



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO
FACULTAD DE INFORMÁTICA Y ELECTRÓNICA
ESCUELA DE INGENIERÍA ELECTRÓNICA EN CONTROL Y REDES
INDUSTRIALES

**“IMPLEMENTACIÓN DE UN DRON ANTICOLISIONES CAPAZ
DE SEGUIR UNA TRAYECTORIA PRE ESTABLECIDA PARA LA
ELABORACIÓN DE UN MAPA DE LA TASA DE OCUPACION
VEGETAL EN UNA ZONA.”**

TRABAJO DE TITULACIÓN
TIPO: DISPOSITIVO TECNOLÓGICO
Presentado para optar al grado académico de:
INGENIERO EN ELECTRÓNICA CONTROL Y REDES
INDUSTRIALES

AUTORES: JHONY PAUL TACURI AUQUILLA
WILSON JHONATAN OLMEDO CARRILLO
TUTOR: ING. HENRY VALLEJO VIZHUETE

Riobamba – Ecuador

2019

©2019, Jhony Paul Tacuri Auquilla, Wilson Jhonatan Olmedo Carrillo.

Se autoriza la reproducción total o parcial, con fines académicos, por cualquier medio o procedimiento, incluyendo la cita bibliográfica del documento, siempre y cuando se reconozca el Derecho de Autor.

-

ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

FACULTAD DE INFORMÁTICA Y ELECTRÓNICA
ESCUELA DE INGENIERÍA ELECTRÓNICA EN TELECOMUNICACIONES Y
REDES

El Tribunal del Trabajo de Titulación certifica que la propuesta tecnológica “IMPLEMENTACIÓN DE UN DRON ANTICOLISIONES CAPAZ DE SEGUIR UNA TRAYECTORIA PRE ESTABLECIDA PARA LA ELABORACIÓN DE UN MAPA DE LA TASA DE OCUPACION VEGETAL EN UNA ZONA.”, de responsabilidad del señor Jhony Paul Tacuri Auquilla y señor Wilson Jhonatan Olmedo Carrillo, ha sido minuciosamente revisado por los Miembros del Tribunal del Trabajo de titulación, quedando autorizada su presentación.

NOMBRE	FIRMA	FECHA
Dr. Julio Santillán C. VICEDECANO DE LA FACULTAD DE INFORMÁTICA Y ELECTRÓNICA	_____	_____
Ing. Freddy Chávez V. DIRECTOR DE ESCUELA DE INGENIERÍA EN CONTROL Y REDES INDUSTRIALES	_____	_____
Ing. Henry Vallejo V. DIRECTOR DEL TRABAJO DE TITULACIÓN	_____	_____
Ing. Fausto Cabrera A. MIEMBRO DEL TRIBUNAL	_____	_____

Nosotros, Jhony Paul Tacuri Auquilla y Wilson Jhonatan Olmedo Carrillo. Somos responsable de las ideas, doctrinas y resultados expuestos en esta Trabajo de Titulación y el patrimonio intelectual de la Trabajo de Titulación de Grado pertenece a la Escuela Superior Politécnica de Chimborazo.

Jhony Paul Tacuri Auquilla,

Wilson Jhonatan Olmedo Carrillo.

DEDICATORIA

A Dios por darnos todo lo necesario y apartar de nuestro camino lo que no hace falta, en la culminación de una etapa más de muchas que vendrán en el transcurso de nuestras vidas.

“IN GOD WE TRUSTH”

TABLA DE CONTENIDO

RESUMEN	xiv
ABSTRACT	xv
INTRODUCCIÓN	1
ANTECEDENTES	1
OBJETIVOS	5
CAPÍTULO I	
1. MARCO TEÓRICO	6
1.1. Drones	6
1.1.1. Tipos	7
1.1.1.1. <i>Según su uso</i>	7
1.1.1.1.1. <i>Militares</i>	7
1.1.1.1.2. <i>Civiles</i>	7
1.1.1.2. <i>Según su diseño</i>	7
1.1.1.2.1. <i>Ala Fija</i>	7
1.1.1.2.2. <i>Ala rotatoria</i>	8
1.1.2. Quacopter	8
1.1.2.1. <i>Partes de un Dron</i>	8
1.1.2.1.1. <i>Estructura</i>	9
1.1.2.1.2. <i>Motores Y Hélices</i>	9
1.1.2.1.3. <i>ESC Controlador electrónico de velocidad</i>	9
1.1.2.1.4. <i>Control de vuelo</i>	9
1.1.2.1.5. <i>Radio Control</i>	9
1.1.2.1.6. <i>Batería</i>	10
1.1.3. Programación y desarrollo de aplicaciones	10
1.1.3.1. <i>DroneKit</i>	10
1.1.3.2. <i>Las APIs de DJI</i>	11
1.1.3.3. <i>ROS</i>	11
1.1.4. Control desde una estación en tierra	13
1.1.4.1. <i>Mission Planner</i>	13
1.1.4.2. <i>QGround Control</i>	14
1.1.5. Aplicaciones	15
1.1.5.1. <i>Inspección Aérea de Líneas Eléctricas</i>	15
1.1.5.2. <i>Transporte de Materiales</i>	15
1.1.5.3. <i>Agricultura de Precisión</i>	16
1.1.5.4. <i>Drones para sustitución militar</i>	16

1.1.5.5.	<i>Vista de águila “Fotoperiodismo”</i>	16
1.1.5.6.	<i>Aplicación en la seguridad</i>	17
1.1.6.	<i>Kits de desarrollo de drones</i>	17
1.1.6.1.	<i>Parrot SLAMdunk</i>	17
1.1.6.2.	<i>Intel Aero Ready to Fly</i>	18
1.2.	Cámara Xioma	19
1.3.	Sistema anticollisiones	20
1.3.1.	Sistemas basados en Radio Frecuencia	20
1.3.2.	Sistemas basados en Reconocimientos de Patrones	21
1.3.2.1.	<i>Intel RealSense en detección de obstáculos</i>	21
1.3.2.2.	<i>Especificaciones de la cámara R200</i>	21
1.3.2.3.	<i>Biblioteca de prevención de colisiones</i>	22
1.3.2.4.	<i>Estrategias de evasión y detectores</i>	22
1.3.2.5	<i>Algoritmo de detección y evasión de obstáculos</i>	22
1.3.3.	Sistemas basados en sensores ópticos-infrarrojos	23
1.3.4.	Sistemas basados en sensores ultrasónico	24
1.4.	Procesamiento de imágenes	24
1.4.1.	Image-Stitching	25
1.4.1.1.	<i>Detector de puntos clave</i>	26
1.4.1.2.	<i>Descriptores de Puntos de Interés</i>	27
1.4.1.3.	<i>Resolución de Homografía</i>	31
1.4.1.4.	<i>Unión de imágenes</i>	31
1.4.2.	Segmentación de imágenes	32
1.4.2.1.	<i>Umbralización</i>	32
1.4.2.2.	<i>Discontinuidad</i>	33
1.4.2.3.	<i>Similitud</i>	33
CAPITULO II		
2.	DESARROLLO E IMPLEMENTACIÓN	35
2.1.	Elección del kit de desarrollo	35
2.1.1.	Intel Aero RTF	36
2.1.1.1.	<i>Intel Aero Compute Board</i>	37
2.1.1.2.	<i>Intel Aero Vision</i>	37
2.1.1.3.	<i>Intel Aero controller With</i>	37
2.1.1.4.	<i>Partes de ensamblaje del dron</i>	38
2.2.	Montaje	39
2.3.	Configuración Inicial del Intel Aero RTF	43
2.3.1.	Arquitectura y Especificaciones de Hardware de la placa Intel Aero	43

2.3.2.	<i>Consideraciones en la configuración y especificaciones de software</i>	45
2.3.3.	<i>Configuración Inicial</i>	45
2.4.	Calibración	48
2.4.1.	<i>Selección del programa de calibración</i>	48
2.4.2.	<i>Resumen del estado de calibración</i>	48
2.4.3.	<i>Configuración de frame (Fuselaje)</i>	49
2.4.4.	<i>Configuración del radio control</i>	49
2.4.5.	<i>Configuración de los sensores</i>	50
2.4.6.	<i>Configuración del modo de vuelo</i>	51
2.5.	Sistema anticollisiones	51
2.5.1.	<i>Sistema anticollisiones basado en sensores ultrasónicos</i>	52
2.6.	Control en tierra	57
2.6.1.	<i>Planificación del vuelo autónomo</i>	58
2.7.	Captura de imágenes y procesamiento	59
2.8.	Reconstrucción	60
2.8.1.	<i>Algoritmo de Reconstrucción de Imagen</i>	60
2.8.1.1.	<i>Algoritmo: reconocimiento panorámico</i>	60
2.8.2.	<i>Implementación del algoritmo</i>	61
2.8.2.1.	<i>Librería vlfeat</i>	61
2.8.3.	Componentes del proyecto	61
2.8.3.1.	<i>Función codigo.m</i>	61
2.8.3.2.	<i>Función interfaz.m</i>	62
2.8.3.3.	<i>Función graficar.m</i>	63
2.8.4.	Matlab y Guide	64
2.8.5.	Obtención de la tasa de ocupación Vegetación	66
2.8.5.1.	<i>Segmentación de Imagen</i>	66
2.8.5.2.	<i>Descripción de funcionamiento</i>	67
2.8.6.	<i>Calculo del porcentaje de ocupación vegetal</i>	67
CAPITULO III		
3.	EVALUACIÓN DE RESULTADOS	69
3.1.	Parámetros de vuelo	69
3.2.	Características del sistema anticollisiones	69
3.3.	Planificación de la Misión	71
3.4.	Capturas de imágenes	73
3.5.	Reconstrucción de imagen	74
3.6.	Determinación del índice de ocupación vegetal	75
CONCLUSIONES		77

RECOMENDACIONES.....	78
BIBLIOGRAFÍA	
ANEXOS	

ÍNDICE DE TABLAS

Tabla 1-1: Descripción de las características principales que posee la cámara Xiaomi YI Lite Action.....	19
Tabla 2-1: Características fundamentales en las técnicas de evasión.....	22
Tabla 1-2: Características de los elementos que poseen los kits SLAMdunk e Intel Aero.....	35
Tabla 2-2: Comparación entre SLAMdrunk e Intel Aero.....	36
Tabla 3-2: Características de los principales elementos del Drone.....	38
Tabla 1-3: Valores tolerables en los cuales el sistema basado en ultrasónico será eficiente.....	70
Tabla 2-3: Comparación entre los valores estimados y reales del sistema de detección de obstáculos.....	71
Tabla 3-3: Valores medidos en misiones a diferentes alturas.....	73
Tabla 4-3: Comparación de resultados de la reconstrucción de imágenes, variando parámetros.....	75
Tabla 5-3: Tabla de comparación de resultados de los índices de ocupación vegetal mostrada en el programa.....	76

ÍNDICE DE GRÁFICAS

Gráfico 1-2: Código para cargar los archivos de las imágenes.....	62
Gráfico 2-2: Código para realizar segmentación de imagen.....	63
Gráfico 3-2: Código para graficar líneas.....	63
Gráfico 4-2: Botones de control del programa.....	65
Gráfico 5-2: Botones de subdivisión y Muestra de Resultados.....	65
Gráfico 6-2: Pruebas de similitud de color.....	66
Gráfico 7-2: Pruebas de similitud cambiando el pixel de comparación.....	67
Gráfico 8-2: Subdivisión del mapa de ocupación vegetal en áreas pequeñas.....	68

ÍNDICE DE FIGURAS

Figura 1-1: Estructura común en un UAV (Vehículo Aero no Tripulado).....	6
Figura 2-1: Modelo básico de un UAV de ala fija.....	7
Figura 3-1: 020Modelo UAV de ala rotatoria “Drone H529 de Yunecc”	8
Figura 4-1: Esquema general de un UAV con sus elementos principales	9
Figura 5-1: Trayectoria que el UAV Mission Planner.....	13
Figura 6-1: Planificación de trayectoria, QGround Control	14
Figura 7-1: Envío de paquetes mediante un dron.	15
Figura 8-1: Dimensiones frontal y lateral de la cámara Xiomia YI Lite	19
Figura 9-1: Tarjeta RFID con Antena.....	20
Figura 10-1: Principio de funcionamiento de Intel RealSense	21
Figura 11-1: Diagrama de flujo base sobre detección y evasión	23
Figura 12-1: Detección de obstáculos con sensores infrarrojos.....	23
Figura 13-1: Dji Matriz 100, con sistema ultrasónico	24
Figura 14-1: Unión de imágenes con Stitching.....	25
Figura 15-1: Diferencia Gaussiana, pirámide de escala de David Lowe	28
Figura 16-1: Gradientes en la ventana Gaussiana e Histograma de puntos	30
Figura 17-1: Espacio escala mediante SIFT (izq.) y SURF (dcha.).....	30
Figura 18-1: Histograma imagen con fondo blanco	32
Figura 19-1: Histograma imagen con discontinuidad.....	33
Figura 20-1: Segmentación por color	34
Figura 1-2: Frame F50,	39
Figura 2-2: Motor Typhoon H 940kv	39
Figura 3-2: ESC, Yunecc FOC ESC.....	39
Figura 4-2: Intel Aero Compute Board.....	40
Figura 5-2: Placa de Energía.....	40
Figura 6-2: Cámara Intel RealSense R200.....	40
Figura 7-2: Módulo GPS con brújula	40
Figura 8-2: Ensamblaje del Drone y pruebas de funcionamiento	41
Figura 9-2: Dron Intel Aero	41
Figura 10-2: Protectores de hélices, Solid Works.....	42
Figura 11-2: Protector de hélices en la base del motor.....	42
Figura 12-2: Vista inferior del dron, con cámara.....	43
Figura 13-2: Estructura final del Drone	43
Figura 14-2: Diagrama de bloques Intel Aero Compute Board.....	44

Figura 15-2: Diagrama de bloques Intel Aero Flight Controller	44
Figura 16-2: Resultados de actualización del sistema de Intel Aero RTF	47
Figura 17-2: Ventana de estados de los sensores del UAV en QGC (QGroundControl)	48
Figura 18-2: Selección del Fuselaje o Airframe del UAV	49
Figura 19-2: Calibración del radio control	50
Figura 20-2: Calibración de sensores.....	50
Figura 21-2: Configuración de los canales de los modos de vuelo.....	51
Figura 22-2: Sensor ultrasónico.....	52
Figura 23-2: Módulo NRF24101 y Antena 2.4GHz.....	53
Figura 24-2: Diagrama de flujo del transmisor.....	54
Figura 25-2: Diagrama de flujo del receptor.....	55
Figura 26-2: Valores estimados del ángulo de giro respecto al voltaje del acelerador.....	55
Figura 27-2: Diagrama del transmisor del sistema anticollisiones	55
Figura 28-2: Diagrama del receptor del sistema anticollisiones	56
Figura 29-2: Sistema de paro de emergencia sujetado.....	56
Figura 30-2: Planificación de una misión en QGroundControl	58
Figura 31-2: Ventana del plan de vuelo.....	59
Figura 32-2: Interfaz para la determinación de la tasa de ocupación vegetal	64
Figura 1-3: Estimación de la distancia de descenso del dron.....	70
Figura 2-3: Imagen de las zonas de cultivos de la parroquia de Yaruquies.....	71
Figura 3-3: Esquema de la altura.....	72
Figura 4-3: Ruta pre establecida a cubrir.....	72
Figura 5-3: Secuencia de imágenes obtenidas en el cultivo de alfalfa.....	73
Figura 6-3: Secuencia de imágenes	74
Figura 7-3: Imagen reconstruida.....	74
Figura 8-3: Imagen reconstruida y resultados obtenidos	76

RESUMEN

Este trabajo de titulación tuvo como objetivo la implementación de un sistema de vuelo no tripulado con la finalidad de brindar ayuda en aspectos específicos dentro del cálculo y estimación de ocupación vegetal en una zona determinada, el desarrollo del prototipo se basó en tres etapas. La primera etapa: establecer, adquirir e implementar los elementos necesarios; siendo el kit de desarrollo Intel Aero, el mismo que es un dron pre armado y listo para volar, además se realizó un sistema anticollisiones usando un Arduino y sensor ultrasónico para la detección de obstáculos. La segunda etapa consistió en la calibración del dron y adquisición de imágenes en vuelo, para lo cual se utilizó el programa QGround Control, con el cual se logró establecer comunicación con el dron para calibrar, configurar y monitorear los parámetros de vuelo. Las imágenes fueron capturadas con una cámara tipo deportiva, colocada en la parte inferior del dron con la cual se capturaron imágenes aéreas que abarca el espacio de la ruta trazada por el dron. Por último, se realizó una reconstrucción de imágenes, que se llevó a efecto con el módulo Image-Stitching del entorno de desarrollo Matlab, dentro de este utilizamos un algoritmo de segmentación el cual nos permite separar a los pixeles según su color obteniendo así el valor de ocupación vegetal en la imagen reconstruida. Las pruebas fueron realizadas en un cultivo de alfalfa de 200 metros cuadrados, en el mismo que se pudo determinar que existía en promedio 65% de ocupación vegetal en todo el terreno, es decir que 139 metros cuadrados eran de vegetación. Logrando así, a partir de imágenes aéreas capturadas por un dron, determinar los lugares que poseían menos densidad de plantas; recomendando los sectores del cultivo donde se deben aplicar más recursos agrícolas.

Palabras clave: <TECNOLOGÍA Y CIENCIAS DE LA INGENIERÍA>, <CONTROL Y REDES INDUSTRIALES>, <DRON <ANTICOLISIONES>, <TRAYECTORIA>, <SEGEMENTACIÓN>, <VEGETACIÓN>, <RECONSTRUCCIÓN>, <ARDUINO>

ABSTRACT

The objective of this degree work was to implement an unmanned flight system with the purpose of providing assistance in specific aspects of calculating and estimating vegetation occupation in a specific area. The development of the prototype was based on three stages. The first stage: to establish, to acquire and to implement the necessary elements; being the Intel Aero development kit, the same as a pre-armed and ready-to-fly drone, an anti-collision system was also made using an Arduino and ultrasonic sensor for obstacle detection. The second stage consisted of the drone calibration and flight image acquisition, for which the QGround Control program was used, it was used to establish communication with the drone to calibrate, configure and monitor the flight parameters. The images were captured with a sports-type camera, placed in the lower part of the drone with which images were captured that cover the space of the route traced by the drone. Finally, an image reconstruction was carried out with the Image-Stitching module of the Matlab development environment, within this we use a segmentation algorithm which allows us to separate the pixels according to their color obtaining the value of vegetable occupation in the whole land, it means that 139 square meters were vegetation. Achieving this way, from aerial images captured by a drone, determine the places that had less density of plants; recommending crop sectors where more agricultural resources should be applied.

Keywords: <ENGINEERING TECHNOLOGY AND SCIENCE> <CONTROL AND INDUSTRIAL NETWORKS> <DRON> <ANTI-COLLISIONS> <TRAJECTORY> <SEGMENTATION> <VEGETATION> <RECONSTRUCTION> <ARDUINO>

INTRODUCCIÓN

Con el desarrollo continuo de la electrónica y la aceptación positiva de la sociedad se han buscado medios que faciliten las actividades que deben tratarse día a día, por tanto, reemplazar actividades desarrolladas por el hombre, con máquinas, es un medio que facilita y acelera el proceso de desarrollo de cada una de ellas debido a que las máquinas no generan agotamiento por repetición o tampoco presentan la dificultad conocida como “error humano” para llevar a cabo muchas tareas. Por lo tanto, en este proyecto, adquirir imágenes sobre terrenos de cultivo, con una mayor facilidad, es la tarea que se llevará a cabo. Mediante un dispositivo de vuelo no tripulado reutilizable es como se pretende dar solución a esta problemática.

ANTECEDENTES

Anteriormente, las dimensiones de cultivos no eran muy extensas, por lo tanto, no era difícil recorrerlas completamente. No obstante, la creciente demanda de alimentos necesita de cultivos más extensos, que se vuelven demasiado grandes para poderlos controlar perfectamente cómo se hacía (González, Amarillo, Amarillo, & Sarmiento, 2016, p2). Esta problemática genera que las personas no puedan conocer de manera puntual las necesidades de los cultivos que manejan y tomen decisiones generalizadas de manera equivocada. (González et al., 2016, p2).

Los UVA, comúnmente conocidos como drones, no disponen de un piloto en su interior, estos están dirigidos por una persona o sistema electrónico externo, que decide en cada momento la acción a tomar, la habilidad y los sentidos del piloto son reemplazadas por sensores electrónicos de gran precisión (García, 2017, p13). Estas características los hacen que puedan ser utilizados en varias áreas como un simple jovi, seguridad, deportes, en el área de búsqueda, exploración hasta su uso en la agricultura como, por ejemplo: Se llevó a cabo un proyecto que consistía en encontrar maleza en los cultivos de maíz, cuando estos cultivos empezaban a desarrollarse, haciendo el estudio con imágenes adquiridas por un dron (GIUSIANO MONICA, 2017, p9) proyecto que fue llevado a cabo por Giusiano, Monica Noemí en Argentina.

El reciente interés de utilizar imágenes aéreas adquiridas por drones se debe al potencial de información que se puede obtener de una simple fotografía, la integración de estas con sistemas de posicionamiento global, sistemas de control de campo y percepción remota, ayudan a la planeación y ejecución de las labores agrícolas (C. Garcia, Herrera, & Erazo Mesa, 2015, p2). Como ejemplo de esto tenemos: “Sistema de procesamiento de imágenes RGB aéreas para

agricultura de precisión” (Martinez, 2016, p1). Este trabajo en comparación con el nuestro, tienen en común el realizar un procesamiento de las imágenes obtenidas por drones, esto con el fin de obtener información sobre el estado de los cultivos, demostrando la importancia del procesamiento de las imágenes para una buena interpretación de los datos obtenidos.

Gracias a la implementación de UVA, la agricultura tiene una nueva herramienta para realizar tareas más eficientes, a modo de ejemplo, la localización e identificación exacta de las zonas con presencia de malezas, se genera un ahorro de insumos(herbicidas) debido que estos se aplicarían solo donde son necesarios (Garcia, Herrera, Mesa, 2015, p2).

“Se evidencia que a partir de los drones, vistos como una herramienta tecnológica innovadora, es posible resolver las problemáticas expuestas en campos de cultivos de gran extensión” (Koh & Wich 2012) citado por (González et al., 2016^a, p2), Al aplicar las nuevas tecnologías a la agricultura actual pretendemos mejorar la producción de los cultivos así como minimizar el esfuerzo que deben realizar los agricultores, es decir pretendemos dar a conocer las técnicas actuales que muchos agricultores desconocen, intentando introducir poco a poco el concepto de agricultura de precisión.

JUSTIFICACIÓN

Justificación Teórica

El mapeo de las plantaciones para detectar plagas y definir la cobertura vegetal, si se la quiere llevar a cabo de una manera eficaz, es una tarea de gran complejidad. (Martinez, 2016, p9), entonces se puede optar por una manera eficaz de obtener información sobre el estado de los cultivos con la ayuda de los drones.

“Adquirir los datos necesarios, monitorear continuamente y calificar el estado de los sembríos, son tareas que requieren un conjunto concreto de instrucciones que definen una secuencia de pasos ordenados para así poder cumplir con objetivos necesarios para el cuidado de los cultivos, todo esto lleva mucho tiempo debido a que anteriormente dichas tareas se podían llevar a cabo con la ayuda de un esfero y un papel, no obstante con el crecimiento de la demanda de alimentos, este método es mucho más propenso a errores” (Palmer, 2012, p2) citado por (Rea Sanchez, Maldonado Cevallos, & Villao Santos, 2015, p3).

Debido a estos problemas expuestos por los autores anteriormente citados se ha implementado un dron que permite recorrer un cultivo de dimensiones determinadas en menor tiempo, además se

realizaron pruebas de un sistema anticollisiones y por medio de una aplicación de rutas nuestro dispositivo es capaz de seguir una trayectoria pre-establecida, con ello se realizaron pruebas para que el dron sea capaz de evitar los posibles obstáculos que se puedan presentar en su recorrido. Posteriormente se ha conseguido la obtención de imágenes aéreas las cuales son necesarias para obtener información relevante sobre el estado de los cultivos.

Con este proyecto lo que se desea es realizar un monitoreo más exhaustivo en los cultivos y de este modo localizar las zonas donde la producción del terreno se encuentre por debajo del rango aceptable, superando así los inconvenientes que conlleva un monitoreo y control constante de un cultivo, ya que, con el uso de tecnologías, se podrá realizar esta tarea de forma repetitiva las veces que se crea necesario. Al obtener y almacenar datos del cultivo podremos realizar una mejor utilización de los recursos, ya que nos permitirá estimar zonas específicas en las cuales el cultivo necesita de estos, ya sea agua, abono o insecticidas. Al lograr determinar la ubicación exacta donde se encuentra el problema podremos centrar los recursos a la misma, obteniendo así un ahorro de recursos que generará un ahorro económico al agricultor y contribuyendo así a la conservación y sostenibilidad ambiental.

Justificación Aplicativa

El trabajo de recorrer el terreno se vuelve tedioso, cansado y repetitivo, además que demanda de mucho tiempo, por este motivo muchos no realizan este control, y siendo más complicado realizarlo en terrenos de gran extensión, este es el motivo por el cual muchos agricultores optan por realizar un trabajo rutinario sobre sus cultivos es decir que cada cierto tiempo determinado realizan el riego, fumigación, abono, etc.

Se desarrolló entonces un dron, que servirá para recorrer el cultivo en su totalidad, este prototipo se realizara en una tarjeta de desarrollo específica para vehículos aéreos no tripulados, la cual será seleccionada en base a los requerimientos de vuelo, así como condiciones ambientales y a las necesidades que se puedan presentar durante el desempeño de su trabajo.

Con este dron se realizará un mapeo de los cultivos, recorriendo una trayectoria dada por el usuario, estableciendo comunicación con una estación de control en tierra, la misma que tendrá el control del dron, recibirá datos y por donde se podrá ingresar la trayectoria que desee el usuario. Con este dron se procederá a la captura y almacenamiento de imágenes aéreas de los cultivos, las cuales serán sometidas a un procesamiento digital, y mediante la realización de cálculos con los datos obtenidos se podrá determinar la tasa o índice de ocupación vegetal en el cultivo. Además, se probará un sistema anticollisiones, el cual mejora la autonomía del dron, esto permitirá que el

mismo pueda tener una mejor protección en su integridad física debido a que se puede desplazar evitando obstáculos presentes en la trayectoria establecida.

Conocer si existe una buena salud y cobertura vegetal, brinda grandes ventajas al momento de tomar medidas. La información obtenida es útil para ayudar a tomar decisiones en el momento de suministrar insumos agrícolas y mejorar el crecimiento de los cultivos (Martinez, 2016, p10).

OBJETIVOS

Objetivo General

Implementar un dron anticollisiones capaz de seguir una trayectoria pre establecida para elaborar un mapa de la tasa de ocupación vegetal en una zona.

