



## **Sistema de patrullaje de drones para detección de personas en espacios restringidos**

Mancero Viscaino, Luis Antonio y Viteri Basantes, Javier Alejandro

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

Ph.D. Aguilar Castillo, Wilbert Geovanny

6 de julio del 2021

## Urkund Analysis Result

**Analysed Document:** Tesis\_ManceroViteri.pdf (D110159789)  
**Submitted:** 7/5/2021 2:27:00 PM  
**Submitted By:** wgaguilar@espe.edu.ec  
**Significance:** 3 %

### Sources included in the report:

tesis\_calderon\_merizalde.pdf (D75308789)  
Tesis\_Yamberla\_Jonathan.pdf (D98724338)  
Tesis\_Yamberla\_Jonathan.pdf (D98672911)  
TrabajoTitulaciónVinicioSalcedo.pdf (D35374643)  
tesis\_grijalva\_santiago.pdf (D57099779)  
TESIS\_CHAUCA\_BRYAN\_urkund.pdf (D78121896)  
<https://docplayer.es/201888935-Seguimiento-y-busqueda-de-objetivos-en-entornos-complejos-usando-micro-vehiculos-aereos-con-camaras-monoculares-para-aplicaciones-militares.html>  
<https://core.ac.uk/download/pdf/288501214.pdf>  
[https://e-archivo.uc3m.es/bitstream/handle/10016/29343/TFG\\_Rocio\\_Sieiro\\_Alfonsin.pdf?sequence=1](https://e-archivo.uc3m.es/bitstream/handle/10016/29343/TFG_Rocio_Sieiro_Alfonsin.pdf?sequence=1)

### Instances where selected sources appear:

24



Firmado digitalmente por:  
WILBERT GEOVANNY  
AGUILAR CASTILLO

**Dr. Wilbert G. Aguilar Castillo**  
Director de Tesis



**DEPARTAMENTO DE ELÉCTRICA, ELECTRÓNICA Y  
TELECOMUNICACIONES**

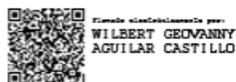
**CARRERA DE INGENIERÍA EN ELECTRÓNICA, AUTOMATIZACIÓN Y  
CONTROL**

**CERTIFICACIÓN**

Certifico que el trabajo de titulación, “**Sistemas de patrullaje de drones para detección de personas en espacios restringidos**” fue realizado por los señores **Mancero Viscaino, Luis Antonio** y **Viteri Basantes, Javier Alejandro** el cual ha sido revisado y analizado en su totalidad por la herramienta de verificación de similitud de contenido; por lo tanto cumple con los requisitos legales, teóricos, científicos, técnicos y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, razón por la cual me permito acreditar y autorizar para que lo sustente públicamente.

Sangolquí, 21 de junio de 2021

Firma:



**Dr. Wilbert Geovanny Aguilar Castillo**

C. C. 070384469-6



DEPARTAMENTO DE ELÉCTRICA, ELECTRÓNICA Y  
TELECOMUNICACIONES

CARRERA DE INGENIERÍA EN ELECTRÓNICA, AUTOMATIZACIÓN Y  
CONTROL

RESPONSABILIDAD DE AUTORÍA

Nosotros, **Mancero Viscaino, Luis Antonio** y **Viteri Basantes, Javier Alejandro**, con cédulas de ciudadanía n° 1751637248 y 0802375014, declaramos que el contenido, ideas y criterios del trabajo de titulación: **Sistema de patrullaje de drones para detección de personas en espacios restringidos** 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, 21 de junio de 2021

Firma

Mancero Viscaino, Luis Antonio

C.C.: 1751637248

Viteri Basantes, Javier Alejandro

C.C.: 0802375014



DEPARTAMENTO DE ELÉCTRICA, ELECTRÓNICA Y  
TELECOMUNICACIONES  
CARRERA DE INGENIERÍA EN ELECTRÓNICA, AUTOMATIZACIÓN Y  
CONTROL

AUTORIZACIÓN DE PUBLICACIÓN

Nosotros **Mancero Viscaino, Luis Antonio** y **Viteri Basantes, Javier Alejandro**, con cédulas de ciudadanía n°1751637248 y 0802375014, autorizamos a la Universidad de las Fuerzas Armadas ESPE publicar el trabajo de titulación: **Sistema de patrullaje de drones para detección de personas en espacios restringidos** en el Repositorio Institucional, cuyo contenido, ideas y criterios son de mi/nuestra responsabilidad.

Sangolquí, 21 de junio de 2021

Firma

**Mancero Viscaino, Luis Antonio**

C.C.: 1751637248

**Viteri Basantes, Javier Alejandro**

C.C.: 0802375014

### **Dedicatoria**

Este proyecto de titulación lo dedico a mi padre y madre, por siempre impulsarme a alcanzar todas las metas que me he propuesto en la vida con su infinito amor. A mi hermana Marjorye, por estar conmigo en todos los momentos difíciles a lo largo de mi carrera y siempre ser mi ejemplo a seguir para alcanzar mis sueños.

De la misma manera dedico este trabajo a todos mis compañeros y profesores de las diferentes carreras de la ESPE que tuve el gusto de conocer a través de mi formación educativa. Y sobre todo a mis compañeros y hermanos en la disciplina de Crossfit por estar ahí para mí siempre que los necesité

**Javier Alejandro Viteri Basantes**

A mis padres Antonio y Susy que con su cariño incondicional y consejos siempre me han guiado y motivado a seguir adelante a lo largo de los años, gracias a ustedes por enseñarme que con perseverancia se puede llegar a lograr alcanzar mis objetivos. A mis hermanas que me han enseñado a través de sus experiencias a sobrellevar las dificultades que la vida nos presenta a diario, por todas esas anécdotas y vivencias juntos. A mis compañeros por su apoyo, ánimos en cada escalón personal y académico.

**Luis Antonio Mancero Viscaino**

## **Agradecimiento**

Agradezco especialmente a toda mi familia que me acompañaron hasta este punto de mi vida. A mi padre por enseñarme a enfocarme en los detalles en la vida para tomar las mejores decisiones. A mi madre por guiarme con su inmenso amor. A mi hermana por seguir a mi lado en esta apasionante aventura que es la ingeniería.

Quiero agradecer a mi amigo Paul Torres por apoyarme al transmitir su conocimiento en ingeniería geográfica que fue de gran importancia en el desarrollo de este trabajo de titulación.

Agradezco a todos los ingenieros que ayudaron a mi formación académica. Y un especial agradecimiento al ingeniero Richard Bernis por apoyarme en mi crecimiento tanto académico como personal desde mi primer día en la universidad.

**Javier Alejandro Viteri Basantes**

Agradezco a toda mi familia por siempre apoyarme y guiarme a través de las adversidades, este logro se lo debo a ustedes. A mi compañero Javier por tenerme paciencia con las dificultades de tiempo, pero sin su apoyo no se habría logrado el objetivo.

Y finalmente a Francini que fue un apoyo incondicional a lo largo de los años de estudio, por todas tus enseñanzas y reprimendas para que no me conforme con pequeñeces si no que cada vez busque realizar un trabajo excelente para superarme cada día más, te lo agradezco de todo corazón.

**Luis Antonio Mancero Viscaino**

## Índice de contenidos

|  |    |
|--|----|
| Resultados de Urkund.....                | 2  |
| Certificación .....                      | 3  |
| Responsabilidad de autoría.....          | 4  |
| Autorización de publicación.....         | 5  |
| Dedicatoria.....                         | 6  |
| Agradecimiento.....                      | 7  |
| Indice de contenidos .....               | 8  |
| Indice de tablas.....                    | 11 |
| Indice de figuras.....                   | 13 |
| Resumen .....                            | 15 |
| Abstract.....                            | 16 |
| Glosario .....                           | 17 |
| Capítulo I .....                         | 18 |
| Introducción .....                       | 18 |
| Antecedentes .....                       | 18 |
| Justificación e importancia.....         | 22 |
| Alcance del proyecto .....               | 25 |
| Objetivos .....                          | 27 |
| <i>Objetivo General</i> .....            | 27 |
| <i>Objetivos Específicos</i> .....       | 28 |
| Capítulo II .....                        | 29 |
| Marco Teórico.....                       | 29 |
| Métodos de navegación de un UAV .....    | 29 |
| Métodos de aterrizaje de un UAV.....     | 30 |
| Algoritmos de detección de personas..... | 30 |

|  |    |
|--|----|
| <i>Haar Cascade</i> .....  | 31 |
| <i>HOG + SVM</i> .....   | 32 |
| <i>Yolo</i> .....  | 33 |
| Software de análisis y manipulación de datos geográficos ArcGis.....     | 34 |
| Capítulo III .....   | 36 |
| Descripción del Sistema.....   | 36 |
| Descripción de Hardware .....  | 36 |
| <i>Estación terrestre</i> .....  | 36 |
| <i>Estación aérea (UAV)</i> .....  | 37 |
| Descripción de Software.....   | 40 |
| <i>Robot Operating System (ROS)</i> .....                                | 41 |
| <i>Comunicación del Sistema</i> .....                                    | 41 |
| Capítulo IV .....  | 43 |
| Sistema de Navegación .....  | 43 |
| Navegación Gruesa.....   | 43 |
| <i>Navegación basada en GPS</i> .....                                    | 43 |
| <i>Diseño del controlador de navegación autónoma</i> .....               | 46 |
| Navegación Fina .....  | 48 |
| <i>Diseño del controlador a partir de estimación de movimiento</i> ..... | 51 |
| Capítulo V .....   | 56 |
| Detección de Personas .....  | 56 |
| Introducción.....  | 56 |
| Implementación del Algoritmo de Detección de Personas .....              | 57 |
| Capítulo VI .....  | 60 |
| Pruebas de patrullaje y Resultados.....                                  | 60 |
| Circuito cuadrado .....  | 61 |

|   |     |
|---|-----|
| Circuito en forma de rombo .....  | 68  |
| Circuito en forma de trapecio.....  | 75  |
| Pruebas de reconocimiento de personas.....                                      | 82  |
| <i>Pruebas de desempeño Haar Cascade</i> .....                                  | 83  |
| <i>Pruebas de desempeño Hog + SVM</i> .....                                     | 85  |
| <i>Pruebas de desempeño Yolo</i> .....  | 87  |
| Pruebas de desempeño de HOG + SVM en el sistema de patrullaje .....             | 89  |
| <i>Pruebas de desempeño de HOG + SVM en circuito cuadrado</i> .....             | 89  |
| <i>Pruebas de desempeño de HOG + SVM en circuito en forma de rombo</i> .....    | 91  |
| <i>Pruebas de desempeño de HOG + SVM en circuito en forma de trapecio</i> ..... | 93  |
| Pruebas de controlador de aterrizaje .....                                      | 96  |
| Capítulo VII .....  | 100 |
| Conclusiones y Recomendaciones .....  | 100 |
| Conclusiones.....   | 100 |
| Recomendaciones.....  | 102 |
| Referencias.....  | 104 |
| Anexos.....   | 120 |

## Índice de tablas

|   |    |
|---|----|
| <b>Tabla 1.</b> Características de cuadricóptero Bebop Parrot 2 .....                                 | 38 |
| <b>Tabla 2.</b> Movimientos básicos de Bebop Parrot 2 .....   | 39 |
| <b>Tabla 3.</b> Características de cámara en Bebop Parrot 2.....                                      | 40 |
| <b>Tabla 4.</b> Condiciones de operación para modelamiento .....                                      | 52 |
| <b>Tabla 5.</b> Datos de longitud tomados por Google Earth y por el cuadricóptero Bebop Parrot 2..    | 62 |
| <b>Tabla 6.</b> Datos de latitud tomados por Google Earth y por el cuadricóptero Bebop Parrot 2 ..... | 63 |
| <b>Tabla 7.</b> Tabla de errores en circuito cuadrado.....  | 66 |
| <b>Tabla 8.</b> Tabla de errores de la misión de vuelo .....  | 67 |
| <b>Tabla 9.</b> Datos de longitud tomados por Google Earth y por el cuadricóptero Bebop Parrot 2..    | 69 |
| <b>Tabla 10.</b> Datos de latitud tomados por Google Earth y por el cuadricóptero Bebop Parrot 2...70 | 70 |
| <b>Tabla 11.</b> Tabla de errores en circuito en forma de rombo.....                                  | 73 |
| <b>Tabla 12.</b> Tabla de errores de la misión de vuelo .....   | 74 |
| <b>Tabla 13.</b> Datos de longitud tomados por Google Earth y por el cuadricóptero Bebop Parrot 2 76  | 76 |
| <b>Tabla 14.</b> Datos de latitud tomados por Google Earth y por el cuadricóptero Bebop Parrot 2...77 | 77 |
| <b>Tabla 15.</b> Tabla de errores en circuito en forma de trapecio .....                              | 80 |
| <b>Tabla 16.</b> Tabla de errores de la misión de vuelo .....   | 81 |
| <b>Tabla 17.</b> Resumen de resultados de la implementación del algoritmo Haar Cascade .....          | 85 |
| <b>Tabla 18.</b> Resumen de resultados de la implementación del algoritmo Hog + SVM.....              | 87 |
| <b>Tabla 19.</b> Resumen de resultados de la implementación del algoritmo Yolo.....                   | 89 |
| <b>Tabla 20.</b> Resumen de resultados de la implementación de detección de personas .....            | 91 |
| <b>Tabla 21.</b> Resumen de resultados de la implementación de detección de personas .....            | 93 |
| <b>Tabla 22.</b> Resumen de resultados de la implementación de detección de personas .....            | 95 |
| <b>Tabla 23.</b> Prueba de aterrizaje del circuito cuadrado .....                                     | 96 |
| <b>Tabla 24.</b> Prueba de aterrizaje del circuito con forma de rombo .....                           | 97 |

|  |    |
|--|----|
| <b>Tabla 25.</b> Prueba de aterrizaje del circuito con forma de trapecio.....                    | 98 |
| <b>Tabla 26.</b> Errores globales del controlador de aterrizaje en circuito los 3 circuitos..... | 99 |

## Índice de figuras

|  |    |
|--|----|
| <b>Figura 1.</b> Esquema general del sistema de patrullaje .....   | 27 |
| <b>Figura 2.</b> Etapas de un clasificador Haar Cascade .....  | 32 |
| <b>Figura 3.</b> Extracción de características HOG.....  | 33 |
| <b>Figura 4.</b> Proceso de detección de objetos mediante Yolo .....   | 34 |
| <b>Figura 5.</b> Interfaz gráfica de ArcGis.....   | 35 |
| <b>Figura 6.</b> Ejemplo de plataforma y dimensiones propuestas para el proyecto .....                                 | 37 |
| <b>Figura 7.</b> Cuadricóptero Bebop Parrot 2.....   | 37 |
| <b>Figura 8.</b> Interfaz gráfica de la aplicación FreeFlight PRO y función de calibración .....                       | 43 |
| <b>Figura 9.</b> Ejemplo de movimiento misión de vuelo del dron e .....  | 44 |
| <b>Figura 10.</b> Traducción de puntos al origen para facilitar el cálculo del ángulo $\alpha$ .....                   | 45 |
| <b>Figura 11.</b> Datos de odometría entregado por Bebop Parrot 2 y equivalencia en grados .....                       | 46 |
| <b>Figura 12.</b> Diagrama de bloques del sistema de control autónomo .....  | 47 |
| <b>Figura 13.</b> Ecuación de movimiento del cuadricóptero .....   | 48 |
| <b>Figura 14.</b> Diagrama de bloques extracción de características y detección del objeto.....                        | 49 |
| <b>Figura 15.</b> Área de aterrizaje del UAV .....   | 52 |
| <b>Figura 16.</b> Gráficas para el modelamiento del controlador .....  | 53 |
| <b>Figura 17.</b> Comparación entre estimación de movimiento y validación de movimiento.....                           | 55 |
| <b>Figura 18.</b> Histograma de gradiente orientado.....   | 58 |
| <b>Figura 19.</b> Detección de personas con HOG + SVM implementado en la Estación en Tierra ...                        | 59 |
| <b>Figura 20.</b> Ubicación de circuitos de prueba en el perímetro de Universidad de las Fuerzas<br>Armadas ESPE ..... | 60 |
| <b>Figura 21.</b> Circuito cuadrado con 42 coordenadas generadas por ArcGis.....                                       | 61 |
| <b>Figura 22.</b> Gráfica de coordenadas obtenidas en el circuito cuadrado .....                                       | 65 |
| <b>Figura 23.</b> Circuito en forma de rombo con 42 coordenadas generadas por ArcGis.....                              | 68 |

|  |    |
|--|----|
| <b>Figura 24.</b> Gráfica de coordenadas obtenidas en circuito en forma de rombo .....                           | 72 |
| <b>Figura 25.</b> Circuito en forma de trapecio con 42 coordenadas generadas por ArcGis .....                    | 75 |
| <b>Figura 26.</b> Gráfica de coordenadas obtenidos en circuito en forma de trapecio.....                         | 79 |
| <b>Figura 27.</b> Resumen de resultados de reconocimiento de Haar Cascade .....                                  | 84 |
| <b>Figura 28.</b> Resumen de resultados de reconocimiento de Hog + SVM.....                                      | 86 |
| <b>Figura 29.</b> Resumen de pruebas de reconocimiento de Yolo .....   | 88 |
| <b>Figura 30.</b> Resultados de pruebas de reconocimiento de personas en circuito cuadrado.....                  | 90 |
| <b>Figura 31.</b> Resultados de pruebas de reconocimiento de personas en circuito en forma de<br>rombo .....     | 91 |
| <b>Figura 32.</b> Resultados de pruebas de reconocimiento de personas en circuitos en forma de<br>trapecio ..... | 94 |

## Resumen

Los vehículos aéreos no tripulados (UAV) han ganado gran popularidad debido a su aplicación en el ámbito militar, sector privado y público. Se puede encontrar una amplia gama de aplicaciones en estos vehículos aéreos tales como: monitoreo en tiempo real de una variable cualquiera, apoyo a entidades encargadas de búsqueda y rescate de civiles, sistemas de seguridad. En este proyecto de titulación se utilizó un cuadricóptero Parrot Bebop 2 para un sistema de patrullaje autónomo con detección de personas. Se implementó 2 metodologías para el sistema de navegación. El primero es la estimación de estado, la cual se refiere al proceso de seguimiento de posición actual del vehículo por medio de GPS. Con este, se obtiene un modelo matemático el cual describirá el movimiento del dron a lo largo de su trayectoria. El segundo método de navegación es el sistema de percepción, se realiza un control con base en las imágenes capturadas desde una cámara monocular y la información inercial de unidad de medición inercial del UAV para su aterrizaje. Finalmente, la detección de personas se la realiza por medio de la extracción de características y clasificación ORB implementado en OpenCV.

-Palabras clave:

- **NAVEGACIÓN AUTONOMA**
- **DRONES**
- **DETECCIÓN PERSONAS**
- **CONTROL UAV**

### **Abstract**

Unmanned aerial vehicles (UAVs) have gained great popularity due to their application in the military, private and public sectors. A wide range of applications can be found in these air vehicles such as: real-time monitoring of any variable, support for entities in charge of search and rescue of civilians, security systems. In this titling project a Parrot Bebop 2 quadcopter was used for an autonomous patrol system with human detection. Two methodologies were implemented for the navigation system. The first is the state estimation, which refers to the process of tracking the current position of the vehicle by means of GPS. With this, a mathematical model is obtained which describes the movement of the drone along its path. The second navigation method is the perception system, a control is carried out based on the images captured from a monocular camera and the inertial information from the UAV's inertial measurement unit for landing. Finally, the detection of people is carried out through the extraction of characteristics and classification ORB implemented in OpenCV.

-Key words:

- **AUTONOMOUS NAVIGATION**
- **DRONE**
- **DETECTION OF PEOPLE**
- **UAV CONTROL**

## Glosario

|              |   |
|--------------|---|
| <b>BRIEF</b> | Binary Robust Independent Elementary Features |
| <b>FAST</b>  | Features from accelerated segment test        |
| <b>GPS</b>   | Global Positioning System                     |
| <b>ORB</b>   | Oriented FAST and Rotated BRIEF               |
| <b>RGB</b>   | Red Green Blue                                |
| <b>ROS</b>   | Robot operating system                        |
| <b>SIFT</b>  | Scale-invariant feature transform             |
| <b>SURF</b>  | Speeded up robust features                    |
| <b>SVM</b>   | Support Vector Machine                        |
| <b>SLAM</b>  | Simultaneous Localization and Mapping         |
| <b>UAV</b>   | Unmanned aerial vehicle                       |

## Capítulo I

### Introducción

#### ***Antecedentes***

Los vehículos aéreos no tripulados (UAV) (Orbea, y otros, Vertical take off and landing with fixed rotor, 2017) están ganando popularidad debido a su aplicación en el ámbito militar, sector privado y público, especialmente en campos en los cuales no es requerido un operador humano (Goh, et al., 2017) (Sánchez Martínez, 2012). Los UAVs pueden ser utilizados en muchas aplicaciones civiles debido a su fácil implementación, bajo coste de mantenimiento y alta movilidad (Hayat, Yanmaz, & Muzaffar, 2016). Algunas de las aplicaciones de los UAVs son monitoreo en tiempo real del tráfico (Barrientos, et al., 2007), operaciones de búsqueda y rescate, entrega de bienes, seguridad y vigilancia (Gallardo Valdivia, 2007), inspección de agricultura e infraestructura civil (Shakhatreh, et al., 2019). El número de UAVs ha incrementado drásticamente a lo largo de los años, así como las empresas dedicadas a la fabricación y uso de estos vehículos. Así mismo, investigadores de múltiples disciplinas encuentran nuevos usos a los UAVs (Birnbaum, et al., 2016).

El control manual del UAV brinda rápida adaptabilidad y alta precisión requerida para navegación y aterrizaje del vehículo (Aguilar, Cobeña, Rodriguez, Salcedo, & Collaguazo, 2018). El control y manejo de un UAV no es tarea sencilla y dependerá de la destreza del piloto (Gandhi, Pinto, & Gupta, 2017). El crecimiento tecnológico de los UAV ha generado nuevos campos de investigación e innovadores algoritmos con el objetivo de lograr navegación autónoma que minimice el número de accidentes (Chmaj & Selvaraj, 2015). Para la navegación autónoma de UAVs existen dos metodologías ampliamente utilizadas:

GPS: El Sistema de posicionamiento global (GPS) es la tecnología central para la localización de UAVs, los cuales pueden volar de forma autónoma con una cámara basada en GPS (Chmaj & Selvaraj, 2015). Los múltiples sensores (acelerómetros y giroscopios) proporcionan la información sobre el desplazamiento y rotación del UAV (Kwak & Sung, 2018). Las rutas de vuelo se las realiza ingresando secuencialmente los puntos de dirección (waypoints) que contienen las ubicaciones y alturas (Kissai & Smith, 2019).

Control servo visual: La navegación basada en visión depende de la correcta captura de imágenes y selección (Aguilar, Quisaguano, Alvarez, Pardo, & Zahira, 2018) (Shirai, Yamaguchi, & Takaba, 2017). Al tomar imágenes de un vehículo en movimiento (Aguilar W. G., y otros, Pedestrian detection for UAVs using cascade classifiers with meanshift, 2017) puede generar efectos no deseados en las imágenes, los efectos de traslación y rotación que se generan en el UAV son muy importantes debido a la inestabilidad que se puede generar en el vuelo, por lo cual los comercializadores de UAVs han optado por implementar métodos de estabilización de video como un control servo visual para la corrección de imágenes (Aguilar, Angulo, & Pardo, Motion intention optimization for multicopter robust video stabilization, 2017) (Yang & Quan, 2020).

En el control servo visual se pueden identificar diferentes métodos, como pueden ser: La navegación por medio de una cámara monocular como sensor principal (Aguilar, y otros, 2018), lo cual hace que no sea necesario ayudas externas como un GPS o marcadores visuales (Suárez-Sánchez & Loiza-Correa, 2015), un filtro Kalman extendido (Segarra, Caballeros, Aguilar, Samà, & Rodríguez-Martín, 2019) para la fusión de datos y estimación de estado y un controlador PID para generar comandos de dirección (Aguilar, Álvarez, Grijalva, & Rojas, 2019) (Engel, Sturm, & Cremers, 2014). Otros trabajos utilizan una cámara RGB-D para realizar la localización y mapeo en tiempo real de un UAV. SLAM (Simultaneous Localization and Mapping) (Aguilar, y otros, Real-Time 3D Modeling with a RGB-D Camera and On-Board

Processing, 2017) es una forma versátil de solucionar el problema, el cual es un importante reto en la historia de visión por computadora y robots móviles, el cual ha sido utilizado para extraer puntos de interés de las imágenes de la cámara y simplificarla para su asociación (Strasdat, Montiel, & Davison, 2010) (Cremers, Engel, & Sturm, 2012).

Una de las partes más importantes de los UAV es el aterrizaje automático, mediante el cual se puede ampliar las condiciones operativas del uso de UAVs permitiendo disminuir la necesidad de soporte manual (Azinhiera, Rente, & Kellet, 1998). En (Azinhiera & Rives, 2008) se busca realizar el aterrizaje automático realizando una estimación movimiento y posición del vehículo basado en imágenes. En (Wenzel, Masselli, & Zell, 2011) se realiza el seguimiento y aterrizaje sobre una plataforma en movimiento para lo cual utilizaron una cámara de infrarroja remota (IR) y un patrón de luces IR para el seguimiento del objetivo. En (Huh & Shim, 2010) los autores diseñaron un control no lineal utilizando el enfoque de inversión dinámica para el aterrizaje.

