entrenamiento en función de las imágenes y etiquetas (Galindo, Aguilar, & Reyes Ch, 2019).

Datos de Entrenamiento

Los datos para entrenar serán imágenes que muestren personas tal como se observa en la Figura 22. Teniendo en cuenta que el ángulo de inclinación de la toma sea de más o menos 45 grados y estén a una altura aproximadamente de 10 metros (Córdova, Jiménez, Pardo, Toalombo, & Aguilar, 2018). Para la selección de imágenes se usaron 400 imágenes, 300 para entrenamiento y 100 para validación.

Figura 22

Imágenes base para la recolección de datos



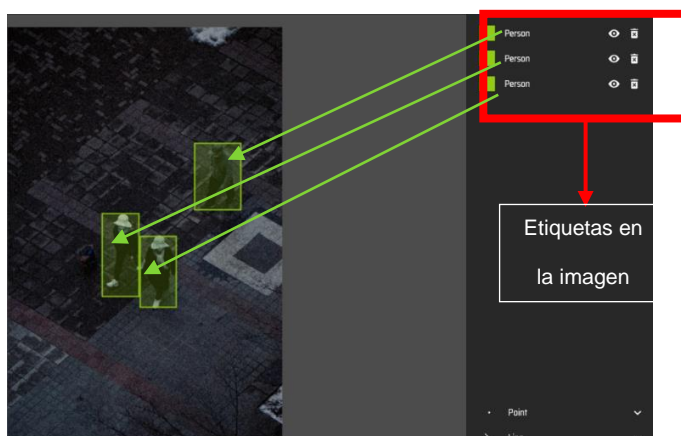
Nota. Estas imágenes fueron tomadas de la plataforma Unsplash, una fuente de internet de imágenes de uso libre. Tomado de *Unsplash*, por Images, 2022.

Etiquetado de personas

El siguiente paso es etiquetar las imágenes utilizadas como se puede observar en la Figura 23, la herramienta para este proceso de etiquetado fue makesense.ai que su uso es en línea y gratuito. Gracias al uso de un navegador, no requiere ninguna instalación.

Figura 23

Etiquetado de imágenes en makesense.ai



Nota. En la figura se puede observar la imagen con sus respectivas etiquetas, realizado en makesense.ai

Cuando haya finalizado el proceso de etiquetado tanto para las imágenes de entrenamiento como para validación, se deben exportar los datos en formato YOLO ya que este será usado en el siguiente paso, entrenamiento (Grijalva, Cueva, Ramírez, & Aguilar, 2019).

Entrenamiento

Posterior a tener los datos de entrenamiento y validación, se debe configurar el archivo “.yaml” que indica la ruta de almacenamiento de las imágenes que serán la entrada de la red neuronal a entrenar, también se debe configurar las clases a entrenar con referencia a los distintos objetos a detectar (Cobeña, Rodríguez, Salcedo, Collaguazo, & Aguilar, 2018), en este caso personas. En la Figura 24 se muestra dicha configuración.

Figura 24

Configuración archivo “.yaml” de datos a entrenar y clase de datos

```
path: C:/Users/Usuario/PycharmProjects/ENTRENAR_MODELO/data
train: images/train
val: images/val
nc: 1
# Clases
names:
0: Person
```

Nota. La Figura muestra la configuración del *dataset* de entrenamiento.

Después de haber configurado los parámetros del modelo, se deben configurar los parámetros de entrenamiento especificando el modelo, el conjunto de datos y número de épocas. En la Figura 25 se muestra dicha configuración.

Figura 25

Configuración para entrenamiento del modelo

```
from ultralytics import YOLO
# Modelo de Yolo a utilizar
model = YOLO("yolov8n.yaml")
# Entrenamiento
model.train(data="custom.yaml", epochs=100)
```

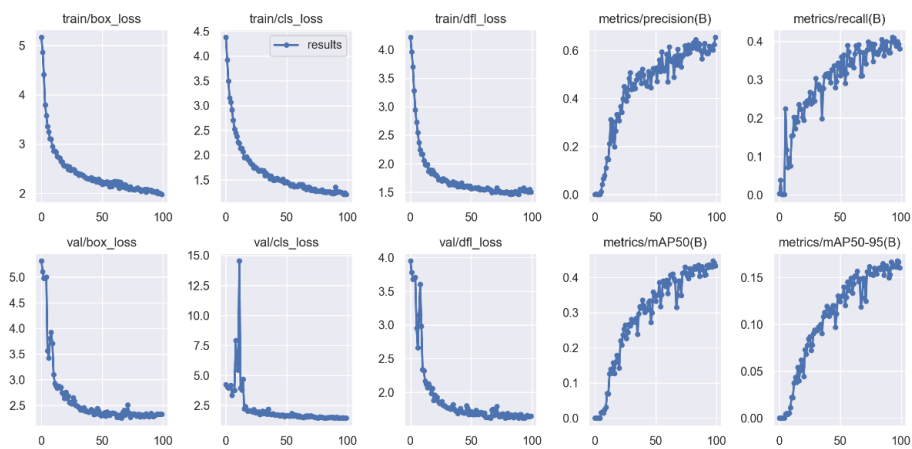
Nota. La Figura muestra el modelo de YOLO para ser usado.

Resultados

En la Figura 26 se muestran los resultados obtenidos del entrenamiento del modelo, el valor de pérdida disminuye; esto debido a que la red neuronal adquiere más información y por ende es más robusta en la detección. Entonces se puede apreciar que los valores de pérdida disminuyen y permiten un mejor resultado del modelo, el cual tardó 12 horas en ser entrenado.

Figura 26

Resultados de entrenamiento del modelo para la detección de intrusos



Nota. Los resultados del entrenamiento muestran que el valor de pérdida disminuye.

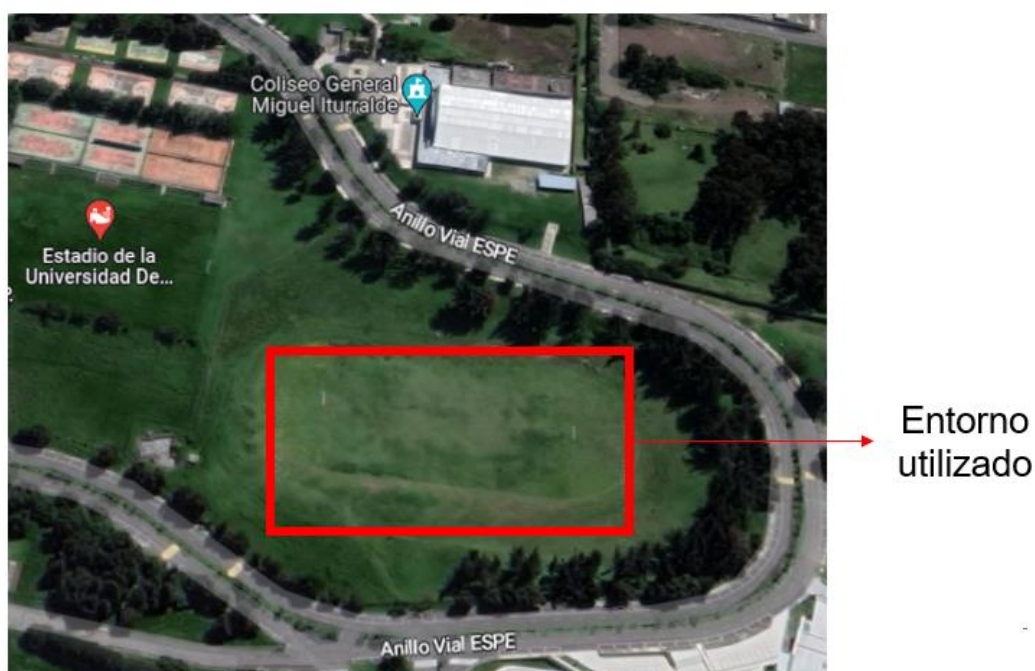
Capítulo IV. Navegación Autónoma en Entornos Externos

Descripción del Entorno

El entorno utilizado se encuentra ubicado en la Universidad de las Fuerzas Armadas ESPE en la provincia de Pichincha, campus principal. Específicamente en una de las cachas del estadio de la universidad tal como se muestra en la Figura 27.

Figura 27

Estadio de la Universidad de las Fuerzas Armadas, entorno utilizado



Nota. La Figura muestra el campus de la Universidad de la Fuerzas Armadas ESPE, específicamente el estadio.

Se seleccionó este espacio debido a que no existe ningún obstáculo que impida las pruebas a realizar. A continuación, en la Figura 28 se muestra la ubicación de la estación a tierra (marcador) y el espacio donde el micro-UAV hará el recorrido.

Figura 28

Zona de vuelo y aterrizaje del micro-UAV

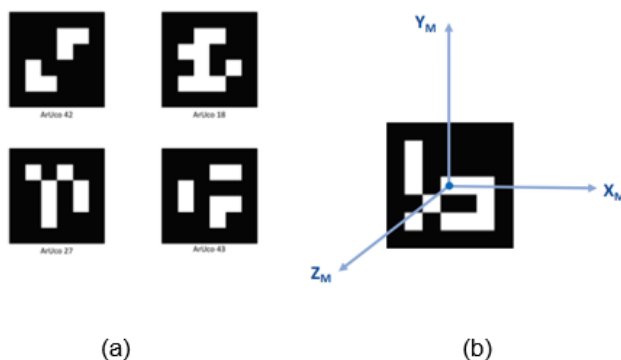


Marcadores de referencia

El marcador utilizado definirá la posición de aterrizaje del micro UAV, será un código ArUco. El código ArUco es un marcador compuesto cuadrilátera que consta de un borde negro y una matriz binaria interna que se utiliza para determinar el identificador. La matriz interna será dada por el tamaño del marcador (Hoefgeest Toribio, Fernández García, Álvarez García, & González, 2019). Existen distintos diccionarios ArUco que agrupan tokens con múltiples identificadores, en este caso el diccionario utilizado es 5x5_100(Ridlwan, Nugraha, Riansyah, & Trilaksono, 2017).

Figura 29

MarcadorArUco



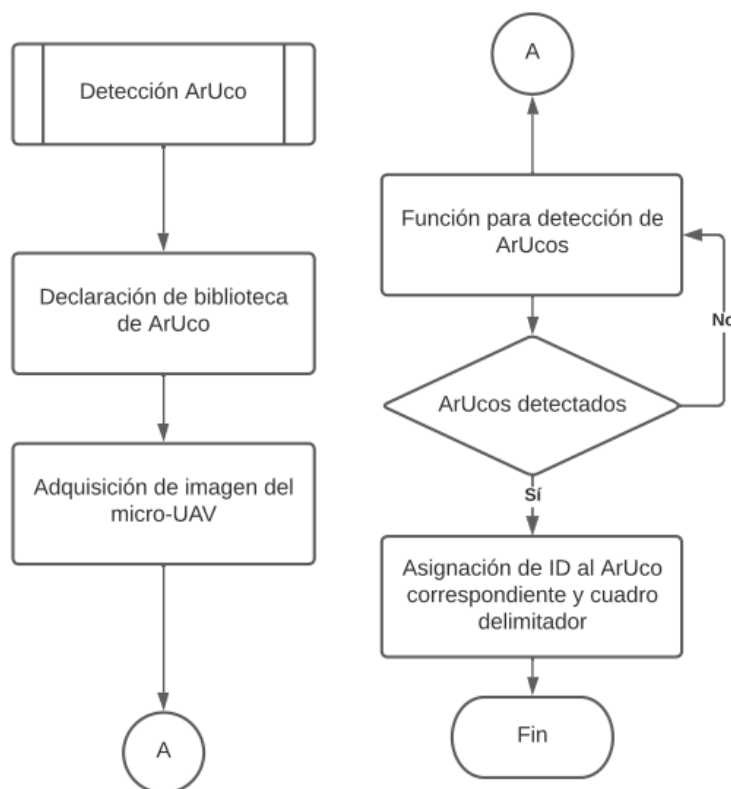
Nota. Diferentes marcadores ArUco (a), y el sistema de coordenadas de un marcador (b).

Tomado de Localización de Robots Móviles en Entornos Industriales usando un Anillo de Cámaras, por Hoefgeest Toribio, S. R., Fernández García, Á., Álvarez García, I., & González, R., 2019, Universidad de Oviedo.

El proceso de detección del marcador ArUco necesario para el aterrizaje del micro UAV se puede observar en la Figura 30, se debe tomar en cuenta la biblioteca a utilizar tal como se mencionó anteriormente.

Figura 30

Diagrama de flujo para la detección de marcador



Nota. Diagrama de flujo a la detección del marcador haciendo uso de la librería de ArUco, identificando los distintos marcadores de la biblioteca usada.

Estimación del Modelo del micro-UAV

En el presente trabajo se han establecido dos tipos de navegación, fina y gruesa, debido a la exactitud que debe establecerse. Por lo que es necesario estimar el modelo del micro-UAV haciendo referencia a la planta de un sistema de control estableciendo los parámetros del controlador a implementarse, teniendo en consideración un modelo matemático con aproximación numérica de tipo SISO (Aguilar, Costa-Castelló, & Angulo, Autonomous Navigation Control for Quadrotors in Trajectories Tracking, 2017)(Aguilar & Angulo, 2015)(Orbea, Moposita, Aguilar, Paredes, León, & Jara-Olmedo, 2017).