Objetivos Específicos

- Investigar los conceptos, modelos, tipos y aplicaciones principales de los UAV.
- Investigar sobre las tecnologías de sistemas anticollisiones, para conocer los distintos usos y elementos necesarios para su aplicación.
- Identificar el kit de desarrollo que mejor se adapte a las necesidades de vuelo y adquisición de imágenes.
- Establecer una comunicación entre el dron y una estación de control en tierra.
- Desarrollar un procesamiento de imágenes para determinar la tasa de ocupación vegetal en una zona especificada.
- Realizar la reconstrucción de imágenes obtenidas para elaborar un mapa de la zona.
- Verificar el funcionamiento tanto del vuelo del dron, así como de la captura y procesamiento de imágenes.

CAPÍTULO I

1. MARCO TEÓRICO

1.1. Drones

Un dron se lo puede definir como un vehículo aéreo no tripulado (VANT, o su nombre en inglés UAV), capaz de mantener un nivel de vuelo controlado y sostenido (Salvador, 2016, p.10). Pueden estar controlados remotamente desde tierra por operadores o ser autónomos siguiendo una trayectoria previamente definida (Iglesias, 2016, p.18).



Figura 1-1: Estructura común en un UAV
(Vehículo Aero no Tripulado)

Fuente: <http://us.yuneec.com/hd-racer>

Estos vehículos no tripulados tuvieron un origen militar, eran manejados mediante radio control y se los usaba para practica de tiro, para las unidades antiaéreas del ejercito: Pero con el pasar de los años se fueron implementando sistemas complementarios como cámaras de video las cuales permitían realizar reconocimientos de terrenos; con el avance de la tecnología estos fueron sufriendo cambios muy grandes y rápidos, los cuales permitieron que los UAV sean capaces de realizar otro tipo de tareas. Actualmente, el uso comercial de UAV ha tenido un gran crecimiento, debido a la implementación de diferentes tecnologías, las cuales los hacen capaces de realizar tareas sin necesidad de tener una persona controlándoles, esto ha permitido que esta industria crezca y alcance gran popularidad dentro de un gran número de actividades.

1.1.1. Tipos

Existen varios tipos o clasificación entre las más importantes podremos citar:

1.1.1.1 Según su uso

1.1.1.1.1. Militares

Son aquellos de uso exclusivo militar que sirven para espionaje, reconocimiento o equipados con armas para realizar ataques. Generalmente estos están equipados con sistemas de alta tecnología que permiten que su radio de acción sea muy grande, además que alcanzan grandes velocidades, y posean gran precisión en su localización.

1.1.1.1.2. Civiles

Son aquellos que usan las personas para realizar varias actividades desde simple juguetes, tomas de imágenes hasta aplicaciones áreas como la agricultura, ingeniería, cartografía, etc. Estos han tenido una gran acogida, ya que estos permiten tomar fotos y realizar videos, desde una vista aérea logrando así una experiencia nueva para los usuarios. Además, por estas mismas razones, su uso en varias actividades, que necesiten una recolección de datos e información.

1.1.1.2. Según su diseño

1.1.1.2.1. Ala Fija

Son aquellos cuyas alas se encuentran unidas con el resto de elementos de la aeronave y no poseen movimiento (Santana, 2017, p.9), es decir son parecidos a los aviones.

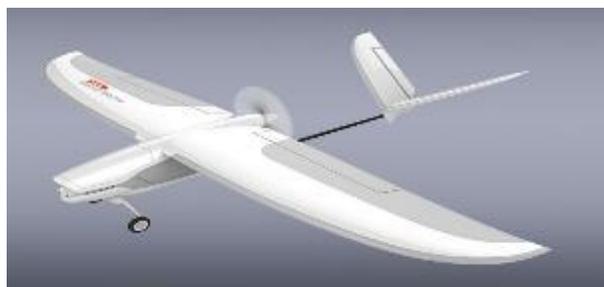


Figura 2-1: Modelo básico de un UAV de ala fija

Fuente: <http://us.yuneec.com/firebird-fpv>

Este tipo de drones sirven para desarrollar trabajos en los cuales se deban recorrer grandes distancias a velocidades y altura mayores, por lo que su maniobrabilidad no es muy buena, entre sus principales aplicaciones son en el campo militar, y cartografía.

1.1.1.2.2. *Ala rotatoria*

Son todos aquellos que sus alas giran alrededor de un eje, consiguiendo sostenerse en el aire (Santana, 2017, p.10.). dentro de estos se encuentra cada multirotor que son aquellos que poseen 3 o más rotores. Dentro del ámbito comerciales los más utilizados son lo de 4 rotores llamados quadcopteros.



Figura 3-1: 020Modelo UAV de ala rotatoria
“Drone H529 de Yunecc”

Fuente: <http://commercial.yuneec.com/comm-en-h520-overview>

1.1.2. *Quaqcopter*

Es el modelo más común que se encuentra hoy en día, en estos, dos hélices giran en sentido horario y las otras dos en sentido antihorario para generar una fuerza de empuje que supere a la fuerza de sustentación (Serrano & Pérez, 2016, p.37). Este tipo nos permite una maniobrabilidad mayor, además, de darnos la posibilidad de permanecer estáticos en el aire.

1.1.2.1. *Partes de un Dron*

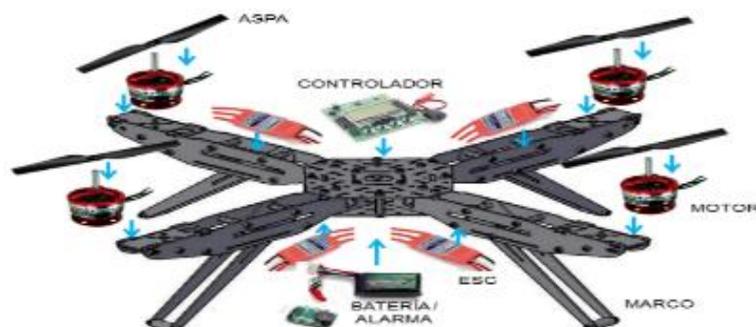


Figura 4-1: Esquema general de un UAV con sus elementos principales

Fuente: http://4.bp.blogspot.com/-c2lLr1396CE/Vn7w0AV_wZl/AAAAAAAAADHo/OL1sRN30ppY/s1600/partes%2Bdrone%2Bbasico.png

1.1.2.1.1. Estructura

Es el componente central, que conforma el cuerpo del dron, siendo donde se alojarán los distintos componentes mecánicos, eléctricos y electrónicos de la aeronave (Iglesias, 2016, p.22). La forma de esta estructura dependerá del tipo de dron que se vaya a implementar y el material deberá ser resistente y lo más ligero posible.

1.1.2.1.2. Motores Y Hélices

Los más utilizados son de tipo brushless DC. Son motores que presentan una reacción rápida y son alimentados por una señal AC que se aplica a los electroimanes para su funcionamiento (Iglesias, 2016, p.22). Son los componentes que permitirán al dron mantenerse suspendido en el aire, junto con las hélices giran a grandes velocidades para lograr la fuerza suficiente para elevar y mantener el dron en el aire.

1.1.2.1.3. ESC Controlador electrónico de velocidad

Los ESC regulan la potencia eléctrica para lograr el giro de los motores con agilidad y eficiencia (Serrano & Pérez, 2016, p.40), estos nos permiten controlar la velocidad de giro de los motores, convirtiendo la corriente continua a una señal AC, con lo cual podremos lograr el movimiento del dron.

1.1.2.1.4. Control de vuelo

Es el componente encargado de censar y controlar todo lo que sucede con el dron, dentro de este se encuentran los componentes necesarios para el control del dron como son: acelerómetro, giroscopio, magnetómetro, GPS, el microcontrolador (Serrano & Pérez, 2016, p.41), esta es la parte principal del dron ya que es la encargada de realizar el procesamiento de los datos y la toma de decisiones para el correcto funcionamiento.

1.1.2.1.5. Radio Control

El radio receptor es el encargado de recibir la señal desde el control remoto y transformarla en datos para que el controlador de vuelo ejecute la instrucción que envía el piloto (Serrano & Pérez,

2016, p.41). Es el dispositivo que nos permite una comunicación entre el dron y el piloto que lo controla.

1.1.2.1.6. Batería

Son las encargadas de proporcionar al dron de energía eléctrica para su funcionamiento. Estas unidades suelen utilizar baterías de Polímero de litio, (Li-Po) ya que ofrecen una mayor densidad de carga a las tradicionales de ion-litio, un tamaño más reducido y menor peso (Iglesias, 2016, p.23).

1.1.3. Programación y desarrollo de aplicaciones

Ya se dispone de APIs (interfaz de programación de aplicaciones) para que los desarrolladores programen aplicaciones y servicios relacionados con drones, así también se puede realizar en distintos lenguajes de programación. Las compañías 3D Robotics y DJI, han lanzado Kits de desarrollo de aplicaciones (SDKs) y APIs para desarrolladores. A continuación, detallamos algunas de los sistemas y lenguajes de programación usados en el desarrollo de aplicaciones para vehículos aéreos no tripulados.

1.1.3.1. DroneKit,

3D Robotic dispone de una API en Python flexible y abierta para el desarrollo de aplicaciones que se ejecutan desde el ordenador de a bordo y comunican con el controlador de vuelo ArduPilot con un enlace de baja latencia.

La API se conecta con los drones a través del protocolo de comunicación MAVLink, Gracias a este protocolo, la API tiene acceso a los datos de la telemetría a través de Bluetooth, Wi-Fi o el sistema de radio, a los parámetros del dron y también permite controlar el movimiento y las operaciones del propio aparato no tripulado.

Este SDK permite a los desarrolladores, entre otras funciones

- Desarrollar aplicaciones que mejoran el vuelo en piloto automático
- Proporcionar características inteligentes al dron como visión por ordenador, planificación de ruta o el modelado 3D

DroneKit se puede instalar en sistemas operativos Linux, Windows y Mac OS X y funciona tanto en ordenadores móviles como de escritorio.

1.1.3.2. *Las APIs de DJI*

El primero de los SDK son los que permiten programar el sistema de control a bordo, permite a los profesionales supervisar y controlar el dron desde cualquier sistema conectado directamente con el piloto automático de la aeronave. Un desarrollador puede crear un algoritmo que sea capaz de controlar el dron usando cualquier tipo de lenguaje de programación y para distintos sistemas operativos.

Las características más importantes:

- Comunicación confiable, protocolo para evitar pérdidas de datos.
- Sistemas de control flexible diferentes métodos de control del vuelo, desde la información de su posición hasta su velocidad.
- Monitorización de datos configurable los datos de vuelo se pueden obtener de forma personalizada.

Autonomía de vuelo el modo de control de vuelo y la información obtenida en ruta están diseñados para facilitar una navegación autónoma (BBVA, s. f.).

1.1.3.3. *ROS*

Robot Operating System, es un sistema flexible para el desarrollo de software para robots. Es un conjunto de herramientas, bibliotecas, que tienen como objetivo simplificar la tarea de crear un comportamiento complejo y robusto en una amplia variedad de plataformas robóticas. (ROS.org, s. f.). Dentro de los componentes principales tenemos:

- Infraestructura de comunicaciones:

Este es el nivel más bajo de ROS, donde se ofrece una interfaz que permite el paso de mensajes para la comunicación entre los diferentes dispositivos que componen el sistema robótico. Dentro de este se puede realizar grabaciones y reproducción de mensajes, llamadas a procedimientos remotos, contribución entre tareas, y en si todos los protocolos necesarios para poder realizar una buena captación y distribución de la información y comandos de ejecución.

- Características del Robot:

Dentro de estos componentes, ROS también proporciona bibliotecas útiles que permiten configurar de manera fácil y rápida, algunas características principales. Dentro de estas tenemos:

- Mensajes estándar de Robot:

Aquí es donde se proporciona ayuda con la lectura de diversos formatos de datos, que se adquieren con diferentes tipos de sensores, para que el usuario pueda conectar y leer sensores de diferentes tipos, y con diferentes formatos de salida. Esta herramienta lo que hace es interpretar los diferentes formatos para acoplarlos con el resto del sistema ROS, para poder usar los valores sin ningún tipo de inconveniente.

- Robot Geometry Library:

Esta herramienta nos permite conocer la posición de las articulaciones de los robots, es decir esta herramienta nos permite administrar datos de transformación de coordenadas para los robots con más de 100 grados de libertad, (ROS.org | Componentes principales, s. f.).

Esta librería nos ayuda mucho para conocer y analizar la cinemática y dinámica de un robot, dado que esto es muchas veces complejo y tedioso, esta herramienta nos facilita mucho ese trabajo ahorrándonos tiempo valioso, que puede ser destinado a otra etapa de desarrollo.

- Robot Descripción Idioma:

Esta herramienta nos permite modelar nuestro robot, en el sentido que nosotros diseñamos la mecánica del robot y esta herramienta lo describe de una manera tal que pueda ser interpretado por una máquina. Podemos modelar, según nuestras necesidades, y no estaremos limitados a los modelos estándar. Con esto podemos describir desde la longitud de las extremidades hasta el tamaño de las ruedas, ubicación de los sensores y el aspecto visual de nuestro robot.

- Estimación de la posición, localización y navegación:

Ros, proporciona una herramienta que nos permite estimar la posición, orientación de nuestro robot. También permite la construcción de un mapa e incluso la navegación móvil.

Esta herramienta es una muy útil, ya que, es uno de los problemas que se necesita resolver al momento de poner en marcha un robot.

- Herramientas:

Dentro de estas encontramos un gran grupo que ayudan mucho dentro del proceso de desarrollo y programación de aplicaciones para robots. Alguna como la Línea de comandos la cual nos

permite interactuar con el robot sin la necesidad de una GUI. Otra importantes es la raíz, que nos proporciona una visualización tridimensional de varios tipos de sensores. Rqt, basada en Qt para desarrollar interfaces gráficas para un robot, donde se puede crear interfaces personalizadas. Entre otras herramientas que ayudan al desarrollo de prototipos de manera sencilla y rápida.

1.1.4. Control desde una estación en tierra

El sistema de control que maneja el usuario es aquel encargado de llevar a cabo las tareas de planificación y control de la misión que el VANT debe ejecutar, además debe ejecutar un control de la carga útil y distribución de la información a usuarios exteriores o finales. Además, la estación posee sistemas requeridos de comunicación y enlace de datos, debido a que éstos son los encargados para acceder a la tarjeta de control y a los elementos electrónicos que componen el dron.

Funciones

- Establecer la ruta o misión. – la trayectoria se enfoca principalmente en crear la ruta que el dron va a recorrer y las tareas que se deben llevar a cabo.
- Comunicarse con el VANT. – Es necesario una comunicación constante y en tiempo real con la finalidad de enviar tareas que deberán ejecutarse y recibir información sobre el estado del vehículo como datos de velocidad, altitud y posición actual del dron.

Existen varias alternativas dentro de las cuales se puede mencionar.

1.1.4.1. Mission Planner



Figura 5-1: Trayectoria que el UAV Mission Planner

Fuente:http://ardupilot.org/planner/_images/mission_planner_screen_flight_plan.jpg

Mission Planner es una estación de control en tierra para vehículos aéreos no tripulados. Es compatible con Windows solamente. Puede utilizarse como una utilidad de configuración o como un suplemento de control dinámico para su vehículo autónomo.

Características principales:

- Cargar el firmware en el piloto automático (APM, PX4) que controla el vehículo
- Configurar y ajustar el vehículo para un rendimiento óptimo
- Planificar, guardar y cargar misiones autónomas en el piloto automático
- Descargar y analizar los registros de misión creados por el piloto automático
- Con el hardware de telemetría adecuado se puede:
- Controlar el estado del vehículo mientras está funcionando
- Llevar un registro de telemetría que contiene mucha más información
- Ver y analizar los registros de telemetría
- Utilizar el vehículo en FPV

1.1.4.2. QGround Control



Figura 6-1: Planificación de trayectoria, QGround Control

Fuente: <https://www.rhinohawk.com/wp-content/uploads/2017/08/qgroundcontrol-768x436.png>

QGround Control proporciona un control sobre el vuelo y la configuración del vehículo, con piloto automático PX4 o ArduPilot. Proporciona un uso sencillo y directo para principiantes y también ofrece funciones de gama alta para expertos. (Dronecode.org, s. f.)

Características principales:

- Configuración completa de vehículos con PX4 y ArduPilot,
- Soporte de vuelo para vehículos con PX4 y ArduPilot, o cualquier piloto automático con comunicación MAVLink.

- Planificación de misiones para vuelo autónomo.
- Visualización del mapa mientras se muestra la posición del vehículo.
- Transmisión de video con superposición de instrumentos en la pantalla.
- Soporte para administrar múltiples vehículos.
- Se ejecuta en Windows, Mac OS X, plataformas Linux, dispositivos IOS y Android.

1.1.5. Aplicaciones



Figura 7-1: Envío de paquetes mediante un dron.

Fuente: <https://www.engadget.com/de/2016/12/22/frankreich-auch-dpd-setzt-auf-lieferung-per-drohne/>

Los vehículos aéreos no tripulados gracias a su potencial y la tecnología que se pueden implementar en ellos, hacen que estos equipos se conviertan en una gran herramienta para la realización de una gran cantidad de actividades las cuales el hombre consideraría repetitiva, difícil de realizar o ambos, en una amplia cantidad de aplicaciones en diversos campos. Algunos de estos espacios pueden ser:

1.1.5.1. Inspección Aérea de Líneas Eléctricas

Se usa drones para la revisión visual de las torres de transmisión, equipados con cámaras HD que pueden captar con más detalle los posibles problemas que se presentan, además se puede realizar un registro más completo de toda la estructura y así también con imágenes térmicas poder determinar lugares con pérdida de energía y posibles incendios (Drones: General aspects and social applications. Pinzón, Ortega, & Rosa, 2016, P.6).

1.1.5.2. Transporte de Materiales

Existen varios proyectos en los cuales se pretende transportar mercadería y otros objetos, usando drones, por ejemplo, transporte de insumos médicos en la zona rural, este proyecto busca mejorar

la respuesta de atención en centros de salud, en zonas rurales donde la topografía hace difícil que llegue la atención médica primaria (Drones: General aspects and social applications, Pinzón et al., 2016, P.7).

1.1.5.3. Agricultura de Precisión

Esta es una rama en la cual el uso de los drones ha tenido gran impacto ya que existe varios proyectos y trabajos que se realizan en este campo. La principal problemática que se quiere resolver aquí, es el desconocimiento del estado actual del cultivo en puntos determinados haciendo que los agricultores tomen medidas que no están acorde a la realidad de ciertas aéreas, que traerán con el tiempo como resultado perdidas y gastos superfluos (González, Amarillo, Amarillo, & Sarmiento, 2016b). Con el uso de drones se puede crear aplicaciones para detectar mala hierba, zonas con falta de agua, tasa de crecimiento, detección de plagas entro otras aplicaciones que servirán para mejorar la producción agrícola.

1.1.5.4. Drones para sustitución militar

La utilización de estos equipos de vuelo no tripulados ha tenido una gran aceptación en muchos campos, uno de ellos en el ámbito militar se utilizan estos equipos debido a que representan una inversión significativamente menor en comparación con equipos de vuelo militares tradicionales y adicional a ello, tienen una gran preferencia debido a que esta nueva técnica de llevar a cabo una misión resulta en gran parte, segura para la tripulación que anteriormente poseía la obligación de transportarse en la aeronave.

Como dato interesante el ejército de los Estados Unidos posee una flota de 8000 drones aproximadamente, éstos dispositivos tienen una gran aceptación en el uso en misiones debido a que al no poseer un tripulante las misiones de espionaje y ataque presentan un menor riesgo para las personas que forman parte de la fuerza a la que representan, son utilizados para atacar lideres terroristas, esta técnica de uso de drones tomo un incremento significativo en la administración de Obama, otros países como Reino Unido entre otros de primer mundo, en síntesis el uso de drones es la nueva generación muy cercana en el uso de defensa y ataques militares.(Jaramillo, 2018, p.1).

1.1.5.5. Vista de águila “Fotoperiodismo”

Con la ayuda de los UAV el periodismo ha alcanzado un nuevo nivel en la difusión de información a la sociedad debido a que cada vez las imágenes de eventos de interés social han ido incrementando su precisión y fiabilidad, pues gracias a estos equipos la obtención de imágenes de ángulos distintos en cada evento que se suscite es una misión que puede llevarse a cabo, la

seguridad de los periodistas se ha incrementado a la hora de captar imágenes de difícil acceso o en desastres naturales, por otro lado las fuerzas de la ley y el orden se han visto beneficiados debido a la ayuda de estos equipos en la misión de control de tráfico de drogas y personas. (Villalobos, 2015, p.1).

1.1.5.6. Aplicación en la seguridad

En los últimos años la utilización de los UAV han tenido una aceptación en América Latina, una gran aceptación ha sido en las áreas de seguridad y monitoreo para la facilitar y brindar una mejor atención y cuidado a la ciudadanía, el uso de estos equipos ha sido destinada en la vigilancia aérea, en el monitoreo de construcciones ilegales, control de zonas públicas, supervisión de áreas afectadas por contaminación entre otros usos, su éxito se debe a la accesibilidad y alto nivel de seguridad que pueden llegar a ofrecer, el único inconveniente es el que de acuerdo al país que se encuentre, se debe respetar las normas de uso y límites de acuerdo a las políticas públicas latinoamericanas.(Gomis-Balestreri & Falck, 2015, p.2).

En general el uso de los vehículos aéreos no tripulados en diferentes campos, se ha incrementado gracias su accesibilidad y gran diversidad de uso, debido a que se pueden realizar tareas que en el pasado conllevaban un tiempo y esfuerzo considerablemente alto si pretendíamos una alta fidelidad en los resultados que obtendríamos. Entre las muchas aplicaciones que se pueden atribuir a los drones podemos decir que en la gran mayoría de campos lo que se pretende es alcanzar una nueva expectativa en cuanto se hace referencia a la fotografía y video aéreo; esto se ha vuelto muy popular tanto para personas aficionadas a estos atractivos equipos de vuelo como para profesionales y experimentadas que tratan de obtener fotos videos de lugares desde un punto de vista totalmente nuevo o de acceso remotamente inasequible.

1.1.6. Kits de desarrollo de drones

1.1.6.1. Parrot SLAMdunk

El Parrot SLAMdunk es un kit de desarrollo para aplicaciones de navegación autónoma, evasión de obstáculos y mapeo 3D, para drones y otro tipo de robots, esta plataforma integra aplicaciones de software avanzadas para algoritmos SLAM, lo cual permite que el vehículo pueda realizar un mapeo en 3D. Este kit está diseñado para realizar investigaciones y desarrollo de aplicaciones y sus características son:

- Procesador NVIDIA Tegra K1
- RAM 2 GB DDR3

- Memoria de 16 GB
- Una cámara estéreo con resolución 1500 x1500 a 60 FPS (Cuadros por segundo)
- IMU (Unidad de medición Inercial)
- Sensor ultrasónico 15cm a 15 m
- Magnetómetro
- Barómetro

1.1.6.2. Intel Aero Ready to Fly

Es una plataforma de desarrollo de un vehículo aéreo con todo lo necesario para volar, que funciona con la placa Compute Board y equipada con capacidades de visión de profundidad y ejecutada bajo el sistema operativo Linux de código abierto. Este kit está dirigido a investigaciones y desarrollo que deseen un camino rápido para probar aplicaciones creadas para drones.

Este kit está diseñado para probar y desarrollar aplicaciones, y sus características son:

- Intel Compute Board, con procesador Intel Atom x7-z8750
- RAM 4GB LPDDR3
- Memoria 32 GB eMMC
- Ranura para MicroSD
- Conector M.2 para SSD
- Intel Dual Band Wireless
- Altera Max 10 FPGA
- Kit de cámaras RGB 8MP y VGA monocromático
- Cámara Intel RealSense R200
- Controlador de vuelo Intel Aero, STM32
- IMU, Magnetómetro y Barómetro
- Fuselaje de fibra de carbono
- GPS y brújula
- Placa de distribución de energía
- 4 ESC y 4 motores
- En relación al lenguaje de programación, es básicamente el preferido por el usuario, dentro de los cuales los más usados son:
 - Programación en C/C++
 - Programación en Python
 - ROS

1.2. Cámara Xioma

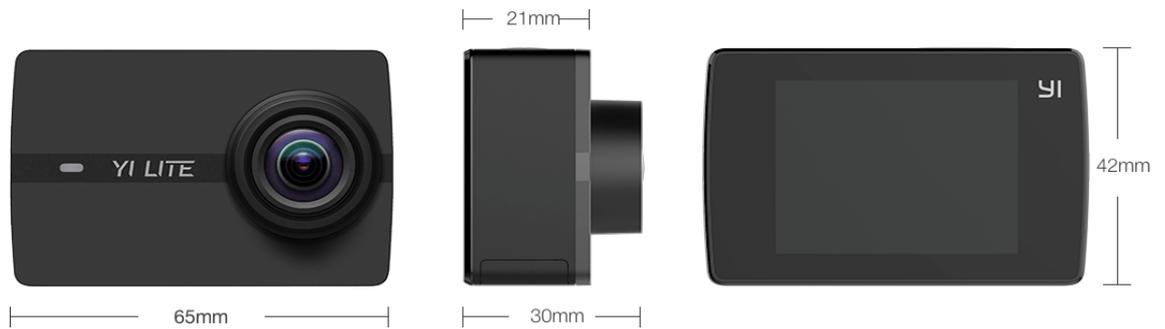


Figura 8-1: Dimensiones frontal y lateral de la cámara Xioma YI Lite

Fuente: <https://www.yitechnology.com/yi-lite-specs>

La cámara deportiva Xiaomi YI Lite Action Cam, es una cámara de gama media que permite obtener imágenes de gran calidad de la zona en la cual se realizó la trayectoria del sobrevuelo.

La tabla siguiente contiene detalles principales de la cámara:

Tabla 1-1: Descripción de las características principales que posee la cámara Xiaomi YI

Característica	Descripción
-Dimensiones	3 x 6,5 x 4,2 cm
-Peso	70 g
-Chip	Hisilicon Hi3556
-Sensor	Sony Exmor de 16 MP
-Gran Angular	120°
-Apertura Focal	2.8
-Video	4K (20 fps), 1080p (60 fps)
-Fotografía	JPG 16 MP
-Autonomía de la batería	130 minutos
-Conectividad	Wi-Fi, Bluetooth
-Funciones	4 modos de foto, time lapse, foto ráfaga

Fuente: (YI LITE Action Camera Specs, YI Technology, YI LITE ACTION CAMERA, 2017, pp2-6)

Para la captura de las imágenes dentro de la misión se utilizó la función temporizador, de 3 segundos, la cual nos permite obtener imágenes dentro de un intervalo de tiempo adecuado, para lograr tener imágenes con un solapamiento entre el 50 % y 80 %. La resolución de las imágenes es de 16 Mpx, logrando cubrir aproximadamente 15 metros cuadrados por cada 10 metro de altura.

1.3. Sistema anticollisiones

Los drones son vehículos no tripulados, pero no autónomos, esto quiere decir que necesitan de un operador que los dirija y de indicaciones de sus movimientos desde la distancia. Para lograr que un dron sea autónomo necesita de un sistema anticollisiones el cual nos permita determinar si delante de la trayectoria que sigue existe un obstáculo, para poder evitarlo y seguir con la ruta establecida.