Otro desafío que se planea encarar es la detección de personas. Este es un tema de gran dificultad ya que en lugares complejos comúnmente se tienen escenarios con múltiples personas, oclusiones complicadas y fondos desordenados o incluso en movimiento (Andriluka, Roth, & Schiele, 2008). Se ha demostrado que los detectores de personas pueden localizar peatones incluso en escenas de calles complejas, pero los falsos positivos siguen siendo frecuentes (McKenna, Jabri, Duric, Rosenfeld, & Wechsler, 2000). Se debe entender que la detección de personas consta de 2 partes importantes, la extracción de características la cual se refiere a la abstracción de información de imágenes para tomar decisiones locales en cada punto de la imagen (Faytong, Moggia, & Vintimilla, 2009). Y la etapa de clasificación en el cual se discrimina la información obtenida para reconocer objetos de interés (Vargas Rojas, 2018).

Existen varios algoritmos de clasificación para la identificación de objetos de interés, a continuación, se presentan 4 ampliamente conocidos.

HAAR: Es un clasificador en cascada que se entrena para reconocimiento de objetos utilizando cientos de muestras, tanto positivas como negativas, para que el clasificador pueda reconocer la forma del objeto que se desea. Clasificador en cascada se refiere a que debe pasar por diferentes etapas que son aplicadas secuencialmente con un proceso de optimización (Aguilar W. G., y otros, Pedestrian Detection for UAVs Using Cascade Classifiers and Saliency Maps, 2017) (Visakha & Prakash, 2018).

SVM: Dentro de la tarea de clasificación, las SVM (Grijalva, Cueva, Ramírez, & G, 2019) (Galindo, Aguilar, & Reyes Ch, 2019) pertenecen a la categoría de los clasificadores lineales, puesto que inducen separadores lineales o hiperplanos, ya sea en el espacio original de los ejemplos de entrada, si éstos son separables o cuasi-separables (ruido), o en un espacio transformado (espacio de características), si los ejemplos no son separables linealmente en el espacio original (Avidan, 2004) (Suarez, 2014).

Neural Networks: Las redes neuronales son algoritmos de procesamiento que permite reconocer patrones en los datos, a partir de modelos que simulan los sistemas biológicos de aprendizaje. De manera general, una red neuronal está compuesta por unidades de procesamiento llamadas neuronas, distribuidas en diferentes capas, conectadas entre sí por una serie de pesos que establecen las relaciones entre ellas (Bayona & Salcedo, 2012) (Hinton, 1992).

ADABOOST: El nombre del algoritmo proviene de 'Adaptive Boosting' y se le llama así por su propiedad de adaptarse al nivel de error de cada algoritmo débil, es decir, tiene capacidad para utilizar el nivel de error de cada algoritmo para poner más o menos atención en él. La propiedad más importante de Adaboost es su habilidad para reducir el error de entrenamiento, el cual decrementa de manera exponencial conforme aumentan las rondas (Enríquez, 2006) (Schapire, 2013).

De igual manera se presentan algunos algoritmos de extracción de características tales como:

HOG: Histograma Orientado a Gradientes (HOG) es un algoritmo de extracción de características que convierte la información local de los gradientes para cada pixel, en una representación de la imagen que captura la forma global del objeto en un vector de características (Alcántara-Montiel, et al., 2019) (Catalán Vitas, 2017).

LBP: El operador de patrones locales binarios (LBP) obtiene información local en vecindades pequeñas de píxeles, basándose en el hecho de que, a menudo, la correlación entre píxeles disminuye a medida que aumenta la distancia entre éstos. Se trata de detectar patrones binarios locales en vecindades determinadas por una resolución espacial y una cuantización del espacio angular (Alvarado & Fernández, 2012) (Velasco Montero, 2016).

Tomando en cuenta los antecedentes mencionados y con el objetivo de superar las dificultades que representa la navegación y aterrizaje de un vehículo no tripulado, se propone en este proyecto de investigación una navegación gruesa basada en GPS y una navegación fina basada en percepción con una cámara monocular. Adicionalmente contará con un sistema de detección de personas en base a uno de los algoritmos detallados anteriormente, implementado en una estación en tierra (computadora).

### ***Justificación e importancia***

En el ámbito militar, la seguridad en instalaciones de almacenamiento de materiales explosivos es de alta importancia ya que pueden causar grandes daños por explosiones o en caso de robos puede significar una pérdida económica de millones de dólares (Montañez & Almendra, 2021).

El almacenamiento de explosivos, que por su naturaleza intrínseca generan un riesgo potencial de afectación al entorno circundante, se debe realizar en lugares seguros y aislados;

es decir, en los denominados polvorines, y estos deben estar contruidos específicamente para esta finalidad (López, 2021). No se debe permitir el almacenamiento de cantidades de explosivos que sobrepasen el 70 % de la capacidad del polvorín, ya que el 30 % restante debe destinarse para zonas de circulación y ventilación. No se deben almacenar en un mismo polvorín explosivos o agentes de voladuras que pertenezcan a grupos diferentes (Dasso Celis, 2016).

La instalación de cámaras y sensores para seguridad en estos espacios aislados tienden a ser de difícil instalación o muy costosos ya que toda instalación eléctrica debe ser a prueba de explosión, haciéndose desde el exterior por medio de claraboyas practicadas en la pared. Al existir aumentos de temperatura por estos dispositivos (cámaras y sensores), puede existir explosiones por lo cual no se suele optar por esta metodología de seguridad (Ejercito Ecuatoriano, 2018).

Como medida de seguridad anti-robos, estas instalaciones solo son accesibles durante un cierto horario determinado por las autoridades a cargo de la unidad militar. Al ser los polvorines instalaciones aisladas de otras, toma mucho tiempo identificar un robo, dar la alarma y actuar (León Urbina, 2020).

Normalmente se designa personal militar que realice rondas de patrullaje alrededor de estas zonas, sin embargo, esto tiende a ser ineficiente ya que, de presentarse una emergencia, la notificación de alarma normalmente llega muy tarde y además al tener a muchas personas patrullando en diferentes lugares no se cuenta con el suficiente personal para tomar acción (León Urbina, 2020) (Jara-Olmedo A. , Medina-Pazmiño, Tozer, Aguilar, & Pardo, 2018) (Jara-Olmedo A. , y otros, 2018).

Este sistema de seguridad aérea cuenta con varias ventajas en relación con sistemas de seguridad móviles terrestres como pueden ser:

Desplazamiento: Normalmente los cuarteles militares se ubican en zonas montañosas o de difícil acceso incluso para personas por lo cual la navegación alrededor de estas se torna una tarea difícil para un sistema terrestre (Caiza, Aguilar, Albán, & Fernández, 2020). Un drone puede desplazarse libremente por estas zonas siempre y cuando se le especifique bien el recorrido que debe realizar en el patrullaje (Montañez & Almendra, 2021).

Velocidad y autonomía: Si bien es cierto que un sistema terrestre cuenta con un mayor tiempo de autonomía que puede extenderse hasta horas, los drones pueden recorrer amplias distancias en una fracción de tiempo del que le tomaría al sistema terrestre, haciendo del sistema aéreo una alternativa mucho más veloz y eficiente (Katherine & Gutierrez Barboza, 2019) (Orbea, y otros, Math Model of UAV Multi Rotor Prototype with Fixed Wing Aerodynamic Structure for a Flight Simulator, 2017).

Costos y mantenimiento: Los sistemas terrestres para aplicaciones militares deben ser más robustos para soportar áreas de desplazamiento complejo y deben tener baterías enormes que aseguren el cumplimiento de la misión de reconocimiento. Esto hace que su costo sea demasiado elevado y que su mantenimiento sea igual de costoso ya que algunos elementos necesarios para dicho mantenimiento requieran ser importados de otros países o fabricados en su defecto (Katherine & Gutierrez Barboza, 2019).

Este proyecto de investigación se plantea el desarrollo de un sistema de seguridad autónomo por medio de vehículos aéreos no tripulado (UAV) con el fin de presentar una alternativa para la seguridad de polvorines en instalaciones militares. En base a los resultados obtenidos, se plantea extrapolar el sistema de seguridad a otras aplicaciones militares.

### ***Alcance del proyecto***

El presente proyecto de investigación tiene como objetivo el desarrollo de un sistema de patrullaje mediante el uso de un cuadricóptero. El cual contará con un sistema de navegación enfocado en 2 aspectos importantes:

El primero será la estimación de estado, la cual se refiere al proceso de seguimiento de posición actual del vehículo, que puede ser absoluta o relativa a la ubicación inicial (plataforma de despegue y aterrizaje) (Kendoul, 2012). En este aspecto se halla la navegación gruesa (navegación autónoma) la cual se la llevará acabo por medio de un control proporcional basado en un sistema de posicionamiento global (GPS) el cual proporciona al controlador de vuelo información de altitud y posición suficiente para poder opera en entornos libres de obstáculos (Romero Serna, 2015).

El segundo aspecto se referirá al sistema de percepción, en el que expone la necesidad que existe de reconocer el entorno en el que UAV va a situarse (Kendoul, 2012). Como se verifico en los antecedentes un aterrizaje autónomo permite disminuir la necesidad de soporte manual. Se llevará a cabo la navegación fina (aterrizaje automático) donde se realizará un control con base en las imágenes capturadas desde una cámara monocular y la información inercial de la IMU (Unidad de Medición Inercial) del UAV (Benini, Mancini, & Longhi, 2013). El objetivo es detectar una referencia específica en la plataforma de aterrizaje y estimar su posición con respecto al UAV. Esta posición relativa puede expresarse en un marco inercial en metros o en el marco de la imagen en píxeles (Kendoul, 2012).

Como parte del procesamiento de imágenes se utilizarán algoritmos para la detección de puntos de interés para estimar la posición mediante transformaciones geométricas (Martínez Redondo, 2016). Conjuntamente para determinar la posición se establecerá una estimación de movimiento del UAV, permitiendo realizar una compensación de movimientos para el seguimiento de la trayectoria de aterrizaje (Barranco, Fermüler, & Aloimonos, 2014).

Para la detección de personas se analizará los múltiples métodos de extracción de características y clasificación expuestos en los antecedentes de este proyecto. Procediendo a seleccionar e implementar la metodología que mejor se acople con el sistema de navegación. Los algoritmos desarrollados serán implementados en una estación en tierra la cual tendrá una comunicación inalámbrica mediante Wi-Fi con el UAV.

El sistema estará constituido por un cuadricóptero Bebop 2.0 y una estación de tierra encargada de procesar los datos del GPS, imágenes monoculares y datos inerciales para estimar la posición del UAV y enviar las señales de control, además de las imágenes que serán procesadas para verificar la detección de presencia de personas.

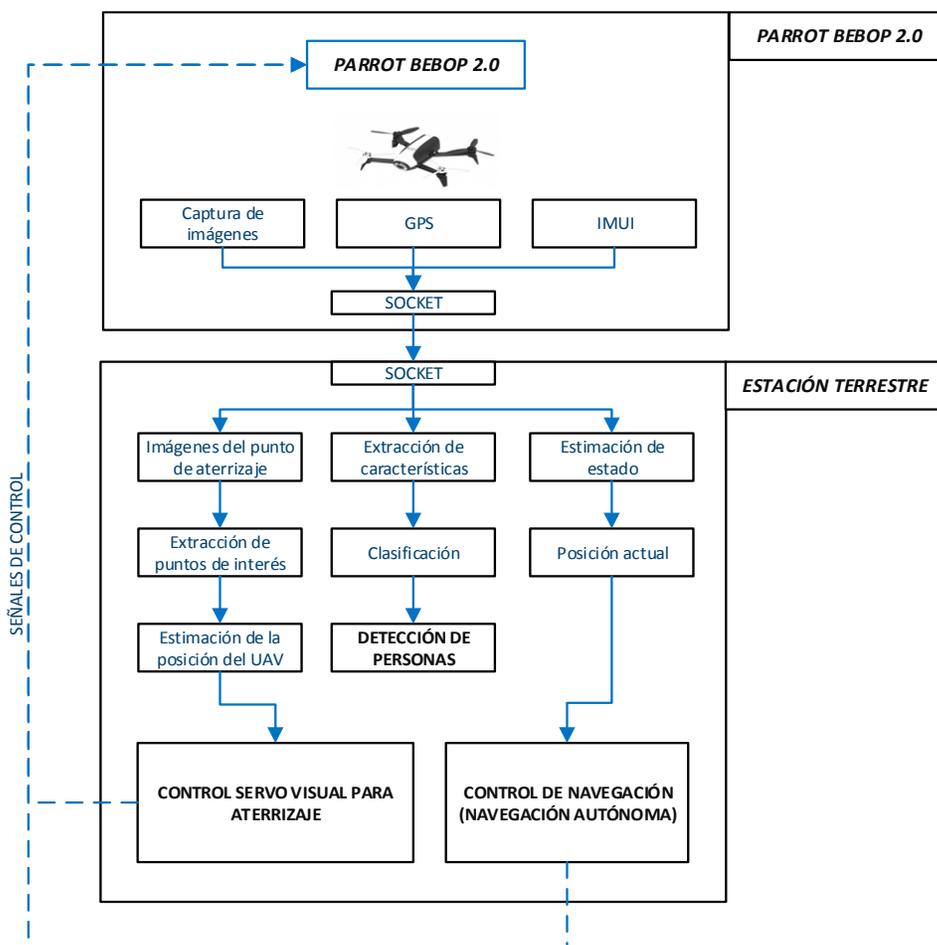
El área disponible para realizar las pruebas de funcionamiento tiene dimensiones de 45 metros de ancho y 90 metros de largo como máximo en espacio abierto. La plataforma de aterrizaje tendrá unas dimensiones de 841×1189 [mm]. La autonomía de vuelo del UAV es de 15 minutos a una velocidad de 7 a 8 km/h, considerando que el patrullaje se lo realizará cada hora permitiendo un lapso prudente de tiempo para un cambio de batería para el próximo patrullaje.

La comunicación se realizará mediante una ad-hoc creada por el Parrot Bebop 2.0 a la cual se conectará la estación terrestre. Para poder realizar la conexión ad-hoc se realizarán transacciones mediante protocolo TCP y un puerto específico. Para negociar la conexión se debe conectar un socket al puerto y enviar un objeto JSON (JavaScript Object Notation) que contenga la información del puerto, tipo y nombre del controlador e ID del dispositivo. El dispositivo responderá la solicitud con otra cadena JSON enviada por el mismo socket (Parrot SA, 2019).

La información disponible en la interfaz contará con visualización de imágenes, una alarma visual y auditiva que serán implementados en la estación en tierra comunicada con el UAV, la cual será desarrollada en Python 3.4 y Robot Operating System (ROS).

**Figura 1.**

*Esquema general del sistema de patrullaje*



## Objetivos

### Objetivo General

Desarrollar un sistema de patrullaje de drones para detección de personas en espacios restringidos.

### **Objetivos Específicos**

- Desarrollar el algoritmo para navegación autónoma basada en control del sistema de posicionamiento global (GPS).
- Desarrollar el algoritmo para el control servo visual mediante el análisis de fotogramas consecutivos para el aterrizaje del UAV con una estimación de movimiento para realizar una compensación en el seguimiento de la trayectoria de aterrizaje.
- Realizar un estudio sobre la extracción de características y clasificación para la detección de personas.
- Implementar el algoritmo de detección de personas basado en imágenes monoculares.
- Realizar pruebas experimentales para el sistema de navegación y aterrizaje autónomo.
- Evaluar el desempeño del algoritmo de detección de personas.

## Capítulo II

### Marco Teórico

#### ***Métodos de navegación de un UAV***

El uso de vehículos aéreos no tripulados autónomos (UAV) puede ser utilizado para cumplir ya sea tareas militares o tipo laboral, para lo cual conocer la posición, dirección y velocidad de la aeronave. Un conjunto de sensores y métodos de navegación nos va a permitir conocer información acerca del vehículo y el espacio por el cual se desplazará permitiéndole cumplir satisfactoriamente la misión de vuelo que le sea asignada (Concia, Zumarraga, & Mitidieri, 2019).

Debido a la gran exactitud, simplicidad y disponibilidad de los sistemas de posicionamiento global (GPS), es posible utilizarlos como principal sistema de navegación en los UAV (Lange, Sunderhauf, & Protzel, 2009). A lo largo de los años estos sistemas han progresado permitiendo dar ubicaciones con rangos de error de metros a centímetros el cual varía dependiendo el tipo de observaciones y métodos utilizados para su procesamiento. Es muy importante considerar el entorno en el cual va operar el GPS para garantizar la calidad de medidas, para lo cual es necesario mantener una línea de vista con los satélites que tienen enganchado al receptor del GPS (Foeng Vergel & Zabaleta Hernández, 2005).

La navegación no solo depende del posicionamiento global también está involucrado un sistema de navegación inercial (INS) el cual a partir de un punto inicial se estima las posiciones futuras mediante la velocidad, orientación y aceleración (Aguilar & Morales, 3D Environment Mapping Using the Kinect V2 and Path Planning Based on RRT Algorithms, 2016). El funcionamiento continuo de este sistema permite extraer en periodos de alta frecuencia de muestreo datos de altitud, rotación, aceleración (Zabala, Cuenca, León, & Cabrera, 2018). Para la obtención de estos datos se utiliza una unidad de medición inercial (IMU) el cual es un

conjunto de sensores conformado por acelerómetro, magnetómetro y giroscopio que viene integrado al UAV (Ferrer, 2009).

### ***Métodos de aterrizaje de un UAV***

En la navegación de UAV de forma autónoma es muy importante realizar el control de la aeronave en la etapa de aterrizaje ya que es una de las fases más críticas y delicadas, uno de los más populares son los controladores PID encargado de brindar estabilidad al sistema para el aterrizaje antes factores externos (Romero Paredes, 2014). Este control permite aterrizar la aeronave de forma autónoma pero no considera el área donde va a realizar el aterrizaje ya que es asistida por un supervisor o piloto que determina cuando es adecuado realizar un aterrizaje (Belmonte, 2019).

Otro método utilizado en el aterrizaje autónomo son los sistemas de detección basados en visión, la mayor parte de aeronaves viene equipado con un sensor de cámara el cual permite estimar la velocidad y distancia al área de aterrizaje, mediante la utilización de esta información el UAV puede alinearse a la plataforma de aterrizaje modificando su velocidad y aceleración permitiendo un aterrizaje seguro (Fernández, 2016) (Lee, Ryan, & Kim, 2012).

### ***Algoritmos de detección de personas***

Para la identificación y detección de objetos, existen múltiples metodologías, pero no se puede decir que un algoritmo sea mejor que otro ya que se puede seleccionar el algoritmo que se adapte mejor a los requerimientos de cada proyecto. Los sistemas de detección de objetos han experimentado múltiples cambios a través de los años para mejorar el rendimiento, velocidad y precisión, debido al desarrollo continuo de investigadores interesados en esta área, gracias a esto se han mejorado los algoritmos en aplicaciones populares como detección de personas, imágenes médicas, robótica y detección de rostros (Adarsh, Rathi, & Kumar, 2020) (Barrachina Jordá, 2021).

Dentro de los algoritmos desarrollados podemos encontrar Haar Cascade, HOG (Histogramas de gradientes orientados) y YOLO (You Only Look Once).

### ***Haar Cascade***

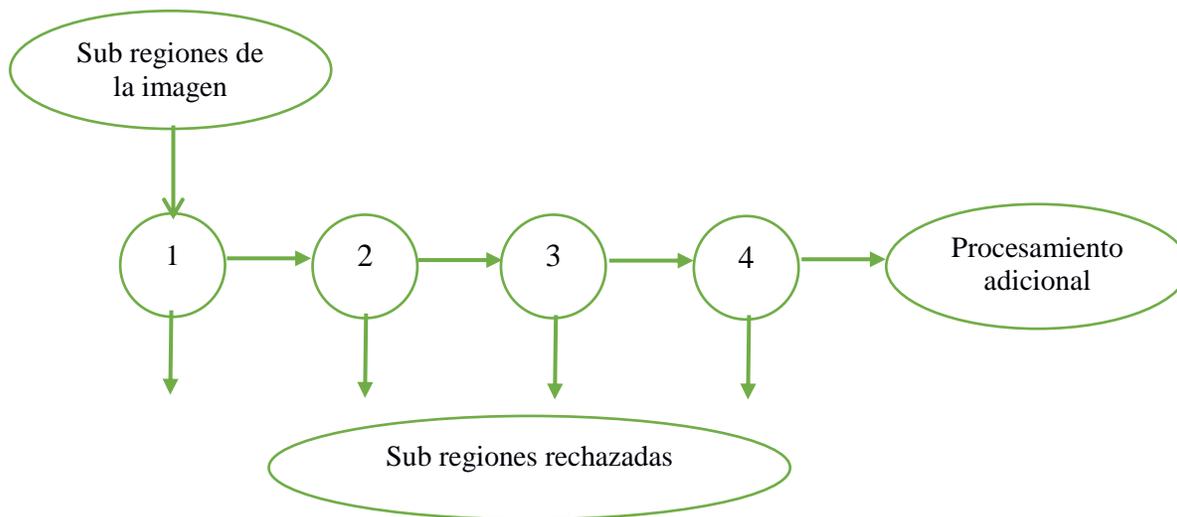
La base para la detección de objetos mediante Haar Cascade es mediante la utilización de características Haar-like. Este extractor de características utiliza el cambio de contraste en un grupo de píxeles adyacentes. La diferencia de contraste en este grupo de píxeles es usada para determinar áreas relativamente claras y oscuras. Dos o tres grupos con una variación de contraste forman una característica Haar-Like. Cada una de estas características son utilizadas para detectar una imagen (Jalied & Voronkov, 2016) (Lienhart & Maydt, 2002) (Wilson & Fernandez, 2006).

Un clasificador en cascada consiste en una serie de etapas, en donde se desarrolla una pequeña porción de procesamiento, en la Figura 2 se puede visualizar la estructura del clasificador de forma gráfica. El sistema detecta objetos que se van moviendo a través de la imagen mediante ventanas (Jalied & Voronkov, 2016). Cada ventana de detección etiqueta una región de la imagen como positiva o negativa, positiva si el objeto fue encontrado en esa región o negativa si el objeto no fue encontrado (Guevara, Echeverry, & Urueña, 2008).

Si el clasificador etiqueta una región como negativa la ventana de detección se mueve a una nueva región de la imagen. Para una etiqueta positiva el clasificador procede a mover el clasificador a la siguiente etapa. Para dar un veredicto final de positivo de que un objeto se ha encontrado dentro de la imagen, cuando todas las etapas de las cuales está conformado el clasificador entrega un resultado de que el objeto se encuentra en la imagen (Soo, 2014).

**Figura 2.**

*Etapas de un clasificador Haar Cascade (Soo, 2014).*



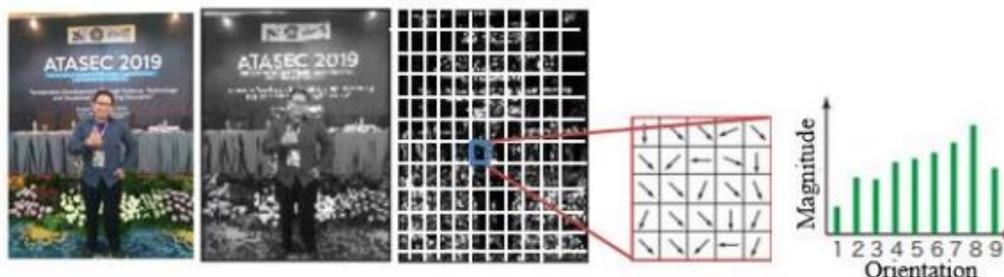
Nota: En la figura se observa cómo está estructurado un clasificador en cascada.

### **HOG + SVM**

Los Histogramas de Gradientes Orientados (HOG) son un descriptor de características que ha sido usado exitosamente utilizado para la detección de objetos y personas (Steinera, Shapiro, & Destefanis, 2019). Este permite representar un objeto como un vector de valor único contrario a un conjunto de vectores de características donde se representa una región de la imagen, el cual es calculado por el computador deslizando una ventana detectora a través de la imagen (Rahmad, et al., 2020). En la Figura 3 se muestra la extracción de características completa.

### Figura 3.

*Extracción de características HOG (Rahmad, et al., 2020).*



Nota: En esta figura verificar el proceso para la extracción de características HOG.

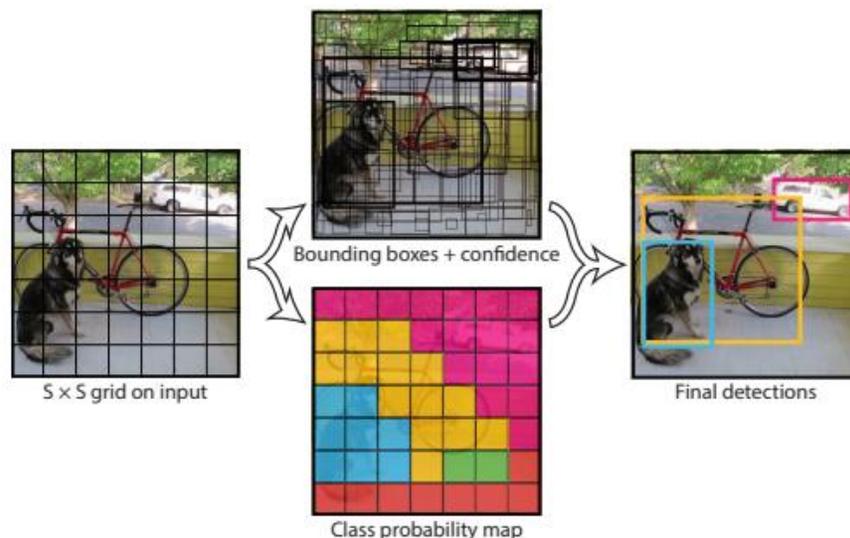
Un clasificador SVM (Maquina de Vectores de Soporte) con un extractor de características HOG es uno de las técnicas más populares utilizadas, los SVM son conocidas como máquinas de aprendizaje supervisado implementadas para tareas de clasificación y regresión (Martínez, 2018).

