FACULTAD DE INGENIERÍA

ESCUELA DE INGENIERÍA CIVIL MECATRÓNICA

SISTEMA DE ADQUISICIÓN Y ANÁLISIS DE DATOS PARA EL SEGUIMIENTO DE OBJETOS POR MEDIO DE VEHÍCULO AUTODIRIGIDO

Memoria para optar al Título de

Ingeniero Civil Mecatrónica

Profesor Guía:

Roberto Ramírez

JUAN FRANCISCO OSSES CANCINO

CURICÓ-CHILE

2021

CONSTANCIA

La Dirección del Sistema de Bibliotecas a través de su encargado Biblioteca Campus Curicó certifica que el autor del siguiente trabajo de titulación ha firmado su autorización para la reproducción en forma total o parcial e ilimitada del mismo.

Curicó, 2023

SISTEMA DE ADQUISICIÓN Y ANÁLISIS DE DATOS PARA EL SEGUIMIENTO DE OBJETOS POR MEDIO DE VEHÍCULO AUTODIRIGIDO

Juan Francisco Osses Cancino

Enero, 2021

Resumen

En robótica, el seguimiento visual a objetos es un proceso que ayuda a obtener información en tiempo real o no, de distintas áreas de competencias, tales como tecnología, deporte, salud, agrícola, entre otros. Logrando así adquirir resultados con respaldo visual, sin necesidad de moverse del lugar de trabajo.

En la actualidad, a medida que la tecnología avanza, el seguimiento visual va teniendo una mejora continua, logrando captar datos por medio de objetos y/o personas que estén en constante movimiento. De esta manera, se puede inspeccionar y procesar la información recolectada de una forma más precisa y eficaz sin necesidad de detener o pausar la actividad.

Para realizar las evaluaciones en tiempo real, se necesita un equipo que vaya adquiriendo y procesando las imágenes, además de un robot (aéreo, terrestre o acuático) que se traslade gracias a la recepción de ciertos parámetros (entregados mediante control manual o control autónomo) que permitan a los sensores de imagen mantenerse siempre en dirección al sujeto o persona en estudio.

El proyecto por trabajar, el objetivo yace de realizar un seguimiento autónomo continuo por medio de un equipo aéreo no tripulado (Drone), sin necesidad de que alguien deba estar controlando su posición respecto a cómo se mueva el objeto. Esto se llevaría a cabo por medio de la adquisición de imágenes mediante la utilización de una cámara montada en el mismo Drone, con la que se estima la posición del objeto u persona en estudio para ejecutar la acción de control de posición de este, manteniendo al sujeto a una distancia fija y en un rango cercano al centro focal de la cámara.

Como propuesta, se desea integrar tecnologías que logren desarrollar un control de vuelo autónomo mediante la utilización de *softwares* de procesamiento de imágenes, en conjunto con un algoritmo que logre interpretar los datos para enviar las señales correspondientes a los análogos del joystick del Drone y genere los movimientos correspondientes para mantener siempre en visual al objeto.

Dedicado en especial a mi familia, amigos, y cada uno que me ha invitado a crecer…

Agradecimientos

Quiero enfatizar que no soy bueno expresando emociones ni nada, pero quisiera agradecer a la vida que me ha llevado hasta este punto, mi familia que es una base fundamental de mi formación, a mi entrenador que me enfatizó en fijar objetivos y tener las metas claras, como también mis amigos que son parte de mí.

"Solo hay una pequeña parte del universo de la que sabrás con certeza que puede ser mejorada, y esa parte eres tú, Aldous Huxley."

Contenido

Índice de Figuras

Índice de ecuaciones y tablas

Capítulo 1 Introducción

1.1 Introducción General

La robótica está hoy inserta en prácticamente todas las industrias y hogares de Chile. Desde pequeños robots que limpian pisos y ventanas, hasta sistemas que detectan temperaturas y humedad para mantener en buen estado frutas y verduras durante semanas.

Hoy, estas tecnologías son fundamentales para las empresas. Es que automatizar procesos permite que las compañías ahorren dinero y realicen sus tareas con mayor rapidez y efectividad [1].

Son 5 áreas donde automatización está cambiando la forma de ver y solventar los trabajos: Minería, Agricultura, Comercio, Gastronomía y Drones. Aunque en particular los Drones no son una industria, más bien una tecnología, han transformado la forma que vemos el mundo y sus múltiples aplicaciones han ayudado a facilitar tareas y llegar de manera fácil donde las personas no pueden o tienen difícil acceso [1] .

En este último tiempo, la creación de UAV (*Unmanned Aerial Vehicle*) han crecido a un gran nivel por la cantidad de empresas que se afilian a este mercado, y a la vez, la tecnología que se integra es mayor (diseño, comunicación y sistema de control) lo que permite a muchas empresas o personas particulares trabajar en proyectos con UAV [2]. Los Drones ofrecen posibilidades extraordinarias para usuarios empresariales y privados y serán empleados de manera creciente en aplicaciones de observación ya que pueden contener cámaras de alta resolución, sensores y geolocalización para ubicar cualquier problema o defectos. También pueden reforzar las misiones de rescate y búsqueda a través del uso de Drones equipados con cámaras infrarrojas o simplemente realizar seguimiento a vehículos, personas u objetos dependiendo del objetivo en que se encuentre [3].

En la actualidad, se han generado proyectos evocados al seguimiento de personas u objetos mediante de distintos métodos, algunos de los trabajos investigados son como el Diseño e implementación del sistema de control de posición para un Drone basado en seguimiento de objetos con el Drone Ar. Drone 2.0 [4] o el desarrollo de un vehículo no tripulado que de forma autónoma para que realice el vuelo siguiendo a un objeto [5] [6] y el modelado y control de un Micro Vehículo Submarino Autónomo (AUV), nombrado AR2D2 [7].

El proyecto presenta un equipo que tiene la capacidad de realizar un seguimiento continuo a un objeto, mediante la utilización de una cámara montada en un Drone que visualice a un objetivo en particular. Para ello, se genera un algoritmo con la capacidad de detectar objetos que, conectada a un microcontrolador, es capaz de procesar las imágenes y generar señales de control para mantener siempre en visual al objetivo, como para el seguimiento en una competencia atlética, el estudio biomecánico de una persona, seguimiento de un vehículo, según sea su requerimiento. Ya que actualmente, el control de los Drone lo realiza una persona que tiene las habilidades de pilotear esta tecnología o también un control automático existente con el análisis previo del circuito que atravesará o más bien de un sistema GPS [5]. La idea de generar un seguimiento continuo a partir de una cámara montada en el Drone es integrar tecnologías que logra un proyecto de rápida respuesta, con un valor menor al del mercado y con un mínimo margen de error en el respectivo análisis, dando la posibilidad de entrar a múltiples áreas de trabajo con robot de seguimiento de control automático para distintas áreas en las que se pueda trabajar tanto para la región como a nivel nacional.