Un sistema anticollisiones es un sistema de seguridad primaria, que permite al UAV detectar obstáculos que atenten contra sí (Arreguín, Soto, & Arreguín, 2017, p.94).

Existen hoy en día muchos drones comerciales que poseen sistemas para evitar obstáculos los cuales utilizan diferentes clases de sensores de proximidad que nos permiten determinar la presencia de algún objeto, Al momento de hablar de un sistema anticollisiones podemos mencionar que existen desde los más sencillo que detectan un obstáculo y se detienen para evitar el choque, esperando la orden del piloto para continuar, hasta los más avanzados los cuales crean un mapa 3D de la zona en tiempo real para poder así navegar autónomamente.

A continuación, se detalla algunas de las tecnologías utilizadas para realizar sistemas anticollisiones.

1.3.1. Sistemas basados en Radio Frecuencia



Figura 9-1: Tarjeta RFID con Antena

Fuente:<https://i2.wp.com/peihispano.com/contenido/uploads/2013/12/Capture.jpg>

Estos sistemas se basan en Radio Frecuencia, el cual está compuesto por tarjetas RFID colocadas en varias partes del terreno por el cual se va a desplazar el dron, este sistema toma datos de las tarjetas las cuales nos proporcionan la ubicación de obstáculos en el campo. La ventaja de estos sistemas es que no requiere equipo auxiliar de visualización, es ligero y económico. Pero su principal desventaja es que debemos colocar una tarjeta RFID por cada posible obstáculo en la trayectoria del dron.

1.3.2. Sistemas basados en Reconocimientos de Patrones



Figura 10-1: Principio de funcionamiento de Intel RealSense

Fuente:https://www.researchgate.net/publication/313468896_De_ficcion_a_realidad_drones_y_seguridad_ciudadana_en_America_Latina

Estos sistemas incorporan cámaras frontales, estereoscópicas, que permiten capturar imágenes que luego serán procesadas para calcular distancias y posiciones de los posibles obstáculos para el vehículo, la ventaja de este tipo de sistemas es que su distancia de detección es mayor, además que permite reconocer y diferenciar los tipos de obstáculos que se presentan. La desventaja de estos, está en que solo pueden detectar objetos en una única dirección, generalmente la parte frontal, siendo los laterales un punto ciego, esto se puede corregir añadiendo más cámaras, aumentando así la otra gran desventaja, que es, la necesidad de usar algoritmos complejos lo que radica en una latencia mayor en el procesamiento de la información.

1.3.2.1. Intel RealSense en detección de obstáculos

“La Tecnología Intel RealSense utiliza distintas tecnologías de detección para lograr la percepción de profundidad, las imágenes en 3D, el mapeo interior y el seguimiento de características. Desde realidad virtual hasta el desarrollo de robótica y la creación en 3D para consumo, el uso de la tecnología Intel RealSense lidera la industria y ofrece creatividad y libertad sin límites” («Intel® RealSense™ Data Ranges | Intel® Software», s. f.).

Esta tecnología comprende cuatro importantes modos de comunicación: las manos, el rostro, la voz y el entorno que rodea al usuario. Todo esto se lo puede llevar a cabo con la cámara 3D Intel RealSense, que puede comprender e interpretar la profundidad.

1.3.2.2. Especificaciones de la cámara R200

El módulo R200 posee tres cámaras una RGB y dos Infrarrojas estereoscópicas de profundidad, que con la ayuda de un proyector laser, puede escanear en 3D, la distancia puede ser entre 0,5 y

3,5 m en interiores y sobrepasa los 5 m en exteriores, pudiendo alcanzar hasta los 10 m dependiendo de las condiciones ambientales, estado que se encuentra el módulo y la iluminación.

1.3.2.3. Biblioteca de prevención de colisiones

Collision Avoidance Library (Coav) es un entorno para probar y evaluar estrategias de prevención de colisiones, desarrollada para drones.

Esta proporciona la infraestructura para desarrollar y realizar pruebas de nuevos algoritmos de evitación y prevención de colisiones (Intel Corporation, "collision-avoidance-library", 2016/2018, p.1). La biblioteca incluye un entorno de pruebas automatizado y herramientas para ejecutar la implementación en un entorno simulado, dentro del Sistema Operativo Ubuntu, junto con ROS.

1.3.2.4. Estrategias de evasión y detectores

Tabla 2-1: Características fundamentales en las técnicas de evasión

TÉCNICAS DE EVASIÓN		
Estrategias	SHIFT EVITACIÓN	STOP AVOIDANCE
	Desplaza el vehículo a la izquierda hasta sortear por completo el obstáculo y continuar con la misión establecida.	Cuando encuentra un obstáculo, detiene el dron generando una interrupción en la misión establecida.
Detectores	OBSTÁCULOS DE IMAGEN DE PROFUNDIDAD	HISTOGRAMA POLAR DE IMAGEN DE PROFUNDIDAD
	Con la lectura de objetos binarios de gran tamaño "Blobs" que son datos proporcionados por el sensor, si están en una distancia segura se guardarán los datos en una etiqueta, por el contrario, en otra etiqueta y se creará un obstáculo con el blob que tiene la menor distancia.	El sensor apunta en la dirección de movimiento del vehículo, si se capta un objeto, la distancia se divide en segmentos los cuales serán comparados bajo un umbral determinado y si estos datos están por debajo del nivel aceptable, se crea un obstáculo centrado en la lectura mínima de distancia

Realizado por: Tacuri & Olmedo, 2018

1.3.2.5. Algoritmo de detección y evasión de obstáculos

Para realizar un algoritmo capaz de evaluar obstáculos y evitarlos es necesario realizar un algoritmo que se compone básicamente en crear un objeto del vehículo para la comunicación con el vehículo real, crear un objeto sensor el cual es el sensor real del vehículo, con el sensor se

deberá plantear una instancia del detector que interpreta los datos del sensor y al final con esos datos se plantea una estrategia de evasión.

A continuación, se plantea el esqueleto base para la elaboración de un algoritmo de detección y evasión:

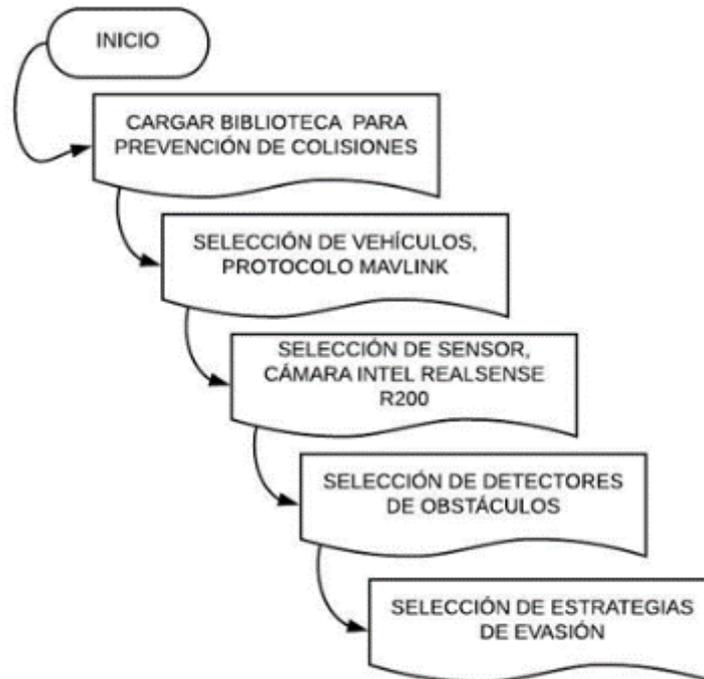


Figura 11-1: Diagrama de flujo base sobre detección y evasión

Realizado por: (Tacuri & Olmedo, 2018)

1.3.3. *Sistemas basados en sensores ópticos-infrarrojos*



Figura 12-1: Detección de obstáculos con sensores infrarrojos

Fuente: <http://www.unmannedsystemstechnology.com/wp-content/uploads/2015/03/Panoptes-Systems-eBumper4-Drone-Obstacle-Avoidance-System.jpg>

Estos sistemas se basan en sensores infrarrojos, los cuales emiten un rayo de luz, laser, que puede alcanzar distancias de 20 cm a 14 m para detectar obstáculos en la trayectoria de vuelo (Arreguín et al., 2017, p.94-97). La principal ventaja de esta clase de sistemas es que son muy pequeños y ligeros, además, se adaptan a casi cualquier clase de dron, el tiempo de respuesta es mejor ya que no necesita algoritmos complejos para su lectura. La desventaja de esta clase de sistemas es que los sensores infrarrojos no son óptimos en días nublados y que la señal puede ser absorbida por superficies en el entorno (Arreguín et al., 2017, p.94-97).

1.3.4. Sistemas basados en sensores ultrasónico



Figura 13-1: Dji Matriz 100, con sistema ultrasónico

Fuente:https://d2otfaypcda2dg.cloudfront.net/blog/wpcontent/uploads/2015/08/DJI-Matrice-100-unboxing_frame-constructed-ultrasonics-base-1024x684.jpg

Estos sistemas utilizan sensores ultrasónicos que pueden detectar obstáculo mediante ondas de sonido de alta frecuencia las cuales son emitidas y al chocar contra un objeto rebotan u vuelven al sensor, para determinar la distancia debemos realizar un cálculo externo, sin embargo, en la actualidad ya existen módulos que integran una unidad de procesamiento que nos da como resultado la distancia calculada. La ventaja de este tipo de sistemas es que pueden detectar objetos a distancias grandes mayores a los 10 m y que se pueden montar varios de ellos para lograr un sistema de detección en varias direcciones, así mismo la desventaja es principalmente el ruido, producido por sonidos y vibraciones en el ambiente y el propio dron, así como la superficie que se pretende detectar.

1.4. Procesamiento de imágenes

“Es la ciencia de manipulación de imágenes usando computadores para realizar procedimientos específicos según las aplicaciones y requerimientos del usuario, tales como: filtrado, recorte, segmentación, compresión y reconocimiento” (Jácome, Torres, & Araujo, 2017).

Como lo mencionan los autores antes citados el procesamiento no es más que la manipulación de las imágenes en una computadora con el objetivo de extraer de ella información de interés para el usuario. Esta manipulación se lo realiza en diferentes entornos con softwares específicos para este propósito.

Para este proyecto se trabajará principalmente con dos técnicas de procesamiento y visión por computadora que son: Stitching y Segmentación.

1.4.1. Image-Stitching

El stitching o un significado en español costura, no es más que el proceso para la unión de varias imágenes para formar una imagen más grande o panorama.



Figura 14-1: Unión de imágenes con Stitching

Fuente: <http://winbizsolutionsindia.com/wpcontent/uploads/2018/02/image-stitching.jpg>

Para realizar un procesamiento de este tipo se usa generalmente técnicas de visión por computadora y procesamiento de imágenes como detección de puntos de interés, descriptores invariantes locales, coincidencia de puntos clave, RANSAC y perspectiva de deformación.

Para poder realizar la unión de imágenes primero necesitamos encontrar un conjunto de puntos distintivos, luego definir una región alrededor de cada punto significativo, luego extraer y normalizar la región y luego calcular un descriptor local desde la región normalizada.

Para poder realizar una correcta unión de imágenes hay que tomar en cuenta algunos parámetros como la estabilidad de la cámara, es decir que si la cámara está en un punto fijo sin movimiento. La secuencia de imágenes que se requiere, para un buen resultado debemos tener una cantidad razonable de imágenes y solapamiento, se recomienda por lo menos 50 – 80 %, Esto servirá para superar la distorsión de lentes. Otro factor a tomar en cuenta es la exposición, en lo posible se

trata de mantener una exposición constante entre todas las imágenes, pero sabemos que no es posible por esa razón lo mejor es tener unas transiciones suaves entre imágenes.

Para realizar un algoritmo Stitching, se debe considerar:

- Intensidad variable del ambiente.
- Intensidad variable y el contraste entre imágenes.
- Distorsión de la lente, acerico, barril y ojo de pez. Distancia focal entre otros.
- Dinámica / movimiento en la escena, causa efecto fantasma.
- Desalineación entre imágenes.
- Resultado agradable a la vista, donde se elige el tamaño del panorama generalmente aspecto 10:3.

Un algoritmo Stitching consta generalmente de las siguientes etapas:

1.4.1.1. *Detector de puntos clave*

Los detectores de puntos clave es una técnica de visión por computadora que sirve para extraer rasgos de interés dentro de una imagen. Un punto de interés es un punto que posee una posición definida dentro de la imagen y puede ser detectada de manera fácil sin importar si cambian las condiciones de la imagen.

Dentro de estos podemos mencionar los siguientes:

- Harris Corner Detector: este es uno de los detectores más usados actualmente, esto se debe a su gran invariancia ante la escala, rotación, cambios de iluminación y ruido. El punto característico viene dado por aquel pixel que posea los valores elevados en el cálculo de la matriz de segundo orden (Ballesta et al., 2010, p.69). Este detector se basa en una matriz $\mathbf{C}(\mathbf{x}, \mathbf{y})$ con una subventana de $\mathbf{p} \times \mathbf{p}$, que se calcula para cada punto.

$$\mathbf{C}(\mathbf{x}, \mathbf{y}) = \begin{pmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{pmatrix}$$

Donde: I_x, I_y son los gradientes de la imagen en dirección horizontal y vertical respectivamente.

Donde la función de autocorrelación $R = \lambda_1 \lambda_2 - k(\lambda_1 + \lambda_2)$, Tendrá un pico si ambos valores son altos, entonces se dice que desplazamientos en cualquier dirección producirán un aumento significativo, indicando que se trata de una esquina. Siendo λ_1 y λ_2 los valores propios de la matriz (Ballesta et al., 2007, p.3).

- Harris Laplace: En este detector, los puntos son invariantes a rotación y escala. Los puntos son detectados mediante una función de Harris adaptada a la escala, que son seleccionadas en el espacio de escalas por un operador Laplaciano, la región de interés se da de acuerdo a la escala seleccionada, este detector detecta características locales de forma más precisa espacialmente, que son mucho mejores para ser usadas en reconstrucción y localización (Ballesta et al., 2007, p.3).
- Susan: este detector aplica una máscara circular sobre un pixel. En este caso se compara la similitud de los pixeles vecinos en el círculo, determinando en función de la cantidad de pixeles similares con el pixel central situado dentro de la máscara si es no un punto característico (Ballesta et al., 2010, p.69). Se usa principalmente en la detección de objetos.

1.4.1.2. Descriptores de Puntos de Interés

Un descriptor obtiene de cada punto de interés, información sobre la región que lo rodea, para que la misma estructura pueda ser reconocida si es encontrada en otra imagen (Albán, 2014, p.71-73). Podemos mencionar:

Sift

Este algoritmo es tanto un detector de puntos como un descriptor de puntos de interés, realiza la extracción de características distintivas de imágenes en escala de grises, permitiendo así reconocer la imagen dentro de un grupo o incluso dentro de una imagen de mayor tamaño. (Flores & Braun, 2011, p.1).

Este algoritmo se lo realiza de la siguiente manera:

- Detección de extremos en el espacio-escala

El objetivo es hallar en el espacio-escala, de la imagen, puntos invariantes a las características invariantes, traslación, escalado y rotación de la misma. Estos puntos son los puntos máximos o mínimos, resultado de las diferencias Gaussianas aplicadas en el espacio-escala de la imagen.

La Diferencia Gaussiana $D(x, y, s)$ se define de la siguiente manera:

$$D(x, y, \sigma) = L(x, y, k\sigma) - L(x, y, \sigma)$$

Donde:

$$L(x, y, \sigma) = I(x, y) * G(x, y, \sigma)$$

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

Se aplica esta diferencia Gaussiana n veces, con distintos tamaños de región, es decir, aumentando la desviación estándar, buscando máximos y mínimos locales en cada iteración.

Los puntos máximos o mínimos son posibles candidatos a ser puntos clave, “Una vez que se obtiene la “pirámide” anterior, se buscan para cada “piso” de la misma extremos locales quienes se transformarán en candidatos a puntos clave” (Flores & Braun, 2011, p.2). Para considerarse como punto clave relativos se realiza una comparación entre sus vecinos debiendo cumplir: si es mayor o menor a sus 8 puntos vecinos dentro de su propio piso y a sus 9 puntos vecinos de los niveles inferior y superior.

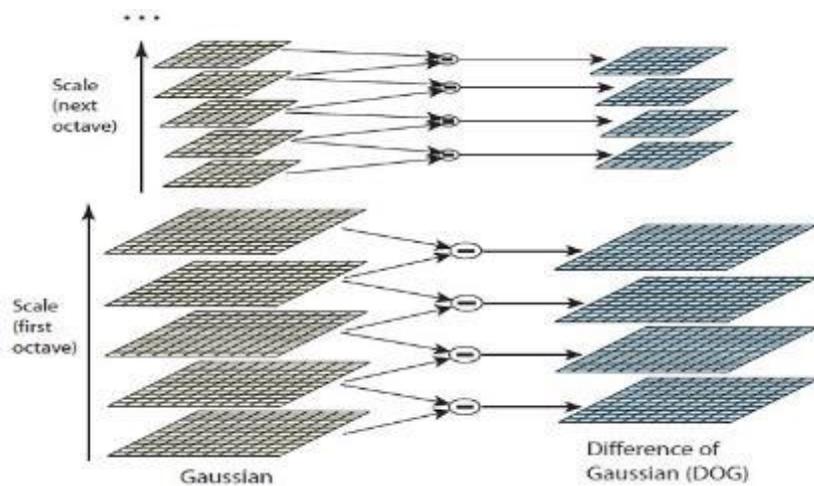


Figura 15-1: Diferencia Gaussiana, pirámide de escala de David Lowe

Fuente: <http://aishack.in/tutorials/sift-scale-invariant-feature-transform-log-approximation/>

- Localización exacta del punto clave

La detección de extremos produce muchos puntos en el espacio-escala, que son posibles candidatos, entre ellos se encuentran puntos afectados por la variación de iluminación y ruido. Se procede a realizar una estimación de la función en la diferencia Gaussiana mediante una serie de Taylor de grado 2:

$$D(x) = D + \frac{\partial D^T}{\partial x} x + \frac{1}{2} x^T \frac{\partial^2 D}{\partial x^2} x$$

Donde D y sus derivadas son evaluadas en el punto en cuestión, si este valor es menor al umbral, el punto es descartado. También es necesario descartar los puntos que vienen de una línea y no de una esquina, para ello de manera similar al algoritmo de Harris.

Esto se lo realiza como se detalla a continuación según (Flores & Braun, 2011, p.3):

Sea H la matriz Hessiana de $D(x, y, \sigma)$ evaluada en un punto del espacio-escala determinado (X_o, Y_o, σ_o) , estaremos en presencia de un borde (línea) si sus valores propios α y β son uno grande y el otro pequeño. La anterior es equivalente a trabajar con el conocimiento de:

$$T_{raza}(H) = \frac{\partial^2 D}{\partial x^2} + \frac{\partial^2 D}{\partial y^2} = \alpha + \beta$$

$$Det(H) = \frac{\partial^2 D}{\partial x^2} \times \frac{\partial^2 D}{\partial y^2} = \alpha \cdot \beta$$

Sea la $\alpha = \gamma \cdot \beta$, la condición se reduce a:

$$\frac{T_{raza}(H)^2}{Det(H)} < \frac{(r + 1)^2}{r}$$

Lowe propone un umbral de $r=10$. Se puede ver que conforme aumenta la relación r entre ambos valores propios también lo hace la relación entre el cuadrado de la traza de la matriz Hessiana y su determinante.

- Asignación de orientación

Para asegurar la invariancia a la rotación de las imágenes, se asigna una orientación a cada punto de la imagen, luego se determina el módulo de su gradiente $m(x, y)$ y la fase $\theta(x, y)$ utilizando diferencias entre píxeles:

$$m(x, y) = \sqrt{(\Delta L_x)^2 + (\Delta L_y)^2}$$

$$m(x, y) = \sqrt{[L(x + 1, y) - L(x - 1, y)]^2 + [L(x, y + 1) - L(x, y - 1)]^2}$$

$$\theta(x, y) = \tan^{-1} \left(\frac{\Delta L_y}{\Delta L_x} \right)$$

$$\theta(x, y) = \tan^{-1} \left(\frac{L(x, y + 1) - L(x, y - 1)}{L(x + 1, y) - L(x - 1, y)} \right)$$

Una vez calculado la magnitud y dirección del gradiente del punto y sus vecinos, se genera un histograma de direcciones con los valores que varían de a 10 grados ponderado por la magnitud del gradiente del punto clave en cuestión.

El mayor porcentaje indica la orientación del punto de interés, si este valor sobrepasa el 80% del máximo principal, esto generara un nuevo punto clave en la misma posición y escala. (Flores & Braun, 2011, p.4).

- Descriptor de puntos clave

Aquí cada punto posee una asignación de escala, localización y orientación. Un descriptor es la concatenación de los valores de cada histograma, hallado mediante el gradiente de la sub división en 4 regiones de 4x4 vecinos del punto (Flores & Braun, 2011, p.4).

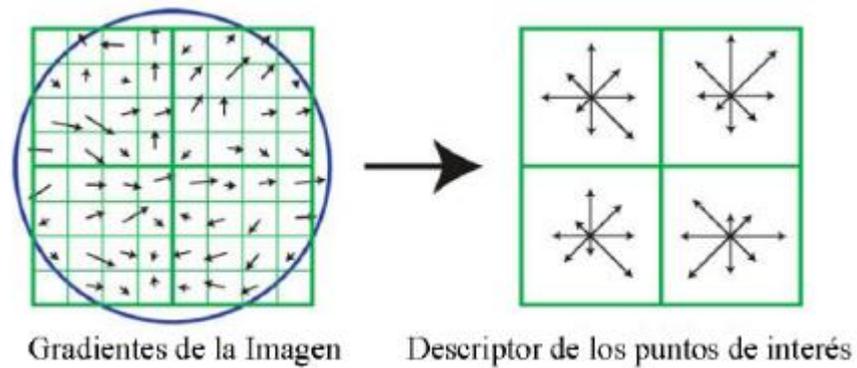


Figura 16-1: Gradientes en la ventana Gaussiana e Histograma de puntos

Fuente: https://www.researchgate.net/figure/Usando-una-ventana-gaussiana-los-valores-m-y-TH-examina-en-la-vecindad-del-punto-de_fig4_266488057

Surf

El algoritmo SURF es un detector y descriptor de puntos de interés, consta de etapas similares al algoritmo SIFT, su principal diferencia es el resultado del vector descriptor que es de un tamaño de 64, a comparación de SIFT que es de 128. Además, según sus autores es computacionalmente más rápido. Las etapas son: generación de imagen integral para agilizar cálculos, creación del espacio escala, localización de puntos invariantes máximos y mínimos del espacio de escala; asignación de orientación y generador del descriptor.

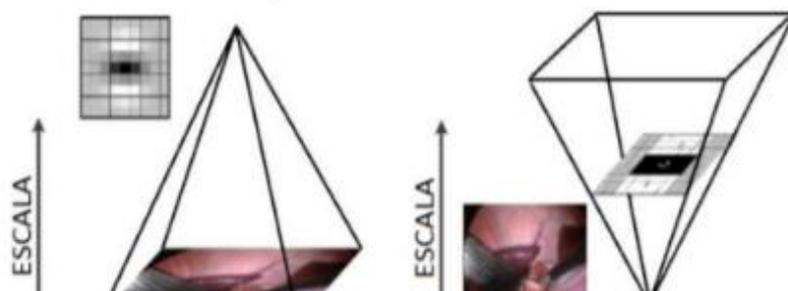


Figura 17-1: Espacio escala mediante SIFT (izq.) y SURF (dcha.)

Fuente: http://oa.upm.es/20480/1/INVE_MEM_2012_135438.pdf

1.4.1.3. Resolución de Homografía

Se denomina homografía a la proyección que determina una correlación entre dos figuras planas, de tal manera que a cada punto y recta de una de ellas le corresponde un punto y una recta de la otra respectivamente.

Ransac

Es un algoritmo iterativo utilizado para estimar los parámetros de un modelo matemático de un conjunto de datos, se utiliza para estimar una homografía a partir de puntos correspondientes obtenidos mediante algoritmos de detección y descriptores de puntos de interés (Flores & Braun, 2011a, p.1).

La idea es utilizar la menor cantidad de puntos posibles para estimar el modelo y luego ver cuántos puntos se ajustan al modelo estimado en un principio (Flores & Braun, 2011a, p.1).

El algoritmo es el siguiente:

- Dado un modelo con un número mínimo n para determinar sus parámetros y un conjunto de datos P tal que $\text{datos de } P > n$, se escoge al azar un subconjunto $S1$ de n puntos de P para instanciar el modelo $M1$, con este se determina un subconjunto de $S1^*$ de puntos de P que estén a menos de una distancia t de $M1$ (Flores & Braun, 2011a, p.1-2).
- Si el número de puntos en $S1^* > T$, siendo T un umbral, entonces se elige el subconjunto de decisión $S1^*$ para computar e nuevo modelo $M1^*$. Si el número de puntos en $S1^*$ es menor que T , se sortea un nuevo subconjunto $S2$ y se repite el proceso. Esto se repite N veces y si no se obtiene un subconjunto de decisión que cumpla con el umbral T , se resuelve el modelo con el subconjunto de decisión más grande obtenido, o se termina sin devolver el modelo (Flores & Braun, 2011a, p.1-2).
- Dentro de este algoritmo se debe estimar 3 parámetros, el umbral de distancia t , para determinar compatibilidad del modelo, para que se pueda dar una probabilidad de que un punto pertenezca o no; número de iteraciones N que se calcula en función de la probabilidad de que por lo menos un subconjunto que se elige esté libre de outliers; y número mínimo T de puntos en el subconjunto de decisión que debe ser similar al número de inliers que se cree que existen.

1.4.1.4. Unión de imágenes

Aquí se proyecta cada imagen en una misma superficie y se realiza la mezcla en función a los casos anteriores. Aquí se realiza la unión de las imágenes realizando una compensación de ganancia esto con el fin de minimizar la diferencia de intensidad entre los píxeles superpuestos.

Siendo el pixel más cerca del centro de la imagen el de mayor peso, realizando una mezcla en las tres bandas para evitar el desenfoco (Hoiem, 2010, p.58-60).

1.4.2. Segmentación de imágenes

La segmentación de imágenes es la subdivisión de una imagen en partes según una característica, objeto o grupo de píxeles iguales o similares. El objetivo principal es separar regiones de interés del resto de la imagen, por lo cual se debe realizar una segmentación en función al problema a resolver (Palomino & Concha, 2014, p.10).

Dentro de los métodos que se usan para realizar algoritmos de segmentación, se basan en propiedades de discontinuidad la cual se basa en los cambios en los niveles de intensidad, similitud donde se realiza la segmentación en zonas que posean valores similares y umbralización que se usa para diferenciar objetos del fondo de la imagen mediante una binarización.

1.4.2.1. Umbralización

Es un método que busca dividir las imágenes creando una máscara binaria de las intensidades de la imagen, esto con el fin de determinar el valor de intensidad llamado umbral, que sirve para separar las clases deseadas. La segmentación se hace agrupando todos los píxeles con mayor valor que el umbral y todos los píxeles con menor valor que el umbral en otra (Coto, 2003, p.9-10).