### Yolo

También conocido como You Only Look Once. Es una nueva forma para la detección de objetos, la cual es capaz de predecir que objetos y donde se encuentran de un solo vistazo (Bathija & Sharma, 2019). En la Figura 4 se modela la detección como una regresión, en donde se separan cuadros delimitadores y probabilidades de clase, de esto está encargada una sola red neuronal convolucional encargada de predecir estas probabilidades de clase asociadas (Du, 2018). En la Figura 4 se muestra el proceso de detección de objetos mediante Yolo.

**Figura 4.**

*Proceso de detección de objetos mediante Yolo*



Nota: El gráfico muestra el proceso realizado por Yolo para la detección de objetos, dividiendo la imagen en una matriz  $S \times S$ , para cada cuadrícula de la matriz predice cuadros delimitadores para otorgarles una probabilidad de clase (Redmon, Divvala, Girshick, & Farhadi, 2016).

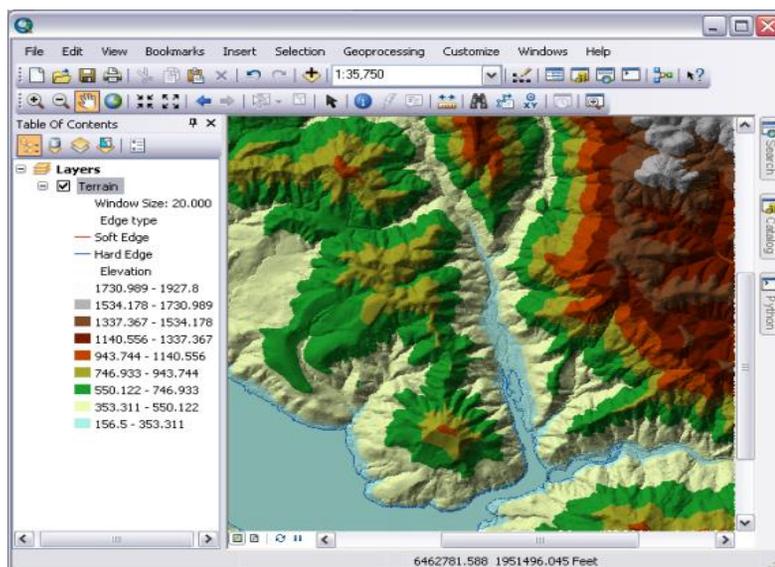
### ***Software de análisis y manipulación de datos geográficos ArcGis***

Con más de un millón de usuarios de software en todo el mundo e instalaciones en más de 5.000 universidades, Environmental Systems Research Institute, Inc. (ESRI), establecido en 1969, es líder mundial en el diseño y desarrollo de software de Sistemas de Información Geográfica (GIS) (Scott & Janikas, 2010). La tecnología GIS permite la organización, manipulación, análisis y visualización de datos espaciales, a menudo descubriendo relaciones, patrones y tendencias (Scott & Janikas, 2010). ArcGis es una herramienta importante para muchos proyectos de ingeniería como en el monitoreo del estado de carreteras y puentes para la creación de mapas en planificación para desastres naturales, mapeo de zonas para instalación de antenas y georreferenciación (Booth & Mitchell, 2001) (Basantes, y otros, 2018)

(Zurita, Aguilar, & Enríquez, 2019). Las técnicas de análisis de ArcGis incluyen consultas espaciales, superposición de mapas, análisis de búfer, interpolación y cálculos de proximidad (Childs, 2004). Junto con las herramientas cartográficas y de gestión de datos básicas, estas técnicas analíticas han sido durante mucho tiempo la base del software de información geográfica (Scott & Janikas, 2010). En la Figura 5 se muestra la interfaz gráfica de ArcGis.

### Figura 5.

*Interfaz gráfica de ArcGis (Scott & Janikas, 2010).*



## Capítulo III

### Descripción del Sistema

#### ***Descripción de Hardware***

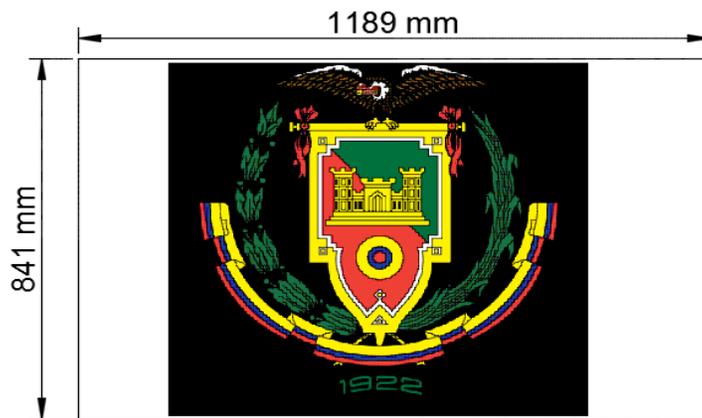
El sistema de patrullaje autónomo consta de dos estaciones, una aérea y una terrestre. La estación terrestre costará del cuarto de control asignado para monitoreo y seguridad. La estación aérea constará del cuadricóptero o UAV comunicado por medio de Wi-Fi a la estación terrestre.

#### **Estación terrestre.**

La estación terrestre que se propone en este proyecto de titulación consta de un cuarto de control equipado con un ordenador Lenovo Y700 que cuenta con un procesador Intel® Quad-Core™ i7 vPro 6° generación, una memoria RAM de 16GB DDR4 y disco duro de 1TB. En la parte exterior del cuarto de control se deberá situar una plataforma de 841×1189 mm que puede ser hecha de cualquier material sólido como madera, concreto, etc. Esta tendrá un logo distintivo con colores vivos o diferentes a su entorno para que el drone de patrullaje pueda identificar su zona de aterrizaje y no se confunda con otros elementos que puedan existir en su entorno (Gómez Redondo, 2020). En la Figura 6 se ilustra un ejemplo del diseño de una plataforma.

**Figura 6.**

*Ejemplo de plataforma y dimensiones propuestas para el proyecto*

**Estación aérea (UAV).**

La estación aérea de constará de un cuadricóptero Bebop 2 de la marca Parrot. En la Figura 7 se ilustra la apariencia del cuadricóptero seleccionado.

**Figura 7**

*Cuadricóptero Bebop Parrot 2 (Parrot SA, 2019)*



Una de las características más llamativas de este cuadricóptero es que es un UAV de código abierto el cual da apertura a desarrollo de software y aplicaciones de vuelo tal como se

requiere en este proyecto de titulación (Aguilar, y otros, Visual SLAM with a RGB-D Camera on a Quadrotor UAV Using on-Board Processing, 2017).

### **Características Bebop Parrot 2**

En la tabla 1 se especifican las características más relevantes de este cuadricóptero en cuestión.

**Tabla 1**

*Características de cuadricóptero Bebop Parrot 2 (Grijalva S. , 2018)*

| <b>Características</b>       |   |
|------------------------------|---|
| <b>Peso</b>                  | 500 Gr  |
| <b>Autonomía de vuelo</b>    | 25 min  |
| <b>Material</b>              | Fibra de vidrio   |
| <b>Elementos del Sistema</b> |   |
| <b>Sensores</b>              | GPS integrado, IMU  |
| <b>CPU</b>                   | Dual Core con GPU Quad-Core   |
| <b>Almacenamiento</b>        | 8 GB de almacenamiento flash  |
| <b>Interfaz</b>              | Aplicación vía Smart-Phone o computadora                              |
| <b>Construcción</b>          |   |
| <b>Motores</b>               | 4 motores sin escobillas Outrunner                                    |
| <b>Estructura</b>            | PA12 estructura reforzada de fibra de vidrio (20%) y Grilamid (Casco) |
| <b>Peso</b>                  | 500 gramos, batería :192 gramos                                       |
| <b>Autonomía de Batería</b>  | 2700 mAh  |
| <b>Dimensiones</b>           | 38 x 33 x 9 cm  |
| <b>Conectividad</b>          |   |
| <b>Estándar</b>              | Wi-Fi IEEE 802.11 a/b/g/n/ac  |
| <b>Antena Wi-Fi</b>          | MIMO de banda dual con 2 antenas dipolo dual de 2.4 y 5 GHz           |

|                            |  |
|----------------------------|--|
| <b>Alcance de la señal</b> | 300m con dispositivo móvil o hasta 2Km con Skycontroller 2 |
|----------------------------|--|

### **Características de movimiento**

El Bebop Parrot 2 cuenta con 4 tipos de movimiento básicos que serán implementados en el sistema de patrullaje. En la tabla 2 se especifican los rangos de movimiento junto a su velocidad para ejecutarlos.

**Tabla 2**

*Movimientos básicos de Bebop Parrot 2 (Grijalva S. , 2018).*

| <b>Ilustración de movimiento</b>  | <b>Tipo de movimiento</b>              | <b>Velocidad de movimiento</b>    |
|---|--|-----------------------------------|
|    | Movimiento lineal en el eje X          | Máximo: 16 m/s                    |
|  | Movimiento lineal en el eje Y          | Máximo: 16 m/s                    |
|  | Movimiento lineal en el eje Z          | Mínimo: 0.5 m/s<br>Máximo: 6 m/s  |
|  | Movimiento angular alrededor del eje Z | Mínimo: 10 °/s<br>Máximo: 200 °/s |

### **Características de la cámara para detección de personas**

La cámara a ser implementada en el reconocimiento de personal no autorizado es la integrada en el Bebop Parrot 2. Las características de mayor relevancia que se buscan son el área de cobertura y resolución ya que dependiendo de estas, se determinara la calidad del

reconocimiento de personas en plena misión de vuelo. La tabla 3 muestra las características de la cámara incorporada en el cuadricóptero.

**Tabla 3**

*Características de cámara de Bebop Parrot 2 (Vera, Andrés, Castillo, & Geovanny, 2020).*

| <b>Características</b>             |  |
|------------------------------------|--|
| <b>Resolución</b>                  | 14 megapíxeles<br>3800 x 3188            |
| <b>Lente</b>                       | Gran angular                             |
| <b>Angulo de visión total</b>      | 180°                                     |
| <b>Angulo de visión disponible</b> | Eje vertical: 80°<br>Eje horizontal: 50° |
| <b>Resolución de video</b>         | 1920x1080p (30fps)                       |
| <b>Codificación del video</b>      | H264                                     |
| <b>Formato de imágenes</b>         | JPEG, RAW, DNG                           |

### ***Descripción de Software***

En este proyecto de titulación se implementó una serie de recursos de software instalados en el sistema operativo Ubuntu. Específicamente se utilizó *Linux Ubuntu* 16.04 LTS. Esta distribución de Ubuntu es gratuita y de software abierto con licencia de GNU (General Public Licencia), junto con varias otras licencias gratuitas (Tabassum & Mathew, 2014). Esta versión, aunque no es la más actual del mercado, cuenta con documentación detallada acerca de la instalación y compatibilidad de programas relacionados con la programación de cuadricópteros como el Bebop Parrot 2. Los programas como ROS no presentaron ninguna incompatibilidad con esta versión de Linux.

## **Robot Operating System (ROS)**

El Sistema Operativo de Robot (ROS) es un conjunto de bibliotecas de software y herramientas que se utilizan para construir sistemas robóticos. ROS es conocido por poseer un diseño distribuido y modular (Cashmore, et al., 2015) (Quigley, et al., 2009). Su enfoque es el proporcionar servicios para trabajar con elementos de hardware, drivers y controladores de dispositivos vinculados, servicios de transmisión de mensajes entre procesos, además de un conjunto de herramientas y bibliotecas para desarrollo, visualización y administración (Vera, Andrés, Castillo, & Geovanny, 2020). ROS está basado en una arquitectura de grafos que le permite la multiplexación de la información, en otras palabras, el procesamiento se lleva a cabo en "nodos" independientes, mismo que pueden publicar y suscribirse a mensaje de otros nodos (Aguilar, y otros, On-Board Visual SLAM on a UGV Using a RGB-D Camera, 2017). La independencia de lenguaje de programación es una gran característica de ROS tales como la implementación de esta en *Python*, *C++* y *Lisp* (Galarza, Pérez, Serrano, Tapia, & G, 2018).

## **Comunicación del Sistema**

Para realizar la comunicación entre la estación de tierra y Bebop 2 es necesario la implementación de ROS el cual es un sistema meta-operating de código abierto para controlar robots, el cual brinda las mismas bondades de un sistema operativo como es la abstracción de hardware, control de dispositivos de bajo nivel, implementación de funciones de uso común y gestión de paquetes. ROS permite el desarrollo de nuevos paquetes con librerías que permiten la compatibilidad a diferentes robots. Uno de los drivers disponibles en ROS es el *bebop\_autonomy* compatible con Parrot Bebop 1 y 2, el cual es basado en ARDroneSDK3 oficial de Parrot (Daramouskas, Perikos, & Hatzilygeroudis, 2012). Este driver fue desarrollado por Autonomy Lab de Simon Fraser University por Mani Monajjemi y otros contribuidores. El cual es mantenido gracias a los aportes de la Universidad Simon Fraser, Universidad de Toronto y el Laboratorio Interactivo de Tecnologías de Zurich.

El driver para la comunicación puede ser ejecutado como nodo o un nodelet, un nodo permite realizar varios procesos computacionales y se comunica con otros nodos mediante topics mientras que un nodelet puede ejecutar múltiples algoritmos o nodos en el mismo proceso. El driver de Bebop recomienda el uso de un nodelet al realizar algún tipo de procesamiento en la transmisión de video (Valseca Martínez, 2018).

Para realizar el levantamiento del nodelet es necesario la instalación del driver `bebop_autonomy` y que el computador se encuentre en la misma red Wi-Fi que le proporciona el UAV. Desde el terminal de Ubuntu se utiliza como fuente al driver de `bebop_autonomy` ingresando el comando:

```
source bebop_ws/devel/setup.bash
```

Añadido la fuente procederemos a ejecutar el nodelet mediante el comando `roslaunch` el cual permite ejecutar archivos de extensión `.launch`, al ejecutar este nodelet tendremos acceso a los topics del UAV mediante el comando `rostopic list` y se desplegara una ventana la cual nos permitira visualizar la trasmision de la camara, para ejecutar el nodelet ingresamos el comando:

```
roslaunch bebop_tools bebop_nodelet_iv.launch
```

## Capítulo IV

### Sistema de Navegación

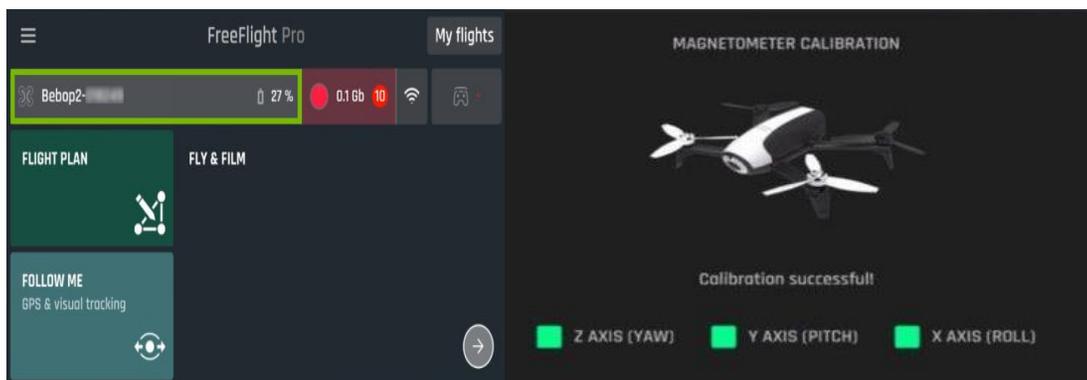
#### Navegación Gruesa

#### Navegación basada en GPS

Una de las partes más importantes del sistema de navegación desarrollado es el GPS incorporado. El Bebop Parrot 2 consta de un GPS estándar que entrega tanto las coordenadas de longitud como las coordenadas de latitud. Para que estas coordenadas sean precisas, primero se debe calibrar el magnetómetro del cuadricóptero utilizando la aplicación libre FreeFlight PRO. En la Figura 8 se puede observar el proceso de calibración mediante la aplicación en cuestión.

#### Figura 8.

*Interfaz gráfica de la aplicación FreeFlight PRO y función de calibración (Parrot SA, 2019)*

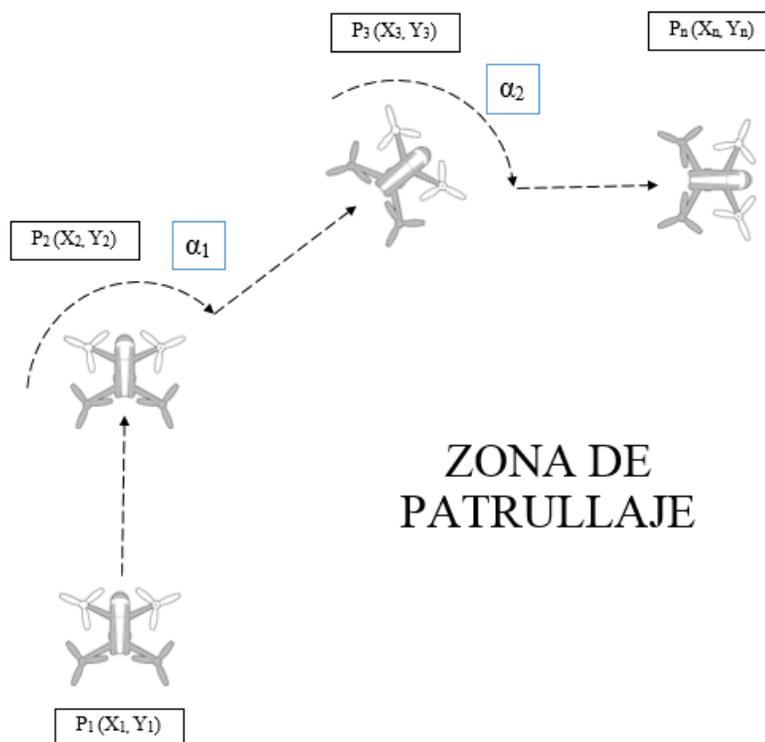


Una vez calibrado el dron, se debe especificar las coordenadas de los vértices por los cuales se requiere que el cuadricóptero recorra para completar su misión de vuelo. El desplazamiento del dron de un punto a otro será de manera rectilínea y una vez llegado a uno de los vértices este calculará el ángulo que debe girar para estar totalmente alineado entre la coordenada en el que se encuentra y la siguiente coordenada a la que se debe dirigir.

En la Figura 9 se explica los movimientos que debe realizar el cuadricóptero durante su misión de vuelo.

### Figura 9.

*Ejemplo de movimiento misión de vuelo del drone*

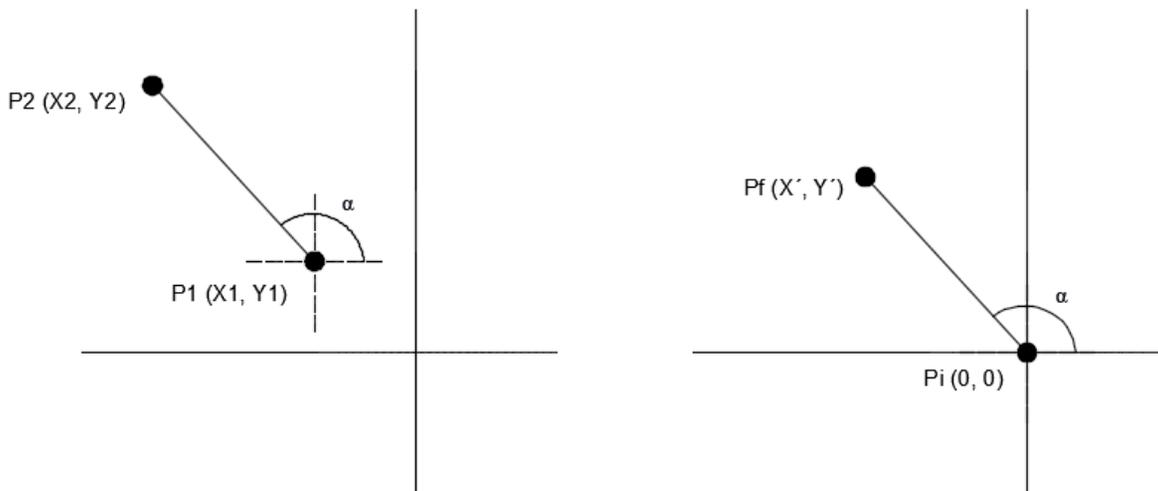


Nota.  $P_n$ : Punto o coordenada parte de la misión de vuelo.  $X_n$ : Valor de longitud de la coordenada.  $Y_n$ : Valor de la latitud de la coordenada.  $\alpha_n$ : Valor del ángulo de rotación realizado por el dron.

El ángulo de giro que debe realizar el dron en cada uno de los vértices se lo calcula tomando como coordenada de inicio un punto  $P_i(0, 0)$  y un punto final  $P_f(X', Y')$ . Estos puntos nacen de la translación al origen de las coordenadas  $P_1$  y  $P_2$ , que son los puntos de inicio y fin de la trayectoria (Enrique, 2017) (Cabras, Rosell, Pérez, Aguilar, & Rosell, 2011). En la Figura 10 se ilustra cómo entiende el cuadricóptero está translación.

**Figura 10.**

*Traslación de puntos al origen para facilitar el cálculo del ángulo  $\alpha$  (Enrique, 2017).*



Nota.  $P_f$  y  $P_i$  representan los puntos desplazados al origen.

Como podemos observar en la figura xx, los puntos  $P_f$  y  $P_i$  se los calcula con las siguientes ecuaciones.

$$P_f = P2 - P1 \quad P_i = P1 - P1 \quad (1)$$

Estas nuevas coordenadas tienen valores diferentes a las coordenadas originales pero el ángulo  $\alpha$ , que es de nuestro interés, se mantiene intacto.

Con esta translación el ángulo  $\alpha$  puede ser calculado fácilmente con la ecuación.

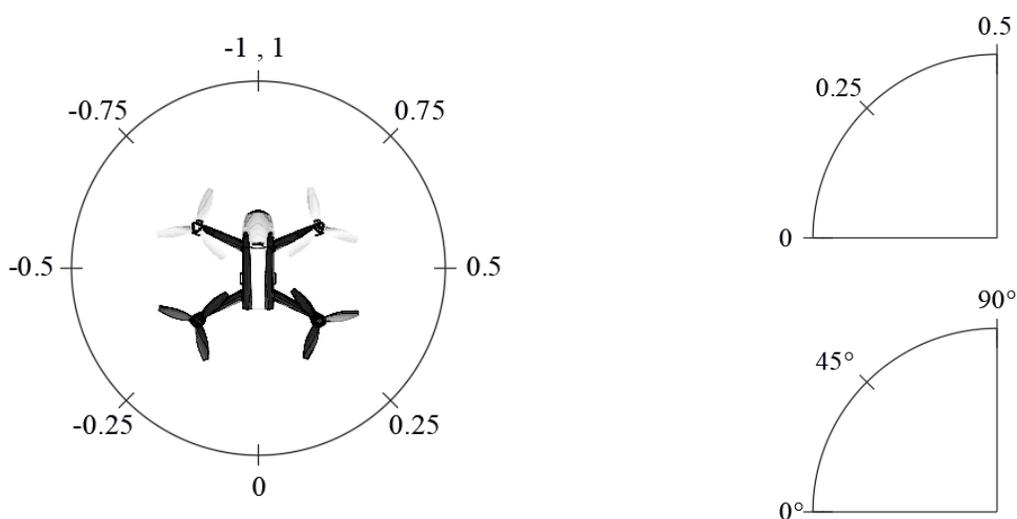
$$\alpha = \left( \frac{Y'}{X'} \right) \quad (2)$$

Una vez calculado el ángulo de giro, se debe transformar esta información al formato de la odometría del cuadricóptero. Como ya se mencionó previamente, la odometría es forma en la que el dron determina su posición relativa respecto a su localización inicial. Esta nos brinda información de posición, ángulo de inclinación, ángulo de giro en sus ejes, etc. (Jiménez &

Baturone, 1996). Con el fin de orientar al dron en la dirección requerida se obtendrá la dirección del cuadricóptero en el momento preciso de giro y se le sumará el ángulo previamente calculado para alinearlo al siguiente punto al que debe dirigirse. En la siguiente figura se ilustra los valores de la odometría que muestran la dirección del mismo.

**Figura 11.**

*Datos de odometría entregados por Bebop Parrot 2 y equivalencia en grados*



a) Valores de odometría

b) Equivalencia en grados.

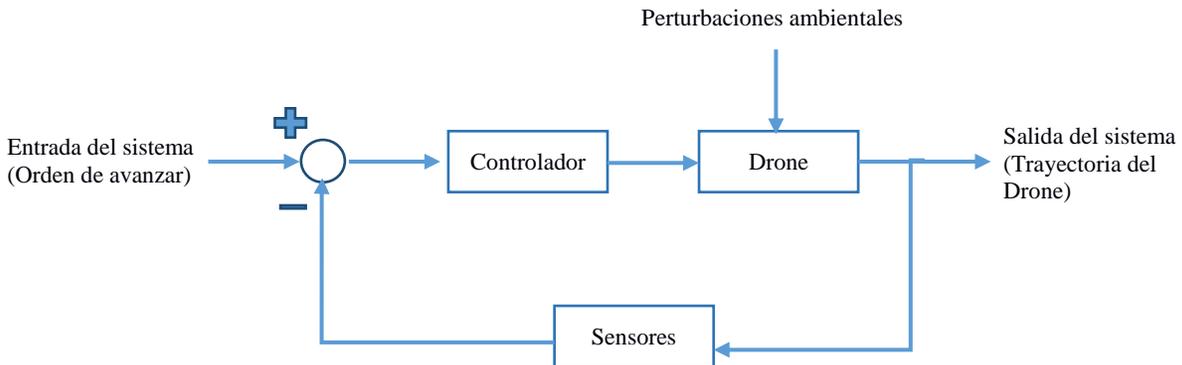
Nota. Los datos entregados por el dron son adimensionales y solo indican la dirección en la que se encuentra en ese momento.