El documento está organizado de la siguiente manera: el capítulo 1 se presenta el estado del arte, donde se argumenta las bases actuales de este proyecto respecto el procesamiento de imágenes, tipos de Drones, control de robot actualmente utilizados y de definir los objetivos, alcances, limitaciones y metodología de trabajo que tendrá el proyecto propuesto. Para el capítulo 2 habla de los implementos utilizados tanto físicos como de los *softwares* utilizados para llevar a cabo el proyecto, como también el tipo de conexión realizado con el fin de enviar las señales que permitan el control del Drone. El capítulo 3 describimos el procesamiento de imágenes utilizado en el proyecto, identificando desde la calibración de la cámara utilizada hasta encontrar el área y centro de la persona que se utiliza en el proceso de este proyecto. De igual forma, el capítulo 4 habla sobre el algoritmo de control que genera las señales para modificar la posición del Drone, explicando inicialmente la comunicación efectuada con el algoritmo de procesamiento de imágenes a fin de poder transmitir los datos necesitados, y con ello, el capítulo finaliza exponiendo los procesos que el algoritmo desarrollado tiene para lograr el control autónomo continuo. Por último, el capítulo 5 se describe los resultados experimentales del proyecto basado en la secuencia de imágenes obtenidas de las grabaciones que se realizan para verificar el correcto funcionamiento del

algoritmo de control autónomo según los datos adquiridos por el procesamiento de la imagen captada de la cámara. Por último, se entrega las conclusiones del desarrollo de este proyecto y los trabajos futuros en los que se puede modificar e implementar.

1.2 Estado del Arte

En esta sección se presenta la investigación previa desarrollada para el proyecto. Esta revisión se orienta al seguimiento de objetos y personas, al control de sistemas robóticos y los tipos de robots aéreos existentes. Para terminar, se presenta la propuesta de este trabajo y se definen objetivos sobre los que este se enfocará.

1.2.1 Seguimiento de objetos y personas

La detección de objetos es una tecnología de ordenador relacionada con la visión artificial y el procesamiento de imagen que trata de detectar casos de objetos semánticos de una cierta clase (como humanos, edificios, o coches) en vídeos e imágenes digitales. El control visual consiste en integrar, en tiempo real, las informaciones visuales en el bucle de control de un robot manipulador. Una aproximación muy utilizada, consiste en dotar al robot de un sistema de visión compuesto por una cámara de forma que se permita determinar la localización de los objetos en cuestión [8].

La visión artificial, visión por computador o visión técnica es una disciplina que tiene como finalidad, reproducir artificialmente el sentido de la vista mediante el procesamiento e interpretación de imágenes, capturadas con distintos tipos de sensores (en la mayoría de los casos cámaras) y utilizando para ello las prestaciones de los ordenadores [9] [10]. La visión artificial es una gran herramienta para establecer la relación entre el mundo tridimensional y sus vistas bidimensionales. Por medio de esta teoría se pueden hacer, por una parte, una reconstrucción del espacio tridimensional a partir de sus vistas, y, para llevar a cabo una simulación de una proyección de una escena tridimensional en la posición deseada a un plano bidimensional.

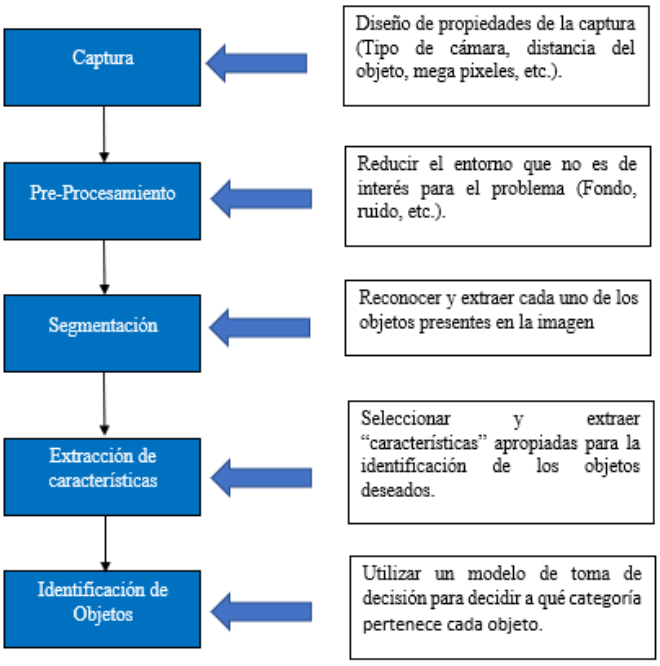

Fig. 1.1 Etapas del procesamiento de imagen [11].

Como se muestra en la Fig. 1.1, la primera fase que corresponde a una etapa sensorial, que consiste en la captura o adquisición de la imagen del mundo real en imágenes digitales, a través de algún tipo de sensor.

La segunda etapa consiste en el tratamiento digital de las imágenes obtenidas, que tiene como objeto facilitar las etapas posteriores. Esta etapa, llamada procesamiento previo, es donde, mediante filtros y transformaciones geométricas, entre otros, se eliminan partes indeseables de la imagen o se realzan partes interesantes de la misma.

La siguiente fase se le conoce como segmentación, y consiste en aislar una zona o elementos que son relevantes en una escena para poder ser comprendida.

La cuarta fase corresponde a identificar las características propias de la detección (color, forma, tamaño, entre otras).

Por último, se llega a la etapa de reconocimiento o clasificación, en ella se obtiene la distinción de los objetos segmentados de la escena, gracias a los análisis de ciertas características que fueron establecidas previamente para diferenciarlos [11].

El procesamiento de imágenes tiene como objetivo mejorar el aspecto de las imágenes y hacer más evidentes en ellas ciertos detalles que se desean hacer notar. La imagen puede haber sido generada de muchas maneras, por ejemplo, fotográficamente, o electrónicamente

o por medio de monitores de televisión. El procesamiento de las imágenes se puede en general hacer por medio de métodos ópticos, o bien por medio de métodos digitales, en una computadora [12]. Los ámbitos mejor desarrollados de detección de objetos incluyen detección de caras y detección de personas. La detección de objetos tiene aplicaciones en muchas áreas de visión artificial, incluyendo recuperación de imágenes y videovigilancia. El objetivo de un sistema de visión consiste en extraer la información de interés del ambiente y proveerla para tomar decisiones de alto nivel. El componente de visión es el primer mecanismo que tiene un robot para censar su ambiente [13].

