



ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

**DEPARTAMENTO DE CIENCIAS DE LA ENERGÍA Y
MECÁNICA**

CARRERA DE INGENIERÍA EN MECATRÓNICA

**TRABAJO DE TITULACIÓN, PREVIO A LA OBTENCIÓN DEL
TÍTULO DE INGENIERA/O EN MECATRÓNICA**

**TEMA: INVESTIGACIÓN E IMPLEMENTACIÓN DE UN
SISTEMA DE SEGURIDAD FIJO Y MÓVIL MEDIANTE UN DRON,
USANDO VISIÓN ARTIFICIAL PARA DETECCIÓN Y
SEGUIMIENTO DE PERSONAS EN UN AMBIENTE EXTERNO
ESPECÍFICO, DE LA UNIVERSIDAD DE LAS FUERZAS
ARMADAS ESPE-L.**

**AUTORES: LILIA MARISOL MORALES CACPATA
CARLOS MANUEL PAUCAR TIPÁN**

DIRECTOR: ING. DARÍO MENDOZA

LATACUNGA

2017



**DEPARTAMENTO DE CIENCIAS DE LA ENERGÍA Y
MECÁNICA**

CARRERA DE INGENIERÍA EN MECATRÓNICA

CERTIFICACIÓN

Certificó que el trabajo de titulación, “**INVESTIGACIÓN E IMPLEMENTACIÓN DE UN SISTEMA DE SEGURIDAD FIJO Y MÓVIL MEDIANTE UN DRON, USANDO VISIÓN ARTIFICIAL PARA DETECCIÓN Y SEGUIMIENTO DE PERSONAS EN UN AMBIENTE EXTERNO ESPECÍFICO, DE LA UNIVERSIDAD DE LAS FUERZAS ARMADAS ESPE-L**” realizado por la señorita **Lilia Marisol Morales Cacpata** y el señor **Carlos Manuel Paucar Tipán**, ha sido revisado en su totalidad y analizado por el software anti-plagio, el mismo que cumple con los requisitos teóricos, científicos, técnicos, metodológicos y legales establecidos por la Universidad de la Fuerzas Armadas ESPE, por lo tanto me permito acreditarlo y autorizar a la señorita **Lilia Marisol Morales Cacpata** y el señor **Carlos Manuel Paucar Tipán** para que lo sustente públicamente.

Latacunga, 15 de Junio del 2017

Atentamente;

Ing. Darío José Mendoza Chipantasi

DIRECTOR



**DEPARTAMENTO DE CIENCIAS DE LA ENERGÍA Y
MECÁNICA**

CARRERA DE INGENIERÍA EN MECATRÓNICA

AUTORÍA DE RESPONSABILIDAD

Nosotros, **Lilia Marisol Morales Cacpata**, con cédula de ciudadanía N° 0503166746 y **Carlos Manuel Paucar Tipán**, con cédula de ciudadanía N° 1718827858, declaramos que este trabajo de titulación **“INVESTIGACIÓN E IMPLEMENTACIÓN DE UN SISTEMA DE SEGURIDAD FIJO Y MÓVIL MEDIANTE UN DRON, USANDO VISIÓN ARTIFICIAL PARA DETECCIÓN Y SEGUIMIENTO DE PERSONAS EN UN AMBIENTE EXTERNO ESPECÍFICO, DE LA UNIVERSIDAD DE LAS FUERZAS ARMADAS ESPE-L”** ha sido desarrollado considerando los métodos de investigación existentes, así como también se ha respetado los derechos intelectuales de terceros considerándose en las citas bibliográficas.

Consecuentemente declaramos que este trabajo es de nuestra autoría, en virtud de ello nos declaramos responsables del contenido, veracidad y alcance de la investigación mencionada.

Latacunga, 15 de Junio del 2017

Lilia Marisol Morales Cacpata

C.C.: 0503166746

Carlos Manuel Paucar Tipán

C.C.: 1718827858



**DEPARTAMENTO DE CIENCIAS DE LA ENERGÍA Y
MECÁNICA**

CARRERA DE INGENIERÍA EN MECATRÓNICA

AUTORIZACIÓN

Nosotros, **Lilia Marisol Morales Cacpata y Carlos Manuel Paucar Tipán**, autorizamos a la Universidad de las Fuerzas Armadas ESPE publicar en la Biblioteca Virtual de la Institución el presente trabajo de titulación **“INVESTIGACIÓN E IMPLEMENTACIÓN DE UN SISTEMA DE SEGURIDAD FIJO Y MÓVIL MEDIANTE UN DRON, USANDO VISIÓN ARTIFICIAL PARA DETECCIÓN Y SEGUIMIENTO DE PERSONAS EN UN AMBIENTE EXTERNO ESPECÍFICO, DE LA UNIVERSIDAD DE LAS FUERZAS ARMADAS ESPE-L”** cuyo contenido, ideas y criterios son de nuestra autoría y responsabilidad.

Latacunga, 15 de Junio del 2017

Lilia Marisol Morales Cacpata

C.C.: 0503166746

Carlos Manuel Paucar Tipán

C.C.: 1718827858

DEDICATORIA

Este proyecto está dedicado a los pilares fundamentales de mi vida, mis padres, Fanny Cacpata y Gustavo Morales, sin ellos, y gracias a sus consejos y educación, jamás hubiese podido conseguir lo que hasta ahora.

También dedico este proyecto a mis hermanos, Diana, Kerly y Gustavito, mismos que han sido los cómplices de momentos de alegría, apoyo y compañía. A ellos este proyecto que, sin ellos, no hubiese podido ser y es por ellos que soy lo que soy ahora. Los amo con mi vida

Lilia Marisol Morales Cacpata

DEDICATORIA

“Sólo dos legados duraderos dejan los padres a sus hijos: uno, es la vida y el otro es la educación”

El presente proyecto dedico muy orgullosamente a mis Padres Ana Tipán y Homero Paucar, que cada uno deposito su confianza, su amor y su apoyo incondicional, quienes han sido un ejemplo de fortaleza y perseverancia quienes de una u otra forma han sabido corregirme día a día, y darme enseñanzas duraderas que siempre las llevare en mi alma y mi corazón.

A mi belleza quien vino a poner mi mundo de cabeza Anita Gabriela y que más que mi hermana eres mi confidente, a ti mi hijo Javier Alejandro que hiciste que ponga los pies sobre la tierra y que eres un incentivo para forjar un nuevo futuro para ti.

A ti mi amor Katty, que llegaste a completar y ordenar mi mundo, haciendo que vea las cosas con más calma, siendo así otra integrante más de mi familia.

Carlos Paucar

AGRADECIMIENTO

En primer lugar, a Dios por haberme guiado por el camino hacia mis sueños; en segundo lugar, a cada uno de los que son parte de mi familia a mi *madre*, mi *padre*, mis hermanas y hermano; por siempre haberme dado su fuerza y apoyo incondicional que me han ayudado y llevado hasta donde estoy ahora.

también mis más sinceros agradecimientos a mis *abuelos* y *tíos* este proyecto es el resultado del esfuerzo conjunto de todos los que formamos el grupo de trabajo. Por esto agradezco a mi compañero y amigo Carlos Paucar quien, y Nuestros sinceros agradecimientos están dirigidos hacia Ing. Diego Maldonado de la Escuela Politécnica Nacional, quien, con su ayuda desinteresada, nos brindó consejos e información relevante, para la iniciación del presente proyecto. Y finalmente agradezco a esta prestigiosa universidad la cual abrió sus puertas a jóvenes como nosotros, preparándonos como profesionales de calidad.

Lilia Marisol Morales Cacpata

AGRADECIMIENTO

Primeramente, agradezco a Dios por haber cuidado en todo este trayecto estudiantil, a mi madre Anita, a mi padre Homero, a mi hermana Gaby, y a mi hijo Alejo, que desde el comienzo fueron quienes han estado dándome su confianza, su apoyo y ayuda, sin ellos nada de esto hubiese sido posible, a mi familia por siempre estar pendiente de mí.

Para continuar quiero agradecer a nuestro tutor Darío Mendoza, que con su reducido tiempo, siempre saco espacio para brindarnos su ayuda, que nos presionó cuando debía hacerlo y nos dio ánimos cuando parecía todo perdido.

También agradezco a una personita muy especial que de una u otra forma siempre está a mi lado dándome su apoyo incondicional, que ha soportado mis buenos y malos momentos e incluso situaciones difíciles, muchas gracias Katty.

Antes de finalizar debo dar gracias a mi compañera de tesis, mi mejor amiga Marisol, que sin ella no existiera esta tesis, que entre perezas, sustos y bromas logramos culminarla.

Carlos Paucar

ÍNDICE DE CONTENIDOS

| | |
|---|-------------|
| CARÁTULA | i |
| CERTIFICACIÓN | ii |
| AUTORÍA DE RESPONSABILIDAD | iii |
| AUTORIZACIÓN | iv |
| DEDICATORIA | v |
| AGRADECIMIENTO | vii |
| ÍNDICE DE CONTENIDOS | ix |
| ÍNDICE DE TABLAS | xiv |
| ÍNDICE DE FIGURAS | xv |
| NOMENCLATURA | xvii |
| RESUMEN | xix |
| ABSTRACT | xx |
| | |
| CAPÍTULO I | 1 |
| 1. MARCO METODOLÓGICO DE LA INVESTIGACIÓN | 1 |
| 1.1. Introducción | 1 |
| 1.2. Antecedentes..... | 1 |
| 1.3. Justificación | 2 |
| 1.4. Planteamiento del problema | 2 |
| 1.5. Hipótesis..... | 3 |
| 1.6. Estado del arte sistemas de seguridad autónomos con dron | 4 |
| 1.6.1. Ventajas..... | 4 |
| 1.6.2. Desventajas | 5 |
| 1.7. Investigaciones realizadas | 6 |
| 1.7.1. Vigilancia con drones para evitar robos | 6 |
| 1.7.2. Guardia de seguridad volador | 6 |

| | |
|--|-----------|
| 1.7.3. Rastreo y seguimiento de objetos basado en visión sin GPS por vehículos aéreos no tripulados..... | 7 |
| 1.7.4. Visión para la detección y el seguimiento de objetos mediante un vehículo aéreo multirrotor no tripulado..... | 8 |
| 1.7.5. Sistema aéreo no tripulado basado en la visión contra cazadores furtivos de la fauna silvestre | 9 |
| 1.8. Análisis de investigaciones de sistemas seguridad mediante dron actuales y futuros. | 10 |
| CAPÍTULO II | 12 |
| 2. DESCRIPCIÓN DE COMPONENTES | 12 |
| 2.1. Conceptos generales | 12 |
| 2.1.1. Sistema de seguridad | 12 |
| 2.2. Vehículo Aéreo no Tripulado..... | 12 |
| 2.2.1. Aplicaciones de los UAV | 13 |
| 2.2.2. Ventajas de UAV | 13 |
| 2.2.3. Desventajas técnicas del vehículo aéreo no tripulado | 13 |
| 2.3. Cuadricóptero | 14 |
| 2.4. Hardware | 15 |
| 2.5. El cuadricóptero: Parrot Bebop 2 | 16 |
| 2.6. Sistema operativo | 17 |
| 2.7. Ubuntu Linux..... | 18 |
| 2.8. ROS (Robot Operating System)..... | 19 |
| 2.8.1. Conceptos de ROS | 21 |
| 2.8.2. Desarrollo del Espacio de trabajo (Workspace) | 22 |
| 2.8.3. Beneficios de ROS..... | 22 |
| 2.8.4. Paquete Bebop Drone autonomy | 23 |

| | |
|--|-----------|
| 2.8.5. OpenCv | 23 |
| CAPÍTULO III | 26 |
| 3. DESARROLLO DEL ALGORITMO PARA PROCESAR LA IMAGEN | 26 |
| 3.1. Visual servoing | 26 |
| 3.1.1. Control servo visual basado en imagen (IBVS) | 28 |
| 3.2. Adquisición de la imagen | 29 |
| 3.2.1. Captura de imágenes desde una cámara | 29 |
| 3.2.2. Conversión de imágenes de ROS a imágenes de OpenCv | 29 |
| 3.3. Reconocimiento de objetos en imagen | 30 |
| 3.3.1. Reconocimiento de objetos usando OpenCv | 30 |
| 3.4. Descriptores HOG (Histogramas de gradientes orientados)..... | 31 |
| 3.4.1. Descripción básica de HOG | 34 |
| 3.4.2. Descripción matemática de HOG | 36 |
| 3.5. Máquina de vectores soporte SVM (Suport Vector Machine) | 39 |
| 3.5.1. Descripción básica de SVM | 40 |
| CAPÍTULO IV | 42 |
| 4. IMPLEMENTACIÓN DE SEGUIMIENTO | 42 |
| 4.1. ROS..... | 42 |
| 4.2. Paquete bebop_autonomy | 42 |
| 4.3. Configuración del paquete y librerías | 43 |
| 4.4. Configuración de los archivos “launch” | 45 |
| 4.5. Ejecutando el sistema en ROS | 46 |
| 4.6. Órdenes del usuario por teclado | 50 |
| 4.7. Estimación de pose..... | 51 |
| 4.7.1. Determinación del error de Pose..... | 52 |

| | |
|---|------------|
| 4.7.2. Conversión error de posición a velocidad | 54 |
| 4.8. Modelado del cuadricóptero Bebop 2..... | 58 |
| 4.9. Controlador PID | 64 |
| 4.10. Controlador PID para cada eje..... | 66 |
| CAPITULO V | 68 |
| 5. PRUEBAS Y RESULTADOS | 68 |
| 5.1. Área de detección útil de la cámara estática y cámara del cuadricóptero .. | 68 |
| 5.2. Evaluación del Algoritmo de detección usado para la cámara estática | 71 |
| 5.3. Evaluación del Algoritmo de detección dentro de las distancias delimitadas, para el seguimiento del cuadricóptero. | 79 |
| 5.3.1. Comprobación de hipótesis de detección..... | 81 |
| 5.4. Pruebas de seguimiento | 83 |
| 5.4.1. Comprobación de hipótesis de seguimiento..... | 85 |
| 5.5. Pruebas de velocidad de la persona en seguimiento. | 87 |
| 5.5.1. Velocidad promedio de una persona por tramos. 2..... | 88 |
| 5.6. Pruebas del Sistema de Seguridad..... | 90 |
| 5.7. Comprobación de la hipótesis del sistema..... | 92 |
| CAPITULO VI..... | 95 |
| 6. CONCLUSIONES Y RECOMENDACIONES..... | 95 |
| 6.1. Conclusiones | 95 |
| 6.2. Recomendaciones | 98 |
| REFERENCIAS BIBLIOGRÁFICAS..... | 100 |
| ANEXOS | 105 |
| Anexo A Características técnicas Parrot Bebop 2 | |

Anexo B Requerimiento para la instalación

Anexo C Manual de instalación

Anexo D Manual de usuario

CERTIFICACIÓN.....1

ÍNDICE DE TABLAS

| | |
|---|----|
| Tabla 1. Parámetros de selección del cuadricóptero..... | 16 |
| Tabla 2. Selección de Sistemas Operativos..... | 18 |
| Tabla 3. Órdenes por teclado para control del Bebop 2 | 50 |
| Tabla 4. Funciones de transferencia de cada eje..... | 62 |
| Tabla 5. Valores para el cálculo de la inercia en los diferentes ejes | 64 |
| Tabla 6 Ganancias del control PID | 67 |
| Tabla 7. Área de detección en base a la profundidad del sujeto | 70 |
| Tabla 8. Escalonamiento con respecto al área | 71 |
| Tabla 9. Detección de personas según la profundidad | 72 |
| Tabla 10. Detección de personas por algoritmo..... | 74 |
| Tabla 11. Delimitación de distancias para una buena detección | 76 |
| Tabla 12. Evaluación del algoritmo detector | 79 |
| Tabla 13. Datos para calcular chi-cuadrado..... | 82 |
| Tabla 14. Calculo de frecuencia teórica | 82 |
| Tabla 15 Validación del seguimiento de 0-40m..... | 84 |
| Tabla 16. Datos para calcular chi-cuadrado..... | 86 |
| Tabla 17. Velocidad de seguimiento de una persona en una longitud de 40m. | 87 |
| Tabla 18. Tiempo y velocidad por tramos de seguimiento | 89 |
| Tabla 19. Pruebas del sistema de seguridad (Detección y seguimiento) | 90 |
| Tabla 20. Datos chi-cuadrado hipótesis | 92 |

ÍNDICE DE FIGURAS

| | |
|---|----|
| Figura 1. Dron Security Guard realizado por Autonomy | 7 |
| Figura 2. Imágenes obtenidas por el AR Dron 2.0. | 8 |
| Figura 3. UAV with NavStik and Odroid | 8 |
| Figura 4. Detección y seguimiento de un rinoceronte | 9 |
| Figura 5. Reacción de torques en cada motor de un Cuadricóptero | 14 |
| Figura 6. Compensación de reacciones de torques de un Cuadricóptero | 14 |
| Figura 7. Parrot Bebop 2..... | 17 |
| Figura 8. Logotipo Ubuntu GNU..... | 19 |
| Figura 9. Terminal de Ubuntu. | 19 |
| Figura 10. Logotipo ROS (Robotic operating System) | 20 |
| Figura 11. Estructura de comunicación ROS | 20 |
| Figura 12. Sistema de control del VSC | 27 |
| Figura 13. Proceso de conversión de imagen ROS a imagen OpenCv | 30 |
| Figura 14. Reconocimiento de Objetos mediante OpenCv..... | 31 |
| Figura 15. Obtención de imagen y aplicación del descriptor HOG | 32 |
| Figura 16. Imágenes, histogramas y gradientes del descriptor HOG | 32 |
| Figura 17. Bloques para normalización de una imagen | 33 |
| Figura 18. Solapamiento de bloques e Histogramas de cada celda por bloque | 35 |
| Figura 19. Proceso de extracción, cálculo de gradientes y almacenamiento de características | 36 |
| Figura 20. Hiperplano obtenido dentro del espacio de características | 39 |
| Figura 21. Clasificación de muestras Persona y no persona..... | 41 |
| Figura 22. Comunicación Operacional ROS | 42 |
| Figura 23. Bebop_autonomy..... | 43 |
| Figura 24. CMakeList.txt..... | 44 |
| Figura 25. Package.xml | 45 |
| Figura 26. Ejemplo de configuración del archivo launch (PID) | 46 |
| Figura 27. Nodos ejecutados con archivo launch..... | 46 |
| Figura 28. Suscripción y publicación de tópicos..... | 47 |

| | |
|--|----|
| Figura 29. Comunicación entre nodos mediante tópicos..... | 48 |
| Figura 30. Pantalla de interfaz principal | 48 |
| Figura 31. Campo visual de cámara Bebop2 | 51 |
| Figura 32. Tamaño de la imagen 640x480p..... | 52 |
| Figura 33. Posición de la persona con respecto al tiempo | 52 |
| Figura 34. Punto centro y error de posición | 53 |
| Figura 35. Punto de origen en el centro del campo de visión..... | 54 |
| Figura 36. Relación error de posición – velocidad | 55 |
| Figura 37. Grafica error de posición Y (roll) vs. velocidad | 55 |
| Figura 38. Grafica área (profundidad) vs. velocidad X | 57 |
| Figura 39. Marcos de referencia para el modelado | 58 |
| Figura 40. Rotación de roll con respecto a X | 59 |
| Figura 41. Rotación de pitch con respecto a Y..... | 59 |
| Figura 42. Rotación de yaw con respecto a Z..... | 60 |
| Figura 43. Longitudes del motor | 63 |
| Figura 44. Diagrama de bloques de funcionamiento de control PID..... | 66 |
| Figura 45. Distancias del centroide a X, Y | 67 |
| Figura 46. Área de detección de la cámara estática | 68 |
| Figura 47. Área de detección útil de cuadricóptero | 69 |
| Figura 48. Campo visual de detección | 70 |
| Figura 49. Muestra de imágenes cámara estática | 72 |
| Figura 50. Detección vs distancia cámara estática | 74 |
| Figura 51. Evaluación del algoritmo de detección | 76 |
| Figura 52. Muestra de imágenes cámara cuadricóptero | 76 |
| Figura 53. Delimitación de distancias en profundidad | 78 |
| Figura 54. Evaluación del Algoritmo con la cámara del Dron en vuelo..... | 81 |
| Figura 55. Tabla chi-cuadrado 2 GDL error 5% | 83 |
| Figura 56. Medición de intervalos de distancia | 84 |
| Figura 57. Tabla Chi- cuadrado 7 GDL error 5%..... | 86 |
| Figura 58. Pruebas de seguridad..... | 92 |
| Figura 59. Tabla Chi- cuadrado 2 GDL error 5%..... | 93 |

NOMENCLATURA

d_{cg} : Distancia del motor al centro de gravedad del cuadricóptero

$e(t)$: Error= valor medido – valor requerido

m_m : Masa del motor

$u(t)$: Respuesta del controlador ante el error

$A_{m\acute{a}x}$: Área máxima de detección $60000px^2$

A_{min} : Área máxima de detección $10000px^2$

A_{refx} : Área de referencia para la distancia de la persona

BSD: Berkeley software distribution

GNU: Acrónimo recursivo que significa GNU No es Unix (GNU is Not Unix)

ITS: Intelligent traffic systems / Sistemas inteligentes de trafico

I_x : Longitud del rotor en X

I_y : Longitud del rotor en Y

I_z : Longitud del rotor en Z

I_{xx} : Inercia en el eje x

I_{yy} : Inercia en el eje y

I_{zz} : Inercia en el eje z

K_p : Ganancia proporcional del controlador PID

K_i : Ganancia integral del controlador PID

K_d : Ganancia derivativa del controlador PID

OpenCV: Open computer vision

P: Punto centro del sujeto.

P_c : Punto centro del plano de imagen

P_e: Punto para el cálculo del error.

ROS: Robot operating system / Sistema operativo robótico

SO: Sistema operativo / Operating system

UAV: Unmanned aerial vehicle / Vehículo aéreo no tripulado.

RESUMEN

El presente proyecto tiene la finalidad de utilizar visión artificial, para controlar el vuelo de un cuadricóptero logrando que este se vuelva autónomo. Esto se realiza por medio del uso software libre y un sistema operativo dedicado explícitamente al control y comunicación de robots. Por medio de visión artificial se conseguirá la detección de personas que es el objetivo principal de este proyecto. Todo esto se lo realiza por medio de una cámara estática, que adquirirá las imágenes para ser procesadas, y así enviar la orden de despegue y persecución al cuadricóptero. El uso del cuadricóptero es esencial ya que es el encargado de realizar el seguimiento a las personas que han sido detectadas por la cámara estática, realizándose el control por medio de un software libre llamado ROS. Este entre sus características tiene librerías que facilitan la comunicación con diferentes plataformas robóticas. Al tener estas características se requiere: crear un *workspace* o espacio de trabajo donde los paquetes a ser usados se almacenan y a su vez estos puedan ser creados, modificados o instalados, realizar una comunicación entre nodos ya que de esa manera estos comparten información para realizar una ejecución compleja, suscribirse a temas (tópicos) es decir identificarse ante un nodo para que este le envíe datos o señales de control y publicar tópicos es decir un nodo declara que va a publicar un tema. Las actividades mencionadas son necesarias para realizar el control y cumplir con el objetivo, logrando un vuelo autónomo del cuadricóptero, mientras este persigue a una persona; obteniéndose como resultados una comunicación y un control óptimo del mismo, teniendo presente que al ser un prototipo se puede realizar más investigaciones.

PALABRAS CLAVE

- **VISIÓN ARTIFICIAL**
- **SISTEMA OPERATIVO ROBÓTICO**
- **VEHÍCULO AÉREO NO TRIPULADO**
- **HISTOGRAMA DE GRADIENTES ORIENTADOS**
- **MÁQUINA DE VECTORES SOPORTE**

ABSTRACT

The present project's purpose is to use, artificial vision to control a flight quadcopter's, making it autonomous. This is done by means of free software and an operating system dedicated to the control and communication of robots. Through artificial vision, will be accomplished the detection of people, which is the main objective of this project, this will be done by means of a static camera, that will acquire to the images to be processed, and thus send the order of takeoff and persecution to the quadcopter; The use of the quadcopter is essential since it tracks people who have been detected by the static camera, performing the control by means of a free software called ROS. Among its features, this software, has libraries that facilitate communication with different robotic platforms. Having these characteristics, it is required to create a workspace where the packages to be used are stored and so that they can be created, modified, or installed, perform a communication between nodes for share information to execute a complex application, subscribe to topics that is to identify themselves before a node so that this sent data or control signals and publish topics so a node declares that it will publish a topic. All these mentioned activities are necessary to be able to perform the control, and accomplish the objective of performing an autonomous flight of a quadcopter while it follows a person, obtaining as results an optimal communication and control, keeping in mind that since it is a prototype, more research can be done with it.

KEYWORDS:

- **ARTIFICIAL VISION**
- **ROBOTIC OPERATING SYSTEM**
- **UNMANNED AERIAL VEHICLE**
- **HISTOGRAM OF ORIENTED GRADIENTS**
- **SUPORT VECTOR MACHINE**

CAPÍTULO I

1. MARCO METODOLÓGICO DE LA INVESTIGACIÓN

1.1. Introducción

En vista de la creciente evolución de la tecnología, al estar relacionada con dos ramas muy importantes como son la visión artificial y la robótica, por lo que ha despertado el interés de investigadores para desarrollar aplicaciones que permitan la sinergia de estas ramas. Para este proyecto se trabajará en un sistema de seguridad que detecte personas y realice el seguimiento con un vehículo aéreo no tripulado; en este caso uno de las características más relevantes es el uso de visión artificial, la cual estará encargada de la detección de personas con ayuda de las librerías OpenCv (Open Computer Vision), ROS (Robot Operating System), como interfaz de comunicación y procesamiento de control para el vehículo aéreo no tripulado.

1.2. Antecedentes

En la actualidad el uso de visión artificial para la seguridad de lugares, ya sean públicos o privados, se están afianzando como una tecnología imprescindible en los sistemas de vigilancia gracias a la posibilidad de obtener la máxima información del objeto por medio de la adquisición y procesamiento de imágenes, recibiendo datos como la trayectoria seguida, velocidad, coordenadas de su última posición, entre otros. El caso más común del uso de estos sistemas es el control de tráfico (ITS), sistemas de seguridad en entornos militares, espacios públicos; todo esto tiene como finalidad brindar seguridad a las personas que habitan estos lugares. (INFAIMON S.L, 2016).

