



ESPE

UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

Sistema de visión por computador para detectar automáticamente líneas de cultivo y mejorar la precisión del guiado durante misiones autónomas de vuelo de un vehículo aéreo no tripulado

Chasi Taimbud, Christian Stalin

Vicerrectorado de Investigación, Innovación y Transferencia de Tecnología

Centro de Posgrados

Maestría en Electrónica y Automatización Mención Redes Industriales

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

Ing. Minchala Ávila, Luis Ismael PhD.

6 de marzo del 2022

Latacunga



Proyecto titulación_Cristian Chasi.pdf

Scanned on: 14:11 March 10, 2022 UTC



Overall Similarity Score



Results Found



Total Words in Text

Identical Words	864
Words with Minor Changes	0
Paraphrased Words	0
Omitted Words	0



Digitally signed
by LUIS ISMAEL
MINCHALA
AVILA
Date:
2022.03.11
10:20:27 -05'00'



ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

VICERRECTORADO DE INVESTIGACIÓN, INNOVACIÓN Y TRANSFERENCIA TECNOLÓGICA

CENTRO DE POSGRADOS

CERTIFICACIÓN

Certifico que el trabajo de titulación, **“Sistema de visión por computador para detectar automáticamente líneas de cultivo y mejorar la precisión del guiado durante misiones autónomas de vuelo de un vehículo aéreo no tripulado”** fue realizado por el ingeniero **Chasi Taimbud Christian Stalin**, el mismo que ha sido revisado y analizado en su totalidad, por la herramienta de verificación de similitud de contenido; por lo tanto cumple con los requisitos legales, teóricos, científicos, técnicos y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, razón por la cual me permito acreditar y autorizar para que lo sustente públicamente.

Latacunga, 06 de marzo de 2022

.....

Ing. Minchala Ávila Luis Ismael PhD.

Director



ESPE

UNIVERSIDAD DE LAS FUERZAS ARMADAS

INNOVACIÓN PARA LA EXCELENCIA

C.C.: 030145348-6

VICERRECTORADO DE INVESTIGACIÓN, INNOVACIÓN Y TRANSFERENCIA TECNOLÓGICA

CENTRO POSGRADOS

RESPONSABILIDAD DE AUTORÍA

Yo, Chasi Taimbud Christian Stalin con cédula de ciudadanía N.º 171890290-9, declaro que el contenido, ideas y criterios del trabajo de titulación: "Sistema de visión por computador para detectar automáticamente líneas de cultivo y mejorar la precisión del guiado durante misiones autónomas de vuelo de un vehículo aéreo no tripulado", es de mi autoría y responsabilidad, cumpliendo con los requisitos legales, teóricos, científicos, técnicos y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, respetando los derechos intelectuales de terceros y referenciando las citas bibliográficas.

Latacunga, 06 de marzo de 2022



Chasi Taimbud, Christian Stalin

C.C.: 171890290-9



ESPE

UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

VICERRECTORADO DE INVESTIGACIÓN, INNOVACIÓN Y TRANSFERENCIA TECNOLÓGICA

CENTRO DE POSGRADOS

AUTORIZACIÓN DE PUBLICACIÓN

Yo, **Chasi Taimbud Christian Stalin** con cédula de ciudadanía N.º 171890290-9, autorizo a la UNIVERSIDAD DE LA FUERZAS ARMADAS-ESPE, publicar el trabajo de titulación: "**Sistema de visión por computador para detectar automáticamente líneas de cultivo y mejorar la precisión del guiado durante misiones autónomas de vuelo de un vehículo aéreo no tripulado**", en el Repositorio Institucional, cuyo contenido, ideas y criterios son de mi responsabilidad.

Latacunga, 05 de marzo de 2022

A handwritten signature in blue ink, appearing to read 'Chasi Taimbud', is written over a horizontal dashed line.

Chasi Taimbud, Christian Stalin

C.C.: 1718902909

Dedicatoria

Dedico este proyecto a Dios por acompañarme siempre y en todo lugar, por brindarme la oportunidad de fortalecer mis conocimientos para culminar con éxito esta etapa de mi vida profesional.

A mis queridos padres, Gladys y Ángel, quienes me han brindado su apoyo incondicional para alcanzar mis metas, mi eterno agradecimiento mamá y papá.

A mis familiares y amigos, especialmente a Jonathan, mejor amigo y consejero, pues han sido ellos quienes me han motivado a ser mejor cada día y han formado parte de este camino de aprendizaje.

Christian Chasi Taimbud

Agradecimiento

Agradezco primeramente a Dios por darme la sabiduría y el estímulo constante para alcanzar mis sueños, gracias por darme la fuerza necesaria para seguir adelante en todo momento.

A Grace, compañera incondicional que me apoyó durante toda mi etapa de estudio y la realización de este trabajo, gracias a ti culminé con éxito esta carrera profesional.

Quiero expresar un sincero agradecimiento a mi director, Ingeniero Ismael Minchala, por brindarme su apoyo y por compartirme su conocimiento durante el desarrollo de este proyecto.

Y a todas las personas que han estado presentes durante esta etapa de mi vida, estoy seguro de que cada uno de ustedes aportó de manera asertiva en mi vida.

Christian Chasi Taimbud

Tabla de contenidos

Carátula.....	1
Reporte de Verificación de Contenido	2
Certificación	3
Responsabilidad de autoría	4
Autorización de publicación	5
Dedicatoria	6
Agradecimiento	7
Tabla de contenidos	8
Índice de Tablas	13
Índice de Figuras	14
Índice de Ecuaciones	17
Lista de abreviaturas	18
Resumen	19
Abstract.....	20
Capítulo I	21
Definición del Proyecto	21
Antecedentes	21
Líneas de investigación.....	22
Sub líneas de investigación.....	22

Área de influencia	22
Planteamiento del problema	23
Trabajos relacionados	24
Justificación, importancia y alcance del proyecto	25
Objetivos	27
<i>Objetivo general del proyecto</i>	27
<i>Objetivos específicos del proyecto</i>	27
Hipótesis de investigación	28
Categorización de las variables de investigación	28
Marco teórico referencial	29
<i>Marco legal</i>	29
<i>Marco conceptual</i>	30
Diseño de la investigación	31
Capítulo II	34
Fundamentación teórica	34
Agricultura de precisión	34
Tecnologías aplicadas a la agricultura de precisión	34
<i>Sistemas de posicionamiento</i>	35
<i>Tecnologías de tasa variable</i>	35
<i>Sensores remotos</i>	35
<i>Aeronaves pilotadas remotamente</i>	35

Vehículos aéreos no tripulados.....	35
<i>Clasificación de drones</i>	36
ArduPilot	37
<i>Hardware</i>	38
<i>Sistema de propulsión.</i>	38
<i>Sistema de posicionamiento.</i>	39
<i>Sistema de telemetría.</i>	39
<i>Sistema de seguridad del vehículo.</i>	40
<i>Sistema de radio control.</i>	40
<i>Sistema estabilizador.</i>	41
<i>Firmware</i>	41
<i>Software</i>	42
Micro aerial vehicle link (MAVLink)	43
Dronekit	44
Visión por computador	44
<i>Iluminación</i>	45
<i>Iluminación direccional.</i>	45
<i>Iluminación difusa</i>	46
<i>Iluminación a contraluz.</i>	47
<i>Sistema de adquisición</i>	47
<i>Procesamiento y análisis</i>	48
Open Source Computer Vision Library (OpenCV)	49
Detección de líneas	49

	11
<i>Transformada Hough</i>	50
Capitulo III	53
Metodología	53
Ensamblaje y configuración del VANT	53
<i>Firmware</i>	55
<i>Calibración de componentes</i>	56
<i>Acelerómetro</i>	57
<i>Brújula</i>	57
<i>Transmisor RC</i>	59
<i>Controlador electrónico de velocidad</i>	59
<i>Estabilizador de cámara</i>	60
Configuración de Raspberry Pi	62
<i>Software</i>	62
<i>Entorno virtual</i>	66
<i>Dronekit-Python</i>	68
<i>OpenCV</i>	68
<i>MAVLink</i>	69
Calibración de la cámara	71
<i>Adquisición de imágenes</i>	74
<i>Parámetros intrínsecos y coeficiente de distorsión</i>	76
<i>Conversión pixel a mm</i>	78
Detección de líneas de cultivo	80
<i>Método de visión</i>	81

<i>Segmentación</i>	82
<i>Detección de líneas de cultivo curvas y rectas</i>	85
<i>Obtención de coordenadas</i>	87
Planeación de misión de vuelo automática	90
Capítulo IV	93
Resultados	93
Detección de líneas de cultivo	93
Precisión del sistema	96
Misión de vuelo autónoma	96
Capítulo V	98
Conclusiones y recomendaciones	98
Conclusiones	98
Recomendaciones	100
Trabajos Futuros	101
Bibliografía	102
Anexos	107

Índice de Tablas

Tabla 1 <i>Trabajos Relacionados</i>	24
Tabla 2 <i>Operacionalización de las Variables</i>	28
Tabla 3 <i>Matriz de Diseño de Investigación</i>	33
Tabla 4 <i>Clasificación de drones por tamaño y peso</i>	36
Tabla 5 <i>Componentes del VANT</i>	54
Tabla 6 <i>Información técnica GoPro Hero 3+</i>	79
Tabla 7 <i>Niveles HSV</i>	83
Tabla 8 <i>Precisión del sistema</i>	96

Índice de Figuras

Figura 1 <i>Cableado del sistema de propulsión</i>	38
Figura 2 <i>Cableado sistema de posicionamiento</i>	39
Figura 3 <i>Sistema de telemetría</i>	39
Figura 4 <i>Cableado sistema de seguridad</i>	40
Figura 5 <i>Sistema de radio control</i>	40
Figura 6 <i>Sistema estabilizador de cámara</i>	41
Figura 7 <i>Tipos de Firmware de ArduPilot</i>	42
Figura 8 <i>Mission Planner</i>	43
Figura 9 <i>Elementos de la visión por computador</i>	45
Figura 10 <i>Iluminación direccional</i>	46
Figura 11 <i>Iluminación difusa</i>	46
Figura 12 <i>Iluminación a contraluz</i>	47
Figura 13 <i>Campo de Visión</i>	48
Figura 14 <i>Detección de líneas</i>	50
Figura 15 <i>Transformada Hough</i>	50
Figura 16 <i>Ecuación de la recta</i>	51
Figura 17 <i>Acumulador de la transformada Hough</i>	52
Figura 18 <i>Diagrama de conexión del sistema</i>	55
Figura 19 <i>Versión de firmware</i>	56
Figura 20 <i>Tasa de baudios MP</i>	56
Figura 21 <i>Calibración acelerómetro MP</i>	57
Figura 22 <i>Calibración brújula Mission Planner</i>	58
Figura 23 <i>Posiciones de calibración brújula</i>	58

Figura 24 <i>Calibración Radio Control Mission Planner</i>	59
Figura 25 <i>Calibración ESC MP</i>	60
Figura 26 <i>Estabilizador de cámara</i>	61
Figura 27 <i>Software de calibración del gimbal</i>	61
Figura 28 <i>Raspberry Pi Imager</i>	63
Figura 29 <i>Conexión SSH</i>	64
Figura 30 <i>VNC Viewer</i>	64
Figura 31 <i>Software RaspAP</i>	65
Figura 32 <i>WEBGUI RaspAP</i>	66
Figura 33 <i>Entorno virtual</i>	67
Figura 34 <i>Biblioteca Dronekit</i>	68
Figura 35 <i>Biblioteca OpenCV</i>	69
Figura 36 <i>Conexión Pixhawk-Raspberry</i>	70
Figura 37 <i>Parámetros TELEM2</i>	70
Figura 38 <i>Habilitación puerto UART</i>	71
Figura 39 <i>Distorsión radial</i>	72
Figura 40 <i>Distorsión tangencial</i>	72
Figura 41 <i>Patrón de calibración</i>	74
Figura 42 <i>Capturas del patrón de calibración</i>	76
Figura 43 <i>Detección de esquinas en el patrón</i>	77
Figura 44 <i>Cultivo con diferente intensidad de luz solar</i>	81
Figura 45 <i>Arquitectura del sistema</i>	81
Figura 46 <i>Paleta de color verde HSV</i>	82
Figura 47 <i>Cultivo en el espacio de color HSV</i>	83

Figura 48 <i>Identificación de hileras</i>	84
Figura 49 <i>Imagen original filtrada empleando operaciones morfológicas</i>	84
Figura 50 <i>Detección de bordes en la imagen binaria</i>	85
Figura 51 <i>Recta proyectada en línea de cultivo curva</i>	86
Figura 52 <i>Detección de líneas curvas y rectas en el cultivo</i>	86
Figura 53 <i>Coordenadas rectangulares de las líneas de cultivo en px</i>	87
Figura 54 <i>Coordenadas rectangulares de las líneas de cultivo en mm</i>	88
Figura 55 <i>Coordenadas del sistema de detección de líneas curvas y rectas</i>	88
Figura 56 <i>Coordenadas de las líneas de cultivo en secuencia</i>	89
Figura 57 <i>Secuencia de vuelo</i>	89
Figura 58 <i>Ubicación del cultivo</i>	90
Figura 59 <i>Planificación de misión de vuelo automática</i>	91
Figura 60 <i>Waypoints de la misión de vuelo automática</i>	92
Figura 61 <i>Comparativa de capturas con y sin distorsión</i>	93
Figura 62 <i>Capturas en diferentes condiciones de iluminación</i>	94
Figura 63 <i>Imágenes filtradas con diferentes elementos estructurantes</i>	95
Figura 64 <i>VANT durante una misión de vuelo autónoma</i>	97

Índice de Ecuaciones

Ecuación (1) <i>Fórmula de la matriz de la cámara</i>	73
Ecuación (2) <i>Matriz de rotación de la cámara</i>	73
Ecuación (3) <i>Matriz de distorsión radial corregida en X</i>	73
Ecuación (4) <i>Matriz de distorsión radial corregida en Y</i>	73
Ecuación (5) <i>Matriz de distorsión tangencial corregida en X</i>	73
Ecuación (6) <i>Matriz de distorsión tangencial corregida en Y</i>	73
Ecuación (7) <i>Valores de la matriz de la cámara</i>	78
Ecuación (8) <i>Valores de los coeficientes de distorsión de la cámara</i>	78
Ecuación (9) <i>Fórmula del campo de visión</i>	78
Ecuación (10) <i>Fórmula del tamaño del pixel</i>	78
Ecuación (11) <i>Valor del campo de visión horizontal</i>	79
Ecuación (12) <i>Valor del tamaño del pixel horizontal</i>	79
Ecuación (13) <i>Valor del campo de visión vertical</i>	79
Ecuación (14) <i>Valor del tamaño del sensor vertical</i>	79

Lista de abreviaturas

VANT	Vehículo aéreo no tripulado
MP	Mission Planner
ISPA	Sociedad Internacional de Agricultura de Precisión
GPS	Sistema de posicionamiento global
UAS	Sistema de aeronaves no tripuladas
GCS	Estación terrestre de control
SS	Tamaño del sensor
API	Interfaz de programación de aplicaciones
GUI	Interfaz gráfica de usuario
ESC	Controlador electrónico de velocidad
VNC	Equipo de red virtual
GPIO	Entrada/Salida de propósito general
UART	Transmisor-receptor asíncrono universal

Resumen

En la actualidad, los Vehículos Aéreos no Tripulados (UAV, por sus siglas en inglés) o drones y su tecnología se han desarrollado rápidamente. Este desarrollo ha logrado que los mismos puedan ser utilizados para realizar tareas como: entregar productos, grabar videos, tomar fotografías sin incurrir en costos elevados, monitorear y adquirir datos e inclusive atacar a un objetivo específico. Utilizar el recurso humano para realizar actividades de monitoreo y adquisición de datos, representa un alto porcentaje de los costos de producción debido a que no solo implica la mano de obra sino también su tiempo de ejecución, además de que no se garantiza que la actividad cubra toda el área geográfica de los cultivos, ya sea por su extensión, difícil acceso o baja precisión de guiado. Este trabajo consiste en desarrollar e implementar un sistema a bordo de un cuadricóptero para la detección automática de líneas de cultivo, orientado a la agricultura de precisión, basado en el uso de técnicas de visión por computador aplicadas a imágenes capturadas mediante una cámara digital, con el objetivo de planificar una ruta de vuelo precisa de manera autónoma en misiones de monitoreo. Este trabajo es una base para al desarrollo e implementación de aplicaciones que empleen visión por computador a bordo de un vehículo aéreo no tripulado. En este primer desarrollo se logró emplear una cámara con una lente gran angular para detectar características de las líneas de un cultivo y usarlas para diseñar un algoritmo que genere coordenadas para crear una misión de vuelo autónoma.

Palabras Clave:

- **UAV**
- **AGRICULTURA DE PRECISIÓN**
- **VISION POR COMPUTADOR**
- **MISSION PLANNER**

Abstract

Currently, Unmanned Aerial Vehicle (UAV) or drones and their technology have developed rapidly. This development has allowed them to be used to perform tasks such as: delivering products, recording videos, taking photographs without incurring high costs, monitoring and acquiring data and even attacking a specific target. The use of human resources to carry out monitoring and data acquisition activities represents a high percentage of production costs because it involves not only labor but also execution time, in addition to the fact that the activity is not guaranteed to cover the entire geographical area of the crops, either because of its extension, difficult access or low guidance precision. The work consists of developing and implementing a system on board a quadcopter for the automatic detection of crop lines, oriented to Precision Agriculture, based on the use of computer vision techniques applied to images captured by a digital camera, with the objective of planning a precise flight path autonomously in monitoring missions. This work is a basis for the development and implementation of applications that employ computer vision on the edge of an unmanned aerial vehicle. In this first development it was possible to use a camera with a wide-angle lens to detect characteristics of the lines of a crop and use them to design an algorithm that generates coordinates to create an autonomous return mission.

Key words:

- **UAV**
- **PRECISION AGRICULTURE**
- **COMPUTER VISION**
- **MISSION PLANNER**

Capítulo I

Definición del Proyecto

Antecedentes

Los Vehículos Aéreos no Tripulados (UAV, por sus siglas en inglés) o drones se han convertido en un recurso importante para realizar misiones que dependen principalmente de una planificación previa y un Sistema de Posicionamiento Global (GPS, por sus siglas en inglés) para definir uno o varios puntos de aterrizaje aproximados. Sin embargo, las variaciones del entorno pueden afectar el lugar de aterrizaje e inclusive el paisaje puede interferir en la señal del satélite (Safadinho, Ramos, Ribeiro, Filipe, Barroso y Pereira, 2020).

El uso de métodos agrícolas tradicionales y el creciente costo de productos químicos generan el incremento de costos de producción que aunados a la contaminación ambiental provocada por los residuos de herbicidas incitan a buscar métodos alternativos para proteger los cultivos y optimizar los recursos. Una forma de disminuir estos problemas es utilizar técnicas de precisión para operaciones agrícolas específicas, de modo que se obtenga resultados óptimos (Astrand y Baerveldt, 2005).

Los sistemas de visión por computador presentan inconveniente al momento de obtener las imágenes desde un vehículo en movimiento debido a las condiciones cambiantes de iluminación, sombras y posibles vibraciones. En este contexto, se ha incrementado la necesidad de implementar mecanismos tecnológicos automáticos en la agricultura, que sean capaces de superar estos inconvenientes y aporten información para el guiado autónomo de vehículos en tiempo real (Barreda, 2011).

El ritmo de vida actual, el avance tecnológico y las capacidades de los UAVs han influido en la creación de nuevas aplicaciones comerciales para la industria e inclusive la defensa civil.

Estos vehículos son utilizados en diferentes campos como: seguridad, gestión de desastres, entrega de paquetes, video vigilancia, silvicultura, agricultura, entre otros.

La agricultura es un área viable para la implementación de nuevas tecnologías que promueven herramientas para mejorar su eficiencia operativa y gestión en la toma de decisiones. Según la International Society of Precision Agriculture (ISPA), la Agricultura de Precisión es una estrategia de gestión que recoge, procesa y analiza datos temporales, espaciales e individuales y los combina con otras informaciones para respaldar decisiones.

En este trabajo se describe el desarrollo e implementación de un sistema a bordo de un cuadricóptero para la detección automática de líneas de cultivo, basado en el uso de técnicas de visión por computador, con el objetivo de planificar rutas de vuelo precisas para realizar misiones de monitoreo.

Líneas de investigación

- Automática y control
- Tecnologías de la información y comunicación
- Ingeniería de software

Sub líneas de investigación

- Comunicaciones inalámbricas
- Procesamiento digital de señales e imágenes
- Inteligencia artificial y realidad virtual

Área de influencia

- Agricultura de precisión en la zona central del Ecuador.

Planteamiento del problema

En la actualidad, se ha determinado que la obtención de datos y su posterior análisis juegan un papel importante en la toma de decisiones, mejora de procesos y optimización de recursos dentro de las industrias y procesos comerciales.

La agricultura se ve impulsada por las nuevas tecnologías que favorecen su crecimiento, y es por eso que nace la agricultura de precisión que es una estrategia innovadora de gestión eficiente de recursos, que hace uso de tecnologías emergentes para recoger, procesar y analizar datos obtenidos mediante diferentes tipos de sensores.