### **Diseño del controlador de navegación autónoma.**

En cada uno de los tramos que debe recorrer el cuadricóptero existe la posibilidad de que se generen vientos, llovizna u otros factores climáticos provocando que el dron se desvíe de su trayecto y cambie totalmente su plan de vuelo. Para este proyecto se propuso la implementación de un control ON/OFF para mitigar este inconveniente (Ogata, 2010). En la Figura 12 se establece el diagrama de bloques del control a ser implementado.

**Figura 12.**

*Diagrama de bloques del sistema de control autónomo*

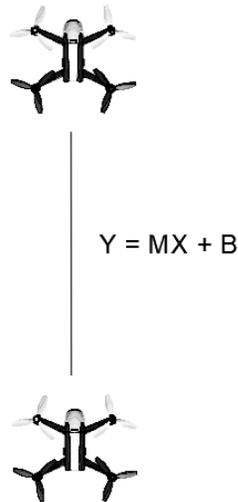


Como podemos observar en la figura 12 la entrada del sistema será la instrucción de avanzar hacia delante el dron y su salida será la trayectoria que debe tomar el mismo. El controlador se encargará de hacer que el cuadricóptero se desplace a la derecha o izquierda si sale de su trayectoria. Y una vez se haya realizado el desplazamiento, los sensores de posicionamiento del dron indicarán al controlador si se debe volver a corregir la trayectoria o solo mantenerla normalmente.

Para el diseño del controlador, primero modelaremos su movimiento en una ecuación que describa su trayectoria de un punto a otro. En la Figura 13 se muestra este modelamiento junto a sus variables.

**Figura 13.**

*Ecuación de movimiento del cuadricóptero (Arauz)*



Nota. Los elementos que se encuentran en la ecuación son:

- Y: Coordenada de latitud del drone
- M: Pendiente entre puntos
- X: Coordenada de longitud del drone
- B: Coordenada de latitud de posición inicial del drone

Como podemos observar, el movimiento que queremos que realice el drone es un movimiento rectilíneo por lo cual la ecuación de su trayectoria es una recta (Zambrano, 2010). Al obtener los datos de la coordenada que debe seguir el drone, si este se sale de esa trayectoria recta, automáticamente el controlador le indicará si debe moverse a la izquierda o a la derecha mientras avanza en su misión de vuelo (Aranda Romasanta, 2017).

***Navegación Fina***

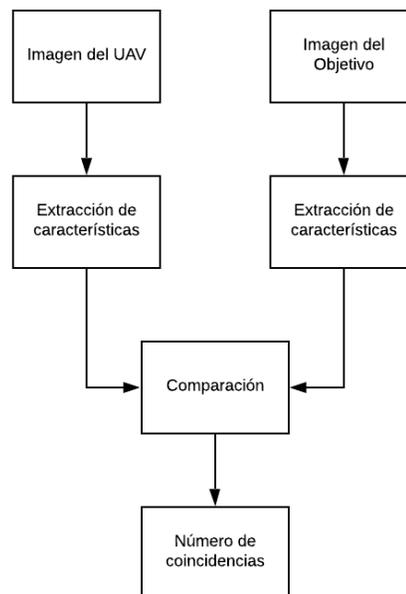
Para realizar la navegación fina del drone es necesario realizar un tratamiento de las imágenes para una buena detección y seguimiento a los puntos de interés de la plataforma de

aterrijaje. La identificación del objetivo se la realiza mediante algoritmo de detección de características en (Aguilar, Casaliglla, & Pólit, Obstacle Avoidance Based-Visual Navigation for Micro Aerial Vehicles, 2017) (Amaguaña, Collaguazo, Tituaña, & Aguilar, 2018) se hace una comparación entre los tres algoritmos más conocidos para la caracterización de imágenes SIFT, SURF y ORB para determinar cual tiene un mejor desempeño, en sus resultados determinaron que el algoritmo ORB permite optimizar el procesamiento con un alto desempeño, el criterio que utilizaron es la cantidad de puntos que puede detectar en el tiempo con múltiples cambios de contraste (Dang, et al., 2015) (Rubleee, Rabaud, Konolige, & Bradski, 2011).

Considerando el análisis realizado por salcedo (Salcedo, Aguilar, Cobeña, Pardo, & Zahira, 2018) (Aguilar, Salcedo, Sandoval, & Cobeña, 2017) se procedió a diseñar un algoritmo para la detección del área de aterrijaje el cual esta descrito en la Figura 14.

#### Figura 14.

*Diagrama de bloques extracción de características y detección del objetivo*



Nota: Imagen del UAV es la imagen obtenida de la transmisión de video del dron; Imagen del objetivo es la imagen precargada de la plataforma de aterrizaje.

Como se observa en la Figura 14 los puntos extraídos de la imagen del UAV y de la imagen del objetivo deben ser comparados pueden presentar cambios o deformaciones los cuales son conocidos como cambios geométricos relacionados entre sí, por lo cual existe una transformación geométrica. OpenCV permite realizar el cálculo de las transformaciones geométricas recurriendo a la transformada de homografía la cual permite representar cualquier deformación en un espacio euclidiano realizando una estimación de la transformada entre dos conjuntos de puntos. La función “findHomography” entregará los cambios geométricos sufridos entre la imagen que visualiza el UAV y la imagen del objetivo, conjuntamente esta función trabaja con el método RANSAC el cual descartará los puntos que no correspondan al objetivo y entregará únicamente los puntos correctos o correspondientes (Bradski & Kaehler, 2000).

Obtenidos los puntos correspondientes se realiza la transformación de perspectiva conjuntamente con la matriz de homografía trasladando el recuadro a las dimensiones necesarios para encapsular el objetivo. En esta etapa se compara los puntos de interés de cada fotograma con los puntos de interés hasta que superen el umbral que permite detectar el área de aterrizaje y delimitar el área objetivo a través de un recuadro, logrando con esto que no se pierda el objetivo.

Para realizar el modelamiento para el aterrizaje del UAV en una plataforma tomaremos en consideración trabajos previamente establecidos en (Tenezaca, Canchignia, Aguilar, & Mendoza, 2019) donde se proponen algoritmos para la estabilización de video utilizados en tiempo real durante la navegación de UAVs, permitiendo mantener una imagen estable simultáneamente se realizan movimientos reales, estos algoritmos fueron aplicados por (Merizalde, Aguilar, & Calderón, 2020) para modelar un sistema de aterrizaje de un dron

basado en estimación de movimiento entre fotogramas consecutivos (Aguilar, Manosalvas, Guillén, & Collaguazo, 2018).

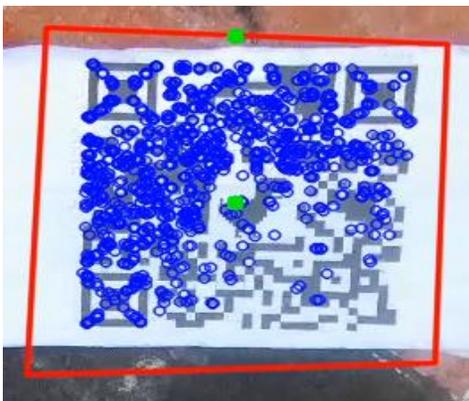
En el desarrollo es necesario considerar la fidelidad de los datos obtenidos y que el estado del drone sea similar o más veloz que el procesamiento de imágenes debido a que se deben tomar acciones de control sobre las velocidades, para lo cual no es posible recurrir a la unidad de medición inercial (IMU) incorporada al Parrot Bebop 2, el cual transmite los datos de navegación al controlador ROS a una tasa de refresco de 5Hz lo cual hace que el control ocurra a una baja frecuencia. Debido a este problema se ha optado por la estimación de movimiento basado en visión artificial para poder realizar el modelamiento del controlador del drone.

### **Diseño del controlador a partir de estimación de movimiento**

Para tomar una acción de control sobre el UAV se utilizara los datos de la cámara que son enviados a una frecuencia de 30Hz los cuales son convenientes para obtener un modelo servo visual mediante el cual se puede aproximar los movimientos independientes que posee un UAV como un modelo SISO (Aguilar, Angulo, & Costa-Castello, Autonomous Navigation Control for Quadrotors in Trajectories Tracking, 2017) (Grijalva & Aguilar, Landmark-Based Virtual Path Estimation for Assisted UAV FPV Tele-Operation with Augmented Reality, 2019) (Aguilar & Angulo, Real-Time Model-Based Video Stabilization for Microaerial Vehicles, 2016). En la Figura 15 se puede visualizar la plataforma de aterrizaje sobre la cual debe descender la aeronave para lo cual es necesario un controlador PID para controlar el posicionamiento tomando como referencia el centro de la plataforma de aterrizaje. El posicionamiento se lo realizara mediante movimientos a lo largo del eje X y eje Y, para lo cual es necesario dos controladores para generar movimientos en Pitch (eje X) y en Roll (eje Y).

**Figura 15**

*Área de aterrizaje del UAV.*



Nota: En la Figura se visualiza el área delimitada e identificada como zona de aterrizaje.

Para el modelamiento alrededor del eje X se publica un tren de impulsos en el tópic `/bebop/cmd_vel` que se asemeja a un movimiento a través del eje `linear.x` y se procede a grabar el movimiento generado por el tren de impulsos mediante la cámara a bordo del dron, en la Tabla 4 se especifican las condiciones mediante las cuales se realizó la grabación del video.

**Tabla 4**

*Condiciones de operación para modelamiento.*

| Iluminación | Altura | Angulo cámara | Estado                     |
|-------------|--------|---------------|----------------------------|
| 2000 lux    | 1.5m   | -90°          | Desplazamiento en el eje X |
| 2000 lux    | 1.5m   | -90°          | Desplazamiento en el eje Y |

Obtenido el video se procede a realizar el proceso de estimación de movimiento desarrollado en (Aguilar & Angulo, Real-time video stabilization without phantom movements for

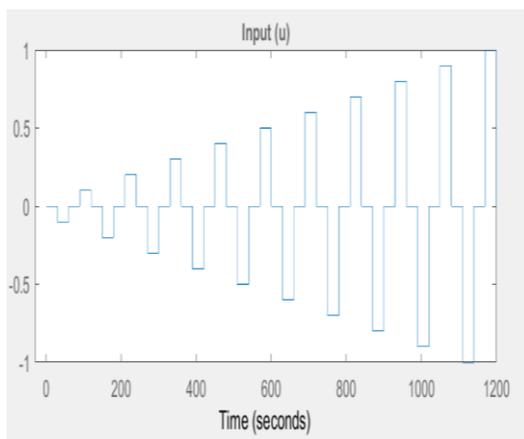
micro aerial vehicles, 2014) en el cual se realiza la estimación de fotograma a fotograma realizando una comparación de los puntos de interés mediante la función `cv2.estimateAffinePartial2D` de OpenCV en el cual se obtiene una matriz de transformación o matriz de similitud.

$$M = \begin{pmatrix} \cos \theta . s & -\sin \theta . s & t_x \\ \sin \theta . s & \cos \theta . s & t_y \end{pmatrix} \quad (3)$$

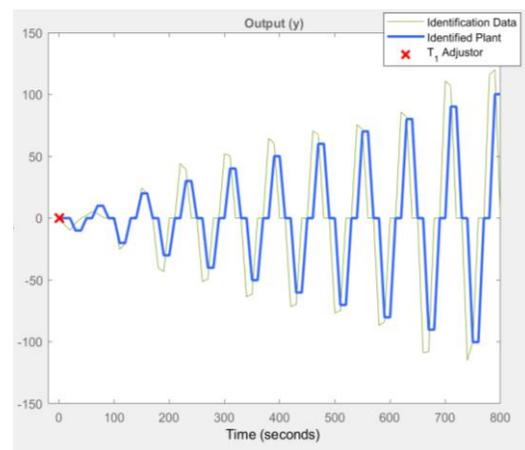
A partir de la matriz de transformación de similitud obtenemos el ángulo de rotación que existe entre frame a frame y se grafica la función de estimación de rotación para el tren de impulsos ingresado al UAV. Las gráficas para el movimiento para el tren de impulsos se visualizan en la Figura 16.

**Figura 16.**

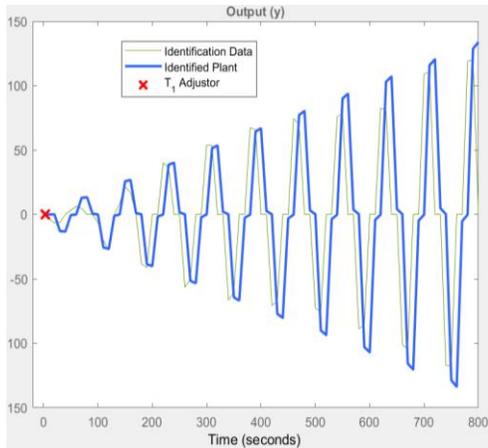
*Gráficas para el modelamiento del controlador.*



c) Tren de impulsos para modelamiento servo visual en el eje X y eje Y



d) Estimación de movimiento obtenido a partir del tren de impulsos en el eje X.



- e) Estimación de movimiento obtenido a partir del tren de impulsos en el eje Y.

Con los datos de la estimación de movimiento a lo largo del eje X se procede a realizar la identificación de la planta utilizando el Toolkit de Matlab systemIdentification, se aproxima a una planta de primer orden con función de transferencia:

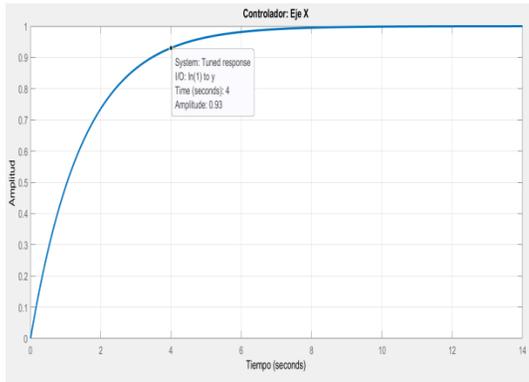
$$G(s) = \frac{K_p}{1 + T_p \cdot s} \quad (4)$$

Obteniendo los parámetros  $K_p = 115.46$  y  $T_p = 1.55$  para Pitch, se procede a sintonizar el PID con la ayuda del Toolkit de Matlab pidTuner con  $M_p = 0\%$  ,  $t_s = 4$ .

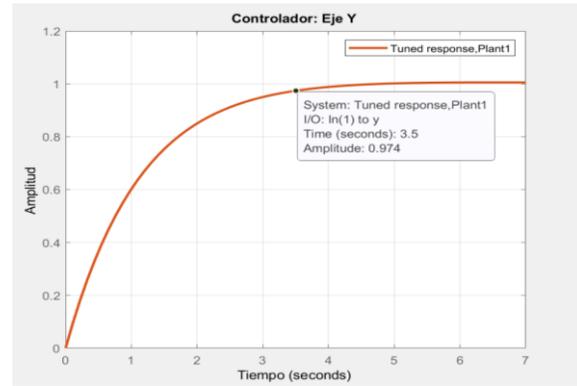
En el caso de Roll se obtuvieron  $K_p = 90.59$  y  $T_p = 1.84$ , el cual se procede de igual forma a sintonizar con ayuda de Matlab con los parámetros  $M_p = 0\%$  ,  $t_s = 3.5$ .

**Figura 17.**

*Controladores obtenidos para el control en Roll y Pitch.*



a) Controlador para el movimiento en el eje X



b) Estimación de movimiento obtenido a partir del tren de impulsos en el eje X.

## Capítulo V

### Detección de Personas

#### **Introducción**

Los sistemas de detección de objetos tienen múltiples aplicaciones como pueden ser el conteo de personas (Andrea, Byron, Jorge, Inti, & Aguilar, 2018), análisis estadístico (Aguilar W. G., y otros, 2017), seguridad (Pardo, Aguilar, & Toulkeridis, 2017), video vigilancia (Calderón, Aguilar, & Merizalde, 2020), etc. (Vera Meaurio & Martínez Jara, 2015). Existen múltiples trabajos e investigaciones que abordan este tema. Haar Cascade es una opción para poder realizar la detección de personas, es un algoritmo sencillo de comprender e implementar, a pesar de eso Haar Cascade presenta un problema en su parámetro de multi-escala y factor de escala. Si el factor de escala es demasiado bajo se evaluarán muchas capas, esto ayudará a detectar más de una persona en una imagen, pero el proceso de detección será lento e incrementará la cantidad de falsos positivos, de otra forma si el factor de escala es muy grande el algoritmo no será capaz de reconocer personas en pequeños píxeles (Rahmad, et al., 2020).

El algoritmo basado en HOG + SVM para detección de personas se destaca por sus características discriminativas basadas en márgenes, el cual se concentra en el contraste de la silueta de las personas. Diferentes personas pueden tener apariencias diferentes pero su contorno es similar (Pang, Yuan, Li, & Pan, 2011), el algoritmo puede provocar falsos positivos, pero se puede ajustar realizando una calibración del umbral de superposición (Cao, Wu, Yan, & Li, 2011).

Yolo es uno de los algoritmos más veloces en la detección de objetos y genera un excelente equilibrio entre velocidad y precisión, lo cual lo hace idóneo para trabajar si se necesita una detección rápida y en tiempo real (Du, 2018), a pesar de esto, Yolo tiene limitaciones como la detección de objetos pequeños en grupos. La implementación oficial de

Yolo está desarrollada con CUDA, por lo cual es necesario de cómputo mediante una GPU compatible. Es posible implementar el algoritmo solo en una CPU, pero su rendimiento será muy bajo provocando retardos en la imagen de hasta 30s (Huang, Pedoeem, & Chen, 2018).

Al no disponer de una GPU para la implementación y correcto funcionamiento de Yolo, se consideró los recursos disponibles para la elaboración del proyecto, el cual fue el algoritmo HOG + SVM, con el objetivo de evitar retardos en la detección de personas.

### ***Implementación del Algoritmo de Detección de Personas***

El algoritmo para tener un mejor rendimiento y reducir el tiempo de detección necesita que se redimensione la imagen, realizar una reducción permitirá que se evalúen menos ventanas deslizantes y de igual forma mejorará la precisión general del algoritmo.

Para la extracción de características HOG primero hay que obtener los coeficientes diferenciales (Kadota, et al., 2009),  $f_x(x, y)$  y  $f_y(x, y)$  que son obtenidos mediante la ecuación:

$$\begin{cases} f_x(x, y) = f(x + 1, y) - f(x - 1, y), \\ f_y(x, y) = f(x, y + 1) - f(x, y - 1), \end{cases} \quad (5)$$

Donde  $f(x, y)$  es la luminosidad en  $(x, y)$ . Después hay que obtener la magnitud  $m$  y dirección  $\theta$  mediante:

$$m(x, y) = \sqrt{f_x(x, y)^2 + f_y(x, y)^2} \quad (6)$$

$$\theta(x, y) = \tan^{-1} \frac{f_x(x, y)}{f_y(x, y)} \quad (7)$$

Una vez obtenidos los valores de magnitud y dirección se procede a calcular los histogramas de gradientes detallados a continuación:

1. Determinar a qué clase pertenece cada  $\theta(x, y)$ ,
2. Incrementar el valor de la clase determinado por el paso anterior,

3. Repetir las operaciones anteriores para todos los gradientes que pertenecen a la celda.

Finalmente hay que proceder a normalizar el histograma, hay que proceder a crear un histograma grande combinando todos los histogramas generados pertenecientes a un bloque que está conformado por varias celdas (Kadota, et al., 2009), para normalizar se debe aplicar la siguiente ecuación:

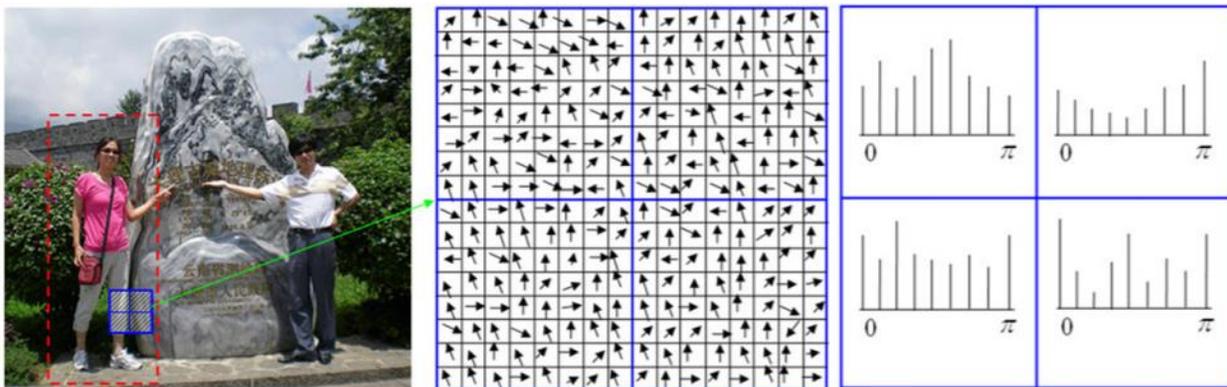
$$v = \frac{V_k}{\sqrt{\|V_k\|^2 + 1}} \quad (8)$$

Donde  $V_k$  es el vector correspondiente a la combinación de los histogramas del bloque,  $v$  es un vector normalizado el cual es una característica HOG.

Una vez definida como se realiza la extracción de características para el clasificador se utiliza la función *cv2.HOGDescriptor()* el cual se encarga de iniciar el descriptor, realiza el proceso detallado en la Figura 18.

### Figura 18.

*Histograma de gradiente de orientado: a) Ventana de detección en una imagen b) Bloque de 16x16 conforman 4 celdas c) Histogramas de gradientes orientados correspondientes a 4 celdas (Pang, Yuan , Li, & Pan, 2011).*



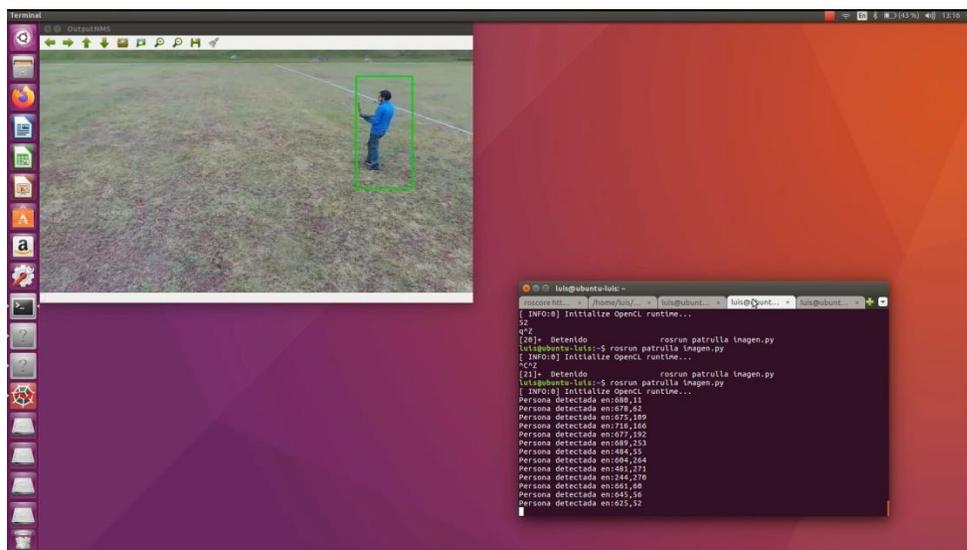
Nota: En la Figura se visualiza el proceso que realiza el extractor de características HOG para generar los histogramas de gradiente orientados.

Llamamos al clasificador el cual es una máquina de soporte vectorial mediante `setSVMDetector` y lo configuramos para que sea un detector de peatones pre entrenado incluyendo la función `cv2.HOGDescriptor getDefaultPeopleDetector()`.

La detección de personas se la realiza mediante el método `detectMultiScale` descriptor HOG entrega dos arreglos, el primero contiene las coordenadas del cuadro delimitador de donde se encuentra cada persona en la imagen y la segunda el valor de confianza devuelto por el SVM para cada una de las detecciones realizadas. En la Figura 19 observamos el algoritmo implementado en la estación en tierra.

## Figura 19

*Detección de personas con HOG +SVM implementado en la Estación en Tierra*



## Capítulo VI

### Pruebas de patrullaje y Resultados

Las pruebas de desempeño del sistema de patrullaje evaluarán la calidad en la navegación en espacios abiertos. Considerando que este proyecto de titulación está enfocado en el patrullaje de polvorines para vigilancia militar, se han establecido 3 circuitos de diferentes características y dimensiones. Al considerar el tipo de entorno en el que se desarrollará el patrullaje y la precisión del GPS incorporado de  $\pm 3$  [m] (Parrot SA, 2019), se establece un error máximo de desviación de trayecto de 1.5 [m], que es el valor experimental de fluctuación en la estabilización del GPS interno del cuadricóptero. De esta manera se analizará la influencia del entorno en el reconocimiento de personas en la misión de vuelo. En la Figura 21 se muestra los circuitos elegidos alrededor de la Universidad de las Fuerzas Armadas ESPE.