Actualmente existen muchas variantes de softwares para el desarrollo de este tipo de aplicaciones, una de estas es ROS que contiene características de código abierto que proporcionan varios servicios e incluso presenta librerías que

potencian considerablemente la creación de aplicaciones para visión artificial. También presenta entre sus características la comunicación predefinida con varias plataformas ya sean móviles o fijas. (Gallardo H., 2011)

1.3. Justificación

El presente proyecto nace debido al aumento de la inseguridad en la mayor parte de nuestro país (CEDATOS, 2016); con lo que se busca minimizar la inseguridad en diferentes entornos, al proporcionar una herramienta tecnológica fácilmente reproducible al usar software libre.

El proyecto contempla un concepto del uso de visión artificial enfocado a la seguridad y defensa, con un especial énfasis en la investigación de detección y seguimiento de personas, que interrumpen un espacio delimitado, usando hardware y software que trabajando en conjunto prestarán un servicio.

Un cuadricóptero es una plataforma UAV, que tiene una gran ventaja para la creación de aplicaciones de seguridad y su versatilidad al momento de ser controlados por sistemas remotos. Con la vinculación de una cámara de seguridad externa hace que este sistema sea propicio para trabajar en entornos

La unión de herramientas de software y hardware permitirá que el proyecto actual se use para la vigilancia de áreas restringidas, detectando la presencia de un intruso humano, mediante el flujo de imágenes otorgada por una cámara fija, para dar paso al seguimiento de la persona mediante un dron.

1.4. Planteamiento del problema

Desde siempre el tema de vigilancia ha sido un tema primordial, para resguardar la seguridad de personas y proteger bienes ante amenazas externas, ya sea de instalaciones sean públicas o privadas,.

A medida que pasan los años la tecnología avanza y permite tener sistemas más sofisticados, innovadores y precisos, referentes al tema de incrementar el

nivel de seguridad en diferentes áreas como reservas ecológicas, áreas verdes, y lugares con poca vigilancia. En vista de este precedente, se da una alternativa para elevar la seguridad en áreas específicas con la detección de personas en un entorno y su posterior seguimiento mediante un cuadricóptero, que recopila imágenes desde su cámara incorporada.

Tal es el caso de EP Petroecuador que desea incorporar la tecnología dron para seguridad, con un sistema basado en visión artificial, que deberá incluir procesamiento de imágenes obtenidas gracias al uso de cámaras convencionales de video. Cuyo objetivo es contar con la vigilancia y el control de riesgos para reforzar a la seguridad e integridad de poliductos en el Ecuador, para identificar personas, animales u otros objetos en los sitios monitoreados, (Freire J., 2015).

Países vecinos como Venezuela, están apoyándose también en sistemas de seguridad mediante visión artificial usando drones que permiten ampliar la capacidad de respuesta operacional para robustecer la seguridad en diferentes zonas de este país. (GLOBOVISION, 2016).

Planteados los problemas actuales, relacionadas a la vigilancia de grandes superficies, control perimetral, vigilancia de infraestructuras, entre otros, surgen soluciones tecnológicas que abren nuevas ramas en el campo científico y de investigación, con el uso de herramientas como son: visión artificial y dron, siendo el campo de este último uno de los segmentos con mayor crecimiento en la industria aeronáutica a nivel mundial y al combinarlo con la primera permitirá la adquisición de video en tiempo real y al acoplar con la detección de personas permitirá plantear una alternativa de solución para vigilancia de áreas.

1.5. Hipótesis

¿Se mejorará la seguridad de un área delimitada al implementar un sistema de visión artificial que usa en conjunto una cámara fija y otra móvil, para detección y seguimiento de personas?

1.6. Estado del arte sistemas de seguridad autónomos con dron

A lo largo de los últimos años, el uso sistemas de seguridad autónomos, especialmente los basados en el uso de dron para la vigilancia y seguridad, es un tema que ha ido tomando mucha relevancia. Esta tecnología puede ser muy provechosa sobre todo en situaciones de extremo peligro, investigaciones policiales o misiones del ejército. (Zahumenszky, 2013)

Dentro de las aplicaciones otorgadas a los drones, una que destacará a futuro es la de seguridad y vigilancia perimetral, ya que los vehículos aéreos no tripuladas brindan un punto de vista único con bajo costo en sus operaciones. (Torres, 2017)

Actualmente el interés en este tema ha crecido dando realce a la investigación, en varias universidades y organizaciones, ya que sin lugar a duda esta tecnología beneficiará en varios aspectos y no únicamente en seguridad.

1.6.1. Ventajas

La aplicación de la tecnología dron genera ventajas en ámbitos civiles, militares, industriales y comerciales, como los que a continuación, se indican.

- La vigilancia de zonas rurales mejora y disminuye los robos en el campo.
- Reducen los índices de delincuencia y logran un monitoreo en tiempo real de las diferentes situaciones que se registran en la ciudad a través de la captura de imágenes y la visualización de las mismas en tiempo real, (Redacción Bogotá, 2016).
- Compañías de seguridad aprovechan las múltiples virtudes de estas pequeñas aeronaves para ofrecer unos niveles de seguridad más elevados en espacios abiertos.
- Empresas de cartografía acuden con frecuencia a este tipo de sistemas para pulir y ajustar los mapas que se desarrollan.

- Algunos cuerpos del ejército de los EE.UU., entre otros, ya hacen uso de drones para planificar y realizar ataques a los enemigos de una forma más silenciosa y eficiente, (Rivera, 2016).
- Ayudan en la repartición de envíos de una forma más eficiente y veloz, como es el caso de hoy en día de Amazon.
- Pueden ser capaces de trabajar en ambientes de calor para realizar tareas de vigilancia aérea más específicas, por ejemplo: vigilancia del cráter de un volcán.
- No ponen en peligro la integridad física de las personas que los monitorean o se encuentren cercanas.

1.6.2. Desventajas

- La tecnología dron aún no está establecida como un recurso estandarizado, es decir que su uso, puede llegar a ser ilegal en ciertas zonas, dependiendo de las normativas aéreas de cada país.
- La legislación para estas aeronaves no tripuladas, es inexistente, inválida o parcialmente indefinida en algunos países del mundo, dejando camino libre al espionaje tanto industrial como de personas. (Rivera, 2016).
- Por el momento, no se permiten a los operadores de drones efectuar vuelos nocturnos, tampoco vuelos en las proximidades de aeropuertos. Además, restringe los vuelos sobre edificios y exige supervisar continuamente los vuelos ejecutados de forma autónoma.
- Es necesario homologar el sistema de vigilancia que se vaya a utilizar, para apegarse a las reglas que se rigen en la legislación española actual, así como a las normas que rigen en Estados Unidos, ya que ambas son similares.

1.7. Investigaciones realizadas

1.7.1. Vigilancia con drones para evitar robos

El Ayuntamiento de Algemesí (Valencia-España) y la Universidad Politécnica de Valencia realiza un proyecto de vigilancia sobre los campos de cultivo mediante aeronaves por control remoto desde el 2016. La propuesta incorpora un dron controlado desde un coche-patrulla, lo que facilitaría a la policía un radio de alcance de vigilancia de un kilómetro, para evitar el robo de cultivos de naranja en épocas de cosecha. (Montero, 2016)

1.7.2. Guardia de seguridad volador

Dentro del ámbito comercial, la empresa americana Autonomy ha desarrollado un dron para seguridad, mismo que es un octocóptero con características similares al DJI S-1000+, que cuenta con cámara y se usa comúnmente por productores de cine para grabar escenas desde el aire. (García, 2016)

El dron tiene un diámetro de más de un metro, constituido de sensores personalizados, 8 hélices, así como un sistema de control. Autonomy añade luces azules y rojas, y un foco blanco y dos altavoces para hacer frente a los intrusos, como se observa en la figura 1.

El vehículo aéreo no tripulado realiza capturas de video usando cámaras comunes. Un avión no tripulado Autonomy estará programado para patrullar un área de forma automática, y utilizar sus cámaras a bordo para detectar y abordar cualquier persona que entra en un área sin autorización.



Figura 1. Dron Security Guard realizado por Autonomy

Fuente: (Zone, 2017)

El avión no tripulado destellaría sus luces de advertencia, enciende una luz central dirigida a la persona intrusa y entrega una advertencia. Un guardia de seguridad en un centro de control sería notificado, y el guardia podría tomar el control del avión no tripulado y hablar a través de él. (Simonite, 2016)

1.7.3. Rastreo y seguimiento de objetos basado en visión sin GPS por vehículos aéreos no tripulados

Investigación realizada por el grupo Vision4UAV de la universidad Politécnica de Madrid en conjunto con la Universidad Estatal de Arizona, el cual presenta una estrategia de control basada en la visión artificial. Se desarrolló un método de control visual basado en una imagen que utiliza sólo una cámara con visión frontal para el seguimiento a partir de un UAV multi-rotor, que no necesita la iteración con el GPS para realizar el control de vuelo. El método propuesto sigue un objeto especificado por el usuario continuamente, manteniendo una distancia fija del objeto y también simultáneamente manteniéndolo en el centro del plano de imagen.

El algoritmo se validó utilizando un Parrot AR.Drone 2.0 en condiciones al aire libre, el seguimiento de personas u objetos en movimiento rápido; muestra la robustez de los sistemas propuestos contra perturbaciones y cambios de iluminación. La figura 2 muestra características del seguimiento e imagen de la cámara a bordo del UAV. (Pestana J. , Sanchez, Campoy, & Saripalli, 2013)



Figura 2. Imágenes obtenidas por el AR Dron 2.0.

Fuente: (Pestana J. , Sanchez, Campoy, & Saripalli, 2013)

1.7.4. Visión para la detección y el seguimiento de objetos mediante un vehículo aéreo multirotor no tripulado

El sistema de navegación basado en visión para la detección de objetos y el seguimiento mediante un vehículo aéreo multiusos no tripulado (UAV) es presentado por el equipo NavStik and Odroid. El sistema consta de un ordenador de bordo único integrado con una cámara orientada hacia abajo para la detección de objetos y seguimiento acompañado por el piloto automático de NavStik para estabilizar el multirotor, el cual se aprecia en la figura 3. (Kaingade, More, Dhule, Gaidhani, & Gupta, 2015)



Figura 3. UAV with NavStik and Odroid

Fuente: (Pestana J. , Sanchez, Campoy, & Saripalli, 2013)

El procesamiento se caracteriza por desventajas tales como los retrasos en la comunicación y el riesgo de fallo de la comunicación.

En este trabajo, el sistema es capaz de estimar la posición del objeto y enviar puntos de ajuste de posición al piloto automático en tiempo real. La detección de objetos en color y el algoritmo de seguimiento se ejecuta en una sola placa Odroid, donde los puntos de control para multirotor se calculan y envía a NavStik.

NavStik, ejecutando RTOS (sistema operativo en tiempo real), acepta comandos de Odroid y acciona actuadores de multirrotor para navegar y volar. (Kaingade, More, Dhule, Gaidhani, & Gupta, 2015)

1.7.5. Sistema aéreo no tripulado basado en la visión contra cazadores furtivos de la fauna silvestre

Esta investigación está diseñada para prestar apoyo en la vigilancia de grupos de animales, el reconocimiento de cazadores furtivos, así como el seguimiento de sus vehículos. Se desarrolló capacidades para definir misiones de vigilancia completamente autónomas, el despegue autónomo y el seguimiento de una lista de posiciones predefinida, seguimiento de animales, detección de caras de cazadores furtivos, seguimiento vehículos de caza furtiva y retorno, a través de un escenario de aterrizaje autónomo, en estaciones para recargar las baterías y prepararse para el próximo vuelo de vigilancia, en la figura 4 se muestra la detección de un rinoceronte y como es marcado por software. (Olivares, y otros, 2015)



Figura 4. Detección y seguimiento de un rinoceronte

Fuente: (Olivares, y otros, 2015)

En el área que será monitoreada debe ser instalado un conjunto de estaciones estáticas que permitirán realizar el aterrizaje así como la carga. Estas estaciones son necesarias para recuperar UAVs en los casos en que el vehículo de patrulla de seguridad más cercano está fuera del alcance de la resistencia de la batería de un UAV. Una vez que el UAV despegue de una de estas estaciones móviles o estáticas, seguirá patrullando la trayectoria. Esta trayectoria podría

modificarse de forma autónoma en los casos en que se detecte un animal o un grupo de animales. Las cámaras a bordo están transmitiendo las imágenes a un vehículo de patrulla en el que un guardia de seguridad identifica si un animal está herido.

La trayectoria predefinida también puede modificarse de forma autónoma en caso de detectar un grupo de cazadores furtivos, en este caso, el sistema debe detectar sus caras para agregarlos en una base de datos. Esta base de datos debe ser compartida con las autoridades para identificar a estos individuos y hacerlos procesar. Otro caso en el que el sistema UAV debe modificar su trayectoria es cuando se detecta un vehículo, en este caso, el sistema debe poder seguir el vehículo con el fin de ayudar a las autoridades de seguridad a perseguir y atrapar a posibles cazadores furtivos. Una vez que la misión se haya completado o cuando las pilas de un UAV estén a punto de agotarse, este regresa a la estación de aterrizaje o carga ya sea estática o móvil que se encuentre más cercana, para finalmente realizar un aterrizaje de forma autónoma.

Basado en los escenarios de misión previamente establecidos, este documento presenta una serie de algoritmos, algunos de estos utilizan imágenes aéreas que captaran zonas en las que se encuentren animales para realizar su seguimiento discriminándolos de acuerdo a características específicas. Otros algoritmos incluidos utilizan sistemas de control basados en la visión para seguir vehículos o para aterrizar UAV de forma autónoma tanto en movimiento y plataformas estáticas. (Olivares, y otros, 2015)

1.8. Análisis de investigaciones de sistemas seguridad mediante dron actuales y futuros.

Actualmente el uso de drones enfocados para mantener la seguridad en espacios abiertos para zonas rurales o suburbanas ha aumentado. Las investigaciones ya sea dentro del ámbito comercial o académico no se han hecho esperar, encontrando varios proyectos futuristas, que parten desde una plataforma móvil

comercial como es el Parrot AR Dron o desde plataformas diseñadas que se equipan con sensores y placas de control.

A pesar de los beneficios que brinda esta creciente tecnología, ciertos países no le permiten desarrollarse ampliamente, debido a sus legislaciones, a excepción de lugares donde las legislaciones permiten su uso como en áreas de desastres naturales, reservas de flora y fauna, zonas rurales, construcciones privadas e incluso prisiones, para preservar la seguridad e inclusive bajar el índice de robos.

La plataforma móvil Parrot AR Drone 2.0, lanzada al mercado en 2012, dada la versatilidad que ofrece para obtener los datos de velocidad, posición, altura, entre otros, desde sus sensores, así como la calidad de imagen y video que este proporciona y sobre todo un conector Wi-Fi, que permite vincular el dron a dispositivos móviles personales con sistemas operativos como iOS, Android o Linux, ha permitido que varios investigadores elijan y basen su trabajo para detección de personas u objetos, en dicho dron. Sin embargo, Parrot ha mejorado las características de sus drones, tal es el caso del Bebop 2, que cuenta con cámara de 14 megapíxeles, para proporcionar una mejor imagen al momento de realizar las detecciones, tiempo de vuelo más prolongado, además de su robustez y estabilidad en vuelo ante corrientes de aire medianamente fuertes (60km/h).

Las características mejoradas que Parrot Bebop 2 presenta, permitirá una detección y seguimiento de personas u objetos más exacta, dada su resolución de imagen, principalmente para investigaciones de visión, por lo que sería conveniente investigar y trabajar en dicha plataforma. Al acoplar con un sistema operativo de buenas prestaciones se creará un sistema de seguridad que tiende a caracterizarse por su velocidad de procesamiento de imagen, detección de personas en tiempo real y robustez ante perturbaciones de luz y viento.

CAPÍTULO II

2. DESCRIPCIÓN DE COMPONENTES

2.1. Conceptos generales

2.1.1. Sistema de seguridad

Los sistemas de seguridad están determinados como el conjunto de elementos, dispositivos e instalaciones que se encuentran encaminadas a dar resguardo y preservar la integridad tanto de personas como de hogares, escuelas, instalaciones públicas o privadas, ante amenazas externas que pueden ser infiltraciones, robos, atracos. Estos darán una señalización para realizar una acción que ayude a dar una solución con algún mecanismo.

Los sistemas de seguridad tienen variables según la exigencia de la vigilancia, el presupuesto y la calidad del mismo, como es el caso de uso de gendarmería, cercas eléctricas, biometría, detectores, etc.; todos estos con el mismo objetivo el de dar seguridad a la ciudadanía y objetos de alto valor. (CEIL, 2012)

2.2. Vehículo Aéreo no Tripulado

Un vehículo aéreo no tripulado o UAV denominado así por sus siglas en inglés (Unmanned Aerial Vehicle), es un dispositivo que puede ser totalmente autónomo o pilotado, según se requiera. Existe una gran variedad de estos dispositivos al igual que sus aplicaciones, que en su mayoría se han desarrollado para uso militar y de investigación. Sus diferentes aplicaciones ya sean militares o civiles tienen grandes ventajas frente a plataformas aéreas tripuladas ya que al no poseer pilotos humanos son perfectos para realizar operaciones peligrosas o donde se necesita de largos periodos de tiempo, o la vida del piloto afronta alto riesgo; se puede decir que estas plataformas tienen un beneficio significativo al no depender del factor humano. (Sanchez, Mulero, & Saumeth, 2013)

2.2.1. Aplicaciones de los UAV

Son aplicables en ambientes de alta toxicidad química y radiológica, donde se pone en riesgo las vidas humanas, siendo de ayuda incluso en el ámbito civil, en actividades como las que se mencionan a continuación.

- Monitoreo y vigilancia.
- Monitoreo ambiental.
- Búsqueda y rescate.
- Detección de incendios forestales.
- Fotografías aéreas y filmación.
- Cartografía, modelado de superficies terrestres.
- Distribución de semillas.
- Recopilación de información.
- Publicidad y promoción.

2.2.2. Ventajas de UAV

- En la actualidad, la ventaja más contundente de estos vehículos es el uso en zonas de alto riesgo.
- No necesita de una tripulación humana porque su funcionamiento es por control remoto o son completamente autónomos.

2.2.3. Desventajas técnicas del vehículo aéreo no tripulado

- Por ser vehículos en desarrollo aún se tiene deficiencias tanto en la transmisión y recepción de datos, como en la seguridad en los firmwares que pueden ser hackeados y la autonomía de vuelo que viene implícita con el uso de baterías (Maldonado & Calderon, 2014).
- Al ser vehículos que, en cierto modo, son de fácil acceso, no tienen inteligencia propia, no pueden discriminar el futuro de sus acciones, tampoco determinar si su objetivo es el adecuado (Hinojosa & Montaluisa, 2014).

2.3. Cuadricóptero

Es un vehículo aéreo no tripulado que pueden elevarse y desplazarse por medio de cuatro rotores cooplanares, cuyos movimientos dependen de la variación de potencia en cada uno, provocando una diferencia entre el par generado por cada uno de ellos, consiguiendo de esa manera el movimiento deseado (Blasco, García, & Reynoso, 2012). Estos se encuentran instalados en un marco en forma de cruz, la configuración de giro de las hélices de estos es de forma que dos giran en sentido horario mientras los otros dos giran en sentido anti horario como se muestra en la figura 5. (Blasco, García, & Reynoso, 2012)



Figura 5. Reacción de torques en cada motor de un Cuadricóptero

Fuente: (Pestana M. J., Sanchez, Saripalli, & Campoy, 2017)

Las líneas de investigación sobre las que se puede trabajar al utilizar como plataforma este tipo de vehículos es múltiple, clasificada en la literatura generalmente en 3 principales tópicos: control, navegación y guiado. (Bristeau, Callou, Vissiere, & Petit, 2011)



Figura 6. Compensación de reacciones de torques de un Cuadricóptero

Para realizar un vuelo adecuado existen parámetros, como posición, altitud, velocidad, ángulo de ataque, etc. Todos estos se unen para permitir el movimiento adecuado y que no exista ningún daño en la aeronave. Para cada caso hay dispositivos que entregan los datos necesarios y estos a su vez son usados y procesados por programas especializados, (Hinojosa & Montaluisa, 2014). Los dispositivos más relevantes, se detallan a continuación:

- GPS
- Altímetro barométrico
- Magnetómetro
- Sensores de velocidad
- Sensores ópticos de línea de vista
- Radar, sonar

2.4. Hardware

Los dispositivos que se utilizan en este proyecto son una cámara de vigilancia, un cuadricóptero y un computador portátil. El computador portátil es el dispositivo encargado de ejecutar todos los programas y enviar las órdenes de control al cuadricóptero mediante señal Wi-Fi. La cámara de vigilancia se encargará de captar las imágenes para que el algoritmo identifique a la persona y el cuadricóptero recibirá comandos de control que son obtenidos por medio de la ejecución del segundo algoritmo para realizar el seguimiento adecuado de la persona. En la tabla 1 se indican los parámetros selección del cuadricóptero para este proyecto, donde se considera con "0" la referencia base del hardware, + si supera la referencia base y - cuando no supera la referencia base.

Tabla 1.

Parámetros de selección del cuadricóptero

| Criterios de selección | Phantom 2 | Ardrone | Bebop 2 |
|-------------------------------|------------------|----------------|----------------|
| Tiempo de uso | + | - | + |
| Velocidad de vuelo | + | - | + |
| Resolución de cámara | 0 | - | 0 |
| Resistencia al viento | + | - | + |
| Comunicación | + | - | + |
| SDK | - | + | + |
| Robustez estructural | + | 0 | + |
| Tamaño | - | - | + |
| Peso | - | + | + |
| TOTAL POSITIVOS + | 5 | 2 | 8 |
| TOTAL NEGATIVOS - | 3 | 6 | 0 |
| VALORACIÓN FINAL | 1 | -3 | 7 |

2.5. El cuadricóptero: Parrot Bebop 2

Tomando en cuenta la valoración final de la tabla 1, el cuadricóptero Bebop 2 que se observa en la figura 7, se ha elegido por tener mejores características en lo que se refiere a velocidad de vuelo tanto lineal como de ascenso, robustez estructural al ser fabricado con filamentos de fibra de carbono, con respecto a otros cuadricópteros que se han considerado.

Es fabricado por la empresa Parrot en 2015, cuenta con cuatro motores configurados en forma de “X”, con un peso de 500 gramos y una autonomía de 25 minutos. Sus prestaciones le permiten volar en interiores y exteriores, filmar y tomar fotos al mismo tiempo, todo esto gracias a la cámara a bordo de 14 megapíxeles. En el anexo A se aprecian las características completas. (Parrot, 2015)



Figura 7. Parrot Bebop 2

Fuente: (Parrot, 2015)

2.6. Sistema operativo

“Un Sistema Operativo (S.O), es un conjunto de programas de proceso con las rutinas de control necesarias para mantener continuamente operativos dichos programas” (Matinez, 2013).

En el mercado existe una gran variedad de sistemas operativos de venta, siendo algunos de ellos privativos y otros libres.

ROS es un sistema dedicado explícitamente al desarrollo de aplicaciones de robótica, el cual tiene características que ayudan implícitamente a la comunicación y control; todo esto se encuentra basado en la programación y ejecución de nuevos algoritmos . Este sistema operativo es de licencia libre, lo que limita su funcionamiento en sistemas anfitriones, los más usados como anfitriones son LINUX y WINDOWS.

Windows: es el nombre que se da al sistema operativo que basa su funcionalidad por medio de ventanas para ser amigable con el usuario. Este en la actualidad lleva el liderazgo absoluto en el mercado de sistemas operativos para ordenadores, este sistema operativo tiene una licencia privativa ya que es propiedad de la reconocida empresa Microsoft, (Microsoft, 2017).

Linux: es un sistema operativo que se encuentra al alcance de todos, ya que es completamente gratuito, aparte de esto cuenta con GNU que es una parte

esencial ya que cuenta con herramientas básicas indispensables, cualquiera lo puede modificar según sus necesidades. Además, es bastante seguro.

Se puede observar en la tabla 2, las características de cada sistema operativo, que se ha tomado en cuenta para realizar la selección y que pueden ser anfitriones.

Tabla 2.

Selección de Sistemas Operativos

| Criterios de selección | Windows | Linux/Ubuntu |
|-------------------------------|----------------|---------------------|
| Fiabilidad | - | + |
| Rendimiento | - | + |
| Seguridad | - | + |
| Drivers | + | 0 |
| Aplicaciones | + | 0 |
| Costo de licencia | - | + |
| TOTAL POSITIVOS + | 4 | 5 |
| TOTAL NEGATIVOS - | 3 | 1 |
| Valoración final | 1 | 4 |

2.7. Ubuntu Linux

El software usado para el desarrollo de este proyecto es Ubuntu 14.04 LTS por ser una actualización estable y disponible. Ubuntu es un sistema operativo al igual que Windows o Mac Os, pero a diferencia de estos tiene licencia GNU y se encuentra basado en Debian - Linux, lo que conlleva que es de código abierto (Camazón, 2011).



Figura 8. Logotipo Ubuntu GNU.

Fuente: (Camazón, 2011)

Una de las características más relevantes de este sistema operativo es que no necesita grandes recursos para operar de manera correcta, y al ser de licencia libre la mayor parte de empresas industriales lo prefieren. Ubuntu tiene un gestor de paquetes donde se pueden añadir nuevas aplicaciones el cual es llamado Centro de Software, donde se encuentran varias aplicaciones que son compatibles y ya están probadas para este sistema. (Camazón, 2011)

El uso del terminal es de gran ayuda para el desarrollo de este proyecto, ya que permite introducir comandos para: la compilación y ejecución de algoritmos, instalación de paquetes, descarga de programas, etc., debido a que por este medio dichas operaciones resultan mucho más rápido y fácil hacerlas. (Sanchez L. , 2016)

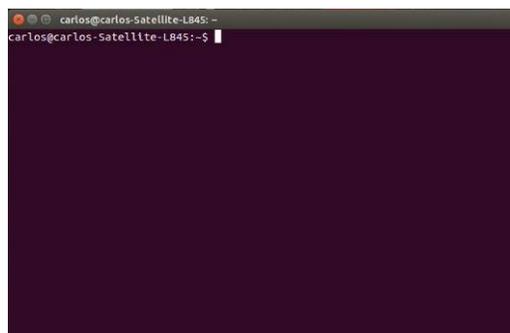


Figura 9. Terminal de Ubuntu.