Tomando como base la metodología desarrollada en (Chauca Vera, 2020)y (Aguilar, Salcedo, Sandoval, & Cobeña, 2017), se define un modelo basado en visión en el que se establecen las siguientes consideraciones:

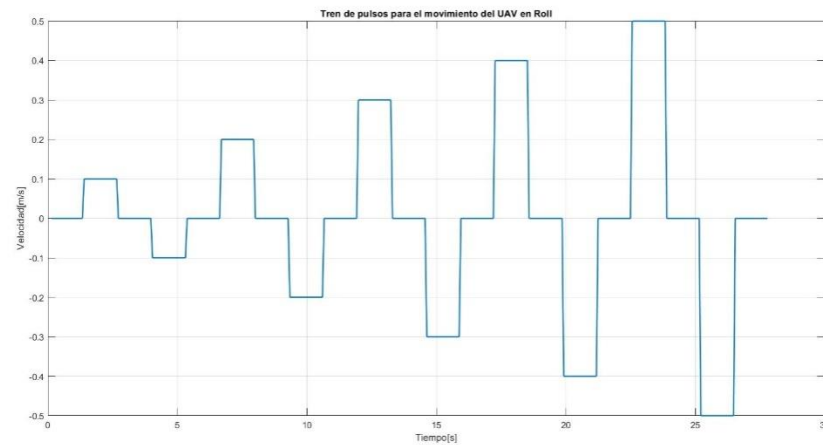
1. El sistema cuenta con la detección del marcador (Código ArUco) para establecer el tiempo de muestreo.
2. La salida del sistema será el desplazamiento en píxeles entre un *frame* y otro, en el que se encuentra situado el marcador, y una vez obtenidas las coordenadas del centro del marcador (c_x, c_y) se podrán estimar cada uno de los movimientos correspondientes al sistema Pitch y Roll.
3. La entrada del sistema será un tren de pulsos que se publicará en el tópic de la velocidad del micro-UAV, en el parámetro correspondiente. Los valores tienen un rango entre -1 y 1, pero se ha dispuesto como máximo 0.5 y mínimo -0.5 debido a que si su magnitud es mayor el marcador detectado se encontrará fuera del área de visión del micro-UAV.

Modelo de la Planta para el Movimiento en Roll

En la Figura 31y Figura 32 se visualiza la variación del desplazamiento entre un *frame* y otro respecto a la acción generada por el valor de velocidad introducido al micro-UAV, para el parámetro “*linear.y*”

Figura 31

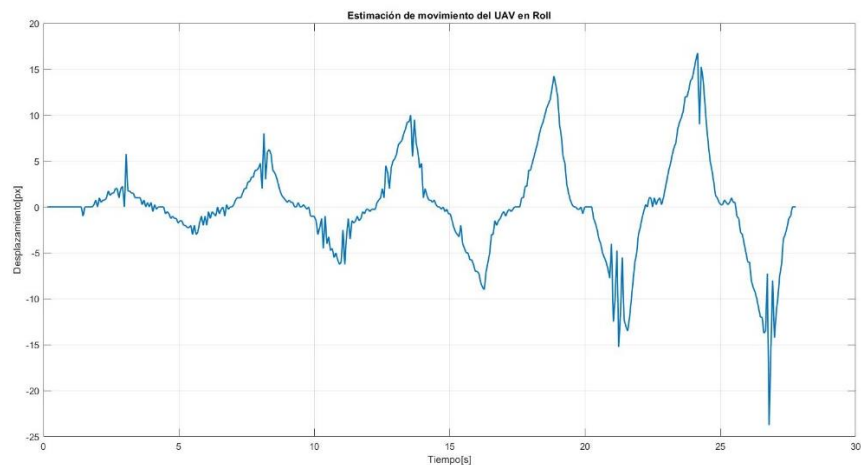
Entrada del sistema para estimación de modelo del movimiento en Roll



Nota. La figura muestra un tren de pulsos, que es la entrada para la estimación del modelo en Roll.

Figura 32

Salida del sistema para estimación de modelo del movimiento en Roll



Nota. La figura muestra los resultados obtenidos de la variación del desplazamiento en pixeles entre un *frame* y otro.

Se observa que la salida del sistema presenta oscilaciones esto se debe a las acciones generadas por el viento, dando como resultado una variación al obtener el valor de desplazamiento respecto a los pixeles.

A través del *software* Matlab se accede a una de las aplicaciones, PID Tuner, por medio de esta se estimará la planta, para lo cual se requieren 3 parámetros:

- Entrada del sistema: Tren de pulsos, son los datos obtenidos en la Figura 31.
- Salida del sistema: Delta desplazamiento entre un *frame* y otro, datos obtenidos de la Figura 32.
- Tiempo de muestreo: Es la frecuencia con la que se han tomado los datos, es decir, los datos obtenidos entre cada *frame*. Este valor se obtiene a través de los FPS de ejecución del sistema, en este caso es de 16 FPS dando como resultado un tiempo de muestreo de 62.5 ms

Haciendo uso de la función que permite autoestimar los valores se obtiene el modelo de la planta de primer orden con la siguiente configuración.

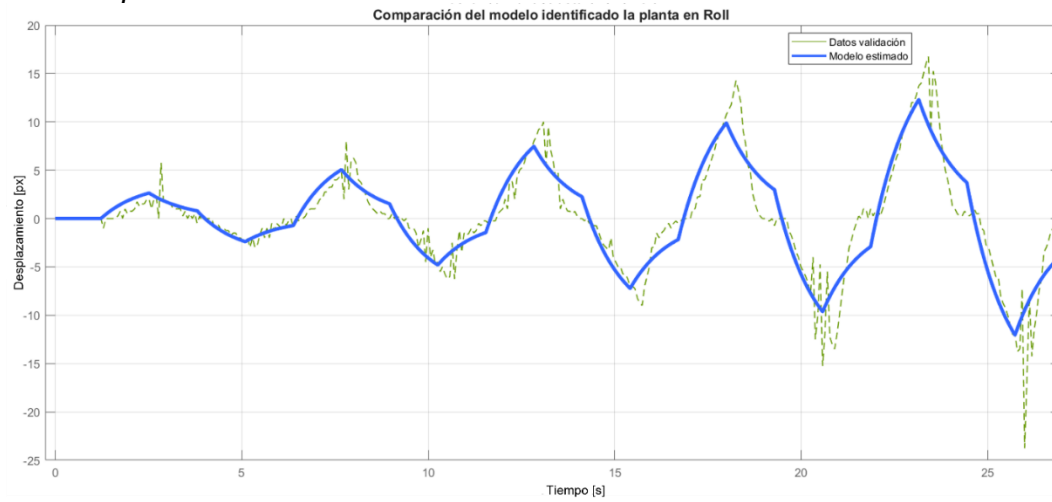
$$G(s) = \frac{K}{T_1s+1} \quad (22)$$

Obteniendo:

$$K = 37.71 \quad (23)$$

$$T_1 = 1.078 \quad (24)$$

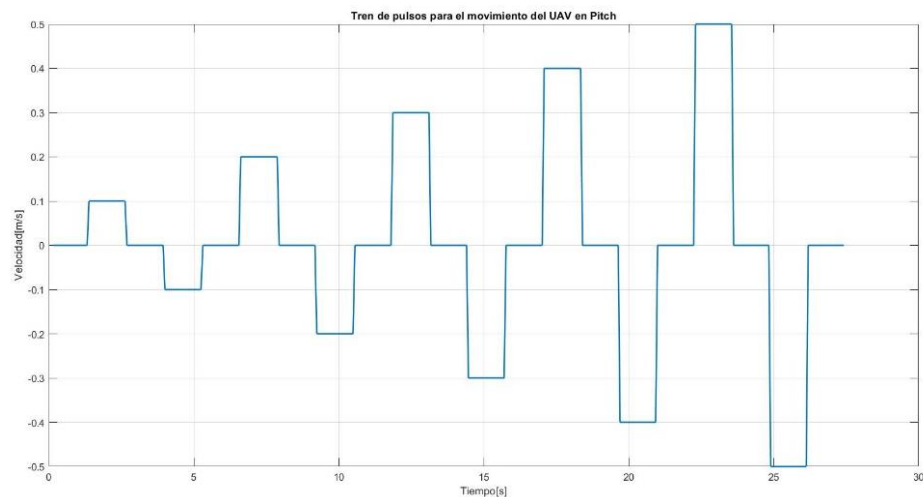
En la Figura 33 se muestra la comparación de la estimación del modelo de la planta y los datos obtenidos, se puede apreciar que se consigue una estimación adecuada de los datos.

Figura 33*Identificación de la planta en Roll*

Nota. La figura muestra la comparación entre los datos de validación y el modelo estimado.

Modelo de la Planta para el Movimiento en Pitch

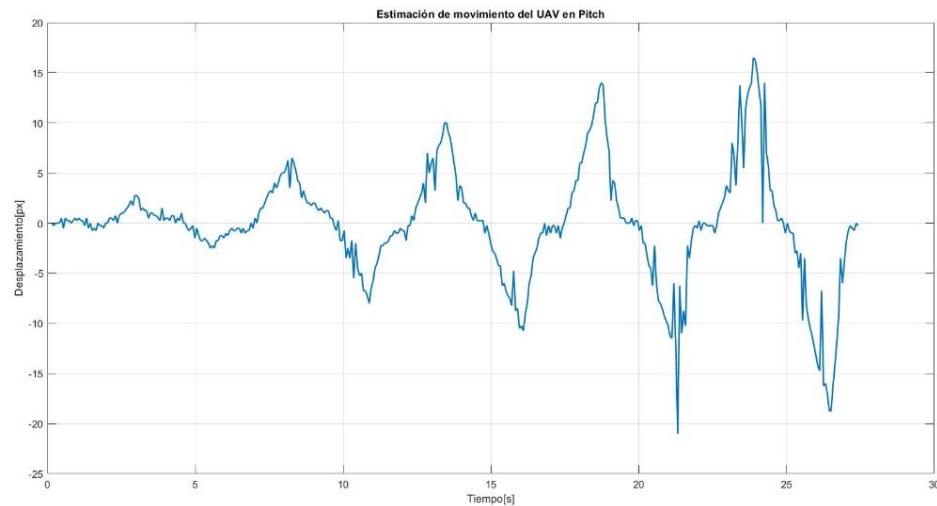
En la Figura 34 y Figura 35 se visualiza la variación de desplazamiento entre un *frame* y otro respecto a la acción generada por el valor de velocidad introducido al micro-UAV en este caso el parámetro “*linear.x*”

Figura 34*Entrada del sistema para estimación de modelo del movimiento en Pitch*

Nota. La figura muestra un tren de pulsos, que es la entrada para la estimación del modelo en Pitch.

Figura 35

Salida del sistema para estimación de modelo del movimiento en Pitch



Nota. La figura muestra los resultados obtenidos de la variación del desplazamiento en píxeles entre un frame y otro.

Del mismo modo las oscilaciones que se visualizan son generadas por el viento como se mencionó en el modelo de Roll, así como también el procedimiento usando Matlab para el modelo de la planta.

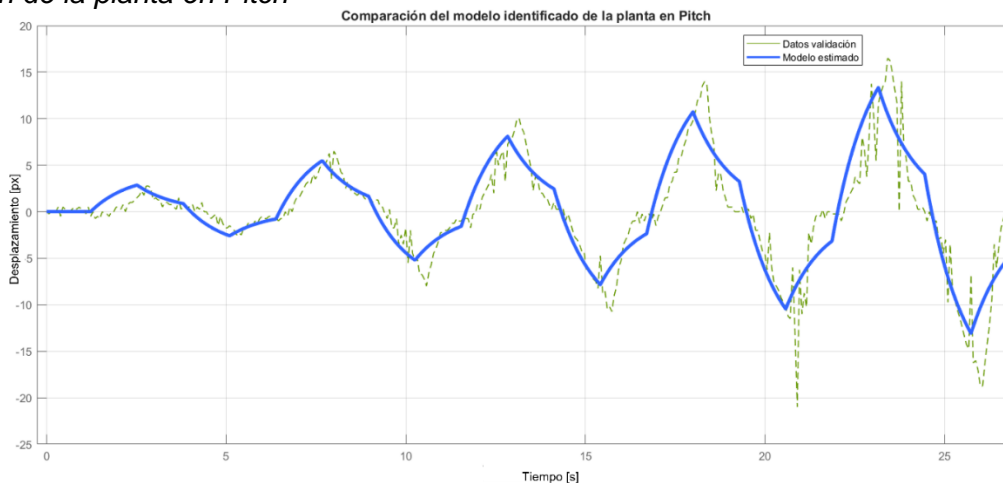
$$G(s) = \frac{K}{T_1 s + 1} \quad (25)$$

Obteniendo los siguientes resultados:

$$K = 41.02 \quad (26)$$

$$T_1 = 1.078 \quad (27)$$

En la Figura 36 se muestra la comparación de la estimación del modelo de la planta y los datos obtenidos, para lo cual se puede apreciar que se consigue una estimación adecuada.

Figura 36**Identificación de la planta en Pitch**

Nota. La figura muestra la comparación entre los datos de validación y el modelo estimado.

Diseño del Controlador

En el diseño del controlador para las plantas obtenidas de Roll y Pitch, se considera que este permita tener una respuesta rápida, estable y una consigna en donde el error sea cero (Lopez-Franco, Gomez-Avila, & Alanis, 2017) (Xu, Hu, Liu, Wang, & Zhang, 2020) (Wang, Liu, & Wang, 2019). Mediante la aplicación *PID Tuner* se establecerá el tipo de controlador a usar, el cual será PID, además esta herramienta cuenta con la opción para variar el tiempo de respuesta y el comportamiento transitorio que permitirá que este sea más robusto o agresivo.