#### Figura 20.

*Ubicación de circuitos de prueba en el perímetro de Universidad de las Fuerzas Armadas ESPE*



Nota: Las coordenadas de los circuitos fueron dibujados por medio de ArcGIS.

### ***Circuito cuadrado***

El primer entorno consta de una zona cuadrada alejada de objetos y al aire libre. Este circuito tiene aproximadamente 23[m] de lado y cubre un área de 529 [m<sup>2</sup>]. En la Figura 21 se ilustra la forma del circuito cuadrado con 42 puntos de muestreo.

### **Figura 21.**

*Circuito cuadrado con 42 coordenadas generadas por ArcGis*



Una vez establecido la misión de vuelo del cuadricóptero, se entrega al mismo las coordenadas de los vértices del recorrido y por medio de los algoritmos de navegación desarrollados se procede a hacer el patrullaje. Una vez se inicie la misión de vuelo, el dron de patrullaje toma un total de 42 puntos para ser contrastados.

Según el Sistema Geodésico Mundial (WGS), el CRS (Estándar Común de Comunicación) o sistema de proyección que se utilizará para este proyecto de titulación al estar en Ecuador es el WGS 1984 UTM Zone 17S. El cuál es apropiado para las zonas que abarcan entre 84 ° W y 78 ° W, hemisferio sur entre 80 ° S y el ecuador, en tierra y mar adentro. El error de distancia que existe entre un punto tomado por el GPS de Google Earth y el tomado por el

cuadricóptero se lo calcula transformando las coordenadas en grados a coordenadas en metros. Esto se lo logra con la siguiente relación establecida a nivel mundial.

$$1^\circ = 111,32 \text{ [km]}$$

Teniendo establecido nuestro sistema de coordenadas, Se puede observar los resultados obtenidos en longitud y latitud en las tablas 5 y 6 respectivamente. En los Anexos A y B se presenta la información de las tablas 5 y 6 expresada en metros.

**Tabla 5**

*Datos de longitud tomados por Google Earth y por el cuadricóptero Bebop Parrot 2*

| <b>Coordenadas Longitud<br/>Google Earth</b> | <b>Coordenadas Longitud<br/>Bebop Parrot 2</b> | <b>Proporción del error<br/>longitud con respecto al<br/>error máximo admisible<br/>(1.5[m])</b> |
|--|--|--|
| -78,44581800000                              | -78,44581936270                                | 10,11%   |
| -78,44579836420                              | -78,44580435860                                | 44,49%   |
| -78,44577872840                              | -78,44578757080                                | 65,62%   |
| -78,44575909260                              | -78,44576935680                                | 76,17%   |
| -78,44573945680                              | -78,44574931840                                | 73,19%   |
| -78,44571982100                              | -78,44572921280                                | 69,70%   |
| -78,44570018520                              | -78,44570926270                                | 67,37%   |
| -78,44568054940                              | -78,44568939870                                | 65,67%   |
| -78,44566091350                              | -78,44566969400                                | 65,16%   |
| -78,44564127770                              | -78,44564998360                                | 64,61%   |
| -78,44562164190                              | -78,44563026810                                | 64,02%   |
| -78,44560297490                              | -78,44561060640                                | 56,64%   |
| -78,44560247810                              | -78,44560278310                                | 2,26%  |
| -78,44560198140                              | -78,44560249100                                | 3,78%  |
| -78,44560148470                              | -78,44560177090                                | 2,12%  |
| -78,44560098790                              | -78,44560081640                                | 1,27%  |
| -78,44560049120                              | -78,44560075340                                | 1,95%  |
| -78,44559999450                              | -78,44560375910                                | 27,94%   |
| -78,44559949770                              | -78,44560163330                                | 15,85%   |
| -78,44559900100                              | -78,44559687790                                | 15,76%   |
| -78,44559850430                              | -78,44559911280                                | 4,52%  |
| -78,44559800750                              | -78,44560267190                                | 34,62%   |
| -78,44561736320                              | -78,44562173620                                | 32,45%   |
| -78,44563702520                              | -78,44564157520                                | 33,77%   |
| -78,44565668720                              | -78,44566144750                                | 35,33%   |
| -78,44567634920                              | -78,44568133550                                | 37,00%   |
| -78,44569601120                              | -78,44570136940                                | 39,76%   |

|                 |                 |        |
|-----------------|-----------------|--------|
| -78,44571567320 | -78,44572144600 | 42,84% |
| -78,44573533520 | -78,44574147920 | 45,60% |
| -78,44575499720 | -78,44576148730 | 48,17% |
| -78,44577465920 | -78,44578143830 | 50,31% |
| -78,44579432120 | -78,44580138920 | 52,45% |
| -78,44580651800 | -78,44580709930 | 4,31%  |
| -78,44580779380 | -78,44581012450 | 17,30% |
| -78,44580906950 | -78,44580911140 | 0,31%  |
| -78,44581034530 | -78,44580822570 | 15,73% |
| -78,44581162110 | -78,44581206030 | 3,26%  |
| -78,44581289690 | -78,44581462640 | 12,84% |
| -78,44581417270 | -78,44581542850 | 9,32%  |
| -78,44581544840 | -78,44581548670 | 0,28%  |
| -78,44581672420 | -78,44581507470 | 12,24% |
| -78,44581800000 | -78,44581910720 | 8,22%  |

Nota: El error porcentual presentado muestra la relación de distancia que existe entre el punto original en Longitud tomado por Google Earth y el tomado por el cuadricóptero Bebop Parrot 2 respecto a 1.5 [m] que se estableció como error máximo admisible en el circuito cuadrado.

**Tabla 6**

*Datos de latitud tomados por Google Earth y por el cuadricóptero Bebop Parrot 2*

| <b>Coordenadas Latitud Google Earth</b> | <b>Coordenadas Latitud Bebop Parrot 2</b> | <b>Proporción del error latitud con respecto al error máximo admisible (1.5[m])</b> |
|---|---|---|
| -0,31759660000                          | -0,31759356074                            | 22,56%  |
| -0,31759556798                          | -0,31758582116                            | 72,33%  |
| -0,31759453596                          | -0,31759648588                            | 14,47%  |
| -0,31759350394                          | -0,31760509104                            | 85,99%  |
| -0,31759247191                          | -0,31760496930                            | 92,75%  |
| -0,31759143989                          | -0,31760402155                            | 93,37%  |
| -0,31759040787                          | -0,31760122961                            | 80,31%  |
| -0,31758937585                          | -0,31759794852                            | 63,62%  |
| -0,31758834383                          | -0,31759376223                            | 40,21%  |
| -0,31758731181                          | -0,31758960269                            | 17,00%  |
| -0,31758627979                          | -0,31758546740                            | 6,03%   |
| -0,31758629492                          | -0,31758733846                            | 7,74%   |
| -0,31760595156                          | -0,31760237363                            | 26,55%  |
| -0,31762560819                          | -0,31762251462                            | 22,96%  |
| -0,31764526482                          | -0,31764264625                            | 19,43%  |
| -0,31766492146                          | -0,31766276772                            | 15,98%  |
| -0,31768457809                          | -0,31768283863                            | 12,91%  |

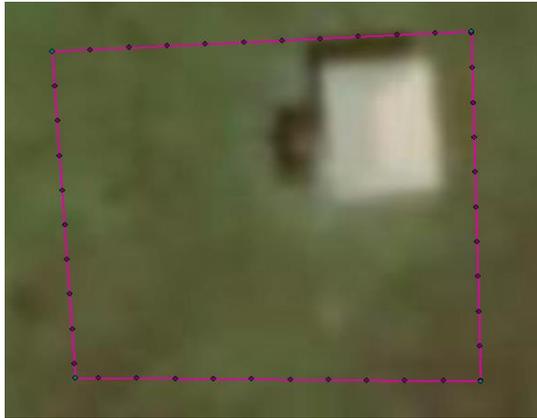
|                |                |        |
|----------------|----------------|--------|
| -0,31770423472 | -0,31770275764 | 10,96% |
| -0,31772389135 | -0,31772244930 | 10,70% |
| -0,31774354799 | -0,31774202445 | 11,31% |
| -0,31776320462 | -0,31776183794 | 10,14% |
| -0,31778286125 | -0,31778166364 | 8,89%  |
| -0,31778297382 | -0,31778510341 | 15,80% |
| -0,31778278476 | -0,31778160805 | 8,73%  |
| -0,31778259570 | -0,31778267278 | 0,57%  |
| -0,31778240664 | -0,31778587722 | 25,76% |
| -0,31778221759 | -0,31778532270 | 23,04% |
| -0,31778202853 | -0,31778366968 | 12,18% |
| -0,31778183947 | -0,31778156627 | 2,03%  |
| -0,31778165041 | -0,31778037391 | 9,47%  |
| -0,31778146135 | -0,31778315963 | 12,60% |
| -0,31778127230 | -0,31778594536 | 34,68% |
| -0,31777319329 | -0,31777156006 | 12,12% |
| -0,31775357181 | -0,31775164401 | 14,31% |
| -0,31773395034 | -0,31773160923 | 17,37% |
| -0,31771432886 | -0,31771159788 | 20,27% |
| -0,31769470738 | -0,31769182171 | 21,42% |
| -0,31767508591 | -0,31767189814 | 23,66% |
| -0,31765546443 | -0,31765176961 | 27,42% |
| -0,31763584295 | -0,31763163388 | 31,24% |
| -0,31761622148 | -0,31761149359 | 35,09% |
| -0,31759660000 | -0,31759193063 | 34,65% |

Nota: El error porcentual presentado muestra la relación de distancia que existe entre el punto original en Latitud tomado por Google Earth y el tomado por el cuadricóptero Bebop Parrot 2 respecto a 1.5 [m] que se estableció como error máximo admisible en el circuito cuadrado.

Al graficar las coordenadas obtenidas y la trayectoria del drone se obtienen los siguientes resultados. En la Figura 22 se ilustran las coordenadas generadas por ambos GPS, Google Earth y Bebop Parrot 2 junto a la dirección de la misión de vuelo realizado por el cuadricóptero.

**Figura 22.**

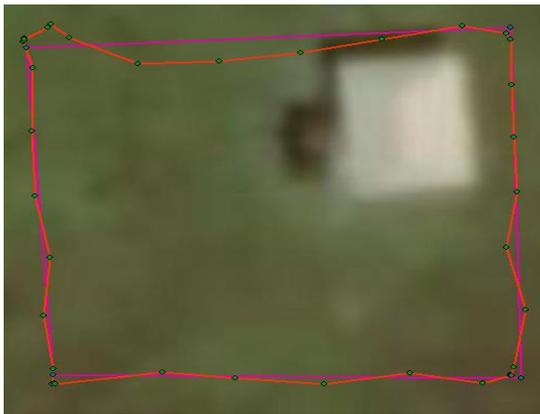
*Gráfica de coordenadas obtenidas en el circuito cuadrado.*



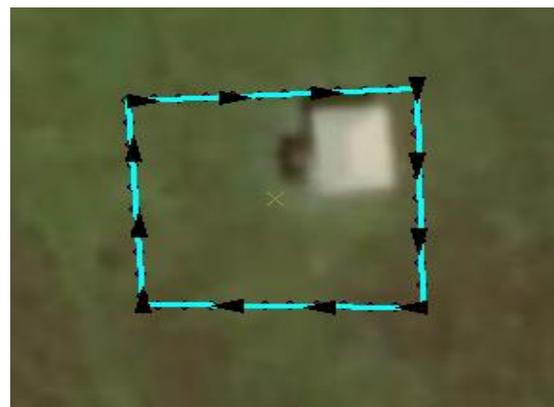
a) Coordenadas obtidas por GPS  
Google Earth



b) Coordenadas obtidas por GPS  
Bebop Parrot 2



c) Contraste de coordenadas en  
misión de vuelo



d) Dirección de movimiento de  
cuadricóptero

Una vez analizado gráficamente los resultados, se establece el error de distancia que existe entre la coordenada establecida por el GPS de Google Earth y la del quadricóptero Bebop Parrot 2. En la tabla 7 se especifica la proporción de error que existe entre cada una de las muestras tomadas anteriormente respecto al error máximo admisible en el circuito cuadrado. En el Anexo C se presenta la información de la tabla 7 expresada en metros.

Tabla 7

*Tabla de errores en circuito cuadrado.*

| <b>Numero<br/>asignado a<br/>dato<br/>coordinado</b> | <b>Proporción del error total con respecto al<br/>error máximo admisible (1.5[m])</b> |
|--|---|
| 1  | 24,72%  |
| 2  | 84,92%  |
| 3  | 67,20%  |
| 4  | 114,88%   |
| 5  | 118,15%   |
| 6  | 116,52%   |
| 7  | 104,83%   |
| 8  | 91,44%  |
| 9  | 76,57%  |
| 10   | 66,81%  |
| 11   | 64,30%  |
| 12   | 57,16%  |
| 13   | 26,65%  |
| 14   | 23,27%  |
| 15   | 19,55%  |
| 16   | 16,03%  |
| 17   | 13,05%  |
| 18   | 30,01%  |
| 19   | 19,12%  |
| 20   | 19,39%  |
| 21   | 11,10%  |
| 22   | 35,74%  |
| 23   | 36,10%  |
| 24   | 34,88%  |
| 25   | 35,33%  |
| 26   | 45,09%  |
| 27   | 45,96%  |
| 28   | 44,54%  |
| 29   | 45,64%  |
| 30   | 49,09%  |
| 31   | 51,86%  |
| 32   | 62,88%  |
| 33   | 12,87%  |
| 34   | 22,45%  |
| 35   | 17,38%  |
| 36   | 25,66%  |
| 37   | 21,66%  |
| 38   | 26,92%  |
| 39   | 28,96%  |
| 40   | 31,24%  |

|    |        |
|----|--------|
| 41 | 37,16% |
| 42 | 35,61% |

Nota: El error porcentual presentado muestra la relación de distancia que existe entre el punto original tomado por Google Earth y el tomado por el cuadricóptero Bebop Parrot 2 respecto a 1.5 [m] que se estableció como error máximo admisible en el circuito cuadrado.

Calculados todos los errores de las muestras tomadas, se establece la siguiente tabla de errores globales. La tabla 8 muestra los errores existentes en el posicionamiento de la misión de vuelo.

**Tabla 8**

*Tabla de errores de la misión de vuelo.*

| <b>Errores globales estadísticos</b>      |   |  |                                       |   |
|---|---|--|---------------------------------------|---|
| <b>Variación máxima entre coordenadas</b> | <b>Variación mínima entre coordenadas</b> | <b>Error cuadrático medio Longitud</b> | <b>Error cuadrático medio Latitud</b> | <b>Porcentaje de puntos con un error menor o igual a valor admisible (1.5[m])</b> |
| 118,15%                                   | 11,10%                                    | 0,375718464                            | 0,302424268                           | 92,85 %   |
| <b>Errores globales de trayectoria</b>    |   |  |                                       |   |
| <b>Tramo de análisis</b>                  | <b>Error medio [m]</b>                    | <b>Distancia recorrida [m]</b>         | <b>Error relativo</b>                 | <b>Error relativo acumulado</b>   |
| 1   | 1,2686                                    | 23,15                                  | 5,47%                                 | 11,81%  |
| 2   | 0,3530                                    | 21,93                                  | 1,60%                                 |   |
| 3   | 0,6642                                    | 23,15                                  | 2,87%                                 |   |
| 4   | 0,3898                                    | 20,81                                  | 1,87%                                 |   |

Como se puede observar en la tabla 8, la variación máxima entre coordenadas respecto al error máximo admisible de 1.5 [m] es de 118,15% el cual excede el límite establecido de 1.5 [m] de desviación en su trayectoria, este exceso representa una desviación de 0,27[m].

También se puede apreciar que el error cuadrático medio tanto de longitud como de latitud son bajos, lo que implica que no existen picos grandes o dispersos de errores. Por último, al establecer un error admisible de 1.5 [m] podemos observar que un 92,85% de los puntos se

encuentran dentro de este límite establecido. Dentro de los errores globales de distancia se realizó un análisis por tramos en el cual se obtuvo el error medio que corresponde a la desviación de cada tramo, tomando en cuenta la distancia recorrida obtendremos un error relativo que representa la desviación en su trayectoria. El máximo error medio generado es de 1,2686[m] el cual representa una desviación del 5,47% de su trayectoria, esta desviación es poco significativa para la trayectoria lo cual demuestra que existe un buen control para la navegación gruesa del circuito y el controlador implementado funciona adecuadamente.

### ***Circuito en forma de rombo***

El segundo entorno consta de una zona en forma de rombo, cerca de algunos objetos como árboles y objetos metálicos. Este circuito tiene aproximadamente 33[m] de diagonal superior y 25[m] de diagonal inferior, cubriendo un área de 412 [m<sup>2</sup>]. En la Figura 23 se ilustra la forma del segundo circuito con 42 puntos de muestreo.

### **Figura 23.**

*Circuito en forma de rombo con 42 coordenadas generadas por ArcGis*



En este circuito se planteó evaluar el rendimiento de navegación del cuadricóptero Bebob Parrot 2 en curvas más cerradas y en presencia de objetos cercanos que podrían repercutir en falta de señal para la comunicación con el UAV.

Tomando en cuenta estos parámetros, en las tablas 9 y 10 se reflejan los resultados obtenidos en longitud y latitud respectivamente. En los Anexos D y E se presenta la información de las tablas 9 y 10 expresada en metros.

**Tabla 9**

*Datos de longitud tomados por Google Earth y por el cuadricóptero Bebop Parrot 2*

| <b>Coordenadas Longitud<br/>Google Earth</b> | <b>Coordenadas Longitud<br/>Bebop Parrot 2</b> | <b>Proporción del error de<br/>longitud con respecto al<br/>error máximo admisible<br/>(1.5[m])</b> |
|--|--|---|
| -78,44504900000                              | -78,44523625580                                | 69,48%  |
| -78,44506821430                              | -78,44521563250                                | 54,70%  |
| -78,44508742850                              | -78,44519370120                                | 39,43%  |
| -78,44510664280                              | -78,44517208380                                | 24,28%  |
| -78,44512585710                              | -78,44515039330                                | 9,10%   |
| -78,44514507130                              | -78,44512850110                                | 6,15%   |
| -78,44516428560                              | -78,44510607610                                | 21,60%  |
| -78,44518349990                              | -78,44508636020                                | 36,05%  |
| -78,44520271410                              | -78,44506529320                                | 50,99%  |
| -78,44522192840                              | -78,44504230160                                | 66,65%  |
| -78,44523027560                              | -78,44502020350                                | 77,95%  |
| -78,44520888770                              | -78,44499819020                                | 78,18%  |
| -78,44518749990                              | -78,44497861080                                | 77,51%  |
| -78,44516611200                              | -78,44496224020                                | 75,65%  |
| -78,44514472410                              | -78,44494871180                                | 72,73%  |
| -78,44512333630                              | -78,44493088730                                | 71,41%  |
| -78,44510194840                              | -78,44491288110                                | 70,16%  |
| -78,44508056060                              | -78,44489605900                                | 68,46%  |
| -78,44505917270                              | -78,44487929820                                | 66,75%  |
| -78,44503778490                              | -78,44486208220                                | 65,20%  |
| -78,44501639700                              | -78,44484187820                                | 64,76%  |
| -78,44499661880                              | -78,44482383910                                | 64,11%  |
| -78,44497953940                              | -78,44483940970                                | 52,00%  |
| -78,44496245990                              | -78,44486203400                                | 37,26%  |
| -78,44494538050                              | -78,44488505010                                | 22,39%  |
| -78,44492830100                              | -78,44490755380                                | 7,70%   |
| -78,44491122160                              | -78,44492914370                                | 6,65%   |
| -78,44489414210                              | -78,44495201720                                | 21,48%  |
| -78,44487706270                              | -78,44497489650                                | 36,30%  |
| -78,44485998320                              | -78,44499724400                                | 50,93%  |
| -78,44484290380                              | -78,44501953400                                | 65,54%  |
| -78,44482622920                              | -78,44504033260                                | 79,45%  |
| -78,44484850620                              | -78,44505638810                                | 77,14%  |
| -78,44487078330                              | -78,44507846510                                | 77,06%  |

|                 |                 |        |
|-----------------|-----------------|--------|
| -78,44489306040 | -78,44510027870 | 76,89% |
| -78,44491533750 | -78,44511794490 | 75,18% |
| -78,44493761460 | -78,44513728030 | 74,09% |
| -78,44495989170 | -78,44515780680 | 73,44% |
| -78,44498216870 | -78,44517740390 | 72,45% |
| -78,44500444580 | -78,44519613570 | 71,13% |
| -78,44502672290 | -78,44521375160 | 69,40% |
| -78,44504900000 | -78,44523530000 | 69,13% |

Nota: El error porcentual presentado muestra la relación de distancia que existe entre el punto original en Longitud tomado por Google Earth y el tomado por el cuadricóptero Bebop Parrot 2 respecto a 1.5 [m] que se estableció como error máximo admisible en el circuito con forma de rombo.

**Tabla 10**

*Datos de latitud tomados por Google Earth y por el cuadricóptero Bebop Parrot 2*

| <b>Coordenadas Latitud<br/>Google Earth</b> | <b>Coordenadas Latitud<br/>Bebop Parrot 2</b> | <b>Proporción del error de<br/>latitud con respecto al<br/>error máximo admisible<br/>(1.5[m])</b> |
|---|---|--|
| -0,31596900000                              | -0,31584962080                                | 44,30%   |
| -0,31595666999                              | -0,31584128246                                | 42,82%   |
| -0,31594433998                              | -0,31583365698                                | 41,07%   |
| -0,31593200997                              | -0,31582515153                                | 39,65%   |
| -0,31591967996                              | -0,31581683463                                | 38,16%   |
| -0,31590734995                              | -0,31580913204                                | 36,45%   |
| -0,31589501994                              | -0,31580306814                                | 34,12%   |
| -0,31588268993                              | -0,31579079812                                | 34,10%   |
| -0,31587035992                              | -0,31578222450                                | 32,70%   |
| -0,31585802991                              | -0,31577890175                                | 29,36%   |
| -0,31584686256                              | -0,31577201263                                | 27,77%   |
| -0,31583887653                              | -0,31576604924                                | 27,02%   |
| -0,31583089051                              | -0,3157750285                                 | 19,81%   |
| -0,31582290448                              | -0,31579369040                                | 10,84%   |
| -0,31581491846                              | -0,31581257520                                | 0,87%  |
| -0,31580693243                              | -0,31582726818                                | 7,55%  |
| -0,31579894641                              | -0,31584189357                                | 15,94%   |
| -0,31579096039                              | -0,31585791462                                | 24,84%   |
| -0,31578297436                              | -0,31587399960                                | 33,78%   |
| -0,31577498834                              | -0,31588946791                                | 42,48%   |
| -0,31576700231                              | -0,31590093325                                | 49,70%   |

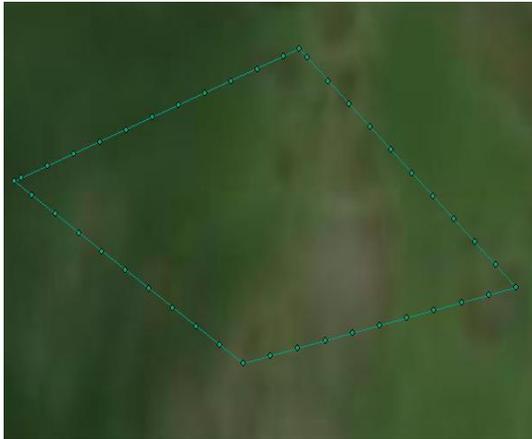
|                |                |        |
|----------------|----------------|--------|
| -0,31576766014 | -0,31591513174 | 54,72% |
| -0,31578280971 | -0,31592454880 | 52,59% |
| -0,31579795928 | -0,31592981300 | 48,93% |
| -0,31581310885 | -0,31593296182 | 44,47% |
| -0,31582825842 | -0,31593806019 | 40,74% |
| -0,31584340799 | -0,31594663520 | 38,30% |
| -0,31585855756 | -0,31595016698 | 33,99% |
| -0,31587370713 | -0,31595386666 | 29,74% |
| -0,31588885670 | -0,31596021052 | 26,48% |
| -0,31590400627 | -0,31596675217 | 23,28% |
| -0,31591905138 | -0,31596781577 | 18,09% |
| -0,31592404624 | -0,31595102664 | 10,01% |
| -0,31592904110 | -0,31594566939 | 6,17%  |
| -0,31593403597 | -0,31593824284 | 1,56%  |
| -0,31593903083 | -0,31592315780 | 5,89%  |
| -0,31594402569 | -0,31591053068 | 12,43% |
| -0,31594902055 | -0,31589965509 | 18,32% |
| -0,31595401541 | -0,31588718074 | 24,80% |
| -0,31595901028 | -0,31587354053 | 31,71% |
| -0,31596400514 | -0,31585839672 | 39,19% |
| -0,31596900000 | -0,31585014655 | 44,10% |

Nota: El error porcentual presentado muestra la relación de distancia que existe entre el punto original en Latitud tomado por Google Earth y el tomado por el cuadricóptero Bebop Parrot 2 respecto a 1.5 [m] que se estableció como error máximo admisible en el circuito con forma de rombo.

Al graficar las coordenadas obtenidas y la trayectoria del dron se obtienen los siguientes resultados. En la Figura 24 se ilustran las coordenadas generadas por ambos GPS, Google Earth y Bebop Parrot 2 junto a la dirección de la misión de vuelo realizado por el cuadricóptero.