Para cualquier sistema de control, se dispone de la medición del sistema actual, posición de referencia y la respectiva estrategia de control, en este caso se hará referencia a la estimación entregada por la cámara y un equipo robótico que permita el seguimiento como se señala en la Fig. 1.2.

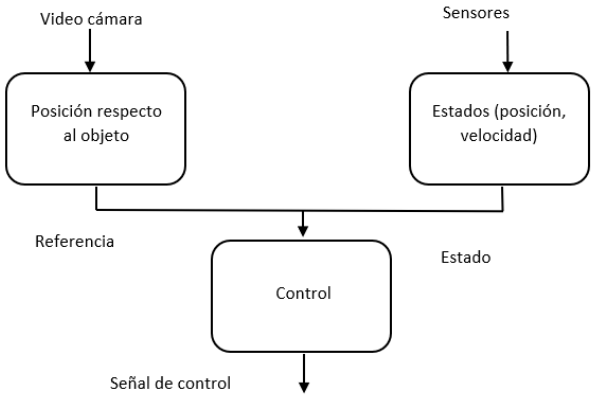

Fig. 1.2 Arquitectura general del sistema [4].

En cuanto a la estimación de la posición respecto del objeto, se tiene dos procesos mostrados en la Fig. 1.3 los que se reconocen como detección de colores y SURF. La detección de color es la capacidad de un sensor o algoritmo de distinguir colores a partir de la extracción de información de la luz. La manera básica de detectar el color consiste en captar la luz incidente en un sensor. Este, mediante un conjunto de celdas de fotones que forman una matriz de puntos, uno por cada píxel, es capaz de medir la cantidad de luz llegada a cada uno de estos, produciendo una corriente eléctrica que varía en función de la intensidad.

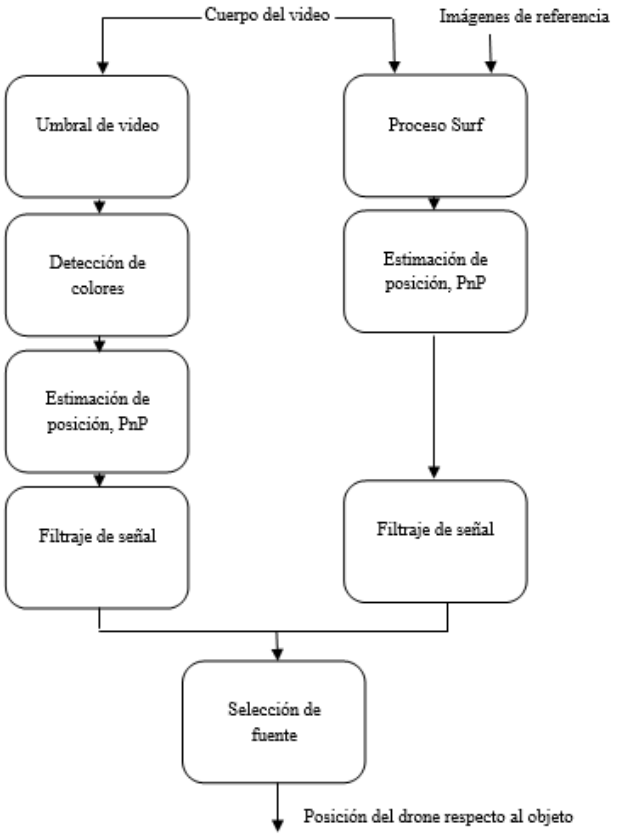

Fig. 1.3 Diagrama para posición del objeto [4].

SURF es un algoritmo de visión por computador, capaz de obtener una representación visual de una imagen y extraer una información detallada y específica del contenido. Esta información es tratada para realizar operaciones como por ejemplo la localización y reconocimiento de determinados objetos, personas o caras, realización de escenas 3D, seguimiento de objetos y extracción de puntos de interés. Este algoritmo forma parte de la mencionada inteligencia artificial, capaz de entrenar un sistema para que interprete imágenes y determine el contenido [4] [14]. Otros algoritmos investigados fueron SIFT, ASIFT y Tracking. SIFT es un algoritmo para detectar y describir las características locales en las imágenes, su metodología de trabajo es más robusto siendo que sea más lento que SURF [14]. SIFT puede extraer características relevantes de las imágenes que posteriormente pueden usarse en reconocimiento de objetos, detección de movimiento, estereopsis, registro de la imagen y otras tareas. ASIFT a diferencia del método SIFT, simula tres parámetros: el zoom, el ángulo de la cámara en latitud y el ángulo de la cámara en longitud, y normaliza los otros 3 parámetros: la traslación, rotación y escala. Para Tracking, en la forma más simple,

se puede definir como el problema de estimar la trayectoria de un objeto en el plano de la imagen a medida que se mueve alrededor de una escena. En otras palabras, un rastreador asigna etiquetas consistentes a los objetos rastreados en diferentes fotogramas de un vídeo [15].

Para desarrollar los algoritmos planteados con anterioridad, se debe tener consideración un manejo de los distintos lenguajes que se puede aplicar estos algoritmos. Los lenguajes investigados para poder realizar el procesamiento de imagen son Python, C++, Matlab y ROS.

1.2.1.1 Python

Python es un lenguaje de programación interpretado de tipado dinámico cuya filosofía hace hincapié en una sintaxis que favorezca un código legible. Se trata de un lenguaje de programación multiparadigma y disponible en varias plataformas [16].

Dicho de otro modo, Python es:

Interpretado**:** Se ejecuta sin necesidad de ser procesado por el compilador y se detectan los errores en tiempo de ejecución.

Multiparadigma: Soporta programación funcional, programación imperativa y programación orientada a objetos.

Tipado dinámico: Las variables se comprueban en tiempo de ejecución.

Multiplataforma: disponible para plataformas de Windows, Linux o MAC.

Gratuito: No dispone de licencia para programar.