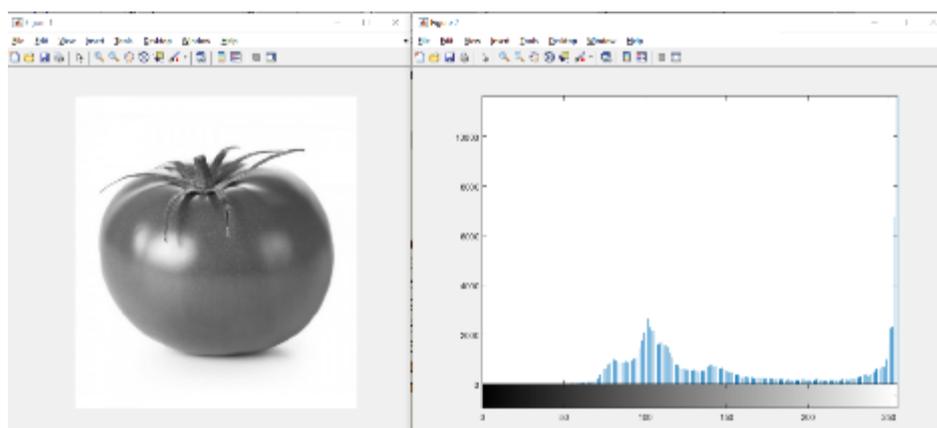


Figura 18-1: Histograma imagen con fondo blanco

Realizado por: Tacuri & Olmedo, 2018

Como se puede apreciar en la figura 17-1, claramente podemos determinar mediante el histograma, que la imagen posee 2 clases lo cual nos proporciona el valor del umbral para la segmentación de imagen.

La umbralización es una técnica efectiva para segmentación de imágenes, donde estructuras diferentes tienen intensidades contrastantes u otras características diferenciables (Coto, 2003, p.9-10), por eso no se recomienda para imágenes multibandas.

1.4.2.2. *Discontinuidad*

Este método busca los cambios bruscos de niveles de intensidad, ya que, debido a propiedades de iluminación y color, por ejemplo, existen cambios dentro del histograma, que son muy notorios, debido a estos cambios bruscos podemos separar regiones. Este método se utiliza para segmentar un objeto del fondo de una imagen. Siendo lo más relevante de este la detección de puntos aislados, la detección de líneas y detección de bordes de una imagen (Palomino & Concha, 2014, p14-15).

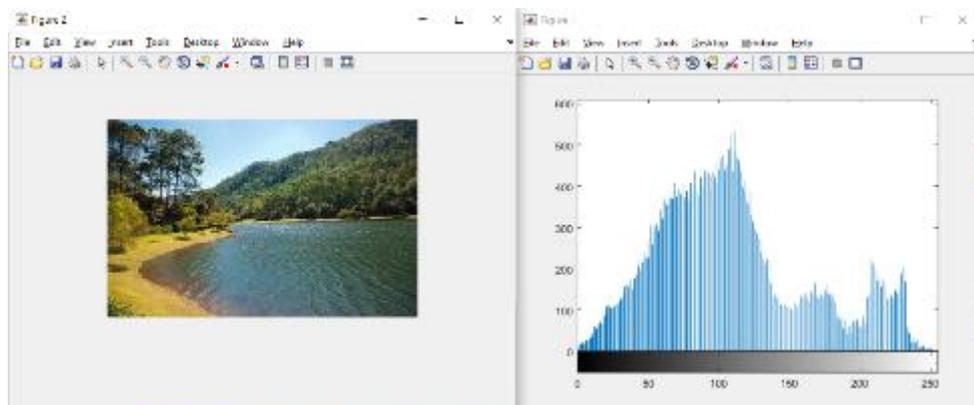


Figura 19-1: Histograma imagen con discontinuidad

Realizado por: (Tacuri & Olmedo, 2018)

1.4.2.3. *Similitud*

Este método consiste en dar o conocer características específicas de un pixel, por ejemplo, intensidad, luminosidad, color y en si propiedades bien definidas. Se orienta a agrupar los pixeles que se acerquen o poseen las características, similares o iguales al pixel base. Es decir, lo que se hace es ir comparando los valores de pixel con el que necesitamos separar, y si está dentro de un rango de tolerancia entra o no en el grupo. Con este método podemos segmentar en un número indefinido de regiones, ya que podemos ajustar la segmentación a una característica específica y con una tolerancia baja, lo que haría que los pixeles seleccionados sean muy similares entre sí.

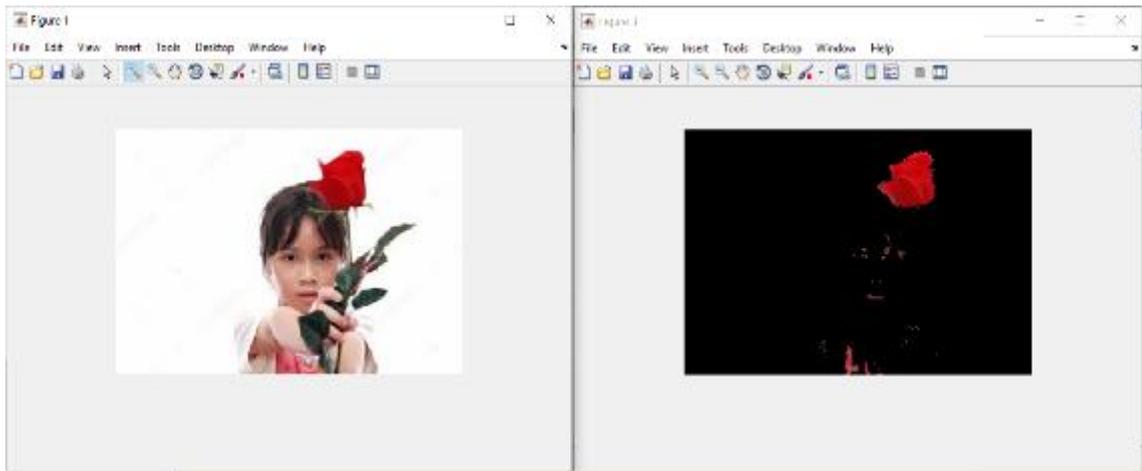


Figura 20-1: Segmentación por color

Realizado por: (Tacuri & Olmedo, 2018)

CAPÍTULO II

2. DESARROLLO E IMPLEMENTACIÓN

2.1. Elección del kit de desarrollo.

Dentro del desarrollo del proyecto se puede realizar una comparación entre dos kits de desarrollo de drones, pertenecientes a diferentes marcas y con características propias que los convierten en buenos puntos de inicio para el desarrollo del proyecto.

Se puede evidenciar, que dado la gran acogida que ha tenido los drones en los últimos años, hoy en día existen una gran cantidad de componentes, que sirven como punto de partida para el desarrollo de un prototipo de vehículo, que se adapte a nuestras necesidades tanto en hardware y software.

Pero dado a la inversión tanto económica como en tiempo, que se necesita para poder armar un vehículo con los componentes electrónicos y de software, que se ajusten a las necesidades que se presentan al momento de desarrollar este proyecto, se ha optado por elegir equipos de desarrollo completos que abarca la mayor cantidad de características para ser útiles al momento de desarrollar este proyecto.

Por lo tanto, se ha tomado en consideración éstos dos kits, realizando una breve comparación mencionando las características de cada uno de ellos, detallados a continuación:

Tabla 1-2: Características de los elementos que poseen los kits SLAMdunk e Intel Aero

	Parrot SLAMdunk	Intel Aero Ready to Fly
Procesador	NVIDIA Tegra K1	Intel Atom x7-z8750
RAM / Memoria	2 GB DDR3 / 16 GB	4 GB DDR3 / 32 GB
Sensores	IMU, Magnetómetro, barómetro, ultrasónico	IMU, Magnetómetro, Barómetro
Cámaras	Resolución 1500 x1500 a 60 FPS	RGB 8MP y VGA monocromático, RealSense R200
Controlador de vuelo	-----	Controlador Intel Aero, SMT 32
Sistema Operativo	Ubuntu, ROS	Linux Yocto, soporte para Ubuntu/ ROS
Aplicación	Sistema para navegación Autónoma, evasión de obstáculos y mapeo 3D, aplicable a drones y otra clase de robots.	Desarrollo y prueba de aplicaciones para drones, soporte de tecnología Intel RealSense para interacción con el usuario, detección de objetos y mapeo 3D

Realizado por: Tacuri & Olmedo, 2018

Tabla 2-2: Comparación entre SLAMdrunk e Intel Aero

	Parrot SLAMdunk	Intel Aero RTF
Mayor capacidad de procesamiento	NO	SI
Soporte para varios sensores	SI	SI
Mayor soporte para cámaras	NO	SI
Sistema Operativo de código abierto	SI	SI
Controlador de vuelo incluido	NO	SI
Versatilidad para desarrollar sistemas de navegación autónoma	SI	NO
Partes y componentes necesarios para montaje de un UAV	NO	SI

Realizado por: Tacuri & Olmedo, 2018

La versión de Parrot es una opción a tomar en cuenta debido a su gran versatilidad al momento de desarrollar proyecto, no solo con vehículos aéreos no tripulados sino también a otra clase de robots, posee las características necesarias para elaborar aplicaciones de navegación autónoma, necesarias para el proyecto propuesto, sin embargo, ya que solo es un módulo, que necesita ser conectado a un dron ya armado para su uso, se hace necesario adquirir las partes y componentes faltantes de otros fabricantes, pudiendo generar problemas de compatibilidad al momento de querer expandir el rendimiento del vehículo en desarrollo.

En cambio, la versión de Intel, es un kit completo, con todos los elementos necesarios para el montaje y armado de un prototipo de manera rápida y sencilla. Además de poseer características superiores al kit anterior, como mejor procesador, mayor cantidad de memoria, entre otras. Debido a esto, lo hace una plataforma interesante para el desarrollo de múltiples aplicaciones para vehículos aéreos.

Después de realizar un análisis y comparación entre los dos kits propuestos, se ha llegado a la conclusión de elegir al kit proporcionado por Intel, debido a que este posee la ventaja de ser mucho más completo y con una mejor capacidad de desarrollo de aplicaciones.

2.1.1. Intel Aero RTF

Como se mencionó anteriormente este es un kit completo de un vehículo aéreo no tripulado, es de mucha utilidad debido a que brinda opciones de configuración para usuarios que están

comenzando en el mundo de los drones así como opciones avanzadas en programación de SDKs y APIs.

A continuación, mencionamos los componentes que el kit posee.

2.1.1.1. Intel Aero Compute Board

Es la placa principal del vehículo, donde se alojan los componentes, sensores y conectores. La placa está diseñada específicamente para UAV, al tener un tamaño reducido dispone de los siguientes elementos:

- Intel® Atom™ x7-Z8750 processor
- 4 GB LPDDR3 -1600
- 32 GB eMMC
- MicroSD* memory card slot
- M.2 connector 1 lane PCIe for SSD
- Intel® Dual Band Wireless-AC 8260
- USB 3.0 OTG
- Reprogramable I/O vía Altera® Max® 10 FPGA
- Open-source embedded Linux*, Yocto Project
- Inside Software InsydeH2O* UEFI BIOS optimized for the Intel® Aero Platform for UAVs.

2.1.1.2. Intel Aero Vision

Está conformado por tres cámaras, Intel RealSense R200, una cámara de 8MP y una cámara VGA, además de incluir cables y accesorios para su montaje. Con esto proporcionamos capacidades de visión y detección para el vuelo.

- 8MP RGB camera (front-facing)
- VGA camera, global shutter, monochrome (down-facing)
- Intel® RealSense™ camera (R200)

2.1.1.3. Intel Aero controller With

Se conecta a la placa Aero Compute Board mediante HSUART y se comunica mediante el protocolo MAVLink.

- Soporte para PX4 y ArduPilot
- Microcontrolador STM32
- 6 DoF IMU, Magnetómetro y Sensor de altitud

2.1.1.4. Partes de ensamblaje del drone

- Estructura de fibra de carbono
- GPS y brújula
- Módulo de distribución de energía
- 4 ESC Yunecc FOC ESC 11.1 – 14.8 v
- 4 motores Motor Typhoon H 940kv, 11.1 – 14.8 v con montaje de hélices
- 8 hélices Yunecc Typhoon H, de sujeción rápida
- Receptor Spektrum DSMX
- Transmisor Spektrum DXe (2,4GHz DSMX)
- Tornillos y cables para montaje y conexión

Todo el kit hace que el drone tenga las siguientes características:

Tabla 3-2: Características de los principales elementos del Drone

Dimensiones: extremo a extremo	360 mm
Altura del Drone: Desde la base a la parte superior de la antena del GPS	222 mm
Hélices (Yunecc Typhoon H): Longitud de extremo a extremo	230 mm
Peso del Drone: Estructura básica sin considerar la batería	865 g
Peso total máximo: Peso máximo que garantice el despegue	1900 g
Tiempo de vuelo: Con una batería Lipo 4S, 4000mAh	20 min
Distancia máxima de control: Con el control remoto Spektrum DXE	300 m
Límite de Altura de vuelo: Sobre el nivel del suelo	122 m
Temperatura: mínima y máxima	0 C / +45 C
Motores: tipo Brushless A.C. 940Kv	11.1 – 14.8 V
ESC (Controlador de velocidad electrónico) Diseñados y fabricados por Yuneec, modificados por Intel Aero - Interfaz de control de entrada	11.1 -14.8 V 20 A UART

Fuente: <https://software.intel.com/es-es/aero/drone-kit>

2.2. Montaje

Una vez conocidos los puertos y lugares donde se ubican cada uno de los componentes, se procede con el armado del vehículo UAV, utilizando el diagrama de bloques como referencia. Primero se monta en el chasis los cables de conexión que parten desde la placa de distribución de energía hacia los ESC y posteriormente hacia los motores en cada una de sus ubicaciones, procurando que los motores A y B se encuentren en posiciones alternas respectivamente, luego la placa Intel Aero debe tener una comunicación física con cada uno de los componentes principales del Drone, la placa de distribución de energía, las cámaras, el módulo GPS y finalmente los motores.



Figura 1-2: Frame F50,

Realizado por: (Tacuri & Olmedo, 2018)



Figura 2-2: Motor Typhoon H 940kv

Realizado por: (Tacuri & Olmedo, 2018)



Figura 3-2: ESC, Yunecc FOC ESC

Realizado por: (Tacuri & Olmedo, 2018)



Figura 4-2: Intel Aero Compute Board

Realizado por: (Tacuri & Olmedo, 2018)



Figura 5-2: Placa de Energía

Realizado por: (Tacuri & Olmedo, 2018)



Figura 6-2: Cámara Intel RealSense R200

Realizado por: (Tacuri & Olmedo, 2018)



Figura 7-2: Módulo GPS con brújula

Realizado por: (Tacuri & Olmedo, 2018)



Figura 8-2: Ensamblaje del Drone y pruebas de funcionamiento

Realizado por: (Tacuri & Olmedo, 2018)



Figura 9-2: Dron Intel Aero

Realizado por: (Tacuri & Olmedo, 2018)

Para realizar pruebas de vuelo y adaptar el vehículo a las necesidades que se presentan, se realiza adaptaciones y cambios dentro del vehículo original como se muestra en la Figura 10-2. El principal inconveniente a superar al momento de realizar las primeras pruebas de vuelo es la protección de hélices y motores razón por la cual se debe adaptar unos protectores con la finalidad de evitar daños significativos y considerablemente altos en la integridad de los motores y principalmente de las hélices al momento de realizar pruebas de vuelo en la de detección y evasión de obstáculos.

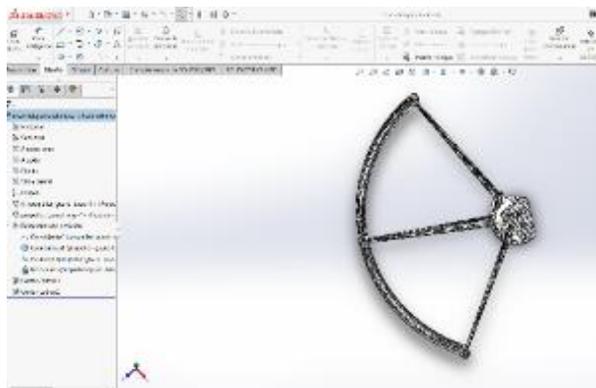


Figura 10-2: Protectores de hélices, Solid Works

Realizado por: (Tacuri & Olmedo, 2018)



Figura 11-2: Protector de hélices en la base del motor

Realizado por: (Tacuri & Olmedo, 2018)

Estos protectores son diseñados en el programa de modelamiento SolidWorks y posteriormente construidos en una impresora 3D con material ABS, este material se ha elegido debido a la precisión que ofrece la máquina con un error de 0.5mm.

Debido a la necesidad de imágenes de buena calidad se optó en montar una cámara tipo deportiva, la cual permite capturar imágenes de 16 MP (Megapíxeles) y con un gran foco angular, logrando obtener imágenes de mayor calidad y mucho más confiables pudiendo de este modo, capturar una mayor y más definida área por imagen, en comparación con la cámara original.

La nueva cámara es sujeta en la parte inferior del dron con una correa que se adapta perfectamente a la cámara y los orificios de sujeción del dron, son suficientes estas adaptaciones debido a que se obtienen los resultados esperados con la adición de un peso lo más bajo posible.



Figura 12-2: Vista inferior del dron, con cámara.

Realizado por: (Tacuri & Olmedo, 2018)

En el momento de accionamiento el dispositivo no presenta vibraciones de consideración elevada y debido a que la velocidad de recorrido es baja, las imágenes adquiridas son aceptables para un posterior procesamiento.



Figura 13-2: Estructura final del Drone

Realizado por: (Tacuri & Olmedo, 2018)

2.3. Configuración Inicial del Intel Aero RTF

2.3.1. *Arquitectura y Especificaciones de Hardware de la placa Intel Aero*

Para una mejor comprensión del proceso de calibración y configuración del Intel Aero RTF, es necesario investigar sobre las conexiones existentes dentro de la placa, A continuación, en la figura 14-2, se presenta un diagrama conexión y las especificaciones que posee esta placa.

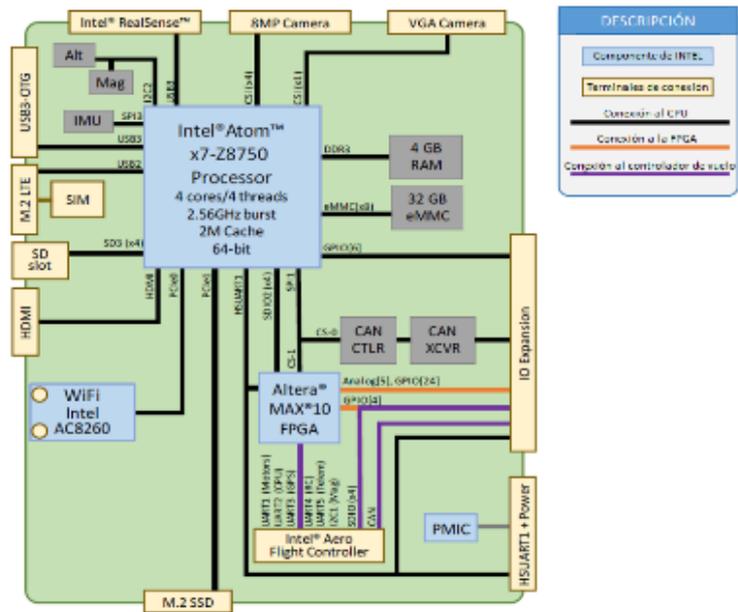


Figura 14-2: Diagrama de bloques Intel Aero Compute Board

Fuente: <https://www.intel.com/content/dam/support/us/en/documents/drones/development-drones/intel-aero-compute-board-guide.pdf>

A continuación, en la Figura 15-2, se presenta un diagrama de conexiones y especificaciones del controlador de vuelo.

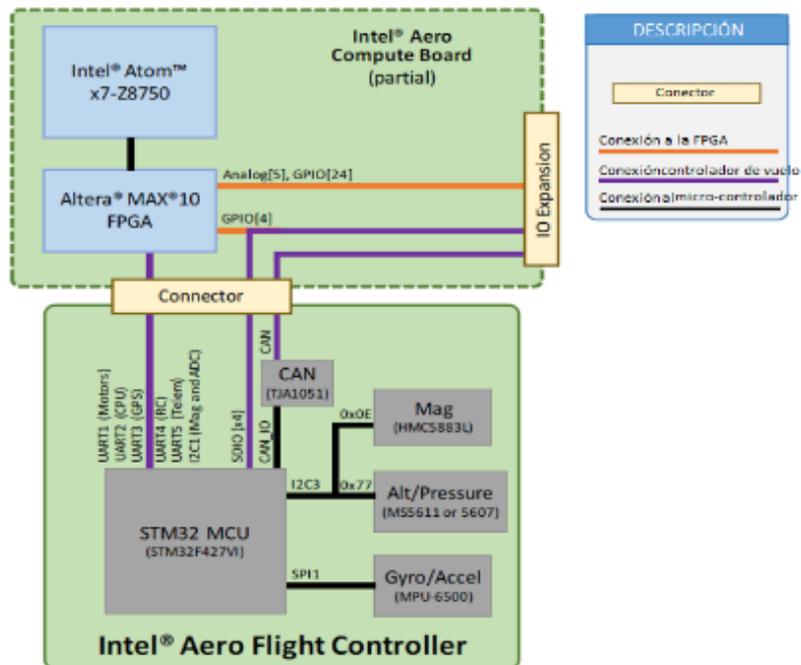


Figura 15-2: Diagrama de bloques Intel Aero Flight Controller

Fuente: <https://www.intel.com/content/dam/support/us/en/documents/drones/development-drones/intel-aero-compute-board-guide.pdf>

Con esta información podemos saber cuáles son elementos internos de hardware que son utilizados en este proyecto, esto permite conocer la estructura física de estas tarjetas controladoras, pudiendo de este modo conocer la ubicación de cada dispositivo de control en caso de que existan inconvenientes y sucesos no deseados en la utilización de éstos delicados elementos.

Para una información más detallada acerca de estos dispositivos se puede consultar la guía “Intel Aero Compute Board Hardware Features and Usage Rev. 1.5”, proporcionada por el fabricante.

2.3.2. Consideraciones en la configuración y especificaciones de software

La plataforma Intel Aero RTF es de sistema Linux y de código abierto, de forma predeterminada viene con una versión del específica de Intel del Proyecto Yocto, BIOS UEFI por InsydeH20. No obstante el sistema operativo Yocto no puede resultar una plataforma de desarrollo muy interactiva con el usuario, por esa razón se debe mencionar que la placa Intel también puede soportar una distribución típica de Linux Ubuntu, que puede ser instalada de manera manual.

Para poder realizar la configuración es necesario tener una disponibilidad de los siguientes elementos:

- Adaptador USB OTG, el puerto disponible es 3.0 pero también funcionan adaptador 2.0
- Un HUB USB, para conectar varios periféricos, como teclado, mouse y una memoria USB
- Cable Micro HDMI para realizar una programación con salida de video de la placa.
- Teclado USB
- Memoria flash USB de 2 GB o más para actualizaciones de firmware

2.3.3. Configuración Inicial

Para realizar la configuración inicial se debe contar con los elementos mencionados anteriormente, y cabe resaltar que una de las características más importantes a nivel de software que posee este kit, es la posibilidad de formatear tres partes. La primera parte consiste en la actualización de BIOS, la segunda es la actualización de la FPGA y finalmente la actualización del controlador de vuelo.

La configuración inicial se realiza descargando la imagen .iso del sistema operativo desde la página oficial de Intel Aero, la cual contiene todos los archivos necesarios para realizar todo el proceso de configuración.

Podemos realizar la configuración usando uno de los dos métodos, el primer método que es el recomendado es el que consiste en conectar un cable micro HDMI y un monitor para poder

trabajar directamente en la Intel Aero Board; el segundo método es de forma remota conectándonos al punto de acceso del Dron y realizando una conexión SSH (Secure Shell, este intérprete de órdenes es utilizado para acceder a servidores privados a través de una puerta trasera), para tener acceso a un terminal del Intel Aero Board.

La actualización se hace como si se instalara cualquier sistema operativo desde una USB, es decir:

- Entrar en la BIOS, al inicio del sistema operativo, pulsando ESC en el teclado
- Seleccionar inicio desde llave USB
- Instalar

Todo el proceso de instalación y mensajes de configuración se puede observar en la pantalla HDMI conectada al equipo o desde el ordenador conectado remotamente a la tarjeta del dron.

2.3.3.1. *Configuración de la tarjeta paso a paso*

Se debe utilizar el siguiente comando en el terminal:

```
aero-reboot-update.py
```

Sin importar cual fuera la opción seleccionada los comandos utilizados para realizar el proceso de configuración y actualización son los siguientes:

2.3.3.2. *Actualización de la BIOS*

Se debe descargar la última versión de la BIOS desde la página oficial y copiarla en la Intel Aero Board, y seguido ejecutar los siguientes comandos:

```
rpm -ev aero-bios-01.00.13-r1.corei7_64  
rpm -i aero-bios-01.00.16-r1.corei7_64.rpm  
aero-bios-update  
reboot
```

Estos comandos realizan las siguientes acciones: la primera línea sirve para retirar la versión antigua, la segunda línea para instalar la versión actual y la tercera línea ejecuta la actualización.

2.3.3.3. *Actualización de FPGA*

Para realizar la actualización de la FPGA se debe escribir en el terminal los siguientes comandos:

```
cd / etc / fpga /
```

```
jam -aprogram aero-rtv f|2f.jam
```

Estos comandos realizan las siguientes acciones: la primera línea ubica el directorio donde se encuentra el archivo necesario, y la segunda línea procede con la actualización.