La necesidad de automatizar las actividades agrícolas, con el objetivo de mejorar la productividad, disminuir la contaminación ambiental y reducir los costos de producción por mano de obra y materia prima, específicamente de productos químicos, presenta un incremento no sólo en países desarrollados sino también países en vías de desarrollo, tal es el caso de Ecuador que ha empezado a aplicar nuevas tecnologías en sus cultivos.

Implementar infraestructura de transmisión cableada para recolectar datos y cubrir toda el área geográfica de los cultivos resulta costoso y presenta limitaciones como: poca capacidad de mantenimiento y baja precisión. Por lo tanto, se recomienda implementar dispositivos móviles con sensores on-board para realizar actividades de monitoreo y recolección de datos, tales como: humedad, temperatura o captura de imágenes específicas para analizar el ritmo de crecimiento de los cultivos.

Actualmente la navegación autónoma de UAVs utiliza dispositivos GPS. Estos dispositivos proporcionan un nivel alto de precisión para el guiado de este tipo de vehículos cuando las líneas de cultivo presentan desviaciones o su distribución no es lineal.

En Ecuador la agricultura de precisión es empleada en florícolas, bananeras y empresas azucareras. Se emplean UAVs o drones como protagonistas de este tipo de agricultura gracias a

su flexibilidad y bajo costo. Sin embargo, los sistemas de visión por computador on-board presentan dificultades que afectan el proceso de identificación de las líneas de cultivo, tales como: iluminación no controlada, desnivel del terreno, discontinuidades o espacios libres de plantas o cantidad elevada de malas hierbas.

Los problemas presentados sugieren desarrollar un sistema de detección automático de líneas de cultivo, a bordo de un cuadricóptero, utilizando técnicas de visión por computador aplicadas a imágenes capturadas mediante una cámara digital, con el objetivo de mejorar la precisión del guiado del vehículo durante misiones autónomas de vuelo aplicadas al monitoreo de cultivos.

Trabajos relacionados

En función de la investigación previa fue posible identificar los trabajos relacionados que se describen en la Tabla 1.

Tabla 1

Trabajos Relacionados

No.	Artículo / Proyecto	Autor	Año	Descripción
1	Sistema de visión artificial para el análisis de imágenes de cultivo basado en texturas orientadas	Sotomayor, F., Gómez, A., y Cela	2014	Presenta el desarrollo de un sistema de visión que detecta la dirección de las líneas de cultivos de trigo en imágenes áreas, basándose en el análisis de texturas orientadas, se implementan 3 estrategias para afrontar situaciones como: poca capa vegetal, rodales de mala hierba, iluminación variable, escalado de la imagen, entre otros.
2	Técnicas de visión artificial para la segmentación y detección de líneas de cultivo en imágenes agrícolas	Montalvo, M.	2011	Presenta un trabajo enfocado a la agricultura de precisión para optimizar recursos agrícolas usando robots, emplea técnicas de visión por computador con la finalidad de detectar línea de cultivo y rodales de malas hierbas en campos de maíz bajo condiciones ambientales de iluminación variable.

No.	Artículo / Proyecto	Autor	Año	Descripción
3	Detección automática de líneas de cultivo de papa utilizando imágenes digitales	Santillán, I., Herrera, C., y Ortega, J.	2018	Este trabajo presenta dos métodos para detectar líneas campos de maíz durante los estados iniciales de crecimiento de cultivos y malas hierbas. El objetivo del trabajo es identificar las líneas de cultivo para guiar con precisión vehículos autónomos y realizar tratamientos específicos que incluyan la eliminación de malas hierbas situadas entre las líneas de cultivo.
4	Métodos de visión por computador para detección automática de líneas de cultivo curvas/rectas y malas hierbas en campos de maíz.	García, I.	2018	Este trabajo utiliza un UAV para mejorar la conectividad de la red terrestre actuando como una mula que recopila datos y los descarga en nodos de sensores terrestres. Evalúa el rendimiento de la red, midiendo la tasa promedio de pérdida de paquetes.
5	Sistema de visión para agricultura de precisión: identificación en tiempo real de líneas de cultivo y malas hierbas en campos de maíz	Guerrero, J.	2015	Este trabajo presenta una investigación respecto a la identificación de líneas de cultivo de maíz y la discriminación entre mala hierba y cultivo. Además, presenta el diseño de un sistema experto de visión por computador capaz de identificar en tiempo real líneas de cultivo y malas hierbas en campos de maíz, y controla el guiado de un tractor y el solapamiento de zonas de tratamiento con el objetivo de implementar un tratamiento selectivo.

Nota. Esta tabla muestra el estado del arte actual referente a trabajos de detección de líneas de cultivo utilizando métodos de visión por computador.

Justificación, importancia y alcance del proyecto

La agricultura tradicional puede generar desperdicio de recursos tales como: mano de obra y tiempo de ejecución en actividades necesarias dentro de cultivos con áreas geográficas extensas, además de que pueden existir lugares de difícil acceso que no son monitoreados

constantemente. El uso de recursos humanos dentro de la agricultura representa un alto porcentaje de los costos variables de producción, situación que busca mejorarse con la implementación de la automatización y la robótica en este sector (Roldán, Joossen, Sanz, Del Cerro y Barrientos, 2015).

Durante los últimos años, la tecnología en los procesos comerciales e industriales ha evolucionado rápidamente, exigiendo que las industrias adapten métodos automatizados para cumplir los requerimientos de sus procesos y optimizar sus recursos. En la actualidad, el recurso humano está pasando de realizar tareas de ejecución a tareas de supervisión y análisis, debido a la implementación de dispositivos y sistemas automáticos capaces de realizar tareas específicas.

En este sentido los UAVs ofrecen una gran variedad de posibilidades en la agricultura de precisión, como por ejemplo: sobrevolar áreas geográficas extensas de cultivos de forma veloz recolectando datos de diversas magnitudes (humedad, temperatura, entre otras) gracias a sus sensores. Una de las aplicaciones más importantes que ofrecen estos vehículos en la agricultura es la localización prematura de plagas y enfermedades, información que permite a los productores agrícolas tomar decisiones tempranas.

Es importante mantener información continua de los cultivos, para ello es necesario monitorear constantemente toda su área geográfica con el objetivo de recolectar información. Sin embargo, es una tarea cuyo tiempo de ejecución es elevado y en caso de ser realizada con la ayuda de un UAV o dron requiere de personal calificado para pilotar el vehículo.

Implementar misiones automáticas puede ser una solución, hoy en día los drones utilizan GPS para obtener las coordenadas de vuelo, utilizar este sistema de posicionamiento implica un bajo nivel de precisión cuando se tiene cultivos en terrenos irregulares o en hileras cuya distribución no es lineal, lo que da como resultado un monitoreo incompleto del cultivo.

El alcance del proyecto está limitado a desarrollar e implementar un sistema de detección automático de líneas de cultivo curvas y rectas, a bordo de un cuadricóptero, utilizando técnicas de visión por computador en imágenes capturadas mediante una cámara digital para mejorar la precisión del guiado del vehículo durante misiones autónomas de vuelo aplicadas al monitoreo de cultivos. Los beneficios del proyecto serán, entre otros, los siguientes:

- económicos, porque se incrementará la capacidad de producción y se disminuirán gastos por mano de obra y tiempos de ejecución manual de tareas;
- tecnológicos, porque se implementará técnicas de visión por computador para detectar curvas y rectas patrón para guiar el vehículo; y,
- sociales, porque ayudará en las tareas agrícolas de los productores ecuatorianos.

El presente trabajo dejará la posibilidad de desarrollar proyectos futuros que permitirán realizar monitoreo directo para detectar anomalías en los cultivos, tales como plagas o enfermedades.

Objetivos

Objetivo general del proyecto

Diseñar e implementar un sistema de detección automática de líneas de cultivo para mejorar la precisión de guiado autónomo de un cuadricóptero en misiones de monitoreo.

Objetivos específicos del proyecto

- Determinar los parámetros intrínsecos y extrínsecos de la cámara para mitigar los efectos de distorsión en las imágenes
- Implementar un sistema de visión a bordo de un cuadricóptero para capturar imágenes digitales de cultivos en diferentes etapas de crecimiento.
- Implementar algoritmos de visión por computador para detectar automáticamente líneas de cultivo curvas y rectas utilizando software de código abierto.

- Utilizar las coordenadas obtenidas por el sistema de detección para planificar una misión de vuelo y monitorear un cultivo.
- Evaluar el desempeño del sistema en un ambiente controlado.

Hipótesis de investigación

Un sistema de detección automático de líneas de cultivo curvas y rectas permite mejorar la precisión de guiado de un vehículo aéreo no tripulado durante misiones automáticas de monitoreo.

Categorización de las variables de investigación

En la hipótesis planteada se identifican dos variables:

- **Variable independiente:** implementación de un sistema de detección automático de líneas de cultivo curvas y rectas para un cuadricóptero.
- **Variable dependiente:** Mejorar la precisión de guiado de un vehículo aéreo no tripulado durante misiones automáticas de monitoreo.

Tabla 2

Operacionalización de las Variables

Variables	Tipo	Definición Conceptual	Definición Operacional	Dimensiones	Indicadores
Implementación de un sistema de detección automático de líneas de cultivo curvas y rectas para un cuadricóptero	Independiente	Un sistema de detección de líneas de cultivo ayuda a optimizar los procesos de siembra, control de malas hierbas y cosecha.	Un sistema de detección de líneas basado en visión por computador es un sistema de control que obtiene coordenadas para guiar un vehículo.	- Costo de implementación - Facilidad de implementación - Tiempo de implementación	- Mínimo costo de implementación. -Facilidad mínima de implementación. - Tiempo mínimo de implementación

Variab les	Tipo	Definición Conceptual	Definición Operacional	Dimensiones	Indicadores
Mejorar la precisión de guiado de un vehículo aéreo no tripulado durante misiones automáticas de monitoreo	Dependiente	Las misiones de monitoreo son funciones o trabajos que tienen como objetivo inspeccionar un área en busca de información	Las misiones de monitoreo son tareas cuyo objetivo es desempeñar una función específica en algún lugar y registrar información del entorno.	- Precisión del guiado. - Tiempo de respuesta. - Duración de la misión.	- Porcentaje de precisión del guiado. - Tiempo máximo de respuesta. - Tiempo máximo de duración de la misión.

Marco teórico referencial

Marco legal

- **Dirección General de Aviación Civil. Resolución 251 / 2015.**

Esta resolución dictada por el Director General de Aviación Civil de la República del Ecuador aprueba la Operación de los Sistemas de Aeronaves Pilotadas a Distancia (RPAS) o conocidas como DRONES o Sistemas de Aeronaves No Tripuladas (UAS). Establece además disposiciones complementarias para poder pilotar este tipo de vehículos.

- **Tercer Suplemento – Registro Oficial N° 439. Ley Orgánica de Telecomunicaciones.**

Esta Ley establece el régimen general de telecomunicaciones y del espectro radioeléctrico como sectores estratégicos del Estado. Además, establece que es facultad del Estado Central, a través del Ministerio rector de las Telecomunicaciones y de la Sociedad de la Información y de la Agencia de Regulación y Control de las Telecomunicaciones, establezca políticas, requisitos, norma y condiciones para el despliegue de infraestructura alámbrica e inalámbrica de telecomunicaciones a nivel nacional.

- **Acuerdo Ministerial N° 299 – Registro Oficial N° 34.**

Este Acuerdo establece las responsabilidades que tiene el Estado para alcanzar la soberanía alimentaria, como por ejemplo: fortalecer la diversificación y la introducción de tecnología en la producción agropecuaria, regular bajo normas de bioseguridad el uso y desarrollo de biotecnología, así como también proteger a la población del consumo de alimentos contaminados.

Marco conceptual

La optimización de recursos depende de la administración y planeamiento de actividades de previsión y control, es por eso que la aplicación de nuevas tecnologías está produciendo un cambio importante en el modo de operación de los agricultores.

Los UAVs son una gran opción para este cambio porque permiten reducir costos, disminuir el impacto ambiental, aumentar la productividad gracias a la información que éstos brindan a los agricultores sobre el estado hídrico, nivel de sanidad y desarrollo de sus cultivos.

Los siguientes conceptos sintetizan los aspectos fundamentales de las principales áreas de estudio del proyecto:

- **Agricultura de precisión:** está formada por un conjunto de tecnologías que combinan sensores, sistemas de información, máquinas inteligentes y gestión de información para optimizar la producción tomando en cuenta variabilidades e incertidumbres propias de los sistemas agrícolas. La agricultura de precisión proporciona un medio para supervisar y controlar la cadena de producción y gestionar tanto la cantidad como la calidad de los productos agrícolas (Gebbers y Adamchuk, 2010).
- **Vehículo aéreo no tripulado:** Proviene de la terminología inglesa UAV (Unmanned Aerial Vehicle), es un vehículo controlado mediante navegación autónoma o por medio de un

radio control, basado en la sustentación dinámica y el empuje aerodinámico (Nonami, Kendoul, Suzuki, Wang y Nakazawa, 2010).

- **Visión por computador:** Es la transformación de datos de una cámara fotográfica o de video en una decisión, como por ejemplo detectar la presencia de una persona en un entorno o una nueva representación, como convertir una imagen en color en una imagen en escala de grises o quizá eliminar el movimiento de una cámara en una secuencia de imágenes. Las transformaciones son realizadas para alcanzar objetivos específicos (Bradski y Kaehler, 2008).
- **OpenCV:** Es una biblioteca de código abierto que incorpora una gran cantidad de algoritmos de visión por computador, esta librería está formada por varios módulos entre los cuales se encuentran: procesamiento de imágenes, transformación geométrica, filtros, cambio de espacio de color, entre otros. Por otro lado, permite realizar análisis de video, calibración de cámaras para eliminar la distorsión, detección y seguimiento de objetos (Culijak, Abram, Pribanic, Dzapo y Cifrek, 2012)

Diseño de la investigación

El presente trabajo está basado en investigación aplicada, cuyo objetivo es resolver un problema socio-económico, para lo cual se utilizará técnicas de visión por computador con el objetivo de detectar líneas de cultivo para mejorar el guiado de un UAV. Se realizará el armado y montaje de un cuadricóptero al que se le implementará el sistema propuesto para resolver un modelo hipotético-deductivo, el cual propone una hipótesis en base a la observación del fenómeno en el estudio. La investigación finaliza con la verificación de la hipótesis en base a la evaluación del desempeño del sistema implementado en un ambiente controlado. La secuencia de investigación será la siguiente:

- **Identificación del problema y estudios relacionados.**

Mediante investigación bibliográfica de diferentes fuentes como: artículos científicos, libros, informes, normas, leyes y reglamentos, se obtiene el estado del arte y las soluciones existentes a problemas similares.

- **Implementar un sistema de visión a bordo de un cuadricóptero para capturar imágenes digitales de cultivos en diferentes etapas de crecimiento**

Mediante algoritmos de visión por computador se determinará los parámetros intrínsecos y extrínsecos de la cámara con la finalidad de mitigar su distorsión radial y tangencial, para luego capturar imágenes digitales de cultivos en diferentes etapas de crecimiento con el sistema de visión a bordo de un cuadricóptero.

- **Implementar algoritmos de visión por computador para detectar automáticamente líneas de cultivo curvas y rectas utilizando software de código abierto.**

Mediante técnicas de visión por computador aplicadas a las imágenes digitales de los cultivos, capturadas con el sistema de visión, implementar un sistema resistente a perturbaciones como cambios de iluminación y hierbas malas con la finalidad de detectar automáticamente líneas de cultivo, curvas y rectas, que servirán como coordenadas guía.

- **Planificar una misión de vuelo y monitorear un cultivo utilizando las coordenadas obtenidas por el sistema de detección.**

Emplear las coordenadas obtenidas por el sistema de detección de líneas para mejorar el guiado de vuelo del cuadricóptero con el objetivo de monitorear un cultivo.

- **Implementar el sistema propuesto y evaluar su desempeño.**

Mediante la implementación e integración de la propuesta, se realizará la validación del sistema y a través de la evaluación de su desempeño se podrá verificar la hipótesis expuesta.

Tabla 3

Matriz de Diseño de Investigación

Objetivo Específico	Técnica	Instrumentos	Variables
Recolectar información sobre las técnicas de aterrizaje de los UAVs	Investigación bibliográfica.	Recopilación documental. Computador Internet	Técnicas de recolección de datos.
Implementar un sistema de visión a bordo de un cuadricóptero para capturar imágenes digitales de cultivos en diferentes etapas de crecimiento	Investigación bibliográfica Desarrollo de programas y comprobación de funcionamiento	Ambiente de desarrollo de programas Computador	Técnicas de reconocimiento de objetos. Eficiencia del algoritmo.
Implementar algoritmos de visión por computador para detectar automáticamente líneas de cultivo curvas y rectas utilizando software de código abierto.	Investigación bibliográfica Investigación aplicada Desarrollo de programas y comprobación de funcionamiento	Ambiente de desarrollo de programas Computador Controlador Hoja de datos de referencia.	Técnicas de reconocimiento de esquinas, líneas y curva. Eficiencia del algoritmo. Facilidad de implementación. Disponibilidad de medios.
Planificar una misión de vuelo y monitorear un cultivo utilizando las coordenadas obtenidas por el sistema de detección.	Investigación bibliográfica Desarrollo de programas	Ambiente de desarrollo de programas. Computador.	Tiempo de ejecución. Disponibilidad de medios.
Implementar el sistema propuesto y evaluar su desempeño.	Investigación aplicada	Hoja de datos de referencia Ambiente de desarrollo de programas	Efectividad del sistema. Facilidad de implementación. Tiempo de implementación.

Capítulo II

Fundamentación teórica

En la actualidad la agricultura se ha convertido en uno de los principales ejes de consumo en el Ecuador y el mundo, más aún cuando nos encontramos en una época dónde las personas buscan alimentos cultivados naturalmente. Emplear métodos de cultivo de forma tradicional, sin implementar herramientas tecnológicas, puede llevar a tener un uso exagerado de químicos en los cultivos, mismos que a su vez incrementan el costo de producción, provocan el deterioro del suelo y contaminan el ambiente (Kataoka, 2003).

En este contexto, nace la necesidad de crear máquinas y equipos capaces de evaluar de forma visual las características de los cultivos con el objetivo de tomar decisiones a partir de una gran cantidad de datos recolectados a través de sensores. La revolución tecnológica y el internet de las cosas ha llegado a ser utilizado en la agricultura de precisión e incluso se ha tornado un apoyo en el manejo de vehículos agrícolas.

Agricultura de precisión

La agricultura de precisión surgió en Estados Unidos y hace referencia al uso adecuado y correcto de materias primas, tales como: químicos, abonos, fertilizantes, etc. Este tipo de agricultura emplea tecnologías de la información para tener un óptimo manejo del suelo y sus cultivos, además de que emplea sistemas de posicionamiento global y de otras herramientas tecnológicas para obtener la mayor cantidad de datos de los cultivos (García, 2008).

Tecnologías aplicadas a la agricultura de precisión

Las tecnologías, métodos y elementos empleados en la agricultura de precisión son extensas, todos con un mismo objetivo, mejorar la productividad y disminuir costos de producción. A continuación, se describe algunos de las tecnologías más utilizadas dentro de la agricultura de precisión.

Sistemas de posicionamiento

Sistemas importantes dentro del control agrícola ya que proveen información en tiempo real sobre su ubicación y facilitan la implementación de rutas óptimas de monitoreo. Dentro de los más destacados tenemos: GPS, GLONASS, Beidou y Galileo (Arley y Llano, 2016).

Tecnologías de tasa variable

Corresponde a elementos para la aplicación de fertilizantes, herbicidas, pesticidas, agua y suplementos, necesarios para los cultivos, de forma autónoma dependiendo de la calidad del suelo, ya que dependiendo de la zona el requerimiento puede ser diferente (Arley y Llano, 2016).

Sensores remotos

Elementos cuyo objetivo es recolectar datos del cultivo, tales como: calidad del suelo, humedad, precipitaciones, temperatura, entre otros, con el uso de redes inalámbricas como Wifi, Bluetooth o redes celulares para conocer la condición del cultivo (Arley y Llano, 2016).

Aeronaves pilotadas remotamente

Vehículos que ofrecen una solución novedosa en la obtención de imágenes en áreas de difícil acceso, adquisición de datos y monitorización de cultivos. El uso de estos dispositivos en la agricultura se ha incrementado en los últimos años ya sea por la reducción de costos producción y mano de obra o su fácil adquisición (Arley y Llano, 2016).

Vehículos aéreos no tripulados

Los vehículos aéreos no tripulados o UAV, más conocidos como drones, se destacan por no llevar un piloto en su interior y ser dirigidos por una persona o un sistema electrónico externo, mismo que toma decisiones y envía señales de activación del paso a paso a seguir durante el vuelo. Desde sus inicios, los drones fueron creados con el objetivo de salvaguardar la seguridad del ser humano y facilitar tareas que puedan resultar complejas en zonas de difícil

acceso, con el paso de los años y el desarrollo de tecnologías como el GPS, se han creado modos de vuelo autónomo para realizar cualquier intervención, despegar y aterrizar sin ningún tipo de manipulación ni presencia humana (García, 2017).

Existe una variedad amplia de modelos de drones tomando en cuenta su forma, configuración, volumen y funcionalidad, lo que ha permitido que los drones se adapten con facilidad a varias actividades o aplicaciones como misiones de monitoreo, recopilación de datos a través de sensores, apoyo en incendios, entregas a domicilio, levantamientos topográficos, rescates, entre otros (Lee y Choi, 20016).