2.8. ROS (Robot Operating System)

“ROS es un sistema operativo de código abierto para desarrollar aplicaciones para robots o plataformas de investigación” (ROS.org, 2015). Provee servicios

similares a un sistema operativo, incluyendo abstracción de hardware, control de dispositivos de bajo nivel, implementación de la funcionalidad de uso común, mediante el uso de herramientas y bibliotecas para la obtención, construcción, escritura y ejecución de código en varios equipos (Willow Garage, 2015).



Figura 10. Logotipo ROS (Robotic operating System)

Fuente: (Clearpath Robotics Inc, 2016)

Está basado en una arquitectura de grafos donde el procesamiento toma lugar en los nodos que permitirán recibir, enviar y multiplexar mensajes de sensores, control, estados y actuadores, etc. La librería está orientada para un sistema Ubuntu. (Garcia A. , 2013).

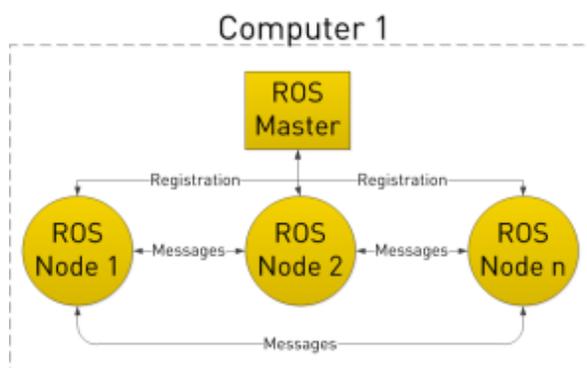


Figura 11. Estructura de comunicación ROS

Fuente: Clearpath Robotics Inc (2016)

ROS inicia al ejecutar ROS Master, mismo que permite que los nodos creados se encuentren y hablen entre sí, sin especificar detalladamente una dirección, simplemente podemos decirle al nodo 1 que envíe mensajes al nodo 2. (Clearpath Robotics Inc, 2016)

El registro de los nodos en el ROS Master permite que estos encuentren exactamente a donde enviar mensajes, y definir si un nodo publica un tema (topic), o si este a su vez está suscrito a alguno.

2.8.1. Conceptos de ROS

Paquetes (*Package*): Son la unidad principal para la organización del software en ROS. Un paquete puede contener procesos de ejecución llamados nodos (1 o más), bibliotecas ROS, bases de datos, archivos de configuración, entre otros. Los paquetes son importantes en la construcción de nuevas aplicaciones al momento de ejecutar ROS. (Chamizo, 2013)

Maestro (*Master*): Permite el registro de nombres para que el resto de los archivos puedan encontrarlos, es decir, sin el Maestro, los nodos no podrían encontrar los mensajes que son enviados de algún otro nodo.

Nodos (*Node*): Es un proceso que realiza algún tipo de computación en el sistema; comparten información entre ellos para realizar una ejecución compleja, teniendo presente que para una aplicación existen varios nodos al momento de ejecutar el sistema de control de una plataforma robótica. Por ejemplo: un nodo controlaría un telémetro láser, otro nodo controla los motores de las ruedas, otro nodo realiza localización, otro nodo realiza planificación de la trayectoria y así sucesivamente. Un nodo de ROS está escrito mediante las bibliotecas de ROS, tales como roscpp (C++) o Rospypy(Python). (Martinez, 2014)

Temas (*Topics*): Son canales de información entre nodos, que contienen los mensajes y los enrutan a través de un sistema de transporte con publicaciones o suscripciones. Estos tienen una identificación para determinar su contenido. Un nodo que está interesado en un determinado tipo de datos va a suscribirse al tema apropiado, es decir habiendo varios publicadores y suscriptores concurrentes para un solo tema, En general, existen publicadores y suscriptores los cuales no interactúan entre ellos. Se puede pensar en un tema como un bus de mensajes inflexible con un determinado nombre, al cual cualquier persona

puede conectarse para enviar o recibir mensajes, percatándose que estos sean los adecuados.

Mensajes: Estructura de datos, que permiten la comunicación entre nodos y contienen tanto el tipo de datos y el mensaje que se envía. Los tipos estándar: enteros, punto flotante, booleanos, etc., así como arreglos, se pueden usar, se pueden incluir estructuras y arreglos C arbitrariamente anidados.

Message (msg) types: Definen la estructura de datos de los mensajes que se enviarán en ROS, se almacenan en `my_package/msg/MyMessageType.msg`.

Paquete manifest: Contiene la información del paquete creado tales como: nombre, versión, descripción, información de la licencia, dependencias, y otra información. Toda esta información se la encuentra en el archivo `package.xml`. (Martinez, 2014).

2.8.2. Desarrollo del Espacio de trabajo (*Workspace*)

Como en todo software, se requiere de un espacio de trabajo y esencialmente para el sistema ROS, en el que debe existir espacio para los paquetes que se usen, sean creados, modificados o instalados. Al mantener todos los paquetes y archivos dentro de un espacio de trabajo, se llevará un orden que permitirá el fácil acceso a ellos para modificarlos y compilarlos mediante Cmake. (Clearpath Robotics, 2015)

2.8.3. Beneficios de ROS

ROS ha sido creado y diseñado para facilitar el intercambio de software entre los aficionados y los profesionales de la robótica en todo el mundo debido a su enfoque didáctico y abierto, lo que ha permitido la construcción de una gran comunidad de colaboradores, así como una plataforma de proyectos de investigación ya probados. (Clearpath Robotics Inc, 2016)

A la hora de desarrollar, ROS admite el uso de diferentes lenguajes de programación de forma oficial soportan Python, C++, Lisp, Lua, etc. ROS puede ejecutarse sobre equipos computacionales de tipo Unix, principalmente Ubuntu y

Windows, aunque por parte de la comunidad se puede encontrar ayuda para su manejo en otras plataformas como Fedora, Gentoo, etc. (Garcia A. , 2013).

ROS cuenta con una comunidad grande de desarrollo, que se compone de investigadores, aficionados, fabricantes de hardware y empresas que le dan soporte como Willow Garage, o incluso Google.

A lo expuesto, sumado a la amplia cantidad de ejemplos, librerías listas para funcionar (con algoritmos de navegación, visión artificial, etc.), y la creciente comunidad ROS es un punto de partida para iniciar el camino en el mundo de la Robótica, (Garcia A. , 2013).

2.8.4. Paquete Bebop Drone autonomy

Bebop autonomy es un controlador ROS creado específicamente para el Parrot Bebop en sus versiones 1.0 y 2.0, por desarrolladores de la Universidad Simon Fraser, basado en Parrot ARDronSDK3. Este controlador contiene el nodo bebop_driver, los mensajes personalizados creados por este nodo y los servidores para los servicios ofrecidos. (Monajjemi Mani., AutonomyLab, Simon Fraser University, 2015)

2.8.5. OpenCv

OpenCv (código abierto visión por ordenador) es liberado bajo una licencia BSD y por lo tanto está libre tanto para uso académico de investigación y comercial. Tiene interfaces de C ++, C, Python y Java y es compatible con Ubuntu Linux. Esencialmente para el diseño de OpenCv se tomaron en cuenta aspectos como la eficiencia computacional con un enfoque para aplicaciones que requieren que el procesamiento se realice en tiempo real, tratando de no perder información relevante. OpenCv es la biblioteca de código más popular y avanzada para aplicaciones relacionadas con la visión por ordenador, (Creative Commons, 2016).

Rasgos de OpenCv

Actualmente la librería OpenCv, abarca muchas tareas muy básicas como: captura y pre-tratamiento de los datos de imagen, además en los algoritmos de alta complejidad en los que se requiere: extracción de características, seguimiento de movimiento, aprendizaje automático, (OpenCV, 2015). Es software libre y provee una API en C, C ++, Java y Python. La biblioteca en sí es independiente de la plataforma y, a menudo utilizado para el procesamiento de imágenes en tiempo real y la visión por ordenador. (OpenCV, 2015).

Estructura de OpenCv 2.4.8

OpenCv tiene una estructura modular, lo que significa que el paquete incluye varias bibliotecas compartidas o estáticas. Los siguientes módulos están disponibles:

- ✓ **Funcionalidad básica.** Un módulo compacto que define las estructuras de datos básicas, incluyendo las Mat (matriz multidimensional) y funciones básicas utilizadas por todos los demás módulos.
- ✓ **Procesamiento de imágenes.** Un módulo de procesamiento de imágenes que incluye el filtrado de imágenes no lineal y lineal, transformaciones geométricas de imágenes (redimensionar, afín y la perspectiva de deformación, Reasignación genérico basado en tablas), la conversión de espacio de color, histogramas, etc.
- ✓ **Procesamiento de vídeo.** Contiene un módulo de análisis de video que contiene la estimación de movimiento, la sustracción del fondo, y los algoritmos de seguimiento de objetos, (OpenCV, 2015).
- ✓ **calib3d.** Módulo básico con múltiples algoritmos de visión geométrica, calibración de cámara; algoritmos de correspondencia estéreo y elementos de la reconstrucción 3D.
- ✓ **features2d.** Detectores de características sobresalientes, descriptores y comparadores de descriptores.

- ✓ **objdetect.** Detección de objetos e instancias de las clases predefinidas (por ejemplo, caras, ojos, tazas, personas, vehículos, etc.).
- ✓ **highgui.** Una interfaz fácil de usar para las capacidades de interfaz de usuario sencilla.
- ✓ **videoio.** Una interfaz fácil de usar para la captura de vídeo y codecs de vídeo. (OpenCV, 2015)

CAPÍTULO III

3. DESARROLLO DEL ALGORITMO PARA PROCESAR LA IMAGEN

3.1. Visual servoing

Esta metodología llamada visual servoing o control servo visual, hace el uso de sistemas de visión por computador para controlar movimientos o acciones de un robot o plataforma, como en este caso es el control del vuelo del Dron Bebop 2, para realizar seguimiento de una persona en un área externa. Mediante esta técnica se puede adquirir datos de imágenes desde una cámara que se encuentre montada sobre una plataforma móvil o desde una cámara que se encuentre fija en un lugar de trabajo estacionario. En el primer caso, el movimiento de la plataforma móvil ocasiona el movimiento de la cámara. En el segundo caso se pueden considerar el colocar cámaras fijas en un área que observen los movimientos de robot, cualquier configuración similar está considerada Visual Servoing. (Maldonado & Calderon, 2014)

Las técnicas de procesamiento de imágenes conjuntamente con la teoría de control constituyen una base fundamental para el control servo visual; dicha técnica usa una cámara para suministrar información referente a la localización de un objeto con respecto a la imagen recogida y controlar el movimiento de un robot (Hutchinson, Hanger, & Corke, 1996). La información recopilada se usará para encontrar el error al relacionar la característica de referencia con la característica actual y con la realimentación se intentara corregir el error.

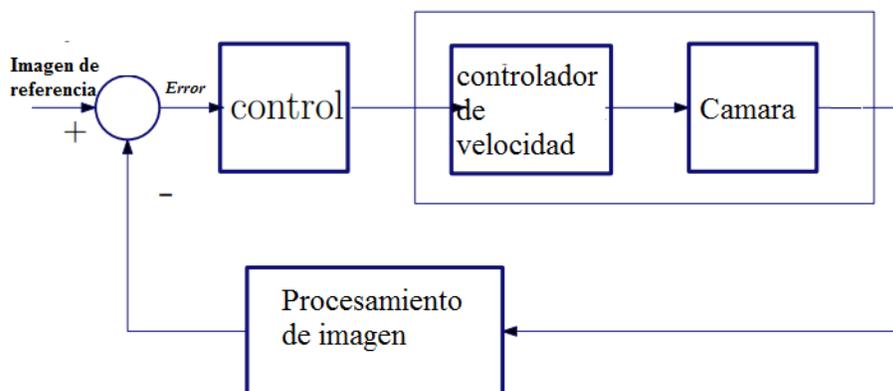


Figura 12. Sistema de control del VSC

El control servo visual se clasifica de acuerdo a las principales técnicas usadas para el control, las cuales se detallarán a continuación:

IBVS (Control servo visual basado en imagen): Los valores de control son calculados basándose en las características que se encuentren dentro de una imagen adquirida, comparando las características de referencia con las características actuales. Los valores mencionados de las imágenes 2D, se usarán para estimar el movimiento del robot respecto al objetivo, (Hutchinson, Hanger, & Corke, 1996).

PBVS (Control servo visual basado en posición): Las características son extraídas de una imagen y usadas en conjunto con un modelo geométrico del objetivo para estimar la posición del objeto con respecto a la cámara. La característica principal es recuperar la información 3D de la escena. (Hutchinson, Hanger, & Corke, 1996)

Ambas técnicas se usan para estimar los movimientos del robot o plataforma, mediante el uso de imágenes, sin embargo, IBVS, usa únicamente imágenes en 2D y características que benefician al proyecto actual, por ejemplo: reducción del tiempo computacional. (Hutchinson, Hanger, & Corke, 1996)

3.1.1. Control servo visual basado en imagen (IBVS)

Mediante esta técnica se extraen características desde una imagen, las cuales se usan para estimar la posición del objetivo con respecto a la cámara, no necesita el conocimiento de un modelo en el espacio 2D. De esta manera, el 2D control servo visual llega a denominarse también *image-based visual servoing*.

En general, el control servo visual basado en imágenes es mucho más robusto, no únicamente con respecto a la cámara sino también a los errores de calibración del robot. Sin embargo, su convergencia está asegurada sólo en una región (difícil de determinar analíticamente) alrededor de la posición deseada. Excepto en casos muy simples, los análisis de la estabilidad con respecto a los errores de calibración parecen imposibles ya que el sistema es acoplado y no lineal. (Maldonado & Calderon, 2014)

Como se mencionó anteriormente, la ley de control que rige esta técnica se basa en el error entre las características actuales y las deseadas en el plano de la imagen y no implica ninguna estimación de la pose (posición y orientación) del objetivo. Las características pueden ser las coordenadas de características visuales, las líneas o los momentos de las regiones. IBVS tiene ciertas dificultades cuando se trabaja con movimientos con rotaciones muy grandes, que son conocidos como retiro de cámara. A partir de esto, se entiende que primero es importante hacer un pre procesamiento de la imagen que contiene el objetivo a seguir, para después aplicar los algoritmos de interpretación y seguimiento.

A continuación, se consideran las siguientes etapas para conseguir el reconocimiento preciso del objetivo que será sometido a un proceso de seguimiento, con la manipulación de un cuadricóptero:

- 1) Adquisición de las imágenes (captura).
- 2) Pre procesamiento de la imagen; para mejorar la calidad o resaltar detalles que interesan.
- 3) Suavizar o eliminar ruido.
- 4) Segmentación: dividir la imagen en segmentos manipulables.

- 5) Extracción de características: tamaño, forma, características geométricas, etc.
- 6) Reconocimiento e interpretación.

3.2. Adquisición de la imagen

Las imágenes a obtener pueden provenir de cámaras convencionales existentes en el mercado con características que permitan una imagen óptima para la detección, tales como: calidad de imagen, comunicación en tiempo real, robustez y compatibilidad con el sistema operativo en uso para el desarrollo del proyecto.

3.2.1. Captura de imágenes desde una cámara

El procedimiento para la captura de imágenes, ya sea desde una cámara web o desde la cámara del Bebop2, es diferente dado a que estas inicialmente se manejan con ROS y por ende el video que se obtiene está en un formato de mensajes de ROS.

3.2.2. Conversión de imágenes de ROS a imágenes de OpenCv

ROS obtiene la imagen y la transporta bajo su formato de mensaje `sensor_msgs/image`, pero para el procesamiento de imagen en OpenCv es necesario cambiar el formato de imagen, por tal razón se usa el `cv_bridge` package, el cual define a `CvBridge` que es una biblioteca de ROS que proporciona una interfaz entre ROS y OpenCv, permitiendo convertir las imágenes de ROS en OpenCv (`cv::Mat`). (ROS.org, 2015)

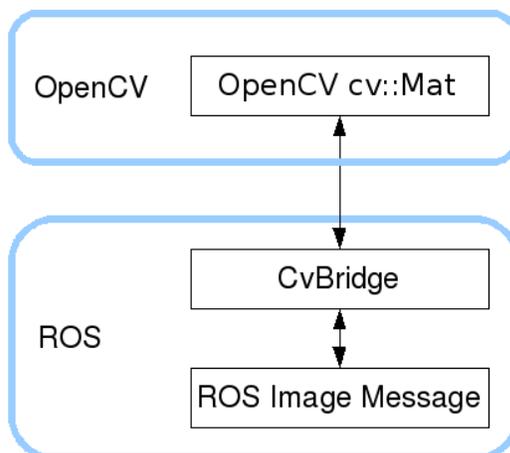


Figura 13. Proceso de conversión de imagen ROS a imagen OpenCv

Fuente: (ROS.org, 2015)

Una vez que la imagen se establezca manipulable para OpenCv, se desplegará en una ventana usando el comando `imshow()`, con la información exacta, sin sufrir ningún tipo de pérdida.

3.3. Reconocimiento de objetos en imagen

La visión artificial se constituye como un aspecto muy importante para el reconocimiento o identificación de objetos, ya sea clasificándolos por formas o colores, de manera independiente según sea el método de identificación. En ciertos casos permite la identificación de patrones establecidos, por lo que el reconocimiento identifica desde formas muy simples hasta formas complejas, permitiendo su aplicación en sistemas avanzados que en un futuro podrían ser capaces de reconocer personas por sus caras o aspectos físicos de manera similar a como lo hacen los humanos.

3.3.1. Reconocimiento de objetos usando OpenCv

OpenCv permite usar sus recursos como librerías especializadas en visión artificial, brindando recursos que permiten discriminar objetos o personas de una determinada imagen de entrada, mediante técnicas de extracción de características. Entre las más conocidas se encuentra el algoritmo de HOG

(Histograma de gradientes orientados), mismo que debido a su robustez ante cambios de luz, permitirá definir de mejor manera el contorno de una persona.



Figura 14. Reconocimiento de Objetos mediante OpenCv

Por otro lado, OpenCv, permite el uso de clasificadores pre-entrenados que permitirá una detección de personas óptima y más aproximada, permitiendo reducir falsos positivos dentro de la imagen de detección, siendo el más recomendable el clasificador SVM (Máquina de Soporte Vectorial), porque ha mostrado tener gran desempeño, inclusive más que las máquinas de aprendizaje tradicional como las redes neuronales. (Betancourt, 2005)

3.4. Descriptores HOG (Histogramas de gradientes orientados)

Los descriptores de HOG trabajan basándose en la orientación del gradiente dentro de áreas locales de una imagen. Dicha imagen se divide en varias celdas, las cuales contienen direcciones del histograma de gradiente u orientaciones de los bordes de los pixeles de las celdas, como se muestra en la figura 15.

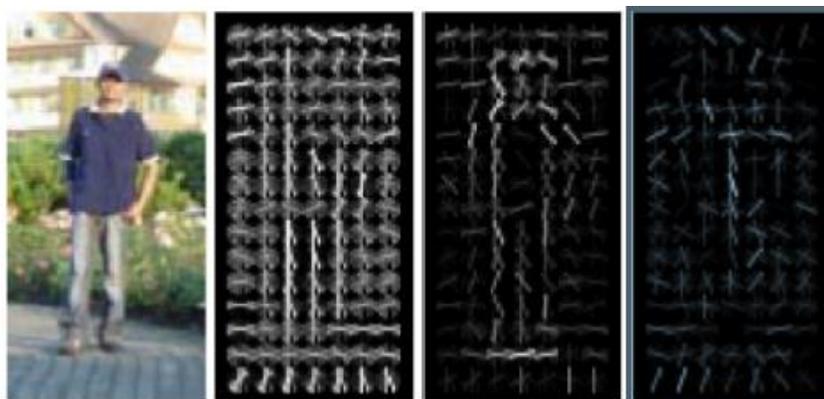


Figura 15. Obtención de imagen y aplicación del descriptor HOG

Fuente: (Intxaurbe J., 2013)

El principal objetivo del descriptor de HOG, es conseguir la máxima invariancia posible ante cambios de posición, iluminación, etc., por ejemplo, al obtener 2 imágenes, con diferente intensidad de luminosidad. La figura 16 muestra a la izquierda una imagen menos iluminada que al tomar una celda de la misma, nos proporcionará un histograma que refleja valores menores (un gradiente más pequeño), mientras que, en la segunda imagen con más iluminación, los valores del histograma son mayores (el valor de su gradiente será más grande).

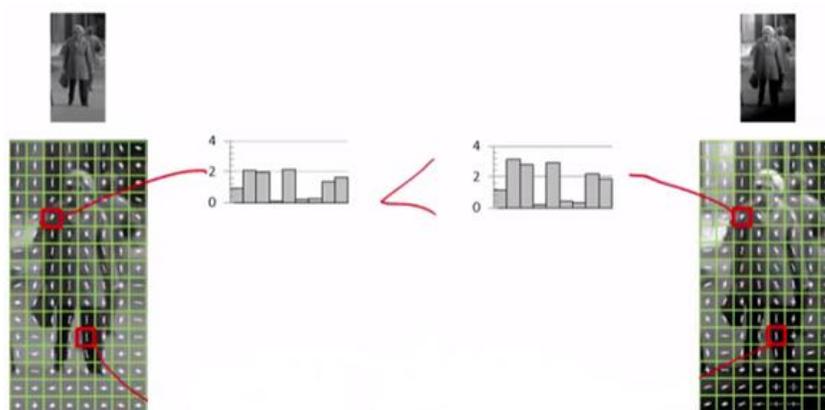


Figura 16. Imágenes, histogramas y gradientes del descriptor HOG

Fuente: (Valveny E., 2015)

Para minimizar estas diferencias en la descripción de las imágenes, es conveniente normalizar los valores de los histogramas, con el objetivo de conseguir que las magnitudes del gradiente sean similares en ambas imágenes. Dicha normalización es recomendable realizar en zonas más grandes llamadas bloques (agrupaciones de celdas generalmente de $b \times b$), para obtener mejores resultados. Estos bloques de descriptores normalizados son lo que los autores denominan descriptores HOG, mostrados en la figura 17.

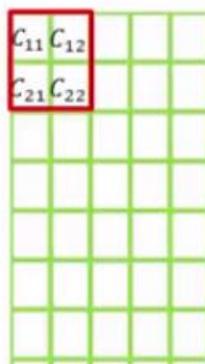


Figura 17. Bloques para normalización de una imagen

Fuente: (Valveny E., 2015)

Como resultado de la normalización, los valores globales (gradiente) de los histogramas son iguales, por lo tanto, las diferencias de valores se han reducido para la representación final entre imágenes similares, obteniendo así un descriptor final robusto. (Valveny E., 2015)

Finalmente, encontrados los gradientes normalizados de cada bloque de la imagen obtenida con la cámara, se utilizan como datos de entrada para un clasificador SVM.

La información que los descriptores HOG proporcionan son los cambios de intensidad luminosa, debido a los contornos o bordes de una imagen, al tener en cuenta la relación entre sus zonas vecinas y colindantes, se puede reconocer la frontera de cada objeto, permitiendo aislar uno de otro, independientemente de su forma geométrica o color. Por lo tanto, se puede identificar objetos con siluetas más pronunciadas o más suaves, fijándonos más en la relación de los objetos en

relación con el entorno, distinguiendo los cambios más pronunciados. Por lo tanto, HOG es adecuado para la detección de personas.

3.4.1. Descripción básica de HOG

El método de HOG calcula los gradientes de una imagen, dividiendo dicha imagen en bloques, y estos a su vez se dividen en celdas y calcula en cada una de ellas, su histograma y gradiente.

El algoritmo HOG es capaz de detectar la presencia de personas presentes en el campo de visión de la cámara. Una vez detectado un sujeto, el dron puede ser alertado de la presencia de este con la suficiente antelación como para poder despegar y realizar el seguimiento, es por ello que se solicita que el programa procese las imágenes tomadas con la mayor brevedad posible. El cálculo de los descriptores HOG presenta un coste computacional un poco elevado por el hecho de calcular los gradientes HOG en cada una de las celdas.

Este método para detección de personas en la escena se destaca por su robustez frente a diferentes condiciones de iluminación, pequeños cambios en el contorno de la imagen y diferencias de fondos y de escalas.

Dada una imagen en color, lo primero que se hace es transformar a escala de grises. A continuación, se calculan los gradientes espaciales sobre toda la imagen. Posteriormente, se divide la imagen en bloques solapados cierta área para lograr un mejor barrido. El avance de bloques se realiza eliminando la columna de las celdas de la izquierda y añadiendo la columna de la derecha para el desplazamiento horizontal, mientras que para el vertical se elimina la fila de las celdas de arriba, añadiendo la fila de celdas de abajo. A su vez, cada bloque se divide en subregiones o celdas, calculándose en cada uno de ellos el histograma de los gradientes orientados, de tal forma que se logra mejorar el rendimiento.

A continuación, como se puede apreciar en la figura 18, se aplica una ventana o filtro gaussiano sobre cada bloque, dado que este tiene un fundamento matemático y un comportamiento muy similar al sistema de visión humano (HVS);

por ello es muy importante ya que se especializa en la detección de cambios bruscos como los bordes. (Escalante Boris, 2006)

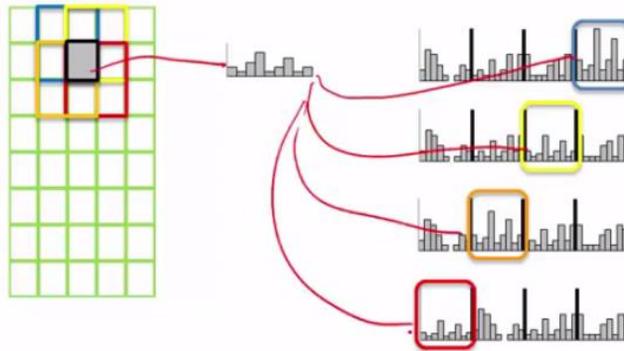


Figura 18. Solapamiento de bloques e Histogramas de cada celda por bloque

Fuente: (Valveny E., 2015)

Finalmente, el resultado de aplicar el filtro, se almacena en el vector de características de la imagen.

El cálculo de los gradientes de HOG de una imagen, su normalización de bloques y el almacenamiento del vector de características que se explicó anteriormente, sigue el procedimiento que se muestra en la figura 19.