Las métricas para ajustar al controlador son las siguientes:

- Tiempo de establecimiento (*Setting Time*) < 1.5 s
- *Overshoot* = 2%

Una vez realizado el ajuste en cada una de las plantas correspondientes se obtienen los siguientes resultados:

En la Figura 37 se observan los valores para cada una de las constantes del controlador obteniendo los siguientes resultados:

$$K_p = 0.061569 \quad (28)$$

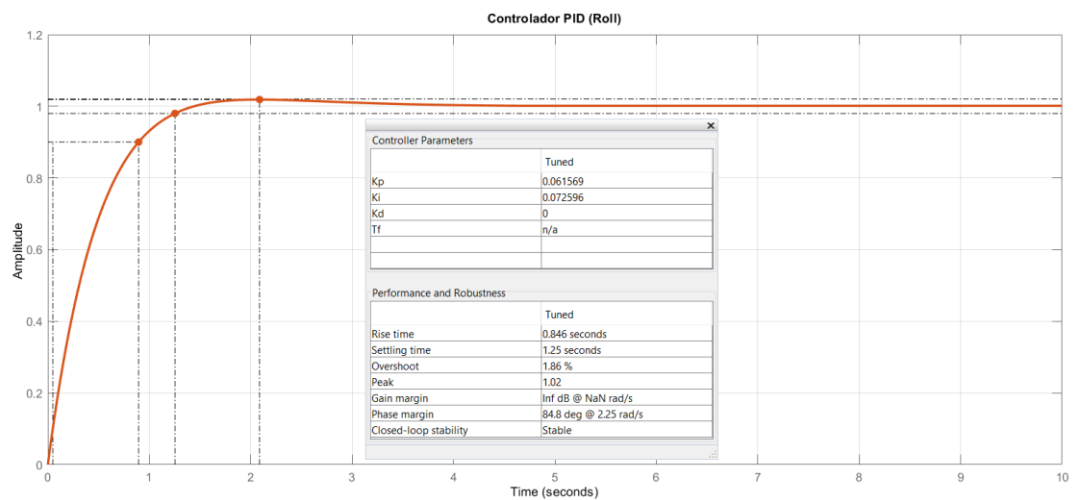
$$K_i = 0.072596 \quad (29)$$

$$K_d = 0 \quad (30)$$

Además, las métricas que fueron impuestas para el diseño del controlador PID se encuentran dentro del rango, lo que permite una respuesta adecuada del sistema.

Figura 37

Respuesta al escalón del controlador PID para Roll



Nota. La figura muestra el controlador PID de Roll, y sus respectivos parámetros.

En la Figura 38 se pueden observar los valores para cada una de las constantes del controlador:

$$K_p = 0.054676 \quad (31)$$

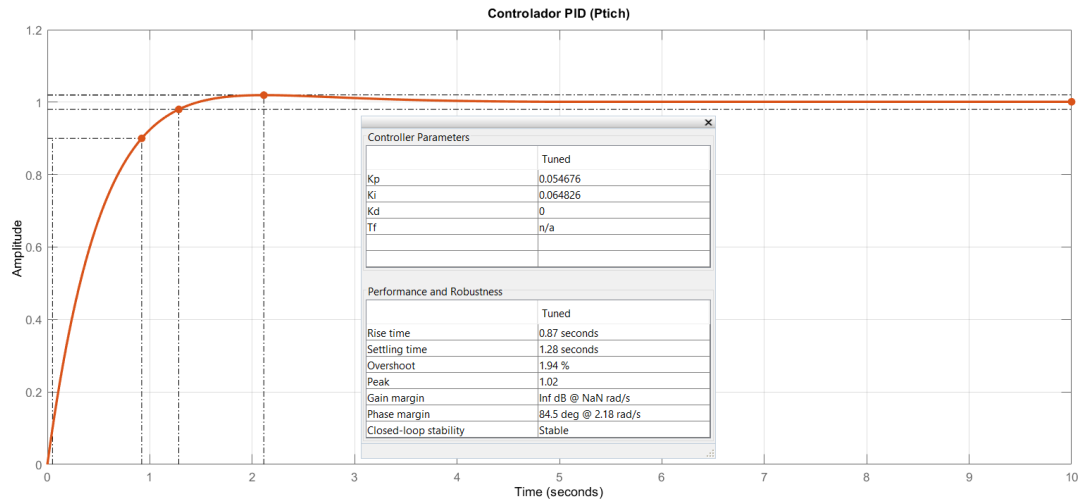
$$K_i = 0.064826 \quad (32)$$

$$K_d = 0 \quad (33)$$

Además, las métricas que fueron impuestas para el diseño del controlador PID se encuentran dentro del rango lo que permite una respuesta adecuada del sistema.

Figura 38

Respuesta al escalón del controlador PID para Pitch



Nota. La figura muestra el controlador PID de Pitch, y sus respectivos parámetros.

Implementación de Controlador para Posicionamiento del micro-UAV y Acción de Aterrizaje

Para realizar el aterrizaje automático del micro-UAV, se ha diseñado un controlador PID, en el cual se establece el *set point* (SP), este será el píxel localizado en el centro de la imagen tanto como para Roll y Pitch, de tal manera el *set point* y error son:

$$SP_{roll} = 428 [px] \quad (34)$$

$$Error_{roll} = c_x - 428 \quad (35)$$

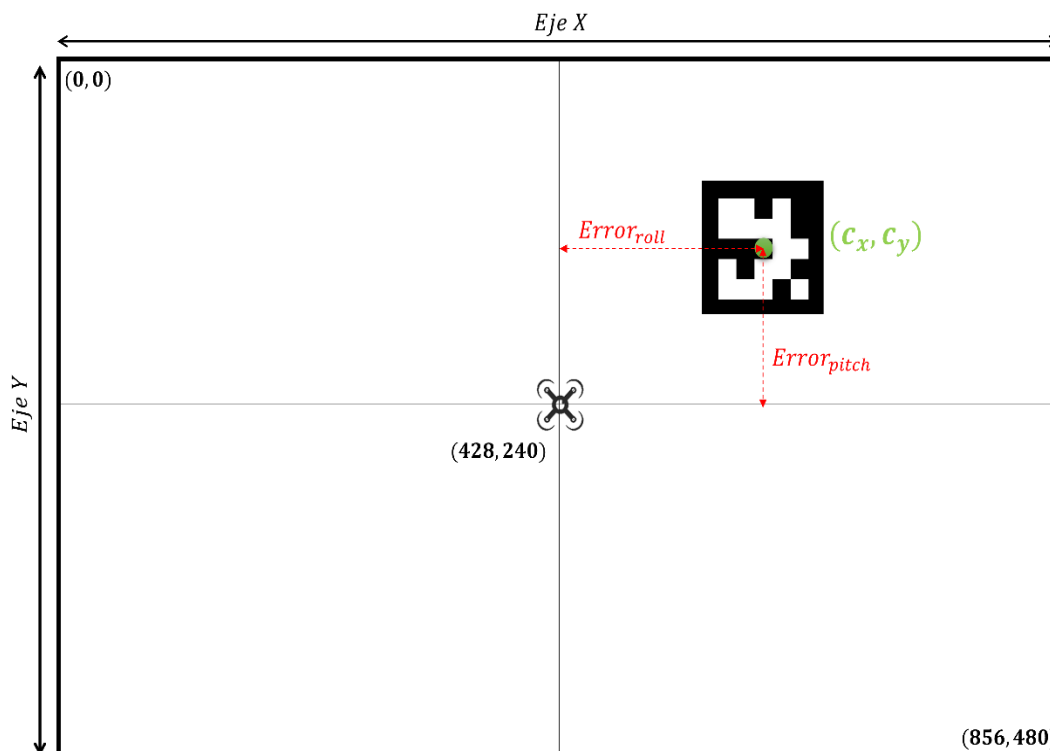
$$SP_{pitch} = 240 [px] \quad (36)$$

$$Error_{pitch} = c_y - 240 \quad (37)$$

Al trabajar con píxeles se debe considerar la altura a la cual se está realizando el control para el aterrizaje ya que de esto depende la escala de trabajo en la detección del marcador y la extracción de los valores para el control del sistema.

Figura 39

Error de posición del marcador respecto a la imagen del micro-UAV



Nota. La figura muestra el error del controlador y cómo reaccionará mediante la acción de control.

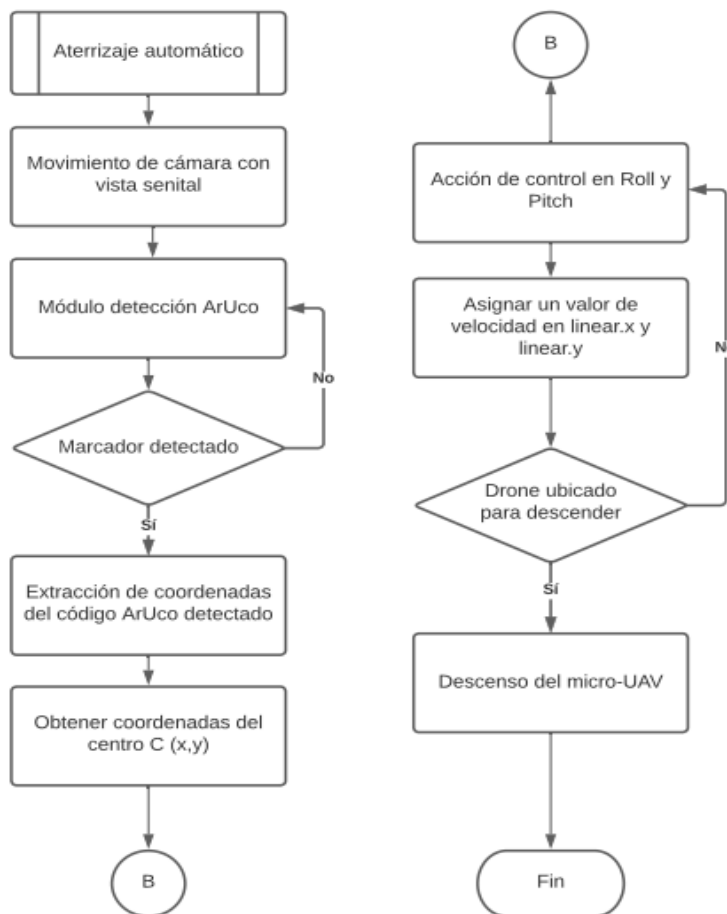
En la Figura 39 se determina el error que será la entrada del controlador que efectuará la acción de control. Por otro lado, la ilustración representa una imagen cenital vista desde el micro-UAV en donde el punto central del marcador será la consigna establecida.

En la Figura 40 se muestra un diagrama de flujo que corresponde al aterrizaje automático que deberá realizar el micro-UAV, este proceso se efectuará siempre y cuando se

haya detectado el código ArUco y este marcador esté dentro de la biblioteca utilizada en este trabajo.

Figura 40

Diagrama de flujo para el aterrizaje con respecto al marcador



Nota. El diagrama de flujo muestra la rutina que debe seguir el micro-UAV para aterrizar en la mitad del código ArUco de la zona de aterrizaje.

Planificación de Trayectoria

En este apartado se menciona la metodología utilizada para la navegación del micro-UAV (Tenezaca, Canchignia, Aguilar, & Mendoza, 2019), en este caso será por GPS. A

continuación, se detallan los diferentes parámetros a tomar en cuenta (Borowczyk, Nguyen, Nguyen, Nguyen, Saussié, & Le, 2017) (Jung, Bang, & Lee, 2015).

Navegación Autónoma por GPS

En este apartado se mencionará la metodología desarrollada para la navegación autónoma GPS del micro-UAV la cual ha sido basada en (Merizalde, Aguilar, & Calderón, 2020) y (Veness, 2020).

Por lo tanto, se realiza el siguiente procedimiento para efectuar la navegación a un punto establecido.