Clasificación de drones

De acuerdo con el departamento de defensa de los Estados Unidos, se puede clasificar un drone por su tamaño en cuatro categorías. En la siguiente tabla se muestra la categorización de estos por el peso de despegue, altitud de funcionamiento respecto al Nivel del Suelo (SNS), o al Nivel del Mar (SNM) y velocidad aérea.

La tabla 4 indica clasificaciones de drones tomando en cuenta su tamaño y peso.

Tabla 4

Clasificación de drones por tamaño y peso

<i>Categoría</i>	<i>Tamaño</i>	<i>Peso máximo de despegue (libras)</i>	<i>Altitud de funcionamiento (pies)</i>	<i>Velocidad aérea (nudos)</i>
<i>Grupo 1</i>	Pequeño	0-20	<1200 SNS	<100
<i>Grupo 2</i>	Mediano	21-55	<3500 SNS	<250
<i>Grupo 3</i>	Grande	<1320	<18000 SNM	<250
<i>Grupo 4</i>	Extragrande	>1320	>18000 SMS	Cualquiera

Nota. Recuperado de métodos de visión por computador para detección automática de líneas de cultivo curvas/rectas y malas hierbas en campos de maíz, por García, 2018.

Debido a una gran comunidad dedicada al mundo de los sistemas de vuelo y radio control se ha desarrollado software de código abierto para armar, configurar y controlar drones personalizados. Dicho esto, cualquier persona tiene la facilidad de armar un drone con elementos de diferentes marcas y modelos comerciales. Sin embargo, es importante que aquella persona que desea armar uno de estos vehículos, tenga conocimiento técnico sobre la estructura y los elementos que lo conforman, a continuación, se detalla una lista de los elementos fundamentales de un drone:

- Controlador de vuelo
- Controlador de velocidad electrónico (ESC Electronic Speed Controller)
- Motores
- Módulo de distribución de potencia
- Batería
- Hélices
- Radio Control
- Módulo GPS
- Brújula
- Marco o Frame

ArduPilot

ArduPilot es un sistema de código abierto que permite la creación y uso de sistemas de vehículos aéreos, terrestres y acuáticos no tripulados y autónomos. Proporciona un conjunto completo de herramientas adecuadas para casi cualquier tipo de vehículo y evoluciona constantemente gracias a los comentarios de una gran comunidad de usuarios que lo utilizan. ArduPilot no es fabricante de hardware, pero funciona con una amplia variedad de dispositivos y permite la fácil integración de periféricos que brindan funcionalidades avanzadas como:

telemetría, comunicación en tiempo real, conexión con estaciones terrestres y computadoras (ArduPilot, 2020).

Hardware

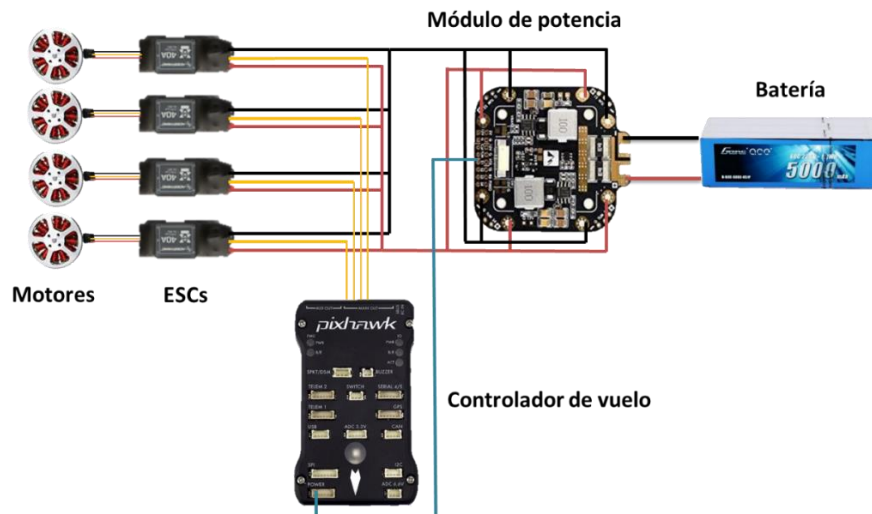
El hardware de ArduPilot consiste en un controlador que recibe información de sensores periféricos y dispositivos de salida que se encargan de emitir señales de activación a los actuadores para ejecutar maniobras controladas en el vehículo (ArduPilot, 2020).

La configuración empleada para el desarrollo de este proyecto está formada por varios subsistemas detallados a continuación.

Sistema de propulsión. El sistema de propulsión de un drone es de vital importancia, ya que es el encargado de elevar la carga montada sobre el marco de este, los elementos empleados en este sistema son: controlador de vuelo, motores, ESCs, módulo de distribución de energía y batería. La figura 1 indica el cableado de energía y control para el correcto funcionamiento del sistema de control.

Figura 1

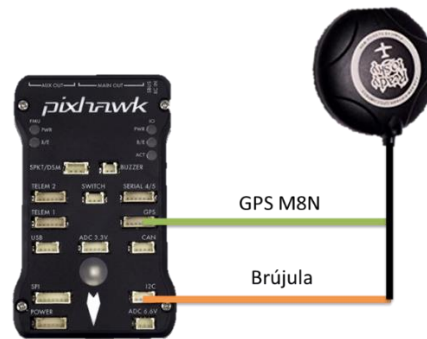
Cableado del sistema de propulsión



Sistema de posicionamiento. El controlador Pixhawk tiene un puerto dedicado para GPS y un puerto I2C que puede ser utilizado para conectar externamente una brújula al sistema. La figura 2 indica la conexión del GPS y la brújula a la tarjeta de control de vuelo.

Figura 2

Cableado sistema de posicionamiento.



Sistema de telemetría. El sistema de telemetría tiene como finalidad transmitir en tiempo real la información del controlador de vuelo y sus periféricos a una estación base para conocer el estado del piloto automático durante misiones de vuelo. Estos sistemas pueden ser implementados en computadores, estaciones de control terrestres y teléfonos inteligentes con sistema Android. La figura 3 indica la conexión del sistema de telemetría además de una representación sobre la comunicación inalámbrica entre la pixhawk y la estación terrestre de control.

Figura 3

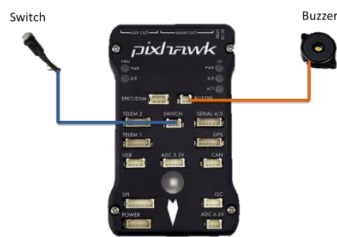
Sistema de telemetría



Sistema de seguridad del vehículo. Todo vehículo aéreo no tripulado requiere de 2 sistemas de seguridad que emitan una alerta y protejan al controlado de vuelo de posibles fallas. En este caso, el controlado debe tener conectado un pulsador de bloqueo de motores, que evite la puesta en marcha de estos mientras no se lo presione y por otro lado, el segundo sistema es un zumbador, que emite señales sonoras para alertar problemas o errores en el vehículo. La figura 4 indica el cableado de los elementos asociados a la seguridad del sistema.

Figura 4

Cableado sistema de seguridad



Sistema de radio control. El sistema de radio control es uno de los más importantes ya que es el elemento que nos permite maniobrar el vehículo. El controlador de vuelo es compatible con un sin número de dispositivos de radio control, en este proyecto se emplea el módulo receptor FlySky FS-IA6 que se comunica mediante serie al controlador de vuelo. La figura 5 indica la conexión del receptor a la tarjeta de control de vuelo.

Figura 5

Sistema de radio control



Sistema estabilizador. Una de las posibilidades que brindan los drones es poder capturar imágenes durante las misiones de vuelo. Sin embargo, las vibraciones generadas por las hélices y los propios movimientos del drone generan inestabilidad y balanceo, lo que dificulta la captura de imágenes. Por lo tanto, para mitigar estos problemas se recomienda utilizar un estabilizador o Gimbal de varios ejes, este dispositivo posee motores que corrigen constantemente la posición de la cámara. La figura 6 indica la conexión de la tarjeta de control de vuelo hacia el estabilizador de la cámara y su software de configuración.

Figura 6

Sistema estabilizador de cámara.

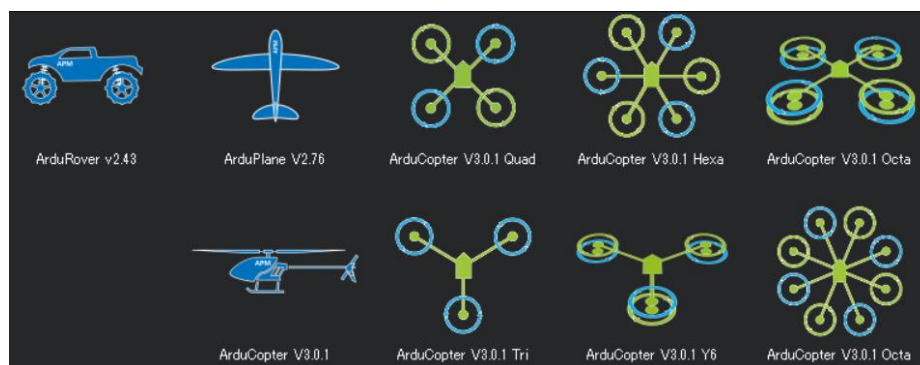


Firmware

El firmware es el código fuente que se ejecuta en el controlador y es importante seleccionar el correcto de acuerdo con el tipo de vehículo que se va a manipular. ArduPilot proporciona 5 tipos de códigos: Copter, Plane, Rover, Sub o Antenna Tracker (ArduPilot, 2020). La figura 7 indica los tipos de firmware compatibles con ardupilot y disponibles en Mission Planner.

Figura 7

Tipos de Firmware de ArduPilot



Nota. Adaptado del software Mission Planner V1.3.75

En este proyecto se emplea el código Copter que es un firmware UAV multicoptero capaz de cumplir con todos los requisitos para una misión de vuelo autónoma programada desde una estación terrestre. Este paquete tiene la ventaja de mostrar herramientas tan sencillas que incluso novatos pueden llegar a configurar aplicaciones complejas en poco tiempo.

Software

Uno de los componentes principales a la hora de pilotar un vehículo aéreo no tripulado es la estación de control o estación terrestre (GCS Ground Control Station). Mission Planner (MP) es un software, destinado para usuarios de Windows, compatible con ArduPilot que permite configurar, modificar, crear, monitorear y calibrar el vehículo de un computador (ArduPilot, 2020).

MP es utilizado para configurar los periféricos del vehículo, establecer parámetros de comunicación serial e inalámbrica con otros dispositivos de visualización y cargar el firmware y la versión más adecuada para el controlador de nuestro vehículo, de modo tal que los usuarios tienen el control total sobre la personalización y adaptación de sus vehículos a cualquier entorno con el menor riesgo de accidentes.

Una de las ventajas más interesantes que presenta MP es la opción de planificar y ejecutar misiones de vuelo autónomas basadas en el GPS montado sobre el vehículo, su interfaz visual acompañada con el mapa del área en tiempo real facilita la creación de la secuencia que debe seguir el multicoptero durante el vuelo.

El registro y análisis de datos es viable debido a la compatibilidad de MP con sistemas de telemetría, esto ayuda a tener información del vehículo durante la misión para que el usuario pueda tomar decisiones asertivas, corregir errores o recibir avisos como: calidad de señal GPS, nivel de batería, posición (latitud, altitud y longitud), velocidad, entre otros, en una interfaz gráfica amigable. La figura 8 indica la interfaz gráfica de usuario del software Mission Planner.

Figura 8

Mission Planner



Nota. Adaptado del software Mission Planner V1.3.75

Micro aerial vehicle link (MAVLink)

MAVLink es un protocolo de comunicación punto a punto utilizado en vehículos no tripulados y estaciones de control terrestre, la compatibilidad con ArduPilot hacen que este protocolo sea el ideal para este proyecto, ya que al permitir una comunicación inalámbrica bidireccional, la manipulación y el monitoreo del vehículo no será complejo.

Su principal ventaja es que admite diferentes medios de transporte de comunicación gracias a su ligera estructura, puede transmitir a través de WiFi, Ethernet o puertos serie mediante telemetría de bajo ancho de banda que generalmente funcionan a frecuencias como 433 MHz, 868 MHz o 915MHz. La tasa de datos puede alcanzar los 250 kbps con un alcance de 500 m. Sin embargo, esto depende en gran medida del entorno y nivel de ruido (Koubaa y Allouch, 2019).

Dronekit

Dronekit es un proyecto de código abierto de Python, conocido como Dronekit-Python, compatible con vehículos que llevan una computadora o controlador de vuelo mediante un enlace de baja latencia. Dronekit es capaz de mejorar significativamente el desempeño de un piloto automático, agregando configuraciones o aplicaciones complejas que requieran de un uso intensivo computacional como por ejemplo: visión por computador, adquisición de datos a través de sensores, planificación de rutas, modelado 3D en tiempo real, entre otros. Este código utiliza el protocolo MAVLink para gestionar misiones por medio del control directo sobre el movimiento y operaciones del vehículo (Robotics, 2019).

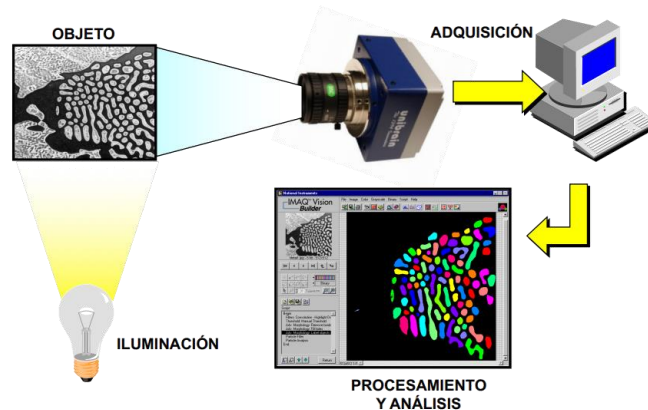
Visión por computador

La visión por computador es la ciencia de programar un computador con el objetivo de que sea capaz de procesar, comprender y analizar imágenes y video para obtener información útil para el observador. La visión por computador tiene varios elementos que la conforman: objeto, iluminación, sistema de adquisición, procesamiento y análisis.

La figura 9 indica los elementos utilizados en el desarrollo de aplicaciones con visión artificial.

Figura 9

Elementos de la visión por computador



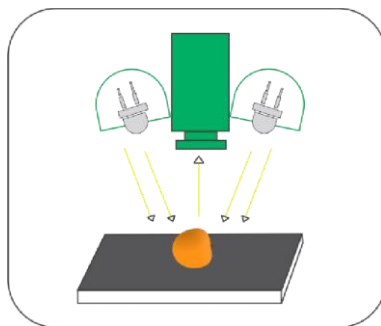
Iluminación

La iluminación es de vital importancia dentro de la visión por computador, ya que facilita de manera considerable la captura y el análisis de las imágenes capturadas al influir directamente sobre la complejidad del algoritmo. Por tal motivo, se recomienda dedicar el mayor esfuerzo a mejorar las condiciones luminosas sobre la región de interés.

Cuando un haz luminoso incide directamente sobre un material, éste puede reaccionar de diferentes formas debido a sus propiedades reflexivas, absorbentes y transmitivas (Hueso, 2018).

- **Espejo:** todo el material se refleja.
- **Cuerpo negro:** el material absorbe la luz.
- **Cristal:** la luz atraviesa el material.

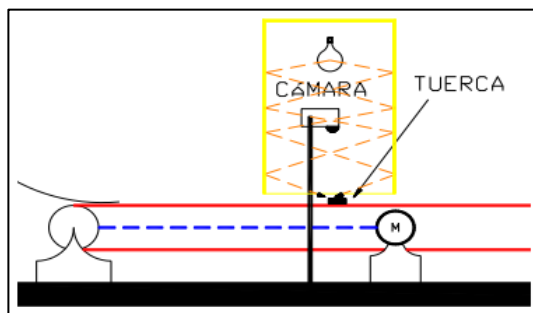
Iluminación direccional. Es utilizada en aplicaciones de localización y reconocimiento de elementos o piezas, inspección de la parte superficial de los objetos, entre otros. El objetivo está en iluminar el objeto con un haz altamente direccional para provocar sombras que resalten defectos en las piezas, es una de las técnicas de iluminación más sencillas y altamente utilizadas (Hueso, 2018). La figura 10 indica la forma correcta de iluminar de manera direccional un objeto.

Figura 10*Iluminación direccional*

Nota. Recuperado de sistema de visión artificial para la identificación de madurez de maracuyá, por Figueroa, 2016.

Iluminación difusa. Es utilizada cuando se va a capturar imágenes por medio de microscopios, en este contexto no es favorable la presencia de sombras o de sombras sin buena iluminación por lo que se recomienda que los haces luminosos incidan sobre el objeto desde todas las direcciones (Hueso, 2018).

La figura 11 indica un sistema de iluminación difusa montado sobre una banda transportadora para identificar tuercas.

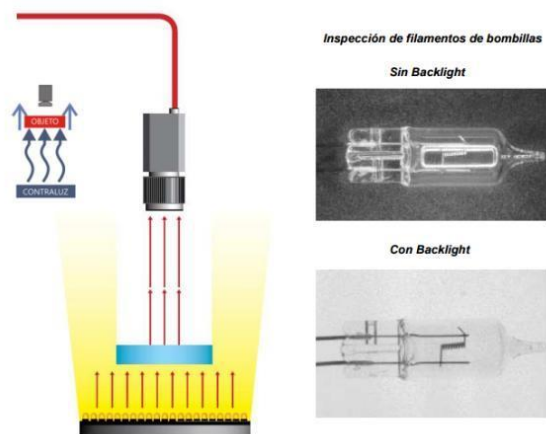
Figura 11*Iluminación difusa*

Nota. Recuperado de sistema de visión artificial para la identificación de madurez de maracuyá, por Figueroa, 2016.

Iluminación a contraluz. Es utilizada en aplicaciones de localización de piezas, control dimensional o búsqueda de todo tipo de perforaciones internas, para ello es necesario únicamente tener una silueta del objeto y es por eso que, en esta técnica la iluminación viene desde atrás del objeto, lo que provoca imágenes binarias, es decir, únicamente 2 niveles de gris. La figura 12 indica la forma de colocar una fuente de luz para crear un efecto contraluz adecuado.

Figura 12

Iluminación a contraluz



Nota. Recuperado de sistema de visión artificial para la identificación de madurez de maracuyá, por Figueroa, 2016.

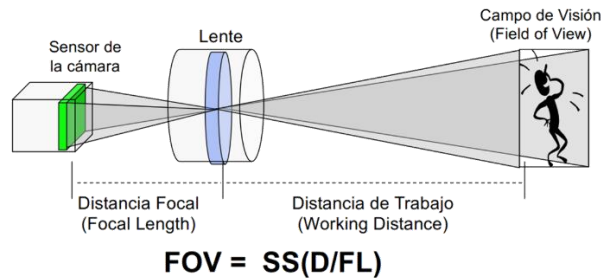
Sistema de adquisición

El sistema de adquisición de datos está formado por un sensor que captura las imágenes, un lente u óptica que debe ser seleccionado de acuerdo con el entorno en el que se va a registrar las imágenes y de ser necesario un filtro para facilitar el algoritmo de análisis. El tamaño del sensor (SS) de la cámara juega un papel importante ya que su correcta selección permitirá cubrir el campo de visión deseado a la distancia de trabajo adecuada. La mayoría de las cámaras no industriales tienen el lente incorporado, sin embargo, es posible adaptar

monturas adicionales de acuerdo con las necesidades del usuario. La figura 13 indica las variables que están directamente relacionadas con el campo de visión de una cámara.

Figura 13

Campo de Visión



FOV, campo de visión

Especificaciones del lente:

FL, distancia focal

D, distancia de trabajo

Especificaciones de la cámara:

SS, tamaño del sensor

Procesamiento y análisis

Dentro del mundo de la visión por computador existen varias formas de procesar y analizar imágenes, en este caso las agruparemos únicamente en 3 niveles de procesamiento de imágenes: Bajo, Medio y Alto nivel, todos con algoritmos distintos y cada vez más complejos.

- **Procesamiento de Bajo Nivel**
- Operaciones primitivas
- Reducción de ruido
- Manejo de contraste
- **Procesamiento de Nivel Medio**
- Segmentación
- Descripción de imágenes
- Clasificación de objetos individuales
- Manejo de atributos (Bordes, contornos, huecos, etc.)
- **Procesamiento de Alto Nivel**
- Algoritmo para toma de decisiones

- Análisis de imágenes
- Aplicaciones varias

Open Source Computer Vision Library (OpenCV)

OpenCV es una biblioteca de código abierto que permite realizar el procesamiento de imágenes a todo nivel y en varios lenguajes de programación, basado principalmente en una API de C++. Esta biblioteca tiene una estructura modular dentro de los cuales se encuentran: procesamiento de imágenes, análisis de video, calibración de cámaras, reconstrucción tridimensional, detección de objetos, descriptores de características, entrada/salida de video y GUI de alto nivel (OpenCV, 2021).

Detección de líneas

Detectar líneas de todo tipo a sido estudiado bajo varias perspectivas, para el caso de líneas rectas existe un número extenso de literatura que aborda el tema, mientras que al hablar de la detección de líneas curvas pasa todo lo contrario, la literatura es muy escasa. La detección de líneas ha sido un avance tecnológico estupendo ya que en la actualidad ha servido para crear vehículos autónomos, seguimiento de rutas, sistemas de seguridad en carretera y más.