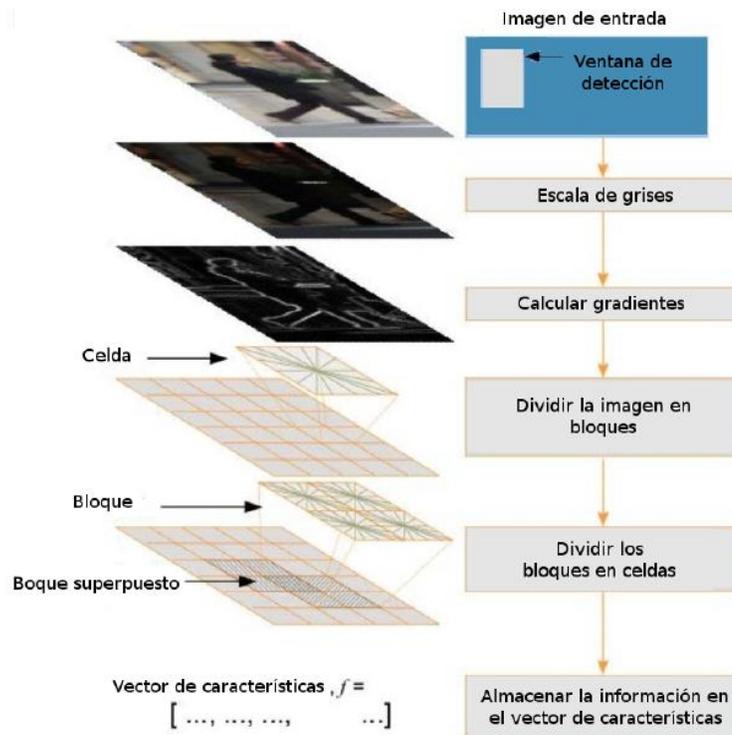


Figura 19. Proceso de extracción, cálculo de gradientes y almacenamiento de características

Fuente: (Intxaurbe J., 2013)

3.4.2. Descripción matemática de HOG

El histograma de una imagen divide el rango de valores posibles de los píxeles de la imagen en una serie de sub rangos o clases de mismo o distinto tamaño entre ellos. Por ejemplo: dado el rango de valores de píxel $[0,255]$, se realiza una división en ocho clases del mismo tamaño: $[0,32)$, $[32, 64)$, $[64, 96)$ $[224, 255]$, y almacena en cada clase la frecuencia de píxeles con un valor comprendido entre ese sub rango, es decir, el número de píxeles en la imagen cuyo valor está entre los valores de inicio y fin de cada sub rango.

El histograma de gradientes orientados de una imagen tiene como rango de valores posibles las distintas orientaciones que pueden tomar los gradientes de los píxeles, es decir los distintos grados que pueden tomar sus ángulos de gradiente. (Por ejemplo, $[-90^\circ, 90^\circ]$, $[0^\circ, 180^\circ]$, $[0^\circ, 360^\circ]$...). Este rango se divide

en sub clases del mismo tamaño o distintos (para el rango $[0^\circ, 180^\circ]$, dividiendo en nueve sub rangos: $[0^\circ, 20^\circ)$, $[20^\circ, 40^\circ)$ $[160^\circ, 180^\circ]$), y se almacena en cada uno de ellas la suma de las magnitudes de gradiente de los píxeles, cuyo ángulo de gradiente se encuentra comprendido entre esos valores. Partiendo del concepto de los HOG, se puede obtener más información de una imagen por medio de un descriptor de HOG. En éste, la imagen se divide en un cierto número de sub-imágenes del mismo tamaño, denominadas celdas, y éstas se agrupan en bloques con un mismo número de celdas de ancho y alto todos ellos. Además, estos bloques se encuentran solapados de forma que el avance de bloques horizontalmente se realiza eliminando la columna de celdas de la izquierda y añadiendo la columna de la derecha y, verticalmente, eliminando la fila de celdas de arriba y añadiendo la fila de celdas de abajo.

De este modo, dados una imagen A de tamaño $W \times H$; un tamaño de celda $C_W \times C_H$ con $W \bmod C_W = 0$ y $H \bmod C_H = 0$; un tamaño de bloque en celdas $B_W \times B_H$; el ancho y alto de la imagen en celdas, W_C y H_C ; y el número de bloques distribuidos horizontalmente y verticalmente, N_{BW} y N_{BH} , se calculan de la siguiente manera:

$$W_C = \frac{W}{C_W}$$

$$H_C = \frac{H}{C_H}$$

$$N_{BW} = 1 + W_C - B_W$$

$$N_{BH} = 1 + H_C - B_H$$

Y, por tanto, el número total de celdas N_C y el número total de bloques N_B resultantes de la imagen A será igual a:

$$N_C = W_C * H_C$$

$$N_B = N_{BW} * N_{BH}$$

Y la distribución de celdas (C) y bloques (B) es la siguiente:

$$A = \begin{pmatrix} a_{00} & a_{10} & \dots & a_{(W-1)0} \\ a_{01} & a_{11} & \dots & a_{(W-1)1} \\ \vdots & \vdots & \ddots & \vdots \\ a_{0(H-1)} & a_{1(H-1)} & \dots & a_{(W-1)(H-1)} \end{pmatrix}$$

$$C = \begin{pmatrix} c_{00} & c_{10} & \dots & c_{(W_C-1)0} \\ c_{01} & c_{11} & \dots & c_{(W_C-1)1} \\ \vdots & \vdots & \ddots & \vdots \\ c_{0(H_C-1)} & c_{1(H_C-1)} & \dots & c_{(W_C-1)(H_C-1)} \end{pmatrix}$$

$$B = \begin{pmatrix} b_{00} & b_{10} & \dots & b_{(N_B W-1)0} \\ b_{01} & b_{11} & \dots & b_{(N_B W-1)1} \\ \vdots & \vdots & \ddots & \vdots \\ b_{0(B_H-1)} & b_{1(B_H-1)} & \dots & b_{(N_B W-1)(N_B H-1)} \end{pmatrix}$$

A partir de esta estructuración de la imagen, el descriptor de HOG calcula de forma independiente el HOG de cada celda y cada bloque agrupa los HOGs de sus celdas correspondientes. Entonces, el número de HOG que contiene un descriptor será:

$$N_{HOG} = B_W * B_H * N_B$$

Y si se divide el HOG en n clases, dado que cada bloque contiene $B_W * B_H$ descriptores HOG, entonces el número total de valores N_V que se tomará de la imagen A será

$$N_V = n * N_{HOG}$$

Para el cálculo del modelo de detección, sobre la colección de imágenes se debe calcular el descriptor de HOG de cada imagen, etiquetando cada descriptor como positivo si es una imagen de persona (+1) o negativo si no lo es (-1). El cálculo de los descriptores de HOG puede tener un coste en tiempo de computación bastante grande, para la aplicación en este proyecto, pues requiere

del cálculo de un HOG por cada celda. Para agilizar esto se puede usar una técnica denominada Integral de HOG por el cual se mejora el coste de computación a cambio de aumentar el coste en memoria.

3.5. Máquina de vectores soporte SVM (Support Vector Machine)

Las máquinas de vectores soporte son estructuras de aprendizaje basadas en la teoría estadística del aprendizaje, transforman el espacio de entrada en otro de dimensión superior en el que el problema puede ser resuelto mediante un hiperplano óptimo (de máximo margen).

Las máquinas de vectores soporte es un clasificador lineal basado en el concepto de margen máximo a partir de vectores de soporte, dicho margen es la región más amplia del espacio de características en el cual no existen muestras. El plano intermedio (hiperplano) que se obtiene dentro del espacio de características, es la solución de la máquina de vectores soporte, en la figura 22 se aprecia lo estipulado.

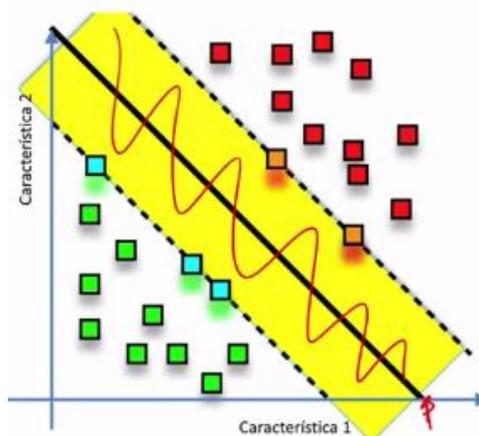


Figura 20. Hiperplano obtenido dentro del espacio de características

Este método está plenamente diseñado para resolver problemas de clasificación, por lo que, dado un conjunto de muestras, se puede etiquetar las clases, para separarlas y entrenar una SVM para construir un modelo que prediga

la clase de una nueva muestra. Una SVM intuitivamente, es un modelo que representa los puntos de muestra en el espacio, alejando las clases por un espacio lo más amplio posible. Cuando las nuevas muestras se ubican en correspondencia con dicho modelo, en función de su cercanía pueden ser clasificadas como pertenecientes o no perteneciente a una u otra clase.

Entonces una SVM construye un hiperplano en medio del espacio de características de dimensionalidad muy alta, este espacio de características está delimitado por los vectores soporte, mismos que podrán ser utilizados en problemas de clasificación. Una buena separación entre las clases permitirá una clasificación correcta.

3.5.1. Descripción básica de SVM

Las máquinas de soporte vectorial o máquinas de vectores de soporte son una colección de algoritmos de aprendizaje supervisado elaborados principalmente para la clasificación. Utilizando un conjunto de ejemplos de entrenamiento se puede etiquetar las clases y entrenar una SVM para elaborar un modelo que prediga la clase de una nueva muestra.

Tomando los datos de entrada como conjuntos de vectores en un espacio n -dimensional, una máquina de vectores soporte construirá un hiperplano de separación en ese espacio. Se considera que es mejor clasificador de datos aquel hiperplano que maximice la distancia con respecto a los puntos que estén más cerca de él. Siendo los vectores de soporte los puntos que tocan el límite del margen.

El uso de la máquina de vectores soporte, para la detección de personas, permitirá separar la clase positiva de la clase negativa, es decir los datos que corresponden a la clase **Persona** son muestras positivas, en cambio la clase **no Persona**, representará muestras negativas como se muestra en la figura 21.

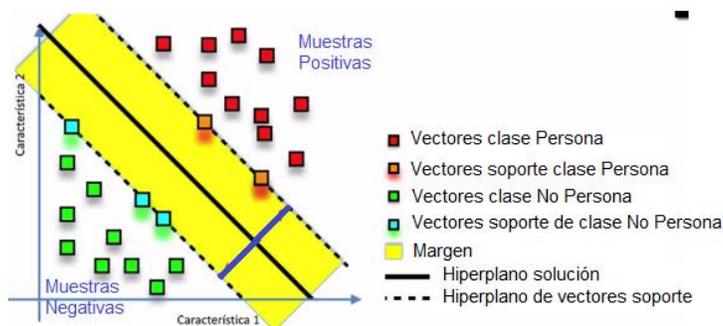


Figura 21. Clasificación de muestras Persona y no persona

En ese concepto de separación óptima es donde reside la característica fundamental de las SVM: este tipo de algoritmos buscan el hiperplano que tenga la máxima distancia (margen) con los puntos que estén más cerca de él mismo. Por tal motivo, a veces se les conoce a las SVM como clasificadores de margen máximo. De esta forma, se separan los puntos del vector que son etiquetados con una categoría a un lado del hiperplano y los casos que se encuentren en la otra categoría estarán al otro lado.

Para separar linealmente los datos se procede a realizar un cambio de espacio mediante una función que transforme los datos de manera que se puedan separar linealmente. Esta función recibe el nombre de **Kernel**.

En este caso, los conjuntos son “personas” y “no personas”. Para ello, se necesita un entrenamiento previo de la máquina, facilitándole ejemplos de personas o “positivos” y ejemplos de no-personas o “negativos”. Con todos los ejemplos de entrenamiento, el algoritmo de clasificación SVM elabora una curva M-dimensional que divide ambos conjuntos, obteniendo de esta forma el kernel de la máquina. Las dimensiones del espacio dependen del número de componentes de cada vector a clasificar.

CAPÍTULO IV

4. IMPLEMENTACIÓN DE SEGUIMIENTO

4.1. ROS

Al ser un Sistema Operativo Robótico (ROS) y contar con varios paquetes, hace de esta plataforma un recurso formidable para el desarrollo de aplicaciones: su principal característica es la reutilización de códigos en la investigación y desarrollo de aplicaciones para robots. Las bases en la que esta instaurado ROS, hace que sea fundamental en este proyecto, por lo que se han creado y configurado nodos para el vuelo autónomo de un cuadricóptero mientras este sigue una persona.



Figura 22. Comunicación Operacional ROS

4.2. Paquete bebop_autonomy

El paquete `bebop_autonomy` tiene mucho peso e importancia para este proyecto, porque cuenta con controladores de comunicación entre la plataforma ROS y el Parrot Bebop2 que será la planta de este proyecto. La versión actualizada, se llama **parrot_arsdk**, y es una dependencia necesaria para el proceso de creación de un paquete para cualquier proyecto en el que se desee trabajar con cuadricópteros de la serie Parrot Bebop. Cabe recalcar que no tiene similitud con el ARSDK que fue creado para poder controlar los cuadricóptero Parrot AR DRON. Este paquete viene instaurado con nodos a los cuales se puede

tener acceso y por ende a la información de cada uno de los sensores e incluso se puede enviar comandos para controlar este cuadricóptero, (Monajjemi Mani., AutonomyLab, Simon Fraser University, 2015).

El paquete `bebop_autonomy` está facultado para ser usado con Parrot Bebop 1 y Parrot Bebop 2, al usar dicho paquete con el cuadricóptero Parrot Bebop 2 se debe actualizar `parrot_arsdk`, puesto a que tiene mejores prestaciones que su antecesor y necesita de ciertos comandos específicos. Por ejemplo `/bebop_ws/land` que sirve para el despegue, `/bebop_ws/reset`, utilizado para el reseteo de las variables que se encuentran cargadas después de un vuelo, y también es usado en caso de emergencia, haciendo que los motores se detengan por completo. En el cuadricóptero no se debe olvidar de tener actualizado su firmware, para tener un mejor desempeño al momento de la comunicación (Monajjemi Mani., AutonomyLab, Simon Fraser University, 2015).



Figura 23. Bebop_autonomy

Fuente: (Parrot S.A, 2016)

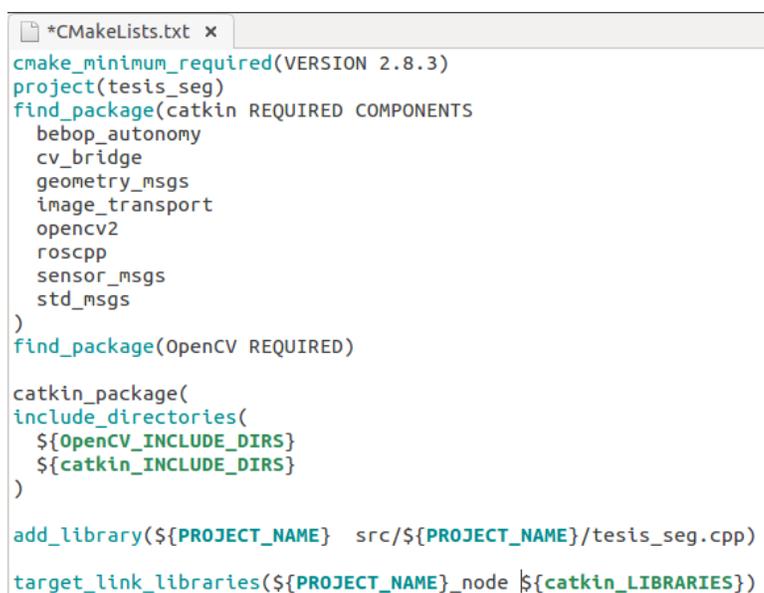
4.3. Configuración del paquete y librerías

La creación del *workspace* (Área de trabajo) y posteriormente la configuración de los diferentes paquetes son indispensables. Al configurar el paquete `bebop_autonomy` se podrá tener acceso a la comunicación e información de diferentes sensores abordo del Parrot Bebop 2, lo cual ayuda a la creación del nuevo paquete que tendrá como objetivo el seguimiento autónomo de personas con dicho cuadricóptero. Esto se hace para que el *firmware* sepa su respectivo direccionamiento de las dependencias.

Cuando se crea un nuevo paquete en la plataforma ROS, se debe declarar las dependencias que se van a ocupar en el código fuente, ya sean librerías o

paquetes. Este proyecto dependerá de paquetes estándar que vienen instalados en ROS, como es el caso `image_transport`, `roscpp`, `std_msgs`, `OpenCv2`, `cv_bridge`, `sensor_msgs`, `geometry_msgs` y un paquete externo llamado `bebop_autonomy`,

A parte de esto se configuran los archivos `manifest.xml` y `CMakeList.txt`, los cuales son usados por el proyecto al momento de la compilación que realiza el CMake, siendo este último utilizado por ROS para verificar incongruencias, errores de declaración de variables, etc. En el archivo `CMakeList.txt` se cargan las dependencias de librerías, como es el caso de `roscpp`, `cv_bridge`, entre otras, librerías externas como lo es `OpenCv`, se especifican los nodos que van a ejecutarse, los archivos fuente, los paquetes que van a ser utilizados al momento de correr la aplicación en ROS, un ejemplo de esto se muestra en la figura 24.



```
*CMakeLists.txt x
cmake_minimum_required(VERSION 2.8.3)
project(tesis_seg)
find_package(catkin REQUIRED COMPONENTS
  bebop_autonomy
  cv_bridge
  geometry_msgs
  image_transport
  opencv2
  roscpp
  sensor_msgs
  std_msgs
)
find_package(OpenCV REQUIRED)

catkin_package(
  include_directories(
    ${OpenCV_INCLUDE_DIRS}
    ${catkin_INCLUDE_DIRS}
  )
)

add_library(${PROJECT_NAME} src/${PROJECT_NAME}/tesis_seg.cpp)

target_link_libraries(${PROJECT_NAME}_node ${catkin_LIBRARIES})
```

Figura 24. CMakeList.txt

El archivo `Package.xml` muestra las dependencias utilizadas para este proyecto; este archivo ayuda en la construcción y ejecución de las mismas al momento de compilar toda la aplicación. En la figura 25 se muestra como quedan configuradas las dependencias, teniendo en cuenta que estas se crean a partir del archivo `CMakeList.txt`.

```

*package.xml x
<?xml version="1.0"?>
<package>
  <name>tesis_seg</name>
  <version>0.0.0</version>
  <description>The tesis_seg package</description>
  <maintainer email="carlos@todo.todo">carlos</maintainer>
  <license>TODO</license>

  <buildtool_depend>catkin</buildtool_depend>
  <build_depend>bebop_autonomy</build_depend>
  <build_depend>cv_bridge</build_depend>
  <build_depend>geometry_msgs</build_depend>
  <build_depend>image_transport</build_depend>
  <build_depend>opencv2</build_depend>
  <build_depend>roscpp</build_depend>
  <build_depend>sensor_msgs</build_depend>
  <build_depend>std_msgs</build_depend>
  <run_depend>bebop_autonomy</run_depend>
  <run_depend>cv_bridge</run_depend>
  <run_depend>geometry_msgs</run_depend>
  <run_depend>image_transport</run_depend>
  <run_depend>opencv2</run_depend>
  <run_depend>roscpp</run_depend>
  <run_depend>sensor_msgs</run_depend>
  <run_depend>std_msgs</run_depend>
  <export>
</export>
</package>

```

Figura 25. Package.xml

4.4. Configuración de los archivos “launch”

Un archivo “launch” se configura para permitir ejecutar de manera rápida y efectiva los diferentes nodos que se hayan creado, ya que dicho archivo, se ejecuta solo con escribir un comando en el terminal (roslaunch [nombre del archivo]). Para este proyecto se ejecutan tres archivos launch: el primero permite ejecutar el nodo del *bebop_driver*, abriendo así el flujo de imagen desde la cámara del cuadricóptero Parrot Bebop 2, hacia el algoritmo que realiza el procesamiento de imágenes, teniendo en cuenta que este ya tiene por defecto calibraciones para la cámara, ya sea en tamaño (640 * 480 pixeles) y posición. El segundo archivo ejecuta el nodo para visualizar la interfaz para el usuario donde se verifica, si se puede empezar o no con el seguimiento, así como la información de salidas tanto de PID, estado del cuadricóptero, velocidad, altura entre otra información. Finalmente, el tercer archivo ejecuta el nodo que se encargará de abrir el flujo de video desde la webcam o cámara estática, hacia el algoritmo de detección de personas. En la figura 26, se puede observar la configuración inicial para el control PID de cada eje, se muestran las diferentes ganancias del controlador antes mencionado (pGain, iGain, dGain), el error

máximo (iMax), y el valor máximo que puede tomar cada ganancia del controlador pudiendo esta variar desde 0.1 hasta 1 (pid_max_[eje]).

```

<!--xml-->
<launch>
  <node name="run_bebop" pkg="bebop_drone"
  type="run_bebop" output="screen">
    <param name="pGain_roll" value="0.62" />
    <param name="iGain_roll" value="0.006" />
    <param name="dGain_roll" value="2" />
    <param name="iMax_roll" value="0.2" />
    <param name="pid_max_roll" value="0.8" />

    <param name="pGain_pitch" value="0.62" />
    <param name="iGain_pitch" value="0.006" />
    <param name="dGain_pitch" value="2" />
    <param name="iMax_pitch" value="0.2" />
    <param name="pid_max_pitch" value="0.5" />
  </node>
</launch>

```

Figura 26. Ejemplo de configuración del archivo launch (PID)

La interconexión de nodos ejecutados para la ejecución total de este proyecto se la puede observar en la figura 27, donde se muestra de color rojo uno de nodos ejecutados y también son visibles al tomar una forma de elipse.

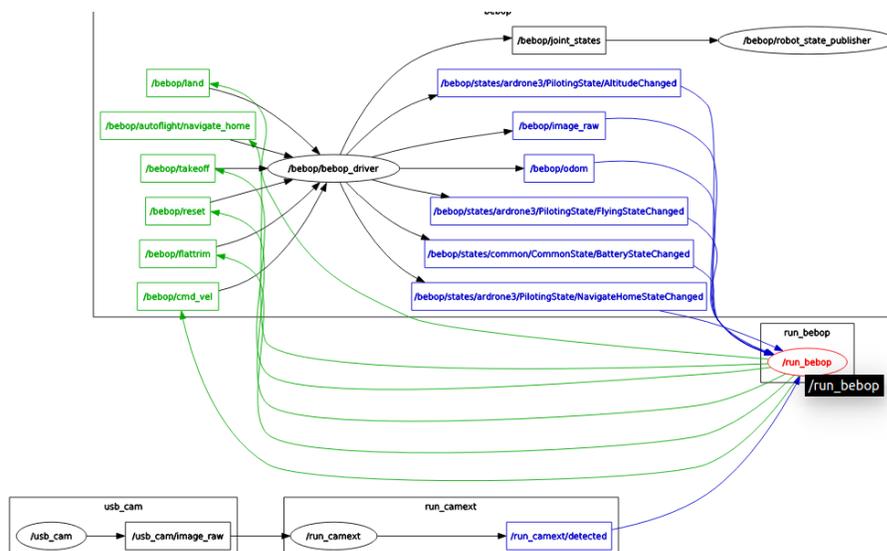


Figura 27. Nodos ejecutados con archivo launch

4.5. Ejecutando el sistema en ROS

Al ejecutar la aplicación se debe tener en cuenta que ROS realiza comunicaciones internas entre nodos, todo esto se hace mediante suscripciones

y publicaciones de tópicos que son los encargados de transmitir y recibir la información; para este caso se debe comunicar el nodo *bebop_driver*, el nodo *run_bebop* y el nodo *cámara_ext*.

Las suscripciones sirven para recibir información que envía el nodo. Entre las más importantes están: el flujo de video que se obtiene de la cámara del cuadricóptero, mediante el uso del tópico llamado */bebop/image_raw*, la posición y velocidad obtenidos del tópico */bebop/odom*. A estos tópicos y otros más se encuentra suscrito el nodo *run_bebop*, mismo que recibirá toda la información necesaria que se requiere para publicarla en el interfaz de usuario.

Como se había mencionado, también existen los tópicos de publicación, los cuales son usados para enviar información desde el nodo *run_bebop* al nodo *bebop_driver*, que en este caso serán comandos para que el cuadricóptero emprenda acciones; ya sean de despegue publicando el tópico */bebop/takeoff* o de aterrizaje publicando el tópico */bebop/land*. Para una mejor comprensión, tanto los tópicos de publicación como suscripción se muestran en la figura 28.

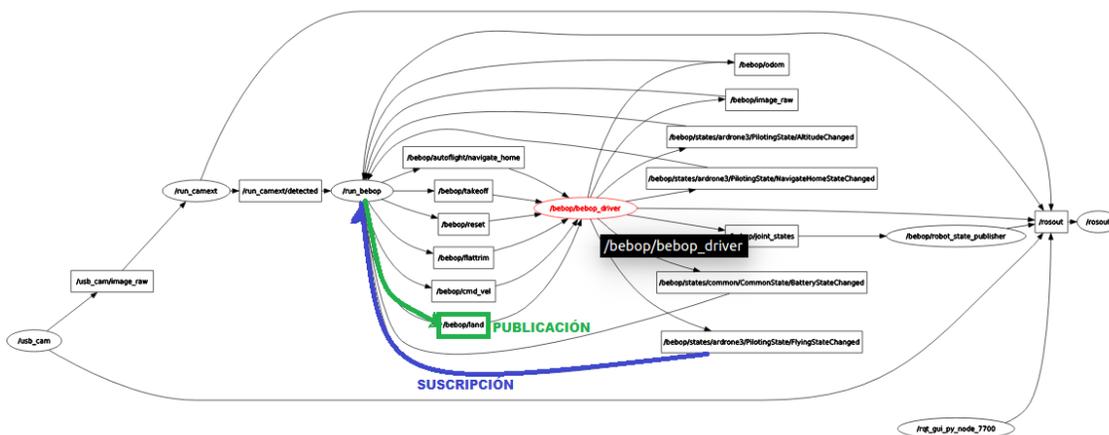


Figura 28. Suscripción y publicación de tópicos

Una vez realizada la comunicación entre los diferentes nodos, por medio de los tópicos antes mencionados, se puede observar en la figura 29 cómo quedan unidos los diferentes nodos teniendo en cuenta que los tópicos de publicación se

muestran con flechas de salida y los tópicos de suscripción tienen saetas de entrada.