2.3.3.4. Actualización del controlador de vuelo

Para realizar la actualización del controlador de vuelo se debe escribir en el terminal los siguientes comandos:

```
cd / etc / arofc / px4 /  
aerofc-update.sh nuttx-aerofc-v1-default.px4
```

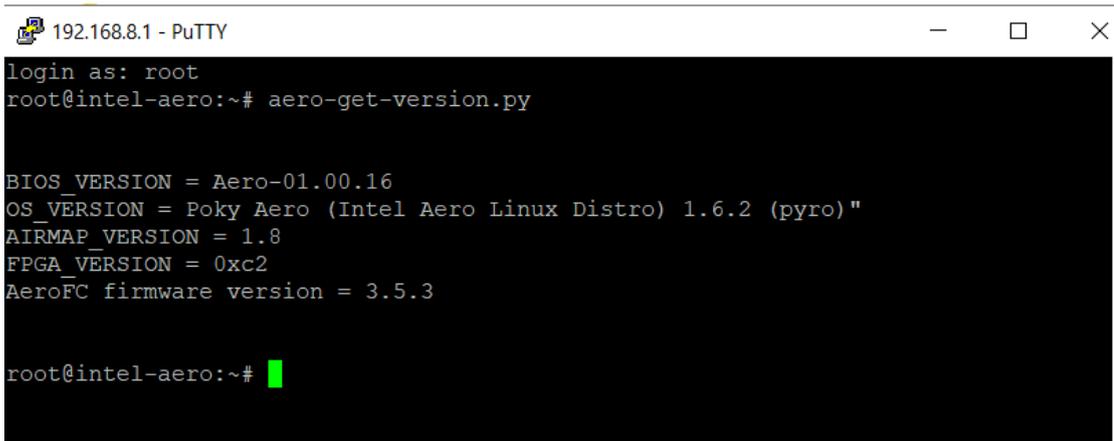
Estas líneas de código se deben tipear con el fin de actualizar la versión más actual del piloto automático PX4, pero también se puede instalar la versión ArduPilot.

```
cd / etc / arofc / ardupilot /  
aerofc-update.sh arducopter-aerofc-v1.px4
```

Una vez terminada la configuración se puede realizar un reinicio del sistema y comprobar que los procesos de actualización se llevaron a cabo de una manera satisfactoria, para ello se utiliza el comando:

```
aero-get-version.py
```

Como resultados de comprobación se obtiene lo siguiente:



```
192.168.8.1 - PuTTY  
login as: root  
root@intel-aero:~# aero-get-version.py  
  
BIOS_VERSION = Aero-01.00.16  
OS_VERSION = Poky Aero (Intel Aero Linux Distro) 1.6.2 (pyro)"  
AIRMAP_VERSION = 1.8  
FPGA_VERSION = 0xc2  
AeroFC firmware version = 3.5.3  
  
root@intel-aero:~# █
```

Figura 16-2: Resultados de actualización del sistema de Intel Aero RTF

Realizado por: (Tacuri & Olmedo, 2018)

2.4. Calibración

Una vez concluido el proceso de configuración y actualización, se procede con la calibración de los sensores que componen el kit de Intel Aero RTF. Este proceso se lo debe realizar cada vez que se realiza una actualización, con el propósito de garantizar un vuelo estable.

2.4.1. Selección del programa de calibración

Para realizar la calibración se necesita conectar a una estación de control en tierra, pudiendo ser “QGroundControl” (QGC) o a su vez también es posible utilizar el software de control en tierra llamado “Mission Planner”; debido a la recomendación del fabricante que es usar QGC, se procede con la utilización del primer software mencionado, debido a que se puede contar con documentación específica para este vehículo, pero cabe mencionar que, también se debe realizar una actualización en Mission Planner, comprobando también su funcionalidad en este software.

2.4.2. Resumen del estado de calibración

En la siguiente imagen se presenta la pantalla general, donde se observa un resumen de diferentes parámetros y características que posee el vehículo sincronizado con este software.

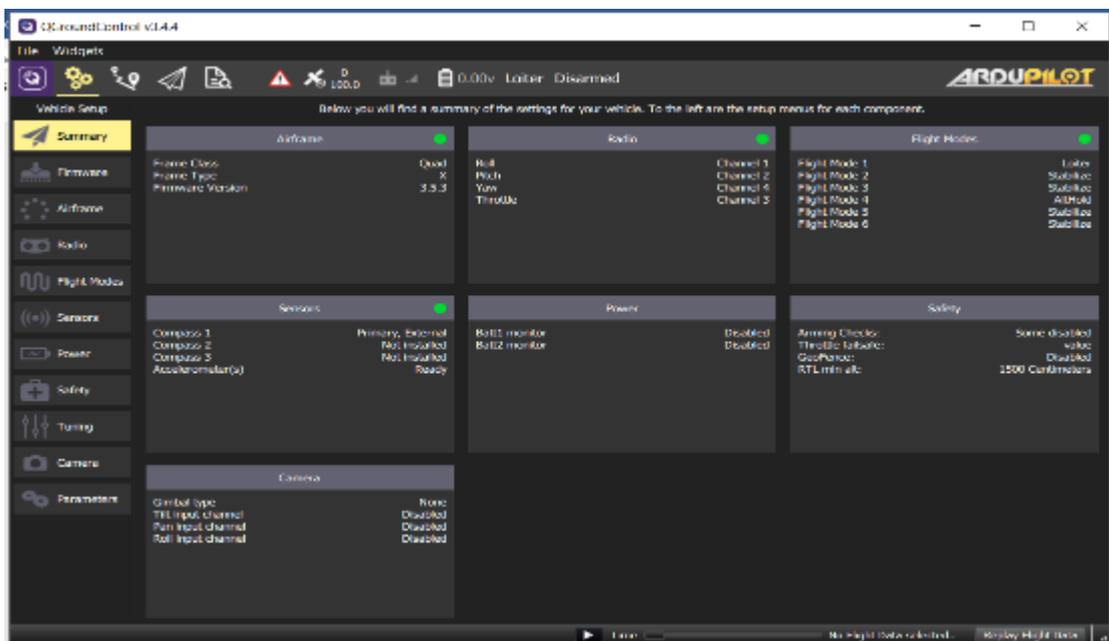


Figura 17-2: Ventana de estados de los sensores del UAV en QGC (QGroundControl)

Realizado por: (Tacuri & Olmedo, 2018)

En la parte izquierda de la imagen se encuentra cada uno de los parámetros a calibrar para una eficaz funcionalidad del vehículo. Dentro de estas las más importantes, y las que se debe calibrar según la documentación proporcionada por Intel es el Airframe, Radio, Sensors y Flight Modes (Ubicados en la segunda, tercera cuarta y quinta opción respectivamente).

El proceso de calibración se lo realiza de manera sencilla e intuitiva siguiendo las instrucciones que el programa QGC muestra paso a paso, para cada uno de los diferentes parámetros.

2.4.3. Configuración de frame (Fuselaje)

La pestaña marcada muestra la configuración con la que se asocia a UAV

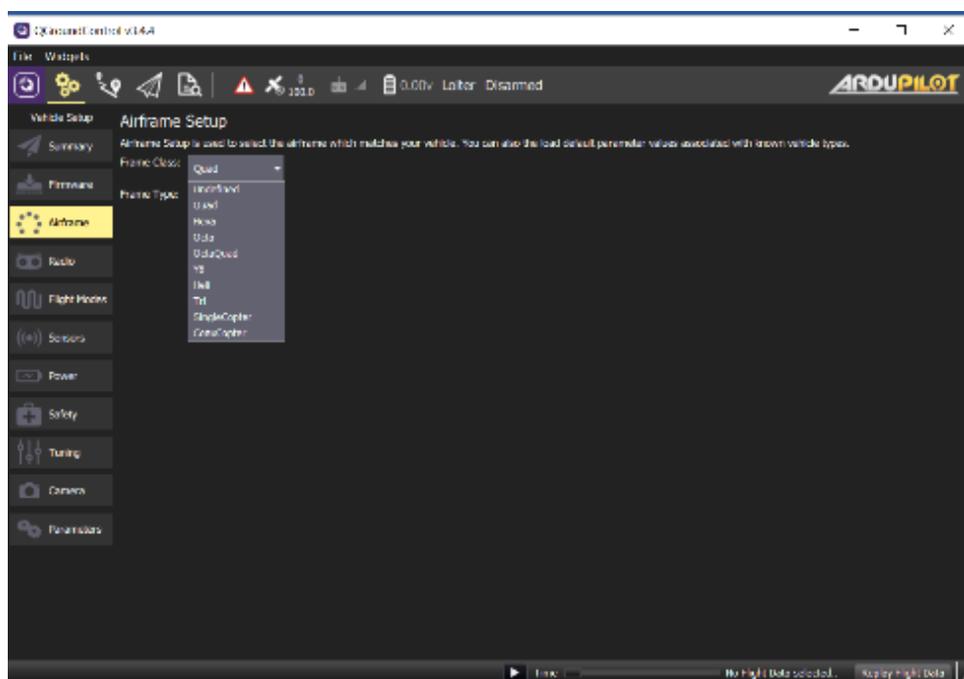


Figura 18-2: Selección del Fuselaje o Airframe del UAV

Realizado por: (Tacuri & Olmedo, 2018)

2.4.4. Configuración del radio control

En la pestaña de radio se procede con la configuración de la palanca de radio con la finalidad de establecer las configuraciones máxima y mínima para los canales de RC. Este proceso se lo lleva

a cabo moviendo los patrones que indique el software uno por uno entonces, la parte superior derecha de la pantalla indica las posiciones que deben tomar cada uno de los canales de control.

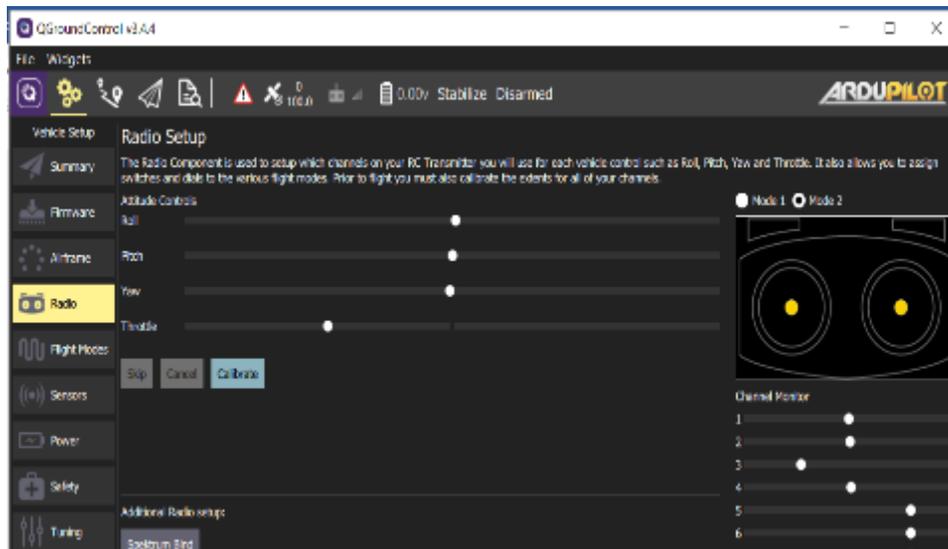


Figura 19-2: Calibración del radio control

Realizado por: (Tacuri & Olmedo, 2018)

2.4.5. Configuración de los sensores

En esta sección del proceso de calibración se procede con la calibración del giroscopio el acelerómetro y la brújula que posee el dispositivo UAV, estos sensores aparecerán enlistados junto a la barra de calibración y están marcados con color rojo si no están calibrados y con un color verde en el caso de que se encuentren listos.

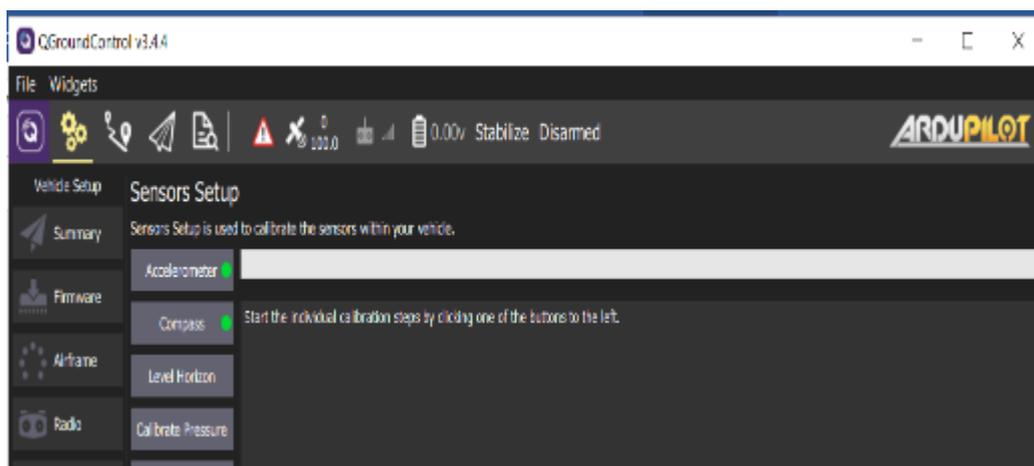


Figura 20-2: Calibración de sensores

Realizado por: (Tacuri & Olmedo, 2018)

2.4.6. Configuración del modo de vuelo

En esta opción se configura y asocia cada uno de los modos de vuelo con los canales disponibles en la palanca de RC. Es recomendable utilizar la sección de un solo canal debido a que es sencillo de entender y configurar.

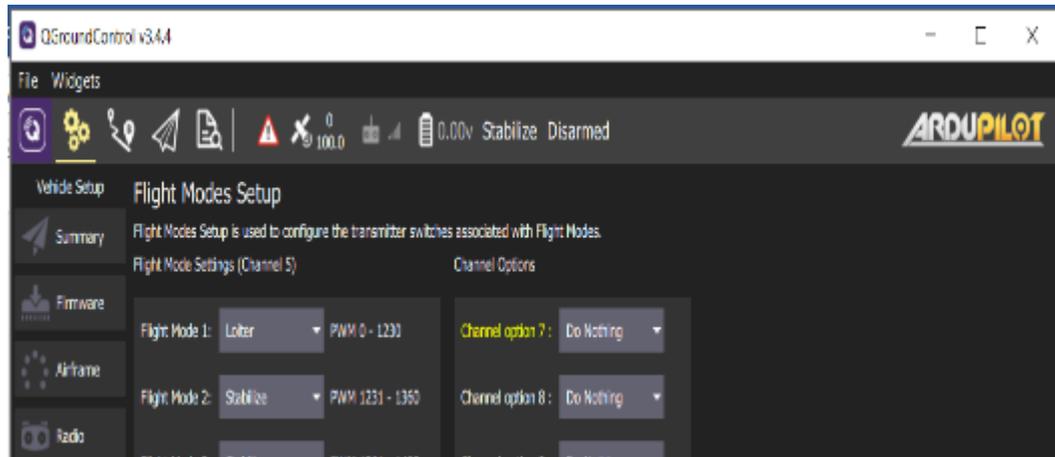


Figura 21-2: Configuración de los canales de los modos de vuelo

Realizado por: (Tacuri & Olmedo, 2018)

Una vez concluido el proceso de calibración podemos proceder a realizar vuelos de comprobación, para la verificación de cada uno de los parámetros.

2.5. Sistema anticollisiones

Para realizar una misión con las debidas precauciones es necesario contar con un sistema que permita detectar obstáculos, evaluarlos y evitarlos, para de este modo proteger el drone. Se ha decidido tomar la opción de implementar un sistema anticollisiones alterno con un sensor ultrasónico y debido a la complejidad de Intel Aero Compute Board y las restricciones que se presentaban en el momento de realizar cambios significativos, la alternativa mencionada nos permitió cumplir con el objetivo planteado.

La desventaja de esto, está en que solo pueden detectar objetos en una única dirección, siendo los laterales un punto ciego, esto se puede corregir añadiendo más sensores, aumentando así la otra gran desventaja, que es, la necesidad de usar algoritmos complejos lo que radica en una latencia mayor en el procesamiento de la información.

2.5.1. Sistema anticollisiones basado en sensores ultrasónicos

Tomando en cuenta que la utilización de las librerías anticollisiones “COAV” junto con la cámara Intel RealSense R200 no se tiene los resultados esperados, se optó por utilizar un sensor ultrasónico el cuál fue situado en la parte frontal del fuselaje del drone. Al trabajar con una frecuencia del orden de 40KHz y con una distancia de detección de hasta 4 metros, su tarea será detectar cualquier obstáculo que se presente y de manera automática reducir las revoluciones de los motores, interrumpiendo la misión y aterrizando el dispositivo y con ello precautelar la integridad del drone ante posibles amenazas.



Figura 22-2: Sensor ultrasónico

Realizado por: Tacuri & Olmedo, 2018

La acción de parar progresivamente las revoluciones en los motores se efectuó en el transmisor Spectrum DXE mediante la reducción automática de la resistencia en el potenciómetro que controla el movimiento vertical del drone, para ello es necesario implementar un sistema de detección individual para el drone, es decir, no se intervine la tarjeta de Intel debido a la complejidad propia del sistema, añadirle un sistema que no sea recomendado por el fabricante puede llegar a descomponer los elementos del kit. Por tanto, esta acción fue llevada a cabo con éxito con la ayuda de un sistema de transmisión y recepción de datos de Arduino y el módulo NRF24L01.



Figura 23-2: Módulo NRF24101 y Antena 2.4GHz

Realizado por: Tacuri & Olmedo, 2018

Estas antenas tienen un protocolo de comunicación SPI, pudiendo así establecer una comunicación con casi cualquier microcontrolador (PIC's, Arduino, Raspberry entre otros). Por lo tanto se optó por el microcontrolador Arduino Nano debido a su peso y sobre todo porque se ha trabajado con estos microcontroladores anteriormente.

Este sistema de transmisión de comunicación nos ayuda en el envío de datos del sensor ultrasónico ubicado en el dron al transmisor DXE Spectrum, el sistema funciona debido a que el alcance de comunicación entre estos dispositivos es de hasta 1km utilizado en modo alto y de 40m en bajo, además podemos alimentar el Arduino junto con el módulo ubicados en el drone con un voltaje de 5v que pueden ser suministrados por Intel Aero compute Board.

Por lo tanto, la tarea de este sistema fue la siguiente:

El principio de funcionamiento del paro de emergencia ante la presencia de obstáculos en la trayectoria del dron es la siguiente.

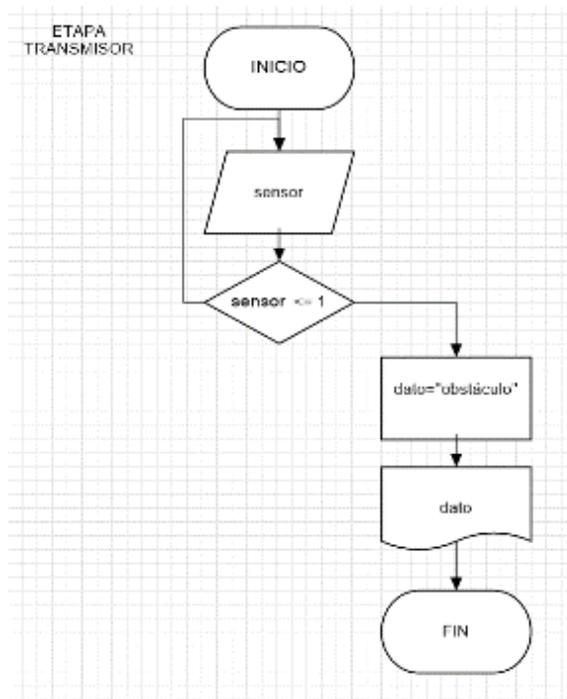


Figura 24-2: Diagrama de flujo del transmisor

Realizado por: Tacuri & Olmedo, 2018

Simultáneamente se envía un dato al receptor ubicado en el TRANSMISOR DXE del dron el cuál mediante la acción de un relé, switcheará el potenciómetro del acelerador hacia otro potenciómetro adaptado con un servomotor y efectúa una disminución mecánica progresiva de la aceleración de los motores.

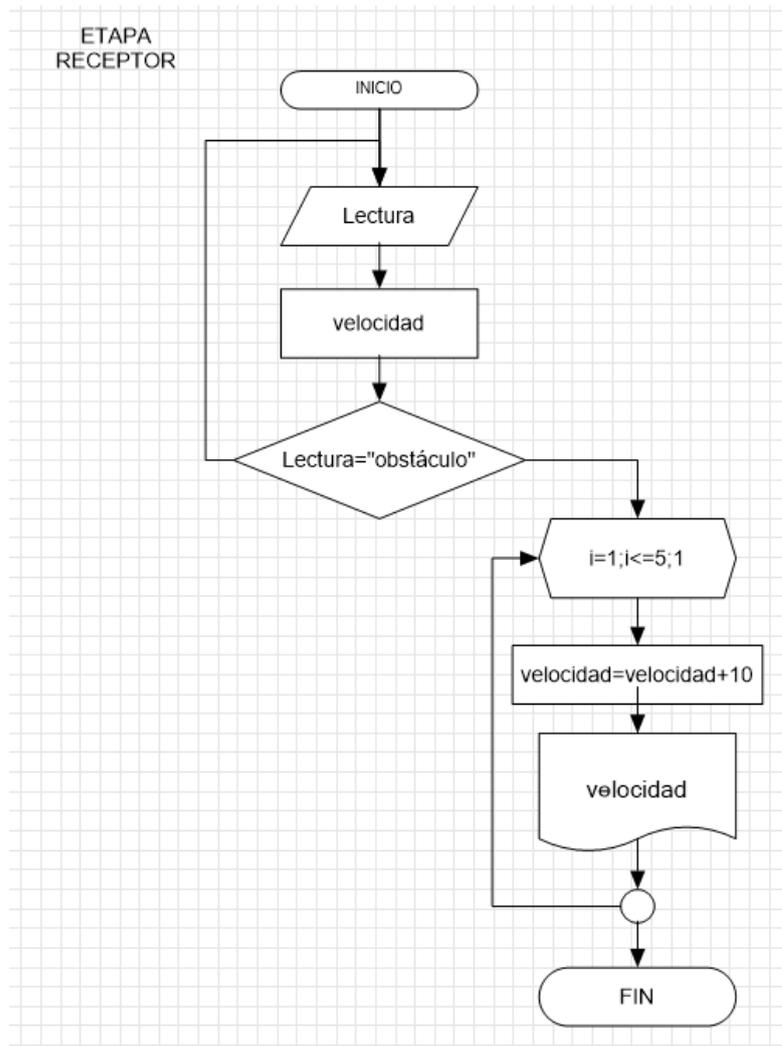


Figura 25-2: Diagrama de flujo del receptor

Realizado por: Tacuri & Olmedo, 2018

Finalmente el objetivo fue hallar una solución adecuada precautelando la integridad del kit y sin la adición de un peso significativo en el dron, la ecuación dominante del servomotor se basó en los valores de la Figura 26-2:

VOLTAJE	GRADOS
2,80	0
2,64	10
2,49	20
2,33	30
2,18	40
2,02	50
1,87	60
1,71	70
1,56	80
1,40	90
1,24	100
1,09	110
0,93	120
0,78	130
0,62	140
0,47	150
0,31	160
0,16	170
0,00	180

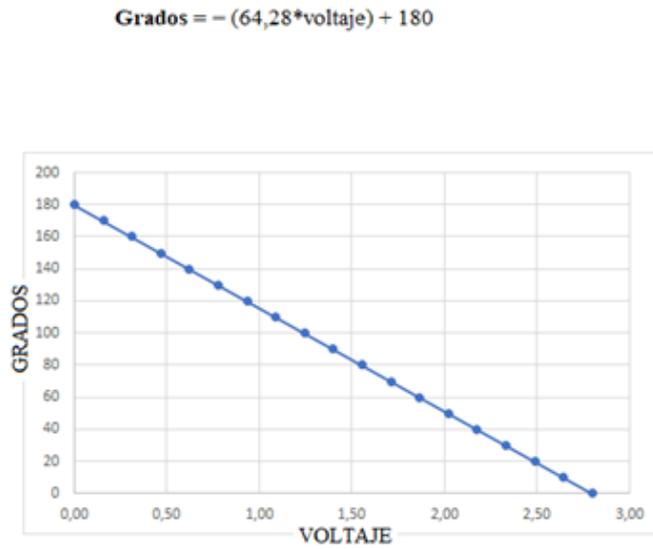


Figura 26-2: Valores estimados del ángulo de giro respecto al voltaje del acelerador

Realizado por: Tacuri & Olmedo, 2018

Los diagramas de conexión del transmisor y receptor se muestran en la Figura 27-2 y Figura 28-2 respectivamente:

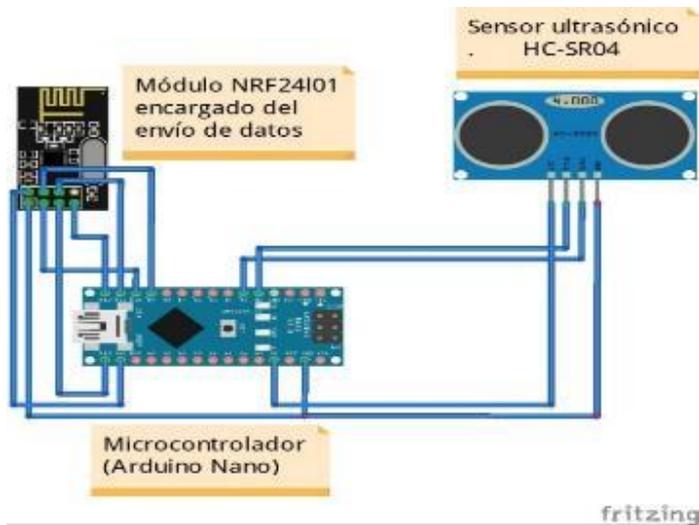


Figura 27-2: Diagrama del transmisor del sistema anticollisiones

Realizado por: Tacuri & Olmedo, 2018

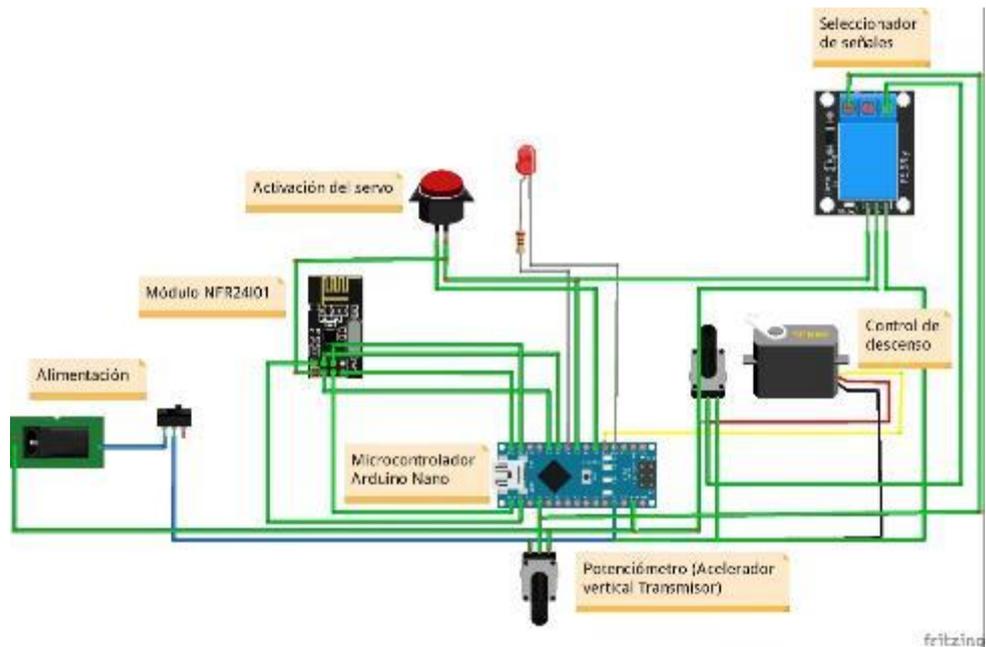


Figura 28-2: Diagrama del receptor del sistema anticollisiones

Realizado por: Tacuri & Olmedo, 2018



Figura 29-2: Sistema de paro de emergencia sujetado

Realizado por: Tacuri & Olmedo, 2018

2.6. Control en tierra

Para realizar la planificación del vuelo y la ruta a seguir por el dron, se procedió a utilizar el programa QGroundControl, por lo siguiente:

- Posee soporte para drones con PX4 y ArduPilot como piloto de vuelo automático.
- Planificación de misiones para vuelo autónomo.
- Soporte para kit Intel Aero Board.
- Software de código abierto y amplia documentación.
- Software recomendado por el fabricante.