1. Establecer un punto en coordenadas GPS, proporcionando los parámetros de:

$$\varphi_1 = \textit{Coordenada de latitud final}$$

$$\theta_1 = \textit{Coordenada de longitud final}$$

$$h = \textit{altura}$$

2. Obtener los parámetros de posicionamiento GPS del micro-UAV, esto se lo realiza a través del tópico “NavSatFix”, que a través de su sensor permite acceder a dichos parámetros que nos permitirán establecer lo siguiente:

$$\varphi = \textit{Coordenada de latitud inicial}$$

$$\theta = \textit{Coordenada de longitud inicial}$$

3. Realizar el cálculo de la distancia de arco, esta debe calcularse con los parámetros en radianes.

$$d = \cos^{-1}(\sin \varphi \sin \varphi_1 + \cos \varphi \cos \varphi_1 \cos(\theta_1 - \theta)) \quad (38)$$

4. Y por último se emplea la fórmula para obtener el ángulo de rumbo

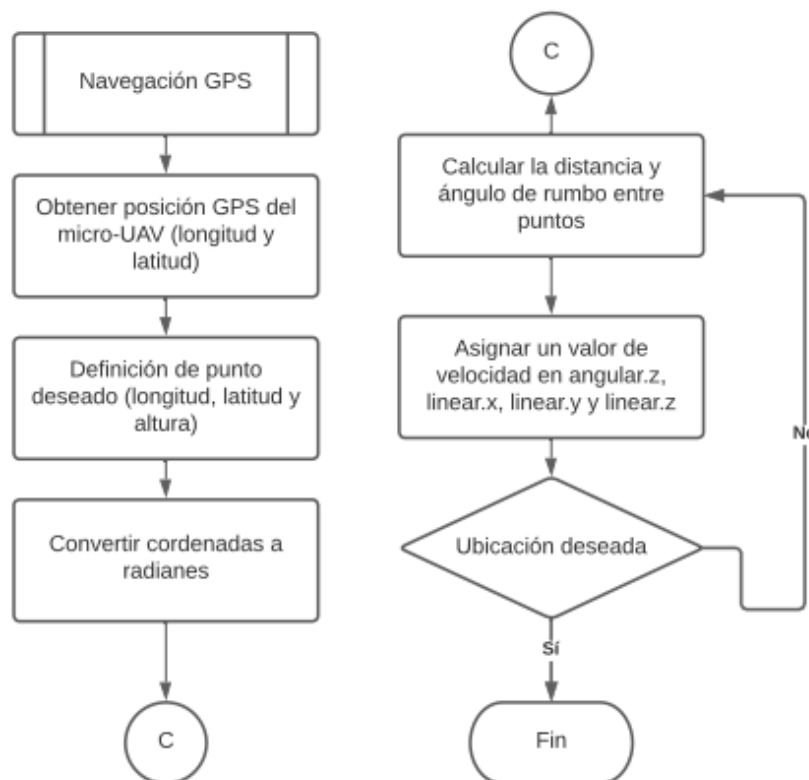
$$\alpha = \cos^{-1} \left(\frac{\sin \varphi - \sin \varphi_1 \cos d}{\cos \varphi_1 \sin d} \right) \quad (39)$$

Luego se implementa el algoritmo que permite realizar un control proporcional para la navegación autónoma a un punto GPS deseado, dicho control fue realizado a partir de esta configuración debido a que al hacer uso del GPS se tienen perturbaciones que afectan en la precisión de establecimiento de coordenadas de este sistema. El error que presenta estas coordenadas teóricamente se encuentra alrededor de los 10 m y dependerá de otros factores como los satélites e interferencias.

En la Figura 41 se muestra un diagrama de flujo que corresponde a la navegación autónoma por medio de GPS que deberá realizar el micro-UAV, este proceso se efectuará siempre y cuando se reciba la señal GPS del micro-UAV.

Figura 41

Diagrama de flujo para la navegación autónoma mediante GPS



Nota. La figura muestra la rutina del micro-UAV para realizar su trayectoria mediante GPS.

Trayectoria aleatoria de ruta GPS

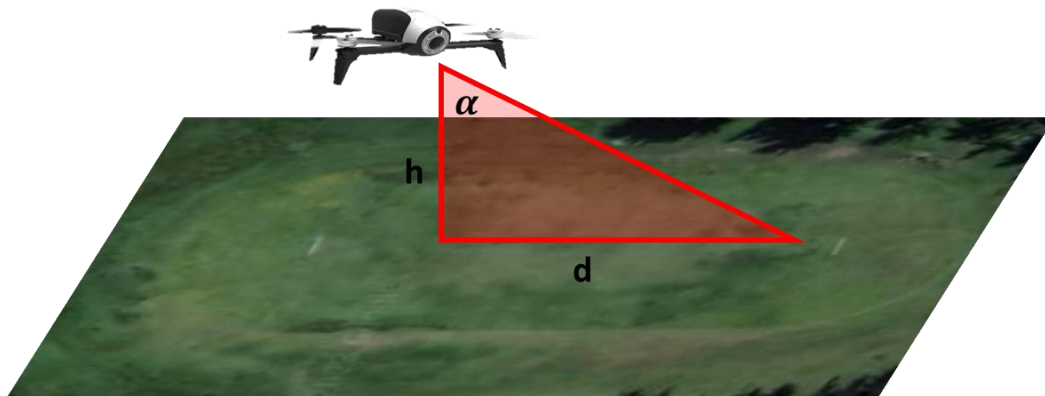
En este apartado se menciona cómo se realiza la trayectoria (Aguilar & Morales, 2016) que efectuará el sistema de detección de intrusos. Es importante mencionar también que existen distintos algoritmos para la planificación de trayectorias dependiendo del enfoque del problema:

- Si el objetivo es dirigirse hacia zonas inexploradas pero que sean libres de colisiones, se usan algoritmos de árboles de exploración rápida (RRT, Rapidly Exploring Random Tree) (Cabras, Rosell, Pérez, Aguilar, & Rosell, 2011).
- Si se busca el camino más corto entre dos puntos se usa el algoritmo A* (A-start) (Hart, Nilson, & Raphael, 1986) (Tierra, F, & Aguilar, 2017).
- O para barridos de zonas un algoritmo basado en mapas probabilísticos (PRM) es ampliamente utilizado (Carrera, Merizalde, Calderón, & Aguilar, 2020).

En la propuesta actual primero se realiza un análisis del rango de visión del micro-UAV. En la Figura 42 se puede visualizar el rango de detección de la cámara incorporada en el micro-UAV.

Figura 42

Rango de detección



Nota. La figura muestra el rango de detección del micro-UAV, esto mediante el Teorema de Pitágoras.

La Ecuación 40 representa la relación entre el ángulo de visión del micro-UAV (α), la altura a la que se encuentra navegando (h) y la distancia que tiene de visión (d).

$$\tan \alpha = \frac{d}{h} \quad (40)$$

Donde el ángulo es de 45° y el micro-UAV realizará la navegación a 10 metros, se calcula la distancia d .

$$d = h * \tan \alpha \quad (41)$$

$$d = 10 [m] \quad (42)$$

Figura 43

Zona de vuelo



Nota. La figura muestra el espacio donde se seleccionarán cuatro puntos para que el micro-UAV los busque y sea capaz de realizar un barrido de toda el área seleccionada.

Posterior a esto se definen dentro del lugar establecido cuatro puntos en coordenadas de longitud y latitud, teniendo en cuenta el área de visión del micro-UAV para que sea capaz de realizar un barrido completo del área.

La Ecuación 43 representa el número de combinaciones que se pueden generar de las rutas, siendo (n) el número de coordenadas geográficas designadas y (m) el tamaño de la combinación a realizarse.

$$P_n^m = \frac{m!}{(m-n)!} \quad (43)$$

Donde el número de coordenadas son 4 y las combinaciones deseadas están conformados por todos los puntos establecidos también son 4, se calcula las combinaciones posibles a realizar por el micro-UAV.

$$P_4^4 = \frac{4!}{(4-4)!} \quad (44)$$

$$P_4^4 = 24 \quad (45)$$

De tal manera el sistema puede presentar 24 diferentes rutas con la finalidad de realizar un barrido de toda el área definida.

La metodología desarrollada para realizar la aleatoriedad se lo realizó de la siguiente manera:

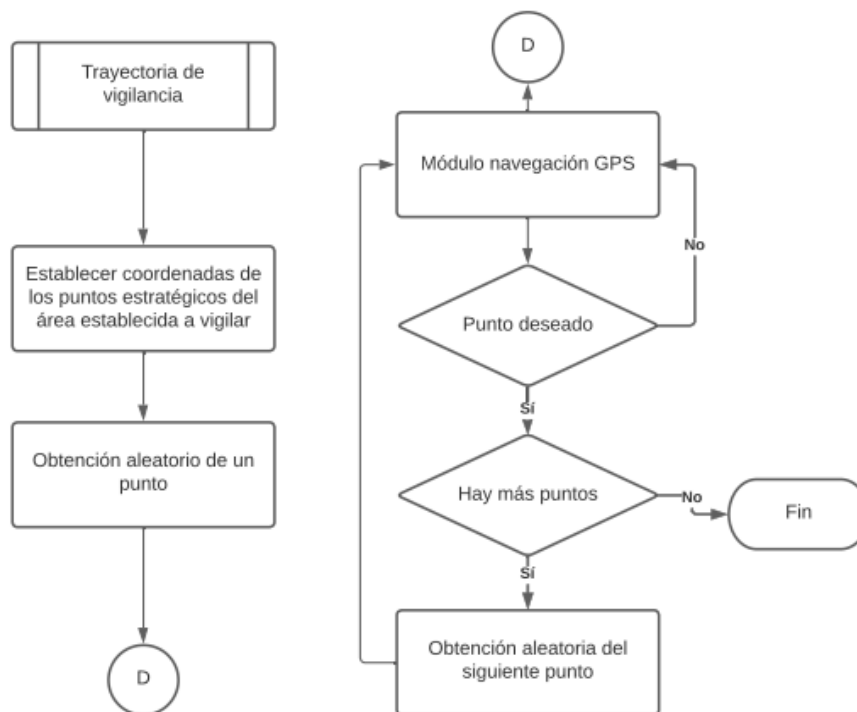
1. Se definen como constantes las 4 coordenadas geográficas en longitud y latitud.

2. Con la función random se establece la primera coordenada al cuál se va a dirigir el micro-UAV.
3. Se descarta la coordenada ya navegada y nuevamente haciendo uso de la función random se establece el segundo punto al cual se debe dirigir.
4. Descartamos la coordenada ya navegada, y con la función random establecemos el tercer punto y se realiza la navegación autónoma.
5. Finalmente, ya nos dirigimos a la coordenada restante.

En la Figura 44 se muestra un diagrama de flujo que corresponde al algoritmo que se realizará en la trayectoria de vigilancia del micro-UAV, este proceso se efectuará siempre y cuando se reciba la señal GPS del micro-UAV.

Figura 44

Diagrama de flujo para la trayectoria de vigilancia del micro-UAV



Nota. El diagrama de flujo muestra la selección de puntos para la trayectoria del micro-UAV.

Capítulo V. Pruebas y Resultados

Al finalizar el sistema de videovigilancia en planificación aleatoria y detección de intrusos, se han realizado varias pruebas con el fin de estimar el desempeño de este sistema. Por tal motivo dentro de las pruebas se considerarán distintos parámetros como las condiciones de viento y luminosidad con el objetivo de verificar el funcionamiento del sistema.

Por lo tanto, el entorno dentro del cual se llevarán a cabo las pruebas es el que se describió en el Capítulo IV (estadio de la Universidad de las Fuerzas Armadas ESPE), los parámetros mencionados con anterioridad se logran al realizar las pruebas en distintos horarios en el transcurso del día, permitiendo así variación en los valores de luminosidad y velocidad del viento, que son factores influyentes en la detección de personas y especialmente en la navegación del micro-UAV.

La estación terrestre fue desarrollada en una laptop MSI Katana F66 con un procesador Intel Core i7, memoria RAM de 16 GB y una tarjeta gráfica NVIDIA GeForce RTX 3060. El micro-UAV empleado fue el *Parrot* Bebop 2 que cuenta con un sistema de 4 rotores, una cámara tipo ojo de pez de 14 megapíxeles y un procesador dual-core con GPU quad-core.

Trayectoria de navegación por GPS

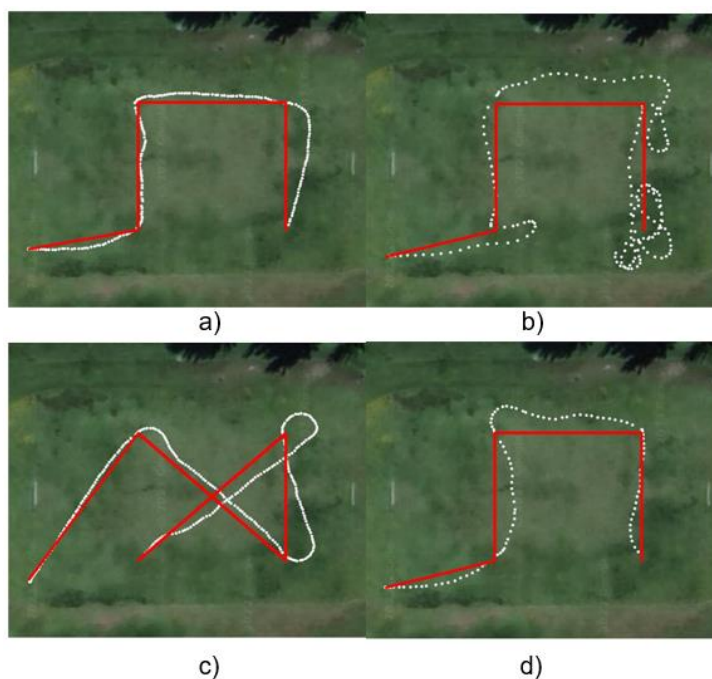
El objetivo de estas pruebas es establecer el error cuadrático medio (RMSE) del sistema de navegación por GPS. Entonces, se obtienen los datos de la ruta que realiza el micro-UAV y la ruta ideal considerando algunas variables para establecer los mejores resultados, los parámetros que serán modificados como la altura a la cual realiza la ruta el micro-UAV, la ruta de trayectoria y la velocidad del micro-UAV (Restrepo, 2022).

En la Figura 45 se presentan varias trayectorias realizadas por el micro-UAV, en donde estas fueron modificadas en altura, ruta y velocidad de movimiento. Se visualizan pequeños puntos blancos que representan las coordenadas GPS que fueron obtenidas y plasmadas en la

imagen representativa. Se efectuaron las pruebas de tal manera que se identifique la trayectoria realizada, además se visualiza una línea de color rojo que representa la trayectoria ideal que se debería seguir. En función a esto se establecerá el error existente entre estos dos conjuntos de datos (Merizalde, Aguilar, & Calderón, 2020).