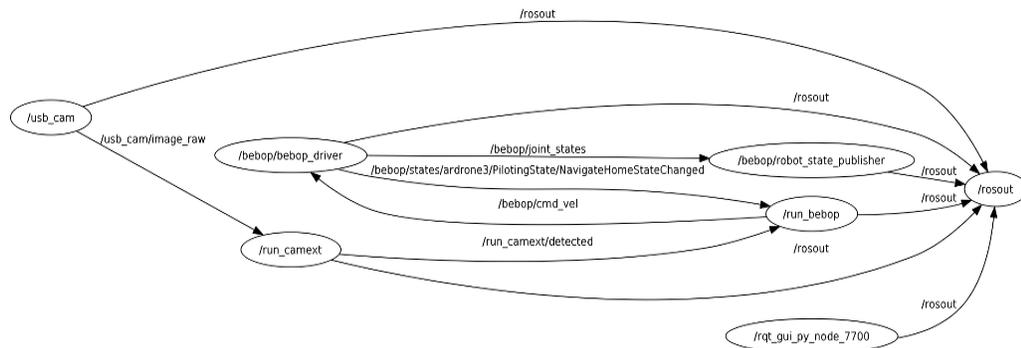


Figura 29. Comunicación entre nodos mediante tópicos

Para empezar la ejecución de la aplicación es necesario acceder al terminal, y ejecutar los archivos launch que se encargarán de la activación de los nodos y la comunicación entre el cuadricóptero y la computadora. Se muestra la ventana principal donde se aprecian los datos que son enviados para el control de cuadricóptero, logrando tener una mejor idea de cómo se está realizando el proceso. En la figura 30 se observa los diferentes estados y datos proporcionados por el nodo **bebop_drive** y por el procesamiento de imagen.

```

/home/carlos/bebop_ws/src/bebop_drone/launch/bebop_run.launch http://localhost:11311
/home/carlos/bebop_ws/sr... x /home/carlos/bebop_ws/sr... x /home/carlos/bebop_ws/sr... x
UNIVERSIDAD DE LAS FUERZAS ARMADAS
ESPE EXTENSION LATACUNGA
CONTROL POR VISION DE UN CUADRICOPTERO UTILIZANDO ROS.
CARLOS PAUCAR MARISOL MORALES

- Fig encontrada : SI - CMD Usuario : HOME
- Per_seguimiento : SI - Estado Drone: landed
- Estado homestate: unavailable
- Estado homereason: userRequest
- Init Seguimiento: OK
Centro/radio=[16928,000000 , -428 , -84]
Set point=[0,081 , 0,998 PERMISO DE DESPEGUE [-1 ]

-Battery=92,0%
-Altura = 0,000 cm
-Velocidad(X,Y)[m/s] = [ -0,000 , 0,000 , 0,000 ]
-Posicion(X,Y)[m/s] = [ 0,000 , 0,000 , 0,000 ]
-Angulos(pitch,roll,yaw)[grad]= [ 0,00 , 0,01 , -0,14 ]]
-PID =[0,121,0,002, 0,12 , -0,01 , -0,0,0,0]]
-PID =[0,303,0,006, 0,30 , 0,42 , 0,1,0,0]]]
-PID =[0,000,0,000, 0,00 , 0,00 , 0,0,0,0]

```

Figura 30. Pantalla de interfaz principal

De la figura 30 se puede deducir sus partes principales:

- **Fig Encontrada:** Muestra si se ha detectado una persona.
- **Per_seguimiento:** Indica que el seguimiento de la persona puede iniciar si este se encuentra marcado con la palabra SI, este estado se lo puede realizar de forma manual o autónoma. En la forma manual debe cumplirse la condición que se haya detectado una persona y se presione la tecla P con lo cual se empieza el seguimiento; en el modo autónomo la cámara estática que en este caso es la webcam enviará el permiso para el seguimiento, haciendo que empiece de forma autónoma el seguimiento.
- **Init_Seguimiento:** Al aparecer la palabra OK, muestra que se ha encontrado la figura y se tiene el permiso de seguimiento sea de modo manual o de modo autónomo, e indica que el computador empezará a enviar las órdenes de control al cuadricóptero.
- **Centro/radio, set_point:** Se aprecia la información de ubicación del objeto encontrado en la imagen y donde se encuentra con respecto al campo visual de la cámara, en el apartado de set_point, se observan los valores de entrada, con lo que se calcula el error.
- **Bateria, altura, velocidad, posición, ángulos:** Esta información es entregada por el computador del cuadricóptero Bebop 2, la cual se aprecia en tiempo real y permite al usuario tener conocimiento inmediato de posición, ángulos, altura y la carga de la batería.
- **PID:** Presenta los valores calculados por el modelo matemático para las diferentes ganancias del controlador PID.
- **CMD Usuario:** Este apartado muestra los diferentes estados que son enviados por el usuario para el control del cuadricóptero.
- **Estado_Dron:** Muestra el respectivo estado del Bebop 2, todo esto lo hace en tiempo real, los estados que se pueden apreciar son: Landed (aterrizado), Hovering (planeando), Landing (aterrizar), Reset (resetear), Takeoff (despegar).
- **Estado homestate:** Nos indica cómo se encuentra el estado de regreso a casa.

- **Estado *homereason*:** Muestra la razón por la cual se está regresando a casa pudiendo ser `userRequest` (orden del usuario), `connectionLost` (Conexión con el controlador perdida), `lowBattery` (batería Baja).
- **Permiso de despegue:** Es el valor que envía la cámara estática, para que el sistema empiece el seguimiento autónomo. Cuando el valor es 1, la cámara ha detectado una persona y envía la orden de despegue, y cuando es -1, indica que no se ha detectado a ninguna persona.

4.6. Órdenes del usuario por teclado

Se puede enviar órdenes de control al cuadricóptero por medio del teclado, mismas que pueden interferir o facilitar dicho control; en la tabla 3 se puede apreciar las teclas asignadas para cada comando en el control manual.

Tabla 3.

Órdenes por teclado para control del Bebop 2

| Comandos desde teclado | | |
|------------------------|------------------------|---|
| Tecla | Comando | Descripción |
| T | Take off | Envía la orden de despegue al cuadricóptero |
| L | Landed | Envía la orden de aterrizaje al cuadricóptero |
| R | Reset | Resetea los parámetros del Bebop2 |
| F | Flatrim | Calibra la cámara a la posición frontal, en función de la orientación del Bebop 2 |
| P | Permiso de Seguimiento | Permite iniciar el seguimiento de modo manual enviando la orden por medio del teclado |
| H | Home | Envía la orden de regreso a casa |

4.7. Estimación de pose

La adquisición de imágenes es fundamental para este proyecto por lo que en parte se seleccionó el dron BEBOP 2, debido a que su cámara frontal incorporada presenta mejores prestaciones en comparación a las del Ar.Dron 2.0. La cámara del Bebop2 tiene una cámara *fish eye Sunny*, diagonal 180°, con sensor CMOS 14Mpx y resolución de vídeo de 1920x1080p (30fps), (Parrot S.A, 2016).

El procesamiento de imágenes debe ser rápido, al tratarse de un sistema de seguridad el seguimiento en vuelo debe ser en tiempo real o lo más aproximado posible. Bajo estas circunstancias se usa la resolución de 640 x 480 pixeles, ya que, al reducir de la resolución original de la cámara, ayudará a procesar rápidamente, por lo tanto, las órdenes de control actuarán adecuadamente al momento de ser enviadas al dron, para el seguimiento de personas.

Las imágenes receptadas por el dron deben contener obligatoriamente a la persona a seguir, para enviar las órdenes de control y realizar las acciones pertinentes. La figura 31, muestra el área del campo visual que abarcará la cámara frontal del dron, dentro de la cual debe estar la persona u objeto a ser seguido, en caso de no encontrar su objetivo dentro del campo visual, se procederá a mantenerse estático en vuelo.

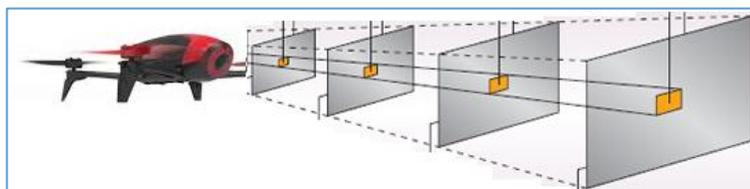


Figura 31. Campo visual de cámara Bebop2

A continuación, la figura 32 representa la resolución de imágenes usada, siendo esta la más apropiada ya que permite captar imágenes claras para el procesamiento rápido, sin retardo como ocurre al usar una resolución más alta.

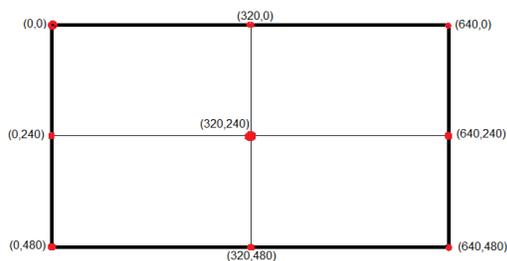


Figura 32. Tamaño de la imagen 640x480p

Establecido el tamaño de la imagen recibida por el cuadricóptero, se debe determinar la posición del objeto en movimiento dentro de este, mismo que se establece por los ejes de referencia del dron: alabeo en inglés *roll* (eje y) y altura (eje z). Con esto la persona describe una trayectoria continua que ocupará una posición (y, z) como se muestra en la figura 33. Sin embargo, en la representación se puede notar que el cuerpo se mueve dentro de la imagen recibida en una trayectoria irregular, a causa del movimiento del sujeto, como del mismo dron al intentar seguir al primero.

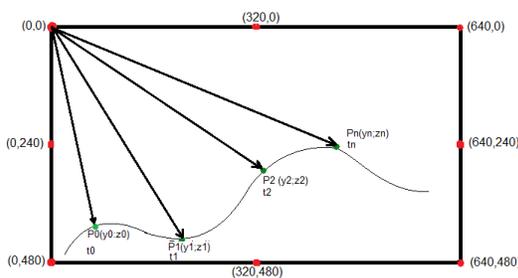


Figura 33. Posición de la persona con respecto al tiempo

4.7.1. Determinación del error de Pose

Determinada la posición del cuerpo u objeto a seguir en cualquier instante, el principal objetivo es ubicarlo y mantenerlo en el centro de la imagen (640x480), por lo que se establece un nuevo punto de centro de la imagen recibida que es $(320;240)$, donde este sirve para que el cuerpo se ubique en el centro, durante el seguimiento.

El punto centro establecido dentro de la imagen, será la referencia o set point para mantener la ubicación requerida. Por lo tanto, la diferencia entre el punto centro de la imagen y el punto centro del objeto, determinará el valor del error de posición a corregir, como se detalla en la figura 34.

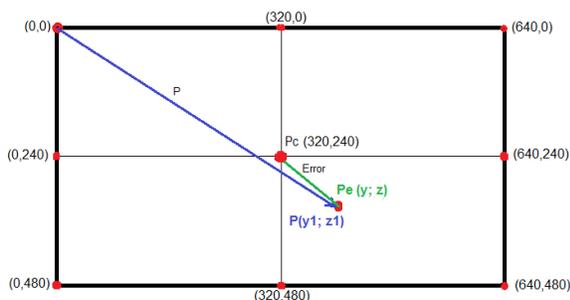


Figura 34. Punto centro y error de posición

Al entrar el sujeto en el campo de visión de la cámara, este se posicionará en un punto $(y_1; z_1)$, con respecto al origen de coordenadas (esquina superior izquierda); conocido el punto centro $P_c(320; 240)$, se determinará el error de posición de la persona respecto al centro de la cámara obteniendo la diferencia entre los dos puntos antes mencionados, como se explica en la siguiente ecuación.

$$Pe(y, z) = P(y_1, z_1) - Pc(320, 240)$$

De la expresión anterior se deduce:

Pe = Punto para cálculo del error.

P = Punto centro del sujeto.

P_c = Punto centro de la imagen

Al operar adecuadamente la ecuación indicada, el error de posición calculado presentara valores dentro del rango de ± 320 pixeles, para el caso del eje y (alabeo), en tanto que referente al eje z (altura) se tendrán valores de ± 240 pixeles, creándose automáticamente un nuevo punto de origen en el centro de la imagen, como se muestra en la figura 35.

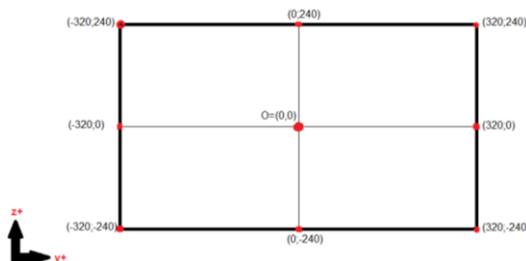


Figura 35. Punto de origen en el centro del campo de visión

Los valores ± 320 en y y ± 240 en z , serán los límites del error de posición es decir los máximos y mínimos dentro de los cuales se podrá detectar y seguir a la persona, enviando órdenes de control para el vuelo del dron.

4.7.2. Conversión error de posición a velocidad

Determinada la posición de la persona con respecto al set point o punto de origen en el centro, se debe enviar las órdenes de control para que el dron las relacione y realice los movimientos adecuados para el seguimiento. El cuadricóptero Bebob 2 para su control a través del sistema operativo ROS, usa em paquete *bebop_autonomy* que contiene un nodo llamado *bebop_drive*. Este último permite la comunicación directa con el dron, para enviar o recibir datos que se usarán como señales de control. Establecida la comunicación, se publican los comandos de vuelo mediante tópicos (temas), en este caso se usa *cmd_vel*, mismo que adquirirá valores dentro del rango de ± 1 , siendo 1 el 100%.

La conversión de los valores del error de posición a valores de velocidad, se lo realiza mediante un escalamiento, donde 320px tendrá el valor de -1 y -320px tomará el valor de 1. Al considerar la conversión mencionada de forma lineal, relacionando los valores máximos y mínimos del error de posición en el eje y o z , de acuerdo a la trayectoria que tome la persona, con los valores máximos y mínimos que puede tomar el comando de velocidad lineal (± 1), como se muestra en la figura 36.

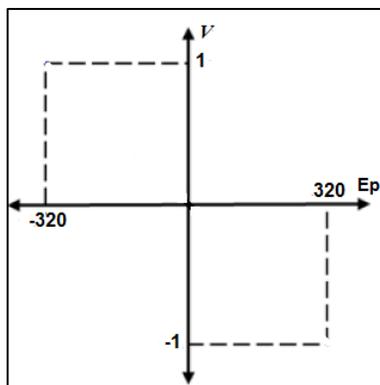


Figura 36. Relación error de posición – velocidad

En la figura 37, los límites del error de posición están entre ± 320 , perteneciente al eje de alabeo (eje Y), por lo tanto, si el sujeto se mueve dentro del campo de vista de la cámara hacia el extremo izquierdo, el valor máximo que toma es -320 , ocurriendo lo contrario si se mueve al extremo derecho donde su valor máximo será 320 . Los valores máximos que maneja el comando de velocidad del dron están determinados ± 1 , por lo tanto, al relacionarlos cuando el sujeto se encuentre al extremo izquierdo, el comando de control enviará el máximo valor, que en este caso sería 1 , para que el dron se dirija hacia la izquierda hasta que la persona se ubique cerca del centro del campo visual o set point. Por otro lado, si el sujeto se encuentra en el extremo derecho (320), se debería enviar el valor máximo -1 .

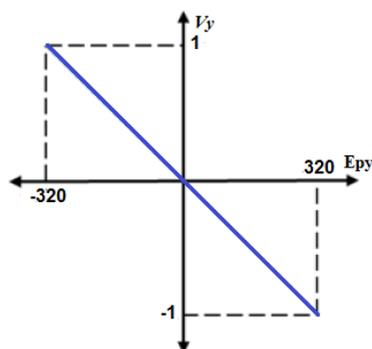


Figura 37. Grafica error de posición Y (roll) vs. velocidad

Cabe recalcar que los valores para el comando de velocidad oscilarán dentro del rango de ± 1 , de acuerdo al cambio de trayectoria y ubicación del sujeto, cuyo comportamiento, lo muestra la gráfica de la figura 37 y lo describirá la siguiente ecuación:

$$Vy = -\frac{Epy}{320}$$

Para el eje de cabeceo (eje X, profundidad), se procede a realizar otro tipo de análisis, mediante la recopilación de las áreas que se obtengan de la persona a ciertas distancias, siempre y cuando se encuentre dentro del campo de visión. Al verificar una distancia en la que el sujeto se encuentre visible y seguro para el seguimiento, se selecciona y guarda dicha área, la cual pertenece a un rectángulo que enmarca al sujeto, obteniendo de esa manera el valor máximo del área que en este caso es 60000 pixeles², es decir, cuando el sujeto esté más cercano al cuadricóptero, teniendo una distancia de 2,35 metros como mínimo, la persona con la que se realizó esta prueba tiene una altura de 1,58 metros y 0,45 metros de ancho, en tanto que al alejarse del mismo, el área se modificará, tornándose más pequeña siendo 10000 pixeles² a una distancia de 6,15 metros como máximo y de esa manera se obtendrá y guardará el valor mínimo del área que rodea a la persona, constituyendo los valores máximos y mínimos de la posición en profundidad.

Usado los valores máximos y mínimos de la posición en profundidad del sujeto, se calcula la diferencia entre estos, se divide para la mitad y sumando el valor mínimo, con lo que se puede establecer el área de referencia o también conocido como set_point, la siguiente ecuación muestra lo antes mencionado.

$$Arefx = \frac{Amáx - Amin}{2} + Amin$$

Donde:

$Arefx$ = Área de referencia para la distancia de la persona

$Amáx$ = Área máxima de detección 60000px²

A_{min} = Área máxima de detección $10000px^2$

Estimando el set point de profundidad, se requiere que el área del sujeto se aproxime a dicho valor, es decir $35000px^2$ que será una distancia de 3,15 metros, para lo cual, cuando la persona se acerque mucho a la cámara frontal y tome el valor máximo del área, se enviará el valor máximo del comando siendo -1, para indicar al dron que debe dirigirse hacia atrás. Cuando el sujeto se aleje del dron y tome el valor mínimo del área, se debe enviar el valor máximo del comando siendo 1, para comunicar que el avión no tripulado debe avanzar directamente hacia su objetivo. En la figura 38, se observa la relación de áreas y velocidad en X, es decir se describe el movimiento del sujeto en relación al eje de profundidad(X), mediante áreas, mismas que relaciona con el comando de velocidad para dicho eje.

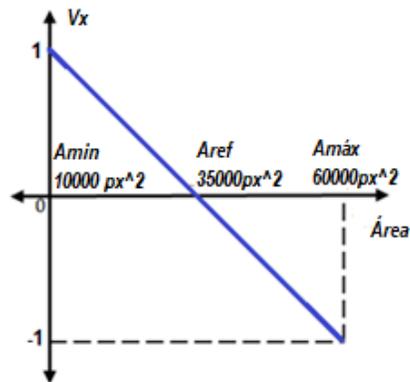


Figura 38. Grafica área (profundidad) vs. velocidad X

El envío de valores para que el dron se mueva hacia adelante o hacia atrás se lo puede obtener con la siguiente ecuación.

$$V_x = \left(1 - \frac{\text{Area}}{A_{refx}}\right) * K$$

De la ecuación anterior K una constante de compensación obtenida mediante pruebas de seguimiento, esto se lo debe hacer para que el dron pueda recibir adecuadamente los comandos para el movimiento en profundidad.

4.8. Modelado del cuadricóptero Bebop 2

Para establecer la ubicación del cuadricóptero en cualquier punto del espacio, se debe conocer la posición de éste con respecto a un marco de referencia fijo y su orientación respecto a un marco inercial, (Barrientos, Peñin, Balaguer, & Aracil, 2007). Cuando el cuadricóptero se desplaza en alguno de sus ejes, la velocidad de giro de los rotores cambia ejerciendo mayor o menor fuerza en estos para lograr el movimiento deseado, tanto en ascenso como en descenso, en cabeceo, ladeo y guiñado.

Para conocer la orientación del cuadricóptero, se debe tener los ángulos de Euler con respecto a C, siendo C el marco que nos indica la rotación del cuadricóptero en el espacio tridimensional (Barrientos, Peñin, Balaguer, & Aracil, 2007), donde: Φ representa el ángulo de balanceo o roll, este representa la rotación con respecto al eje X_b ; θ representa al ángulo de cabeceo o *pitch*, el cual es representa la rotación con respecto al eje Y_b ; Ψ representa el ángulo de guiñada o *yaw*, siendo este la rotación con respecto al eje Z_b . (Reinoso, 2014) En la figura 39 se muestra lo explicado.

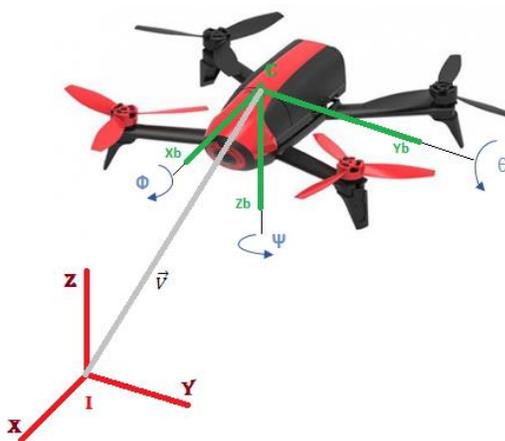


Figura 39. Marcos de referencia para el modelado

Al momento de estimar la posición u orientación del cuadricóptero, se determina con respecto al marco de referencia inercial, se realiza transformaciones al marco de referencia del Bebop 2, para poder obtenerlo con respecto al marco de referencia inercial, teniendo así las diferentes matrices de

rotación; considerando que el marco de referencia inercial y el marco de referencia del Bebop 2, coinciden en el origen y en uno de sus ejes.

En la figura 40 se puede observar la rotación de *roll* con respecto al eje X

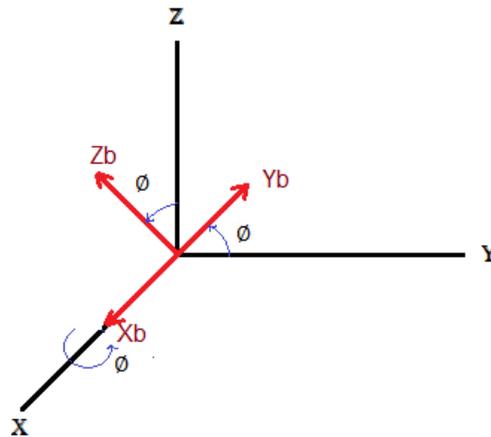


Figura 40. Rotación de roll con respecto a X

$$R(X, \phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & -\text{sen}(\phi) \\ 0 & \text{sen}(\phi) & \cos(\phi) \end{bmatrix}$$

En la figura 41 se observa la matriz de rotación de pitch con respecto al eje Y

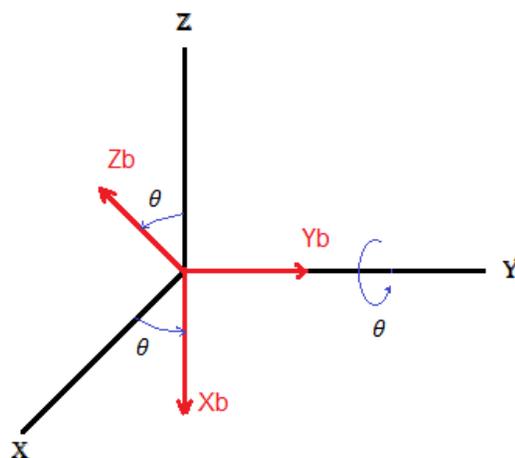


Figura 41. Rotación de pitch con respecto a Y

$$R(Y, \theta) = \begin{bmatrix} \cos(\theta) & 0 & \text{sen}(\theta) \\ 0 & 1 & 0 \\ -\text{sen}(\theta) & 0 & \cos(\theta) \end{bmatrix}$$

Se aprecia la matriz de rotación de yaw con respecto al eje Z en la figura 42

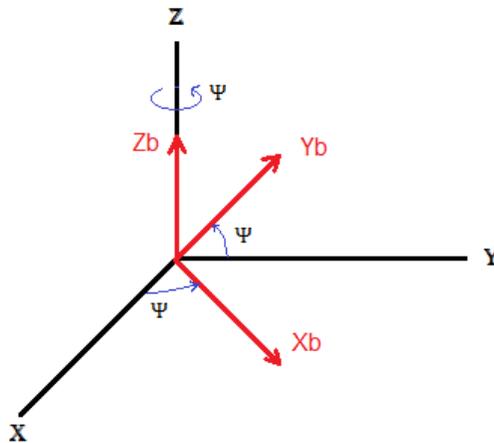


Figura 42. Rotación de yaw con respecto a Z

$$R(Z, \Psi) = \begin{bmatrix} \cos(\Psi) & -\text{sen}(\Psi) & 0 \\ \text{sen}(\Psi) & \cos(\Psi) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Para conseguir la matriz rotacional completa de C con respecto a I, se denomina matriz de cosenos directores y se obtiene mediante el producto de las tres matrices rotaciones, (Barrientos, Peñin, Balaguer, & Aracil, 2007).

$$R_I^C = R(Z, \Psi) \cdot r(Y, \theta) \cdot R(X, \phi)$$

Reemplazando las por las matrices rotacionales de cada eje tenemos:

$$R_I^C = \begin{bmatrix} \cos(\Psi) & -\text{sen}(\Psi) & 0 \\ \text{sen}(\Psi) & \cos(\Psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(\theta) & 0 & \text{sen}(\theta) \\ 0 & 1 & 0 \\ -\text{sen}(\theta) & 0 & \cos(\theta) \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & -\text{sen}(\phi) \\ 0 & \text{sen}(\phi) & \cos(\phi) \end{bmatrix}$$

Obteniendo como resultado la matriz de rotación del marco de referencia C con respecto al marco de referencia fijo I.

$$R_I^C = \begin{bmatrix} \cos \Psi \cos \theta & \cos \Psi \text{sen} \theta \text{sen} \phi - \text{sen} \Psi \cos \phi & \cos \Psi \text{sen} \theta \cos \phi + \text{sen} \Psi \text{sen} \phi \\ \text{sen} \Psi \cos \theta & \text{sen} \Psi \text{sen} \theta \text{sen} \phi - \cos \Psi \cos \phi & \text{sen} \Psi \text{sen} \theta \cos \phi + \cos \Psi \text{sen} \phi \\ -\text{sen} \theta & \cos \theta \text{sen} \phi & \cos \theta \cos \phi \end{bmatrix}$$

Una vez determinada la matriz rotacional del cuadricóptero con respecto al marco referencia fijo, se observa que este se puede dividir en dos subsistemas, el primer subsistema es la posición del cuadricóptero representado en el espacio y el segundo subsistema es la orientación de este.