Las propuestas que se encuentran en la literatura para la detección de líneas curvas y rectas aplicadas a la agricultura son: transformada Hough, regresión lineal, búsqueda de la acumulación de píxeles de tonalidad verde, análisis de las imágenes en el dominio de la frecuencia, búsqueda de patrones regulares, punto de fuga y exploración de franjas horizontales (García, 2018). La figura 14 indica un ejemplo de detección de líneas empleando algoritmos de visión por computador.

Figura 14

Detección de líneas



Nota. Recuperado de detección de bordes canny – OpenCV python, por Kipuna, 2021.

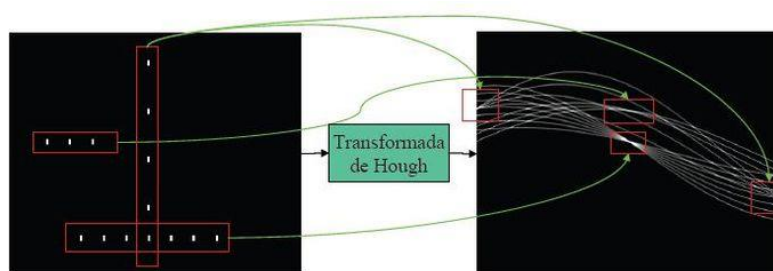
Transformada Hough

La transformada Hough es una técnica ampliamente utilizada dentro del mundo de la visión por computador con la finalidad detectar contornos en imágenes para luego obtener modelos matemáticos que representen rectas, círculos o elipses.

El objetivo principal de la transformada Hough es convertir el espacio cartesiano en una serie de parámetros que a su vez son coordenadas que forman una línea sinusoidal por cada punto encontrado en una recta, haciendo posible la creación de agrupaciones pertenecientes a los bordes de posibles figuras. La figura 15 indica la representación de líneas en el espacio de Hough.

Figura 15

Transformada Hough



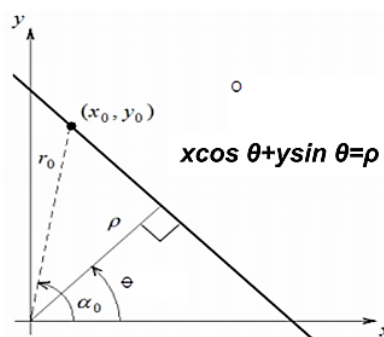
Nota. Recuperado de detección de bordes canny – OpenCV python, por Kipuna, 2021.

Dicho esto, el concepto para determinar si existe o no una línea recta es que si las curvas sinusoidales correspondientes a dos puntos se cruzan en el espacio de Hough entonces corresponde a una línea en el espacio de la imagen (Peralta y Carrasco, 2020).

El algoritmo de la transformada Hough emplea una matriz denominada acumulador, por ejemplo, para una recta el tamaño del acumulador será dos, correspondientes a los valores de p (distancia perpendicular desde origen de coordenadas a la línea) y θ (ángulo formado entre el eje x y la proyección de p), ambos empleados para obtener la ecuación de la recta y de tal forma generar una curva sinusoidal en el espacio (ρ, θ) que es única para ese punto. La figura 16 indica la representación gráfica de la ecuación de la recta que se emplea en el algoritmo de la transformada de Hough.

Figura 16

Ecuación de la recta

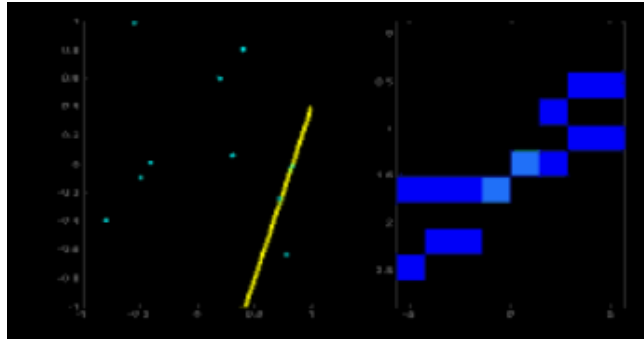


Nota. Recuperado de Comunidad argentina de inteligencia artificial, por López.

Luego de aplicar el algoritmo, cada uno de los elementos del acumulador obtendrá un valor equivalente a la sumatoria de todos los puntos posicionados en la línea representada ahora por los parámetros ρ y θ , de tal modo que el elemento con mayor valor indica una línea recta en la imagen procesada (Peralta y Carrasco, 2020). La figura 17 muestra una representación gráfica de la matriz o acumulador de la transformada de Hough.

Figura 17

Acumulador de la transformada Hough



Nota. Recuperado de Comunidad argentina de inteligencia artificial, por López.

La transformada Hough al igual que otros algoritmos tiene sus limitaciones, y la más importante es que únicamente es efectiva cuando existe una gran cantidad de puntos dentro de los elementos del acumulador y para ello la calidad de la imagen a ser procesada debe ser elevada. Detectar bordes y disminuir el ruido en la imagen son etapas previas a la aplicación del algoritmo de la transformada Hough.

Capítulo III

Metodología

En este capítulo se presenta el diseño e implementación del sistema de visión por computador para detectar automáticamente líneas de cultivo a bordo de un vehículo aéreo no tripulado. El capítulo presenta los elementos, funcionamiento y la configuración empleada para la fabricación del VANT, los algoritmos de calibración de la cámara y detección de las líneas de cultivo.

Ensamblaje y configuración del VANT

El vehículo aéreo no tripulado empleado durante el desarrollo de este trabajo es un drone armable, propulsado por cuatro rotores, cuyo cerebro es el controlador de vuelo Pixhawk compatible con Ardupilot. Una de las ventajas más importantes de emplear un drone armable, es la versatilidad que se tiene a la hora de seleccionar los dispositivos a montar sobre él.

Debido a que el desarrollo de este proyecto contempla el tratamiento de imágenes para detectar las líneas de cultivo, es necesario utilizar una computadora a bordo y en este caso se emplea una Raspberry Pi 4 Modelo B. Por otro lado, es importante considerar el consumo energético del drone durante el vuelo y la adquisición de las imágenes, es por ello que se destina dos baterías: la primera, para el consumo de los motores tanto de propulsión como del gimbal y la segunda, para el controlador de vuelo y el resto de los componentes del vehículo.

La cámara seleccionada fue una GoPro Hero 3+ debido a su compatibilidad con el gimbal del vehículo y la raspberry, además su peso de tan solo 75g y su ángulo de visión de 170°, permite capturar una escena más amplia a una menor distancia.

La masa es un factor que se debe considerar en este tipo de vehículos, ya que al ser necesarios varios componentes para la comunicación, manipulación y adquisición de datos, se necesita aliviar su peso para tener mayor autonomía durante el vuelo, es por eso que el frame

o marco del vehículo utilizado en este proyecto es de fibra de carbono, mismo que posee el espacio suficiente para montar todos los componentes necesarios. La Tabla 5 detalla todos los componentes utilizados para armar el dron.

Tabla 5

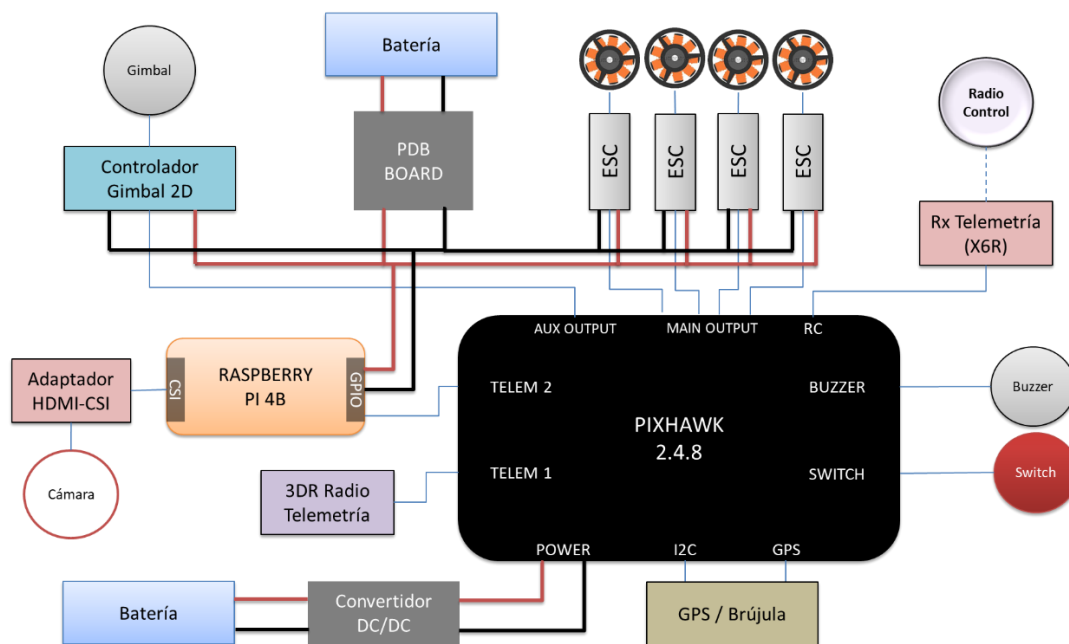
Componentes del VANT

Material	Descripción
Cámara Gopro Hero 3+ Black Edition Adventure	Modelo CHDHX-302.4 Kp15 y 2,7 Kp30, 1440p48, 1080p60, 960p100 y 720p120 fps video 12 MP fotos en hasta 30 frames por segundo.
Marco	Readytosky ZD550 Drone Frame Carbon Fiber Folding Quadcopter Frame with Carbon Fiber Landing Skid.
Gimbal con controlador	TAROT TL2D01 T2-2D 2-Axis Brushless Gimbal PTZ for Hero 4/3+/3 TL2D01FPV Gimbal.
Motor	FEICHAO 5010 350KV / 750KV Brushless Disk Motor High Torque with Mount for Hexacopter Quad Multi Copter Aircraft (350KV, 4Pcs).
Controlador ESC	MAD COMPONENTS AMPX 40A PRO ESC 40A (64.3X19.6X8.0.354 in) sin BEC para el multirotor Quadcopter Drone RC Hobby rig.
Controlador de vuelo	Readytosky Pixhawk PX4 Flight Controller Pix 2.4.8 32 Bit Flight Control Board + Safety Switch + Buzzer + I2C Splitter Expand Module + 4GB SD Card.
Controlador del sistema	Raspberry Pi 4 B+
Hélice	Hélice de fibra de carbono CW CCW CF Prop Con para T-Motor Multicopter Quadcopter Hexacopter para DIY Drone
Batería Drone	6s 6000mAh.
Batería Controlador	3s 2200mAh.
Convertidor DC-DC	DC DC Step Down Buck LM2596
Módulo USB-GPS	Readytosky M8N GPS Module Built-in Compass with GPS Antenna Mount for Standard Pixhawk 2.4.6 2.4.8 Flight Controller.
Control Remoto 3DR Telemetría	FlySky con receptor telemetría FPV Drone 3DR Radio 915MHZ V2 Telemetry 500mw w/OTG for APM2.6 2.8 Pixhawk Flight Controller.
PDB Board	Matek X Class 12S PDB Power Distribution Board FCHUB-12S Supports 8~60V DC Input, with 5V 5A and 12V 4A
Placa amortiguadora de vidrio	ShareGoo - Placa amortiguadora de fibra de vidrio para control de vuelo RC APM2.6 APM2.8 Pixhawk

Cada uno de los componentes debe ser conectado cuidadosamente al puerto que el fabricante de la Pixhawk recomienda. La Figura 18 detalla la conexión de componentes y el cableado correspondiente a la distribución de energía.

Figura 18

Diagrama de conexión del sistema

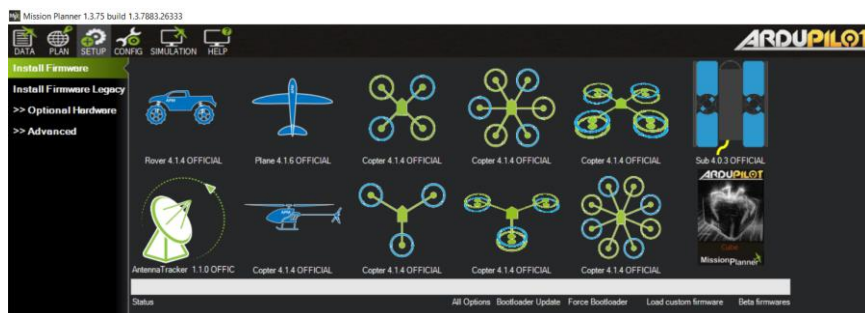


Firmware

El firmware define el tipo de vehículo que será manipulado desde el controlador de vuelo, su actualización es importante y para ello se emplea el puerto micro USB (Universal Serial Bus) del dispositivo y se lo conecta a la computadora que contiene el software de la estación terrestre, Mission Planner en este caso. La versión descargada e instalada para el desarrollo de este proyecto es ArduCopter 4.1.4. La figura 19 indica la versión de firmware utilizada en el desarrollo de este trabajo.

Figura 19

Versión de firmware



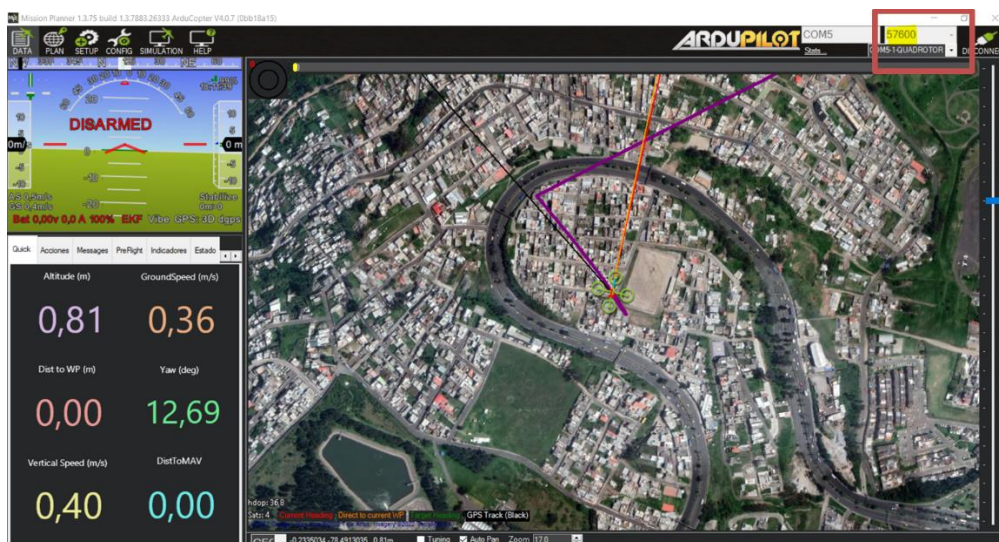
Nota. Adaptado del software Mission Planner V1.3.75

Calibración de componentes

Luego de instalar la versión de firmware correspondiente al vehículo que vamos a manipular, es necesario calibrar los sensores externos conectados al VANT, para conseguir comunicarse mediante telemetría se establece una tasa de baudios de 57600 en la pantalla principal de MP como se indica en la figura 20.

Figura 20

Tasa de baudios MP



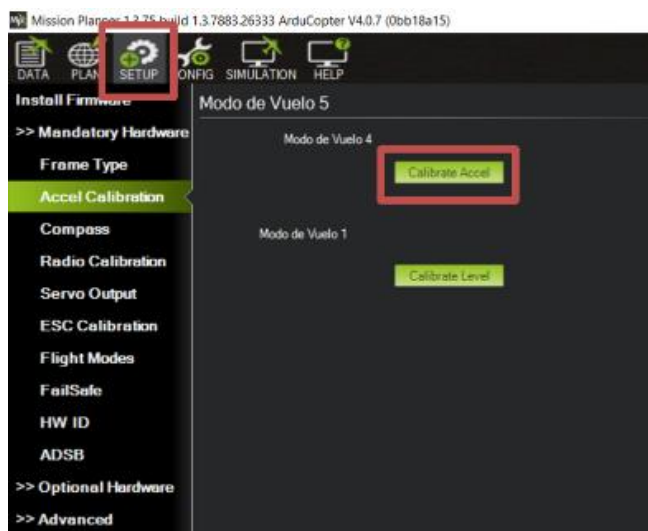
Nota. Adaptado del software Mission Planner V1.3.75

Acelerómetro. El acelerómetro es el dispositivo que mide la aceleración del vehículo, así como también ayuda a controlar su nivel durante el vuelo. Es mandatorio realizar la calibración mientras el vehículo se encuentra desarmado, para empezar, se debe ingresar en *Initial Setup*, a continuación, en *Mandatory Hardware* y luego en *Accel Calibration*.

MP indica las posiciones en las que se debe colocar el vehículo durante la calibración, es importante que el vehículo se mantenga fijo en cada posición tiempo después de haber presionado el botón de calibración paso a paso. La figura 21 indica una guía para el proceso de calibración del acelerómetro.

Figura 21

Calibración acelerómetro MP



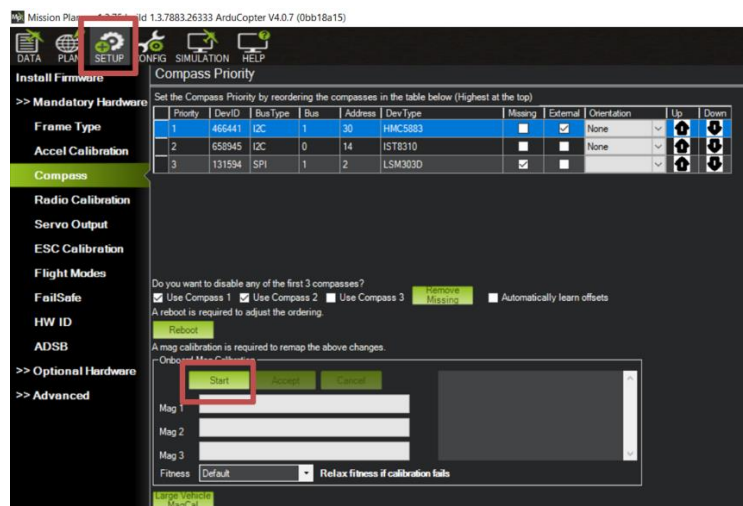
Nota. Adaptado del software Mission Planner V1.3.75

Brújula. La brújula es el dispositivo que direcciona el vehículo durante el vuelo, es importante calibrarlo correctamente y para ello es necesario alejar cualquier tipo de objeto que produzca campo magnético o metálico, debido a que puede producir una calibración errónea. La figura 22 indica la pantalla de configuración de la brújula.

Los vehículos tienen una brújula interna, sin embargo, se recomienda emplear una brújula externa para aumentar el nivel de seguridad. Para empezar la calibración, se debe ingresar en *Initial Setup*, a continuación, en *Mandatory Hardware* y luego en *Compass*.

Figura 22

Calibración brújula Mission Planner



Nota. Adaptado del software Mission Planner V1.3.75

En caso de presentarse errores durante la calibración, se puede desactivar la brújula interna o externa empleando la opción *Use this compass*.

MP solicita colocar al vehículo en las posiciones que se indican en la figura 23 y a su vez girarlo en sentido antihorario en cada una de ellas para lograr una calibración exitosa.

Figura 23

Posiciones de calibración brújula



Nota. Recuperado de documentación Ardupilot.

Transmisor RC. El transmisor RC o radio control es el elemento que permite manipular, controlar, configurar y dirigir el vehículo, así como también activar o desactivar funciones auxiliares como mover el gimbal, capturar imágenes, subir y bajar el tren de aterrizaje, etc.

Debido a la variedad de estos elementos en el mercado, es necesario ajustar los valores mínimos y máximos de cada uno de sus canales en MP. Para empezar la calibración, se debe ingresar en *Initial Setup*, a continuación, en *Mandatory Hardware* y luego en *Radio Calibration*, como se indica en la figura 24.

Figura 24

Calibración Radio Control Mission Planner



Nota. Adaptado del software Mission Planner V1.3.75

Controlador electrónico de velocidad. Los controladores electrónicos de velocidad (ESC Electronic Speed Controller) son los elementos encargados que enviar señales para activar o desactivar los motores a la velocidad deseada y configurada en el controlador de vuelo.

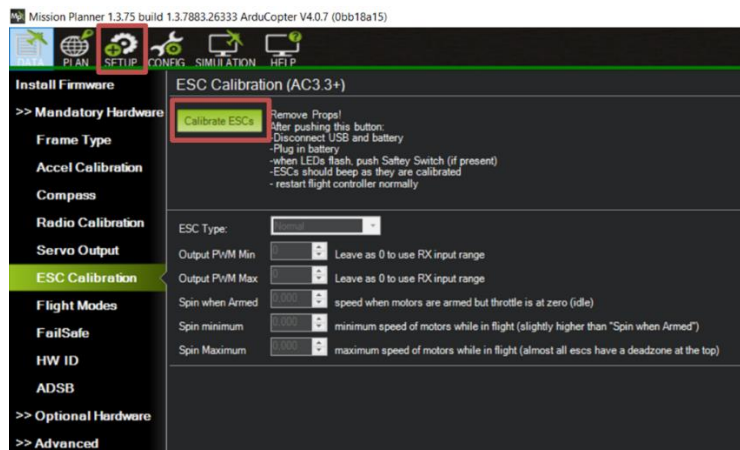
La calibración de estos elementos depende de su marca, por lo que se recomienda siempre verificar la información del fabricante respecto al modo de calibración para evitar

errores o problemas futuros. Es importante antes de empezar la calibración, verificar que no estén ensambladas las hélices y que no se encuentre conectado el controlador al computador ya que puede provocar daños severos en los puertos USB.