1.2.1.2 C++

C++ es un lenguaje imperativo orientado a objetos derivado del C. El resultado es que, como su ancestro, sigue muy ligado al hardware subyacente, manteniendo una considerable potencia para programación a bajo nivel, pero se le han añadido elementos que le permiten también un estilo de programación con alto nivel de abstracción. Todo puede programarse con ellos, desde sistemas operativos y compiladores hasta aplicaciones de bases de datos y procesadores de texto, pasando por juegos, aplicaciones a medida, etc. [17]. En la actualidad, el C++ es un lenguaje versátil, potente y general. Su éxito entre los programadores profesionales le ha llevado a ocupar el primer puesto como herramienta de desarrollo de aplicaciones. El C++ mantiene las ventajas del C en cuanto a riqueza de operadores y expresiones, flexibilidad, concisión y eficiencia. Además, ha eliminado algunas de las

dificultades y limitaciones del C original. La evolución de C++ ha continuado con la aparición de Java, un lenguaje creado simplificando algunas cosas de C++ y añadiendo otras, que se utiliza para realizar aplicaciones en Internet.

1.2.1.3 Matlab

La plataforma de MATLAB está optimizada para resolver problemas científicos y de ingeniería. El lenguaje de MATLAB, basado en matrices, es la forma más natural del mundo para expresar las matemáticas computacionales. Las gráficas integradas facilitan la visualización de los datos y la obtención de información a partir de ellos. Una basta biblioteca de herramientas *(Toolboxes)* integradas le permite empezar a trabajar inmediatamente con algoritmos esenciales para su dominio. El entorno de escritorio invita a experimentar, explorar y descubrir. Todas estas herramientas y funciones de MATLAB están probadas rigurosamente y diseñadas para trabajar juntas. Cuyas características principales son:

Lenguaje de alto nivel para cálculos científicos y de ingeniería.

Entorno de escritorio optimizado para la exploración iterativa, el diseño y la solución de problemas.

Gráficas para visualizar datos y herramientas para crear diagramas personalizados.

Aplicaciones para ajustar curvas, clasificar datos, analizar señales, ajustar sistemas de control y muchas otras tareas.

Toolboxes complementarias para una amplia variedad de aplicaciones científicas y de ingeniería.

Herramientas para crear aplicaciones con interfaces de usuario personalizadas.

Interfaces para C/C++, Java, .NET, Python, SQL, Hadoop y Microsoft Excel.

Opciones de implementación libres de derechos para compartir programas de MATLAB con los usuarios finales [18].

1.2.1.4 Halide

Halide es la respuesta del Massachusetts Institute of Technology (MIT, con la ayuda de Stanford y Adobe), en forma de un lenguaje funcional que permite especificar los algoritmos de procesamiento de imágenes, en la mayoría de los casos, vía los métodos de convolución (que son los que definen finalmente los filtros que vemos en programas como Photoshop, de hecho, son matrices de números que se suman, multiplican o restan, sobre los pixeles de regiones de imágenes). La ventaja es que los métodos de convolución ya hacen la

tarea y no hay que entender siquiera cómo trabajan, solamente se hacen las llamadas a esas rutinas en caso necesario [19].

1.2.1.5 ROS

Si bien ROS no es un lenguaje, más bien un *framework* (marco de trabajo) para el desarrollo de software para robots. ROS provee los servicios estándar de un sistema operativo tales como abstracción del hardware, control de dispositivos de bajo nivel, implementación de funcionalidad de uso común, paso de mensajes entre procesos y mantenimiento de paquetes. Está basado en una arquitectura de grafos (teorías de gráficas) donde el procesamiento toma lugar en los nodos que pueden recibir, mandar y multiplexar mensajes de sensores, control, estados, planificaciones y actuadores, entre otros [20].

1.2.2 Drones

Un Drone en términos tecnológicos es una aeronave no tripulada. Más formalmente los Drones son conocidos como Vehículos Aéreos no Tripulados (VANT) o (UAV – *Unmanned Aerial Vehicles*). En esencia, un Drone es un robot volador que puede ser controlado de forma remota o volar de manera autónoma a través de planes de control de vuelo basados en software y sistemas que trabajan en conjunto tales como sensores y GPS a bordo. En el pasado reciente los VANT estaban más asociados con el sector militar, fueron usados inicialmente en prácticas como blancos aéreos, labores de inteligencia y más controversialmente como plataforma de armamento [3] [21].

1.2.2.1 Tipos de Drones

Debido a la gran diversidad de UAVs existentes y a las múltiples misiones que éstos pueden ejecutar, no existe una clasificación universalmente aceptada. Según el criterio buscado se logra tener la clasificación basada en los métodos de generación de sustentación. Este criterio de clasificación agrupa a los UAVs en dos grandes grupos, en uno de ellos se encuentran todos los aerodinos (aeronaves más pesadas que el aire) y en un segundo grupo los aerostatos (aquellas aeronaves cuya suspensión en el aire se debe al empleo de un gas más ligero que el propio aire). En la Fig. 1.4 se presenta el esquema de dicha clasificación [22].

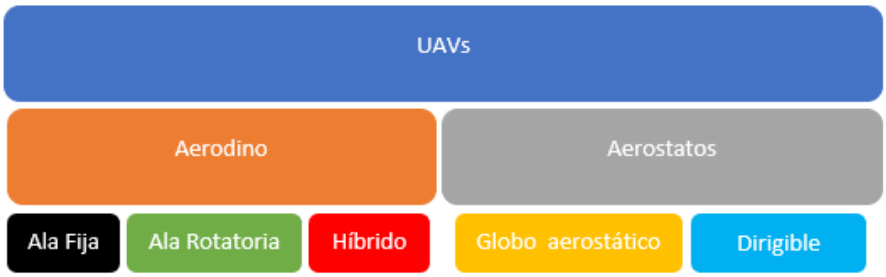

Fig. 1.4 clasificación basada en los métodos de generación de sustentación [22].

Los 3 tipos de aerodino más comunes de este robot son:

a) Ala fija: Los Drones de ala fija son diseñados como la mayoría de las aeronaves, luciendo similar a los aviones (Fig. 1.5). El fuselaje está compuesto por un cuerpo central que tiene dos alas y una sola propela. Una vez en el aire, las dos alas generan elevación que compensa el peso, permitiendo a la aeronave continuar en vuelo. Aunque este tipo de aeronave es menos común, su uso para mapeo, agricultura e inspecciones de plantas de aceite & gas presentan algunas ventajas únicas [23].

Fig. 1.5 Drone de ala fija [23].

b) Multi rotor: Los drones multi rotor son los más comunes para hacer fotogrametría y modelos 3D. Usualmente tienen cuatro rotores (cuadricópteros (Fig. 1.6)), pero pueden tener seis u ocho (hexacóptero (Fig. 1.7) y octacóptero (Fig. 1.8)). Una vez en el aire, estos equipos usan propelas fijas para controlar el movimiento de vuelo, y así cambiar la velocidad relativa en la que se desplaza cada uno de los motores, variando el empuje y el torque producido por cada motor. Esto hace que el Drone tenga un excelente nivel de estabilidad, convirtiéndolo en el ideal para la generación de mapas y demás usos en fotogrametría Aérea [23].