El modelado matemático del cuadricóptero muestra que este se puede descomponer en cada uno de sus ejes, para poder realizar el control y su análisis respectivamente, para esto se debe basar el modelado en sus variables de estado.

$$\dot{\omega} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \omega + \begin{bmatrix} 0 \\ r \end{bmatrix} u$$

Este sistema es muy similar a un sistema de masa sin amortiguamiento, el cual se puede controlar por medio de la entrada, teniendo como resultado la función de transferencia, (Bristeau, Callou, Vissiere, & Petit, 2011). En la tabla 4 se muestran las diferentes funciones de trasferencias para cada eje del cuadricóptero.

$$G = \frac{r}{s^2}$$

Tabla 4.**Funciones de transferencia de cada eje**

Fuente: (Blasco, García, & Reynoso, 2012)

| Eje | Función de transferencia |
|---------|-----------------------------------|
| Altura | $G(s) = \frac{1}{m \cdot s^2}$ |
| Cabeceo | $G(s) = \frac{l1}{I_x \cdot s^2}$ |
| Ladeo | $G(s) = \frac{l2}{I_y \cdot s^2}$ |
| Guiñado | $G(s) = \frac{1}{I_z \cdot s^2}$ |

Los cálculos de la matriz de inercia del cuadricóptero, se los realiza considerando que este es simétrico.

$$I = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix}$$

Se necesitan fórmulas para obtener los momentos de inercia de cada uno de los ejes, (Brito Jorge., 2009).

Calculo de inercia en el eje X

$$I_{x1} = I_{x3} = \frac{1}{12} m_m (I_y^2 + I_z^2)$$

$$I_{x2} = I_{x4} = \frac{1}{12} m_m (I_y^2 + I_z^2) + m_m d_{cg}^2$$

$$I_{xx} = 2I_{x1} + 2I_{x2}$$

Calculo de inercia en el eje Y

$$I_{y2} = I_{y4} = \frac{1}{12} m_m (I_x^2 + I_z^2)$$

$$I_{y1} = I_{y3} = \frac{1}{12} m_m (I_x^2 + I_z^2) + m_m d_{cg}^2$$

$$I_{yy} = 2I_{y1} + 2I_{y2}$$

Calculo de inercia en el eje Z

$$I_{z1} = I_{z2} = I_{z3} = I_{z4} = \frac{1}{12} m_m (I_x^2 + I_y^2) + m_m d_{cg}^2$$

$$I_{zz} = 4I_{z1}$$

Deduciendo las variables:

m_m = Masa del motor

I_x = Longitud del rotor en X

I_y = Longitud del rotor en Y

I_z = Longitud del rotor en Z

d_{cg} = Distancia del motor al centro de gravedad del cuadricóptero

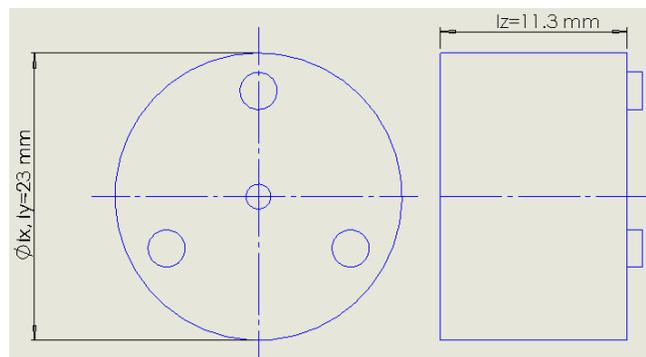


Figura 43. Longitudes del motor

Tabla 5.**Valores para el cálculo de la inercia en los diferentes ejes**

| Variables | Valores |
|--|----------------|
| Masa del Dron | 0.536 kg |
| Masa del Motor | 0.04275 Kg |
| Longitud del rotor en X | 0.023 m |
| Longitud del rotor en Y | 0.023 m |
| Longitud del rotor en Z | 0.0113 m |
| Distancia del motor al centro de gravedad del cuadricóptero | 0.143 m |

Quando se reemplaza los valores en las ecuaciones anteriores, se obtiene como resultado.

$$I_{xx} = 0.00175$$

$$I_{yy} = 0.00175$$

$$I_{zz} = 0.0035$$

Estos valores son necesarios para poder calcular las ganancias del control PID, ya que estos valores son reemplazos en las funciones de transferencia de cada eje, que se mostró en la tabla 4.

4.9. Controlador PID

Existen diferentes tipos de controladores en el mercado, pero el control PID es una de las herramientas más usadas en control automático debido a su rápida y efectiva respuesta al momento corregir o ajustar un proceso, teniendo en cuenta que su principio de funcionamiento es el reducir el error ante cualquier tipo de

perturbación. Las variables a controlar en este proyecto son los ángulos roll y pitch, (Diaz, 2017).

Este tipo de control tiene tres parámetros siendo la parte proporcional (P) la que se encarga de corregir el error actual, corrigiendo perturbaciones transitorias y no perturbaciones permanentes, por lo que se obtiene un error permanente. Usando el control integral(I), genera una corrección proporcional a la integral del error, diciendo que, si es calibrado de manera correcta, el error de seguimiento se reduce a cero. Por último, la parte derivativa (D), intenta que funcione como un predictor de los errores futuros, que ayuda a disminuir el máximo sobrepico, (Maldonado & Calderon, 2014).

La ecuación diferencial que representa a un PID en función del tiempo, está dada por:

$$u(t) = K_p e(t) + K_i \int_0^t e(t) dt + K_d \frac{d}{dt} e(t)$$

Dónde:

$u(t)$: Respuesta del controlador ante el error

$e(t)$: Error= valor medido – valor requerido

K_p : Ganancia proporcional

K_i : Ganancia integral

K_d : Ganancia derivativa

La función antes mencionada no sirve de nada si no se sabe qué tipo de planta o proceso se va a controlar, porque de acuerdo a esto se calcula las diferentes constantes para el controlador PID, siendo estas K_p , K_d , K_i . La manera más fácil de generar el controlador es por medio de una función de transferencia y luego aplicar la transformada de Laplace, obteniendo la ecuación: (Maldonado & Calderon, 2014).

$$G(s) = K_p + \frac{K_i}{s} + K_d s = \frac{K_d s^2 + K_p s + K_i}{s}$$

Cuando se implemente el control PID, es importante recordar que los valores máximos que acepta el cuadricóptero y que son manejados por el bebop_driver, oscilan entre -1 a 1, diciendo así que el control PID no puede enviar valores por encima de los ya establecidos. Por tal motivo se diseñó en la programación un limitador que no deja que se excedan estos rangos.

4.10. Controlador PID para cada eje

En la figura 44 se muestra un diagrama de bloques de cómo se realiza el control del cuadricóptero, teniendo presente que la realimentación se la hace con el procesamiento de la imagen obtenida ya que esta entrega un punto con coordenadas y un área, que son necesarias para el movimiento lateral y de profundidad del dron.

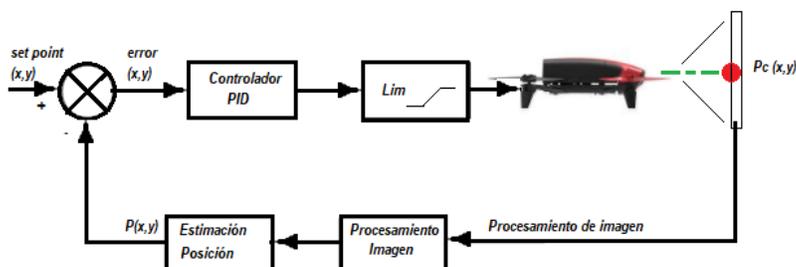


Figura 44. Diagrama de bloques de funcionamiento de control PID

Para tener una mejor respuesta en vuelo del cuadricóptero es recomendable controlar cada eje por separado, siendo el caso de roll y pitch. Un eje que se puede dejar de lado es yaw, ya que la altura a la que se debe encontrar volando, tendrá una configuración nativa, porque el objeto que se va a seguir no va a desplazarse en altura

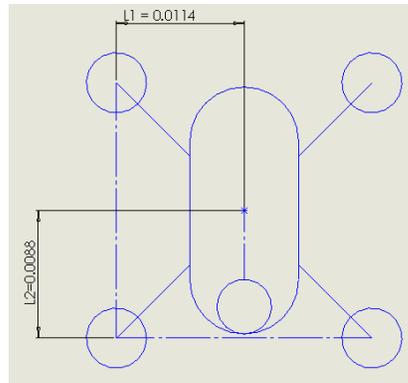


Figura 45. Distancias del centroide a X, Y

Eje de cabeceo

$$G(s) = \frac{l1}{I_x \cdot s^2} = \frac{0.114}{0.00175 \cdot s^2} f_x(s) = \frac{0.3s^2 + 0.603s + 0.006}{s}$$

Eje de ladeo

$$G(s) = \frac{l2}{I_y \cdot s^2} = \frac{0.088}{0.00175 \cdot s^2} f_y(s) = \frac{0.12 + 0.252s + 0.0024}{s}$$

En la tabla 6, se puede apreciar las ganancias de control PID.

Tabla 6

Ganancias del control PID

| Ganancias PID | Cabeceo | Ladeo |
|---------------|---------|--------|
| Kp | 0.603 | 0.252 |
| Ki | 0.006 | 0.0024 |
| Kd | 0.3 | 0.12 |

CAPITULO V

5. PRUEBAS Y RESULTADOS

5.1. Área de detección útil de la cámara estática y cámara del cuadricóptero

Para saber cuál es el área de detección óptima se realizaron pruebas con el algoritmo en ejecución y se obtuvo una referencia de distancia en profundidad lateral de 6m, a partir del eje central, y profundidad frontal de 12m, teniendo presente que las personas son detectables a partir de los 2.9m, en esta distancia a partir del eje central se tiene 0.9m obteniendo como resultado la forma de un cono, la cual va ser el área de detección óptima, esta área tiene 59 m^2 , esto se aprecia en la figura 46.

Esta área será la que la cámara estática monitoree en busca de alguna irrupción, para enviar una señal de despegue al cuadricóptero, y así empezar el seguimiento autónomo del intruso.



Figura 46. Área de detección de la cámara estática

La cámara del cuadricóptero Bebop 2, no presentó problemas al momento de realizar la conexión, ya que este cuenta con un paquete dedicado para la comunicación con ROS, pudiendo tener acceso al flujo de video del

cuadricóptero, la relación de distancias con áreas se las realizó ejecutando el algoritmo y se midió con un flexómetros, obteniendo las siguientes medidas, de profundidad 6.3m, de profundidad horizontal a partir del eje central de la cámara 4m; las personas son detectadas a partir de los 2.35 m, contando desde el centro de la cámara. A partir de esta distancia se tiene 3m, y al igual que la anterior cámara se obtiene un área de visión en forma de trapecio, donde se determina un área de detección útil, teniendo como área útil de 22 m² como se observa en la figura 47.

El área anterior fue delimitada para tener un seguimiento óptimo, restringiendo las áreas.

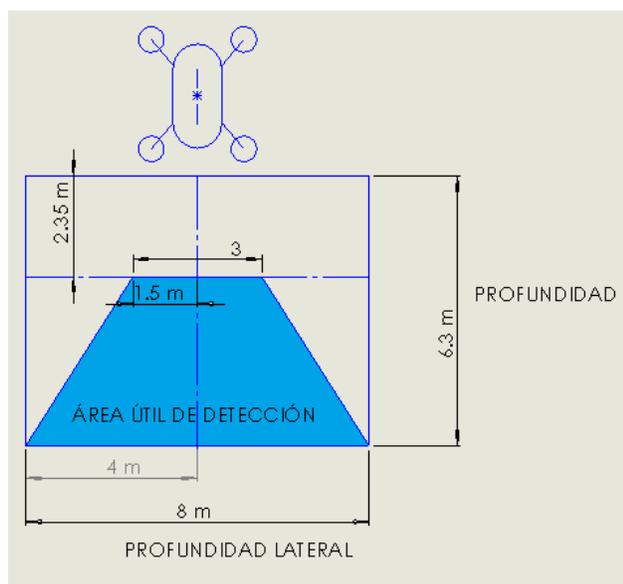


Figura 47. Área de detección útil de cuadricóptero

En la figura 48 se puede observar la perspectiva del campo visual que se obtiene desde la cámara del cuadricóptero; mientras el objeto más cerca este, el área será mayor y se apreciará lo contrario cuando este se encuentre más alejado. En la tabla 7, se puede apreciar una relación de áreas con respecto a la distancia y el envío de valores para el control del cuadricóptero.

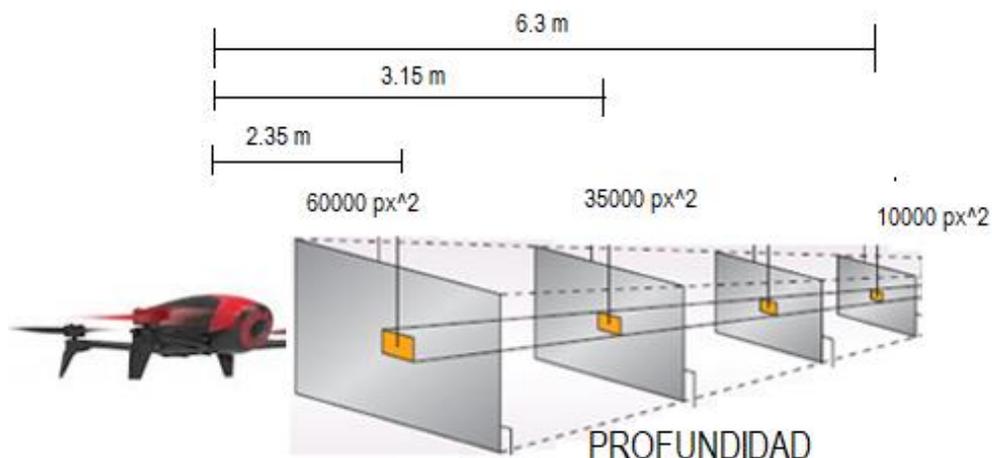


Figura 48. Campo visual de detección

En la tabla 7 se muestran los datos recopilados con la cámara incorporada del dron, para poder identificar el área de detección útil que posteriormente se utilizará en el algoritmo de detección y seguimiento. Estas pruebas se las realizaron con una persona que mide 1,58 metros de altura y 0,45 metros de ancho.

Tabla 7.

Área de detección en base a la profundidad del sujeto

| Altura de vuelo (m) | Área de detección (Píxeles²) | Distancia en profundidad (m) |
|----------------------------|--|-------------------------------------|
| 0,95 | 60000 | 2,35 |
| | 35000 | 3,15 |
| | 10000 | 6,3 |

Una vez iniciado el vuelo del dron, este permanecerá a una altura de 0,95m aproximadamente. Una vez restringida la altura en el eje z, se recopilan datos del área de un rectángulo que rodea al sujeto, proporcionados por el algoritmo de procesamiento de imagen para la detección de personas, es así que el área máxima que el algoritmo detecta es 60000 píxeles², a una distancia en profundidad (eje X) de 2,35 m medidos desde el lente de la cámara del dron, como se muestra en la figura 47.

Por seguridad del sujeto se consideró una distancia en profundidad de 3m con respecto al dron para iniciar el seguimiento hacia adelante con un área de 40000 pixeles². Por otro lado, el área de detección mínima que se ha estimado es de 10000 pixeles² a 6.3m teniendo una distancia de desplazamiento de 3,3 m en línea recta, misma que resulta efectiva para el seguimiento.

Determinadas las áreas de detección de personas máxima y mínima, se procede a determinar los valores que cada área proporciona para enviar el comando de velocidad del dron y este a su vez proceda con las señales de control de vuelo para avanzar o retroceder.

Como se muestra en la siguiente tabla 8, el área de detección de 35000 px² se tomó como **set point** o punto de referencia. Los valores máximo y mínimo de 10000px² y 60000px², respectivamente constituirán los límites del comando de velocidad (± 1). Teniendo presente que 1 es el 100% de la velocidad de vuelo configurada (1m/s).

Tabla 8.

Escalonamiento con respecto al área

| Área de detección en pixeles ² | Valor driver cmd_x; cmd_y | Velocidad (m/s) | Acciones de control |
|---|---------------------------|-----------------|---------------------|
| 10000 | 0,96 | 0,96 | Adelante |
| 20000 | 0,57 | 0,57 | Adelante |
| 30000 | 0,19 | 0,19 | Adelante |
| 35000 | 0 | 0 | |
| 40000 | - 0,19 | - 0,19 | Atrás |
| 50000 | - 0,57 | - 0,57 | Atrás |
| 60000 | - 0,96 | - 0,96 | Atrás |

5.2. Evaluación del Algoritmo de detección usado para la cámara estática

En la tabla 9 se muestra las diferentes distancias en las que se evalúa la detección de personas con la cámara estática, teniendo presente que se marca con un +, si esta detecta a las personas, caso contrario se marcará con un -, y con * diciendo que no existían personas en ese rango. En la figura 49 se puede

apreciar una muestra de las imágenes que se tomaron con la cámara estática. Con esta se realizó la validación del algoritmo de detección de personas.



Figura 49. Muestra de imágenes cámara estática

Tabla 9.

Detección de personas según la profundidad

| N° Pruebas | Rango de detección de personas en profundidad (metros) | | | |
|---------------|---|-------|-------|-------|
| | =3 | 3 a 5 | 5 a 7 | 7 a 9 |
| 1 | + | + | + | + |
| 2 | * | + | + | + |
| 3 | + | - | - | - |
| 4 | * | + | + | + |
| 5 | * | + | + | - |
| 6 | + | + | + | + |
| 7 | * | + | + | - |
| 8 | * | + | + | + |
| 9 | + | + | + | + |
| 10 | + | + | + | + |
| 11 | * | + | + | - |
| 12 | * | + | + | - |
| 13 | * | + | + | + |
| 14 | * | + | + | + |
| 15 | - | + | + | + |
| 16 | + | + | + | + |
| 17 | + | + | + | + |

CONTINUA 

| | | | | |
|---------------------|---------------|------------|------------|------------|
| 18 | + | + | + | + |
| 19 | + | + | + | + |
| 20 | + | + | + | - |
| 21 | + | + | + | + |
| 22 | + | + | - | - |
| 23 | * | + | + | + |
| 24 | * | + | + | - |
| 25 | - | + | + | - |
| 26 | + | - | + | - |
| 27 | - | - | + | - |
| 28 | + | + | + | + |
| 29 | + | - | + | + |
| 30 | + | + | + | + |
| 31 | + | + | + | + |
| 32 | * | + | + | + |
| 33 | + | + | + | + |
| 34 | + | + | + | + |
| 35 | + | + | - | + |
| 36 | + | + | - | + |
| 37 | + | + | - | + |
| 38 | + | + | + | + |
| 39 | - | - | + | + |
| 40 | - | + | + | - |
| 41 | - | - | + | + |
| 42 | + | + | + | + |
| 43 | + | + | + | + |
| 44 | + | + | + | - |
| 45 | + | + | + | + |
| 46 | + | + | + | + |
| 47 | + | + | + | + |
| 48 | + | + | + | + |
| 49 | + | + | + | + |
| 50 | * | + | + | + |
| T. Positivos | 32 | 44 | 45 | 37 |
| T. Negativos | 5 | 6 | 5 | 13 |
| Porcentaje | 86,46% | 88% | 90% | 74% |

De la tabla anterior se tiene como resultados, que a la distancia de 3m se obtiene el 86,46 % de efectividad, en la distancia de 5m se tiene un porcentaje de 88%, a los 7 m se obtuvo el 90% y a los 9m se obtuvo el 74%. Se puede observar que en el rango mayor a 9 m existe un descenso en la detección de imágenes positivas, debido a la deformación de las siluetas de las personas y

también a la calidad del sensor de la cámara. En la figura 50 se muestra las diferencias de la detección en las distancias antes mencionadas.

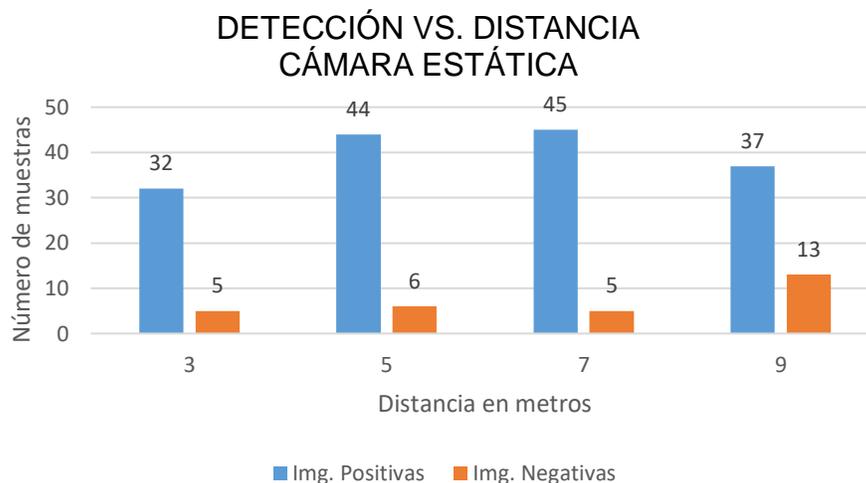


Figura 50. Detección vs distancia cámara estática

La tabla 10 muestra la cantidad de personas detectadas, clasificándolas por detección positiva, negativa o falsos y el porcentaje de personas detectadas en cada imagen.

Tabla 10.

Detección de personas por algoritmo

| N° Pruebas | N° Personas | Detección de personas por algoritmo | | | Porcentaje % |
|------------|-------------|-------------------------------------|------------------------|------------------|--------------|
| | | Personas detectadas | Personas no detectadas | Falsos positivos | |
| 1 | 7 | 5 | 2 | 1 | 71,42 |
| 2 | 8 | 6 | 2 | 0 | 75 |
| 3 | 3 | 2 | 1 | 0 | 66,66 |
| 4 | 7 | 7 | 0 | 0 | 100 |
| 5 | 7 | 6 | 1 | 0 | 85,71 |
| 6 | 6 | 6 | 0 | 0 | 100 |
| 7 | 11 | 10 | 1 | 0 | 90,90 |
| 8 | 9 | 8 | 1 | 1 | 88,88 |
| 9 | 12 | 11 | 1 | 1 | 91,66 |
| 10 | 5 | 5 | 0 | 2 | 100 |
| 11 | 8 | 7 | 1 | 2 | 87,5 |

CONTINUA



| | | | | | |
|--------------|------------|------------|-----------|-----------|--------------|
| 12 | 9 | 9 | 0 | 0 | 100 |
| 13 | 8 | 5 | 3 | 0 | 62,5 |
| 14 | 10 | 10 | 0 | 3 | 100 |
| 15 | 11 | 6 | 5 | 4 | 45,45 |
| 16 | 5 | 4 | 1 | 1 | 80 |
| 17 | 15 | 11 | 4 | 1 | 73,33 |
| 18 | 6 | 6 | 0 | 0 | 100 |
| 19 | 8 | 7 | 1 | 0 | 87,5 |
| 20 | 8 | 8 | 0 | 0 | 100 |
| 21 | 11 | 11 | 0 | 3 | 100 |
| 22 | 9 | 8 | 1 | 2 | 88,88 |
| 23 | 5 | 5 | 0 | 0 | 100 |
| 24 | 10 | 10 | 0 | 4 | 100 |
| 25 | 12 | 11 | 1 | 0 | 91,66 |
| 26 | 6 | 6 | 0 | 1 | 100 |
| 27 | 2 | 2 | 0 | 5 | 100 |
| 28 | 8 | 7 | 1 | 2 | 87,5 |
| 29 | 9 | 8 | 1 | 0 | 88,88 |
| 30 | 7 | 7 | 0 | 0 | 100 |
| 31 | 7 | 7 | 0 | 1 | 100 |
| 32 | 11 | 11 | 0 | 1 | 100 |
| 33 | 10 | 9 | 1 | 1 | 90 |
| 34 | 9 | 8 | 1 | 2 | 88,88 |
| 35 | 13 | 12 | 1 | 3 | 92,30 |
| 36 | 7 | 7 | 0 | 2 | 100 |
| 37 | 7 | 6 | 1 | 1 | 85,71 |
| 38 | 6 | 6 | 0 | 1 | 100 |
| 39 | 9 | 8 | 1 | 3 | 88,88 |
| 40 | 9 | 7 | 2 | 0 | 77,77 |
| 41 | 8 | 7 | 1 | 0 | 87,5 |
| 42 | 8 | 8 | 0 | 0 | 100 |
| 43 | 6 | 6 | 0 | 0 | 100 |
| 44 | 6 | 6 | 0 | 1 | 100 |
| 45 | 11 | 10 | 1 | 3 | 90,90 |
| 46 | 10 | 8 | 2 | 1 | 80 |
| 47 | 9 | 8 | 1 | 1 | 88,88 |
| 48 | 7 | 7 | 0 | 2 | 100 |
| 49 | 7 | 4 | 3 | 2 | 57,14 |
| 50 | 9 | 8 | 1 | 5 | 88,88 |
| Total | 411 | 367 | 44 | 63 | 89,29 |

La figura 51, muestra el porcentaje de detecciones positivas que en este caso son el 89.29%, y el 10.71% de personas no detectadas, haciendo que este algoritmo sea eficiente para la detección de personas. Cabe recalcar que este proyecto tiene como objetivo detectar la irrupción de personas en un espacio

delimitado. Con esto se demuestra que el sistema puede ser utilizado para la vigilancia en espacios delimitados, teniendo presente que las pruebas se realizaron durante el día.

EVALUACIÓN DEL ALGORITMO DE DETECCIÓN CON LA CÁMARA ESTÁTICA

■ Personas detectadas ■ Personas no detectadas

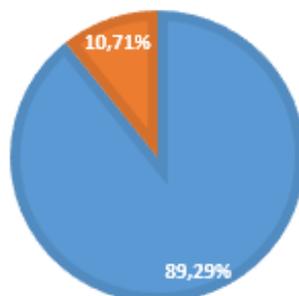


Figura 51. Evaluación del algoritmo de detección

El cuadricóptero mostrará imágenes dentro de las cuales es necesario delimitar la distancia dentro de la cual la detección es óptima. De esta manera lograr datos libres de errores y tomar las respectivas medidas de control de vuelo del dron, por lo tanto, se procede como se indica en la tabla 11, seleccionando imágenes al azar de un total de 50, como se observa en la figura 52.