Figura 45

Trayectoria de navegación aleatoria por GPS.



Nota. La figura muestra cuatro trayectorias. a) es una trayectoria realizada a 10 metros y una velocidad de 0.1 [m/s], b) a 20 metros y una velocidad de 0.3 [m/s], c) a 15 metros y una velocidad de 0.1 [m/s] y finalmente d) a 15 metros y una velocidad de 0.2 [m/s]

Haciendo uso de Matlab se establece el valor RMSE de la trayectoria que realiza el micro-UAV, los datos obtenidos se muestran en la Tabla 10.

Las variables inmersas en el RMSE son los valores teóricos (VT_i) haciendo referencia a aquellos datos de los pixeles de la ruta deseada, los valores prácticos (VP_i) haciendo referencia

a los datos de los píxeles de la trayectoria real del micro-UAV, y el número total de datos (n), la fórmula para calcular esta métrica es la siguiente:

$$RMSE = \sqrt{\frac{\sum(VP_i - VT_i)^2}{n}} \quad (46)$$

Tabla 10

RMSE de la trayectoria realizada por el micro-UAV.

Trayectoria	Velocidad [m/s]	RMSE		
		10 metros	15 metros	20 metros
1	0.1	5.41	7.32	7.44
1	0.2	10.15	8.21	9.16
1	0.3	14.08	13.95	15.42
2	0.1	4.96	6.85	7.95
2	0.2	8.47	11.25	7.98
2	0.3	12.95	17.54	15.48
3	0.1	6.78	5.76	8.32
3	0.2	7.82	8.87	9.41
3	0.3	13.54	16.23	14.89
4	0.1	4.72	4.85	7.74
4	0.2	9.22	8.86	10.27
4	0.3	15.37	15.24	16.83

Nota. La tabla muestra los datos obtenidos en las pruebas realizadas a 10, 15 y 20 metros, a velocidades de 0.1, 0.2 y 0.3 [m/s].

En función a los datos de la Tabla 10 se presenta el RMSE promedio, que es de 10.25, este valor es adimensional y se lo toma como referencia a la dispersión de los datos en píxeles, siendo así que el sistema presenta una dispersión de 10.25 px al realizar su trayectoria.

Estos valores fueron obtenidos de una imagen de 632 x 522 px, en donde el error que se presenta en la trayectoria es de 1.77%, siendo óptimo para el sistema; tomando en cuenta que la ruta que realiza el micro-UAV está sujeta a factores externos como es el viento en consecuencia se desvía de su trayectoria ideal.

Detección de Intrusos (Navegación Fina)

El objetivo de estas pruebas es establecer la fiabilidad del sistema para detectar si existen o no intrusos, es decir, que exista una persona o no al momento de realizar la trayectoria de navegación en el área establecida, para lo cual se ha tomado como referencia a (Calderón, Aguilar, & Merizalde, 2020) para establecer dos métricas en estas pruebas, sensibilidad y especificidad.

Las variables inmersas en la sensibilidad son los verdaderos positivos (VP) haciendo referencia a aquellos objetos identificados como intrusos (personas) y la identificación es correcta, y falsos negativos (FN) a aquellos objetos que no han sido identificados como intrusos (persona) y la identificación es incorrecta, la fórmula para calcular esta métrica es la siguiente:

$$Sensibilidad = \frac{VP}{VP + FN} \quad (47)$$

También se debe evaluar la métrica correspondiente a la especificidad en la cual las variables inmersas son los verdaderos negativos (VN) haciendo referencia a aquellos objetos que no fueron identificados como intrusos (persona) y la identificación es correcta, y falsos

positivos (FP) objetos (personas) que fueron identificados como intrusos y la identificación es incorrecta, siendo así la fórmula para calcular esta métrica es la siguiente:

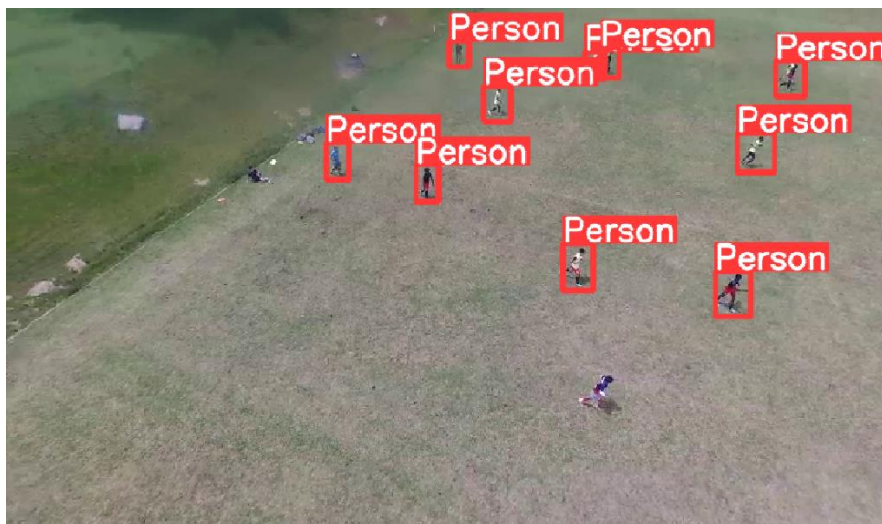
$$Especificidad = \frac{VN}{VN + FP} \quad (48)$$

De tal manera se procede a obtener aleatoriamente algunos *frames* cuando se realiza la detección de intrusos durante la navegación del micro-UAV y considerando en las distintas pruebas las variables como la altura a la cuál realiza la detección el micro-UAV y la luminosidad que varía al transcurrir el día. Además, que los objetos de prueba son jugadores de fútbol, estudiantes y entrenadores, lo que permite tener una variabilidad en cuanto al objeto que se detecta y también determinar la confiabilidad del sistema.

En la Figura 46 la cuál fue extraída a 10 metros se puede visualizar la detección de los jugadores durante un entrenamiento, las cuales se mueven a una velocidad considerable ya que se encuentran corriendo. Obteniendo verdaderos positivos (10), falsos negativos (2), verdaderos negativos (1) y falsos positivos (0). Así mismo se realiza con las distintas imágenes obtenidas para establecer el valor de sensibilidad y especificidad del sistema para detectar intrusos.

Figura 46

Sistema de detección de intrusos.



Nota. La figura muestra jugadores de fútbol en el espacio designado para el recorrido del micro-UAV.

Aplicando el mismo análisis anterior a cada una de las imágenes obtenidas en las distintas pruebas se procede a establecer el valor de sensibilidad y especificidad del sistema de detección de intrusos, en la Tabla 11 se puede observar los valores obtenidos.

Tabla 11

Sensibilidad y especificidad en la detección de intrusos.

Prueba	Iluminación (lux)	10 metros		15 metros		20 metros	
		Sensibilidad	Especificidad	Sensibilidad	Especificidad	Sensibilidad	Especificidad
1	6758 (3pm)	83.33 %	100 %	88.88%	90%	87.50%	75%
2	32767 (12pm)	94.11 %	80 %	92.30%	75%	100%	80%
3	5000 (10am)	100 %	-	80%	100%	86.67%	100%

4	1083 (8am)	94.44%	100%	100%	80%	90%	-
Media	-	92.97 %	93.33%	90.3%	86.25	91%	85%

Nota. Esta tabla muestra la altura vs la iluminación.

Tras finalizar las pruebas teniendo en cuenta factores como la iluminación en el transcurso del día y la altura en las cuales fueron obtenidas las imágenes, el sistema presenta el 91.42% de sensibilidad y 88.19% de especificidad, lo cual tomando como referencia el trabajo de (Merizalde, Aguilar, & Calderón, 2020) se superan los valores, permitiendo que el sistema de detección de intrusos sea adecuado en su aplicación.

Aterrizaje Automático (Navegación Fina)

El objetivo de estas pruebas es establecer la precisión que presenta el sistema respecto al aterrizaje automático que realiza el micro-UAV, en el cual se realizaron 10 pruebas, consisten en que el micro-UAV realiza todas las tareas estipuladas anteriormente sujetas a pruebas (navegación GPS y detección de intrusos) y finalmente aterriza en la zona en la que se encuentra ubicado el marcador (Código ArUco) (Chasillacta, 2020).

El error que se obtendrá se toma respecto al centro del marcador y el centro del micro - UAV la cual será la distancia entre dichos puntos, para poder establecer la precisión del sistema respecto al aterrizaje la métrica establecida parte de la trayectoria realizada por el micro-UA.

En la Figura 47 se presentan distintas tomas realizadas al finalizar el aterrizaje automático del micro-UAV, en las se procederá a realizar las mediciones y establecer la precisión del sistema.

Las variables inmersas en la precisión del sistema es el verdadero total (VT) haciendo referencia a la trayectoria total recorrida por el micro-UAV y el valor de aterrizaje (VA) haciendo referencia a la distancia existente entre el centro del marcador y el centro del micro-UAV, la fórmula para calcular esta métrica es la siguiente:

$$\text{Precisión} = \frac{VT - VA}{VT} * 100\% \quad (49)$$

Figura 47

Tomas aterrizaje automático micro-UAV



a)



b)



c)



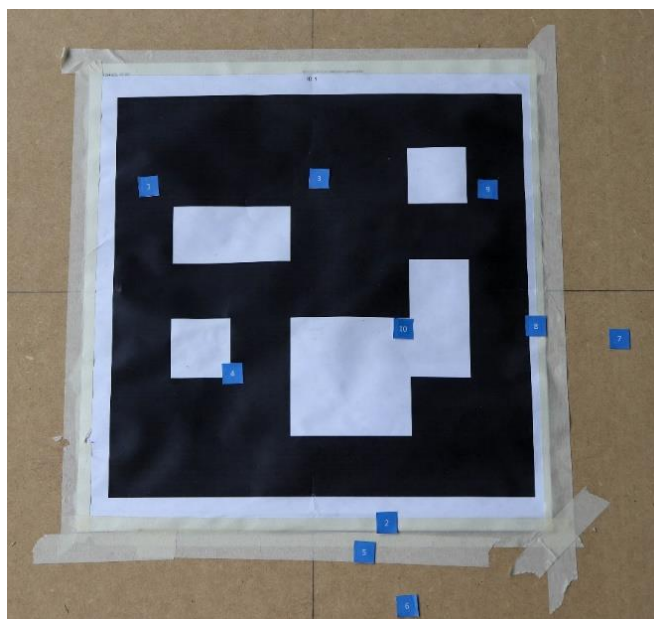
d)

Nota. Se muestra cuatro tomas de las pruebas realizadas. Se puede observar que en a, b y d el micro-UAV aterriza a pocos centímetros del centro mientras que, en c, el micro-UAV se alejó un porcentaje más del centro, esto debido a la velocidad del viento que se tuvo en esa prueba.

A cada prueba realizada, se colocó un identificador que permita conocer el punto central del micro-UAV y el centro del marcador esto con el fin de establecer la distancia que existía entre ellos, como se muestra en la Figura 48.

Figura 48

Identificadores de las pruebas de aterrizaje automático del micro-UAV



Nota. Las viñetas azules muestran los puntos donde el micro-UAV aterrizó, esto con el propósito de encontrar el error.

Los datos obtenidos de la distancia se presentan en la Tabla 12, así como el valor calculado de la precisión del sistema respecto a un aterrizaje mediante GPS.

Tabla 12

Datos obtenidos del aterrizaje automático del sistema

ATERRIZAJE AUTOMÁTICO

Prueba	Distancia [m]	Trayectoria [m]	Precisión [%]
1	0.13	195	99.93
2	0.15	215	99.93
3	0.08	195	99.96
4	0.07	200	99.97
5	0.16	215	99.93
6	0.20	195	99.90
7	0.19	205	99.91
8	0.14	215	99.93
9	0.12	195	99.94
10	0.06	195	99.97
Total	0.13	202.5	99.93

Nota. Datos que permiten saber la distancia del centro del micro-UAV y el centro del marcador para el aterrizaje.

El valor de precisión del sistema al finalizar su trayectoria de navegación presenta una precisión del 99.93 %, cumpliendo con los objetivos del proyecto desarrollado.

También se procede a comparar el sistema de aterrizaje utilizando el controlador PID y el aterrizaje por medio de GPS, el cual es el otro método que permite realizar el aterrizaje estableciendo una coordenada, dirigirse y realizar el descenso del micro-UAV, este valor es de 10 metros debido a que depende de las interferencias al obtener la información del satélite.