MP posee un método de calibración automática que funciona con la mayoría de ESC. Para empezar la calibración, se debe ingresar en *Initial Setup*, a continuación, en *Mandatory Hardware* y luego en *ESC Calibration*. La figura 25 indica la pantalla de calibración del controlador electrónico de velocidad por medio de Mission Planner.

Figura 25

Calibración ESC MP



Nota. Adaptado del software Mission Planner V1.3.75

Estabilizador de cámara

El estabilizador de cámara o gimbal, como se indica en la figura 26, es el elemento que permite mitigar las vibraciones que se generan en el drone durante el vuelo, su soporte giratorio de varios ejes puede ser controlador desde el RC o a través de MP.

Figura 26

Estabilizador de cámara

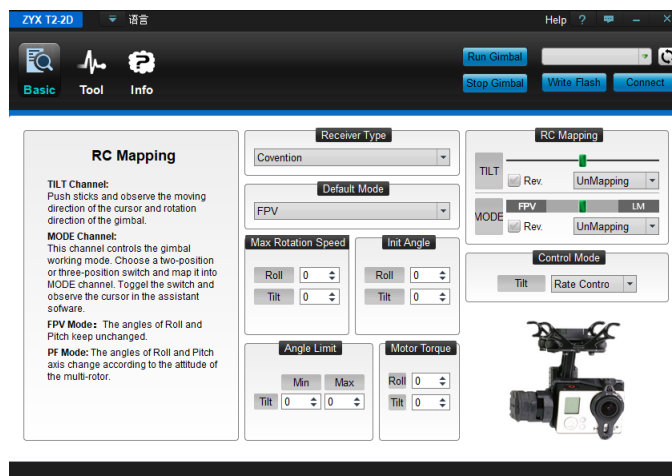


Nota. Recuperado de Wenzhou tarot aviation technology co.

En el desarrollo de este proyecto se manipula el gimbal únicamente por medio del RC empleando el canal VRB para enviar señales de movimiento en el eje Y (Pitch). ZYX T2-2D es el software empleado para calibrar el estabilizador de cámara y a su vez parametrizar las posiciones de arranque de este. La figura 27 indica la interfaz gráfica principal de configuración del estabilizador de la cámara.

Figura 27

Software de calibración del gimbal



Nota. Recuperado de Wenzhou tarot aviation technology co.

Configuración de Raspberry Pi

El control y la comunicación entre el vehículo y la estación terrestre MP, es posible debido a la utilización del protocolo MavLink y la biblioteca de código abierto Dronekit de Python. La adquisición y el tratamiento de las imágenes se lo realiza mediante la biblioteca OpenCV de Python.

A continuación, se presenta la instalación del software montado sobre la Raspberry Pi, así como también la instalación y configuración de las bibliotecas antes mencionadas, durante el proceso se lleva a cabo la instalación de bibliotecas complementarias y de ajustes en la computadora, necesarios para el correcto funcionamiento del sistema propuesto.

Software

El software libre compatible con Raspberry Pi y basado en Debian GN/Linux se lo conoce como Raspbian, debido a que este sistema se encuentra en constante desarrollo, en el presente proyecto se emplea la versión Raspbian Buster 4.19 que es la más estable al momento y cuenta con una interfaz de usuario amigable que admite acceso remoto desde consola utilizando protocolo SSH o escritorio remoto empleando una red virtual de computación (VNC Virtual Network Computing).

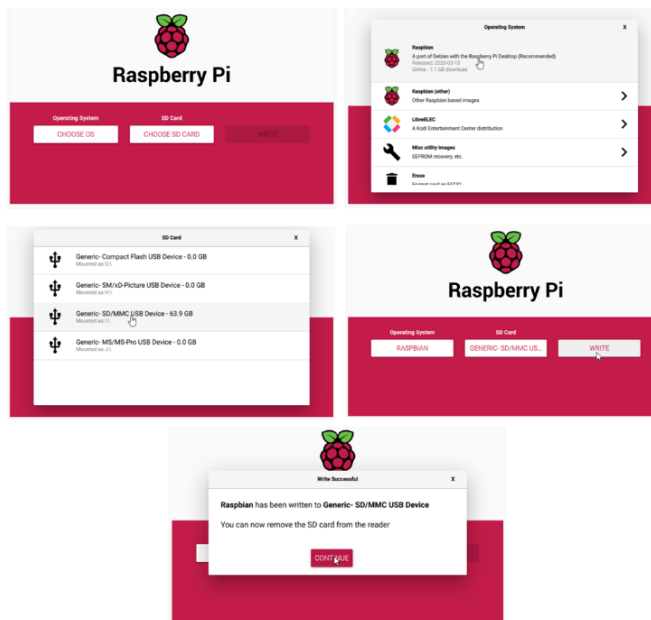
Luego de descargar la imagen o archivo ISO contenedor del sistema operativo, para instalarlo se recomienda emplear el software Raspberry Pi Imager oficial de la marca para evitar errores. Es necesario contar con una tarjeta microSD, con sistema de archivos FAT32, de al menos 16 GB de almacenamiento y como mínimo clase 10.

Una vez montada la tarjeta microSD en un computador con Raspberry Pi Imager instalado, se debe seguir los pasos indicados por el software para el montaje de la imagen sobre

la tarjeta. La figura 28 indica el proceso para instalar el software de la Raspberry Pi en una tarjeta micro SD.

Figura 28

Raspberry Pi Imager



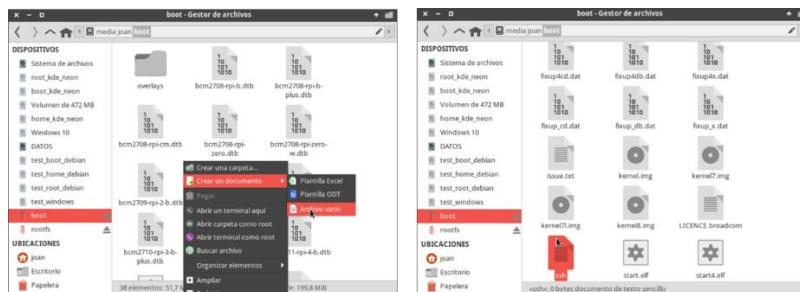
Nota. Recuperado de Raspberry Pi 4, por Raspberry Pi Foundation.

Para realizar el primer acercamiento al sistema operativo instalado se necesita de al menos 3 periféricos: teclado, mouse y pantalla. Sin embargo, si no se dispone de éstos, es posible acceder mediante las 2 alternativas mencionadas al inicio de esta sección.

La comunicación mediante protocolo SSH facilita el acceso a la Raspberry Pi ya que no necesita de periféricos extra para poder hacerlo, este protocolo utiliza una arquitectura cliente-servidor y para habilitar su conexión únicamente es necesario crear un archivo vacío sin extensión de nombre ssh en la raíz de la tarjeta microSD, como se muestra en la figura 29.

Figura 29

Conexión SSH



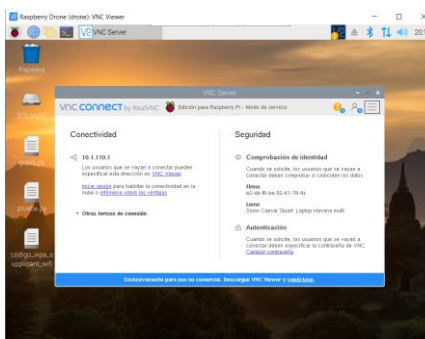
Nota. Recuperado de Raspberry Pi 4, por Raspberry Pi Foundation.

Luego, es necesario conectar la Raspberry a una red para poder acceder mediante su dirección IP, se recomienda acceder a la configuración del router para asignar siempre la misma IP a la Raspberry.

A pesar de que el método presentado anteriormente es uno de los más utilizados y simples, para el desarrollo del presente proyecto, se seleccionó el método de conexión a través de VNC debido a que la interfaz gráfica permite el monitoreo de la actividad del vehículo durante el vuelo y la adquisición de las imágenes. La figura 30 indica el escritorio remoto de Raspberry Pi.

Figura 30

VNC Viewer

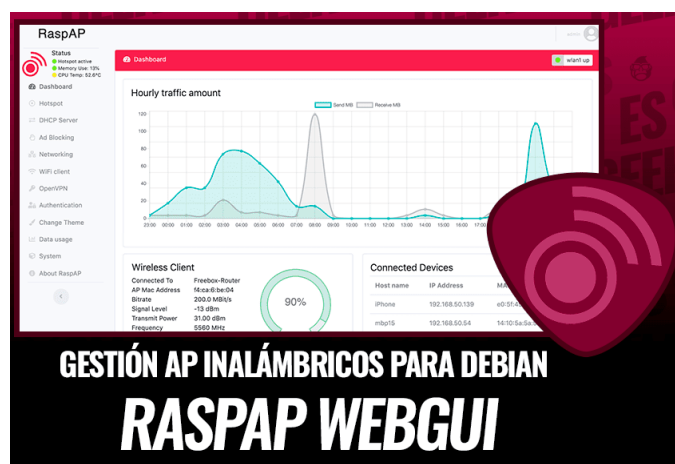


Nota. Presentación del escritorio remoto logrado a través del software VNC Viewer montado sobre una Raspberry Pi 4 B+.

Para empezar, se debe crear un punto de acceso en la placa, RaspAP es un software que posee una interfaz gráfica personalizable sencilla de enrutador inalámbrico compatible con Debian y que permite la creación y configuración de puntos de acceso de forma rápida y sencilla, como se muestra en la figura 31 (RaspAP, 2020).

Figura 31

Software RaspAP



Nota. Adaptado de RaspAP.

Para instalar RaspAP, se debe actualizar el kernel y el firmware del sistema operativo Raspbian a su última versión, seguido del reinicio de la placa, para ello se emplea los siguientes comandos en la terminal del sistema operativo.

-
1. `sudo apt-get update`
 2. `sudo apt-get full upgrade`
 3. `sudo reboot`
-

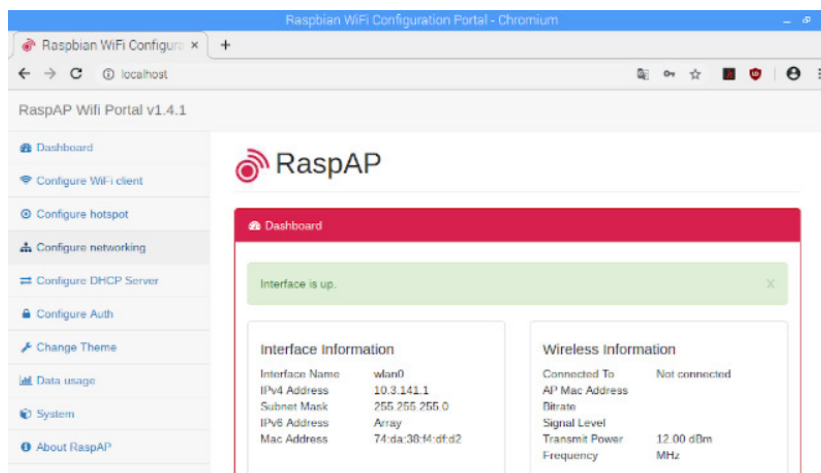
Luego del reinicio, se tiene que acceder a la configuración de la placa mediante el comando `sudo raspi-config` y parametrizar el país Wifi en las opciones de localización. Una vez finalizado ambos pasos, se puede instalar RaspAP a través del siguiente código:

```
1. curl -sL https://install.raspap.com — bash
```

Al terminar la instalación es recomendable cambiar las credenciales predeterminadas para garantizar la seguridad de acceso a la placa, esto se lo puede realizar mediante el WEBGUI de RaspAP al cual se puede acceder desde un navegador web. La figura 32 indica la GUI de RaspAP en el navegador de Raspberry Pi.

Figura 32

WEBGUI RaspAP



Nota. Adaptado de RaspAP.

Entorno virtual. Las aplicaciones desarrolladas en Python en ocasiones necesitan versiones de bibliotecas específicas, esto se vuelve complejo cuando se desea desarrollar varios proyectos con diferentes versiones de una misma librería en una única computadora.

La solución para este problema es el manejo de entornos virtuales, mismo que son espacios o directorios que contienen una versión específica de Python, así como también de sus bibliotecas.

Antes de crear un entorno virtual es necesario instalar un paquete o biblioteca que nos permita gestionarlos, en este proyecto se emplea las bibliotecas *virtualenv 20.8.1* y *virtualenvwrapper 4.8.4*, a continuación, su código de instalación: `sudo pip install virtualenv virtualenvwrapper`

Para su correcto funcionamiento se necesita modificar el archivo *bashrc*:

-
1. nano `./bashrc`
 2. export `WORKON_HOME = $HOME/.virtualenvs`
 3. export `VIRTUALENVWRAPPER_PYTHON = /usr/bin/Python`
 4. source `/usr/local/bin/virtualenvwrapper.sh`
 5. source `./bashrc`
-

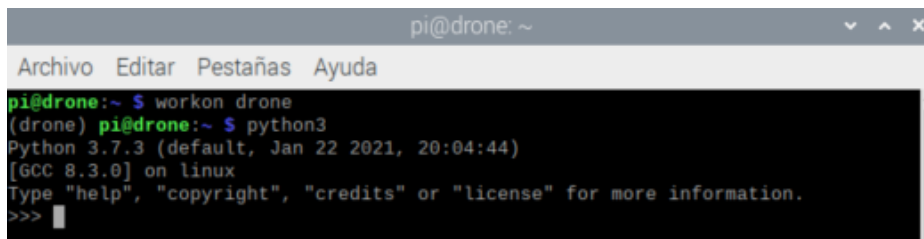
Para la creación del entorno virtual llamado *drone* y con el intérprete Python 3.7 se emplea el siguiente código:

-
1. `mkvirtualenv drone -p python3`
-

Finalmente, para validar que la creación del entorno virtual fue correcta, puede emplear el comando *workon drone*, como se muestra en la figura 33.

Figura 33

Entorno virtual



```

pi@drone: ~
Archivo Editar Pestañas Ayuda
pi@drone:~ $ workon drone
(drone) pi@drone:~ $ python3
Python 3.7.3 (default, Jan 22 2021, 20:04:44)
[GCC 8.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>

```

Nota. Visualización del comando *workon drone* ejecutado sobre la terminal de la Raspberry Pi que se encuentra montada sobre el vehículo aéreo no tripulado.

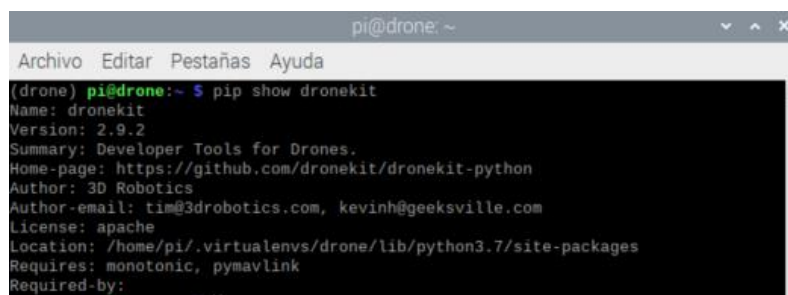
Dronekit-Python. Dronekit-Python es una interfaz de programación de aplicaciones (API) que permite a los desarrolladores de Python comunicarse con vehículos por medio del protocolo MAVLink. Esta API proporciona información del vehículo tal como: parámetros, estado del vehículo, telemetría, posición entre otros, lo cual facilita la gestión de misiones ya que se tiene control directo sobre el vehículo por medio de líneas de código. Para instalarla en la Raspberry se utiliza el siguiente comando:

1. pip install dronekit

La figura 34 indica el resultado correcto luego de instalar la biblioteca.

Figura 34

Biblioteca Dronekit



```
pi@drone: ~  
Archivo Editar Pestañas Ayuda  
(drone) pi@drone:~$ pip show dronekit  
Name: dronekit  
Version: 2.9.2  
Summary: Developer Tools for Drones.  
Home-page: https://github.com/dronekit/dronekit-python  
Author: 3D Robotics  
Author-email: tim@3drobotics.com, kevinh@geeksville.com  
License: apache  
Location: /home/pi/.virtualenvs/drone/lib/python3.7/site-packages  
Requires: monotonic, pymavlink  
Required-by:
```

Nota. Versión de la biblioteca dronekit instalada en la Raspberry Pi.

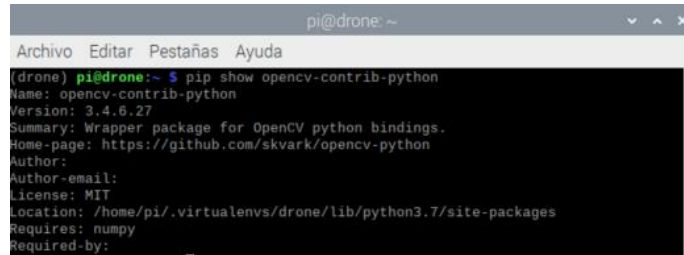
OpenCV. En el presente trabajo se requiere analizar y tratar imágenes, por lo tanto, se emplea la biblioteca de visión por computador (OpenCV), en su versión completa y funcional para Python 3. Esta biblioteca necesita dependencias y otros paquetes para su correcto funcionamiento, utilice las siguientes líneas de código para instalar la versión con módulos principales y adicionales:

-
1. sudo apt-get update
 2. sudo apt-get upgrade
 3. pip install numpy
 4. pip install matplotlib
 5. pip install opencv-contrib-python
-

La figura 35 indica la versión de OpenCV instalada.

Figura 35

Biblioteca OpenCV



```
pi@drone: ~  
Archivo Editar Pestañas Ayuda  
(drone) pi@drone:~$ pip show opencv-contrib-python  
Name: opencv-contrib-python  
Version: 3.4.6.27  
Summary: Wrapper package for OpenCV python bindings.  
Home-page: https://github.com/skvarck/opencv-python  
Author:  
Author-email:  
License: MIT  
Location: /home/pi/.virtualenvs/drone/lib/python3.7/site-packages  
Requires: numpy  
Required-by:
```

Nota. Versión de la biblioteca opencv instalada en la Raspberry Pi, *contrib* indica que se encuentra la versión completa para python.

MAVLink. La comunicación entre el controlador de vuelo y la Raspberry se realiza mediante el protocolo MAVLink a través de conexión serie.

Raspberry Pi posee pines de propósito general (General Purpose Input/Output, GPIO) que pueden ser configurados para establecer comunicación con el controlador de vuelo por medio de sus puertos de telemetría.

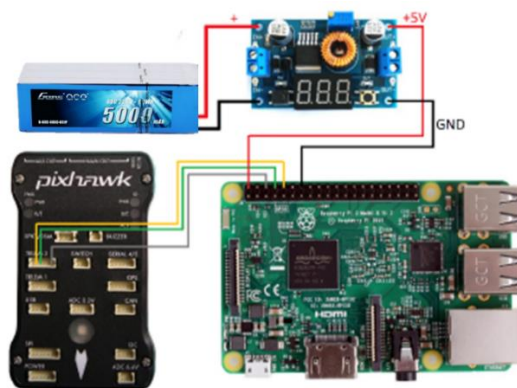
El puerto *TELEM2* de la tarjeta controladora de vuelo debe conectarse a los pines *Tx*, *Rx* y *GND*, el pin de 5V se recomienda conectarlo a una fuente de alimentación diferente a través de un regulador de voltaje.

Es importante verificar el voltaje que recibe la Raspberry, dado que en su versión 4 los requisitos de alimentación son muy exigentes y cualquier variación afecta directamente en su funcionamiento, lo que impide el intercambio de información entre la Raspberry y la Pixhawk.

La figura 36 indica la conexión entre la tarjeta controladora de vuelo y la Raspberry.

Figura 36

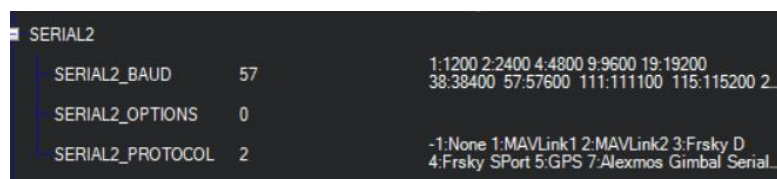
Conexión Pixhawk-Raspberry



En el software Mission Planner, es obligatorio configurar los siguientes parámetros del controlador de vuelo: *Serial2_Protocol = 2* y *SerialBaud = 57*, el primero, para habilitar la comunicación mediante el protocolo MAVLink en el puerto *TELEM2* y el segundo para establecer la velocidad de comunicación en baudios, como se muestra en la figura 37.