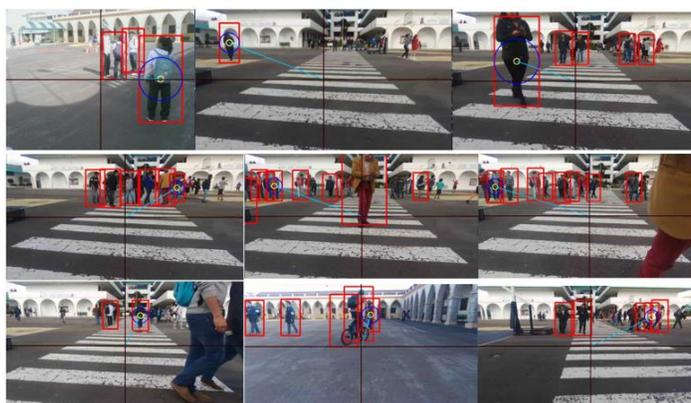


Figura 52. Muestra de imágenes cámara cuadricóptero

Tabla 11.

Delimitación de distancias para una buena detección

| N° Pruebas | Distancias de personas en profundidad | | | | |
|---------------|---------------------------------------|-------|-------|-------|-------|
| | d<=2m | d= 3m | d= 5m | d= 6m | d>=7m |
| 1 | * | * | + | + | - |
| 2 | * | + | * | + | - |
| 3 | * | * | * | + | - |
| 4 | * | * | + | * | - |
| 5 | * | + | + | + | - |
| 6 | * | * | + | + | - |
| 7 | * | + | + | + | - |
| 8 | - | * | + | + | - |
| 9 | - | * | + | + | - |
| 10 | * | + | + | + | - |
| 11 | * | + | + | + | - |
| 12 | * | + | + | + | + |
| 13 | * | + | + | + | + |
| 14 | * | + | + | + | * |
| 15 | * | * | + | + | * |
| 16 | + | * | + | + | + |
| 17 | + | * | + | + | + |
| 18 | + | + | + | + | + |
| 19 | + | * | + | + | + |
| 20 | + | * | + | + | + |
| 21 | + | + | * | + | + |
| 22 | * | * | * | + | * |
| 23 | * | + | + | + | * |
| 24 | * | + | + | + | + |
| 25 | * | * | + | + | - |
| 26 | * | + | + | + | - |
| 27 | * | - | + | + | - |
| 28 | * | + | + | + | - |
| 29 | * | + | + | + | + |
| 30 | * | + | + | + | - |
| 31 | * | + | + | - | - |
| 32 | * | + | + | + | - |
| 33 | * | * | + | + | - |
| 34 | * | + | + | + | - |
| 35 | * | + | + | + | - |
| 36 | * | + | * | + | + |
| 37 | * | + | + | + | - |
| 38 | + | * | + | + | - |
| 40 | + | + | + | - | - |
| 41 | + | + | + | + | - |
| 42 | + | + | + | + | - |
| 43 | * | + | + | + | - |

CONTINUA 

| | | | | | |
|---------------------|----------|-----------|-----------|-----------|-----------|
| 44 | * | + | + | + | - |
| 45 | * | + | - | + | + |
| 46 | * | + | - | - | - |
| 47 | * | + | * | + | - |
| 48 | * | - | + | + | + |
| 49 | * | + | * | + | + |
| 50 | * | + | + | - | + |
| T. Positivos | 9 | 32 | 41 | 45 | 15 |
| T. Negativos | 2 | 3 | 2 | 4 | 31 |

Al evaluar las distancias medidas desde el frente del dron, con referencia a distancias marcadas en el suelo del patio de la Universidad de las Fuerzas Armadas ESPE-L, se obtuvo la gráfica que se observa en la figura 53.

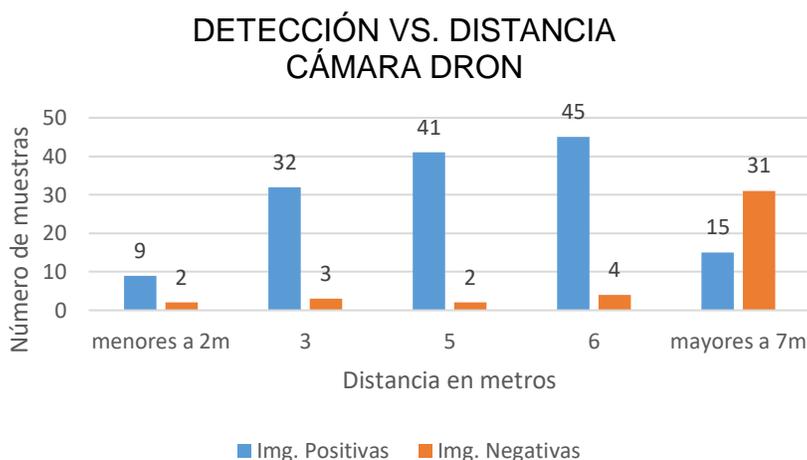


Figura 53. Delimitación de distancias en profundidad

Mediante la gráfica resultante de las tabulaciones, se indica que las personas que se encontraban a distancias menores de 2 metros y mayores a 7 metros no han podido ser detectadas debido a que al estar demasiado cerca o muy lejos la forma que define a una persona tiende a perderse o distorsionarse, por lo tanto, no se ha logrado obtener detecciones.

Por otro lado, dentro del rango de 3 a 6 metros se ha logrado obtener el mayor número de detecciones de manera eficiente y correcta, ya que a partir de estas distancias se puede enfocar y definir totalmente la forma de humano. Cabe

recaltar que los datos fueron tomados mientras el dron Bebop 2, se encontraba en vuelo a una altura constante de 0,95 m aproximadamente, permaneciendo en un punto, sin avanzar o retroceder.

5.3. Evaluación del Algoritmo de detección dentro de las distancias delimitadas, para el seguimiento del cuadricóptero.

Ya que las distancias en línea recta del sujeto con respecto al dron se han definido, se procede a verificar la eficiencia del algoritmo al momento de detectar cuando la persona está dentro de estos rangos de distancia. A continuación, la tabla 13, muestra el número de personas que se encuentran dentro del rango de distancia útil para la detección, en tanto que las que están fuera de dicho rango no se las toma en cuenta.

Tabla 12.

Evaluación del algoritmo detector

| N° Pruebas | No. Personas | Detección de personas por algoritmo | | | Porcentaje % |
|---------------|-----------------|-------------------------------------|---------------------------|---------------------|---|
| | | Personas detectadas | Personas no detectadas | Falsos negativos | |
| 1 | 8 | 5 | 3 | 0 | 62,5 |
| 2 | 9 | 6 | 3 | 1 | 66,66 |
| 3 | 1 | 1 | 0 | 0 | 100 |
| 4 | 1 | 1 | 0 | 0 | 100 |
| 5 | 8 | 6 | 2 | 0 | 75 |
| 6 | 9 | 8 | 1 | 0 | 88,88 |
| 7 | 7 | 5 | 2 | 1 | 71,42 |
| 8 | 9 | 7 | 2 | 0 | 77,77 |
| 9 | 4 | 2 | 2 | 0 | 50 |
| 10 | 2 | 2 | 0 | 1 | 100 |
| 11 | 8 | 7 | 1 | 0 | 87,5 |
| 12 | 4 | 3 | 1 | 2 | 75 |
| 13 | 6 | 6 | 0 | 1 | 100 |
| 14 | 8 | 5 | 3 | 1 | 62,5 |
| 15 | 3 | 3 | 0 | 0 | 100 |
| | | | | CONTINUA |  |
| 16 | 5 | 4 | 1 | 0 | 80 |
| 17 | 8 | 5 | 3 | 0 | 62,5 |
| 18 | 5 | 5 | 0 | 2 | 100 |
| 19 | 6 | 6 | 0 | 0 | 100 |
| 20 | 8 | 6 | 2 | 2 | 75 |

| | | | | | |
|--------------|------------|------------|-----------|-----------|--------------|
| 21 | 9 | 8 | 1 | 1 | 88,88 |
| 22 | 7 | 7 | 0 | 2 | 100 |
| 23 | 6 | 6 | 0 | 0 | 100 |
| 24 | 7 | 5 | 2 | 0 | 71,42 |
| 25 | 10 | 8 | 2 | 0 | 80 |
| 26 | 1 | 1 | 0 | 0 | 100 |
| 27 | 5 | 4 | 1 | 0 | 80 |
| 28 | 10 | 10 | 0 | 1 | 100 |
| 29 | 1 | 1 | 0 | 2 | 100 |
| 30 | 5 | 4 | 1 | 1 | 80 |
| 31 | 6 | 6 | 0 | 0 | 100 |
| 32 | 3 | 3 | 0 | 0 | 100 |
| 33 | 4 | 4 | 0 | 3 | 100 |
| 34 | 9 | 9 | 0 | 0 | 100 |
| 35 | 7 | 6 | 1 | 1 | 85,71 |
| 36 | 8 | 6 | 2 | 1 | 75 |
| 37 | 8 | 7 | 1 | 2 | 87,5 |
| 38 | 7 | 7 | 0 | 1 | 100 |
| 39 | 6 | 4 | 2 | 1 | 66,66 |
| 40 | 3 | 2 | 1 | 0 | 66,6 |
| 41 | 3 | 3 | 0 | 0 | 100 |
| 42 | 4 | 4 | 0 | 0 | 100 |
| 43 | 9 | 9 | 0 | 2 | 100 |
| 44 | 8 | 8 | 0 | 0 | 100 |
| 45 | 9 | 9 | 0 | 1 | 100 |
| 46 | 9 | 7 | 2 | 2 | 77,77 |
| 47 | 7 | 6 | 1 | 1 | 85,71 |
| 48 | 6 | 6 | 0 | 1 | 100 |
| 49 | 4 | 3 | 1 | 0 | 75 |
| 50 | 5 | 5 | 0 | 0 | 100 |
| Total | 310 | 261 | 49 | 34 | 87,09 |

Las personas dentro del rango de distancia delimitado, fueron detectadas en su mayoría. En la figura 54, se observa esta información.

EVALUACIÓN DEL ALGORITMO DE DETECCIÓN CON LA CÁMARA DRON

■ Personas detectadas ■ Personas no detectadas

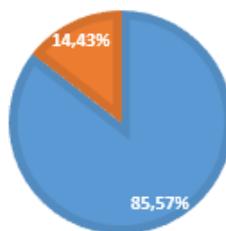


Figura 54. Evaluación del Algoritmo con la cámara del Dron en vuelo

La detección de las personas dentro del rango de distancia de 3 a 6 metros, se realizó de manera adecuada, obteniendo un 85.57% de detecciones correctas, frente a un 14.43% de detecciones que no se realizaron adecuadamente. Sin embargo, se considera que ciertas imágenes contienen personas en grupo por lo que la forma humana se distorsiona, provocando el porcentaje de no detecciones, pero considerando que este proyecto se aplica para la custodia de áreas con poca o ninguna concurrencia de personas, en las que no se tendrá grupos aglomerados, el 14.43% se considera un error bajo, ante la correcta detección de personas que se encuentran individualmente o dispersas dentro del área sin aglomerarse entre sí.

Entonces se puede determinar que el algoritmo es adecuado para la detección, dado su correcto funcionamiento dentro de la distancia delimitada y su robustez ante diferentes horas del día y cambios de luz, que no afectan en lo absoluto a su función obteniendo un relativamente bajo 3% de detecciones de falsos positivos.

5.3.1. Comprobación de hipótesis de detección

- **Hipótesis nula**

Ho= No mejora la seguridad mediante la detección de personas

- **Hipótesis valida**

Hi= Mejora la seguridad mediante la detección de personas

Margen de error considerado = 5%

Margen de error tabla= 0.05

Grados de libertad = 2

$$GDL = (n^{\circ}filas - 1) \times (n^{\circ}columnas - 1)$$

Tabla 13.

Datos para calcular chi-cuadrado

| | Detectadas | No detectadas | Falsos | Total |
|-----------------------------|-------------------|----------------------|---------------|--------------|
| Cámara fija | 367 | 44 | 63 | 474 |
| Cámara Cuadricóptero | 261 | 49 | 34 | 344 |
| Total | 628 | 93 | 97 | 818 |

Tabla 14.

Cálculo de frecuencia teórica

| n | f | ft | n | f | ft |
|----------|------------|-----------|----------|-----------|-----------|
| 0 | 367 | 363.9 | 3 | 49 | 39.11 |
| 1 | 261 | 264.09 | 4 | 63 | 56.20 |
| 2 | 44 | 53.88 | 5 | 34 | 40.79 |

De la tabla 14:

f: frecuencia calculada

ft: frecuencia teórica

$$x^2 = \sum_{i=0}^n \frac{(f - ft)^2}{ft}$$

Donde:

$$n = (nfilas * ncolumnas) - 1$$

$$x^2 = \sum_{i=0}^{n=5} \frac{(f - ft)^2}{ft}$$

$$xcalc^2 = 6.323$$

Al calcular el chi-cuadrado se obtuvo un resultado de 6.323, para ser comparado con el chi-cuadrado de la tabla ($xtbl^2$).

| v/p | 0,001 | 0,0025 | 0,005 | 0,01 | 0,025 | 0,05 | 0,1 | 0,15 |
|-----|---------|---------|---------|---------|---------|---------|---------|---------|
| 1 | 10,8274 | 9,1404 | 7,8794 | 6,6349 | 5,0239 | 3,8415 | 2,7055 | 2,0722 |
| 2 | 13,8150 | 11,9827 | 10,5965 | 9,2104 | 7,3778 | 5,9915 | 4,6052 | 3,7942 |
| 3 | 16,2660 | 14,3202 | 12,8381 | 11,3449 | 9,3484 | 7,8147 | 6,2514 | 5,3170 |
| 4 | 18,4662 | 16,4238 | 14,8602 | 13,2767 | 11,1433 | 9,4877 | 7,7794 | 6,7449 |
| 5 | 20,5147 | 18,3854 | 16,7496 | 15,0863 | 12,8325 | 11,0705 | 9,2363 | 8,1152 |
| 6 | 22,4575 | 20,2491 | 18,5475 | 16,8119 | 14,4494 | 12,5916 | 10,6446 | 9,4461 |
| 7 | 24,3213 | 22,0402 | 20,2777 | 18,4753 | 16,0128 | 14,0671 | 12,0170 | 10,7479 |
| 8 | 26,1239 | 23,7742 | 21,9549 | 20,0902 | 17,5345 | 15,5073 | 13,3616 | 12,0271 |

Figura 55. Tabla chi-cuadrado 2 GDL error 5%

$$xtbl^2 = 5.991$$

$$xcalc^2 > xtbl^2 = Hi \text{ es válida}$$

$$xcalc^2 < xtbl^2 = Ho \text{ es válida}$$

$$6.323 > 5.991 = Hi \text{ es válida}$$

La hipótesis fue verificada por medio de distribución chi-cuadrado, teniendo en cuenta un margen de error del 5%, este porcentaje se tomó debido a las variaciones de luz, procesamiento de los computadores y la distorsión de la persona a distancias lejanas.

5.4. Pruebas de seguimiento

Para la validación del algoritmo de seguimiento y control del vuelo autónomo del cuadricóptero, se procede a recopilar datos en una trayectoria de 40 m. La longitud será marcada en intervalos 5m para determinar la distancia recorrida, iniciando el vuelo en el punto de partida del dron (0m) y continua a lo largo, como se muestra en la figura 56.

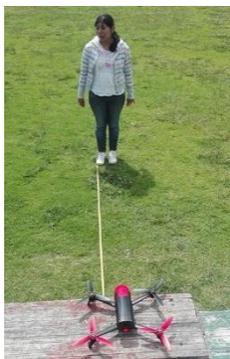


Figura 56. Medición de intervalos de distancia

Tabla 15

Validación del seguimiento de 0-40m

| N° Pruebas | 5m | 10m | 15m | 20m | 25m | 30m | 35m | 40m | Porcentaje seguimiento |
|------------|----|-----|-----|-----|-----|-----|-----|-----|------------------------|
| 1 | + | + | + | + | - | - | - | - | 50% |
| 2 | + | + | + | - | - | - | - | - | 37.5% |
| 3 | + | + | + | + | + | - | - | - | 62.5% |
| 4 | + | + | + | + | + | + | + | + | 100% |
| 5 | + | + | + | + | + | + | + | + | 100% |
| 6 | + | + | + | + | + | + | + | + | 100% |
| 7 | + | + | + | + | + | + | + | + | 100% |
| 8 | + | + | + | + | + | + | + | + | 100% |
| 9 | + | + | + | + | + | + | + | + | 100% |
| 10 | + | + | + | + | + | + | + | + | 100% |
| 11 | + | + | + | + | + | + | + | + | 100% |
| 12 | + | + | + | + | + | + | + | + | 100% |
| 13 | + | + | + | + | + | + | + | + | 100% |
| 14 | + | + | + | + | + | + | + | + | 100% |
| 15 | + | + | + | + | + | + | + | + | 100% |
| 16 | + | + | + | + | + | + | + | + | 100% |
| 17 | + | + | + | + | + | + | + | + | 100% |
| 18 | + | + | + | + | + | + | + | + | 100% |
| 19 | + | + | + | + | + | + | + | + | 100% |
| 20 | + | + | + | + | + | + | + | + | 100% |
| 21 | + | + | + | + | + | + | + | - | 87.5% |
| 22 | + | + | + | + | + | + | - | - | 75% |
| 23 | + | + | + | + | + | - | - | - | 62.5% |
| 24 | + | + | + | + | + | + | + | + | 100% |
| 25 | + | + | + | - | - | - | - | - | 37.5% |
| 26 | + | + | + | + | - | - | - | - | 50% |
| 27 | + | + | + | - | - | - | - | - | 37.5% |
| 28 | + | + | + | - | - | - | - | - | 37.5% |
| 29 | + | + | + | + | + | + | + | + | 100% |

CONTINUA



| | | | | | | | | | |
|----------------|-------------|-------------|-------------|------------|------------|------------|------------|------------|---------------|
| 30 | + | + | + | + | + | + | - | - | 75% |
| 31 | + | + | + | + | + | + | + | + | 100% |
| 32 | + | + | + | + | + | + | + | + | 100% |
| 33 | + | + | + | + | + | - | - | - | 62.5% |
| 34 | + | + | + | + | + | + | - | - | 75% |
| 35 | + | + | + | + | + | + | + | + | 100% |
| 36 | + | + | + | + | + | + | + | + | 100% |
| 37 | + | + | + | + | + | + | + | - | 87.5% |
| 38 | + | + | + | + | + | + | + | + | 100% |
| 39 | + | + | + | + | + | + | + | + | 100% |
| 40 | + | + | + | + | + | + | + | - | 87.5% |
| 41 | + | + | + | + | + | + | + | + | 100% |
| 42 | + | + | + | + | + | + | + | + | 100% |
| 43 | + | + | + | + | + | + | + | + | 100% |
| 44 | + | + | + | + | + | - | - | - | 62.5% |
| 45 | + | + | + | + | + | + | + | + | 100% |
| 46 | + | + | + | + | - | - | - | - | 50% |
| 47 | + | + | + | + | + | + | + | + | 100% |
| 48 | + | + | + | + | + | + | + | + | 100% |
| 49 | + | + | + | + | + | + | + | + | 100% |
| 50 | + | + | + | + | + | + | + | + | 100% |
| P.Total | | | | | | | | | 87% |
| P. | | | | | | | | | |
| Tramos | 100% | 100% | 100% | 92% | 86% | 78% | 72% | 66% | 86.75% |

Realizando un promedio de 50 pruebas del seguimiento del dron se determinó una efectividad del mismo en un porcentaje del 87%, teniendo en cuenta que condiciones como la velocidad del viento, afectan el vuelo de dron y por ende el seguimiento.

También se verificó la efectividad del seguimiento en tramos, teniendo como resultado un porcentaje general de 86.75%, esto es debido al cambio de velocidades en los últimos tramos, se puede apreciar en la tabla 18.

5.4.1. Comprobación de hipótesis de seguimiento

- **Hipótesis nula**

Ho= No existe seguimiento de personas

- **Hipótesis valida**

Hi= Existe seguimiento de personas

Margen de error considerada= 5%

Margen de error tabla= 0.05

Grados de libertad = 7

Tabla 16.

Datos para calcular chi-cuadrado

| | 5m | 10m | 15m | 20m | 25m | 30m | 35m | 40m | Total |
|-----------------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|------------|
| Positivo | 50 | 50 | 50 | 46 | 43 | 39 | 36 | 33 | 347 |
| Negativo | 0 | 0 | 0 | 4 | 7 | 11 | 14 | 17 | 53 |
| Total | 50 | 400 |

$$x^2 = \sum \frac{(f - ft)^2}{ft}$$

$$x_{calc}^2 = 59.72$$

Del cálculo del chi-cuadrado se obtuvo un resultado de 59.72, dicho valor será comparado con el chi-cuadrado de la tabla (x_{tbl}^2), que se muestra en la figura 56.

| v/p | 0,001 | 0,0025 | 0,005 | 0,01 | 0,025 | 0,05 | 0,1 | 0,15 |
|-----|---------|---------|---------|---------|---------|---------|---------|---------|
| 1 | 10,8274 | 9,1404 | 7,8794 | 6,6349 | 5,0239 | 3,8415 | 2,7055 | 2,0722 |
| 2 | 13,8150 | 11,9827 | 10,5965 | 9,2104 | 7,3778 | 5,9915 | 4,6052 | 3,7942 |
| 3 | 16,2660 | 14,3202 | 12,8381 | 11,3449 | 9,3484 | 7,8147 | 6,2514 | 5,3170 |
| 4 | 18,4662 | 16,4238 | 14,8602 | 13,2767 | 11,1433 | 9,4877 | 7,7794 | 6,7449 |
| 5 | 20,5147 | 18,3854 | 16,7496 | 15,0863 | 12,8325 | 11,0705 | 9,2363 | 8,1152 |
| 6 | 22,4575 | 20,2491 | 18,5475 | 16,8119 | 14,4494 | 12,5916 | 10,6446 | 9,4461 |
| 7 | 24,3213 | 22,0402 | 20,2777 | 18,4753 | 16,0128 | 14,0671 | 12,0170 | 10,7479 |
| 8 | 26,1239 | 23,7742 | 21,9549 | 20,0902 | 17,5345 | 15,5073 | 13,3616 | 12,0271 |

Figura 57. Tabla Chi- cuadrado 7 GDL error 5%

$$x_{tbl}^2 = 14.0671$$

$$x_{calc}^2 > x_{tbl}^2 = H_i \text{ es valida}$$

$$x_{calc}^2 < x_{tbl}^2 = H_o \text{ es valida}$$

$$59.72 > 14.0671 = H_i \text{ es valida}$$

Se realizó el cálculo del chi-cuadrado para determinar la validez de la hipótesis, teniendo como resultado el cumplimiento del seguimiento de personas mediante un dron.

5.5. Pruebas de velocidad de la persona en seguimiento.

Cumplida la condición de seguimiento con el dron, es importante establecer la velocidad de desplazamiento de una persona mientras es seguida por el dron en una distancia de 40 m. En la tabla 17 se establecen tiempos obtenidos en varias pruebas, que permitirá obtener la velocidad del sujeto mediante la ecuación que se muestra a continuación.

$$Velocidad = \frac{distancia}{tiempo}$$

Tabla 17.

Velocidad de seguimiento de una persona en una longitud de 40m.

| N° Pruebas | Tiempo (s) | Velocidad (m/s) |
|-------------------|-------------------|------------------------|
| 1 | 52,89 | 0,756 |
| 2 | 50,2 | 0,797 |
| 3 | 54,98 | 0,728 |
| 4 | 47,42 | 0,844 |
| 5 | 53,64 | 0,746 |
| 6 | 45,26 | 0,884 |
| 7 | 48,56 | 0,824 |
| 8 | 47,87 | 0,836 |
| 9 | 49,23 | 0,813 |
| 10 | 53,34 | 0,750 |
| 11 | 48,78 | 0,820 |
| 12 | 51,9 | 0,771 |
| 13 | 52,45 | 0,763 |
| 14 | 52,23 | 0,766 |
| 15 | 53,12 | 0,753 |
| 16 | 48,99 | 0,816 |

CONTINUA



| | | |
|--------------|-------|--------------|
| 17 | 47,44 | 0,843 |
| 18 | 47,87 | 0,836 |
| 19 | 48,01 | 0,833 |
| 20 | 53,68 | 0,745 |
| 21 | 52,78 | 0,758 |
| 22 | 52,98 | 0,755 |
| 23 | 52,76 | 0,758 |
| 24 | 51,43 | 0,778 |
| 25 | 54,88 | 0,729 |
| 26 | 49,59 | 0,807 |
| 27 | 48,92 | 0,818 |
| 28 | 47,67 | 0,839 |
| 29 | 54,09 | 0,740 |
| 30 | 5390 | 0,007 |
| 31 | 50,92 | 0,786 |
| 32 | 46,44 | 0,861 |
| 33 | 53,87 | 0,743 |
| 34 | 54,05 | 0,740 |
| 35 | 50,7 | 0,789 |
| 36 | 46,12 | 0,867 |
| 37 | 54,45 | 0,735 |
| 38 | 49,98 | 0,800 |
| 39 | 53,82 | 0,743 |
| 40 | 49,56 | 0,807 |
| Total | | 0.771 |

Se obtuvo una velocidad promedio de 0,771 m/s (2,76 km/h); por lo tanto, el seguimiento se cumple para la velocidad de caminata promedio de una persona, siendo de **2,5 km/h**. (SALUD Y BIENESTAR, 2017).

5.5.1. Velocidad promedio de una persona por tramos. 2

El recorrido de una persona a lo largo de 40 m, se ha dividido para obtener una velocidad más específica a lo largo de tramos, identificando de esa manera que al inicio del vuelo (despegue), el seguimiento toma una velocidad un poco más lenta debido a la estabilización del dron en el aire. A continuación, en la tabla 18

se observa que la velocidad más baja se ubica en los 20 primeros metros y que esta se incrementa a lo largo del trayecto.

Tabla 18.