**Figura 24.**

*Grafica de coordenadas obtenidas en circuito en forma de rombo.*



a) Coordenadas obtenidas por GPS  
Google Earth



b) Coordenadas obtenidas por GPS  
Bebop Parrot 2



c) Contraste de coordenadas en  
misión de vuelo



d) Dirección de movimiento de  
cuadróptero

Una vez analizado gráficamente los resultados, se establece el error de distancia que existe entre la coordenada establecida por el GPS de Google Earth y la del cuadróptero Bebop Parrot 2. En la tabla 11 se especifica la proporción de error que existe entre cada una de las muestras tomadas anteriormente respecto al error máximo admisible en el circuito con forma de rombo. En el Anexo F se presenta la información de la tabla 11 expresada en metros.

Tabla 11

*Tabla de errores en circuito en forma de rombo.*

| <b>Numero<br/>asignado a<br/>dato<br/>coordinado</b> | <b>Proporción del error total con respecto al<br/>error máximo admisible (1.5[m])</b> |
|--|---|
| 1  | 82,40%  |
| 2  | 69,47%  |
| 3  | 56,94%  |
| 4  | 46,50%  |
| 5  | 39,23%  |
| 6  | 36,96%  |
| 7  | 40,38%  |
| 8  | 49,62%  |
| 9  | 60,58%  |
| 10   | 72,83%  |
| 11   | 82,75%  |
| 12   | 82,72%  |
| 13   | 80,00%  |
| 14   | 76,42%  |
| 15   | 72,74%  |
| 16   | 71,81%  |
| 17   | 71,94%  |
| 18   | 72,83%  |
| 19   | 74,81%  |
| 20   | 77,82%  |
| 21   | 81,63%  |
| 22   | 84,29%  |
| 23   | 73,96%  |
| 24   | 61,50%  |
| 25   | 49,79%  |
| 26   | 41,46%  |
| 27   | 38,88%  |
| 28   | 40,21%  |
| 29   | 46,93%  |
| 30   | 57,40%  |
| 31   | 69,55%  |
| 32   | 81,48%  |
| 33   | 77,79%  |
| 34   | 77,31%  |
| 35   | 76,91%  |
| 36   | 75,41%  |
| 37   | 75,12%  |
| 38   | 75,69%  |
| 39   | 76,57%  |
| 40   | 77,88%  |

|    |        |
|----|--------|
| 41 | 79,70% |
| 42 | 82,00% |

Nota: El error porcentual presentado muestra la relación de distancia que existe entre el punto original tomado por Google Earth y el tomado por el cuadricóptero Bebop Parrot 2 respecto a 1.5 [m] que se estableció como error máximo admisible en el circuito con forma de rombo.

Calculados todos los errores de las muestras tomadas, se establece la siguiente tabla de errores globales. La tabla 12 muestra los errores existentes en el posicionamiento de la misión de vuelo.

**Tabla 12**

*Tabla de errores de la misión de vuelo.*

| <b>Errores globales estadísticos</b>      |   |  |                                       |   |
|---|---|--|---------------------------------------|---|
| <b>Variación máxima entre coordenadas</b> | <b>Variación mínima entre coordenadas</b> | <b>Error cuadrático medio Longitud</b> | <b>Error cuadrático medio Latitud</b> | <b>Porcentaje de puntos con un error menor o igual a valor admisible (1.5[m])</b> |
| 84,29%                                    | 36,96%                                    | 0,843695026                            | 0,247450789                           | 100 %   |
| <b>Errores globales de trayectoria</b>    |   |  |                                       |   |
| <b>Tramo de análisis</b>                  | <b>Error medio [m]</b>                    | <b>Distancia recorrida [m]</b>         | <b>Error relativo</b>                 | <b>Error relativo acumulado</b>   |
| 1   | 0,8695                                    | 27,71                                  | 3,13%                                 | 15,45%  |
| 2   | 1,1440                                    | 26,16                                  | 4,37%                                 |   |
| 3   | 0,8459                                    | 25,82                                  | 3,27%                                 |   |
| 4   | 1,1670                                    | 24,92                                  | 4,68%                                 |   |

Como se puede observar en la tabla 12, la variación máxima entre coordenadas respecto al error máximo admisible de 1.5 [m] es de 84,29% el cual no excede el límite establecido de 1.5 [m] de desviación en su trayectoria. También se puede apreciar que el error cuadrático medio tanto de longitud como de latitud son bajos, lo que implica que no existen picos grandes o dispersos de errores. Por último, al establecer un error admisible de 1.5 [m] podemos observar que ninguno de los datos excedió este límite. Dentro de los errores globales

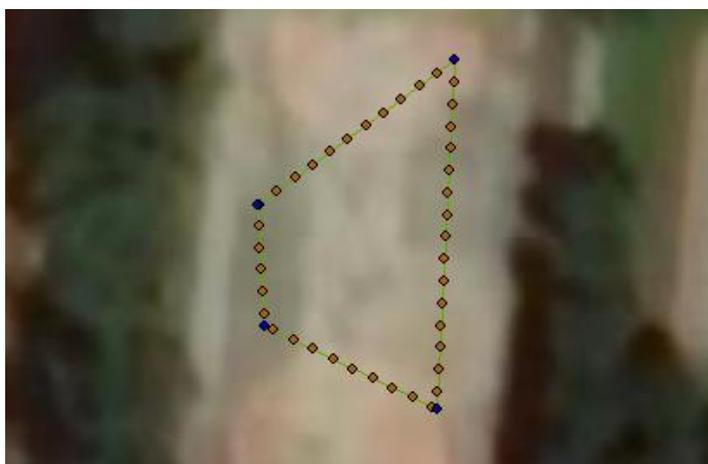
de distancia se realizó un análisis por tramos en el cual se obtuvo el error medio que corresponde a la desviación de cada tramo, tomando en cuenta la distancia recorrida obtendremos un error relativo que representa la desviación en su trayectoria. El máximo error medio generado es de 1,1670[m] el cual representa una desviación del 4,68% de su trayectoria, esta desviación es poco significativa para la trayectoria lo cual demuestra que existe un buen control para la navegación gruesa del circuito y el controlador implementado funciona adecuadamente.

### ***Circuito en forma de trapecio***

El segundo entorno consta de una zona en forma de rombo, cerca de algunos objetos como árboles, objetos metálicos y deferentes formas con colores en el piso. Este circuito tiene aproximadamente 25[m] de base mayor ,10[m] de base menor y 15[m] de altura, cubriendo un área de 262 [m<sup>2</sup>]. En la Figura 25 se ilustra la forma del segundo circuito con 42 puntos de muestreo.

### **Figura 25.**

*Circuito en forma de trapecio con 42 coordenadas generadas por ArcGis*



En este circuito se planteó evaluar el rendimiento de navegación del cuadricóptero Bebop Parrot 2 en curvas más cerradas, presencia de objetos cercanos que podrían repercutir

en falta de señal para la comunicación con el UAV y un ambiente de varias formas y figuras para probar la calidad de reconocimiento de personas y de figuras para el aterrizaje.

Tomando en cuenta estos parámetros, en la tabla 13 y 14 se reflejan los resultados obtenidos en longitud y latitud respectivamente. En los Anexos G y H se presenta la información de las tablas 13 y 14 expresada en metros.

**Tabla 13**

*Datos de longitud tomados por Google Earth y por el cuadricóptero Bebop Parrot 2*

| <b>Coordenadas Longitud<br/>Google Earth</b> | <b>Coordenadas Longitud<br/>Bebop Parrot 2</b> | <b>Proporción del error de<br/>longitud con respecto al<br/>error máximo admisible<br/>(1.5[m])</b> |
|--|--|---|
| -78,44581800000                              | -78,44581936270                                | 21,06%  |
| -78,44579836420                              | -78,44580435860                                | 13,42%  |
| -78,44577872840                              | -78,44578757080                                | 13,28%  |
| -78,44575909260                              | -78,44576935680                                | 11,66%  |
| -78,44573945680                              | -78,44574931840                                | 10,05%  |
| -78,44571982100                              | -78,44572921280                                | 9,61%   |
| -78,44570018520                              | -78,44570926270                                | 8,41%   |
| -78,44568054940                              | -78,44568939870                                | 7,42%   |
| -78,44566091350                              | -78,44566969400                                | 7,38%   |
| -78,44564127770                              | -78,44564998360                                | 7,41%   |
| -78,44562164190                              | -78,44563026810                                | 7,99%   |
| -78,44560297490                              | -78,44561060640                                | 8,23%   |
| -78,44560247810                              | -78,44560278310                                | 8,42%   |
| -78,44560198140                              | -78,44560249100                                | 8,28%   |
| -78,44560148470                              | -78,44560177090                                | 6,80%   |
| -78,44560098790                              | -78,44560081640                                | 5,09%   |
| -78,44560049120                              | -78,44560075340                                | 19,80%  |
| -78,44559999450                              | -78,44560375910                                | 7,30%   |
| -78,44559949770                              | -78,44560163330                                | 30,39%  |
| -78,44559900100                              | -78,44559687790                                | 51,48%  |
| -78,44559850430                              | -78,44559911280                                | 46,08%  |
| -78,44559800750                              | -78,44560267190                                | 40,67%  |
| -78,44561736320                              | -78,44562173620                                | 35,27%  |
| -78,44563702520                              | -78,44564157520                                | 23,31%  |
| -78,44565668720                              | -78,44566144750                                | 10,67%  |
| -78,44567634920                              | -78,44568133550                                | 1,78%   |
| -78,44569601120                              | -78,44570136940                                | 1,79%   |
| -78,44571567320                              | -78,44572144600                                | 0,64%   |
| -78,44573533520                              | -78,44574147920                                | 1,05%   |

|                 |                 |        |
|-----------------|-----------------|--------|
| -78,44575499720 | -78,44576148730 | 0,56%  |
| -78,44577465920 | -78,44578143830 | 0,86%  |
| -78,44579432120 | -78,44580138920 | 3,25%  |
| -78,44580651800 | -78,44580709930 | 7,99%  |
| -78,44580779380 | -78,44581012450 | 13,43% |
| -78,44580906950 | -78,44580911140 | 18,78% |
| -78,44581034530 | -78,44580822570 | 14,99% |
| -78,44581162110 | -78,44581206030 | 7,85%  |
| -78,44581289690 | -78,44581462640 | 0,70%  |
| -78,44581417270 | -78,44581542850 | 1,09%  |
| -78,44581544840 | -78,44581548670 | 2,43%  |
| -78,44581672420 | -78,44581507470 | 2,22%  |
| -78,44581800000 | -78,44581936270 | 21,06% |

Nota: El error porcentual presentado muestra la relación de distancia que existe entre el punto original en Longitud tomado por Google Earth y el tomado por el cuadricóptero Bebop Parrot 2 respecto a 1.5 [m] que se estableció como error máximo admisible en el circuito en forma de trapecio.

**Tabla 14**

*Datos de latitud tomados por Google Earth y por el cuadricóptero Bebop Parrot 2*

| <b>Coordenadas Latitud<br/>Google Earth</b> | <b>Coordenadas Latitud<br/>Bebop Parrot 2</b> | <b>Proporción del error de<br/>latitud con respecto al<br/>error máximo admisible<br/>(1.5[m])</b> |
|---|---|--|
| -0,31759660000                              | -0,31759356074                                | 21,06%   |
| -0,31759556798                              | -0,31758582116                                | 13,42%   |
| -0,31759453596                              | -0,31759648588                                | 13,28%   |
| -0,31759350394                              | -0,31760509104                                | 11,66%   |
| -0,31759247191                              | -0,31760496930                                | 10,05%   |
| -0,31759143989                              | -0,31760402155                                | 9,61%  |
| -0,31759040787                              | -0,31760122961                                | 8,41%  |
| -0,31758937585                              | -0,31759794852                                | 7,42%  |
| -0,31758834383                              | -0,31759376223                                | 7,38%  |
| -0,31758731181                              | -0,31758960269                                | 7,41%  |
| -0,31758627979                              | -0,31758546740                                | 7,99%  |
| -0,31758629492                              | -0,31758733846                                | 8,23%  |
| -0,31760595156                              | -0,31760237363                                | 8,42%  |
| -0,31762560819                              | -0,31762251462                                | 8,28%  |
| -0,31764526482                              | -0,31764264625                                | 6,80%  |
| -0,31766492146                              | -0,31766276772                                | 5,09%  |

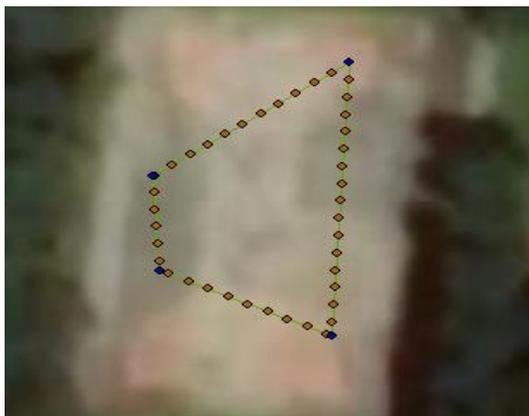
|                |                |        |
|----------------|----------------|--------|
| -0,31768457809 | -0,31768283863 | 19,80% |
| -0,31770423472 | -0,31770275764 | 7,30%  |
| -0,31772389135 | -0,31772244930 | 30,39% |
| -0,31774354799 | -0,31774202445 | 51,48% |
| -0,31776320462 | -0,31776183794 | 46,08% |
| -0,31778286125 | -0,31778166364 | 40,67% |
| -0,31778297382 | -0,31778510341 | 35,27% |
| -0,31778278476 | -0,31778160805 | 23,31% |
| -0,31778259570 | -0,31778267278 | 10,67% |
| -0,31778240664 | -0,31778587722 | 1,78%  |
| -0,31778221759 | -0,31778532270 | 1,79%  |
| -0,31778202853 | -0,31778366968 | 0,64%  |
| -0,31778183947 | -0,31778156627 | 1,05%  |
| -0,31778165041 | -0,31778037391 | 0,56%  |
| -0,31778146135 | -0,31778315963 | 0,86%  |
| -0,31778127230 | -0,31778594536 | 3,25%  |
| -0,31777319329 | -0,31777156006 | 7,99%  |
| -0,31775357181 | -0,31775164401 | 13,43% |
| -0,31773395034 | -0,31773160923 | 18,78% |
| -0,31771432886 | -0,31771159788 | 14,99% |
| -0,31769470738 | -0,31769182171 | 7,85%  |
| -0,31767508591 | -0,31767189814 | 0,70%  |
| -0,31765546443 | -0,31765176961 | 1,09%  |
| -0,31763584295 | -0,31763163388 | 2,43%  |
| -0,31761622148 | -0,31761149359 | 2,22%  |
| -0,31759660000 | -0,31759356074 | 21,06% |

Nota: El error porcentual presentado muestra la relación de distancia que existe entre el punto original en Latitud tomado por Google Earth y el tomado por el cuadricóptero Bebop Parrot 2 respecto a 1.5 [m] que se estableció como error máximo admisible en el circuito en forma de trapecio.

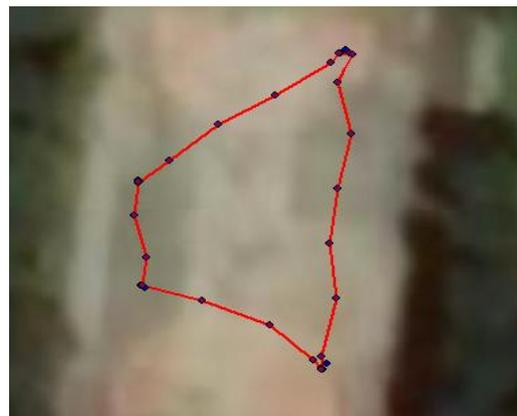
Al graficar las coordenadas obtenidas y la trayectoria del drone se obtienen los siguientes resultados. En la Figura 26 se ilustran las coordenadas generadas por ambos GPS, Google Earth y Bebop Parrot 2 junto a la dirección de la misión de vuelo realizado por el cuadricóptero.

**Figura 26.**

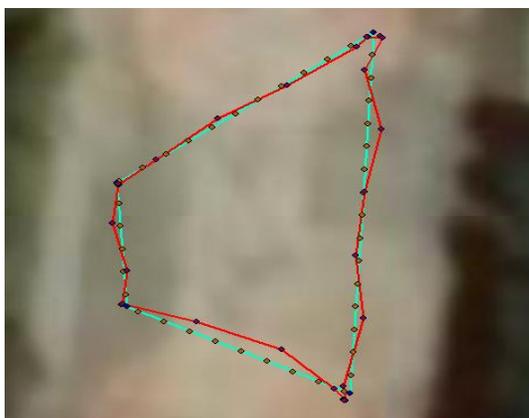
*Grafica de coordenadas obtenidas en circuito en forma de trapecio.*



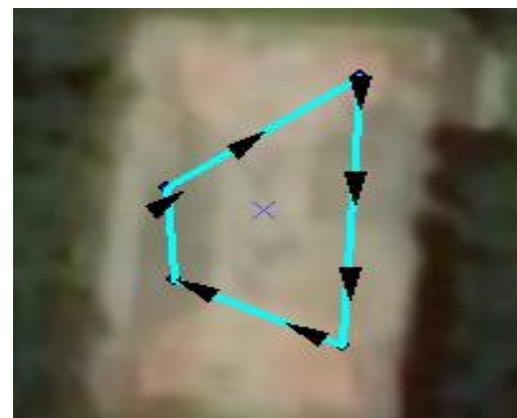
c) Coordenadas obtidas por GPS Google Earth



d) Coordenadas obtidas por GPS Bebop Parrot 2



e) Contraste de coordenadas en misión de vuelo



f) Dirección de movimiento de quadricóptero

Una vez analizado gráficamente los resultados, se establece el error de distancia que existe entre la coordenada establecida por el GPS de Google Earth y la del quadricóptero Bebop Parrot 2. En la tabla 15 se especifica la proporción de error que existe entre cada una de las muestras tomadas anteriormente con respecto al error máximo admisible en el circuito con forma de trapecio. En el Anexo I se presenta la información de la tabla 15 expresada en metros.

Tabla 15

*Tabla de errores en circuito con forma de trapecio.*

| <b>Numero<br/>asignado a<br/>dato<br/>coordinado</b> | <b>Proporción del error total con respecto al<br/>error máximo admisible (1.5[m])</b> |
|--|---|
| 1  | 35,13%  |
| 2  | 14,39%  |
| 3  | 17,40%  |
| 4  | 18,01%  |
| 5  | 39,99%  |
| 6  | 31,39%  |
| 7  | 18,29%  |
| 8  | 8,24%   |
| 9  | 7,95%   |
| 10   | 11,75%  |
| 11   | 11,17%  |
| 12   | 9,64%   |
| 13   | 20,89%  |
| 14   | 16,56%  |
| 15   | 6,81%   |
| 16   | 17,16%  |
| 17   | 23,28%  |
| 18   | 10,18%  |
| 19   | 37,73%  |
| 20   | 62,84%  |
| 21   | 56,69%  |
| 22   | 50,56%  |
| 23   | 44,44%  |
| 24   | 31,84%  |
| 25   | 19,33%  |
| 26   | 7,09%   |
| 27   | 6,45%   |
| 28   | 6,70%   |
| 29   | 21,02%  |
| 30   | 10,46%  |
| 31   | 11,63%  |
| 32   | 9,68%   |
| 33   | 10,06%  |
| 34   | 13,66%  |
| 35   | 18,81%  |
| 36   | 15,17%  |
| 37   | 10,97%  |
| 38   | 13,08%  |
| 39   | 15,04%  |
| 40   | 16,82%  |

|    |        |
|----|--------|
| 41 | 17,31% |
| 42 | 35,13% |

Nota: El error porcentual presentado muestra la relación de distancia que existe entre el punto original tomado por Google Earth y el tomado por el cuadricóptero Bebop Parrot 2 respecto a 1.5 [m] que se estableció como error máximo admisible en el circuito con forma de trapecio.

Calculados todos los errores de las muestras tomadas, se establece la siguiente tabla de errores globales. La tabla 16 muestra los errores existentes en el posicionamiento de la misión de vuelo.

**Tabla 16**

*Tabla de errores de la misión de vuelo.*

| <b>Errores globales estadísticos</b>      |   |  |                                       |   |
|---|---|--|---------------------------------------|---|
| <b>Variación máxima entre coordenadas</b> | <b>Variación mínima entre coordenadas</b> | <b>Error cuadrático medio Longitud</b> | <b>Error cuadrático medio Latitud</b> | <b>Porcentaje de puntos con un error menor o igual a valor admisible (1.5[m])</b> |
| 62,84%                                    | 6,45%                                     | 0,267444671                            | 0,264667002                           | 100 %   |
| <b>Errores globales de trayectoria</b>    |   |  |                                       |   |
| <b>Tramo de análisis</b>                  | <b>Error medio [m]</b>                    | <b>Distancia recorrida [m]</b>         | <b>Error relativo</b>                 | <b>Error relativo acumulado</b>   |
| 1   | 0,2669                                    | 25,02                                  | 1,06%                                 | 9,14%   |
| 2   | 0,5614                                    | 10,79                                  | 5,20%                                 |   |
| 3   | 0,1584                                    | 9,90                                   | 1,60%                                 |   |
| 4   | 0,2396                                    | 18,70                                  | 1.28%                                 |   |

Como se puede observar en la tabla 16, la variación máxima entre coordenadas respecto al error máximo admisible de 1.5 [m] es de 62,84% el cual no excede el límite establecido de 1.5 [m] de desviación en su trayectoria. También se puede apreciar que el error cuadrático medio tanto de longitud como de latitud son bajos, lo que implica que no existen picos grandes o dispersos de errores. Por último, al establecer un error admisible de 1.5 [m] podemos observar que ninguno de los datos excedió este límite. Dentro de los errores globales

de distancia se realizó un análisis por tramos en el cual se obtuvo el error medio que corresponde a la desviación de cada tramo, tomando en cuenta la distancia recorrida obtendremos un error relativo que representa la desviación en su trayectoria. El máximo error medio generado es de 0,5614[m] el cual representa una desviación del 5,20% de su trayectoria, esta desviación es poco significativa para la trayectoria lo cual demuestra que existe un buen control para la navegación gruesa del circuito y el controlador implementado funciona adecuadamente.

### ***Pruebas de reconocimiento de personas***

Las pruebas de desempeño del sistema de reconocimiento de personas evaluarán la calidad del algoritmo seleccionado para dicho propósito. Se procederá a verificar el desempeño los algoritmos en tres diferentes videos con una duración de 30s, para lo cual se considera las decisiones que tomo el algoritmo y estará clasificado en alguna de las opciones listadas a continuación:

- Verdadero Positivo: El algoritmo lo reconoce como una persona y si es una persona.
- Falso positivo: El objeto detectado no es una persona, pero el algoritmo lo detecta como una.
- Falso negativo: La imagen muestra una persona, pero el algoritmo no la puede identificar como tal.

El algoritmo que será utilizado para realizar las pruebas a bordo del sistema de patrullaje será el que presente menor cantidad de falsos positivos y negativos.

### **Pruebas de desempeño Haar Cascade**

En la Tabla 16 se presenta los resultados que se obtuvieron al aplicar el algoritmo de Haar Cascade en imágenes correspondientes a tres diferentes videos, los videos poseen las siguientes características:

- Video 1: Una avenida con flujo de transito medio
- Video 2: La salida de personal de una empresa en un área cerrada
- Video 3: Un área verde similar al área donde se realizará el patrullaje

El algoritmo en la estación en tierra es capaz de procesar la información con un delay de 1.5s entre fotogramas.

**Figura 27.**

*Resumen de resultados de reconocimiento de Haar Cascade*



- a) En la figura se observa que varios peatones no son detectados por el algoritmo, la presencia de falsos positivos y la superposición de recuadros al unir la masa de los objetos.

- b) En la siguiente figura se puede ver que el algoritmo puede generar falsos negativos por factores externos como puede ser brillo e iluminación



- c) En la siguiente figura se puede observar que el algoritmo tiene dificultades para identificar ciertos objetos y los representa como falsos positivos.

Tabla 17

*Resumen de resultados de la implementación del algoritmo Haar Cascade*

|                           | Video | Img 1 | Img 2 | Img 3 | Img 4 | Img 5 | Img 6 | Img 7 | Img 8 | Img 9 | Img 10 |
|---------------------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|--------|
| <b>Verdadero Positivo</b> | 1     | 2     | 1     | 4     | 1     | 3     | 2     | 1     | 2     | 1     | 0      |
| <b>Falso Positivo</b>     |       | 1     | 1     | 2     | 1     | 3     | 1     | 2     | 1     | 3     | 2      |
| <b>Falso Negativo</b>     |       | 4     | 5     | 2     | 3     | 1     | 1     | 2     | 0     | 2     | 3      |
| <b>Verdadero Positivo</b> | 2     | 1     | 1     | 0     | 1     | 1     | 0     | 0     | 1     | 1     | 0      |
| <b>Falso Positivo</b>     |       | 0     | 0     | 1     | 0     | 0     | 1     | 1     | 0     | 0     | 1      |
| <b>Falso Negativo</b>     |       | 1     | 1     | 2     | 0     | 1     | 2     | 1     | 0     | 0     | 0      |
| <b>Verdadero Positivo</b> | 3     | 3     | 1     | 1     | 0     | 0     | 2     | 2     | 2     | 0     | 2      |
| <b>Falso Positivo</b>     |       | 0     | 0     | 0     | 1     | 2     | 1     | 0     | 0     | 1     | 0      |
| <b>Falso Negativo</b>     |       | 3     | 5     | 4     | 5     | 3     | 1     | 1     | 4     | 5     | 2      |

### **Pruebas de desempeño Hog + SVM**

En la Tabla 17 se presenta los resultados que se obtuvieron al aplicar el algoritmo de Hog + SVM en imágenes correspondientes a tres diferentes videos, videos especificados en la sección de pruebas con el algoritmo Haar Cascade. El algoritmo en la estación en tierra es capaz de procesar la información con un delay de 500ms entre fotogramas.