Figura 37

Parámetros TELEM2



Los puertos GPIO 14 y GPIO 15 deben estar configurados como un puerto serie de transmisión y recepción, *Tx* y *Rx*, respectivamente. Raspberry de forma nativa designa el puerto

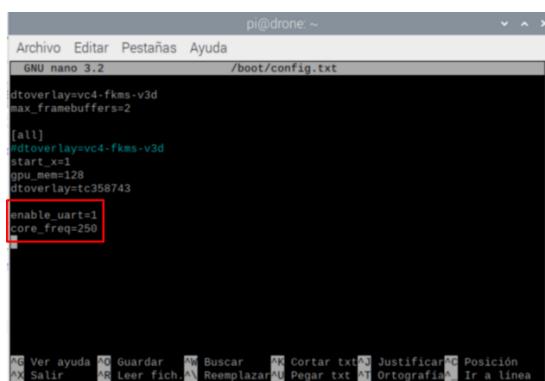
serie *ttyS0* a la interfaz *UART*, para habilitarlo se debe modificar el archivo de arranque del sistema de Raspbian con el siguiente código: `sudo nano /boot/config.txt`

En la parte final del archivo añade `enable_uart = 1` y `core_freq = 250` para que el reloj central de la Raspberry sea utilizado a una frecuencia de 250MHz por el puerto UART.

Finalmente guarde los cambios y reinicie la tarjeta para que los cambios se efectúen. La figura 38 indica el archivo `config.txt` modificado.

Figura 38

Habilitación puerto UART



```
pi@drone: ~  
Archivo Editar Pestañas Ayuda  
GNU nano 3.2 /boot/config.txt  
dtoverlay=vc4-fkms-v3d  
max_framebuffers=2  
  
[all]  
#dtoverlay=vc4-fkms-v3d  
start_x=1  
gpu_mem=128  
dtoverlay=tc358743  
enable_uart=1  
core_freq=250  
  
Ver ayuda Guardar Buscar Cortar txt Justificar Posición  
Salir Leer fich. Reemplazar Pegar txt Ortografía Ir a línea
```

Calibración de la cámara

Las cámaras disponibles en el mercado tienen la capacidad de capturar imágenes de diferentes formas, calidad y ángulo. En la mayoría de los casos estas cámaras presentan distorsión, lo que provoca errores de precisión en sistemas de medición visual para posicionamiento. Por lo tanto, la calibración de la cámara es una tarea obligatoria que está directamente relacionada con este proyecto.

La calibración de la cámara permite mitigar 3 tipos de distorsión:

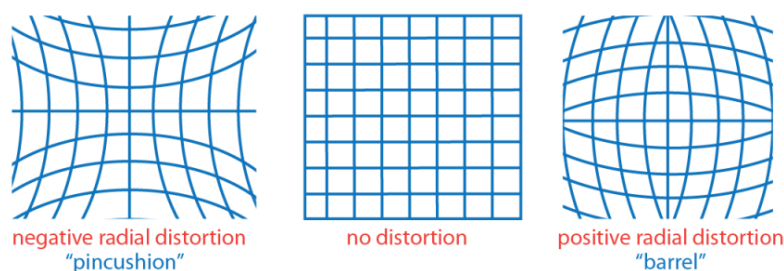
- Distorsión radial positiva: conocida como barril, cuyas líneas se muestran con una curvatura convexa.

- Distorsión radial negativa: conocida como alfiler, cuyas líneas se muestran con una curvatura cóncava.

La figura 39 indica los tipos de distorsión generados por una cámara con lente gran angular.

Figura 39

Distorsión radial

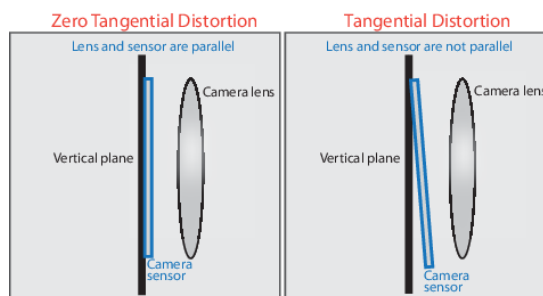


Nota. Representación de los tipos de distorsión que afectan a las imágenes capturas con cámaras que poseen un lente gran angular. Recuperado de Comunidad argentina de inteligencia artificial, por López.

- Distorsión tangencial: imágenes con áreas más cercanas que otras, debido a que el punto central de la lente no se encuentra paralelo al sensor de la cámara, como se muestra en la figura 40.

Figura 40

Distorsión tangencial



Nota. Adaptado de Comunidad argentina de inteligencia artificial, por López.

La aproximación de una escena 3D a una imagen 2D se la puede hacer mediante la utilización de parámetros intrínsecos y extrínsecos de la cámara de acuerdo con el método de calibración del profesor Zhang Zhengyou desarrollado en 1998, método empleado comúnmente en proyecto de visión artificial.

Los parámetros intrínsecos son independientes de la escena, es decir, una vez encontrados, éstos pueden ser utilizados para otras escenas. Los parámetros de su matriz son las distancias focales (f_x, f_y) en píxeles y las coordenadas del punto central de la imagen (C_x, C_y).

$$\bar{A} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \quad (1)$$

Los parámetros extrínsecos hacen referencia a la matriz de rotación-traslación que representa el movimiento de los objetos frente a una cámara estática. El objetivo de estos parámetros es transformar coordenadas (X, Y, Z) de una escena 3D en un punto de coordenadas 2D (x, y).

$$\bar{R}|\vec{t} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \quad (2)$$

La biblioteca de OpenCV tiene la capacidad de obtener la matriz de parámetros intrínsecos junto a los coeficientes de distorsión radial y tangencial, como se representa en las ecuaciones 3, 4, 5 y 6.

$$X_{\text{corregido}} = x'(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \quad (3)$$

$$Y_{\text{corregido}} = y'(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \quad (4)$$

$$x_{\text{corregido}} = x' + [2p_1 x' y' + p_2 (r^2 + 2x'^2)] \quad (5)$$

$$y_{\text{corregido}} = y' + [2p_2 x' y' + p_1 (r^2 + 2x'^2)] \quad (6)$$

Donde (x, y) son las coordenadas no distorsionadas en píxeles, (x', y') son las coordenadas adimensionales que representan la imagen normalizada, k_1, k_2 y k_3 son los

coeficientes de distorsión radial y finalmente p_1 y p_2 son los coeficientes de distorsión tangencial.

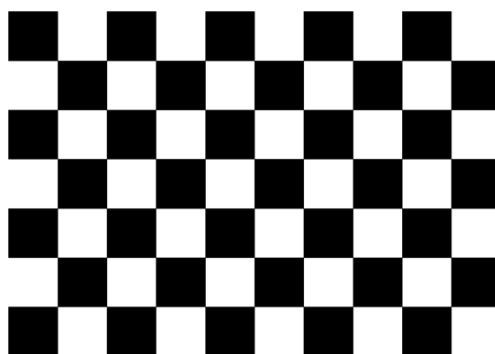
Adquisición de imágenes

Calibrar la cámara tiene como objetivo disminuir la distorsión radial y tangencial generada por la lente y el sensor para obtener imágenes planas, para lograrlo es necesario alimentar al sistema de calibración con al menos diez imágenes capturadas desde varios ángulos y en diferentes escenas.

Para que la calibración sea exitosa se debe emplear un patrón como referencia, en el desarrollo de este proyecto se emplea una imagen de un tablero de ajedrez de 9x6, cuyos cuadrados miden 25x25 mm, similar a la figura 41.

Figura 41

Patrón de calibración



Nota. Recuperado de Comunidad argentina de inteligencia artificial, por López.

En el presente proyecto se emplea una cámara GoPro Hero 3+ para capturar las imágenes, las imágenes fueron capturadas a una distancia del tablero de ajedrez de 600 mm.

El código de programación empleado para realizar la captura y guardarla en la memoria interna de la Raspberry se presenta a continuación.

```
1. import cv2
2. cam = cv2.VideoCapture("http://10.5.5.9:8080/live/amba.m3u8", cv2.CAP
   FFMPEG)
3. cam.set(cv2.CAP_PROP_BUFFERSIZE,0)
4. cv2.namedWindow("test", cv2.WINDOW_AUTOSIZE)
5. img counter = 0
6. while True:
7.     ret, frame = cam.read()
8.     cv2.imshow("test", frame)
9.     if not ret:
10.         break
11.     k = cv2.waitKey(1)
12.     if k%256 == 27:
13.         break
14.     elif k%256 == 32:
15.         img name = "OpenCV .png".format(img counter)
16.         cv2.imwrite(img name, frame)
17.         print(" written!".format(img name))
18.         img counter += 1
19.cam.release()
20.cv2.destroyAllWindows()
```

La cámara GoPro puede ser empleada como un punto de acceso, es por eso que la conexión entre la Raspberry y ésta se realiza mediante WiFi y se emplea el método VideoCapture para acceder a los fotogramas enviados por la cámara mediante protocolo http.

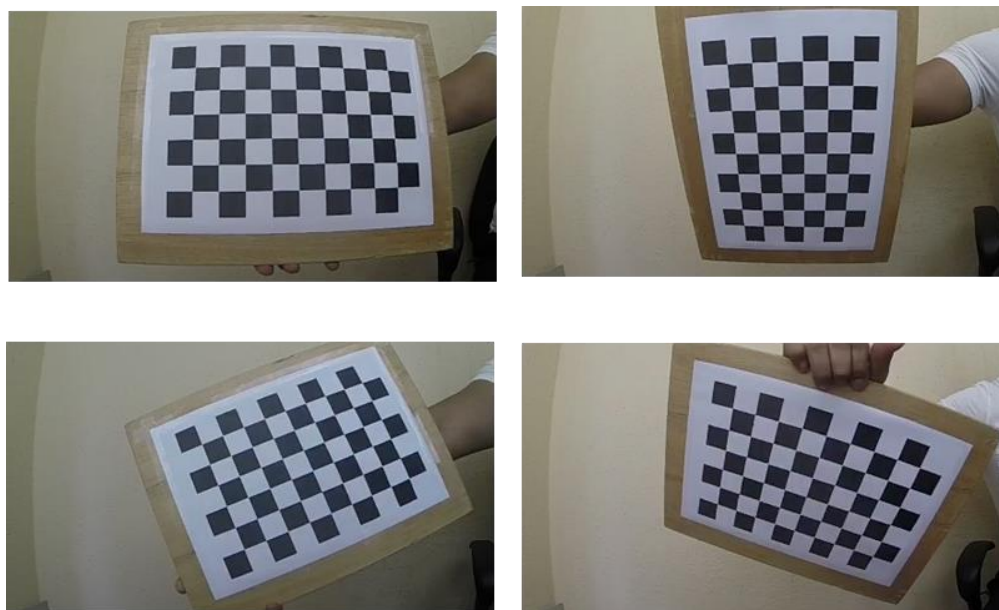
Las diferentes posiciones en las que se capturan las imágenes, tomando en cuenta que la cámara siempre debe estar fija, ayudan a que la calibración sea lo menos errónea posible y permiten identificar los puntos del objeto donde se presenta mayor distorsión. La posición, profundidad e inclinación, son parámetros que están directamente relacionados con la

ubicación de las esquinas del tablero de ajedrez, la distancia medida desde la cámara y el plano en el que se capturan los fotogramas.

La figura 42 presenta algunas de las imágenes utilizadas para calibrar la cámara.

Figura 42

Capturas del patrón de calibración



Nota. Capturas del patrón desde diferentes ángulos para calibrar la cámara GoPro.

Parámetros intrínsecos y coeficiente de distorsión

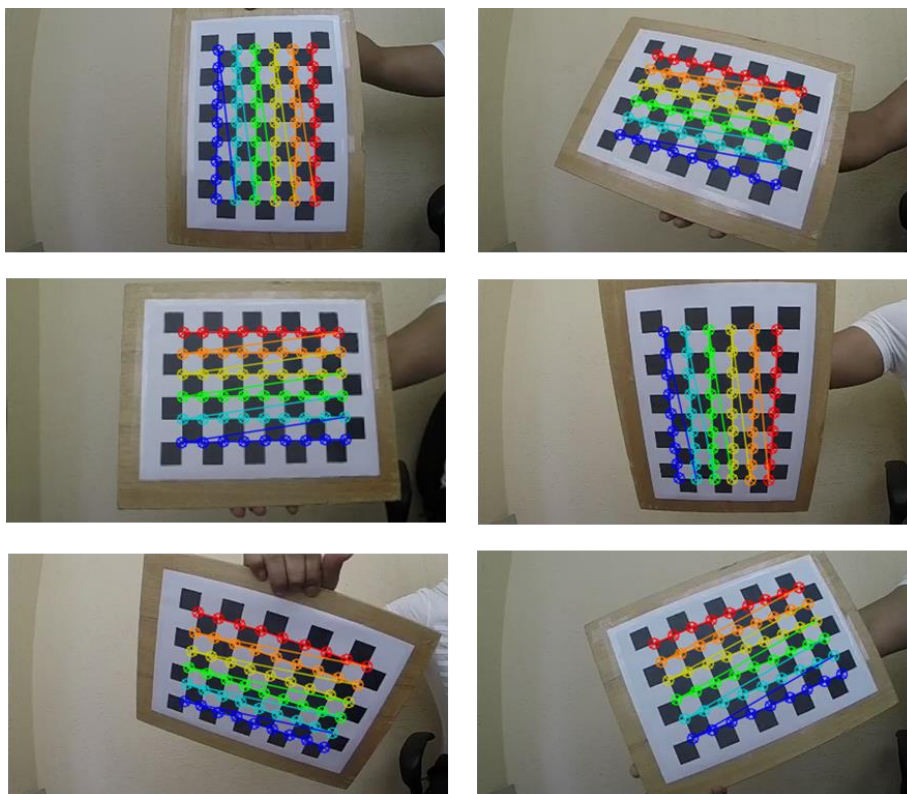
La matriz de parámetros intrínsecos o matriz de cámara y el vector de los coeficientes de distorsión, son el objetivo de la calibración de la cámara, estos valores son los que permiten la corrección de distorsión en los fotogramas.

El proceso consiste en encontrar las posiciones de las esquinas de los cuadrados negros del patrón de calibración y definir sus coordenadas en el mundo real haciendo referencia a los puntos 3D con la ayuda de la medida real de los cuadrados del patrón de calibración, este proceso se debe realizar en cada una de las imágenes capturadas anteriormente previamente transformadas a escala de grises.

La figura 43 indica fotografías capturadas a diferentes ángulos con los puntos de intersección de color negro detectados.

Figura 43

Detección de esquinas en el patrón



Nota. Detección de esquinas utilizando el código de calibración de la biblioteca *opencv*.

Luego de encontrar los vértices en el patrón, se busca reajustarlos para que se coloquen de manera equidistante entre sí empleando el método *cornerSubPix*, su iteración y exactitud se pueden controlar mediante los valores *EPS* y *MAX_ITER*.

A continuación, se busca definir una coordenada en píxeles por cada punto 3D de las imágenes para finalmente obtener la matriz de la cámara y el vector de distorsión utilizando el método *calibrateCamera*.

Las ecuaciones 7 y 8 muestran los valores encontrados durante la calibración desarrollada para el presente proyecto.

$$\text{Matriz de cámara} = \begin{bmatrix} 213.4539 & 0 & 221.1307 \\ 0 & 211.8863 & 132.4178 \\ 0 & 0 & 1 \end{bmatrix} \quad (7)$$

$$\text{Coeficiente de distorsión} = [-0.2657 \ 0.1282 \ 0.001 - 0.0025 - 0.0298] \quad (8)$$

Conversión pixel a mm

El campo de visión (Field Of View, FOV) de la cámara GoPro puede ser modificado en base al área que se necesita capturar y a la distancia en la que ésta se encuentre del objetivo. El tamaño del sensor (Sensor Size, SS), la distancia de trabajo (Working Distance, D) y la longitud focal (Focal Length, FL) de la cámara son parámetros importantes a la hora de encontrar la máxima área que se puede capturar a una resolución específica.

La ecuación 9 representa la relación directa que existe entre los parámetros antes mencionados y el campo de visión. Debido a que la resolución a lo largo y ancho de una captura no es igual en este tipo de cámaras, es necesario determinar el tamaño del pixel de forma independiente, la ecuación 10 indica la fórmula para encontrar el tamaño de cada pixel en milímetros.

$$FOV = SS \frac{D}{FL} \quad (9)$$

$$\text{Tamaño pixel} = \frac{FOV}{\text{Resolución}} \quad (10)$$

GoPro en su información técnica indica los parámetros necesarios para realizar la conversión de píxeles a milímetros.

En la tabla 6 se presenta la información utilizada en el desarrollo del presente proyecto, tales como: tamaño del sensor, longitud focal, campo de visión horizontal y vertical.

Tabla 6*Información técnica GoPro Hero 3+*

Modo de imagen	FOV (°)	FOV (°)	FL (mm)	SS (mm)
	Vertical	Horizontal	Longitud Focal	Tamaño del Sensor
Medium	92.2	94.4	3	4.65x3.49

Nota. Adaptado de cámaras y accesorios GoPro.

La distancia de trabajo empleada para el desarrollo del presente trabajo fue de 5000 mm, valor suficiente para cubrir el área del cultivo.

Las imágenes capturadas fueron obtenidas mediante código de programación en python, el cuál utiliza los parámetros obtenidos en la sección 3.3.1.1 para disminuir la distorsión de la escena, las imágenes presentan una resolución de 320x240 píxeles.

Las ecuaciones 11 y 12 presenta el cálculo realizado para convertir los pixeles horizontales en milímetros.

$$FOV = 4.65 \frac{5000}{3} \quad (11)$$

$$FOV = 7750 \text{ mm}$$

$$\text{Tamaño pixel horizontal} = \frac{7750}{320} \quad (12)$$

$$\text{Tamaño pixel horizontal} = 24.21 \text{ mm}$$

Las ecuaciones 13 y 14 presenta el cálculo realizado para convertir los pixeles verticales en milímetros.

$$FOV = 3.49 \frac{5000}{3} \quad (13)$$

$$FOV = 5816.6 \text{ mm}$$

$$\text{Tamaño pixel vertical} = \frac{5816.6}{240} \quad (14)$$

$$\text{Tamaño pixel vertical} = 24.23 \text{ mm}$$

Detección de líneas de cultivo

Los sistemas de visión por computador a bordo de vehículos aéreos no tripulados son utilizados en varias tareas relacionadas con agricultura de precisión, incluyendo la detección de las hileras de cultivo. Los sistemas de posicionamiento generalmente emplean el GPS como herramienta de guiado, la actualización de sus mapas no es continua y por lo tanto, están muy alejadas de lo que sucede constantemente.

Los cultivos que se encuentran al aire libre se ven afectados directamente por los cambios de iluminación debido a las condiciones climáticas de la zona, esto afecta la calidad de las imágenes. Otros problemas que suelen presentarse son: hileras discontinuas, tamaño del cultivo debido a las diferentes etapas de crecimiento, hierbas malas con colores similares a los sembríos de las hileras e hileras curvas.

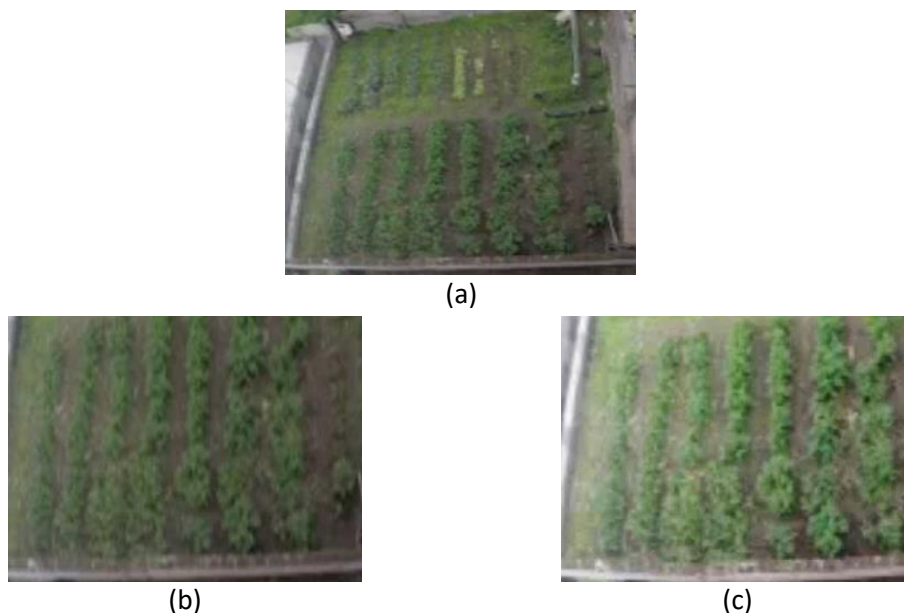
En el presente trabajo se emplea técnicas de tratamiento de imágenes para eliminar el ruido de éstas y se consideraron los siguientes aspectos:

- una línea de cultivo es una distribución espacial acumulada de píxeles con un rango de color verde que sigue un patrón lineal y/o curvo;
- una línea de cultivo curva se proyecta como una o varias líneas rectas para obtener sus coordenadas (X, Y);
- la escena capturada debe contener líneas de cultivo completas para limitar el espacio de búsqueda de la transformada de Hough; y,
- las coordenadas se obtienen de forma relativa al punto central de la imagen, mismo que se considera el punto de inicio de la misión de vuelo.

Tomando en cuenta las consideraciones mencionadas, el cultivo empleado para el desarrollo de este proyecto presenta líneas rectas, curvas, espacios entre las hileras y hierba mala, con el objetivo de demostrar la robustez del sistema.