Fig. 1.6 Drone multi rotor [23].

Fig. 1.7 Drone hexacóptero [24].

Fig. 1.8 Drone octacóptero [25].

c) Mixtos o híbridos: Este tipo de VANT tienen la capacidad de despegar y aterrizar de manera vertical, ventajas que adquieren por la capacidad de las alas rotatorias; por su parte, el ala fija le otorga alta velocidad. Como es de esperar, este tipo de VANT presenta

una mayor complejidad en su sistema [21]. Estas aeronaves (Fig.1.9) poseen redundancia de mecanismos de sustentación, lo que convierte a esta solución en una opción robusta ante fallos inesperados.

Fig. 1.9 Drone híbrido [22].

Los Drones ofrecen amplias posibilidades de aplicación al sector de la ingeniería civil: inspecciones de infraestructuras, investigación atmosférica, levantamientos topográficos, filmación de películas y fotografía deportiva, entre otros. Los países más avanzados se están impulsando diferentes proyectos de investigación y en las distintas áreas señaladas anteriormente y con ímpetu hacia el área deportiva. Desde la utilización de esto de forma militar, hasta actualmente del área comercial, se han desarrollado distintas formas de poder trabajar con los Drones, ya sea de forma dirigida o autodirigida y de ello depende el tipo de requerimiento en específico y necesidad de quien lo utilice. Hoy en día, se observa el avance tecnológico de estos aparatos para lograr formas de navegación autónomos y eficaces. No obstante, la legislación para el uso de estos aparatos es bastante restrictiva: no está permitido volar sobre zonas pobladas, ni aglomeraciones de personas, también siendo necesario obtener una licencia para poder manejarlos [26]. Un ejemplo de estudio los Drones pueden ayudar en los aspectos técnicos y tácticos del Deporte [27]. En el Deporte colectivo, cada vez con más asiduidad, se utiliza la tecnología para mejorar y optimizar el rendimiento de los jugadores. Ahora en el Deporte individual, aquí el Drone se emplea para analizar la optimización del gesto, un uso cada vez más extendido en el alto rendimiento los cuales darán al experto biomecánico datos excepcionales para su posterior estudio. Otro ejemplo sería la utilización de Drones para el seguimiento visual de autos en videos grabados utilizados para diferentes aspectos, tanto seguridad como entretenimiento [28].

1.2.3 Dron y seguimiento a objetos

En principio para el seguimiento de objetos, se estudia la estimación de las trayectorias realizadas por un conjunto de entidades en movimiento en un escenario determinado, cuya entrada al sistema la constituyen imágenes capturadas de la escena de interés. El reconocimiento de las trayectorias realizadas por los objetos se efectúa en función de las posiciones ocupadas por estos a lo largo de la secuencia de cuadros buscando minimizar una función que determina la desviación de una trayectoria [29] [30], con esto, se tiene un patrón establecido donde el objeto iba y a medida que avanzaba, el Drone lo seguía indirectamente [31]. El seguimiento de hace unos años a la fecha es distinto, se utiliza un sistema de control que permite detectar la posición que tiene el objeto o persona en estudio, mediante la utilización de una cámara puesta en el Drone y la posición que este posee, como se mostraba en [4] para realizar una acción determinada [6], donde se utiliza dos sistemas interconectados, el sistema de control de vuelo del Drone, y una placa procesadora de imagen que suministra los comandos de alto nivel para el seguimiento de los objetos o directamente de la computadora [32]. Lo cual se puede realizar desde un Drone hasta un enjambre utilizando el mismo mecanismo anteriormente hablado [33] [34].

En [4] se basa Diseñar e implementar el sistema de control de posición para un Drone basado en seguimiento de objetos con el Drone Air Drone 2.0 de Parrot cual el objetivo es entregar un sistema de reconocimiento de objetos, basado en el lenguaje C++ y las librerías OpenCV. Además, se entregará el módulo de control de un Drone basado en la plataforma ROS corriendo sobre el sistema operativo Ubuntu. Se cumple el objetivo principal, se tiene un Drone que a partir de las imágenes recibidas por la cámara persigue a un objetivo determinado, con algunas falencias propios del sistema.

Para [5] y [26] tiene como finalidad desarrollar un vehículo no tripulado que de forma autónoma realiza el vuelo siguiendo a un objeto, se ha utilizado dos sistemas interconectados, el sistema de control de vuelo del don, con el software *LibrePilot* y una placa procesadora de imagen que suministra los comandos de alto nivel para el seguimiento de los objetos, con *GNU/Linux* y *OpenCV*. Se ha modificado el sistema *LibrePilot* para interconectar este sistema de control del Drone con el sistema de procesamiento de imagen. Como resultados de la investigación, se logra implementar de buena manera para el procesamiento de datos y de imagen para el seguimiento del objeto.

En [34] se presenta el desarrollo y validación experimental de un sistema cuyo objetivo principal es la estimación de posición de un objeto móvil mediante visión, lo que involucra la detección y seguimiento de este, garantizando cierta tolerancia a cambios de luminosidad. Llegando a la conclusión de que el sistema diseñado proporciona una estimación de la posición de gran calidad con un error absoluto máximo de 5 cm.

Describe en [35] el modelado y control de un Micro Vehículo Submarino Autónomo (AUV), nombrado AR2D2, el cual fue construido con dispositivos y materiales de bajo costo. El principal objetivo es la implementación de un controlador PD saturado para estabilizar el avance y arfada del micro vehículo. Lo cual modela adecuadamente y estabiliza al objeto por medio del procesamiento secuencial de imágenes.

En [36] El objetivo de esta tesis es conseguir una navegación basada en visión fiable y explorar el potencial de los algoritmos del estado del arte moderno en visión por computador para la robótica aérea, y especialmente aplicados a Drones de tipo multi rotor. Logrando tener una buena resolución de la posición de los objetos y aplicado a la inspección de las trayectorias que se situaba, analiza distintas áreas tecnológicas para realizar esta experiencia.

1.2.4 Discusión

Según lo revisado en el estado de arte, se han expuestos las características de manejo de robot para el seguimiento de objetos o personas, la estrategia de control y el tipo de robot utilizado. Todo lo expuesto se realizará una discusión y se planteará la propuesta vista para el desarrollo del proyecto.