Debido a que el soporte de Intel utiliza este software, para realizar las configuraciones y pruebas, se optó por elegir dicho programa sobre otros similares, siguiendo así las recomendaciones del fabricante.

2.6.1. Planificación del vuelo autónomo

Para realizar la planificación de la misión de vuelo, se procede a abrir el programa QGroundControl, cabe mencionar que, debe estar conectado a internet para poder descargar el mapa de la zona en la cual se desarrolla la misión, sin embargo, una vez cargada esta zona dentro del programa no es necesario una nueva conexión a internet.

Una vez ingresada a la ventana de planificación se inserta los diferentes puntos de control por donde el vehículo tendrá que sobrevolar.

En la imagen se observa una planificación de una misión, dentro de esta tenemos:

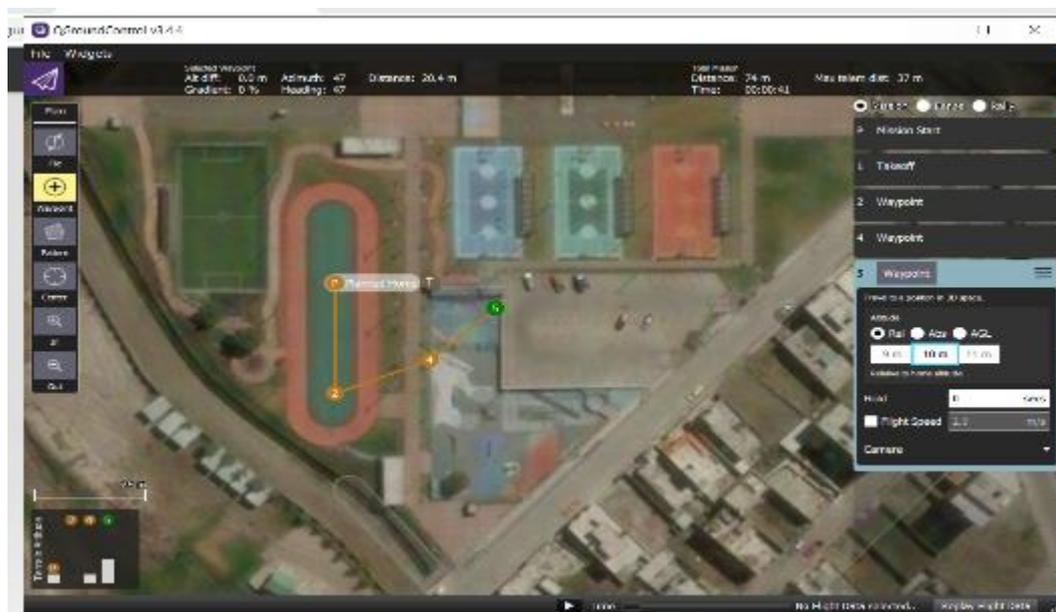


Figura 30-2: Planificación de una misión en QGroundControl

Realizado por: (Tacuri & Olmedo, 2018)

- **MAIN TOOLBAR:** Que indica la información principal de la misión actual que se encuentra llevando a cabo, muestra datos principales de la misión que se planea ejecutar como distancia horizontal y tiempo de la misión.
- **PLAN TOOLS:** Son las herramientas que se usan para crear la misión, en la cual se encuentra la herramienta de “Waypoint” que sirve para añadir un punto de misión, “Pattern” que permite crear misiones dentro de una figura geométrica.
- **COMMAND EDITOR:** Herramienta que permite configurar los parámetros de cada punto de vuelo, entre estos puntos que abarcan tenemos: altura, velocidad y el tipo de acción que hará el dron en dicho punto.

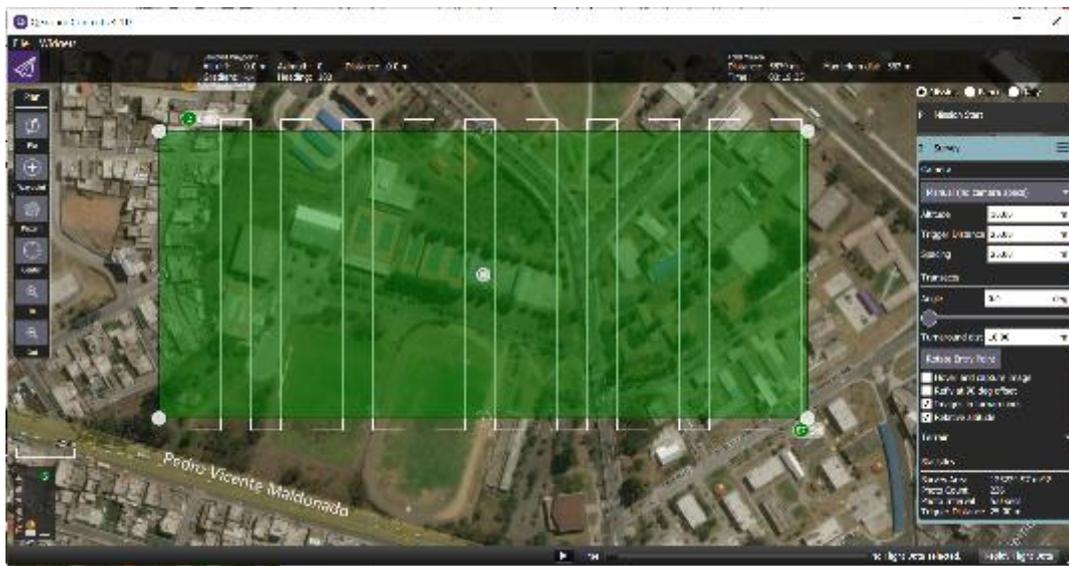


Figura 31-2: Ventana del plan de vuelo

Realizado por: (Tacuri & Olmedo, 2018)

Utilizando las herramientas anteriores se procede a crear una misión, con los puntos que sean necesarios para obtener una cobertura total del terreno con el menor número de imágenes posibles. Cuando se haya elaborado la misión utilizando la herramienta Sync, para poder cargar la misión dentro del vehículo aéreo no tripulado y posteriormente, ejecutar la acción.

2.7. Captura de imágenes y procesamiento

Para la adquisición de las imágenes de las misiones de vuelo realizadas, y su debido procesamiento, se utiliza la cámara Xioma YI Lite, la ventaja de este tipo de cámaras es su reducido tamaño y peso, además que poseen gran ángulo de captura lo cual permite capturar una mayor área en un menor tiempo de vuelo, sin olvidar que el precio es aceptable respecto al costo-beneficio. Para el procesamiento se ha optado por la utilización del software Matlab debido a su

amplia funcionalidad en el procesamiento de imágenes y también a que se posee un conocimiento previo del programa y su lenguaje de programación.

2.8. Reconstrucción

2.8.1. Algoritmo de Reconstrucción de Imagen

A partir de las imágenes obtenidas mediante el vehículo Aero, se realiza un procesamiento de las mismas para la obtención de datos y la construcción de un mapa general de ocupación vegetal en la zona, que estará compuesto por las imágenes capturadas. La técnica utilizada para llevar a cabo, es la de Stitching, que consiste en la unión de varias imágenes para formar una más grande conformada por todas las imágenes anteriores.

Tomando como punto de partida al algoritmo propuesto por M. Brown and D. G. Lowe, en el artículo *Recognising Panoramas*. El cual propone un algoritmo robusto capaz de realizar una imagen panorámica formadas por imágenes más pequeñas que pueden tener variantes como iluminación, orientación, etc.

2.8.1.1. Algoritmo: reconocimiento panorámico

- Entrada: n imágenes desordenadas
- Extraer las características SIFT de todas las n imágenes
- Encontrar k vecinos más cercanos para cada característica usando un árbol k-d
- Para cada imagen:
- Seleccionar m imágenes de coincidencia de candidatos (con el número máximo de coincidencias de características para esta imagen)
- Encontrar coincidencias de características geoméricamente conformes utilizando RANSAC para resolver la homografía entre pares de imágenes
- Verificar que las imágenes coincidan con el modelo probabilístico
- Buscar componentes conectados de coincidencias de imágenes
- Para cada componente conectado:
- Realizar el ajuste del paquete para resolver la rotación θ_1 , θ_2 , θ_3 y la distancia focal f de todas las cámaras
- Renderizar panorámicas usando mezclas multibandas

- Salida: imagen panorámica

Este es el algoritmo propuesto por Matthew Brown y David G. Low; que utiliza características invariantes, para la creación automática de imágenes panorámicas, el mismo que utilizamos como referencia para realizar la reconstrucción de las imágenes captadas por el dron en el proyecto.

2.8.2. Implementación del algoritmo

Para la implementación del algoritmo se procedió con el uso del software MATLAB, debido a que este posee:

- Soporte para múltiples formatos de imágenes
- Soporte para operaciones sobre imágenes como segmentación, morfología, estadística reconocimiento de imágenes, etc.
- Soporte para transformaciones geométricas como rotar, reducción, corrección de distorsiones, etc.
- Toolbox de procesamiento de imágenes
- Amplia cantidad de documentación
- Previo conocimiento del interfaz del programa y su lenguaje de programación

2.8.2.1. Librería vlfeat

Es una biblioteca de código abierto que implementa algoritmos de visión por computadora, especializados en la comprensión, la extracción y coincidencia de características, desarrollada en C para una gran compatibilidad y eficiencia con MATLAB. Esta biblioteca nos permitirá implementar el algoritmo de una manera más rápida debido a que posee librerías como: FISHER VECTOR, VLAD, SIFT, MSER, entre otras.

2.8.3. Componentes del proyecto

El proyecto está compuesto por funciones, que cumplen diferentes propósitos para obtener el resultado esperado. Los dos procesos principales son el algoritmo de Stitching y el algoritmo de segmentación, los cuales nos darán los resultados esperados.

2.8.3.1. Función codigo.m

Esta es la función principal del proyecto, dentro del cual se realiza el proceso de reconstrucción de imagen.

La primera parte que se realiza es la carga de imágenes, para esto se crea un datasets, con el nombre de las carpetas que contienen las imágenes de la misión, estas carpetas se las ha nombrado como Misión seguida del número correspondiente, Misión1, Misión2, etc. Se busca la carpeta especificada y se procede a cargar todas las imágenes en su interior para luego reducir su tamaño con un factor de escala de 0.45, esto con el fin de poder realizar el procesamiento en menor tiempo y con un mayor número de imágenes.

```
function [panorama]=codigo(filename)
% Load images.
%clear all;

% cargamos las carpetas de las imagenes
% datasets crea un vector con los nombres
%          1          2unorder          3          4
datasets=['Mision1','Mision2','Mision3','Mision4','Mision5',...
          'Mision6','Mision7','Mision8','Mision9','Mision10',...]
```

Gráfico 1-2: Código para cargar los archivos de las imágenes

Realizado por: Tacuri & Olmedo, 2018

El segundo paso es encontrar pares imágenes, este proceso se realiza registrando pares de imágenes, detectando características entre $I(n)$ y $I(n-1)$, estimando la transformada geométrica de $I(n)$ a $I(n-1)$ y calculando la misma.

Para empezar con este paso se convierte las imágenes a escala de grises, y utilizando el detector de puntos de interés SURF, para encontrar los puntos característicos de la imagen, y la función matchFeatures para encontrar la semejanza entre los puntos de interés de las imágenes, luego se realiza una estimación de la transformada geométrica entre los puntos antes encontrados y finalmente calculamos la transformada para crear la imagen panorámica.

2.8.3.2. Función interfaz.m

Esta es la función principal del programa debido a que contiene el código de la interfaz gráfica con el usuario y así como el proceso de segmentación de color y el cálculo de los valores de ocupación vegetal.

Dentro de esta se invoca a la función código.m para realizar la reconstrucción de la imagen. Una vez obtenida el resultado de dicha función, obteniendo la imagen sobre la cual realiza el proceso de segmentación de color.

```

#LERNJENQO ENLIIJ DINSJIS CON J QONQO ENJF EJ COJOL A CASO QONQO NO

B=TW(::'j)>=(COJLE[B-FQJ]EJW(::'j)<=(COJLE[B+FQJ]:
C=TW(::'S)>=(COJLE[Q-FQJ]EJW(::'S)<=(COJLE[Q+FQJ]:
E=TW(::'J)>=(COJLE[B-FQJ]EJW(::'J)<=(COJLE[B+FQJ]:
#ENCOCUJENQO JOR COJOLEQ QENLLO QOJ JURELAVIC

COJLE[B=QONQJE(BTX(I'J))I
COJLE[Q=QONQJE(BTX(I'S))I
COJLE[E=QONQJE(BTX(I'J))I: #QENLJENQO JOR COJOLEQ QOJ JURELAVIC
COJ=ENLJENQO ENLIIJ
#QENLJENQO JOR COJOLEQ QOJ JURELAVIC

```

Gráfico 2-2: Código para realizar segmentación de imagen

Realizado por: Tacuri & Olmedo, 2018

En esta parte del código realiza la segmentación de color, utilizando la técnica de similitud, teniendo en cuenta la tolerancia ingresada y el color seleccionado por el usuario.

2.8.3.3. Función graficar.m

Esta sirve para realizar la subdivisión del mapa en sectores más pequeños, además grafica las líneas que sirven para visualizar la división. También en esta función calcula el valor en porcentaje de la ocupación vegetal en los sectores en los cuales se dividió, este valor es con respecto al valor global calculado en el primer resultado.

```

%graficar lineas
a=di;
b=fila;
s=[];
%hold on
for i=1 : n
    plot([a,a],[1,rows],'r')
    a=a+di;
end
for i=1 : m
    plot([1,columns],[b,b],'r')
    b=b+fila;

```

Gráfico 3-2: Código para graficar líneas

Realizado por: Tacuri & Olmedo, 2018

Con el código grafica las líneas de color rojo sobre el mapa de ocupación vegetal obtenido en la segmentación, en función de los datos ingresados por el usuario y usando estos recortes de imágenes para calcular el valor de vegetación existente en esa área, presentando así de mejor manera, los resultados obtenidos.

2.8.4. Matlab y Guide

Para garantizar una buena comunicación con el usuario, y lograr un entorno sencillo de manejar, se procede a realizar una interfaz gráfica, en la cual el usuario pueda ingresar los parámetros de una forma intuitiva y sencilla, y poder obtener los resultados esperados de manera efectiva.

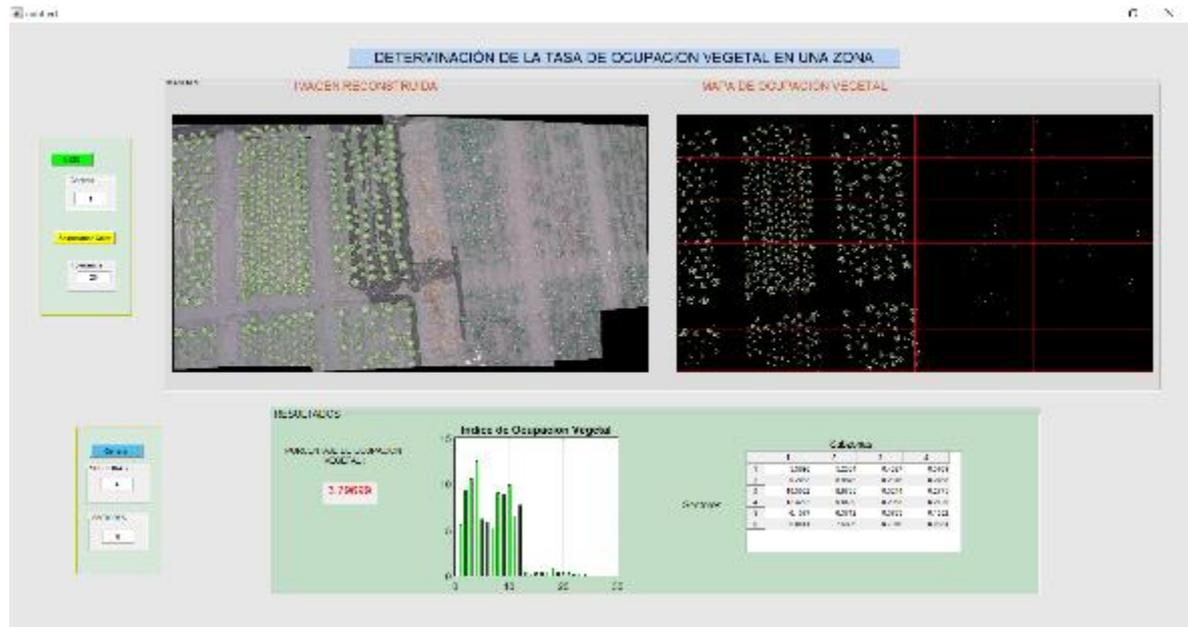


Figura 32-2: Interfaz para la determinación de la tasa de ocupación vegetal

Realizado por: Tacuri & Olmedo, 2018

La primera parte está compuesta por dos botones y dos static text para el ingreso de los parámetros.

El primer botón (INICIO), sirve para iniciar el procedimiento de reconstrucción de la imagen, para esto es necesario ingresar antes el numero de la misión, se ingresa un número, el cual corresponde a una carpeta donde están almacenadas las imágenes capturadas en una misión. El segundo botón (SELECCIONAR COLOR), sirve para seleccionar el área y el color a segmentar dentro de la imagen reconstruida, para esto se ingresa primero un valor de tolerancia comprendido entre 0 y 100, el cual permite ajustar el rango de pixeles que se segmentaran.

En la parte superior de la ventana visualiza las imágenes resultantes siendo la del lado izquierdo el resultado del algoritmo de Stitching (Imagen Reconstruida), y la del lado derecho, el resultado del algoritmo de segmentación (Mapa de Ocupación Vegetal).



Gráfico 4-2: Botones de control del programa

Realizado por: Tacuri & Olmedo, 2018

Al presionar el botón (Seleccionar Color), puede seleccionar de manera manual el área y color que desee segmentar dentro de la imagen resultante del proceso de reconstrucción. Esto dará como resultado una imagen que contiene solo los pixeles que entran en el rango de tolerancia propuesto; así también, entregara un valor el cual es el porcentaje de ocupación vegetal que existe en esa zona, el cual será mostrado en el grupo de resultados.



Gráfico 5-2: Botones de subdivisión y Muestra de Resultados

Realizado por: Tacuri & Olmedo, 2018

Dentro de la segunda etapa del proceso tiene un botón (Generar), el cual permite sub dividir, el mapa de ocupación vegetal en sectores y subzonas, columnas y filas respectivamente; esto con el propósito de obtener un nuevo valor de la tasa de ocupación vegetal, específico para un área más pequeña de la zona, en relación al área global, esto permite tener una mejor visión de los lugares donde existen menos densidad de vegetación.

Los resultados obtenidos, son mostrados en el grupo resultados, siendo el primer valor desde la izquierda el porcentaje de ocupación vegetal de toda la zona en general; los resultados de la subdivisión se muestran mediante un diagrama de barras, permitiendo interpretar de mejor manera

los valores resultantes; finalmente presenta los valores en porcentajes de la ocupación vegetal para cada sector y subzona.

Realizando una breve interpretación de los resultados, y una inspección visual, puede determinar las áreas que poseen el más bajo y el más alto índice de ocupación vegetal, y con esto poder establecer una mejor distribución de recursos a estos sectores.

2.8.5. Obtención de la tasa de ocupación Vegetación

2.8.5.1. Segmentación de Imagen

Para obtener la tasa de ocupación vegetal, una vez logrado reconstruir la zona donde se planea realizar el análisis, se procede a calcular el porcentaje existente de vegetación en la misma. Para esto se utiliza la segmentación de imágenes, que es el proceso de separar zonas, pixeles u objetos con características definidas dentro de una imagen.

En este caso utiliza la segmentación basada en similitud de color. Esto consiste en seleccionar un pixel de color específico al cual se lo compara con todos los pixeles de la imagen y si los valores en los tres canales, están dentro de un rango que es definido por el valor de tolerancia, son considerados semejantes e ingresan a una máscara donde serán almacenados.

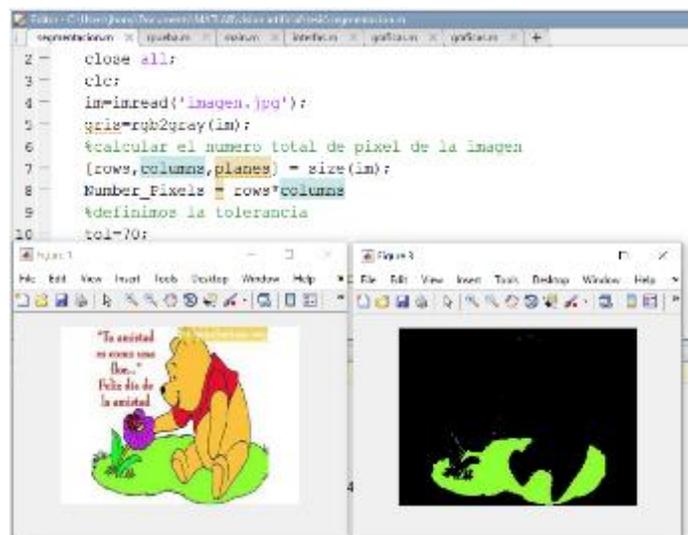


Gráfico 6-2: Pruebas de similitud de similitud de color

Realizado por: Tacuri & Olmedo, 2018

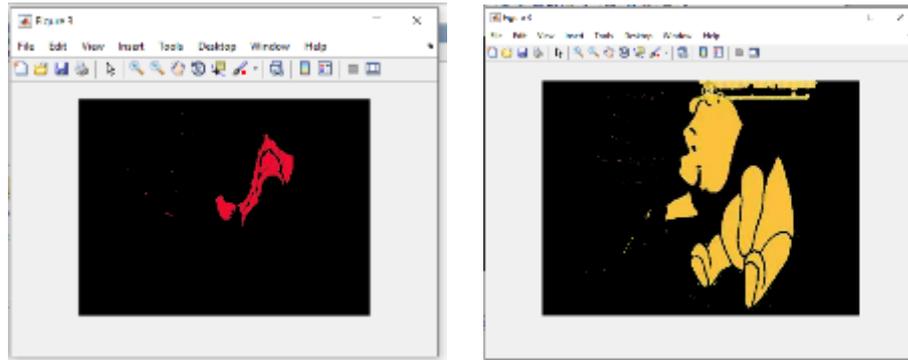


Gráfico 7-2: Pruebas de similitud cambiando el pixel de comparación

Realizado por: Tacuri & Olmedo, 2018

2.8.5.2. Descripción de funcionamiento

Mediante este proceso separa los colores específicos dentro de una imagen, para este caso el color verde, pudiendo ajustar los valores de tolerancia para obtener un resultado más preciso.

En este caso particular se trabaja con tolerancias entre 30 % y 40 % logrando un resultado confiable, mucho depende de la calidad de la imagen obtenida en la reconstrucción. Primero se selecciona mediante el cursor el pixel del color que se desea segmentar, el programa realizara el procedimiento necesario y nos mostrara la imagen resultante, si desea puede volver a seleccionar otro color, actualizando el resultado mostrado cada vez.

2.8.6. Cálculo del porcentaje de ocupación vegetal

Una vez logrado separar los pixeles que representan vegetación dentro de la imagen, se procede a realizar el cálculo para estimar el porcentaje de vegetación dentro de la imagen.

Determinando la escala de la imagen utilizando las siguientes formulas:

$$E = \frac{H + hv}{f}$$

$$vegetación = \frac{n}{r} \times E \times fc$$

Donde:

- E: escala de la imagen
- H: altura desnivel del terreno
- hv: altura de vuelo del dron sobre el terreno
- f: distancia focal de la cámara
- n: número de píxeles segmentados
- r: resolución de la imagen en píxeles por pulgada (ppp)
- fc: factor de conversión de pulgadas a m
- vegetación: área de ocupación vegetal

Obteniendo con este cálculo el área aproximada de ocupación vegetal dentro de la zona en análisis, en metros cuadrados, y obteniendo también el porcentaje en relación al área real que se está analizando.

Para un mejor análisis de resultados e interpretación de este valor se divide a toda la zona en sectores más pequeños que permite conocer de mejor manera en que parte se encuentra la mayor y menor concentración de vegetación.

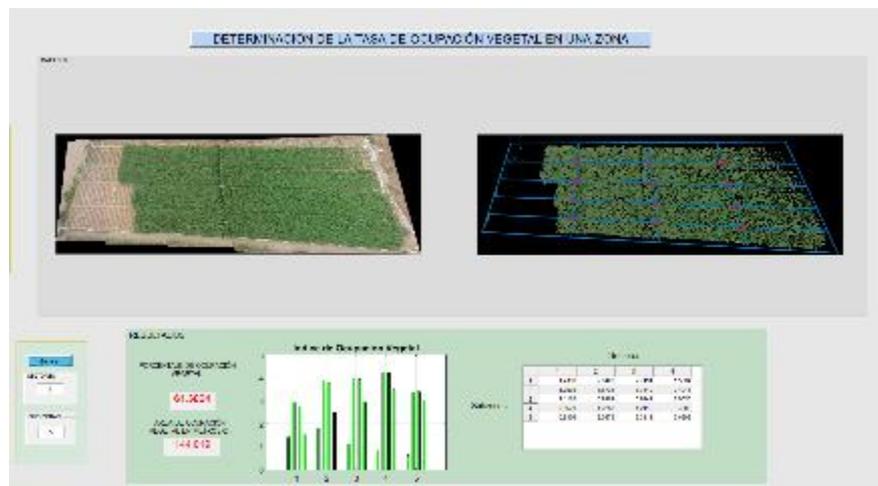


Gráfico 8-2: Subdivisión del mapa de ocupación vegetal en áreas pequeñas

Realizado por: Tacuri & Olmedo, 2018

Permitiendo elaborar estrategias más eficientes de repartición de los recursos agrícolas, puesto que se conoce los lugares donde existe menor densidad de vegetación.

CAPÍTULO III

3. EVALUACIÓN DE RESULTADOS

En este capítulo se describirá el procedimiento a seguir para la captura de datos y el análisis de resultados obtenidos, usando el vehículo aéreo no tripulado implementado. Las pruebas fueron realizadas en un entorno real, con condiciones ambientales óptimas, y tomando todas las seguridades necesarias para verificar cada uno de las etapas que componen el presente proyecto.

3.1. Parámetros de vuelo

Unos de los aspectos principales del proyecto fue determinar los parámetros óptimos de vuelo. El tiempo máximo de vuelo es de 10 minutos en promedio, utilizando una batería de 4s, 3000 mAh, y una tasa de descarga de 30 - 40 C. Este tiempo depende de factores como la velocidad de desplazamiento, la ruta a seguir y velocidad del viento, entre otras condiciones ambientales.

Debido a esto la planificación de la misión no puede superar los 8 minutos de vuelo, si al superar este tiempo, corre el riesgo de ocasionar un accidente con el dron.

La velocidad de vuelo del dron se lo realizo a 2 m/s, la captura de imágenes se lo realizo mediante la función de temporización en un lapso de 3 segundos, con lo cual asegura que las imágenes obtenidas tendrán un solapamiento mayor al 70%.