Figura 44

Cultivo con diferente intensidad de luz solar



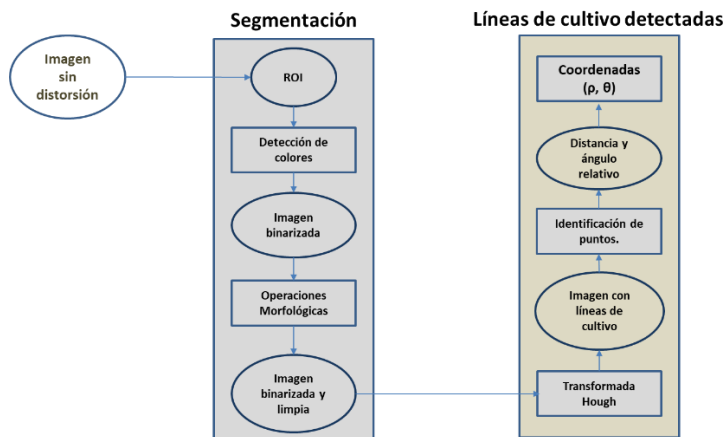
Nota. (a) Cultivo original RGB; (b) Región de interés en día oscuro; (c) Región de interés en día claro.

Método de visión

El sistema propuesto considera tres etapas generales de desarrollo: Segmentación, detección de líneas de cultivo y obtención de coordenadas, como se indica en la figura 45.

Figura 45

Arquitectura del sistema



Segmentación. Detectar hileras en cultivos que se encuentran a campo abierto y bajo condiciones climáticas variables es una tarea que necesita ser lo más precisa posible. En este sentido, para conseguir mejorar el nivel de precisión del sistema, se discrimina hierbas malas, líneas curvas y espacios entre las hileras mediante la aplicación de técnicas de segmentación.

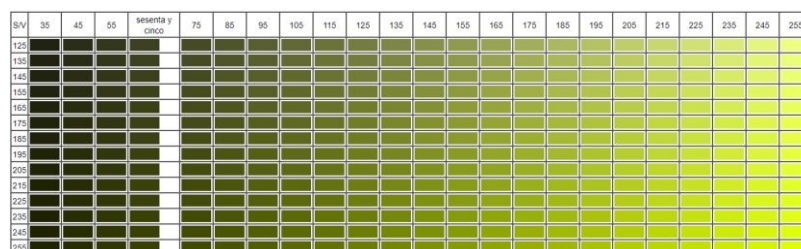
Tomando en cuenta estas consideraciones, la etapa de segmentación fue diseñada con los siguientes procesos:

a) Identificación de hileras

Las hileras de los cultivos pueden ser segmentadas por su tonalidad verdosa, es importante considerar que los niveles de verde no son iguales en todos los cultivos. Por lo tanto, se recomienda trabajar con varios niveles de verde, debido a esto se transformó el espacio de color de la captura de RGB (Red, Green, Blue) a HSV (Hue, Saturation, Value) desarrollado por Smith en 1978. La figura 46 indica un campo de tonalidades de color verde.

Figura 46

Paleta de color verde HSV

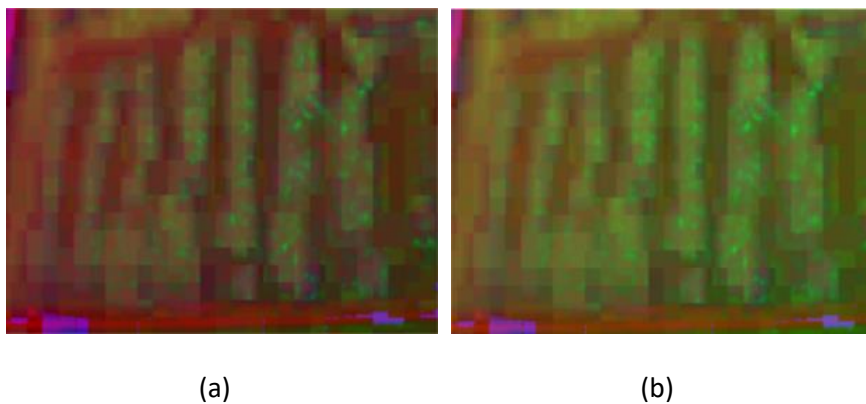


Nota. Recuperado de Learning OpenCV: Computer vision with the OpenCV library, por Bradski.

El cambio de brillo debido a las condiciones climáticas se controla mediante la búsqueda e identificación de píxeles con un nivel por debajo de 205 en el canal de valor o brillo. Una vez identificados los píxeles, se realiza un ajuste de 35 puntos en más, valor encontrado de forma experimental mediante pruebas en campo. La figura 47 indica imágenes con el tratamiento del brillo y sin el tratamiento del brillo.

Figura 47

Cultivo en el espacio de color HSV



Nota. (a) Imagen original HSV; (b) Imagen original con el brillo ajustado

Bajo las consideraciones antes mencionada, se determinó que los rangos óptimos de matiz (hue), saturación (saturation) y valor (value) son los mostrados en la siguiente tabla.

Tabla 7

Niveles HSV

Nivel	Matiz	Saturación	Valor
Mínimo	43	74	0
Máximo	53	255	126

Nota. Niveles HSV utilizados para filtrar las hileras de los cultivos que garantizar la cobertura de varios tipos de colores verde dependiendo de su intensidad.

El resultado de la identificación de hileras sobre la región de interés se en la figura 48.

Figura 48

Identificación de hileras



b) Operaciones morfológicas.

La transformación morfológica fue realizada en la imagen binaria obtenida de la identificación de hileras. Se realizó una operación de erosión (erode) seguida de un filtrado basado en la eliminación de píxeles no correspondientes a las hileras bajo el criterio de un elemento estructurante en forma de cruz y una vecindad cuyo tamaño es de 2x2. Bajo este concepto, el objetivo principal en esta etapa es eliminar el ruido de la imagen provocada por la hierba mala. La figura 49 indica los resultados obtenidos luego de aplicar operaciones morfológicas.

Figura 49

Imagen original filtrada empleando operaciones morfológicas



c) Detección de bordes.

El proceso para detectar bordes tiene su complejidad debido a que es necesario inicializar parámetros que no siempre van a satisfacer las condiciones de iluminación en una determinada área.

En este proyecto se utilizó Sobel como detector de bordes ya que está basado en el cálculo de la primera derivada y su resultado indica cambios de intensidad. Para su desarrollo se utilizó un tamaño de kernel 3 y la convolución con la imagen filtrada se muestra en la siguiente imagen. La figura 50 indica los bordes encontrados con el algoritmo de Sobel.

Figura 50

Detección de bordes en la imagen binaria



El propósito de detectar los bordes fue disminuir la cantidad de datos de la imagen original, conservando únicamente aquellos que son relevantes para encontrar las posiciones de las hileras del cultivo.

Detección de líneas de cultivo curvas y rectas. Las líneas de cultivo curvas y rectas fueron detectadas de acuerdo con los bordes encontrados en la imagen original. En este trabajo, las líneas de cultivo que presentan curvaturas fueron reemplazadas con líneas rectas proyectadas desde la posición inicial de la hilera a su posición final, como se muestra en la figura 51.

Figura 51

Recta proyectada en línea de cultivo curva



Bajo esta consideración, se aplica la transformada de Hough a la región de interés para encontrar una recta entre aquellos pixeles que se encuentren alineados a lo largo de las hileras del cultivo y tengan una longitud mínima de 150 pixeles, equivalente a 3000 milímetros. Debido a que se desconoce de la cantidad de líneas de cultivo que se pueden encontrar, el acumulador tiene un tamaño dinámico y se utilizó una resolución de 1 pixel y 1 grado, valores que proporcionan una precisión suficiente en aplicaciones de agricultura según García.

La figura 52 se presenta la imagen original con las líneas rectas y las proyecciones de las curvas detectadas por el sistema.

Figura 52

Detección de líneas curvas y rectas en el cultivo



Nota. El sistema detecta los bordes de las hileras de cultivo y proyecta una línea recta que se aproxima a cada una de ellas.

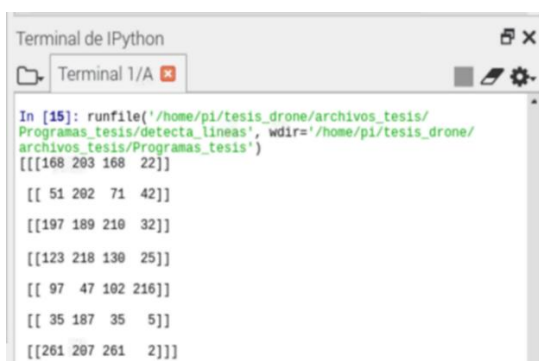
Obtención de coordenadas. Para obtener las coordenadas para planear la misión de vuelo, se toma en consideración lo siguiente:

- La posición inicial del dron en la misión de vuelo es la coordenada (TAKE OFF) central de la imagen.
- Los puntos de inicio y fin de cada línea detectada son las coordenadas (WAYPOINT) que guían al dron durante la misión de vuelo.
- El sistema entrega coordenadas polares y relativas a la posición anterior tomando en cuenta que las líneas de cultivo van de izquierda a derecha en la imagen.

La transformada de Hough entrega las coordenadas rectangulares en pixeles de las líneas detectadas en el cultivo, tomando como origen la esquina superior izquierda de la imagen. La imagen 53 indica las coordenadas en pixeles detectadas por el algoritmo.

Figura 53

Coordenadas rectangulares de las líneas de cultivo en px



```
Terminal de IPython
Terminal 1/A
In [15]: runfile('/home/pi/tesis_drone/archivos_tesis/
Programas_tesis/detecta_lineas', wdir='/home/pi/tesis_drone/
archivos_tesis/Programas_tesis')
[[[168 293 168 22]]
 [[ 51 292 71 42]]
 [[197 189 210 32]]
 [[123 218 130 25]]
 [[ 97 47 102 216]]
 [[ 35 187 35 5]]
 [[261 297 261 2]]]
```

Nota. Posición inicial y final de cada línea de cultivo en coordenadas rectangulares, medidas en pixeles.

En el mundo real es necesario que los valores se encuentren en unidades de longitud. Por lo tanto, se emplea los tamaños de pixeles encontrados en la sección 3.3.1 para transformar los valores de pixeles a milímetros, como se indica en la figura 54.

Figura 54

Coordenadas rectangulares de las líneas de cultivo en mm

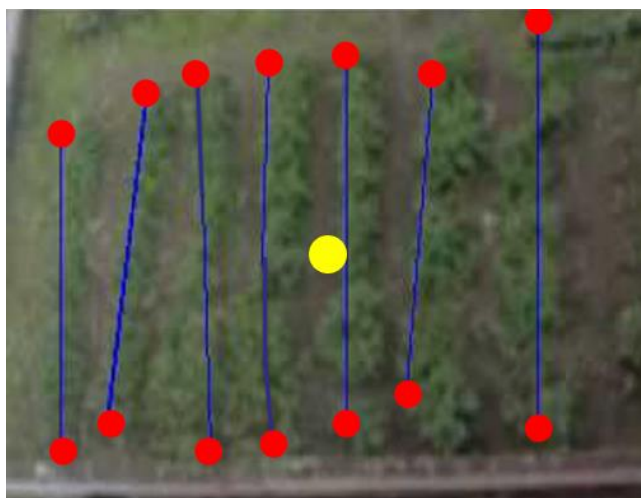
```
Terminal de IPython
Terminal 1/A
In [1]: runfile('/home/pi/tesis_drone/archivos_tesis/
Programas_tesis/detecta_lineas', wdir='/home/pi/tesis_drone/
archivos_tesis/Programas_tesis')
[[[4070.64 4918.69 4070.64 533.06]]
[[1235.73 4894.46 1720.33 1017.66]]
[[4773.31 4579.47 5088.3 775.36]]
[[2980.29 5282.14 3149.9 605.75]]
[[2350.31 1138.81 2471.46 5233.68]]
[[ 848.05 4531.01 848.05 121.15]]
[[6324.03 5015.61 6324.03 48.46]]]
```

Nota. Posición inicial y final de cada línea de cultivo en coordenadas rectangulares, medidas en milímetros.

La figura 55 muestra en color amarillo la coordenada central de la imagen y en color rojo las coordenadas rectangulares de inicio y fin de cada línea de cultivo.

Figura 55

Coordenadas del sistema de detección de líneas curvas y rectas



Para identificar la secuencia, las coordenadas rectangulares en X se ordenan de menor a mayor siempre y cuando Y sea mayor que el valor medio de la resolución vertical.

La figura 56 indica las coordenadas de las líneas de cultivo siguiendo la secuencia de izquierda a derecha de la imagen del cultivo.

Figura 56

Coordenadas de las líneas de cultivo en secuencia

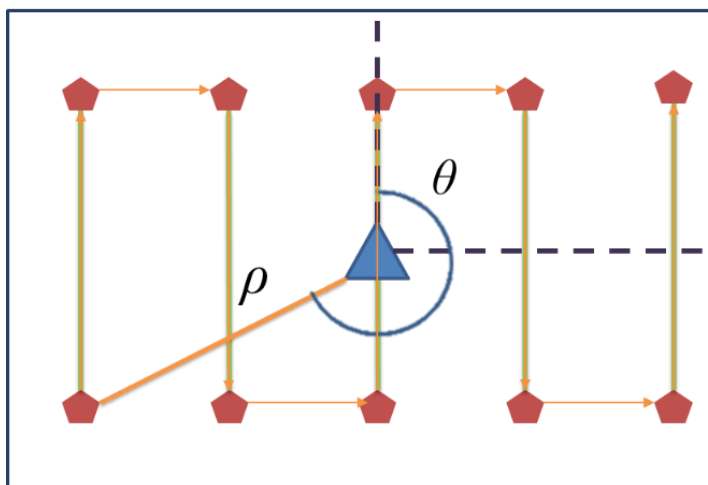
```
In [9]: runfile('/home/pi/tesis_drone/archivos_tesis/
Programas_tesis/detecta_lineas', wdir='/home/pi/tesis_drone/
archivos_tesis/Programas_tesis')
[[ 848.05 4531.01  848.05  121.15]]
[[1235.73 4894.46 1720.33 1017.66]]
[[2350.31 1138.81 2471.46 5233.68]]
[[2980.29 5282.14 3149.9  605.75]]
[[4070.64 4918.69 4070.64  533.06]]
[[4773.31 4579.47 5088.3  775.36]]
[[6324.03 5015.61 6324.03  48.46]]
```

Nota. Coordenadas de las líneas de cultivo ordenadas de izquierda a derecha, secuencia que seguirá el drone cuando se planee la misión de vuelo automática.

Finalmente, se calcula las coordenadas relativas a la posición de la hilera anterior y no al origen de la imagen, como se muestra en la figura 57.

Figura 57

Secuencia de vuelo



Nota. Representación gráfica de la secuencia que el drone seguirá durante la misión de vuelo automática.

Planeación de misión de vuelo automática.

Planear una misión de vuelo puede ser una tarea sencilla siempre y cuando dispongamos de un GPS estable y un mapa actualizado de nuestra región de interés. Para poder hacerlo se debe conocer las coordenadas o waypoints a los que el vehículo tiene que dirigirse. así como también la altura de vuelo, misma que es un valor que puede ser configurado directamente en el planificador de vuelo.

La primera posición de vuelo del vehículo se obtiene mediante la comunicación entre la Raspberry y la PixHawk, a continuación, se muestra el código para adquirir las coordenadas geográficas iniciales.

```
1. from dronekit import connect
2. uav = connect('/dev/ttyS0' wait_ready=True baud = 57600)
3. global location = uav.location.global_frame
```

En el desarrollo del presente proyecto, el mapa no se encuentra actualizado y por lo tanto es complicado crear una misión de vuelo sin conocer las coordenadas del cultivo. La figura 58 muestra el mapa con un espacio vacío donde actualmente se encuentra el cultivo utilizado.

Figura 58

Ubicación del cultivo



Nota. Imagen obtenida mediante el mapa de Mission Planner.

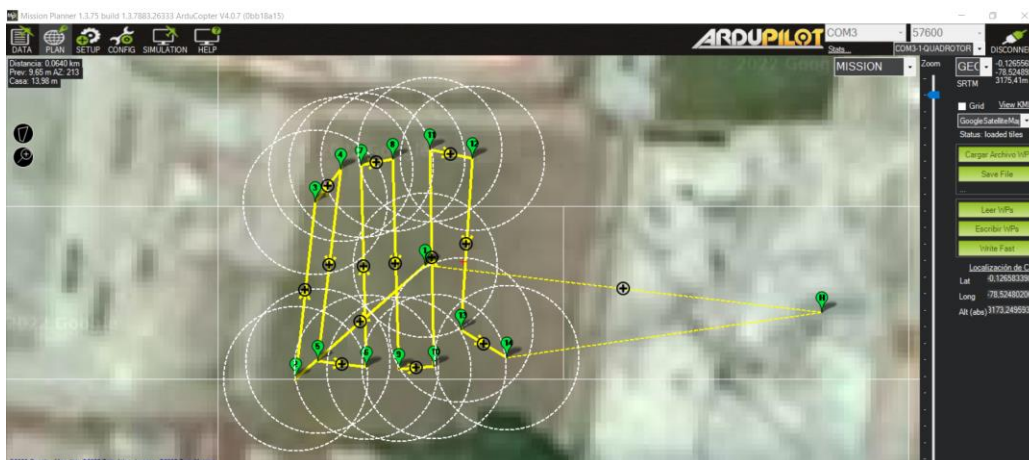
Para crear la misión de vuelo automática se utilizó el planificador de misiones de Mission Planner, empleando las coordenadas obtenidas por el sistema de detección de líneas de cultivo.

Mission Planner dispone de una sección en la esquina superior izquierda donde se observa en tiempo real la distancia y ángulo a la que se colocará el siguiente waypoint, el software tiene la capacidad de transformar estos valores en coordenadas geográficas. Se ingresó de forma manual los valores acordes a los obtenidos en el sistema, cubriendo así el alcance del presente trabajo.

La figura 59 muestra las coordenadas implementadas en el planificador de misiones de Mission Planner.

Figura 59

Planificación de misión de vuelo automática



Nota. Planificación de la misión de vuelo en Mission Planner utilizando las coordenadas relativas obtenidas mediante el sistema desarrollado en el presente proyecto.

Es importante tomar en cuenta que la primera y la última posición de la misión corresponden al modo de despegue (TAKEOFF) y el modo de regreso a casa (RETURN TO

LAUNCH) respectivamente. La figura 60 indica el ejemplo de la configuración de una misión de vuelo automática.

Figura 60

Waypoints de la misión de vuelo automática

Radio WP		Radio Perder Tiempo por D		Absolut		Verify Height		Agregar Abajo		Alt Warn		Spline			
2.00	#5	100								0					
	Comandos	Delay			Lat	Long	Alt	Frame	Borrar	Amb	Abaj	Grad %	Angle	Dist	AZ
1	TAKEOFF	0	0	0	-0.1265717	-78.5249006	10	Relative	X			90.6	42.2	14.9	277
2	WAYPOINT	0	0	0	-0.1265998	-78.5249328	5	Relative	X			-10...	-46.5	6.9	229
3	WAYPOINT	0	0	0	-0.1265561	-78.5249280	5	Relative	X			0.0	0.0	4.9	6
4	WAYPOINT	0	0	0	-0.1265479	-78.5249216	5	Relative	X			0.0	0.0	1.2	38
5	WAYPOINT	0	0	0	-0.1265955	-78.5249273	5	Relative	X			0.0	0.0	5.3	187
6	WAYPOINT	0	0	0	-0.1265970	-78.5249152	5	Relative	X			0.0	0.0	1.4	97
7	WAYPOINT	0	0	0	-0.1265469	-78.5249166	5	Relative	X			0.0	0.0	5.6	358
8	WAYPOINT	0	0	0	-0.1265453	-78.5249087	5	Relative	X			0.0	0.0	0.9	79
9	WAYPOINT	0	0	0	-0.1265975	-78.5249072	5	Relative	X			0.0	0.0	5.8	178
▶ 10	WAYPOINT	0	0	0	-0.1265968	-78.5248985	5	Relative	X			0.0	0.0	1.0	85
11	WAYPOINT	0	0	0	-0.1265430	-78.5248995	5	Relative	X			0.0	0.0	6.0	359
12	WAYPOINT	0	0	0	-0.1265452	-78.5248889	5	Relative	X			0.0	0.0	1.2	102
13	WAYPOINT	0	0	0	-0.1265878	-78.5248918	5	Relative	X			0.0	0.0	4.7	184
14	WAYPOINT	0	0	0	-0.1265946	-78.5248804	5	Relative	X			0.0	0.0	1.5	121
15	RETURN_TO_LAUNCH	0	0	0	-0.1265401	-78.5248770	5	Relative	X			0.0	0.0	6.1	4

Capítulo IV

Resultados

En este capítulo se presentan los resultados obtenidos durante el desarrollo y experimentación del sistema, así como también de los algoritmos y métodos empleados para filtrar las imágenes. A continuación se presentan los resultados evaluados en un cultivo de 45 metros cuadrados con problemas de hierba mala, hileras curvas y discontinuas.

Detección de líneas de cultivo

La cámara GoPro Hero 3+ posee una lente gran angular que genera alta distorsión en las capturas tomadas. Se realizaron varias pruebas para conseguir la menor distorsión posible, consiguiendo como resultado una reducción considerable en la resolución de las imágenes y un porcentaje de error del 2% en los bordes, como se indica en la figura 61.