El seguimiento de objetos o personas se utiliza un robot (ya sea terrestre, aéreo o acuático) con una cámara montada en él, cuyo objetivo principal siempre es captar la imagen y procesar los datos para luego realizar el control de movimientos del robot. Pero la diferencia radica en cómo se procesan estos datos que pueden ser directamente de un computador [4] [34] o más bien desde una tarjeta externa que contenga todos los algoritmos para generar los cálculos de datos que se requieran. También la diferencia va en el tipo de lenguaje utilizado, como es Python, C++ o Matlab o ROS que es un *framework* basado en el lenguaje Python o C++ [4] [5] [26], o inclusive más allá, utilizar algún *softwares* desarrollado por las empresas que se dedican a construir sus propios robots [35]. El tipo de lenguaje para la adquisición de imágenes y el procesamiento dependerá de que tan asociado al lenguaje se esté o que tan rápido se interprete.

En cuanto al tipo de robot a utilizar dependerá netamente de los objetivos que se tengan en desarrollo, aunque si es para un seguimiento en ruta, o de un área deportiva, o una planta con dificultades para llegar, tiene mejor libertad de movimiento el aéreo y no interviene en las actividades ajenas a las que se sitúe la observación .

Para el desarrollo del algoritmo de control del robot seleccionado se ven distintas formas de trabajar y dependerá del tipo de robot utilizado, si es un terrestre, este tendrá que ver un algoritmo que se mueva o detenga según se mueva el objeto en estudio [34] , en cambio sí es un robot aéreo o acuático, tendrán que desarrollar un algoritmo que logre moverse a medida que el objeto se mueva y además incluir en su algoritmo una forma de lograr estabilizarlo en una altura o profundidad determinada, dependiendo del caso [35] [4] [36]. Con relación al tipo de robot a trabajar, actualmente hay algunos de valor comercial, que ya tienen algunas características de estabilización incluidas, lo que permitiría enfocarse solo en el seguimiento (esto si no se desea construir desde cero).

En resumen, lo que se desea lograr en este proyecto es el seguimiento de objetos y/o personas, de manera que no se necesite realizar una inspección de la trayectoria con anterioridad, sino más bien mantener una distancia aproximadamente constante a medida que se vaya moviendo el objeto o persona en estudio, como se muestra en los trabajos analizados con anterioridad. En el procesamiento de imagen se tomarán el lenguaje Python y con ello, se realizará el algoritmo para procesar las imágenes adquiridas y poder calcular los datos necesitados (estimación de distancia y el centro de la figura) que serán enviados al siguiente proceso que es el de control. En algoritmo de control, se trabajará en un microprocesador que adquiera los datos de posición del objetivo y este las interprete, así pueda generar el(los) movimiento(s) para mantener al objetivo en el rango cercano al centro óptico de la cámara y una distancia constante. En esta situación, seleccionamos un robot aéreo ya que este puede supervisar desde el aire, tiene mayor grado de libertad e interactúa sin intervenir en el escenario donde se realiza el testeo (por ejemplo, en un campeonato de atletismo o realizando el seguimiento de algún vehículo en carretera, entre otros), para ello seleccionamos el Drone multi rotor cuadricóptero para el proyecto, ya que su valor comercial es menor que los otros tipos de Drone multi rotor existentes y además, por su mejor estabilidad lo hace apropiado para capturar imágenes para el procesamiento necesitado.

1.3 Hipótesis

Es posible diseñar un sistema de seguimiento continuo a objetos por medio de un Drone utilizando Python y Arduino para el procesamiento de imagen y control de vuelo.

1.4 Objetivos

1.4.1 Objetivo general

Diseñar un equipo montable integrando tecnología, que sea capaz de medir y analizar la ubicación del objeto para realizar un seguimiento automático sin presencia de alguien que lo manipule.

1.4.2 Objetivos específicos

- Diseñar un software para procesar imágenes obtenidas a partir de una cámara.
- Integrar tecnología de seguimiento en Drone.
- Diseñar un algoritmo de control para el seguimiento de objetos y/o personas.
- Elaborar reporte de la construcción del equipo respecto el avance actual de la robótica según la investigación realizada.

1.5 Alcances y Limitaciones

1.5.1 Alcances

- Dron: este deberá tener las condiciones necesarias para soportar el peso al momento de generar el vuelo con el equipo en cuestión, además de mantener la posición cuando no se realice el seguimiento del atleta.
- Control: Para el control de vuelo se analiza el seguimiento del objeto y/o persona mediante un algoritmo de control autónomo continuo.
- Generación de algoritmo que muestre los análisis de la cámara.
- Del vuelo: el vuelo será autónomo con seguimiento en tiempo real, por lo que se deberá generar un control que genere el vínculo con el objeto y/o persona.
- Mediciones: el principal objetivo será medir la posición para mantener una distancia fija y buena nitidez del estudio en cuestión.

1.5.2 Limitaciones

- Tiempo: El uso del equipo estará limitado por el tiempo de carga del Drone, lo que sitúa a trabajos de cortos tiempos.
- El estudio al aire libre no contempla vientos, lluvia, solo debe estar en un escenario ideal, lo que en un principio se realizará en un lugar cerrado que contemple estas características.
- Mediciones: Los cálculos de estimación de distancia y el centro de la figura captada serán del algoritmo de procesamiento de imagen, mientras la de mantener al objeto o persona siempre en visual de la cámara será del algoritmo de control.
- La cámara deberá tener una imagen de excelente calidad y alta resolución dado que la distancia a considerar entre el Drone y el objeto debe mantener siempre una distancia para no distraer en la actividad ejecutada y permita obtener un buen análisis del sistema.
- Cantidad: Por ser prototipo, se realizará el estudio solo a una persona u objeto por investigación, que también está limitado a la memoria del microprocesador que se llegase a utilizar.
- Lugar de estudio: Se tiene posibilidad en un recinto cerrado (preferencia para no tener perturbaciones en el sistema) o un recinto abierto en un horario conveniente por efecto de vientos y sol.

1.6 Metodología

Debemos tener un conocimiento básico de los softwares para proceder a desarrollar el sistema total del equipo haciendo énfasis a los algoritmos tanto de control de vuelo y procesamiento adecuado de las imágenes para su análisis respectivo como se verá a continuación:

1.6.1 Estudio teórico sobre procesamiento de imagen

Se estudiará en base a revistas y/o artículos científicos de exploradores académicos como Scielo, IEEE-Xplore y memorias de título de universidades señaladas en referencias. Haciendo énfasis a la adquisición y aplicación de técnicas de análisis de secuencias de imágenes a termografías de diferentes procesos para nuestro objetivo planteado.