**Figura 28.**

*Resumen de resultados de reconocimiento de Hog + SVM*



a) En la figura se observa que el algoritmo realiza superposición cuando existen varias personas y de igual forma detecta falsos positivos en ambientes con múltiples figuras y objetos.

b) En la siguiente figura se puede visualizar que el algoritmo es capaz de realizar el reconocimiento de personas en espacios reducidos.



c) En la siguiente figura se puede ver que el algoritmo en un ambiente similar al patrullaje no genera falsos positivos, no es capaz de identificar personas que se encuentran muy alejadas.

Tabla 18

*Resumen de resultados de la implementación del algoritmo Hog + SVM*

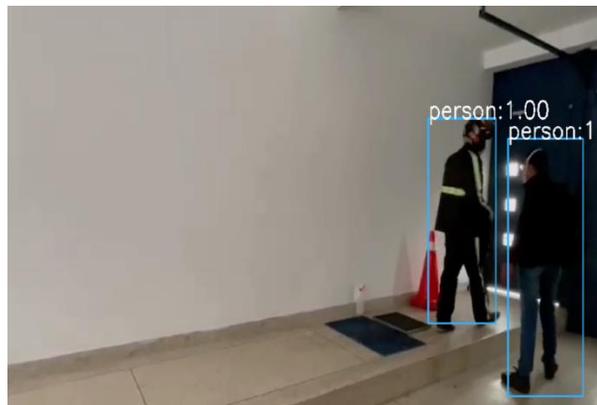
|                           | Video | Img 1 | Img 2 | Img 3 | Img 4 | Img 5 | Img 6 | Img 7 | Img 8 | Img 9 | Img 10 |
|---------------------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|--------|
| <b>Verdadero Positivo</b> |       | 3     | 4     | 4     | 2     | 2     | 2     | 1     | 2     | 2     | 2      |
| <b>Falso Positivo</b>     | 1     | 3     | 1     | 6     | 6     | 4     | 5     | 7     | 7     | 9     | 8      |
| <b>Falso Negativo</b>     |       | 3     | 2     | 2     | 2     | 2     | 1     | 2     | 0     | 1     | 1      |
| <b>Verdadero Positivo</b> |       | 2     | 0     | 2     | 1     | 0     | 1     | 1     | 1     | 0     | 1      |
| <b>Falso Positivo</b>     | 2     | 0     | 2     | 0     | 1     | 1     | 2     | 2     | 0     | 1     | 2      |
| <b>Falso Negativo</b>     |       | 0     | 1     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0      |
| <b>Verdadero Positivo</b> |       | 2     | 1     | 1     | 1     | 2     | 2     | 1     | 1     | 2     | 2      |
| <b>Falso Positivo</b>     | 3     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0      |
| <b>Falso Negativo</b>     |       | 4     | 3     | 3     | 3     | 1     | 1     | 2     | 2     | 3     | 4      |

### **Pruebas de desempeño Yolo**

En la Tabla 18 se presenta los resultados que se obtuvieron al aplicar el algoritmo de Yolo en imágenes correspondientes a tres diferentes videos, videos especificados en la sección de pruebas con el algoritmo Haar Cascade. El algoritmo en la estación en tierra es capaz de procesar la información con un delay de 3s entre fotogramas.

**Figura 29.**

*Resumen de resultados de reconocimiento de Yolo*



- a) En la figura se observa que el algoritmo puede discriminar únicamente a las personas sin importar los objetos y figuras alrededor.

- b) En la figura se visualiza que el algoritmo puede identificar personas con poca iluminación.



- c) En la siguiente figura se puede ver que el algoritmo en un ambiente similar al patrullaje puede identificar personas inclusive a mayores distancias.

Tabla 19

*Resumen de resultados de la implementación del algoritmo Yolo*

|                           | Video | Img 1 | Img 2 | Img 3 | Img 4 | Img 5 | Img 6 | Img 7 | Img 8 | Img 9 | Img 10 |
|---------------------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|--------|
| <b>Verdadero Positivo</b> |       | 4     | 6     | 5     | 4     | 3     | 3     | 3     | 2     | 3     | 3      |
| <b>Falso Positivo</b>     | 1     | 0     | 0     | 2     | 0     | 0     | 0     | 0     | 1     | 0     | 0      |
| <b>Falso Negativo</b>     |       | 2     | 0     | 0     | 0     | 1     | 0     | 0     | 0     | 0     | 0      |
| <b>Verdadero Positivo</b> |       | 2     | 1     | 2     | 1     | 0     | 1     | 1     | 1     | 0     | 1      |
| <b>Falso Positivo</b>     | 2     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0      |
| <b>Falso Negativo</b>     |       | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0      |
| <b>Verdadero Positivo</b> |       | 6     | 6     | 5     | 4     | 2     | 4     | 3     | 3     | 5     | 5      |
| <b>Falso Positivo</b>     | 3     | 0     | 2     | 0     | 0     | 0     | 0     | 1     | 2     | 1     | 1      |
| <b>Falso Negativo</b>     |       | 0     | 0     | 0     | 0     | 1     | 0     | 0     | 0     | 0     | 1      |

### ***Pruebas de desempeño de HOG + SVM en el sistema de patrullaje***

En base a las pruebas realizadas de los diferentes algoritmos implementados se decidió que el algoritmo con mejor desempeño para este proyecto de titulación es el HOG + SVM debido a su bajo costo computacional y su calidad de reconocimiento de personas en ambientes similares al área de patrullaje.

### **Pruebas de desempeño de HOG + SVM en circuito cuadrado**

Para realizar las pruebas en exteriores se aplicó el algoritmo de reconocimiento de personas en los 3 circuitos establecidos anteriormente en las pruebas de navegación. En la Figura 30 se muestra los resultados de reconocimiento en el circuito cuadrado.

**Figura 30.**

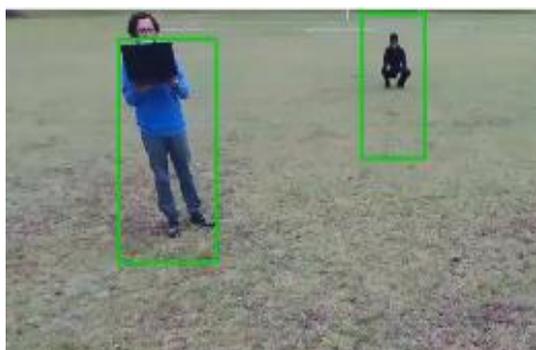
*Resultados de pruebas de reconocimiento de personas en circuito cuadrado*



- a) En la siguiente ilustración se puede observar a un sujeto de prueba con una vestimenta de colores cálidos siendo correctamente reconocido



- b) En la figura se puede observar a un segundo sujeto de prueba con una vestimenta oscura diferente a su entorno y en cuclillas siendo detectado correctamente.



- c) En esta imagen se puede apreciar cómo los 2 sujetos de prueba separados a una distancia aproximada de 4 metros son detectados correctamente.



- d) En esta ilustración se puede ver que el cuadro de detección de personas se sobrepone pero puede detectar a los 2 sujetos de prueba sin problema a una distancia aproximada de 1 metro

Como se puede observar en la figura anterior, no existieron una cantidad considerable de falsos positivos u otros errores del algoritmo en el reconocimiento de personas. En la tabla 20 se presentan los parámetros de las pruebas realizadas y los resultados obtenidos.

**Tabla 20**

*Resumen de resultados de la implementación de detección de personas*

| <b>Datos</b>   | <b>Especificaciones</b> |
|--|-------------------------|
| <b>Circuito</b>                                      | Cuadrado                |
| <b>Tiempo de vuelo</b>                               | 3:50 [min]              |
| <b>Altura de vuelo</b>                               | 3.5 [m]                 |
| <b>Área cubierta</b>                                 | 529 [m <sup>2</sup> ]   |
| <b>Velocidad de drone</b>                            | 7.20 [km/h]             |
| <b>Numero de falsos positivos en misión de vuelo</b> | 2                       |
| <b>Numero de falsos negativos en misión de vuelo</b> | 0                       |

Nota: Los datos obtenidos en las pruebas pudieron ser corroborados en el interfaz de la aplicación *FreeFlight PRO*.

Durante las pruebas, los 2 falsos positivos que se produjeron fueron durante el despegue y el aterrizaje del cuadricóptero, por lo cual pudimos corroborar que durante el patrullaje no existieron errores de reconocimiento y por lo tanto se puede concluir que el algoritmo cumple con los objetivos planteados en este proyecto de titulación.

#### **Pruebas de desempeño de HOG + SVM en circuito en forma de rombo**

De igual manera que el anterior circuito, aquí se realizará pruebas de reconocimiento de personas en el circuito en forma de rombo con un entorno brillante y pocos objetos cercanos. En la Figura 31 se muestra los resultados de reconocimiento en el circuito cuadrado.

**Figura 31.**

*Resultados de pruebas de reconocimiento de personas en circuito en forma de rombo*



- a) En la siguiente ilustración se puede observar a un sujeto de prueba con una vestimenta de colores cálidos siendo correctamente reconocido



- b) En la figura se puede observar a un segundo sujeto de prueba con una vestimenta oscura diferente a su entorno y en cuclillas siendo detectado correctamente.



- c) En esta imagen se puede apreciar cómo los 2 sujetos de prueba separados a una distancia aproximada de 15 metros son detectados, uno correctamente y el otro no, esto se debe a que el sujeto con vestimenta oscura no es identificado al estar muy apartado del rango de visión del UAV.



- d) En esta ilustración se puede ver que el cuadro de detección aparece tanto en el sujeto de prueba como en un poste, este falso positivo se produce por el juego de colores y sombras que existe en el entorno en el que se realiza la detección de personas.

Como se puede observaren la figura anterior, existieron fallas del algoritmo al poner a un sujeto de prueba a una distancia muy lejana o también al poner al UAV en una zona con objetos con tonalidades de colores similares a las de su entorno. En la tabla 21 se presentan los parámetros de las pruebas realizadas y los resultados obtenidos.

**Tabla 21**

*Resumen de resultados de la implementación de detección de personas*

| <b>Datos</b>   | <b>Especificaciones</b> |
|--|-------------------------|
| <b>Circuito</b>                                      | Rombo                   |
| <b>Tiempo de vuelo</b>                               | 3:20 [min]              |
| <b>Altura de vuelo</b>                               | 3.5 [m]                 |
| <b>Área cubierta</b>                                 | 412 [m <sup>2</sup> ]   |
| <b>Velocidad de dron</b>                             | 7.20 [km/h]             |
| <b>Numero de falsos positivos en misión de vuelo</b> | 3                       |
| <b>Numero de falsos negativos en misión de vuelo</b> | 1                       |

Nota: Los datos obtenidos en las pruebas pudieron ser corroborados en el interfaz de la aplicación *FreeFlight PRO*.

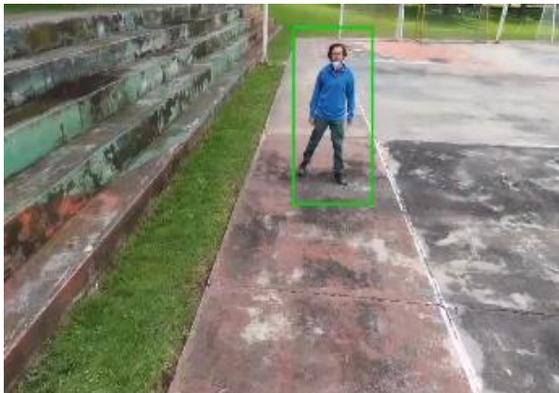
Durante las pruebas, los 3 falsos positivos que se produjeron fueron durante el despegue, el aterrizaje del cuadricóptero y un objeto con colores similares a su entorno, por lo cual pudimos corroborar que, durante el patrullaje, el algoritmo de detección de personas puede cometer errores de reconocimiento al implementarlo en lugares muy brillantes donde los reflejos de luz puedan afectar la imagen. En cuanto al falso negativo presentado en esta prueba, se puede concluir que el reconocimiento de personas se logra sin inconvenientes siempre y cuando el objetivo se encuentre a un rango no mayor a los 20 metros. Esto debido a la resolución de la cámara implementada en este proyecto de titulación.

#### **Pruebas de desempeño de HOG + SVM en circuito en forma de trapecio**

De igual manera que el anterior circuito, aquí se realizará pruebas de reconocimiento de personas en el circuito en forma de trapecio y con un entorno con varios colores y múltiples objetos. En la Figura 32 se muestra los resultados de reconocimiento en el circuito cuadrado.

**Figura 32.**

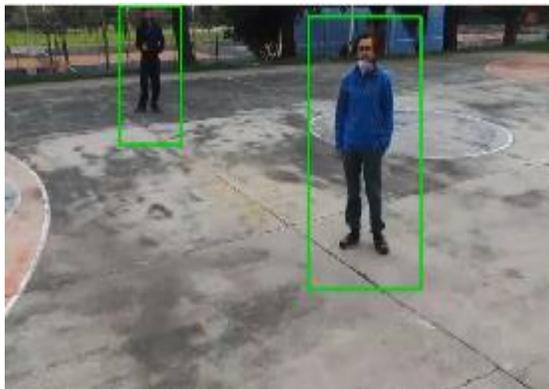
*Resultados de pruebas de reconocimiento de personas en circuito en forma de trapecio*



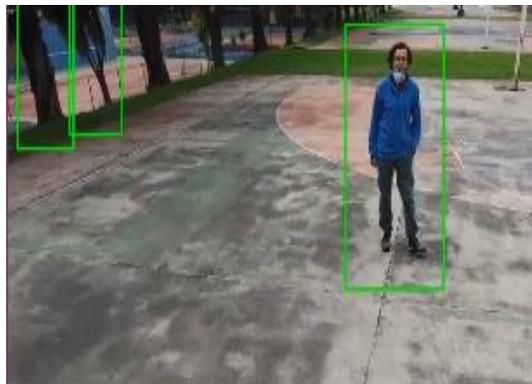
- g) En la siguiente ilustración se puede observar a un sujeto de prueba con una vestimenta de colores cálidos siendo correctamente reconocido



- h) En la figura se puede observar a un segundo sujeto de prueba con una vestimenta oscura con figuras de entorno parecidos siendo detectado correctamente.



- i) En esta imagen se puede apreciar cómo los 2 sujetos de prueba separados a una distancia aproximada de 6 metros y en un entorno con varios colores y formas en su entorno son detectados correctamente.



- j) En esta ilustración se puede ver que el cuadro de detección aparece tanto en el sujeto de prueba como en varios árboles, estos falso positivo se producen por el juego de colores, sombras y formas que existe en el entorno en el que se realiza la detección de personas.

Como se puede observaren la figura anterior, existieron fallas del algoritmo al situar al cuadricóptero en un entorno lleno de figuras con sombra o de siluetas con forma de personas.

Con los ensayos se demostró que estos errores pueden ser corregidos al elevar levemente el UAV, ya que así los objetos que aparentan tener un tronco humano pierden esa forma a lo lejos. Estas correcciones se las debe realizar tomando en cuenta que de igual manera el distanciamiento afecta la calidad de detección de personas. En la tabla 22 se presentan los parámetros de las pruebas realizadas y los resultados obtenidos.

**Tabla 22**

*Resumen de resultados de la implementación de detección de personas*

| <b>Datos</b>   | <b>Especificaciones</b> |
|--|-------------------------|
| <b>Circuito</b>                                      | Trapezio                |
| <b>Tiempo de vuelo</b>                               | 2:50 [min]              |
| <b>Altura de vuelo</b>                               | 2.8 [m]                 |
| <b>Área cubierta</b>                                 | 262 [m <sup>2</sup> ]   |
| <b>Velocidad de dron</b>                             | 7.20 [km/h]             |
| <b>Numero de falsos positivos en misión de vuelo</b> | 4                       |
| <b>Numero de falsos negativos en misión de vuelo</b> | 0                       |

Nota: Los datos obtenidos en las pruebas pudieron ser corroborados en el interfaz de la aplicación *FreeFlight PRO*.

Durante las pruebas, los 4 falsos positivos que se produjeron fueron durante el despegue, el aterrizaje del cuadricóptero y por objetos con siluetas en forma de personas, por lo cual pudimos corroborar que, durante el patrullaje, el algoritmo de detección de personas puede cometer errores de reconocimiento al implementarlo cerca de objetos con estaturas o formas semejantes a las de un ser humano. Como se manifestó anteriormente, estos

problemas se pueden mitigar al modificar la altura de vuelo y el ángulo de la cámara del cuadricóptero de vigilancia.

Con esta recopilación de datos se puede concluir que el sistema de reconocimiento de personas funciona correctamente y puede ser implementado en el sistema de patrullaje tomando en cuenta todas las observaciones antes expuestas.

### Pruebas de controlador de aterrizaje

Las pruebas de desempeño del sistema de aterrizaje autónomo evaluarán la calidad del controlador empleado para detectar la plataforma de aterrizaje y descender en ella. Para este fin, se realizó múltiples aterrizajes en los circuitos antes mencionados. La prueba inicial consta de aterrizar el cuadricóptero realizando solo el último tramo de cada circuito y una vez confirmado que el cuadricóptero aterriza adecuadamente, se realiza una última prueba con un aterrizaje luego de realizar el patrullaje entero. En la tabla 23 se presenta los resultados de aterrizaje en el circuito cuadrado.

**Tabla 23**

*Prueba de aterrizaje del circuito cuadrado.*

| Detección de UAV  | Aterrizaje   | Resultados obtenidos   |
|---|--|--|
|  |  | Porción de plataforma detectada:<br>97%<br>Distancia del centro de la plataforma:<br>33,4 [cm] |

|   |  |   |
|---|--|---|
|  |  | <p>Porción de plataforma detectada:<br/>77%</p> <p>Distancia del centro de la plataforma:<br/>43,2 [cm]</p> |
|  |  | <p>Porción de plataforma detectada:<br/>53%</p> <p>Distancia del centro de la plataforma:<br/>48,2 [cm]</p> |

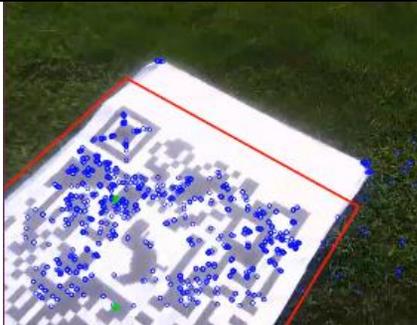
Nota: La información de que cantidad de la plataforma fue detectada es obtenida por el algoritmo implementado y las mediciones del aterrizaje fueron realizadas a mano.

De la misma manera se realizaron pruebas para el circuito en forma de rombo. En la tabla 24 se presentan los resultados obtenidos.

**Tabla 24**

*Prueba de aterrizaje del circuito con forma de rombo.*

| Detección de UAV  | Aterrizaje   | Resultados obtenidos  |
|---|--|---|
|  |  | <p>Porción de plataforma detectada:<br/>88%</p> <p>Distancia del centro de la plataforma:<br/>16,1 [cm]</p> |

|   |  |  |
|---|--|--|
|  |  | Porción de plataforma detectada:<br>74%<br>Distancia del centro de la plataforma:<br>38,6 [cm] |
|  |  | Porción de plataforma detectada:<br>62%<br>Distancia del centro de la plataforma:<br>53,5 [cm] |

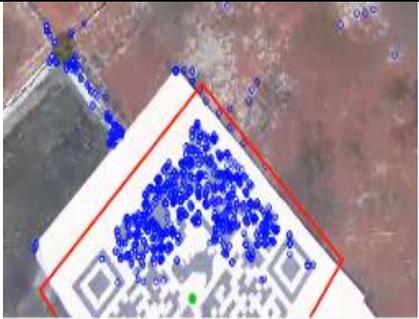
Nota: La información de que cantidad de la plataforma fue detectada es obtenida por el algoritmo implementado y las mediciones del aterrizaje fueron realizadas a mano.

Por último, se realizaron pruebas para el circuito en forma de trapecio. En la tabla 25 se presentan los resultados obtenidos.

**Tabla 25**

*Prueba de aterrizaje del circuito con forma de trapecio.*

| Detección de UAV  | Aterrizaje   | Resultados obtenidos  |
|---|--|---|
|  |  | Porción de plataforma detectada:<br>100%<br>Distancia del centro de la plataforma:<br>34,8 [cm] |

|   |  |  |
|---|--|--|
|  |  | Porción de plataforma detectada:<br>88%<br>Distancia del centro de la plataforma:<br>31,8 [cm] |
|  |  | Porción de plataforma detectada:<br>48%<br>Distancia del centro de la plataforma:<br>69,0 [cm] |

Nota: La información de que cantidad de la plataforma fue detectada es obtenida por el algoritmo implementado y las mediciones del aterrizaje fueron realizadas a mano.

Una vez analizado los datos se calcula los errores del sistema de aterrizaje. En la tabla 26 se exponen los errores globales del sistema de aterrizaje.

**Tabla 26**

*Errores globales del controlador de aterrizaje en circuito los 3 circuitos.*

| <b>Errores globales</b> |                     |                             |
|-------------------------|---------------------|-----------------------------|
| <b>Error máximo</b>     | <b>Error Mínimo</b> | <b>Error medio muestral</b> |
| 69,0 [cm]               | 16,1 [cm]           | 40,9556 [cm]                |

Como se puede ver en la tabla 26, se puede obtener un aterrizaje apropiado con el controlador diseñado en un radio de 40,9556 [cm] al obtener una detección de plataforma mayor al 50%.

## Capítulo VII

### Conclusiones y Recomendaciones

#### **Conclusiones**

En este proyecto de titulación se implementó un sistema de patrullaje de drones para detección de personas en espacios restringidos. Capaz de navegar en un espacio delimitado de forma autónoma, realizar su aterrizaje igualmente sin asistencia sobre una superficie de características establecidas y reconocer personas en el rango de visión del UAV.

En el sistema se implementó un total de 2 controladores. El primero se trata de un controlador ON/OFF encargado de estabilizar la trayectoria del cuadricóptero en caso de que existiera alguna perturbación ambiental que lo desvié de su misión de vuelo. Este al ser implementado presentó pequeños movimientos en zigzag debido a la baja la tasa de refresco del GPS integrado en el Bebop Parrot 2. No se implementó un controlador de mayor complejidad como un control P o PI ya que de igual manera debido a la baja tasa de refresco del GPS integrado entregaría datos fuera de tiempo y provocaría el mismo movimiento en zigzag en el cuadricóptero. Aun con estos leves movimientos, en todas sus misiones de vuelo, el cuadricóptero se mantuvo estable y su error relativo máximo encontrado en los diferentes tramos de sus misiones de vuelo fue de 5,20%, lo cual es aceptable para patrullajes en la intemperie.

El control PID implementado para aterrizaje autónomo requirió de una correcta calibración en el análisis de coincidencias. Si se le exigía al control un alto número de coincidencias para cada corrección de posición el proceso de aterrizaje tomaba más de un minuto en realizarse. Por lo tanto, se optó por un menor número de coincidencias para recortar el tiempo de aterrizaje a menos de un minuto. Por otro lado, el porcentaje mínimo de la plataforma que necesitó ser detectado por el UAV para realizar un aterrizaje exitoso fue del

53%. Lo cual implica que la cámara del cuadricóptero necesita por lo menos la mitad de la imagen para iniciar el proceso de corrección y aterrizaje. De esta forma, al reducirse la cantidad de puntos de interés el error de aterrizaje aumentó hasta alcanzar un pico registrado de 69[cm] de distancia del centro de la plataforma a la posición final del cuadricóptero.

Para el sistema de detección de personas se implementaron diferentes algoritmos de reconocimiento de persona con el fin de encontrar el más factible para este proyecto de titulación. El algoritmo de YOLO entrega excelentes resultados de detección siendo capaz de un reconocimiento de personas a largas, medianas y cortas distancias con una gran precisión. En las pruebas realizadas de este algoritmo, no se encontraron falsos positivos ni errores de detección a menos que se lo expusiera a casos extremos como muchas personas extremadamente juntas o que la detección se realizara a distancias extremadamente cercanas o alejadas del rango de la cámara integrada. Sin embargo, el costo computacional que demanda este algoritmo excede la capacidad de una computadora ordinaria llegando a tener delays de hasta 1 minuto en la estación en tierra. Haar Cascade es una opción sencilla y de bajo coste computacional al no necesitar de una GPU el problema de este tipo de algoritmo es su baja precisión, existen muchos problemas para detectar objetivos a una distancia mayor a 5m además de presentar un problema de parpadeo el cual en ciertos fotogramas detecta a una persona y en otras no, otro inconveniente que presenta este algoritmo son los cuadros delimitadores que no se ajustan apropiadamente. Finalmente se optó por implementar HOG + SVM ya que su costo computacional no es muy exigente siendo capaz de ser implementado en computadoras de gama media sin problemas de delays superiores a 1 segundos. En las pruebas de implementación en las diferentes misiones de vuelo se obtuvo un error máximo de 4 falsos positivos provocados en su mayoría en el despegue y aterrizaje al tener objetos muy cerca de su cámara. Los resultados obtenidos fueron muy prometedores ya que las personas

dentro del rango de visión del UAV fueron correctamente detectadas con la presencia de un solo falso negativo en todas las misiones de vuelo.

### ***Recomendaciones***

Se recomienda implementar este sistema de patrullaje en zonas abiertas sin muchos obstáculos en su misión de vuelo ya que el GPS integrado en cuadricóptero de gama media como es el caso del Bebop Parrot 2, no tienen una alta tasa de refresco de sus coordenadas de posicionamiento por lo cual el control que se realiza en su desplazamiento no es totalmente lineal y puede chocar con objetos que estén cerca de su trayecto.