Figura 61

Comparativa de capturas con y sin distorsión



(a)

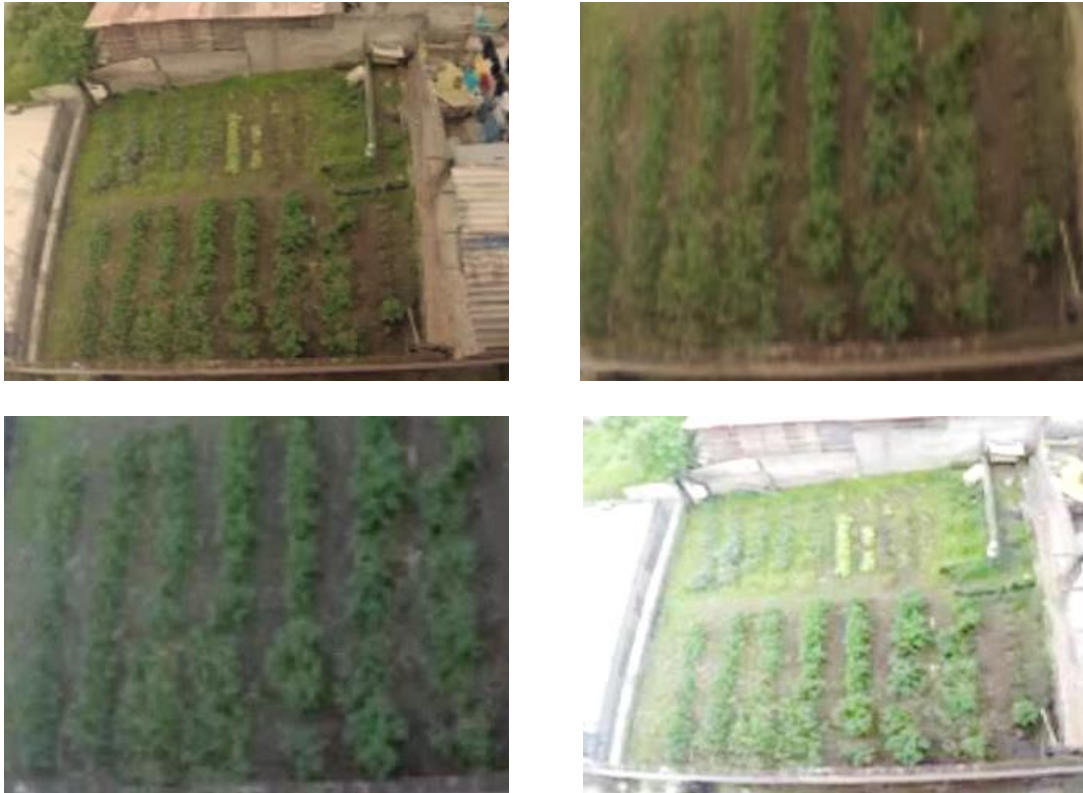
(b)

Nota. (a) Imagen con distorsión; (b) Imagen sin distorsión.

En cuanto a la variación en la iluminación de la escena, se realizaron capturas bajo diferentes tonalidades de brillo y a diferentes alturas, teniendo como resultado que a mayor intensidad de brillo menor error en la detección de líneas ya que los colores verdes resaltan. Sin embargo, en condiciones con poca iluminación, la detección de las líneas de cultivo se ve comprometida ya que debido al bajo contraste el sistema confunde algo de tierra con las hileras.

Figura 62

Capturas en diferentes condiciones de iluminación



Para mitigar este problema se realizó el ajuste del brillo en las capturas mediante el siguiente código.

-
1. `if (imagen_hsv[:, :, 1]).any() + 50 <= 255 :`
 2. `imagen_hsv[:, :, 1] += 34`
-

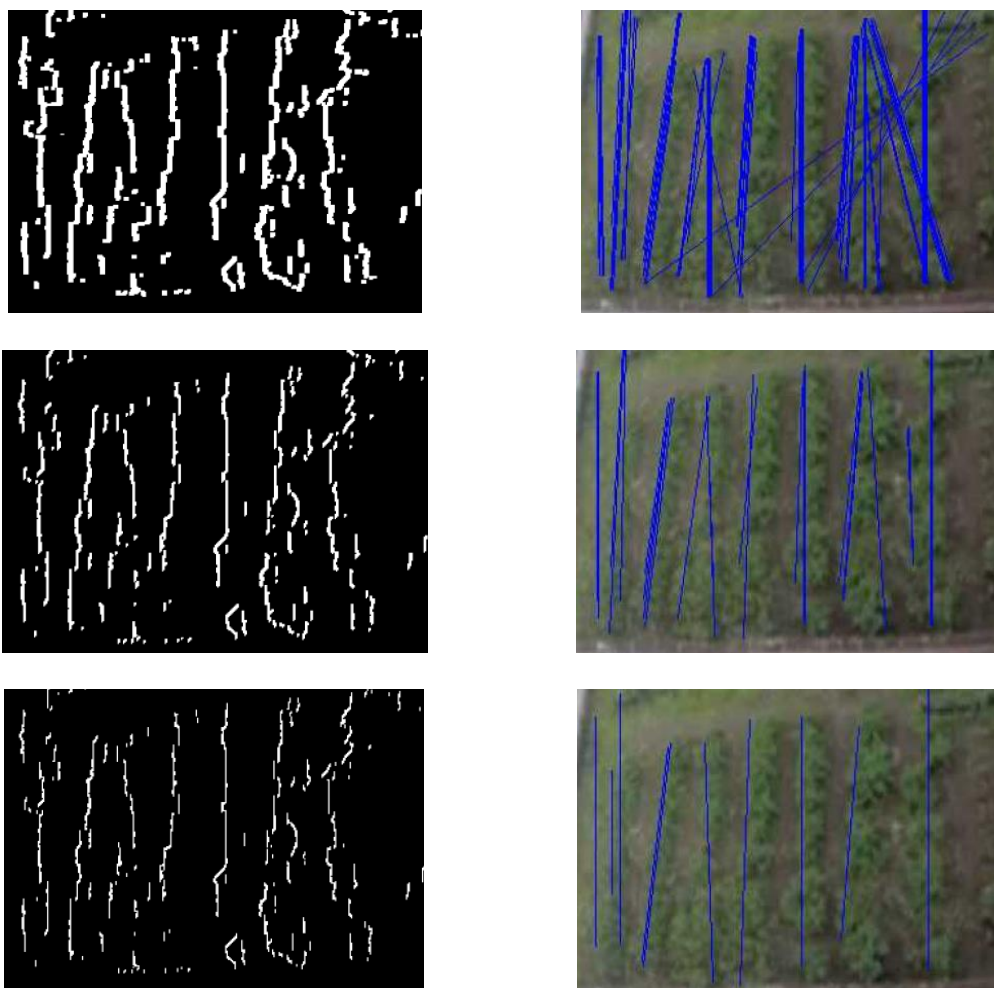
Imagen_hsv hace referencia a la imagen transformada al espacio de color HSV, donde el canal 3 corresponde a los niveles de brillo, mismo que fue manipulado de manera automática cada vez que se detecta un pixel de bajo brillo. El valor de ajuste positivo de 34 puntos fue determinado de forma experimental luego de realizar varias pruebas en imágenes con

diferentes niveles de brillo. Para segmentar de manera correcta las líneas de cultivo de la hierba mala, se realizaron pruebas con diferentes tipos de elementos estructurantes: elipse, rectangular, cuadrado y cruz, siendo el elemento en forma de cruz de tamaño 2x2 el óptimo.

La figura 63 se muestra los resultados obtenidos durante las pruebas realizadas, como se puede ver el elemento estructurante juega un papel muy importante para eliminar líneas que no corresponden a las hileras del cultivo.

Figura 63

Imágenes filtradas con diferentes elementos estructurantes



Precisión del sistema

Con el objetivo de evaluar de forma cuantitativa la precisión del sistema propuesto y validar que se mejora el guiado de un vehículo aéreo no tripulado, se obtuvo la distancia en metros de cada hilera de forma manual para después compararlo con la distancia obtenida por el sistema. La medición manual fue realizada con una cinta métrica de carril y al centro de cada una de las hileras del cultivo.

En promedio se obtuvo un error del 93% en la precisión correspondiente a la medición lineal de las hileras de cultivo. La tabla 8 presenta los resultados comparativos obtenidos en el desarrollo de este trabajo, se considera la enumeración de las hileras de izquierda a derecha en la imagen.

Tabla 8

Precisión del sistema

	Hilera 1	Hilera 2	Hilera 3	Hilera 4	Hilera 5	Hilera 6	Hilera 7
Sistema (mm)	4409,86	3906,97	4096	4679,46	4385,63	3817,12	4967,15
Manual (mm)	4650	4300	4450	4980	4650	4100	5300
Precisión	95%	91%	92%	94%	94%	93%	94%

Misión de vuelo autónoma.

Debido a que los mapas no se actualizan constantemente con Mission Planner, planificar una misión de vuelo sin las imágenes reales del sitio resulta casi imposible y por lo tanto la precisión de las posiciones de vuelo no son acertadas. La precisión del guiado mejora efectivamente ya que el sistema de detección de líneas de cultivo obtiene los waypoints con una precisión del 93% y no depende de imágenes de mapas ni de GPS.

Los resultados de la misión de vuelo automática del vehículo aéreo no tripulado fueron validados de forma visual ya que no existe en el país un sistema similar para realizar una comparativa.

La figura 64 muestra el UAV durante la misión de vuelo autónoma.

Figura 64

VANT durante una misión de vuelo autónoma



Capítulo V

Conclusiones y recomendaciones

Conclusiones.

- La utilización de cámaras con lentes gran angular provocan distorsión en las imágenes y comprometen la precisión en tareas de medición. Por lo tanto, es estrictamente necesario realizar una calibración para obtener sus parámetros intrínsecos y sus coeficientes de distorsión para mitigar este problema.
- Los sistemas de visión por computador a bordo de vehículos aéreos no tripulados ayudan considerablemente a la mejora continua de las tareas en el campo de la agricultura de precisión.
- La utilización de software y hardware libre en el desarrollo de aplicaciones sobre vehículos aéreos no tripulados resulta un reto desafiante debido a la rapidez en la que cambian las versiones de trabajo, lo cual provoca errores de compatibilidad entre sensores, controladores y actuadores.
- La lógica utilizada para el sistema propuesto tiene como base 3 etapas generales de desarrollo: segmentación, detección de líneas de cultivo y obtención de coordenadas.
- Los algoritmos de visión por computador ofrecen ventajas en el desarrollo de proyectos sobre vehículos que utilizan baterías. Sin embargo, el consumo energético se ve comprometido debido al alto consumo de estos sistemas.
- El sistema propuesto presenta una precisión promedio del 93% en la medición lineal de las hileras de cultivo y en cuanto a las coordenadas para la misión de vuelo únicamente se realizó una validación visual ya que en el país no existe un sistema similar para poder realizar una comparativa.

- En la revisión de la literatura se encuentra varios trabajos para detectar líneas de cultivo. Sin embargo, todos se enfocan en implementar algoritmos sobre vehículos terrestres.
- La calibración de los sensores, actuadores y parámetros del vehículo aéreo no tripulado son importantes para evitar accidentes durante el vuelo, ya que estos están directamente relacionados con su estabilidad y rendimiento.

Recomendaciones.

- Configurar velocidades y alturas en el controlador de vuelo para evitar daños y accidentes durante las misiones autónomas.
- Monitorear el vuelo del vehículo desde una estación terrestre permite detectar errores para poder actuar a tiempo sobre ellos.
- Configurar siempre un modo de piloto RTL para que el dron regrese a casa si por algún motivo se pierda el control del mismo.
- La alimentación de la Raspberry Pi debe ser lo más estable posible, las variaciones de voltaje provocan errores en la transmisión de información mediante el protocolo MAVLink.
- La utilización de un entorno virtual y el respaldo de las versiones de las bibliotecas empleadas facilitan el avance de trabajos futuros.
- Realizar pruebas en campo abierto para evitar altercados con personas u objetos ajenos al proyecto.
- Para aplicaciones que empleen video en tiempo real se debe emplear cámaras que permitan acceso cableado a su fuente de video, el uso de comunicación inalámbrica provoca retardo en la adquisición de frames.

Trabajos Futuros

Tomando en cuenta la línea base desarrollada en este trabajo, se puede derivar trabajos futuros:

- Guiado del vehículo en tiempo real durante misiones de vuelo automáticas utilizando las coordenadas obtenidas por el sistema de detección de líneas.
- Adquisición de datos relevantes a agricultura de precisión durante misiones de vuelo automática.
- Análisis de plagas y hierba mala en tiempo real durante misiones de vuelo.

Bibliografía

Acuerdo Ministerial N° 299. Registro Oficial N° 34 de 2013. Normativa General para promover y regular la producción orgánica – ecológica – biológica en el Ecuador. 11 de julio de 2013.

Alarcón, F., García, M., Maza, I., Viguria, A., & Ollero, A. (2019). A precise and gnss-free landing system on moving platforms for rotary-wing UAVs. *Sensors*, *19*(4), 886.

ArduPilot Dev Team, "Documentación ArduPilot ." [En línea]. Disponible: <https://ardupilot.org/ardupilot/>

Åstrand, B., & Baerveldt, A. J. (2005). A vision based row-following system for agricultural field machinery. *Mechatronics*, *15*(2), 251-269.

Barreda Sánchez, J. L. (2011). Seguimiento visual de líneas de cultivo.

Bradski, G., & Kaehler, A. (2008). *Learning OpenCV: Computer vision with the OpenCV library.* " O'Reilly Media, Inc."

Culjak, I., Abram, D., Pribanic, T., Dzapo, H., & Cifrek, M. (2012, May). A brief introduction to OpenCV. In *2012 proceedings of the 35th international convention MIPRO* (pp. 1725-1730). IEEE.

Cho, A., Kang, Y. S., Park, B. J., Yoo, C. S., & Koo, S. O. (2011, October). Altitude integration of radar altimeter and GPS/INS for automatic takeoff and landing of a UAV. In *2011 11th International Conference on Control, Automation and Systems* (pp. 1429-1432). IEEE.

Dai, T., Li, A., & Wang, H. (2019, October). Design and Implementation of a Relay Mechanism in Restricted UAVs Ad-hoc Networks. In *2019 IEEE 19th International Conference on Communication Technology (ICCT)* (pp. 605-611). IEEE.

Figuroa, DE y Guerrero, ER (2016). Sistema de visión artificial para la identificación de la madurez de la fruta de la pasión (granadilla)/Sistema de visión artificial para la identificación del estado de madurez de frutas (granadilla). *Redes de Ingeniería* , *7* (1), 78-87.

García, E., & Flego, F. (2008). *Agricultura de precisión. Revista Ciencia y Tecnología*, *1*(8), 99-116.

García García, I. (2017). Estudio sobre vehículos aéreos no tripulados y sus aplicaciones.

García Santillán, I. D. (2018). *Métodos de visión por computador para detección automática de líneas de cultivo curvas/rectas y malas hierbas en campos de maíz* (Doctoral dissertation, Universidad Complutense de Madrid).

Gebbers, R., & Adamchuk, V. I. (2010). Precision agriculture and food security. *Science*, 327(5967), 828-831.

GoPro. *Cámaras y accesorios GoPro*. Recuperado de <https://gopro.com/es/ec/shop/cameras>

Guato, E. (2019). *ANÁLISIS DE LAS REDES DE SENSORES INALÁMBRICOS EN LA AGRICULTURA DE PRECISIÓN EN EL ECUADOR*. Universidad Agraria del Ecuador.

Guerrero Hernández, J. M. (2015). *Sistema de visión para agricultura de precisión: identificación en tiempo real de líneas de cultivo y malas hierbas en campos de maíz* (Doctoral dissertation, Universidad Complutense de Madrid).

Guy, C. (2006, October). Wireless sensor networks. In *Sixth international symposium on instrumentation and control technology: Signal analysis, measurement theory, photo-electronic technology, and artificial intelligence* (Vol. 6357, p. 635711). International Society for Optics and Photonics.

HUESO, A. D. L. E. (2018). *Visión Por Computador*.

Ikeda, M., Honda, T., Ishikawa, S., & Barolli, L. (2014, November). Performance comparison of DTN routing protocols in Vehicular-DTN environment. In *2014 Ninth International Conference on Broadband and Wireless Computing, Communication and Applications* (pp. 247-252). IEEE.

Kataoka, T., Kaneko, T., Okamoto, H., & Hata, S. (2003, July). Crop growth estimation system using machine vision. In *Proceedings 2003 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM 2003)* (Vol. 2, pp. b1079-b1083). IEEE.

Khalifeh, A. F., AlQudah, M., Tanash, R., & Darabkh, K. A. (2018, October). A simulation study for UAV-aided wireless sensor network utilizing ZigBee protocol. In *2018 14th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)* (pp. 181-184). IEEE.

- Kipuna, 2021. Detección de bordes Canny – OpenCV python. Recuperado de <https://noemioocc.github.io/posts/Detecci%C3%B3n-de-bordes-Canny-openCV-python/>.
- Koubâa, A., Allouch, A., Alajlan, M., Javed, Y., Belghith, A., & Khalgui, M. (2019). Micro air vehicle link (mavlink) in a nutshell: A survey. *IEEE Access*, 7, 87658-87680.
- S. Lee y Y. Choi, "Reviews of unmanned aerial vehicle (drone) technology trends and its applications in the mining industry," pp. 197–204, jul 2016.
- López, R. (s.f.). Libro de IAAR. Comunidad argentina de inteligencia artificial. Buenos Aires.
- Montalvo Martínez, M. (2015). *Técnicas de visión artificial para la segmentación y detección de líneas de cultivo en imágenes agrícolas* (Doctoral dissertation, Universidad Complutense de Madrid).
- Nonami, K., Kendoul, F., Suzuki, S., Wang, W., & Nakazawa, D. (2010). *Autonomous Flying Robots*. Tokyo: Springer.
- OpenCV, "OpenCV: Introduction." [En línea]. Disponible: <https://docs.opencv.org/master/d1/dfb/intro.html>
- O'Shea, K., & Nash, R. (2015). An introduction to convolutional neural networks. *arXiv preprint arXiv:1511.08458*.
- Peralta, R. A., Carrasco, D. C., Tito, P. C., & Delgado, P. (2020). Métodos para el procesamiento de imágenes digitales.
- Rao, A., Shao, H., & Yang, X. (2019, May). The Design and Implementation of Smart Agricultural Management Platform Based on UAV and Wireless Sensor Network. In *2019 IEEE 2nd International Conference on Electronics Technology (ICET)* (pp. 248-252). IEEE.
- Ramírez Atencia, C., & Camacho, D. Planificación automática de misiones para drones mediante computación evolutiva.
- RaspAP. (2020). *RaspAP*. Recuperado de <https://raspap.com/>.
- Raspberry Pi Foundation. *Raspberry Pi 4*. Recuperado de <https://www.raspberrypi.com/>

Resolución 251 de 2015. Comandante Roberto Yerovi de la Calle. Normas de Operación de Drones. 17 de septiembre de 2015.

Registro Oficial N° 439 de 2015. Asamblea Nacional de la República del Ecuador. Tercer Suplemento. Ley Orgánica de Telecomunicaciones. 18 de febrero de 2015.

Robotics, D. (2019). Welcome to DroneKit-Python's documentation!. URL <http://python.dronekit.io>.

Roldán, J. J., Joossen, G., Sanz, D., Del Cerro, J., & Barrientos, A. (2015). Mini-UAV based sensory system for measuring environmental variables in greenhouses. *Sensors*, *15*(2), 3334-3350.

Romero Amondaray, L., Piña Figueredo, F. M., & Goire Castilla, M. M. (2018). Red de sensores inalámbricos para las casas de cultivos protegidos San José. *Ingeniería Electrónica, Automática y Comunicaciones*, *39*(1), 16-26.

Santillán, I. D. G., Herrera, C. D., & Ortega, J. I. M. (2018). Detección automática de líneas de cultivo de papa utilizando imágenes digitales. *SATHIRI*, *12*(2), 46-67.

Safadinho, D., Ramos, J., Ribeiro, R., Filipe, V., Barroso, J., & Pereira, A. (2020). UAV Landing Using Computer Vision Techniques for Human Detection. *Sensors*, *20*(3), 613.

Sarasti, O. A. O., & Ramírez, G. L. (2016). Sistemas de información enfocados en tecnologías de agricultura de precisión y aplicables a la caña de azúcar, una revisión. *Revista Ingenierías Universidad de Medellín*, *15*(28), 103-124.

Sotomayor, J. F., Gómez, A. P., & Cela, A. F. (2014). Sistema de visión artificial para el análisis de imágenes de cultivo basado en texturas orientadas. *Revista Politécnica*, *33*(1).

Sun, P., Boukerche, A., & Tao, Y. (2017, June). Theoretical Analysis of the Area Coverage in a UAV-based Wireless Sensor Network. In *2017 13th International Conference on Distributed Computing in Sensor Systems (DCOSS)* (pp. 117-120). IEEE.

Tarot., Tarot Wenzhou tarot aviation technology. Recuperado de <http://www.tarotrc.com/Product/?Lang=en&CategoryId=f513ce24-ea88-4365-908e-756cd1be9e52>

Vasilakos, A. V., Zhang, Y., & Spyropoulos, T. (Eds.). (2016). Delay tolerant networks: Protocols and applications. CRC press

Yu, G., & Cheng, S. (2011, July). Comparison research on the performance of DTN route protocol and Ad hoc route protocol. In 2011 International Conference on Multimedia Technology (pp. 6273-6277). IEEE.

Anexos