1.6.2 Estudio teórico sobre Drone y el seguimiento de personas

Se analizará preferentemente algoritmos trabajados en artículos científicos y especializado en procesamiento de imagen para gestionar a lo necesario para este proyecto. Anexo a esto, se iniciarán vuelos con Drones que se tengan al alcance (simulaciones y de la universidad) para entender la mecanización de vuelo y explayar de mejor forma los algoritmos en el uso correspondiente. Una vez realizado, se tomará como modelo el Drone disponible en la facultad, para dicha modificación de procesamiento imagen en el seguimiento del objeto o persona en cuestión.

1.6.3 Diseño y creación de estructura para las cámaras de procesamiento de imagen

Se diseñará una estructura para el montaje para las cámaras necesarias para el procesamiento de imagen que se es requerido. Para esto se utilizará un *software* de diseño como inventor o SolidWorks, con su respectivo análisis de fallas según el material que se tenga a disposición poder desarrollarlo.

1.6.4 Implementación sobre algoritmos desarrollados, montaje y puesta en marcha

Luego de tener todos los recursos disponibles para montar en el Drone (algoritmos, estructura, análisis de procesamiento de imagen) se realizará la experimentación del seguimiento continuo y de la lectura del objeto u persona, verificando si los objetivos específicos se cumplen del todo, para luego generar una recopilación de datos y afinar detalles si se es necesario.

1.7 Espacios Utilizados

Contemplando la actual pandemia global, nuestro proyecto se secciono en 2 grupos, denominados “teórico y práctico”. Lo que infiere a que la primera sección de este proyecto solo contempla el desarrollo teórico, simulaciones y desarrollo de las estructuras, lo cual el área de trabajo será en el lugar de estudio que se tenga en la residencia. Por lo que la segunda sección procede a experimentar físicamente el proyecto, lo que se necesitaría un espacio suficiente para volar libremente el Drone.

Capítulo 2 Implementación

2.1 Introducción

Para este capítulo se detallan los componentes que se utilizan en el proyecto, tanto para adquirir y procesar la imagen , los utilizados para enviar las señales de control al Drone y de los *softwares* utilizados en el desarrollo de los algoritmos. El primer algoritmo es de procesamiento de imagen cuya finalidad es calcular el área y el centro de un objeto o persona que es captada por la cámara. El siguiente algoritmo es de control de los movimientos del Drone, cuyo objetivo es realizar el seguimiento continuo, manteniendo al objetivo en el centro óptico de la cámara y a una distancia constante (según los datos que se adquieren del procesamiento de imagen). Además, se explica la comunicación utilizada para la transmisión de datos desde algoritmo de procesamiento de imagen hacia el algoritmo de control, y finalmente describir el tipo de conexionado realizado entre la salida PWM de la tarjeta Arduino y el mando manual del Drone, que es encargada de enviar las señales que el algoritmo de control genera para lograr el control autónomo del Drone del proyecto.

2.2 Componentes utilizados

2.2.1 Drone

El modelo X8HG es un Drone comercial de la serie Syma. Una de sus características es que el sistema de mantenimiento de altitud nos permite mantener estable el Drone a una altura determinada [37]. Este tendrá montado la cámara FPV y el transmisor para la comunicación inalámbrica entre la cámara y el computador.

Fig. 2.1 Drone Syma modelo X8HG [31].

2.2.2 Computador

El computador con que se trabaja es un HP ProBook 430 G5 (Fig. 2.2). Este se utilizará para recibir la imagen de la cámara para poder procesarla y con ello, poder calcular los valores que se necesitan enviar al algoritmo de control así logrando mantener al objeto de estudio en el centro de la cámara y a una distancia constante. Este computador tiene un procesador Intel Core i3-7100U de 2,4 GHz de velocidad, con 2 núcleos y gráficos Intel HD 620 [38]..

Fig. 2.2 Notebook HP ProBook 430 g5 [38].

2.2.3 Transmisión y recepción de la señal de video

Un transmisor (Fig. 2.3) y receptor FPV (Fig. 2.4) no es más que un transmisor de vídeo normalmente analógico o digital, este componente se encarga de recoger la imagen de la cámara FPV y enviarla de forma inalámbrica a un receptor de vídeo. El FPV se refiere a Vista en Primera Persona, un tipo de vuelo muy habitual en los Drones y aviones RC [39].

Fig. 2.3 Transmisor TX TS832 FPV (Referencial) [29].

Transmisor TX TS832 especificaciones:

- Entrada de energía: 7-12.

- Potencia de transmisión: 600 mW.
- Ganancia de la antena: 2dbm.
- Corriente de trabajo: 280mA.
- Ancho de banda de vídeo: 8 MHz.
- Ancho de banda de audio: 6.5 MHz.
- Peso: 7g.
- Dimensión: 23 x 25 x 7.7 mm (sin incluir la antena).

Fig. 2.4 Receptor CX RC832 FPV (referencial) [29].

Receptor CX RC832 especificaciones:

- Frecuencia de trabajo: ISM de 5,8 GHz.
- Fuente de alimentación: DC 12V.
- Consumo de corriente (energía): 200 mA, máx.
- Antena Impedancia de entrada: 500ohm Typ.
- Conector de antena: RP-SMA.
- Ganancia de la antena: 2 dBi.
- Nivel de salida de vídeo: 1,0 Vp-p, 75Ω Typ.
- Nivel de salida de audio: 1,0 Vp-p, Typ 10 k.
- Portadora de audio: 6.5MHz.
- Tipo estándar: NTSC / PAL.
- Dimensión: 80x65x15mm.
- Peso: 85 g.

2.2.4 Arduino

El Arduino Mega 2560 es una placa de desarrollo basada en el microcontrolador ATmega2560. Tiene 54 entradas/salidas digitales (de las cuales 15 pueden ser usadas como

salidas PWM), 16 entradas analógicas, 4 UARTs, un cristal de 16Mhz, conexión USB, jack para alimentación DC, conector ICSP, y un botón de reinicio [40].

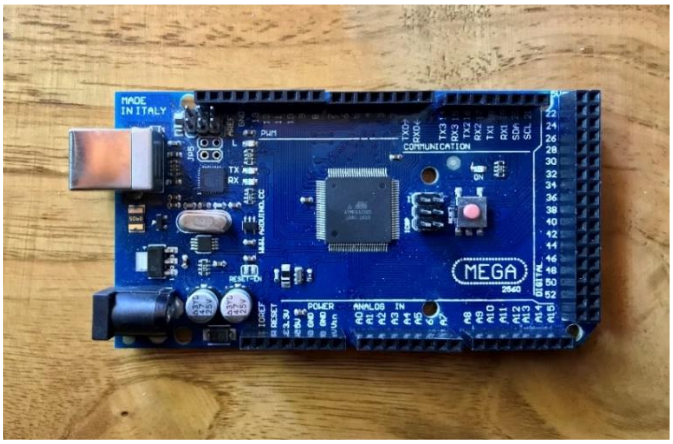

Fig. 2.5 Arduino Mega 2560 utilizado para el control del Drone.