Tabla 13

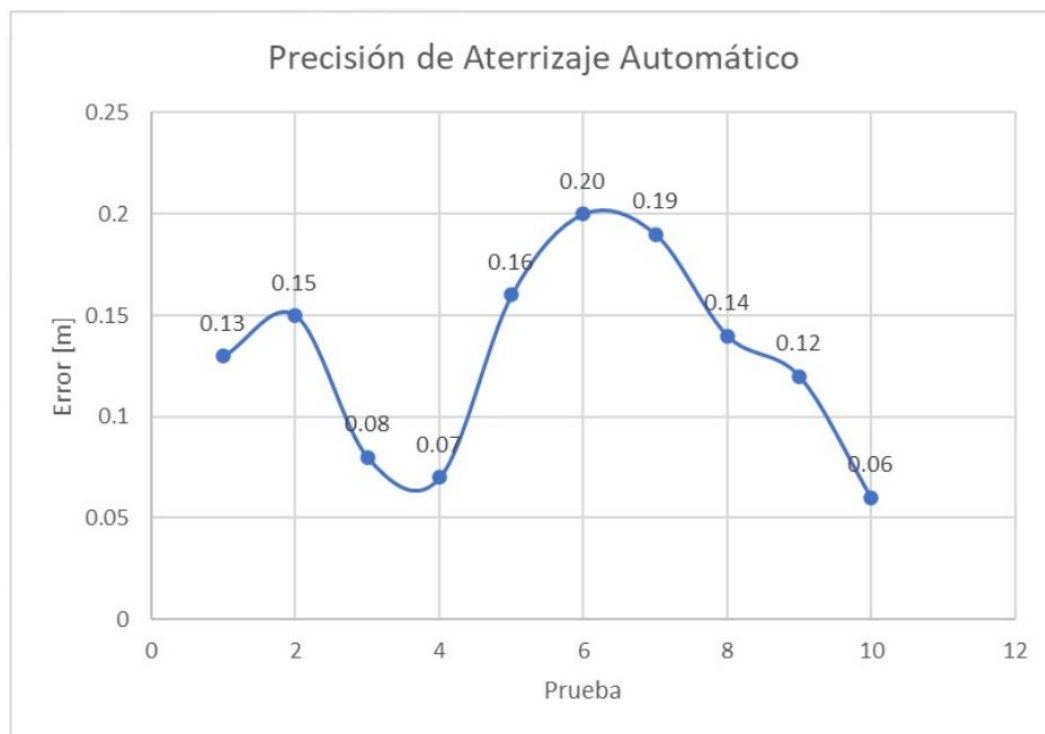
Comparación del sistema de aterrizaje

COMPARACIÓN ATERRIZAJE

Prueba	Distancia [m]	Precisión [%]
1	0.13	98.7
2	0.15	98.5
3	0.08	99.2
4	0.07	99.3
5	0.16	98.4
6	0.20	98
7	0.19	98.1
8	0.14	98.6
9	0.12	98.8
10	0.06	99.4
Total	0.13	98.7

Figura 49

Dispersión del error del aterrizaje automático del micro-UAV



Nota. Se muestra gráficamente la dispersión de 10 pruebas realizadas con respecto al aterrizaje.

En la Figura 49 se visualiza la dispersión que presenta el sistema en las 10 pruebas realizadas, también se observa un valor máximo de error respecto al centro del marcador, esto debido a las condiciones del viento una vez que el micro-UAV apaga sus motores para el descenso, así como también el movimiento propio producido de su aterrizaje (brusco).

Finalmente, el sistema presenta un 98.7% de precisión, este sistema cuenta con un controlador PID basado en visión artificial, en el que se tiene como entrada del sistema la imagen del marcador (código ArUco). Con respecto al aterrizaje mediante GPS, el valor cumple con los objetivos porque se tiene una reducción del error, en donde se obtuvo un porcentaje menor al 2%.

Conclusiones y Recomendaciones

Conclusiones

En función a las investigaciones respecto a las técnicas empleadas basadas en visión por computadora (mediante la cámara incorporada en el micro-UAV), se desarrolló e implementó un sistema de videovigilancia en vista que se deben tomar medidas y generar aplicaciones para combatir la inseguridad que como país estamos atravesando, debido al malestar que ha vulnerado los derechos de los habitantes. Este sistema consta de navegación autónoma y aterrizaje automático, enfocadas en un área exterior delimitada, seleccionando el estadio de la Universidad de las Fuerzas Armadas como el lugar donde se efectuarán las pruebas. Se emplearon técnicas de detección de objetos, navegación por GPS y sistemas de control, dando como resultado este sistema, que representa una parte de la solución de inseguridad que atraviesa la sociedad ecuatoriana.

La comunicación entre la estación terrestre y el vehículo aéreo se llevó a cabo mediante WiFi, en la estación terrestre se utilizó una laptop MSI Katana F66, la cual cuenta con una tarjeta gráfica que permite procesar la imagen obtenida del micro-UAV *Parrot Bebop 2*. Para obtener los parámetros del micro-UAV y realizar el control se llevó a cabo la comunicación maestro-esclavo mediante ROS, a través de sus distintos nodos que tienen la capacidad de suscribirse o publicar la información necesaria para las distintas etapas desarrolladas.

El sistema de planificación de trayectoria se implementó en función del área a vigilar, por lo cual se estableció coordenadas geográficas en distintos puntos y en función a estos se generaba la trayectoria a realizar por el micro-UAV, en la trayectoria se calcula constantemente el ángulo de rumbo y la distancia para generar las acciones de control ya que por las perturbaciones presentadas (viento) es necesario corregir cada cierto tiempo la trayectoria a seguir.

El sistema de detección de intrusos fue implementado mediante YOLO en su versión 8 con un modelo personalizado el cual presenta mejores resultados en comparación a trabajos anteriores desarrollados con YOLOv4 o YOLO v3. El procesamiento fue realizado mediante una tarjeta gráfica NVIDIA GeForce RTX 3060, presentando resultados del 91.42% en sensibilidad y 88.19% en especificidad, esto se debe a que se utilizó PyTorch como marco de aprendizaje por su mayor enfoque a objetos, a diferencia de trabajos anteriores que hacen uso de TensorFlow.

El sistema de aterrizaje automático se desarrolló en función de la detección y estimación respecto a un marcador (código ArUco), implementando dos controladores PID para los movimientos de Pitch y Roll, para lo cual previamente se obtuvo el modelo de la planta mediante la estimación de movimiento entre *frames*. El controlador se encarga de posicionar al micro-UAV en el centro respecto al marcador, posterior a alcanzar el *set point* establecido se realiza el descenso del micro-UAV, se tomó en cuenta las perturbaciones que presenta el sistema al descender de una altura de 6 metros, por lo cual se desciende el micro-UAV a una altura de 80 centímetros y se reubica el micro-UAV en el centro y finalmente se envía la acción de aterrizaje. Presentado el sistema una precisión de 99.93% en función a la trayectoria realizada, y en comparación a un aterrizaje mediante GPS de 98.7%.

El sistema fue implementado en su totalidad, lo que permite al micro-UAV *Parrot Bebop 2*, realizar las distintas tareas de navegación autónomas mediante GPS, detección de intrusos y aterrizaje automático, en donde el sistema fue sometido a distintos cambios de iluminación, altura de vuelo, velocidad de vuelo y trayectoria de navegación. Dando como resultado un error menor al 3% en sus distintas tareas lo cual está relacionado a perturbaciones externas.

Recomendaciones

Se recomienda realizar un mantenimiento preventivo de la parte mecánica del micro-UAV que incluya batería, hélices, motores, estructura. Permitiendo tener un desempeño óptimo

al momento de realizar la navegación de este y prevenir daños por anomalías que se presenten durante el vuelo.

Tener en cuenta el desempeño de la batería, ya que este tiene un comportamiento exponencial, entonces se recomienda establecer una lógica de control que permita al micro-UAV siempre regresar al punto de partida o punto deseado en caso de que el valor de batería se encuentre debajo del requerido para realizar las acciones definidas. En base a las pruebas realizadas se recomienda que el micro-UAV vuelva antes de alcanzar el diez por ciento de su batería.

Considerar la antena de WiFi de la estación terrestre ya que esta limita el área de conexión del micro-UAV, influyendo en la navegación GPS de las coordenadas deseadas, en el trabajo la distancia máxima fue de 300 metros en la cual se podía interactuar.

Se recomienda entrenar el modelo de detección de objetos con un data set personalizado relacionado a la aplicación que se va a definir, teniendo en cuenta parámetros como ángulo de visión, distancia, iluminación, entre otros, lo que va a permitir una sensibilidad y especificidad alrededor del 100%.

Bibliografía

- Aguilar, W. G., & Angulo, C. (2015). Real-Time Model-Based Video Stabilization for Microaerial Vehicles. . *Neural Processing Letters*, XLIII (2), 459–477.
<https://doi.org/10.1007/s11063-015-9439-0>
- Aguilar, W. G., & Morales, S. (2016). 3D Environment Mapping Using the Kinect V2 and Path Planning Based on RRT Algorithms. *Electronics*, 5(4), 70.
- Aguilar, W. G., Casaliglla, V., & Pólit, J. (2017). Obstacle Avoidance Based-Visual Navigation for Micro Aerial Vehicles. *Electronics*, 6(1), 10.
- Aguilar, W. G., Costa-Castelló, R., & Angulo, C. (2017). Autonomous Navigation Control for Quadrotors in Trajectories Tracking. *Lecture Notes in Computer Science*, 287-297.
- Angulo, C., & Aguilar, W. (2013). Estabilización robusta de vídeo basada en diferencia de nivel de gris. *VIII Congreso de Ciencia y Tecnología ESPE*.
- Anónimo. (2022). *Zero to Mastery Learn PyTorch for Deep Learning*. Zero to Mastery Learn PyTorch for Deep Learning: https://www.learnpytorch.io/01_pytorch_workflow/
- Antenucci, A., Mazzaro, S., Fiorilla, A., Messina, L., Massa, A., & Matta, W. (2021). A ROS Based Automatic Control Implementation for Precision Landing on Slow Moving Platforms Using a Cooperative Fleet of Rotary-Wing UAVs. *5th International Conference on Robotics and Automation Engineering (ICRAE)*. Singapore: IEEE.
<https://doi.org/10.1109/ICRAE50850.2020.9310899>
- Antichán, M., Morán, J., & Núñez, S. (2009). Global Positioning System Applied to Intelligent Transit for Emergency Agencies. *Revista Electrónica de Estudios Telemáticos*, XIII(2), 56-69.

- Asimbaya, I., Albán, P., Aguilar, W., & Fernández, Y. (2020). Ackermann UGV with 2D Mapping for Unknown Environments. En *International Conference of Research Applied to Defense and Security* (págs. 45-53).
- Awaludin, M., & Yasin, V. (2020). APPLICATION OF ORIENTED FAST AND ROTATED BRIEF (ORB) AND BRUTEFORCE HAMMING IN LIBRARY OPENCV FOR CLASSIFICATION OF PLANTS. *Journal of Information System, Applied, Management, Accounting and Research, IV(3)*, 51-59.
- Basantes, J., Godoy, L., Carvajal, T., Castro, R., Toulkeridis, T., Fuertes, W., . . . Addison, A. (2018). Capture and processing of geospatial data with laser scanner system for 3D modeling and virtual reality of Amazonian Caves. *IEEE Ecuador Technical Chapters Meeting (ETCM)*. Samborondón, Ecuador.
- Bay, H., Ess, A., Tuytelaars, T., & Van Gool, L. (2008). Speeded-up robust features (SURF). *Computer Vision and Image Understanding, CX(3)*, 346-359.
- Beltrán, D. (2017). *Aplicación de imágenes NDVI para el control de riego y enfermedad en cultivos Agrícolas mediante el uso de aeronaves no tripuladas (UAV) y el software AgVault*. Aporte Santiaguino.
- Benavides, C. (2016). Face Classification by Local Texture Analysis through CBIR and SURF Points. *IEEE Latin America Transactions, XIV(5)*, 2418-2424.
<https://doi.org/10.1109/TLA.2016.7530440>
- Borowczyk, A., Nguyen, D.-T., Nguyen, A., Nguyen, D., Saussié, D., & Le, J. (2017). Autonomous Landing of a Multirotor Micro Air Vehicle on a High Velocity Ground Vehicle. *IFAC-PapersOnLine, L(1)*, 10488-10494.
<https://doi.org/10.1016/j.ifacol.2017.08.1980>

- Caballeros, J., Segarra, D., Castro, P., & Aguilar, W. (2018). RRT Path Planning and Morphological Segmentation Based Navigation for a Tetrapod Robot. *Ubiquitous Networking: 4th International Symposium, UNet* (págs. 273-280). Springer International Publishing.
- Cabero, G., & Barboza, H. (2019). *Content Based Image Retrieval(CBIR) Aplicando descriptor ORB*. Universidad Nacional de Salta.
- Cabras, P., Rosell, J., Pérez, A., Aguilar, W., & Rosell, A. (2011). Haptic-based navigation for the virtual bronchoscopy. *18th IFAC World Congress*. Milano, Italy.
- Caiza, O., Aguilar, W., Albán, P., & Fernández, Y. (2020). Kinect and Manipulator-Based Sample Collection System for Military Robot. En *Smart Innovation, Systems and Technologies* (págs. 75-87).
- Calderón, M., Aguilar, W., & Merizalde, D. (2020). Visual-Based Real-Time Detection Using Neural Networks and Micro-UAVs for Military Operations. En *Smart Innovation, Systems and Technologies* (págs. 55-64).
- Calonder, M., Lepetit, V., Strecha, C., & Fua, P. (2010). Brief: Binary robust independent elementary features. *Computer Vision—ECCV 2010, IV(11)*, 778-792.
- Calonder, M., Lepetit, V., Ozuysal, M., Trzcinski, T., Strecha, C., & Fua, P. (2012). BRIEF: Computing a Local Binary Descriptor Very Fast. *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, 34.
<https://doi.org/10.1109/TPAMI.2011.222>
- Cardoso, E., Fernández, A., Marrero-Osorio, S., & Guardado, P. (2017). Modelos cinemático y dinámico de un robot de cuatro grados de libertad. *SciELO*, XXXVIII(3).