3.2. Características del sistema anticollisiones

De acuerdo a las pruebas establecidas con el dron se ha establecido el rango máximo en la detección de objetos en el sensor ultrasónico con una distancia de detección aproximada de 4 metros debido a la velocidad con la que los motores disminuyen sus revoluciones cuando se lleva a efecto el bloqueo del acelerador en el transmisor del dron se puede definir la distancia de separación con el obstáculo en la que el dron efectuó el aterrizaje.

Se puede estimar la pendiente de descenso que describe el dron pudiendo así conocer la altura máxima en la que el sistema será funcional.

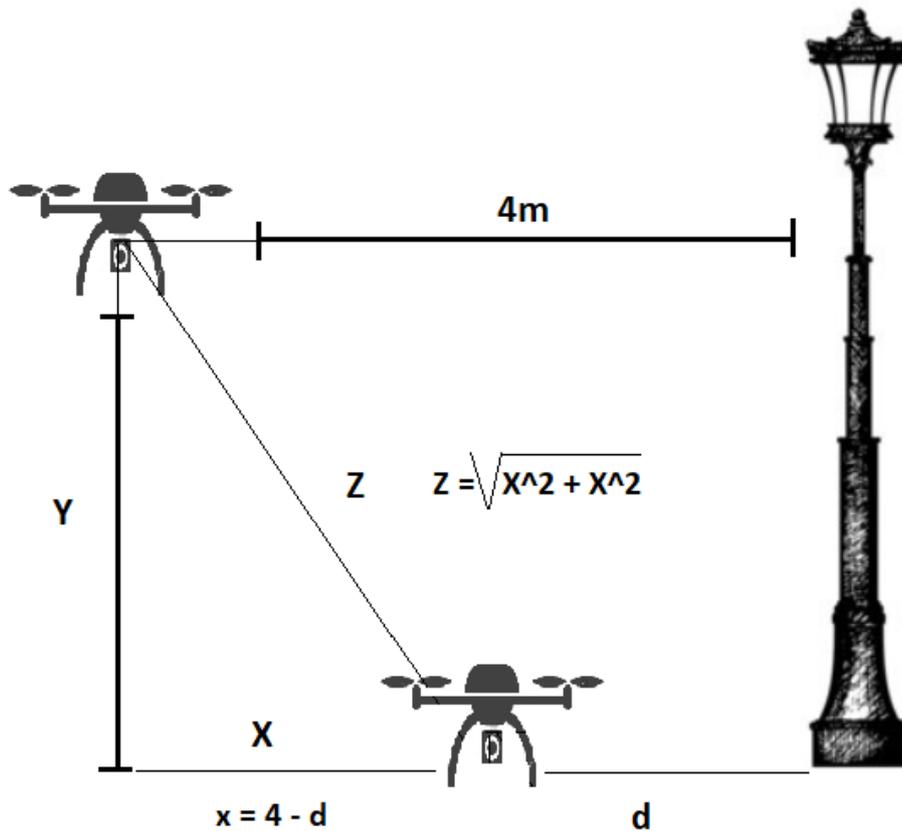


Figura 1-3: Estimación de la distancia de descenso del drone

Realizado por: Tacuri & Olmedo, 2018

Con una calibración de descenso aproximada de 30° se pudo definir el valor de la pendiente y las distancias a las cuales el drone se encontrará al final del proceso. Por lo tanto las estimaciones de las distancias son las siguientes:

Tabla 1-3: Valores tolerables en los cuales el sistema basado en ultrasónico será eficiente

Distancia de detección (metros)	Altura del drone Y (metros)	Distancia X	Distancia de descenso recorrida Z (metros)	Distancia de separación d (metros)	Aceptación con relación a la altura
3	5	2.88	5.77	0.20	NULA
3	4	2.30	4.61	0.70	BAJA
3	3	1.73	3.46	1.27	NORMAL
3	2	1.15	2.30	1.85	ALTA
3	1	0.57	1.15	2.43	ALTA

Realizado por: Tacuri & Olmedo, 2018

Tabla 2-3: Comparación entre los valores estimados y reales del sistema de detección de obstáculos

Altura (metros)	Distancia de separación entre el obstáculo		Tiempo de aterrizaje aproximado(segundos)	Velocidad de descenso (metros/segundo)
	Posición calculada(metros)	Posición real(metros)		
3.5	0.98	1.10	6	0.81
3	1.27	1.35	6	0.70
2,5	1.55	1.62	5	0.67
2	1.85	1.95	4	0.62
1,5	2.13	2.20	3	0.71
1	2.43	2.50	2	0.60
0,5	2.71	2.79	1	0.67

Realizado por: Tacuri & Olmedo, 2018

3.3. Planificación de la Misión

Las pruebas fueron realizadas en la ciudad de Riobamba, en la parroquia Yaruquies, en la imagen se puede apreciar la zona con distintas parcelas donde se realizaron las pruebas.



Figura 2-3: Imagen de las zonas de cultivos de la parroquia de Yaruquies

Realizado por: Tacuri & Olmedo, 2018

Dentro de esta zona se ha seleccionado un terreno, en el cual se va a aplicar el estudio, seleccionando un cultivo de alfalfa con dimensiones de 10x22 metros aproximadamente.

Se realiza una conexión Wifi entre el dron Intel Aero y una computadora en tierra, dentro de la cual se ejecuta el programa QGround Control, este permite configurar y calibrar los sensores y parámetros de vuelo, además que permite trazar la ruta que va a seguir el dron sobre el terreno seleccionado. La altura de vuelo se determina en función del ancho del terreno, para que el ancho del terreno sea cubierto por una sola imagen como se ve en la figura 3-3, y en caso que el ancho sea demasiado grande se determinará de la misma manera, pero con la mitad del ancho del terreno.

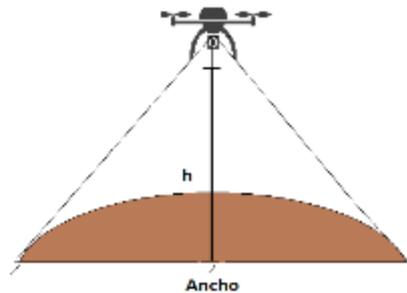


Figura 3-3: Esquema de la altura y área de captura de imágenes

Realizado por: Tacuri & Olmedo, 2018

La finalidad fue definir la trayectoria más sencilla para el dron, es decir una línea recta obteniendo un menor número de imágenes para procesar.

En el caso del terreno seleccionado la altura de vuelo fue de 18 metros sobre el nivel del terreno, y la velocidad de vuelo 2 m/s. La ruta pre establecida fue la que se muestra en la figura 4-3, con un tiempo de vuelo de 3 minutos aproximadamente.



Figura 4-3: Ruta pre establecida a cubrir

Realizado por: Tacuri & Olmedo, 2018

Una vez configurada la ruta, se procede a cargarla en el dron y a ejecutarla para captura de las imágenes, como se muestra en la tabla 3-3, los valores estimados por el programa y los valores reales de desplazamiento son ligeramente menores entre 0-10 centímetros.

Tabla 3-3: Valores medidos en misiones a diferentes alturas

	Altura (metros)	Distancia estimada por el GPS (metros)	Distancia real (Despegue/Aterrizaje) (metros)
MISIÓN 1	1,5	8	7.85
MISIÓN 2	2	8	7.89
MISIÓN 3	2.5	8	7.92
MISIÓN 4	3	7	6.97
MISIÓN 5	3.5	7	6.87

Realizado por: Tacuri & Olmedo, 2018

3.4. Capturas de imágenes

Para la captura de las imágenes se utiliza la cámara deportiva Xiaomi, situado 90° con respecto al dron. Se utilizó la función de time lapse a 3 s, logrando obtener imágenes con un solapamiento aproximado del 60% entre imágenes. Para este ejemplo se obtuvieron capturar una secuencia de 7 imágenes, las cuales fueron almacenadas en la memoria micro SD de la cámara para posteriormente cargarlas a la computadora de manera manual una vez terminada la misión de vuelo.

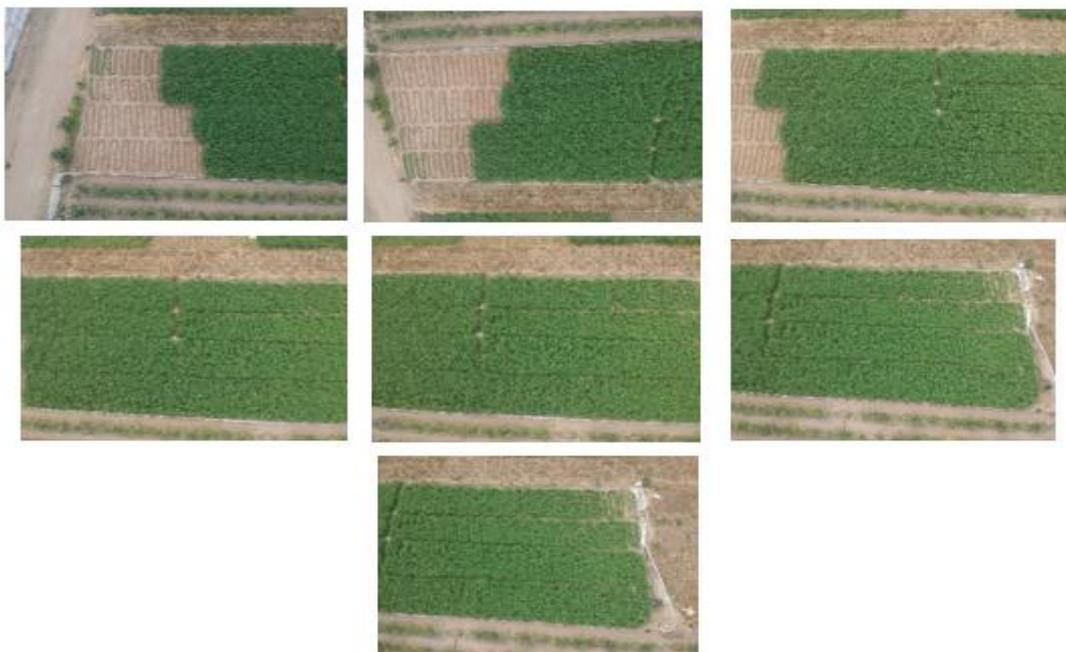


Figura 5-3: Secuencia de imágenes obtenidas en el cultivo de alfalfa

Realizado por: Tacuri & Olmedo, 2018

3.5. Reconstrucción de imagen

El programa de reconstrucción de imágenes en Matlab, realiza una costura de varias imágenes para lograr una imagen panorámica resultante, de la zona en estudio. Para continuar con el ejemplo se procedió a realizar una pre selección de imágenes con la cual elimina imágenes innecesarias que puedan comprometer los resultados finales, en este caso particular las imágenes del inicio y del final del terreno, debido a que no existe una gran diferencia entre una y la otra. La reconstrucción se la realiza con los métodos mencionados en el capítulo anterior

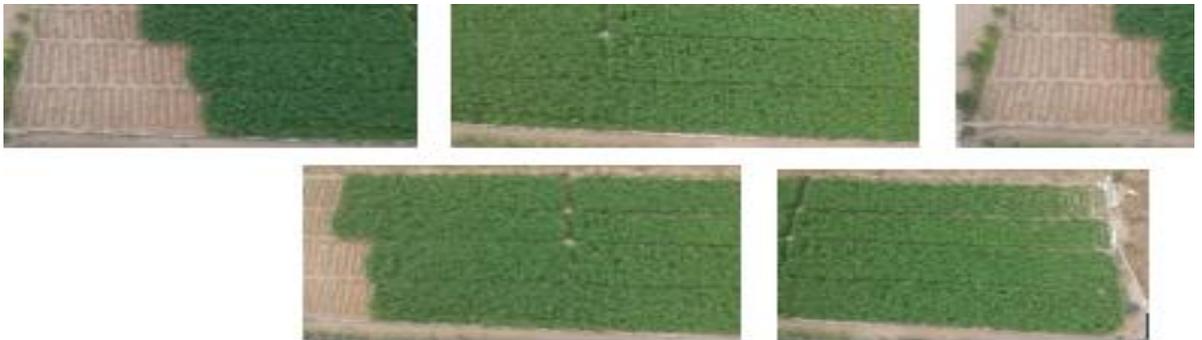


Figura 6-3: Secuencia de imágenes

Realizado por: Tacuri & Olmedo, 2018

En el momento de la reconstrucción se obtienen los siguientes resultados



Figura 7-3: Imagen reconstruida

Realizado por: Tacuri & Olmedo, 2018

Cabe mencionar que el programa soporta cambios en parámetros de rotación, escala y el orden de las imágenes. En la prueba realizada se estima la robustez del algoritmo, procediendo a cambiar el orden de la secuencia inicial de imágenes, obteniendo un resultado similar al original, debido

a que el algoritmo trata de buscar la imagen central dentro de la secuencia y en base a esta, se procede a la unión con las demás imágenes. También se procedió a variar el tamaño y la orientación de una de las imágenes, se utilizaron imágenes de 3968x2976 pixeles y una imagen de 1786x1340 pixeles, con la cual se probó que el tamaño y la orientación de la imagen no afecta al resultado final.

En la tabla siguiente se realizó una comparación entre las diferentes pruebas realizadas.

Tabla 4-3: Comparación de resultados de la reconstrucción de imágenes, variando parámetros.

CARPETA (MISIÓN)	NÚMERO IMÁGENES	PARÁMETRO VARIADO	TIEMPO DE SALIDA (SEGUNDOS)	RESOLUCIÓN DE SALIDA (PIXELES)	RESULTADO
1	4	NINGUNO	00:09.69	2897 x 1508	Aceptable
7	4	DESORDEN	00:10.94	2578 x 1582	Aceptable
8	4	TAMAÑO	00:08.60	2964 x 1585	Aceptable
9	4	ORIENTACIÓN	00:09.29	2567 x 1459	Aceptable

Realizado por: Tacuri & Olmedo, 2018

3.6. Determinación del índice de ocupación vegetal

Una vez culminado las anteriores etapas, se obtuvo una imagen que cubre todo el terreno en cuestión. Para determinar la tasa de ocupación vegetal es necesario introducir en el programa la altura de vuelo de la misión y las dimensiones reales del terreno ya que con estos datos se procedió a calcular el área que es ocupada por la planta de alfalfa. En este ejemplo el área del terreno es de 220 metros cuadrados. Y como se puede apreciar en la imagen las plantas de alfalfa ocupan 139 metros cuadrados aproximadamente del área total del terreno, es decir que, la tasa de ocupación vegetal es del 65%.

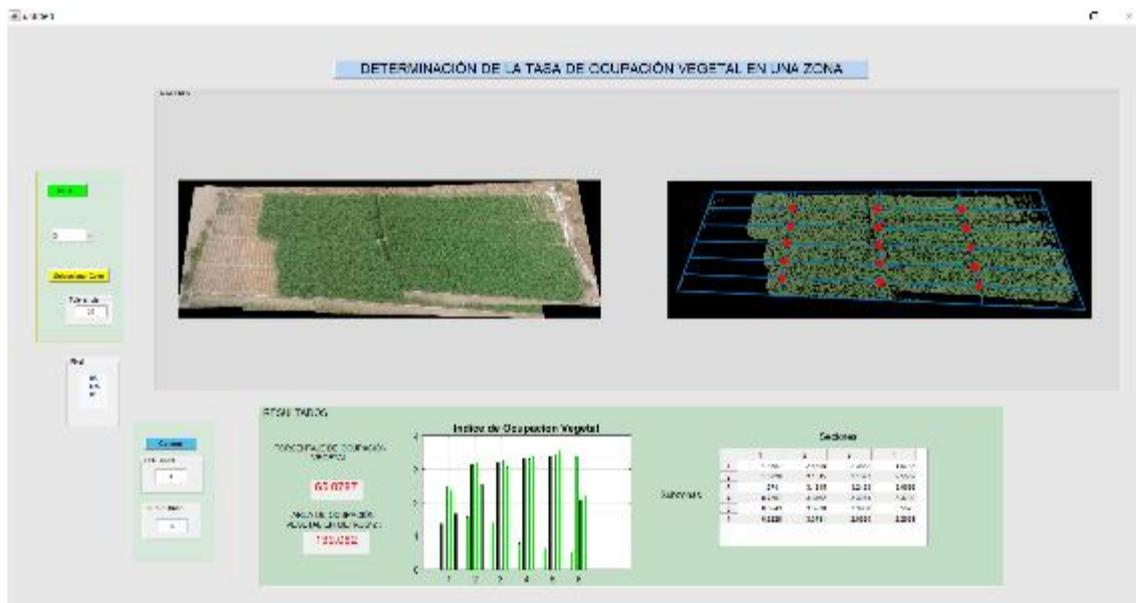


Figura 8-3: Imagen reconstruida y resultados obtenidos

Realizado por: Tacuri & Olmedo, 2018

Además, usando la herramienta de subdivisión, podemos observar el diagrama de barras resultante, donde se puede identificar a simple vista los sectores y la zona donde existe menos ocupación vegetal.

Con esta información podemos informar al agricultor, los sectores en los cuales la producción se ve afectada, para que el afectado lleve a cabo las acciones que crea necesarias.

Se pudieron obtener los valores, de otros cultivos y visualizados los resultados en la siguiente tabla:

Tabla 5-3: Tabla de comparación de resultados de los índices de ocupación vegetal mostrada en el programa.

LUGAR DE VUELO	TIEMPO DE MISIÓN (min)	ÁREA TOTAL medida (m ²)	ALTURA DE VUELO (m)	NUMERO IMÁGENES	ESCALA DE IMAGE N	VEGETACIÓN N (m ²)	ÁREA APROX OCUPADA (%)
Riobamba, Yaruquies, San Vicente	2,53	220	5	6	0,76	110,00	65,07
Riobamba, Yaruquies, Elen	2,11	184	8	4	0,64	134	72,83

Realizado por: Tacuri & Olmedo, 2018

CONCLUSIONES

- Se ha investigado los principales conceptos sobre los vehículos aéreos no tripulados, logrando determinar que un vehículo tipo Quadcopter es la mejor opción para desarrollar el proyecto, debido a que con este tipo puede recorrer el terreno a menores velocidades logrando capturar imágenes más estables y sin desenfoco, además, si se llega a presentar un obstáculo, el vehículo es capaz de efectuar un aterrizaje de emergencia.
- Se logra establecer una comunicación mediante una red wifi entre el dron y una computadora en tierra, usando el protocolo de comunicación MAVLink y el programa QGround Control, consiguiendo pre establecer la ruta que sigue el dron para capturar las imágenes y realizar un monitoreo y control de los parámetros de vuelo.
- Se implementó un sistema anticollisiones con la ayuda de un sensor ultrasónico y los módulos NRF24I01 junto con Arduino, al enviar y recibir datos continuamente sobre la presencia de un posible obstáculo, switchgeando al acelerador del transmisor y reduciendo las revoluciones de los motores hasta desarmarlos.
- Se elaboró un programa que reconstruye una secuencia de imágenes capturadas en vuelo y sin importar su orden, escala u orientación, obtenemos una imagen panorámica resultante, utilizando el programa Matlab y el algoritmo de Stitching.
- Mediante un algoritmo de segmentación y procesamiento de imágenes, se determina el número de pixeles correspondientes a las plantas existentes en un terreno, y utilizando este valor, se logra determinar el área ocupada por el cultivo y el porcentaje de ocupación vegetal en relación al área real del terreno.
- Se verifico que el dron implementado puede seguir una trayectoria pre establecida, además, que las imágenes capturadas presentan condiciones aceptables y que el valor resultante del procesamiento de imágenes, tienen valores proporcionales al aérea total de la zona donde se lleva a efecto el estudio.

RECOMENDACIONES

- Comprobar que las condiciones atmosféricas seas las mejores para realizar un vuelo, sobre todo prestar atención a la velocidad del viento, debido a que afecta significativamente la estabilidad del dron y la duración de la batería, comprobar también todos los componentes del dron principalmente que los conectores de la batería y de toda la placa de control se encuentren bien firmes y conectados correctamente.
- Revisar siempre los valores de GPS y de la calibración de la brújula antes de cada misión, debido a que de estos depende el buen comportamiento del dron durante el vuelo, manteniendo el dron en vuelo unos 30 segundos antes de realizar la misión en modo automático, esto ayuda a comprobar que no existan problemas en el funcionamiento del dron, si se visualiza cualquier comportamiento extraño o mensajes de advertencia en la estación de control de tierra, se debe parar toda acción que se esté llevando a cabo.
- Al momento de ingresar las imágenes a la carpeta de la misión, se recomienda hacer una pre selección de las imágenes para evitar imágenes parasitas que causen un resultado erróneo, evitando imágenes borrosas o desenfocadas por movimientos bruscos del dron, debido al viento u otros factores.
- Al seleccionar el color, siempre intentar seleccionar un lugar donde la vegetación sea más densa y siempre con una tonalidad más oscura, para garantizar una correcta segmentación de la vegetación con el terreno, además de ajustar el valor de tolerancia para que el resultado pueda abarcar la mayor cantidad de pixeles similares al seleccionado.
- Se puede utilizar algoritmos de segmentación y procesamiento de imágenes más complejos, que permitan obtener resultados más precisos, y que permitan obtener mayor información de las imágenes capturada para lograr elaborar estrategias de agricultura de precisión.
- Mantener siempre un control y monitoreo tanto de forma visual como en la estación de control de tierra antes, durante y después de ejecutar una misión en modo automático, para garantizar la integridad del vehículo, así como de las personas presentes, evitando así posibles accidentes.

BIBLIOGRAFÍA

Addati, Gastón. A., & Pérez Lance, Gabriel. *Introducción a los UAV's, Drones o VANTs de uso civil* [en línea], (2014), (Argentina - Buenos Aires), pp.1-12 [Consulta: 1 noviembre 2018], No. 551(2014), Disponible en: <https://www.econstor.eu/handle/10419/130802>

Algoritmo de segmentación, asignatura.us.es [en línea]. Tema5-2_SegmentacionRegionesUmbralizacion [Consulta: 12 de abril de 2018]. Disponible en: http://asignatura.us.es/imagendigital/Tema5-2_SegmentacionRegionesUmbralizacion.pdf

Ballesta Mónica, Gil Arturo, Reinoso Oscar, & Payá Luis. “Evaluación de Detectores de Puntos de Interés para SLAM Visual”. XXVIII Jornadas de Automática. [en línea]. 2007.(España) (p. 9). [Consulta: 1 de abril del 2018]. ISBN: 978-84-690-7497- 8. Disponible en: <https://docplayer.es/64197965-Evaluacion-de-detectores-de-puntos-de-interes-para-slam-visual.html>

Ballesta, M., Gil, A., Reinoso, O., & Úbeda, D. “Análisis de Detectores y Descriptores de Características Visuales en SLAM en Entornos Interiores y Exteriores.” *Revista Iberoamericana de Automática e Informática Industrial RIAI*, [en línea], (2012), 9(4), pp. 68-80, [Consulta: 19 julio 2018], Disponible en: [https://doi.org/10.1016/S1697-7912\(10\)70027-3](https://doi.org/10.1016/S1697-7912(10)70027-3)

Barreiro Elorza Pilar., & Valero Ubierna Constantino. “Drones en la agricultura”. *Tierras de Castilla y León: Agricultura*, [en línea], (2014). España, 220, pp. 36-42., [Consulta: 19 julio 2018], ISSN 1889-0776. Disponible en: <http://oa.upm.es/32561/>

Borra, I., Peña Barragán J. M., Torres Sánchez J, & López Granados F. “Evaluación de la técnica de Resampling en imágenes UAV para la detección de malas hierbas mediante análisis OBIA.” *En XV Congreso de la Sociedad Española de Malherbología: La Malherbología y la transferencia tecnológica*, [en línea], (2015). Sevilla, 209. Junta de Andalucía. Consejería de Agricultura, Pesca y Desarrollo Rural. Recuperado de <https://idus.us.es/xmlui/handle/11441/32507>

Características de cámara, Intel, [en línea] Especificaciones de la Cámara Intel® RealSense™ R200. [Consulta: 1 de mayo de 2018], Disponible en: <https://www.intel.com/content/www/xl/es/support/articles/000016214/emerging-technologies/intel-realsense-technology.html>

Características de cámara, Intel®, [en línea] RealSense™ Data Ranges | Intel® Software. [Consulta: 7 de julio de 2017]. Disponible en: <https://software.intel.com/en-us/articles/intel-realsense-data-ranges>

Características de cámara, Intel®, [en línea] Introducing the Intel® RealSense™ R200 Camera (world facing) | Intel® Software. [Consulta: 7 de julio de 2017]. Disponible en: <https://software.intel.com/en-us/articles/realsense-r200-camera>

Características del programa, elightech Systems [en línea] Sistema de Control de Tierra (GCS). [Consulta: 23 de agosto de 2017], Disponible en: <http://flighttechspanish.weebly.com/control.html>

clarity-from-above-pwc. PwC global report on the commercial applications of drone technology [Consulta: 1 julio 2018]. Disponible en: <https://www.pwc.pl/pl/pdf/clarity-from-above-pwc.pdf>

Díaz Celis Cesar Augusto. Adquisición de imágenes de bajo costo aplicadas a la agricultura de precisión usando vehículos aéreos no tripulados (Tesis)(maestrías). Universidad San Francisco de quito Recuperado, Quito-Ecuador. 2013. pp. 9-15. [Consulta: 1 julio del 2018]. Disponible en: <http://repositorio.usfq.edu.ec/handle/23000/1665>

Domínguez Torres, A. (1996). “Procesamiento digital de imágenes”. *Perfiles Educativos*, [en línea], 1996, México, (72). [Consulta: 10 de julio del 2017]. ISSN: 0185-2698
Disponible en: <http://www.redalyc.org/resumen.oa?id=13207206>

Espinoza, Manolo, Andrade Rafael, Rojas Jorge & Falcón Vladimir, “Tecnologías de la información y comunicación en la agricultura”. *Revista de Ciencia, Tecnología e Innovación*, [en línea], 2017, (Ecuador), 4(1), pp. 1-12, [Consulta: 1 marzo del 2018], ISSN 1390-9150 Disponible en: <https://scholar.google.com.mx/citations?user=pnKBPeEAAAAJ&hl=en>,

Estrada, Ailyn, & Fernández Tatiana, “Tecnologías de la información y la comunicación para el desarrollo sostenible”. *UCE Ciencia. Revista de postgrado*, [en línea] .2017, (República Dominicana), 5(1). [Consulta: 1 marzo del 2018]. ISSN 2306-3556. Disponible en: <http://uceciencia.edu.do/index.php/OJS/article/view/98>

Flores Pablo, & Braun Juan. *RANSAC: fundamento teorico*. [en línea] [Consulta: 16 de abril del 2018]. Disponible en:

<https://iie.fing.edu.uy/investigacion/grupos/gti/timag/trabajos/2011/keypoints/FundamentoRAN SAC.pdf>

Flores Pablo, & Braun Juan. *Algoritmo SIFT: fundamento teórico.* [en línea] [Consulta: 5 de diciembre de 2017]. Disponible en: <https://iie.fing.edu.uy/investigacion/grupos/gti/timag/trabajos/2011/keypoints/FundamentoSIFT.pdf>