Tiempo y velocidad por tramos de seguimiento

| Distancia (m) | Tiempo (s) | Velocidad (m/s) | Tiempo Promedio (s) | Velocidad promedio (m/s) | Velocidad promedio (km/h) |
|------------------|---------------|--------------------|---------------------------|--------------------------------|---------------------------------|
| 0 a 20m | 27,77 | 0,720 | | | |
| | 28,31 | 0,706 | | | |
| | 28,65 | 0,698 | | | |
| | 27,78 | 0,720 | | | |
| | 26,11 | 0,766 | | | |
| | 27,43 | 0,729 | | | |
| | 26,05 | 0,768 | 27.549 | 0.727 | 2.62 |
| | 27,12 | 0,737 | | | |
| | 26,02 | 0,769 | | | |
| | 27,98 | 0,715 | | | |
| | 26,23 | 0,763 | | | |
| | 26,56 | 0,753 | | | |
| | 30,18 | 0,663 | | | |
| | 27,72 | 0,722 | | | |
| | 29,32 | 0,682 | | | |
| 20m a 30m | 12,83 | 0,779 | | | |
| | 13,4 | 0,746 | | | |
| | 14,5 | 0,690 | | | |
| | 14,1 | 0,709 | | | |
| | 13,83 | 0,723 | | | |
| | 13,11 | 0,763 | | | |
| | 14,67 | 0,682 | 13,685 | 0,736 | 2.65 |
| | 12,34 | 0,810 | | | |
| | 15,67 | 0,638 | | | |
| | 13,56 | 0,737 | | | |
| | 13,07 | 0,765 | | | |
| | 15,98 | 0,626 | | | |
| | 14,23 | 0,703 | | | |
| | 12,09 | 0,827 | | | |
| | 11,9 | 0,840 | | | |

CONTINUA



| | | | | | |
|------------------|-------|-------|--------|-------|-------|
| 30m a 40m | 11,41 | 0,876 | | | |
| | 11,46 | 0,873 | | | |
| | 10,95 | 0,913 | | | |
| | 11,17 | 0,895 | | | |
| | 9,82 | 1,018 | 10,728 | 0,935 | 3,367 |
| | 10,23 | 0,978 | | | |
| | 10,75 | 0,930 | | | |
| | 11,23 | 0,890 | | | |
| | 9,7 | 1,031 | | | |
| | 10,56 | 0,947 | | | |

Durante el recorrido de los tramos la velocidad llega hasta un 0,935 m/s (3,367 km/h), en un tiempo promedio de 52 s, a lo largo del trayecto establecido.

5.6. Pruebas del Sistema de Seguridad.

Inicialmente se verificó la funcionalidad de los algoritmos de detección y seguimiento por separado a lo largo de distintas trayectorias de 40m, consiguiendo resultados favorables. Sin embargo, es indispensable acoplar dichos algoritmos para formar el sistema de seguridad para un área delimitada al implementar visión artificial que usa en conjunto una cámara fija y otra móvil, para detección y seguimiento de personas en lugares abiertos. En la tabla 19 se indica el resultado de la sinergia.

Tabla 19.

Pruebas del sistema de seguridad (Detección y seguimiento)

| Nº. Pruebas | Pruebas Positivas | Pruebas Negativas |
|----------------|----------------------|----------------------|
| 1 | + | - |
| 2 | + | - |
| 3 | + | - |
| 4 | + | - |
| 5 | - | + |
| 6 | + | - |
| 7 | + | - |

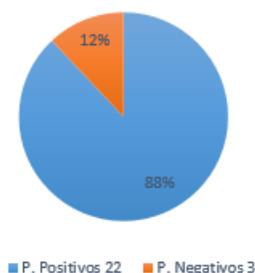
CONTINUA 

| | | |
|-------------------|------------|------------|
| 8 | + | - |
| 9 | + | - |
| 10 | + | - |
| 11 | - | + |
| 12 | + | - |
| 13 | + | - |
| 14 | + | - |
| 15 | + | - |
| 16 | + | - |
| 17 | + | - |
| 18 | - | + |
| 19 | + | - |
| 20 | + | - |
| 21 | + | - |
| 22 | + | - |
| 23 | + | - |
| 24 | + | - |
| 25 | + | - |
| Total | 22 | 3 |
| Porcentaje | 88% | 12% |

Las pruebas realizadas, permiten garantizar que el sistema cumple con los requisitos de seguridad para este proyecto:

- Vigilancia de un área delimitada con una cámara en una estación fija
- Detección de un intruso dentro de un área restringida
- Detección y seguimiento de personas con un dron.

Al implementar el sistema se obtuvo un 88% de eficacia, es decir mejora la seguridad dentro de un área, activando un mecanismo de seguimiento con un dron, monitoreando de esa manera la trayectoria del sujeto, para posteriormente realizar su detención, mientras que el dron recibe la orden de regreso a casa, enviada por el usuario. En la figura 58 se indica los porcentajes de pruebas favorables y no favorables.

PRUEBAS DEL SISTEMA
DE SEGURIDAD**Figura 58. Pruebas de seguridad****5.7. Comprobación de la hipótesis del sistema**

- **Hipótesis nula**

H_0 = No se mejorará la seguridad de un área delimitada al implementar un sistema de visión artificial que usa en conjunto una cámara fija y otra móvil, para detección y seguimiento de personas

- **Hipótesis válida**

H_i = Se mejorará la seguridad de un área delimitada al implementar un sistema de visión artificial que usa en conjunto una cámara fija y otra móvil, para detección y seguimiento de personas

Margen de error considerada= 5%

Margen de error tabla= 0.05

Grados de libertad = 2

Tabla 20.**Datos chi-cuadrado hipótesis**

| | Positivo | Negativo | Total |
|--------------------|------------|-----------|------------|
| Detección | 367 | 44 | 411 |
| Seguimiento | 36 | 14 | 50 |
| Seguridad | 22 | 3 | 25 |
| Total | 425 | 61 | 486 |

| | f | Ft | | f | Ft |
|---|------------|-----------|---|-----------|-----------|
| 0 | 367 | 359.41 | 3 | 44 | 51.58 |
| 1 | 36 | 43.72 | 4 | 14 | 6.27 |
| 2 | 22 | 21.86 | 5 | 3 | 3.13 |

$$x^2 = \sum_{i=0}^n \frac{(f - ft)^2}{ft}$$

$$x_{calc}^2 = 12.16$$

Una vez realizados los cálculos del chi-cuadrado se tiene un resultado de 12.16, dicho valor será comparado con el chi-cuadrado de la tabla ($xtbl^2$), que se muestra en la figura 59.

| v/p | 0,001 | 0,0025 | 0,005 | 0,01 | 0,025 | 0,05 | 0,1 |
|------------|--------------|---------------|--------------|-------------|--------------|---------------|------------|
| 1 | 10,8274 | 9,1404 | 7,8794 | 6,6349 | 5,0239 | 3,8415 | 2,7055 |
| 2 | 13,8150 | 11,9827 | 10,5965 | 9,2104 | 7,3778 | 5,9915 | 4,6052 |
| 3 | 16,2660 | 14,3202 | 12,8381 | 11,3449 | 9,3484 | 7,8147 | 6,2514 |
| 4 | 18,4662 | 16,4238 | 14,8602 | 13,2767 | 11,1433 | 9,4877 | 7,7794 |
| 5 | 20,5147 | 18,3854 | 16,7496 | 15,0863 | 12,8325 | 11,0705 | 9,2363 |

Figura 59. Tabla Chi- cuadrado 2 GDL error 5%

$$xtbl^2 = 5.9915$$

$$x_{calc}^2 > xtbl^2 = Hi \text{ es válida}$$

$$x_{calc}^2 < xtbl^2 = Ho \text{ es válida}$$

$$12.16 > 5.9915 = Hi \text{ es válida}$$

Tras la comparación se determina, que la hipótesis positiva (Hi), se cumple, por lo tanto, la implementación del presente proyecto, permite mejorar la seguridad de un área restringida, aplicando adecuadamente cada uno de los aspectos tomados en cuenta para este propósito, siendo:

- Vigilar y detectar un intruso dentro de un área con una cámara en una estación fija

- Detectar y seguir al intruso, mediante procesamiento de imágenes tomadas desde la cámara incorporada en el dron.

CAPITULO VI

6. CONCLUSIONES Y RECOMENDACIONES

6.1. Conclusiones

- Trabajar con el cuadricóptero Parrot Bebop 2, fue una ventaja muy considerable debido a que se puede tener acceso a los datos de vuelo que son enviados por los diferentes sensores. Gracias a esto se puede desarrollar un sinnúmero de aplicaciones, y al tener su propia señal Wi-Fi hace posible que se puedan correr programas directamente desde la computadora, enviando comandos y receptando los diferentes estados de vuelo.
- En la configuración de la cámara del cuadricóptero Parrot Bebop 2, existe la posibilidad de cambiar la resolución nativa, haciendo que esto sea una ayuda al momento de procesar las imágenes.
- La cámara es una parte esencial en un programa de visión asistida por computador, porque otorga un 50% de eficiencia. En este caso el Parrot Bebop 2 cuenta con una cámara de 14 megapíxeles, que ofrece buenas prestaciones al momento del seguimiento de personas.
- La transmisión de video desde el Parrot Bebop 2 al computador tiene un retardo. Esto se debe a la velocidad de comunicación entre dichos componentes, teniendo así una dificultad en el procesamiento de imágenes.
- El uso de las diferentes herramientas que contiene el sistema operativo robótico (ROS), posibilita el desarrollo de aplicaciones para el control, procesamiento de imágenes, usadas al momento de realizar aplicaciones para el vuelo autónomo de cuadricópteros o plataformas robóticas.
- El sistema operativo robótico (ROS), entre sus características tiene la ventaja de ser código abierto, permitiendo tener acceso a diferentes repositorios e integrando códigos ya probados con anterioridad en sistemas robóticos complejos.

- El sistema operativo robótico (ROS), es un sistema muy utilizado en el campo de los desarrolladores, al contar con repositorios de código libre, al tener esta ventaja también existe su contraparte, existen repositorios no oficiales y al momento de integrarlos pueden ocasionar problemas con la aplicación final.
- Se debe tener en cuenta que para el procesamiento de imagen está inmerso el uso de librerías OpenCv, pero el video se recibe como formato de ROS y se necesita transformarlo a formato de OpenCv porque si no se realiza este paso no es posible realizar el procesamiento de imágenes.
- Al momento de realizar la transformación de video a formato de OpenCv se pierden características debido a la forma que es almacenada por ROS, pero gracias a la resolución obtenida de la cámara a bordo del cuadricóptero Parrot Bebop 2, está perdida no influye en gran forma del desempeño al momento de procesar la imagen, y el resultado final.
- Cuando se trabaja con una resolución de video (640 * 480 pixeles), ayuda al procesamiento, debido a que la información recibida es menor, y gracias a que se puede cambiar la resolución nativa de la cámara del cuadricóptero Parrot Bebop 2, no se pierden características para realizar el seguimiento de la persona.
- Al realizar el procesamiento de la imagen para realizar el reconocimiento y seguimiento de la persona se observa que se necesita una gran cantidad de recursos computacionales tanto memoria como velocidad de procesamiento, haciendo así que el uso de una tarjeta embebida, no sea el adecuado para esta aplicación.
- El desempeño del algoritmo de seguimiento y de control depende explícitamente de la calidad de la imagen obtenida ya sea por la cámara a bordo del Parrot Bebop 2 o de la cámara estática, se puede concluir que si se trabaja con una resolución nativa relativamente baja el procesamiento será mejor y más rápido.

- Al iniciar el seguimiento de las personas, el cuadricóptero intentará mantener el objetivo en el centro de la pantalla y a su vez la distancia predefinida en la programación de 3 metros.
- Para tener un control adecuado del cuadricóptero, se debe obtener de manera adecuada la planta que representará a este, porque de acuerdo a esta se puede realizar un correcto cálculo de las ganancias para el control PID.
- Cuando se implementa el control PID, este debe estar calibrado de la forma adecuada para poder tener un seguimiento óptimo, ya que este es el que tiene el control total del cuadricóptero al momento de estar en modo autónomo.
- El uso de Matlab fue una gran ayuda al momento de implementar el control PID, ya que nos entrega la función de transferencia con sus respectivas ganancias K_p , K_i , K_d , de cada eje a controlar, pero al momento de realizar pruebas se vio la necesidad de calibrar estas usando el método oscilación de Ziegler-Nichols.
- El algoritmo de reconocimiento de personas debe ser robusto para que pueda soportar los diferentes cambios de luz en ambientes no controlados.
- Obtenido el flujo de imagen en formato de ROS, se procede a ecualizar su histograma para obtener una mejor descripción de las características, ya que esto permite que todos los posibles valores de grises se distribuyan de manera uniforme en la imagen, aproximándose a un histograma ideal.
- Mediante el cambio de la imagen original a imagen en escala de grises, se mejoró el tiempo de procesamiento, dado que este último solo necesita un canal a diferencia de las imágenes RGB que necesitan 3.
- El algoritmo de reconocimiento de personas debe ser robusto para que pueda soportar los diferentes cambios de luz en ambientes no controlados, por tal motivo se usa el algoritmo de HOG, el cual brinda estas características.
- Mediante el uso del clasificador SVM, para separar las clases personas y no persona, desde un conjunto de entrenamiento predeterminado, se logró

obtener una detección eficiente, al aplicarlo en los algoritmos de control la cámara estática y del dron.

- La correcta detección de personas, es fundamental, ya que gracias a esto se enviará comandos que rigen el control de vuelo del dron, tales como su despegue, regreso a casa y el principalmente el seguimiento en profundidad y lateralmente.
- La detección en vuelo, permite un adecuado seguimiento en lugares amplios y cerrados, así como en lugares abiertos, en los cuales no haya presencia de viento fuerte, dado que este influye, modificando la altura adecuada en vuelo del dron, por lo cual el cuadricóptero perdería al objetivo a seguir.

6.2. Recomendaciones

- Para empezar las pruebas de esta aplicación, primero se hizo una prueba de control mediante el Smartphone para recolectar posibles problemas en la conexión Wi - Fi, observando que al existir mucho ruido por las redes inalámbricas existentes se pierde el control del cuadricóptero, esto se puede solucionar si se tiene un espacio donde no exista tanta interferencia por señales inalámbricas o a su vez realizando una programación más avanzada en el cuadricóptero.
- Cuando se ejecuta el programa, este configura los parámetros de fábrica, haciendo que se implante la configuración de vuelo en interiores, por lo que es necesario cambiar la configuración a vuelo de exteriores, obteniendo como resultado mejor estabilidad, y mayor potencia en los motores
- Tomar de forma adecuada las medidas del motor y del centro del cuadricóptero, para realizar el cálculo de la inercia, dado que de dicho calculo dependerá el modelado matemático del cuadricóptero.
- Verificar continuamente el porcentaje de la carga de la batería, dada que en un 20%, empieza a dar dificultades en el seguimiento.

- Realizar el vuelo del dron en lugares en donde no se tenga la presencia de viento fuerte, debido a que este provoca un vuelo inestable y por lo tanto la posible pérdida del objetivo.
- Se debe tener un juego de hélices adicional, ya que al realizar pruebas estas pueden sufrir daños, haciendo que el vuelo del cuadricóptero sea inestable.
- Al momento de realizar pruebas del regreso a casa, se debe considerar la altura que se elevará el dron, ya que esta es de 20 metros aproximadamente, esto lo realiza elevándose y haciendo una trayectoria recta hasta el punto de despegue.

REFERENCIAS BIBLIOGRÁFICAS

- Barrientos, A., Peñin, L., Balaguer, C., & Aracil, R. (2007). *Fundamentos de robótica*. Madrid: S.A. MCGRAW-HILL / INTERAMERICANA DE ESPAÑA.
- Blasco, X., García, N. S., & Reynoso, M. G. (2012). *Control autónomo del seguimiento de trayectorias de un vehículo cuatrirrotor*. Valencia - España: Universidad Politecnica de Valencia.
- Bristeau, P., Callou, F., Vissiere, D., & Petit, N. (2011). *The navigation and control technology inside the AR.Dron micro UAV*. Milano: IFAC.
- Brito Jorge. (2009). *Quadrator Prototype*. Lisboa- Portugal: Instituto Superior Tecnico Portugal.
- Camazón, J. N. (2011). *Sistemas Operativos Monopuestos*. Madrid- España: Editex.
- CEDATOS. (4 de 11 de 2016). CEDATOS. Recuperado el 2 de 4 de 2017, de http://www.cedatos.com.ec/detalles_noticia.php?Id=86
- CEIL. (2012). *Sistemas de Seguridad*. Galicia: Universidad Politecnica de Madrid.
- Chamizo Daniel. (2013). *ENTORNO DE PROGRAMACIÓN GRÁFICO OPENCV*. Madrid: ESCUELA POLITÉCNICA UNIVERSIDAD CARLOS III DE MADRID. Obtenido de <http://docplayer.es/17663443-Proyecto-fin-de-grado-entorno-de-programacion-grafico-opencv-autor-daniel-chamizo-alberto-ingenieria-electronica-industrial-y-automatica-tutores.html>
- Clearpath Robotics. (2015). *CREATING A WORKSPACE AND PUBLISHER*. Recuperado el 11 de 2016, de ROS Tutorials: <https://www.clearpathrobotics.com/assets/guides/ros/Creating%20publisher.html>
- Clearpath Robotics Inc. (2016). *ROS 101: INTRODUCCIÓN AL SISTEMA OPERATIVO DEL ROBOT*. Recuperado el 11 de 2016, de <https://www.clearpathrobotics.com/2014/01/how-to-guide-ros-101/>

- Creative Commons. (28 de 11 de 2016). *Ubuntu documentation*. Recuperado el 2016, de OpenCV: <https://help.ubuntu.com/community/OpenCV>
- Diaz, S. M. (18 de 2 de 2017). *Diseño y Construcción de un Quadcopter*. Chile: Universidad de Chile. Obtenido de <https://upcommons.upc.edu/bitstream/handle/2099.1/21902/102664.pdf?sequence=1&isAllowed=y>
- Escalante Boris. (2006). *Procesamiento Digital de Imágenes*. Mexico: UNAM.
- Freire J. (08 de 2015). *La seguridad a través de DRONS será el reto de EP Petroecuador en el #CPEC5*. Recuperado el 12 de 07 de 2016, de <http://www.doctortecno.com/noticia/seguridad-traves-drons-sera-reto-ep-petroecuador-cpec5>
- Gallardo H., J. J. (11 de 2011). *Navegación por Procesamiento de Imágenes Usando OpenCV*. Monterrey - Mexico: Instituto Tecnológico y de Estudios Superiores Monterrey Campus San Luis Potosí.
- Garcia Alvaro. (07 de 09 de 2013). *ROS: Robot Operating System*. Cartagena: Universidad Politecnica de Cartagena. Obtenido de <http://docplayer.es/2377199-Ros-robot-operating-system.html>
- Garcia, A. (08 de 2016). *Este temible dron autónomo sustituirá a los guardas de seguridad*. Recuperado el 15 de 11 de 2016, de ADSLZONE: <https://www.adslzone.net/2016/08/22/este-temible-dron-autonomo-sustituira-los-guardas-seguridad/>
- GLOBOVISION. (01 de 2016). *Ministro González: Sistema de seguridad usa drons para apoyar operaciones*. Recuperado el 07 de 2016, de <http://globovision.com/article/ministro-gonzalez-sistema-de-seguridad-usa-drons-para-apoyar-operaciones>
- Hinojosa, D., & Maontaluca, C. (2014). *Diseño y construcción de una aeronave no tripulada tipo octocóptero*. Latacunga: Universidad de las Fuerzas Armadas ESPE.

- Hutchinson, S., Hanger, G., & Corke, P. (1996). A Tutorial on Visual Servo Control. *IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION*, 651 - 671.
- INFAIMON S.L. (2016). *INFAIMON Vision artificial*. Recuperado el 07 de 07 de 2016, de seguridad: <http://www.infaimon.com/es/seguridad>
- Intxaurre J. (2013). *Deteccion de personas*. Vizcaya - España: Universidad del Pais Vasco Euskal Herriko Unibertsitatea.
- Kaingade, S., More, V., Dhule, D., Gaidhani, P., & Gupta, N. (2015). *VISION BASED OBJECT DETECTION AND TRACKING USING MULTIROTOR UNMANNED AERIAL VEHICLE*. Pune, Maharashtra - India: NavStik Autonomous Systems.
- Maldonado, D., & Calderon, M. (2014). *Control de un cuadricoptero usando ROS*. Quito: Escuela Pólitecnica Nacional.
- Martinez, A. (06 de 2014). *ROS/Concepts*. Recuperado el 08 de 2016, de <http://wiki.ros.org/ROS/Concepts>
- Matinez, D. L. (2013). *SISTEMAS OPERATIVOS*. Argentina.
- Microsoft. (19 de 1 de 2017). *Microsoft*. Obtenido de <https://www.microsoft.com/es-xl/windows>
- Monajjemi Mani. (2015). *AutonomyLab, Simon Fraser University*. Recuperado el 11 de 2016, de <http://bebop-autonomy.readthedocs.io/en/latest/>
- Monajjemi Mani. (2015). *List of Ardrón3 Settings and Corresponding ROS Parameter*. Recuperado el 01 de 2017, de (AutonomyLab, Simon Fraser University): http://bebop-autonomy.readthedocs.io/en/latest/autogenerated/ardron3_settings_parameters.html#picturesettingsvideoresolutionstype
- Montero, J. (10 de 2016). *Valencia vigilará con drones las zonas rurales para evitar robos*. Obtenido de TODRON: <http://www.todron.com/valencia-vigilara-drones-rurales-evitar-robos/>

- Mott, R. (2009). *Resistencia de materiales*. México: Pearson Educación.
- Olivares, M. A., Fu, C., Ludivig, P., Bissyandé, T., Zurad, S. K., Voos, A. A., & Campoy, P. (2015). *Towards an Autonomous Vision-Based Unmanned Aerial System against Wildlife Poachers*. Basel, Switzerland: MDPI.
- OpenCV. (2015). *OpenCV 3.1.0*. Recuperado el 08 de 2016, de Open Source Computer Vision: <http://docs.opencv.org/3.1.0/d1/dfb/intro.html>
- Parrot S.A. (2016). *Parrot BEBOP 2 FPV*. Recuperado el 12 de 2016, de <https://www.parrot.com/es/drons/parrot-bebop-2-fpv#técnicos>
- Parrot, C. (18 de 11 de 2015). *Parrot*. Obtenido de <https://www.parrot.com/es/drons/parrot-bebop-2#parrot-bebop-2->
- Pestana, J., Sanchez, J., Campoy, P., & Saripalli, S. (2013). *Vision based GPS-denied Object Tracking and Following for Unmanned Aerial Vehicles*. Madrid: IEEE.
- Pestana, M. J., Sanchez, M. J., Saripalli, D. S., & Campoy, D. P. (22 de 3 de 2017). *Vision4UAV*. Obtenido de <http://138.100.76.11/visionguided2/?q=following>
- Redacción Bogotá. (19 de 10 de 2016). *Vigilancia con drons para fortalecer la seguridad en Bogotá*. Obtenido de EL ESPECTADOR: <http://www.elespectador.com/noticias/bogota/vigilancia-drons-fortalecer-seguridad-bogota-articulo-661096>
- Reinoso, M. (2014). *Diseño de un sistema de control por regimen deslizante para el seguimiento de trayectoria lineal de un quadrotor*. Cuenca: Universidad Politecnica Salesiana.
- Rivera Nicolas. (13 de 02 de 2016). *El estado de los drons en 2016*. Obtenido de hipertextual: <https://hipertextual.com/2016/02/drons>
- ROS.org. (05 de 2015). *cv_bridgetutorialesUsingCvBridgeToConvertBetweenROSImagesAndOp*

enCVImages. Recuperado el 10 de 2016, de http://wiki.ros.org/cv_bridge/Tutorials/UsingCvBridgeToConvertBetweenROSImagesAndOpenCVImages

S., H., G., H., & P., C. (1996). A Tutorial on Visual Servo Control. *IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION*, 651 - 671.

Sanchez, G., Mulero, M., & Saumeth, E. (2013). *Vehiculos Aereos no Tripulados en Latinoamérica*. España: Alfredo Medina.

Simonite, T. (2016). *Dron Security Guard Scolds Intruders from the Sky*. Obtenido de MIT Technology Review: <https://www.technologyreview.com/s/602412/dron-security-guard-scolds-intruders-from-the-sky/?set=602416>

Torres Adria. (28 de 12 de 2017). *USO DE DRONS EN SEGURIDAD*. Obtenido de DronSpain: <http://dronspain.pro/uso-drons-en-seguridad/>

Valveny E. (2015). *Lecture 29 - L4.4. HOG – Cálculo del descriptor*. Recuperado el 20 de 03 de 2017, de Coursera: <https://es.coursera.org/learn/deteccion-objetos/lecture/uAEKB/l4-4-hog-calculo-del-descriptor>

WIKILIBROS. (09 de 08 de 2014). *Visual servoing*. Recuperado el 03 de 15 de 2017, de https://es.wikibooks.org/wiki/Visual_servoing#1.2..09Visual_servo_basado_en_imagen_.28IBVS.29

Willow Garage. (2015). *ROS*. Recuperado el 28 de 11 de 2016, de <http://www.willowgarage.com/pages/software/ros-platform>

Zahumenszky Carlos. (12 de 12 de 2013). *Así podría ser un sistema real de vigilancia con drones en pocos años*. Recuperado el 12 de 03 de 2017, de GIZMODO: <http://es.gizmodo.com/asi-podria-ser-un-sistema-real-de-vigilancia-con-drons-1481843805>

Zone, A. (17 de 2 de 2017). *www.adslzone.net*. Recuperado el 11 de 03 de 2017, de www.adslzone.net: www.adslzone.net

ANEXOS



ESPE

UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

DEPARTAMENTO DE CIENCIAS DE LA ENERGÍA Y MECÁNICA

CARRERA DE INGENIERÍA EN MECATRÓNICA

CERTIFICACIÓN

Se certifica que el siguiente trabajo fue desarrollado por los señores: **LILIA MARISOL MORALES CACPATA** y **CARLOS MANUEL PAUCAR TIPÁN**, bajo nuestra supervisión.

Ing. Darío Mendoza

TUTOR DEL PROYECTO

Aprobado por:

Ing. Vicente Hallo

DIRECTOR DE CARRERA

Dr. Rodrigo Vaca

SECRETARIO ACADÉMICO