- Carmichael, G., Laganière, R., & Bose, P. (2010). Global Context Descriptors for SURF and MSER Feature Descriptors. *Canadian Conference on Computer and Robot Vision*. Ontario: IEEE. <https://doi.org/10.1109/CRV.2010.47>
- Carrera, A., Merizalde, D., Calderón, M., & Aguilar, W. (2020). Monte Carlo-Based Localization for Kidnapped Robot Problem. En *International Conference of Research Applied to Defense and Security* (págs. 101-109).
- Chasillacta, M. (2020). *Reconocimiento y seguimiento de plataformas para el aterrizaje automático de un vehículo aéreo no tripulado basado en inteligencia artificial y odometría visual*. Universidad de las Fuerzas Armadas ESPE.
- Chasnov, J. (2012). *Álgebra Lineal Aplicada y Ecuaciones Diferenciales*. Hong Kong University of Science and Technology.
- Chauca Vera, B. (2020). *Seguimiento y búsqueda de objetivos en entornos complejos usando micro vehículos aéreos con cámaras monoculares para aplicaciones militares*. Departamento de Eléctrica, Electrónica y Telecomunicaciones.
- Cleary, A., Yoo, A., Samuel, P., George, S., Sun, F., & Israel, A. (2020). Machine Learning on Small UAVs. *IEEE Applied Imagery Pattern Recognition Workshop (AIPR)*. Washington DC: IEEE. <https://doi.org/10.1109/AIPR50011.2020.9425090>
- Cobeña, B., Rodríguez, G., Salcedo, V. S., Collaguazo, B., & Aguilar, W. (2018). SVM and RGB-D Sensor Based Gesture Recognition for UAV Control. *International Conference on Augmented Reality, Virtual Reality and Computer Graphics* (págs. 713-719). Springer.
- Collaguazo, B., Tituaña, J., Aguilar, W., & Amaguaña, F. (2018). Simulation System Based on Augmented Reality for Optimization of Training Tactics on Military Operations.

- International Conference on Augmented Reality, Virtual Reality and Computer Graphics* (págs. 394-403). Springer.
- Conci, A., Falconi, L. G., Perez, M., & Aguilar, W. (2020). Transfer learning and fine tuning in breast mammogram abnormalities classification on CBIS-DDSM database. *Adv. Sci. Technol. Eng. Syst. J*, 154-165.
- Córdova, A., Jiménez, B., Pardo, J., Toalombo, I., & Aguilar, W. (2018). Geolocation and Counting of People with Aerial Thermal Imaging for Rescue Purposes. *International Conference on Augmented Reality, Virtual Reality and Computer Graphics* (págs. 171-182). Springer.
- Cowan, B., Imanberdiyev, N., Fu, C., Dong, Y., & Kayacan, E. (2016). A performance evaluation of detectors and descriptors for UAV visual tracking. *14th International Conference on Control, Automation, Robotics and Vision (ICARCV)*. Phuket: IEEE.
<https://doi.org/10.1109/ICARCV.2016.7838649>
- Cuerno Rejado, C., García Hernández, L., & Sánchez Carmona, A. (Mayo de 2016). Univerisdad Politécnica de Madrid. *Biblioteca Universitaria*, págs. 282-288.
<https://doi.org/http://dx.doi.org/10.6036/7781>
- Cuerno, C., García, L., Sanchez, A., Carrio, A., & Sanchez, J. L. (2016). Evolución histórica de los vehículos aéreos no tripulados hasta la actualidad. *DYNA, XCI(3)*, 282-288.
<https://doi.org/https://doi.org/10.6036/7781>
- Delgado , A. (2019). *Desarrollo de un sistema de detección y predicción de la pose 3D de objetos en la escena mediante técnicas de deep learning*. Universidad de Alicante.

- Douhet, G. (1921). Empleo de UAV en la Armada. En G. Douhet, *EMPLEO DE UAV EN LA ARMADA ESPAÑOLA. ¿CONCEPTO O CAPACIDAD MILITAR?* (págs. 65-75). GRUESO GARCIA.
- Falconí, L. G., Pérez, M., & Aguilar, W. (2019). Transfer learning in breast mammogram abnormalities classification with mobilenet and nasnet. *International Conference on Systems, Signals and Image Processing (IWSSIP)*, (págs. 109-114).
- Falconi, L. G., Perez, M., Aguilar, W., & Conci, A. (2020). Transfer learning and fine tuning in mammogram bi-rads classification. *IEEE 33rd International Symposium on Computer-Based Medical Systems (CBMS)*, (págs. 475-480).
- Fang, W., Wang, L., & Peiming, R. (2020). Tinier-YOLO: A Real-Time Object Detection Method for Constrained Environments. *IEEE Access*, 8, 1935-1944.
<https://doi.org/10.1109/ACCESS.2019.2961959>
- Galarza, J., Pérez, E., Serrano, E., Tapia, A., & Aguilar, W. (2018). Pose estimation based on monocular visual odometry and lane detection for intelligent vehicles. En *Lecture Notes in Computer Science* (págs. 562-566).
- Galindo, R., Aguilar, W., & Reyes Ch, R. P. (2019). Landmark based eye ratio estimation for driver fatigue detection. En *Lecture Notes in Computer Science* (págs. 565-576). Germany: Springer.
- García, L. (2016). *Interacción hombre-máquina en vehículos aéreos no tripulados: estudio de mejora de comunicación en Aerostack*. Universidad Politécnica de Madrid.
- García, R. (2015). *HERRAMIENTA PARA LA SIMULACIÓN INTEGRADA DE SUBSISTEMAS EN UN EQUIPO QUADROTOR*. Universidad de Sevilla.

- Gary, B., & Kaehler, A. (2008). *Learning OpenCV: Computer Vision with the OpenCV Library*. Oreilly.
- Giernacki, W., Koziarski, P., Michalski, J., Retinger, M., Madonski, R., & Campoy, P. (2020). Bebop 2 Quadrotor as a Platform for Research and Education in Robotics and Control Engineering. *International Conference on Unmanned Aircraft Systems (ICUAS)*. Athens: IEEE. <https://doi.org/10.1109/ICUAS48674.2020.9213872>
- Grijalva, S., & Aguilar, W. (2019). Landmark-Based Virtual Path Estimation for Assisted UAV FPV Tele-Operation with Augmented Reality. En *Lecture Notes in Computer Science* (págs. 688-700).
- Grijalva, S., Cueva, G., Ramírez, D., & Aguilar, W. (2019). HSVM-Based Human Activity Recognition Using Smartphones. En *Lecture Notes in Computer Science* (págs. 217-228).
- Guamba, O. (2010). *Academia*.
https://www.academia.edu/14861114/introduccion_sistema_WGS84
- Guarnizo, M. (2019). *DISEÑO E IMPLEMENTACIÓN DE UN CONTROL SERVOVISUAL PARA UN ROBOT MANIPULADOR TIPO SCARA*. ESCUELA POLITÉCNICA NACIONAL.
- Hart, P., Nilson, N., & Raphael, B. (1986). A formal basis for the heuristic determination of minimum cost paths. *Systems Science and Cybernetics. IEEE Transactions*, 100-107.
- Hartley, R., & Zisserman, A. (2011). *Multiple View Geometry in Computer Vision*. Cambridge University Press. <https://doi.org/10.1017/CBO9780511811685>
- Hassnain, S., Hamood, N., Khan, M., Amjad, H., & Zywolek, J. (2022). A Comprehensive Review of Micro UAV Charging Techniques. *Micromachines*, XIII(977), 1-30.
<https://doi.org/10.3390/mi13060977>

- Hecate, H. (2019). *Synced*. Synced: <https://syncedreview.com/2019/12/16/pytorch-deep-learning-framework-speed-usability/>
- Hoefgeest Toribio, S. R., Fernández García, Á., Álvarez García, I., & González, R. (2019). LOCALIZACIÓN DE ROBOTS MÓVILES EN ENTORNOS INDUSTRIALES USANDO UN ANILLO DE CÁMARAS. *Visión por Computador*. Oviedo: Universidad de Oviedo. <https://doi.org/https://doi.org/10.17979/spudc.9788497497169.849>
- Images, G. (2022). *Unsplash*. <https://unsplash.com/s/photos/people>
- INTEL. (2021). *Intel*. WiFi 6 AX201: <https://www.intel.com/content/www/us/en/products/sku/130293/intel-wifi-6-ax201-gig/specifications.html>
- Jaramillo, A. (1998). *Dinámica de manipuladores robots*. Universidad Pontificia Javeriana.
- Jara-Olmedo, A., Medina-Pazmiño, W., Mesías, R., Araujo-Villaroel, B., Aguilar, W., & Pardo, J. A. (2018). Interface of Optimal Electro-Optical/Infrared for Unmanned Aerial Vehicles. En *Smart Innovation, Systems and Technologies* (págs. 372-380).
- Jara-Olmedo, A., Medina-Pazmiño, W., Tozer, T., Aguilar, W., & Pardo, J. A. (2018). E-services from Emergency Communication Network: Aerial Platform Evaluation. *International Conference on eDemocracy & eGovernment (ICEDEG)* (págs. 251-256). IEEE.
- Jhon, B. (2021). LAS OPERACIONES CON UAV EN LA ARMADA. En J. M. MARTÍN, *People, ideas, hardware*. (págs. 743-751).
- Jung, Y., Bang, H., & Lee, D. (2015). Robust marker tracking algorithm for precise UAV vision-based autonomous landing. *15th International Conference on Control, Automation and Systems (ICCAS)*. Busan: IEEE. <https://doi.org/10.1109/ICCAS.2015.7364957>

- Karami, E., Prasad, S., & Shehata, M. (2017). *Image Matching Using SIFT, SURF, BRIEF and ORB: Performance Comparison for Distorted Images*. Faculty of Engineering and Applied Sciences, Memorial University, Canada.
- Kote, P. (2016). *Xataka*. <https://www.xataka.com/drones/parrot-bebop-2-analisis-autonomia-y-facilidad-de-uso-para-la-renovacion-de-este-divertido-companero-de-vuelo>
- Kumar, S., Brunet, P., & Elias, S. (2023). *AI driven shadow model detection in agropv farms*. Cornell University.
- Kümmerle, R., Grisetti, G., Strasdat, H., Konolige, K., & Burgard, W. (2011). G2o: A general framework for graph optimization. *IEEE International Conference on Robotics and Automation*. Shanghai: IEEE. <https://doi.org/10.1109/ICRA.2011.5979949>
- Leiva, C. (2008). *PARÁMETROS DE TRANSFORMACIÓN ENTRE LOS SISTEMAS GEODÉSICOS DE REFERENCIA PSAD 56 Y SIRGAS95 (WGS 84) PARA EL ECUADOR*. Departamento de Geodesia.
- Limaico, A., Rodríguez, G., Quisaguano, F., Aguilar, W., Sandoval, D., Caballeros, J., & Alvarez, L. (2018). Graph Based RRT Optimization for Autonomous Mobile Robots. *Intelligence Science and Big Data Engineering: 8th International Conference, IScIDE* (págs. 12-21). Springer International Publishing.
- Lopez-Franco, C., Gomez-Avila, J., & Alanis, A. (2017). Visual Servoing for an Autonomous Hexarotor Using a Neural Network Based PID Controller. *MDPI Sensors*, *XVII*(8), 1865. <https://doi.org/10.3390/s17081865>
- Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, *LX*, 91-110.

- Luna, M. A., Moya, J. F., Abad, V., Ruiz, H., Parra, H., Angulo, C., & Aguilar, W. G. (2017). Pedestrian Detection for UAVs Using Cascade Classifiers and Saliency Maps. En *Lecture Notes in Computer Science* (págs. 563-574).
- Luna, M. A., Moya, J. F., Luna, M. P., Abad, V., Ruiz, H., Parra, H., & Aguilar, W. (2017). Real-time detection and simulation of abnormal crowd behavior. En *Lecture Notes in Computer Science* (págs. 420-428).
- Luo, C., Yang, W., & Huang, P. (2019). Overview of Image Matching Based on ORB Algorithm. *Journal of Physics*(1237). <https://doi.org/doi:10.1088/1742-6596/1237/3/032020>
- Manosalvas, J. F., Guillén, J. A., Collaguazo, B., & Aguilar, W. (2018). Robust Motion Estimation Based on Multiple Monocular Camera for Indoor Autonomous Navigation of Micro Aerial Vehicle. *International Conference on Augmented Reality, Virtual Reality and Computer Graphics* (págs. 547-561). Springer.
- Martínez Luna, C. (2013). *Visual Tracking, Pose Estimation, and Control for Aerial Vehicles*. Universidad Politécnica de Madrid.
- Martinez, J. (2020). *ESTUDIO Y CARACTERIZACIÓN DE MATERIALES ESTRUCTURALES PARA DRONES*. UNIVERSIDAD POLITÉCNICA DE CARTAGENA.
- Mella, C. (2 de 10 de 2022). Ecuador alcanza la tasa más alta de muertes violentas de la última década. *PRIMICIAS*. <https://www.primicias.ec/noticias/en-exclusiva/ecuador-tasa-muertes-violentas-ultima-decada/>
- Merizalde, D., Aguilar, W., & Calderón, M. (2020). Autonomous Navigation Based on Proportional Controller with GPS Setpoint for UAV in External Environments. En *Smart Innovation, Systems and Technologies* (págs. 89-99).