Fundación Elika, “Drones y sus usos en agricultura” *Berezi* 35, [en línea], (2014). Vasco, pp. 1-14., [Consulta: 21 julio 2018]. Disponible en: <https://agricultura.elika.eus/wp-content/uploads/articulos/Archivo1388/Berezi%2035%20drones%20y%20sus%20usos%20en%20agricultura.pdf>

García-Cervigón, J. J. Estudio de Índices de vegetación a partir de imágenes aéreas tomadas desde UAS/RPAS y aplicaciones de estos a la agricultura de precisión. [en línea](Tesis)(maestría) Universidad de Madrid, (España). 2015. pp. 10-98, [Consulta: 1 marzo del 2018]. Disponible en: http://eprints.ucm.es/31423/1/TFM_Juan_Diaz_Cervignon.pdf

Giusiano Mónica Noemi. Drones agrícolas en el manejo sitio-específico de malezas, análisis de beneficios económicos [en línea](Tesis)(investigación). Universidad Siglo 21, (Córdoba). 2017. pp 10-81. [Consulta: 1 de marzo del 2018] Disponible en: <https://repositorio.uesiglo21.edu.ar/bitstream/handle/ues21/14078/GIUSIANO%20MONICA.pdf?sequence=1&isAllowed=y>

Guerrero Albán, Wilfrido Rolando. “Revista Tecnológica-ESPOL” *Detección y descripción de puntos característicos en imágenes multispectrales utilizando esquemas clásicos.* [en línea], (2014). (Ecuador), 27(1). pp. 1-10. [Consulta: 1 noviembre 2018], ISSN 1390-3659. Disponible en: <http://www.rte.espol.edu.ec/index.php/tecnologica/article/view/285/208>

Chagoya Ramos Ena. (2008, julio 1). *Métodos y técnicas de investigación.* [en línea]. México. 1 de julio del 2008, [Consulta: 1 de julio del 2017] Disponible en: <https://www.gestiopolis.com/metodos-y-tecnicas-de-investigacion/>

Instrucciones de uso de controlador de vuelo, ardupilot.org [en línea], Mission Planner Overview — Mission Planner documentation. [Consulta: 31 de agosto de 2018], Disponible en: <http://ardupilot.org/planner/docs/mission-planner-overview.html>

Instrucciones de uso de programa, dronecode [en línea] QGroundControl Guía del usuario. [Consulta: 13 de abril de 2018], Disponible en: <https://docs.qgroundcontrol.com/en/>

Instrucciones del uso de drone, github .com, [en línea]. BitBake, Intel® Aero Platform for UAVs. [Consulta: 15 marzo del 2018] Disponible en: <https://github.com/intel-aero/meta-intel-aero>

Librería de procesamiento de imágenes, VLFeat - Home. [en línea]. [Consulta: 5 de diciembre de 2017]. Disponible en: <http://www.vlfeat.org/>

Martínez Rodríguez, A. Sistema de procesamiento de imágenes RGB aéreas para agricultura de precisión [en línea](Tesis). (ingeniería) Universidad Central Marta Abre de Las Villas, Facultad de Ingeniería Eléctrica, Departamento de Automática y Sistemas Computacionales. (Santa Clara). 2016. pp. 30-62. [Consulta: 15 marzo del 2018]Disponible en: <http://dspace.uclv.edu.cu/handle/123456789/6341>

Martínez Rodríguez, A. Sistema de procesamiento de imágenes RGB aéreas para agricultura de precisión [en línea] (tesis)(diplomado) Universidad Central “Marta Abreu” de Las Villas, 2016, pp. 62, [Consulta: 19 julio 2018], Disponible en: <http://dspace.uclv.edu.cu/bitstream/handle/123456789/6341/Aniel%20Martinez%20Rodriguez.pdf?sequence=1&isAllowed=y>

Murilo Belluzzo A. *collision-avoidance-library: A framework for testing and benchmarking collision avoidance strategies. (2018a). C++, Intel Corporation.* [en línea]. USA. 2016. [Consulta: 1 julio del 2018] Disponible en: <https://github.com/intel/collision-avoidance-library>

Ortiz, M. *Procesamiento digital de imágenes. Benemérita universidad Autónoma de Puebla.* [en línea]. México, 2013 [Consulta 16 de abril del 2018]. Disponible en: <http://www.cs.buap.mx/~mmartin/notas/PDI-MM-Rev.2013.pdf>

Palomino Nora, & Concha Ulises. “Técnicas de Segmentación en Procesamiento Digital de Imágenes.” *Revista de Investigación de Sistemas E Informática*, [en línea] 2009. (Lima-Perú) 6(2), pp. 9-16. [Consulta: 12 de abril del 2018]. ISSN: 1816-3823. Disponible en: <http://revistasinvestigacion.unmsm.edu.pe/index.php/sistem/article/view/3299>

Raffaello D'Andrea. "Guest Editorial Can Drones Deliver?" *IEEE Transactions on Automation Science and Engineering*, 11(3), (2014). pp. 647-648. [Consulta: 1 de julio del 2018], ISSN: 1558-3783. Disponible en: <https://ieeexplore.ieee.org/document/6827242>

Sánchez Iglesias Joan-Carles, Diseño de un dron para la lectura de tarjetas RFID. [en línea]. (Tesis).(ingeniería). Universidad Politécnica de Catalunya. España. (2016). pp. 2-60. [Consulta: 10 de julio de 2017], Disponible en: <https://upcommons.upc.edu/handle/2117/88806>

Santana, Ernesto. *Tipos de drones - Conoce todos los tipos de drones que existen* [Blog]. [Consulta: 10 de julio de 2017], Disponible en: <http://www.xdrones.es/tipos-de-drones-clasificacion-de-drones-categorias-de-drones/>

Sandino Mora Juan David. Desarrollo de proyectos en monitoreo de cultivos agrícolas a través de la visión de máquina. [en línea](Tesis). (ingeniería). Universidad militar nueva granada, Facultad de ingeniería. (Bogotá).2014. pp: 10-56. [Consulta: 15 de abril del 2018] Disponible en: <http://repository.unimilitar.edu.co/handle/10654/13158>

Serrano Cordero, C., & Pérez Curutchet, O. Análisis, diseño estructural y construcción de un dron para la detección de minas antipersona [en línea](Tesis)(Ingeniería) Universitat Politècnica de Catalunya. (España). 2016. pp. 15-58. [Consulta: 19 de abril del 2018]. Disponible en: <https://upcommons.upc.edu/handle/2117/102000>

Sistema operativo ros, ROS.org [en línea] Acerca de ROS. [Consulta: 12 de abril de 2018] Disponible en: <http://www.ros.org/about-ros/>

Sistema operativo ros, ROS.org [en línea]. Componentes principales. [Consulta: 12 de abril de 2018] Disponible en: <http://www.ros.org/core-components/>

Victoria de Lecea Cristina. Diseño y desarrollo de modelos y sistemas de comunicaciones para vehículos aéreos no tripulados (UAV) y drones [en línea](Tesis)(ingeniería) Universidad Politécnica de Madrid, ETSI Telecomunicación. (España). 2016. pp. 10-58 [Consulta: 19 de abril 2018] Disponible en: <http://oa.upm.es/43261/>

ANEXOS

Anexo A: Algoritmo de detección y evasión.

- Para realizar combinaciones entre los diferentes detectores y las estrategias de evasión logrando así personalizar el algoritmo a nuestras necesidades, primero cargamos una estructura simple, con memory guiamos nuestros punteros a objetos y con common/common obtenemos las estructuras básicas de las bibliotecas

```
# include <iostream>
# include <memory>
# include "common/common.hh"
Using namespace std;
int main(int argc, char **argv)
{
    cout << "Mi aplicación personalizada" << endl;
    return 0;
}
```

- Para cargar un vehículo utilizamos:

```
#include "vehicles/MavQuadCopter.hh"
```

- Dentro de la función main() instanciamos el vehículo:

```
shared_ptr<MavQuadCopter> vehicle = make_shared<MavQuadCopter>();
```

- Para cargar el sensor, en este caso la cámara R200:

```
#include "common/DepthCamera.hh"
```

```
#include "sensors/RealSenseCamera.hh"
```

- Para inicializar el sensor a una resolución de 640x480 a 30 fps.

```
shared_ptr<DepthCamera> sensor = make_shared<RealSenseCamera>(640, 480, 30);
```

- Para cargar un detector utilizaremos:

```
#include "detection/DepthImageObstacleDetector.hh"
```

- Para cargar el detector en este caso *Depth Image Obstacle Detector*, siendo el mismo caso para cualquier detector mencionado anteriormente o crear uno nuevo; el valor es el umbral en metros si es mayor a este el objeto será ignorado.

```
shared_ptr<Detector<DepthCamera>> detector =
    make_shared<DepthImageObstacleDetector>(sensor, 5.0);
```

- Para cargar una estrategia de evasión usamos:

```
#include "avoidance/QuadCopterStopAvoidance.hh"
```

- Para cargar la estrategia de evasión, en este caso se usa *Quadcopter Stop Avoidance*.

```
shared_ptr<CollisionAvoidanceStrategy<MavQuadCopter>> avoidance =
    make_shared<QuadCopterStopAvoidance>(vehicle);
```

- Para ejecutar el algoritmo compuesto por Intel RealSense, detector de obstáculos basado en 'Blob extraction' y Stop para la estrategia de evasión.

```
./coav-control -d DI_OBSTACLE -a QC_STOP -s ST_REALSENSE
```

- Salida del comando

```
./coav-control --help
```

Anexo B: Código del programa de reconstrucción de imágenes y posterior segmentación de color del mismo.

- La primera parte que se realiza es la carga de imágenes y el posterior cambio de tamaño del mismo

```

1  function [panoramas]=codigo(filename)
2
3  % Load imagenes.
4  %clear all;
5
6  % cargamos las carpetas de las imagenes
7  % datasets crea un vector con los nombres
8  %          1      2unorder      3          4          5
9  datasets={'Mision1','Mision2','Mision3','Mision4','Mision5',...
10           'Mision6','Mision7','Mision8','Mision9','Mision10',...
11           ...%      6          7          8unorder      9      10
12           'ejemplo1','ejemplo'};
13           % 11          %12
14
42
43 %escalamos las imagenes para que no sean mayor a al size_bound y las
44 %guardamos
45 %s.Count contador de imagenes
46 qn=fullfile('C:\','Users','jhony','Desktop','panorama-master','imgo',a,'nuevo');
47 out=char(qn);
48 for i=1:s.Count
49     new_img=read(s,i);
50     new_img1=imresize(new_img,0.45);
51     nombre=['\imagen',sprintf('%d',i)];
52     a=strcat(out,nombre,'.jpg');
53     imwrite(new_img1,a);
54 end

```

Código para cargar y cambiar el tamaño de la imagen

Realizado por: Tacuri & Olmedo, 2018

- El segundo paso es encontrar pares imágenes

```
55 % Lee la primera imagen del grupo.
56 I = read('buildingScene_1');
57
58 % Inicializa la función para I(i)
59 grayImage = rgb2gray(I);
60 points = detectSURFFeatures(grayImage);
61 [features, points] = extractFeatures(grayImage, points);
62
63 % Inicializa todas las transformaciones a la matriz de identidad.
64 numImages = buildingScene.Count;
65 tforms(numImages) = projective2d(eye(3));
66
67 % Inicializamos variable para contener el tamaño de imagen.
68 imageSize = zeros(numImages,2);
69
70 % Iteramos sobre los pares de imágenes restantes
71 for n = 2:numImages
72     % Base de puntos y funciones para I(n-1).
73     pointsPrevious = points;
74     featuresPrevious = features;
75
76     % Leer I(n).
77     I = read('buildingScene_n');
78
79     % Convertir la imagen a escala de grises.
80     grayImage = rgb2gray(I);
81
82     % Cambiar el tamaño de la imagen.
83     imageSize(n,:) = size(grayImage);
84
85     % Detectar y extraer las características SURF para I(n).
86     points = detectSURFFeatures(grayImage);
87     [features, points] = extractFeatures(grayImage, points);
88
89     % Encuentra correspondencias entre I(n) and I(n-1).
90     indexPairs = matchFeatures(features, featuresPrevious, 'Unique', true);
91
92     matchedPoints = points(indexPairs(:,1), :);
93     matchedPointsPrev = pointsPrevious(indexPairs(:,2), :);
94
95     % Estimar la transformación entre I(n) e I(n-1).
96     tforms(n) = estimateGeometricTransform(matchedPoints, matchedPointsPrev, ...
97         'projective', 'Confidence', 95.5, 'MaxNumTrials', 2000);
98
99     % Calcular T(n) * T(n-1) * ... * T(1)
100     tforms(n).T = tforms(n).T * tforms(n-1).T;
101 end
```

Determinación de puntos de interés entre imágenes

Realizado por: Tacuri & Olmedo, 2018

```

% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
%[FileName Path]=uigetfile({'*.jpg*'},'Abrir Imagen')
a=str2double(get(handles.edit3,'String'));
if isnan(a)
errordlg('Error ingrese la tolerancia','Error')
else
FileName ='laguna2.jpg';
Path='C:\Users\jhony\Desktop\panorama-master\';

if isequal(FileName,0)
return
else
axes(handles.axes2)
a=imread(strcat(Path,FileName));
imshow(a);
pix=impixel;
end
im=a;
gris=rgb2gray(im);
%calcular el numero total de pixel de la imagen
[rows,columns,planes] = size(im);
Number_Pixels = rows*columns;
%definimos la tolerancia
tol=handles.edit1;
colrefR=double(pix(1,1)); %definimos los colores de referencia
colrefG=double(pix(1,2));
colrefB=double(pix(1,3));

%encontramos los colores dentro del intervalo
R=im(:,:,1)>=(colrefR-tol)&im(:,:,1)<=(colrefR+tol);
G=im(:,:,2)>=(colrefG-tol)&im(:,:,2)<=(colrefG+tol);
B=im(:,:,3)>=(colrefB-tol)&im(:,:,3)<=(colrefB+tol);

%resultado matriz binaria con 1 donde esta el color y cero donde no
t=R&G&B; % crear matriz binaria
handles.col=t;
n=nnz(t); % numero de elementos distintos de cero
seg=t; % asignamos a cada canal el valor de t
seg(:,2)=t;% a la nueva matriz seg RGB
seg(:,3)=t;
fin=uint8(double(seg).*double(im));
axes(handles.axes3)
handles.finn=fin;
imshow(fin);
title('Vegetacion existente en la zona','FontSize',16);
xlabel('\bfSubZonas','FontSize',12);
ylabel('\bfSectores','FontSize',12);

```

Código para cargar los archivos de las imágenes

Realizado por: Tacuri & Olmedo, 2018

```

% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
% [FileName Path]=uigetfile({'*.jpg'}, 'Abrir Imagen')
a=str2double(get(handles.edit3, 'String'));
if isnan(a)
errordlg('Error ingrese la tolerancia', 'Error')
else
FileName = 'laguna2.jpg';
Path='C:\Users\jhony\Desktop\panorama-master\';

if isequal(FileName,0)
return
else
axes(handles.axes2)
a=imread(strcat(Path,FileName));
imshow(a);
pix=inpixel;
end
in=a;
gris=rgb2gray(im);
%calcular el numero total de pixel de la imagen
[rows,columns,planes] = size(im);
Number_Pixels = rows*columns;
%definimos la tolerancia
tol=handles.edit1;
colrefR=double(pix(1,1)); %definimos los colores de referencia
colrefG=double(pix(1,2));
colrefB=double(pix(1,3));

%encontramos los colores dentro del intervalo
R=im(:, :, 1) >= (colrefR - tol) & im(:, :, 1) <= (colrefR + tol);
G=im(:, :, 2) >= (colrefG - tol) & im(:, :, 2) <= (colrefG + tol);
B=im(:, :, 3) >= (colrefB - tol) & im(:, :, 3) <= (colrefB + tol);

%resultado matriz binaria con 1 donde esta el color y cero donde no
t=R&G&B; % crear matriz binaria
handles.col=t;
n=nnz(t); % numero de elementos distintos de cero
seg=t; % asignamos a cada canal el valor de t
seg(:, :, 2)=t; % a la nueva matriz seg RGB
seg(:, :, 3)=t;
fin=uint8(double(seg).*double(im));
axes(handles.axes3)
handles.finn=fin;
imshow(fin);
title('Vegetacion existente en la zona', 'FontSize', 16);
xlabel('\bfSubZonas', 'FontSize', 12);
ylabel('\bfSectores', 'FontSize', 12);

```

Código para realizar segmentación de imagen

Realizado por: Tacuri & Olmedo, 2018

```

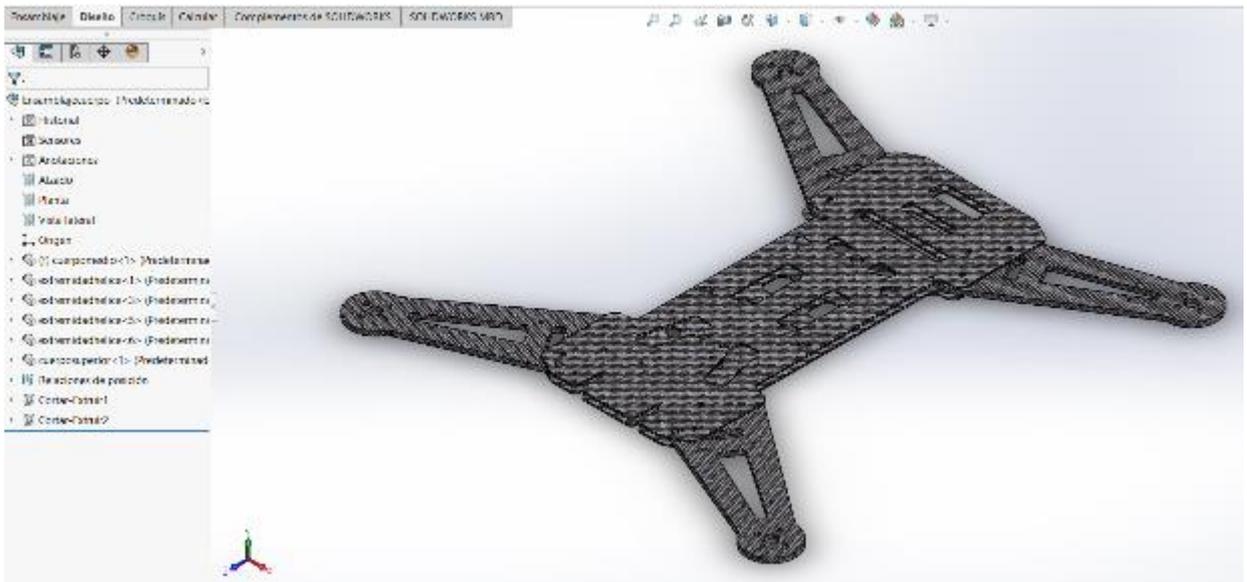
4 | function [porcentaje]=graficar(pix,col, fila, column, Im)
5 |
6 |     n=column; % division en subzonas
7 |     m=fila; % numero de sectores por sub zona
8 |     im=pix; % matriz binaria con los pixeles verdes
9 |
10 |     in2=imread('laguna2.jpg');
11 |     [rows,columns,planes] = size(im);
12 |     % n=b;
13 |     % m=2;
14 |     di=columns/n;
15 |     fila=rows/m;
16 |     %-----
17 |     %graficar lineas
18 |     a=di;
19 |     b=fila;
20 |     s=1;
21 |     %hold on
22 |     for i=1 : n
23 |         plot([a,a],[1,rows], 'r')
24 |         a=a+di;
25 |     -and
26 |     for i=1 : m
27 |         plot([1,columns],[b,b], 'r')
28 |         b=b+fila;

```

Código para graficar líneas y subdividir la imagen

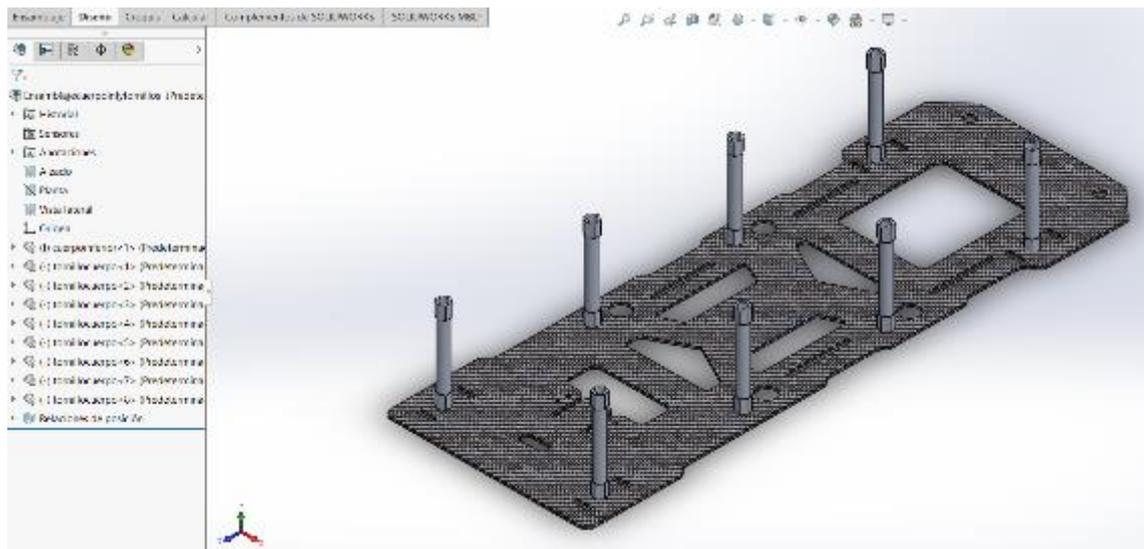
Realizado por: Tacuri & Olmedo, 2018

Anexo C: Estructura del dron realizada en SOLID WORKS



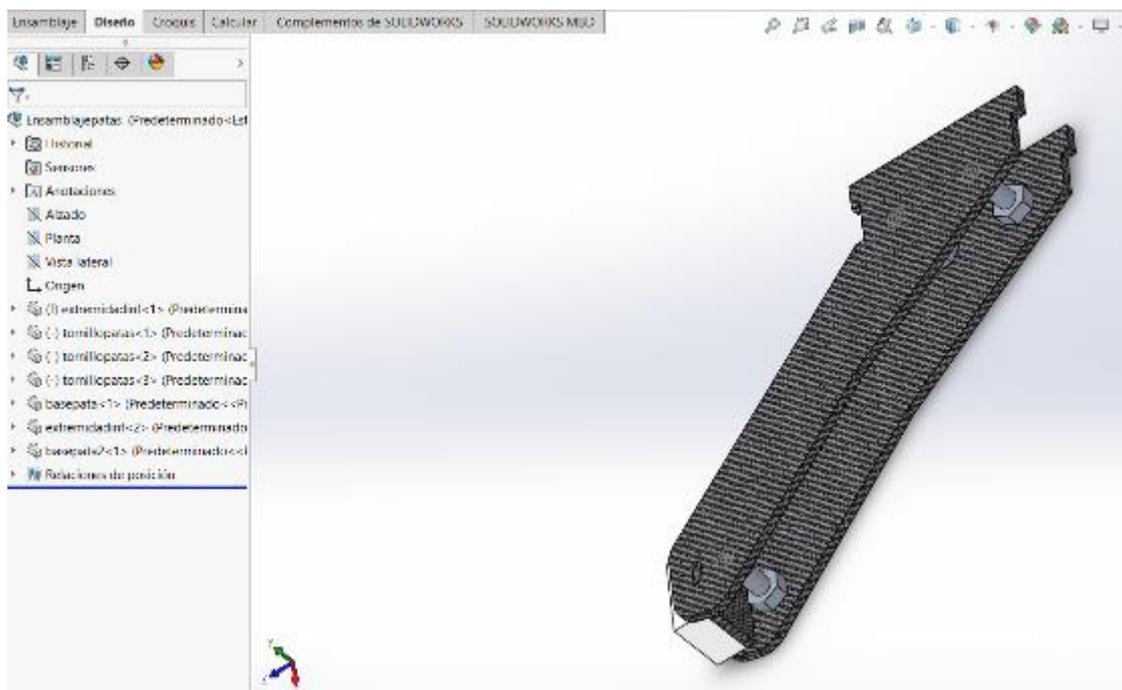
Ensamblaje de las partes superiores del cuerpo

Realizado por: Tacuri & Olmedo, 2018



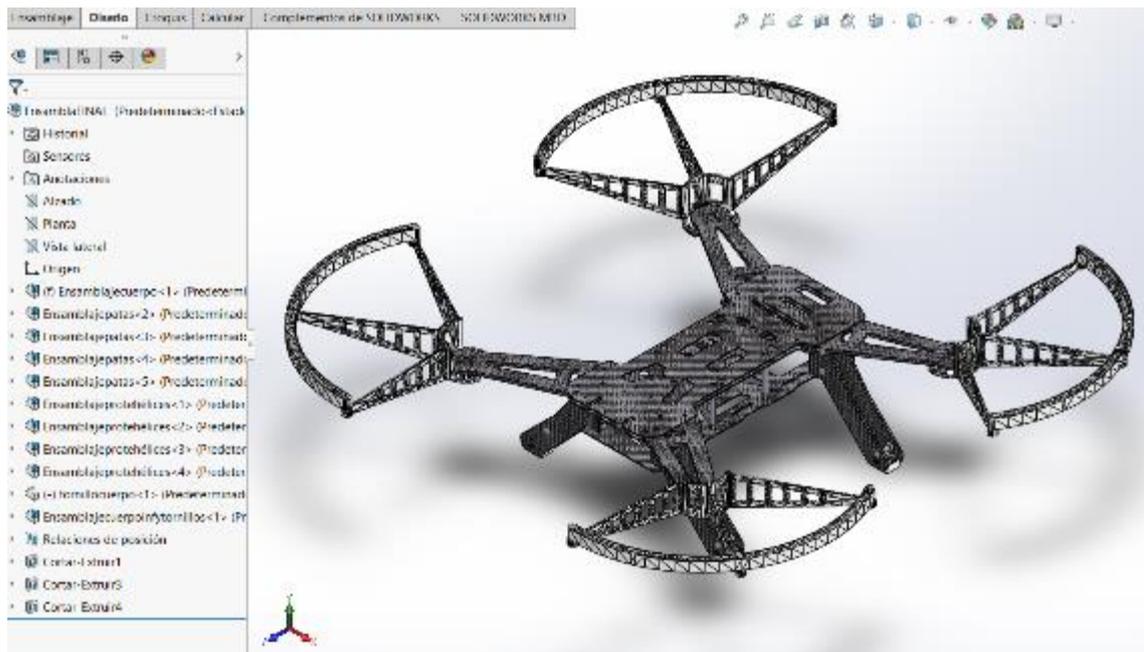
Ensamblaje parte inferior del cuerpo

Realizado por: Tacuri & Olmedo, 2018



Ensamblaje de las patas del dron

Realizado por: Tacuri & Olmedo, 2018



Ensamblaje final del dron incluido protectores de hélices

Realizado por: Tacuri & Olmedo, 2018