Es de gran importancia mantener el UAV encendido en la plataforma de despegue por lo menos un minuto antes de iniciar su misión de vuelo. Esto debido a que, en el proceso de pruebas de patrullaje, se pudo observar que el cuadricóptero necesita un tiempo de estabilización de los datos del GPS. De no esperar este tiempo, el GPS del UAV proporciona datos erróneos de su posición haciendo que el control en su patrullaje se viera afectado.

En cuanto al reconocimiento de personas, se recomienda mantener el patrullaje del drone en un rango de 3 a 4 metros de altura ya que resolución de la cámara integrada del Bebop Parrot 2 puede enfocar bien objetos hasta aproximadamente 20 metros de distancia de él, y por ende realizar un reconocimiento correcto de personas.

De igual manera para el sistema de aterrizaje autónomo, se recomienda no elevar el cuadricóptero más de 4 metros ya que en las pruebas se pudo observar que superada esta altura el UAV no podía reconocer de manera precisa la figura plasmada en la plataforma de aterrizaje por lo cual el control implementado en el cuadricóptero no recibe datos de entrada y solo se mantenía suspendido en el aire.

Para el establecimiento de la estación terrestre, es recomendable que el logo impreso en la plataforma de aterrizaje tenga colores con tonalidades oscuras y diferentes al entorno en que se encuentra para que su reconocimiento no se vea afectado por estas características de su entorno. Además, es muy beneficioso que el logo posea puntas o esquinas que resalten en su forma ya que estas proveen de puntos de interés que el algoritmo de reconocimiento puede detectar fácilmente.

## Referencias

- Adarsh, P., Rathi, P., & Kumar, M. (2020). YOLO v3-Tiny: Object Detection and Recognition using one stage improved model. (IEEE, Ed.) *2020 6th International Conference on Advanced Computing and Communication Systems, ICACCS 2020*, 687-694.  
doi:10.1109/ICACCS48705.2020.9074315
- Aguilar, W. G., & Angulo, C. (2014). Real-time video stabilization without phantom movements for micro aerial vehicles. *EURASIP Journal on Image and Video Processing*, 1, 1-13.
- Aguilar, W. G., & Angulo, C. (2016). Real-Time Model-Based Video Stabilization for Microaerial Vehicles. *Neural Processing Letters*, 43(2), 459-477.
- 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., Álvarez, L., Grijalva, S., & Rojas, I. (2019). Monocular Vision-Based Dynamic Moving Obstacles Detection and Avoidance. En *Lecture Notes in Computer Science* (págs. 386-398).
- Aguilar, W. G., Angulo, C., & Costa-Castello, R. (2017). Autonomous Navigation Control for Quadrotors in Trajectories Tracking. En *Lecture Notes in Computer Science* (págs. 287-297).
- Aguilar, W. G., Angulo, C., & Pardo, J. A. (2017). Motion intention optimization for multicopter robust video stabilization. *Chilean Conference on Electrical, Electronics Engineering, Information and Communication Technologies (CHILECON)*. Pucón, Chile.
- Aguilar, W. G., Casaliglla, V. P., & Pólit, J. L. (2017). Obstacle Avoidance Based-Visual Navigation for Micro Aerial Vehicles. *Electronics*, 6(1), 10.

- Aguilar, W. G., Cobeña, B., Rodriguez, G., Salcedo, V. S., & Collaguazo, B. (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.
- Aguilar, W. G., Luna, M. A., Moya, J. F., Abad, V., Parra, H., & Ruiz, H. (2017). Pedestrian detection for UAVs using cascade classifiers with meanshift. *IEEE 11th international conference on semantic computing (ICSC)*.
- Aguilar, W. G., Luna, M. A., Moya, J. F., Abad, V., Ruiz, H., Parra, H., & Angulo, C. (2017). Pedestrian Detection for UAVs Using Cascade Classifiers and Saliency Maps. En *Lecture Notes in Computer Science* (págs. 563-574).
- Aguilar, W. G., Luna, M. A., Ruiz, H., Moya, J. F., Luna, M. P., Abad, V., & Parra, H. (2017). Statistical Abnormal Crowd Behavior Detection and Simulation for Real-Time Applications. En *Lecture Notes in Computer Science* (págs. 671-682).
- Aguilar, W. G., Manosalvas, J. F., Guillén, J. A., & Collaguazo, B. (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.
- Aguilar, W. G., Quisaguano, F. J., Alvarez, L. G., Pardo, J. A., & Zahira, P. (2018). Monocular depth perception on a micro-UAV using convolutional neuronal networks. En *Lecture Notes in Computer Science* (págs. 392-397).
- Aguilar, W. G., Quisaguano, F. J., Rodríguez, G. A., Alvarez, L. G., Limaico, A., & S, S. D. (2018). Convolutional neuronal networks based monocular object detection and depth perception for micro UAVs. En *Lecture Notes in Computer Science* (págs. 401-410).

- Aguilar, W. G., Rodríguez, G. A., Álvarez, L., Sandoval, S., Quisaguano, F., & Limaico, A. (2017). On-Board Visual SLAM on a UGV Using a RGB-D Camera. En *Lecture Notes in Computer Science* (págs. 298-308).
- Aguilar, W. G., Rodríguez, G. A., Álvarez, L., Sandoval, S., Quisaguano, F., & Limaico, A. (2017). Real-Time 3D Modeling with a RGB-D Camera and On-Board Processing. En *Lecture Notes in Computer Science* (págs. 410-419).
- Aguilar, W. G., Rodríguez, G. A., Álvarez, L., Sandoval, S., Quisaguano, F., & Limaico, A. (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).
- Aguilar, W. G., Salcedo, V., Sandoval, D., & Cobeña, B. (2017). Developing of a Video-Based Model for UAV Autonomous Navigation. *Communications in Computer and Information Science*, 94-105.
- Alcántara-Montiel, C. V., Pedraza-Ortega, J. C., Ramos-Arreguín, J. M., Gorristieta-Hurtado, E., Tovar-Arriaga, S., & Vargas-Soto, J. E. (2019). Detección efectiva de rostros en imágenes utilizando descriptores basados en HOG. *Research in Computing Science*, 148, 371-385.
- Alvarado, J., & Fernández, J. (2012). Análisis de textura en imágenes a escala de grises, utilizando patrones locales binarios (LBP). *ENGI Revista Electrónica de La Facultad de Ingeniería*, 1(1).
- Amaguaña, F., Collaguazo, B., Tituaña, J., & Aguilar, W. G. (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.

- Andrea, C. C., Byron, J. Q., Jorge, P. I., Inti, T. C., & Aguilar, W. G. (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.
- Andriluka, M., Roth, S., & Schiele, B. (2008). People-tracking-by-detection and people-detection-by-tracking. *2008 IEEE Conference on computer vision and pattern recognition*, 1-8.
- Aranda Romasanta, M. (2017). Estudio y aplicación del Filtro de Kalman en fusión de sensores en UAVs.
- Arauz, C. (s.f.). *Práctica: Movimiento Rectilíneo (MRU, MRUV)*.
- Avidan, S. (2004). Support vector tracking. *IEEE transactions on pattern analysis and machine intelligence*, 26(8), 1064-1072.
- Azinhiera, J., Rente, J., & Kellet, M. (1998). Longitudinal Auto-Landing for a Low Cost UAV. *IFAC Proceedings Volumes*, 31(3), 545-550. doi:10.1016/s1474-6670(17)44142-5
- Azinhiera, J. R., & Rives, P. (2008). Imaged-baes visual sevoing for vanishing features and gorund lines tracking: Application to a UAV automatic landing. *International Journal of Optomechatronics*, 2(3), 275-295.
- Barrachina Jordá, J. (2021). Detección e identificación de objetos en movimiento en secuencias de video.
- Barranco, F., Fermüller, C., & Aloimonos, Y. (2014). Estimación de movimiento con sensores asíncronos basados en eventos. *V Jornadas de Computación Empotrada, JCE*, 14.
- Barrientos, A., Del Cerro, J., Gutiérrez, P., San Martín, R., Martínez, A., & Rossi, C. (2007). Vehículos aéreos no tripulados para uso civil. Tecnología y aplicaciones.

- 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.
- Bathija, A., & Sharma, G. (2019). Visual object detection and tracking using Yolo and sort. *International Journal of Engineering Research Technology*, 8(11).
- Bayona, G. A., & Salcedo, I. A. (2012). Evaluación de las redes neuronales artificiales Perceptron Multicapa y Fuzzy-Artmap en la clasificación de imágenes satelitales. *Ingeniería*, 17(1), 61-72.
- Belmonte, C. H. (2019). Desarrollo y comparación de un sistema automático de aterrizaje con arquitecturas moderna y convencional para un UAV de ala Fija.
- Benini, A., Mancini, A., & Longhi, S. (2013). An imu/uwb/vision-based extended kalman filter for mini-uav localization in indoor environment using 802.15. 4a wireless sensor network. *Journal of Intelligent & Robotic Systems*, 70(1), 461-476.
- Birnbaum, Z., Dolgikh, A., Skormin, V., O'Brien, E., Muller, D., & Stracquodaine, C. (2016). Unmanned Aerial Vehicle Security Using Recursive Parameter Estimation. *Journal of Intelligent and Robotic Systems: Theory and Applications*, 84(1-4), 107-120.  
doi:10.1007/s10846-015-0284-1
- Booth, B., & Mitchell, A. (2001). *Getting Started with ArcGIS GIS by ESRI*. ESRI.
- Brachmann, E., & Rother, C. (2019). Neural-guided RANSAC: Learning where to sample model hypotheses. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 4322-4331.
- Bradski, G., & Kaehler, A. (2000). OpenCV. *Dr. Dobb's journal of software tools*, 3.

- Cabras, P., Rosell, J., Pérez, A., Aguilar, W. G., & Rosell, A. (2011). Haptic-based navigation for the virtual bronchoscopy. *18th IFAC World Congress*. Milano, Italy.
- Caiza, O., Aguilar, W. G., 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. G., & 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).
- Cao, X., Wu, C., Yan, P., & Li, X. (2011). Linear SVM classification using boosting HOG features for vehicle detection in low-altitude airborne videos. *2011 18th IEEE International Conference on Image Processing*, 2421-2424.
- Cashmore, M., Fox, M., Long, D., Magazzeni, D., Ridder, B., Carrera, A., . . . Carreras, M. (2015). Rosplan: Planning in the robot operating system. *Proceedings of the International Conference Automated Planning and Scheduling*, 25(1).
- Catalán Vitas, D. (2017). Detección automática de personas mediante Histograma de Gradientes Orientados.
- Childs, C. (2004). Interpolating surfaces in ArcGIS spatial analyst. *ArcUser*, July-September, 3235(569), 32-35.
- Chmaj, G., & Selvaraj, H. (2015). Distributed processing applications for UAV/drones: a survey. *In Progress in Systems Engineering*, 449-454.
- Concia, B., Zumarraga, A., & Mitidieri, P. (2019). Caracterización de sensor de flujo óptico para aplicación en navegación integrada. *V Jornadas de Investigación, Transferencia y Extensión de la Facultad de Ingeniería (La Plata, 2019)*.

- Cremers, D., Engel, J., & Sturm, J. (2012). Camera-based navigation of a low-cost quadcopter. *2012 IEEE/RSJ International Conference on Intelligent Robots and System*, 2815-2821.
- Dang, Q., Le, V., Luqman, M., Coustaty, M., Tran, C., & Ogier, J. (2015). Camera-based document image retrieval system using local features-comparing SRIF with LLAH, SIFT, SURF and ORB. *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, 1211-1215.
- Daramouskas, I., Perikos, I., & Hatzilygeroudis, I. (2012). A Method for Performing Efficient Real-Time Object Tracing for Drones. *Advances in Smart Systems Research*, 6(2), 55.
- Dasso Celis, R. A. (2016). MEDIDAS DE SEGURIDAD EN POLVORINES DE LAS BASES AÉREAS Y ALMACENAMIENTO DE EXPLOSIVOS EN LA FUERZA AÉREA DEL PERÚ-2016.
- Du, J. (2018). Understanding of object detection based on CNN family and YOLO. *Journal of Physics: Conference Series*, 1004(1). doi:10.1088/1742-6596/1004/1/012029
- Ejercito Ecuatoriano. (2018). *NORMA DE SEGURIDAD EN MUNICIÓN EXPLOSIVOS Y EQUIPO ESPECIAL*. Obtenido de [https://www.ejercitoecuadoriano.mil.ec/wp-content/uploads/2018/07/7.-Normas\\_Saneadas.pdf](https://www.ejercitoecuadoriano.mil.ec/wp-content/uploads/2018/07/7.-Normas_Saneadas.pdf)
- Engel, J., Sturm, J., & Cremers, D. (2014). Scale-aware navigation of a low-cost quadcopter with a monocular camera. *Robotics and Autonomous Systems*, 62(11), 1646-1656.
- Enrique, K. (2017). ROTACIÓN, TRASLACIÓN Y COORDENADAS POLARES.
- Enríquez, V. (2006). Adaboost Bayesiano para detección de Rostros-Edición Única.

- Faytong, J., Moggia, G., & Vintimilla, B. (2009). Monitoreo automático de carreteras mediante el uso de un sistema de detección, seguimiento y extracción de características básicas de vehículos con técnicas de visión por computador.
- Fernández, M. (2016). Desarrollo de un sistema de aterrizaje autónomo sobre una plataforma de carga para un UAV tipo Parrot AR. Drone 2.
- Ferrer, G. M. (2009). Integración Kalman de sensores inerciales INS con GPS en un UAV.
- Foeng Vergel, K., & Zabaleta Hernández, J. D. (2005). Vehículos aéreos guiados autonomamente UAVS. *Journal of Chemical Information and Modeling*, 12 Suppl 1(9), 1-29.
- Galarza, J., Pérez, E., Serrano, E., Tapia, A., & G, A. 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. G., & 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.
- Gallardo Valdivia, B. (2007). Estudio de viabilidad de aplicaciones de observación aérea con UAVs y elaboración de un plan de empresa.
- Gandhi, D., Pinto, L., & Gupta, A. (2017). Learning to fly by crashing. *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 3948-3955.
- Goh, G. D., Agarwala, S., Goh, G. L., Dikshit, V., Sing, S. L., & Yeong, W. Y. (2017). Additive manufacturing in unmanned aerial vehicles (UAVs): Challenges and potential. *Aerospace Science and Technology*, 63, 140-151.

- Gómez Redondo, C. (2020). Reconocimiento de una plataforma para aterrizaje de drones mediante técnicas de Visión por Computación.
- Grijalva, S. (2018). SISTEMA DE NAVEGACIÓN AUTÓNOMO BASADO EN VISIÓN PARA PISTAS DE CARRERAS DE DRONES CON MARCAS DE SUPERFICIE. *Tesis de Pregrado, Universidad de las FUerzas Armadas ESPE.*
- Grijalva, S., & Aguilar, W. G. (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., & G, A. W. (2019). HSVM-Based Human Activity Recognition Using Smartphones. En *Lecture Notes in Computer Science* (págs. 217-228).
- Guevara, M., Echeverry, J., & Urueña, W. (2008). Detección de rostros en imágenes digitales usando clasificadores en cascada. *Scientia et technica*, 1(38).
- Hayat, S., Yanmaz, E., & Muzaffar, R. (2016). Survey on unmanned aerial vehicle networks for civil applications: A communications viewpoint. *IEEE Communications Surveys & Tutorials*, 18(4), 2624-2661.
- Hinton, G. (1992). How neural networks learn from experience. *Scientific American*, 267(3), 144-151.
- Huang, R., Pedoeem, J., & Chen, C. (2018). YOLO-LITE: a real-time object detection algorithm optimized for non-GPU computers. *2018 IEEE International Conference on Big Data*, 2503-2510.
- Huh, S., & Shim, D. H. (2010). A vision-based automatic landing method for fixed-wing UAVs. *Journal of Intelligent and Robotic Systems*, 57(1), 217-231.

- Jalled, F., & Voronkov, I. (2016). Object detection using image processing. *ArXiv Preprint ArXiv: 1611.07791*.
- Jara-Olmedo, A., Medina-Pazmiño, W., Mesías, R., Araujo-Villaroel, B., Aguilar, W. G., & 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. G., & Pardo, J. A. (2018). E-services from Emergency Communication Network: Aerial Platform Evaluation. *International Conference on eDemocracy & eGovernment (ICEDEG)* (págs. 251-256). IEEE.
- Jiménez, J., & Baturone, A. (1996). Estimación de la posición de un robot móvil. *Automática*, 29, 3-18.
- Kadota, R., Sugano, H., Hiromoto, M., Ochi, H., Miyamoto, R., & Nakamura, Y. (2009). Hardware architecture for HOG feature extraction. *2009 Fifth International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, 1330-1333.
- Katherine, F. P., & Gutierrez Barboza, E. (2019). Optimización de la asignatura de medidas de seguridad de almacenamiento de clase V y su relación con la instrucción técnica de los cadetes de material de guerra de la Escuela Militar de Chorrillos, año 2018.
- Kendoul, F. (2012). Survey of advances in guidance, navigation and control of unmanned rotorcraft systems. *Journal of Field Robotics*, 29(2), 315-378.
- Kissai, A., & Smith, M. (2019). UAV Dead Reckoning with and without using INS/GPS Integrated System in GPS denied Polar Regions. *Int J Aeronaut Aerosp Eng*, 1(2), 58-67.

- Kwak, J., & Sung, Y. (2018). Autonomus UAV flight control for GPS-based navigation. *IEEE Access*, 6, 37947-37955.
- Lange, S., Sunderhauf, N., & Protzel, P. (2009). A vision based onboard approach for landing and position control of an autonomous multirotor UAV in GPS-denied environments. *2009 International Conference on Advanced Robotics*, 1-6.
- Lee, D., Ryan, T., & Kim, H. J. (2012). Autonomous landing of a VTOL UAV on a moving platform using image-based visual servoing. *2012 IEEE International Conference on Robotics and Automation*, 971-976.
- León Urbina, C. E. (2020). Optimización de los sistemas de seguridad y video vigilancia en los polvorines del Ejército.
- Lienhart, R., & Maydt, J. (2002). An extended set of haar-like features for rapid object detection. *International Conference on Image Processing*, 1, 1-1.
- López, R. (2021). MANEJO RESPONSABLE DE LAS MUNICIONES Y EXPLOSIVOS, CONFIANZA, SEGURIDAD Y NOBLEZA. *Revista de Ciencias de Seguridad y Defensa*, 1(2), 4.
- Martínez Redondo, M. (2016). Evaluación comparativa de técnicas de detección y descripción de puntos de interés en imágenes.
- Martínez, M. S. (2018). DETECCIÓN DE PERSONAS MEDIANTE TÉCNICAS DE PARENDIZAJE AUTOMÁTICO: SVM Y CNN.
- McKenna, S., Jabri, S., Duric, Z., Rosenfeld, A., & Wechsler, H. (2000). Tracking groups of people. *Computer vision and image understanding*, 80(1), 42-56.

- Merizalde, D., Aguilar, W. G., & 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).
- Montañez, M., & Almendra, F. (2021). Manipulación de clase v en los polvorines por los cadetes de tercer año de material de guerra de la Escuela Militar de Chorrillos “Coronel Francisco Bolognesi”--2019.
- Ogata, K. (2010). *Modern control engineering*.
- Orbea, D., Moposita, J., Aguilar, W. G., 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).
- Orbea, D., Moposita, J., Aguilar, W. G., 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.
- Pang, Y., Yuan , Y., Li, X., & Pan, J. (2011). Efficient HOG human detection. *Signal Processing*, 91(4), 773-781.
- Pardo, J. A., Aguilar, W. G., & 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. (2019). *Parrot*. Obtenido de [https://developer.parrot.com/docs/bebop/ARSDK\\_Protocols.pdf](https://developer.parrot.com/docs/bebop/ARSDK_Protocols.pdf)

- Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., . . . Andrew, N. (2009). ROS: an open-source Robot Operating System. *ICRA workshop on open source software*, 3(3.2), 5.
- Rahmad, C., Asmara, R. A., Putra, D. R., Dharma, I., Darmono, H., & Muhiqqin, I. (2020). Comparison of Viola-Jones Haar Cascade classifier and histogram of oriented gradients (HOG) for face detection. *IOP Conference Series: Materials Science and Engineering*, 732(1), 12038.
- Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. *Proceedings of the IEEE Conference on Computer Vision Pattern Recognition*, 779-788.
- Romero Paredes, L. (2014). Diseño y construcción de un módulo de control PID para la estabilización de un cuadricoptero.
- Romero Serna, E. (2015). Desarrollo de un prototipo para el geoposicionamiento referenciado por GPS de un aeromodelo tipo quadcoptero.
- Rublee, E., Rabaud, V., Konolige, K., & Bradski, G. (2011). ORB: An efficient alternative to SIFT or SURF. *2011 International conference on computer vision*, 2564-2571.
- Salcedo, V. S., Aguilar, W. G., 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).
- Sánchez Martínez, L. (2012). Identificación experimental de un avión no tripulado (UAV).
- Schapire, R. (2013). Explaining adaboost. En *Empirical inference* (págs. 37-52).
- Scott, L. M., & Janikas, M. V. (2010). Spatial statistics in ArcGIS. In *HandBook of applied spatial analysis*, 27-41.

- Segarra, D., Caballeros, J., Aguilar, W. G., 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).
- Shakhatreh, H., Sawalmeh, A. H., Al-Fuqaha, A., Dou, Z., Almaita, E., Khalil, I., . . . Guizani, M. (2019). Unmanned aerial vehicles (UAVs): A survey on civil applications and key research challenges. *IEEE Access*, 7, 48572-48634.
- Shirai, J., Yamaguchi, T., & Takaba, K. (2017). Remote visual servo tracking control of drone taking account of time delays. *2017 56th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE)*, 1589-1594.
- Soo, S. (2014). Object detection using Haar-cascade Classifier. *Institute of Computer Science, University of Tartu*, 2(3), 1-12.
- Steinera, G., Shapiro, A., & Destefanis, E. (2019). Implementación de un sistema de detección de troncos de árboles utilizando histogramas de gradientes orientados (HOG).
- Strasdat, H., Montiel, J., & Davison, A. J. (2010). Scale drift-aware large scale monocular SLAM. *Robotics: Science and Systems VI*, 2(3), 7.
- Suarez, E. J. (2014). Tutorial sobre máquinas de vectores soporte (SVM). *Tutorial sobre máquinas de Vectores Voporte (SVM)*, 1-12.
- Suárez-Sánchez, A., & Loaiza-Correa, H. (2015). Esquema de navegación reactiva con sensores RGB-D. *Revista UIS Ingenierías*, 14(1), 7-19.
- Tabassum, M., & Mathew, K. (2014). Software evolution analysis of linux (Ubuntu) OS. *2014 International Conference on Computational Science and Technology (ICCST)*, 1-7.  
doi:10.1109/ICCST.2014.7045194

- Tenezaca, D., Canchignia, C., Aguilar, W. G., & 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).
- Valseca Martínez, V. (2018). Exploración coordinada entre robot móvil terrestre y dron aéreo.
- Vargas Rojas, W. (2018). Tracking de personas en entornos industriales robotizados con imágenes RGB-D.
- Velasco Montero, D. (2016). Extracción de características en imágenes de rayos X de cuadros.
- Vera Meaurio, D., & Martínez Jara, E. (2015). Conteo de personas en imagen y video mediante la técnica de Viola-Jones a través de clasificadores Haar utilizando software libre.
- Vera, B. A., Andrés, B., Castillo, P. A., & Geovanny, W. (2020). Seguimiento y búsqueda de objetivos en entornos complejos usando micro vehículos aéreos con cámaras monoculares para aplicaciones militares.
- Visakha, K., & Prakash, S. (2018). Detection and tracking of human beings in a video using haar classifier. *2018 International Conference on Inventive Research in Computing Applications (ICIRCA)*, 1-4.
- Wenzel, K. E., Masselli, A., & Zell, A. (2011). Automatic take off, tracking and landing of a miniature UAV on a moving carrier vehicle. *Journal of Intelligent & Robotic Systems*, 61(1), 221-238.
- Wilson, P. I., & Fernandez, J. (2006). Facial feature detection using Haar classifiers. *Journal of Computing Sciences in Colleges*, 21(4), 127-133.
- Yang, K., & Quan, Q. (2020). An Autonomous Intercept Drone with Image-based Visual Servo., (págs. 2230-2236).

Zabala, M., Cuenca, L., León, J., & Cabrera, F. (2018). Arquitectura de acoplamiento entre INS/GPS para navegación precisa en trayectorias establecidas. *Maskay*, 8(1), 20-26.

Zambrano, J. (2010). *FISICA VECTORIAL BASICA 1*. EDICIONES MAYA.

Zurita, M., Aguilar, W. G., & 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).

## **ANEXOS**

**Anexo A: Error de distancia Longitud Circuito Cuadrado**

**Anexo B: Error de distancia Latitud Circuito Cuadrado**

**Anexo C: Error de posición total entre coordenadas Circuito Cuadrado**

**Anexo D: Error de distancia de Longitud Circuito con forma de Rombo**

**Anexo E: Error de distancia de Latitud Circuito con forma de Rombo**

**Anexo F: Error de posición total entre coordenadas Circuito con forma de Rombo**

**Anexo G Error de distancia de Longitud Circuito con forma de Trapecio**

**Anexo H Error de distancia de Latitud Circuito con forma de Trapecio**

**Anexo I Error de posición total entre coordenadas Circuito con forma de Trapecio**