- Mikolajczyk, K., & Schmid, C. (2005). A performance evaluation of local descriptors. *IEEE transactions on pattern analysis and machine intelligence*, XXVII(10), 1615-1630.
<https://doi.org/10.1109/TPAMI.2005.188>
- Miranda, R. (2016). *Cinemática y Dinámica de Robots manipuladores*. Alfaomega.
- Montoya, P. B., & Briones, R. V. (2019). Empleo de los UAV, en operaciones de seguridad y vigilancia en las áreas estratégicas en el ecuador. *Revista de Ciencias de Seguridad y Defensa*, 4(4), 15.
- Montoya, P., & Vásquez, R. (2019). EMPLEO DE LOS UAV, EN OPERACIONES DE SEGURIDAD Y VIGILANCIA EN LAS ÁREAS ESTRATÉGICAS EN EL ECUADOR. *Revista de Ciencias de Seguridad y Defensa*, IV(4), 86-100.
- Moposita, J., Orbea, D., Aguilar, W., Paredes, M., Reyes, R. P., & Montoya, L. (2017). Vertical take off and landing with fixed rotor. *Chilean Conference on Electrical, Electronics Engineering, Information and Communication Technologies (CHILECON)*. Pucón, Chile.
- Moya, J. F., Abad, V., Parra, H., Ruiz, H., Aguilar, W., & Luna, M. A. (2017). Pedestrian detection for UAVs using cascade classifiers with meanshift. *IEEE 11th international conference on semantic computing (ICSC)*.
- MSI. (2020). *Katana GF76*. Katana GF76: <https://es.msi.com/Laptop/Katana-GF76-11UX>
- Muhammad , Y. A., Muhammad, M. A., & Sangman, M. (2023). Vision-Based Navigation Techniques for Unmanned Aerial Vehicles: Review and Challenges. *MDPI*, 7(89).
<https://doi.org/10.3390/drones7020089>
- Munin, A., Folarin, A., Munin-Doce, A., Alonso-Garcia, L., Diaz-Casas, V., Ferreno-Gonzalez, S., & Ciriano-Palacios, J. M. (2023). Real Time Vessel Detection Model Using Deep Learning Algorithms for Controlling a Barrier System. *SSRN*.

- Mur-Artal, R., Montiel, J., & Tardos, J. (2015). ORB-SLAM: a versatile and accurate monocular SLAM system. *IEEE transactions on robotics*, XXXI(5), 1147-1163.
<https://doi.org/10.1109/TRO.2015.2463671>
- Murillo Álvarez, J. (2021). *Control servo-visual de una plataforma giroestabilizada*. Universidad de Sevilla. Departamento de Ingeniería de Sistemas y Automática.
- Orbea, D., Moposita, J., Aguilar, W., Paredes, M., León, G., & Jara-Olmedo, A. (2017). Math Model of UAV Multi Rotor Prototype with Fixed Wing Aerodynamic Structure for a Flight Simulator. En *Lecture Notes in Computer Science* (págs. 199-211).
- Oropeza, V. (23 de 9 de 2022). Crisis de seguridad en Ecuador. *BBC News Mundo*.
<https://www.bbc.com/mundo/noticias-america-latina-62976228>
- Oughton , E., Lehr, W., Katsaros, K., Selinis, I., Bublely, D., & Kusuma, J. (2021). Revisiting Wireless Internet Connectivity: 5G vs Wi-Fi 6. *ELSEVIER*, XLV(5).
<https://doi.org/10.1016/j.telpol.2021.102127>
- Palomino, W. (2017). *TRANSFORMACIONES DEL PLANO R2*. UNIVERSIDAD NACIONAL DE EDUCACIÓN ENRIQUE GUZMÁN Y VALLE.
- Pardo, J. A., Aguilar, W., & Toulkeridis, T. (2017). Wireless communication system for the transmission of thermal images from a UAV. *Chilean Conference on Electrical, Electronics Engineering, Information and Communication Technologies (CHILECON)*. Pucón, Chile.
- Parrot SA. (2018). *PARROT*. Dron Parrot Bebop 2: <https://www.parrot.com/en#parrot-bebop-2-fpv-details>

- Qin, Y., Xu, H., & Chen, H. (2014). Image feature points matching via improved ORB. *IEEE International Conference on Progress in Informatics and Computing*. Shanghai: IEEE.
<https://doi.org/10.1109/PIC.2014.6972325>
- Quigley, M., Gerkey, B., & Smart, W. (2007). *Programming Robots with ROS*. OREILLY.
- Quigley, M., Gerkey, B., Conley, K., Faust, J., & Foote, T. (2010). *ROS: an open-source Robot Operating System*. Computer Science Department, Stanford University, Stanford, CA.
- Quisaguano, F. J., Alvarez, L. G., Pardo, J. A., Zahira, P., & Aguilar, W. (2018). Monocular depth perception on a micro-UAV using convolutional neuronal networks. En *Lecture Notes in Computer Science* (págs. 392-397).
- Quisaguano, F. J., Rodríguez, G. A., Alvarez, L. G., Limaico, A., S, S. D., & Aguilar, W. (2018). Convolutional neuronal networks based monocular object detection and depth perception for micro UAVs. En *Lecture Notes in Computer Science* (págs. 401-410).
- Raheem, H., & Al-Assadi, T. (2022). Video Important Shot Detection Based on ORB Algorithm and FLANN Technique. *8th International Engineering Conference on Sustainable Technology and Development (IEC)*. Erbil: IEEE.
<https://doi.org/10.1109/IEC54822.2022.9807488>
- Ramírez, Ó., & Aguilar, W. (2019). Conciencia situacional en la frontera, proporcionada por sensores de alta tecnología. *Revista de Ciencias de Seguridad y Defensa*, 169-178.
- Redacción, E. U. (5 de 8 de 2022). Así están las cifras de la inseguridad en el primer semestre del 2022 comparadas con los tres años previos. *El Universo*.
<https://www.eluniverso.com/noticias/seguridad/asi-estan-las-cifras-de-la-inseguridad-en-el-primer-semestre-del-2022-comparadas-con-los-tres-anos-previos-nota/>
- Redmon, J., & Farhadi, A. (2018). YOLOv3: An incremental improvement. *IEEE* .

- Restrepo, S. (2022). *SISTEMA PARA LA GENERACIÓN AUTOMÁTICA DE TRAYECTORIAS PARA ENJAMBRES DE DRONES CRAZYSWARM EN LA UNIVERSIDAD AUTÓNOMA DE OCCIDENTE (UAO)*. UNIVERSIDAD AUTÓNOMA DE OCCIDENTE.
- Ridlwan, H., Nugraha, Y., Riansyah, M., & Trilaksono, B. (2017). Simulation of Vision-Based For Automatic Takeoff and Landing Hexacopter on a Moving Ground Vehicle. *ACM Journal on Computing and Sustainable Societies*, 502-506.
<https://doi.org/10.1145/3055635.3056658>
- Rodríguez, G. A., Álvarez, L., Sandoval, S., Quisaguano, F., Limaico, A., & Aguilar, W. G. (2017). Visual SLAM with a RGB-D Camera on a Quadrotor UAV Using on-Board Processing. En *Lecture Notes in Computer Science* (págs. 596-606).
- Rodriguez, J. (2016). *Desarrollo del control visual de un UAV para el seguimiento de líneas de cultivo*. Universidad Politecnica de Madrid.
- Rojas, I., Aguilar, W., Álvarez, L., & Grijalva, S. (2019). Monocular Vision-Based Dynamic Moving Obstacles Detection and Avoidance. En *Lecture Notes in Computer Science* (págs. 386-398).
- ROS, W. (2014). *ROS Wiki*. <http://wiki.ros.org/ROS/Concepts>
- Rosten, E., & Drummond, T. (2006). *Machine Learning for High-Speed Corner Detection*. Universidad of Cambridge.
- Rublee, E., Rabaud, V., Konolige, K., & Bradski, G. (2011). ORB: An efficient alternative to SIFT or SURF. *IEEE*, 2564-2571.
- Salcedo, V. S., Aguilar, W., Cobeña, B., Pardo, J. A., & Zahira, P. (2018). On-board target virtualization using image features for UAV autonomous tracking. En *Lecture Notes in Computer Science* (págs. 384-391).

Salgado, M. F., Tierra, A., Sandoval, D., & Aguilar, W. (2017). Travel Time Estimation of Public Transport Networks Based on Commercial Incidence Areas in Quito Historic Center. *Analysis*.

Sánchez, C. (2017). VEHÍCULOS AÉREOS NO TRIPULADOS: DESCRIPCIONES GENERALES Y APLICACIONES. *DESARROLLO TECNOLÓGICO E INNOVACIÓN EMPRESARIAL*, 1(6).

Sanchez, E., & Aguilar, W. (2019). Apoyo logístico de FF.AA ante un desastre natural. *Revista de Ciencias de Seguridad y Defensa*, 23-33.

Sandoval, S., Limaico, A., Villegas-Pico, M., Asimbaya, I., & Aguilar, W. (2019). Path planning based navigation using LIDAR for an Ackerman unmanned ground vehicle. En *International Conference on Intelligent Robotics and Applications* (págs. 399-410).

Segarra, D., Caballeros, J., Aguilar, W., Samà, A., & Rodríguez-Martín, D. (2019). Orientation Estimation Using Filter-Based Inertial Data Fusion for Posture Recognition. En *Lecture Notes in Computer Science* (págs. 220-233).

Shu, C., & Xiao, X. (2018). ORB-oriented mismatching feature points elimination. *IEEE International Conference on Progress in Informatics and Computing (PIC)* (págs. 246-249). Suzhou: IEEE. <https://doi.org/10.1109/PIC.2018.8706272>

Soria, C., Roberti, F., Carelli, R., & Sebastian, J. (2008). Control Servo-Visual de un Robot Manipulador Planar Basado en Pasividad. *ELSEVIER*, V(4), 54-61. [https://doi.org/10.1016/S1697-7912\(08\)70177-8](https://doi.org/10.1016/S1697-7912(08)70177-8)

Szeliski, R. (2010). *Computer Vision: Algorithms and Applications*. Springer.

Tenezaca, D., Canchignia, C., Aguilar, W., & Mendoza, D. (2019). Implementation of Dubin Curves-Based RRT* Using an Aerial Image for the Determination of Obstacles and Path

- Planning to Avoid Them During Displacement of the Mobile Robot. En *Smart Innovation, Systems and Technologies* (págs. 205-215).
- Tierra, A., F, S. M., & Aguilar, W. (2017). Travel Planning in Public Transport Networks Applying the Algorithm A* for Metropolitan District of Quito. *Analysis*.
- Tinizaray, P., Aguilar, W., & Lucio, J. (2022). Fast segmentation of point clouds using a convolutional neural network for helping visually impaired people find the closest traversable region. *Inteligencia Artificial*, 50-63.
- Tituaña, J., Amaguaña, F., Calderón, M., Merizalde, D., & Aguilar, W. (2020). Visual and Inertial Data-Based Virtual Localization for Urban Combat. En *International Conference of Research Applied to Defense and Security* (págs. 65-74).
- Valero, C. (2017). *Desarrollo de aplicaciones basadas en visión con entornos ROS para el DRone*. Universidad de Alcalá.
- Veness, C. (2020). *Movable Type Scripts*. Calculate distance, bearing and more between Latitude/Longitude points: <https://www.movable-type.co.uk/scripts/latlong.html?from=48.955550,-122.05169&to=48.965496,-122.072989>
- Villamarin, D. F. (2015). Estado del Arte, Herramientas y Aplicaciones para Transformaciones geométricas 3D. *Congreso de Ciencia y Tecnología ESPE*, 10, 226-231.
- Villanueva, M. (2018). *Modelado y simulación dinámica de un robot*. Universidad Politécnica de Valencia.
- Wang, G., Liu, Z., & Wang, X. (2019). UAV Autonomous Landing using Visual Servo Control based on Aerostack. *ACM Journals*, 1-6. <https://doi.org/10.1145/3331453.3361667>

- Wang, G., Zhai, Z., Xu, B., & Cheng, Y. (2017). A parallel method for aerial image stitching using ORB feature points. *International Conference on Computer and Information Science (ICIS)*. Wuhan: IEEE. <https://doi.org/10.1109/ICIS.2017.7960096>
- Xu, Z.-C., Hu, B.-B., Liu, B., Wang, X., & Zhang, H. (2020). Vision-based Autonomous Landing of Unmanned Aerial Vehicle on a Motional Unmanned Surface Vessel. *39th Chinese Control Conference (CCC)*. Shenyang: IEEE. <https://doi.org/10.23919/CCC50068.2020.9188979>
- Yang, J., Xiao, B., Fu, Z., & Chaofan, W. (2017). Visual servo control of unmanned aerial vehicles: An object tracking-based approach. *IEEE Xplore*. <https://doi.org/10.1109/CCDC.2017.7979116>
- Zhang, H., Zheng, G., & Fu, H. (2020). Research on Image Feature Point Matching Based on ORB and RANSAC Algorithm. *Journal of Physics: Conference Series*. <https://doi.org/10.1088/1742-6596/1651/1/012187>
- Zhang, Y., Yu, Y., Jia, S., & Wang, X. (2017). Autonomous landing on ground target of UAV by using image-based visual servo control. *36th Chinese Control Conference (CCC)*. Dalian: IEEE. <https://doi.org/10.23919/ChiCC.2017.8029145>
- Zurita, M., Aguilar, W., & Enríquez, V. (2019). Toward the Development of Surveillance and Reconnaissance Capacity in Ecuador: Geolocation System for Ground Targets Based on an Electro-Optical Sensor. En *Smart Innovation, Systems and Technologies* (págs. 123-133).