Esta placa Arduino (Fig. 2.5), tendrá el algoritmo de control que fue desarrollado en el programa de Arduino y subido a la placa, el cual recibe los datos del centro óptico de la cámara y del área del objeto o persona en estudio generados anteriormente por el algoritmo de procesamiento de imágenes en el computador. Estos datos recibidos los procesa y genera las señales correspondientes que serán enviadas por la salida de la PWM al mando manual del Drone para realizar los movimientos requeridos del Drone.

2.3 Softwares

El algoritmo de procesamiento de imágenes será desarrollado en el lenguaje Python, donde se utilizan las bibliotecas de OpenCV, Numpy y Time. Todo esto, se verá reflejado en *Visual Studio Code*, que es un editor de código optimizado que admite operaciones de desarrollo como depuración, ejecución de tareas y control de versiones. Su objetivo es proporcionar las herramientas que un desarrollador necesita para un ciclo rápido de generación y depuración de código y deja flujos de trabajo más complejos a IDE más completos [41].

El algoritmo de control será desarrollado en Arduino. Arduino es una plataforma de desarrollo basada en una placa electrónica de hardware libre que incorpora un microcontrolador reprogramable y una serie de pines hembra. Estos permiten establecer conexiones entre el microcontrolador y los diferentes sensores y actuadores de una manera

muy sencilla. Su lenguaje de programación basado en C++ es de fácil compresión. C++ permite una entrada sencilla a los nuevos programadores y a la vez con una capacidad tan grande, que los programadores más avanzados pueden exprimir todo el potencial de su lenguaje y adaptarlo a cualquier situación. Se puede instalar y ejecutar en sistemas operativos Windows, Mac OS y Linux [42].

2.4 Implementación

Como se mencionaba en la sección 2.3, el lenguaje de trabajo será basado en Python, para el procesamiento de imagen, y la librería Serial ayudará para establecer la comunicación con Arduino que se encargará del control de los movimientos del Drone. Logrando centrar siempre a la persona u objeto en el centro óptico de la cámara con una distancia constante.

La comunicación en serial se define como una interfaz de comunicación de datos digitales que nos permite establecer transferencia de información entre varios dispositivos. Esto nos va a permitir que *Visual Studio Code* pueda comunicarse con Arduino, así transmitir los datos calculados hacia la placa Arduino que tiene el algoritmo de control para los movimientos del Drone [43].

El proceso general del proyecto (Fig. 2.6) inicia por la adquisición de la imagen por medio del sensor de imagen, en dirección donde se encuentra el sujeto en estudio. Luego esa imagen una vez ingresada al computador se procesa en VS Code para calcular los valores del centro y el área (de esta forma determinamos proporcionalmente su distancia). Al tener los datos se transmiten mediante la comunicación serial hacia el Arduino que se encarga de procesar estos datos y generar las señales correspondientes al mando de Drone. De esta manera, modificamos continuamente la posición del UAV para mantener siempre visualizado a la persona u objeto mientras se mueve.

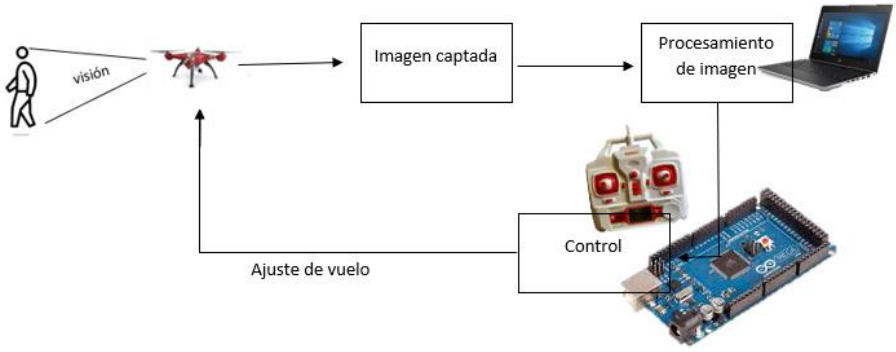

Fig. 2.6 Diagrama de bloques general del proyecto.

2.4.1 Conexionado del mando de Drone y Arduino

Para llevar a cabo el control, se estudian 2 casos donde se puede manipular el Drone, sin necesidad de manipular directamente con el joystick. En primer lugar, al investigar acerca del tema fue posible encontrar un post de un blog en el cual el autor consigue intervenir la conexión entre un Drone y su emisora utilizando un Arduino y así controlarlo con éste [44]. Y el otro proceso (que fue el utilizado) se interviene el control remoto (Fig. 2.7) que viene junto al Drone. En su interior, se sueldan cables en los bornes de conexión de los análogos como se muestra en las Fig. 2.8 y Fig. 2.9, así para conectarlo a los pines del microcontrolador utilizado. Esto permite gracias al algoritmo diseñado, enviar las señales desde el microcontrolador a los análogos del mando manual, permitiendo ejecutar los movimientos del Drone. La Fig. 2.10 muestra las conexiones realizadas que van desde las salidas del PWM de la tarjeta Arduino al mando. Esta es una manera más rápida y económica de poder controlar el Drone, ya que el primer estudio, necesita de un radio transmisor que pueda intervenir en la señal en que opera el joystick y el Drone [44].

Fig. 2.7 Control remoto del Drone Syma X8HG.

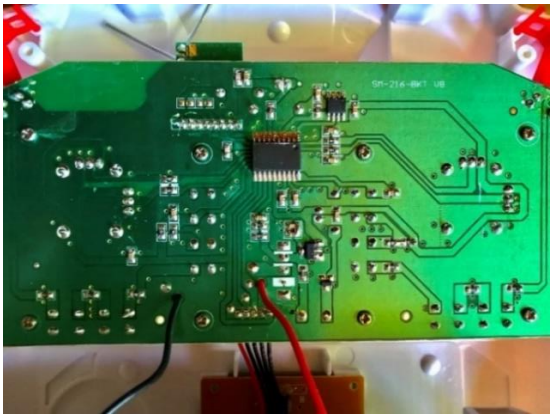

Fig. 2.8 Circuito del control remoto del Drone.

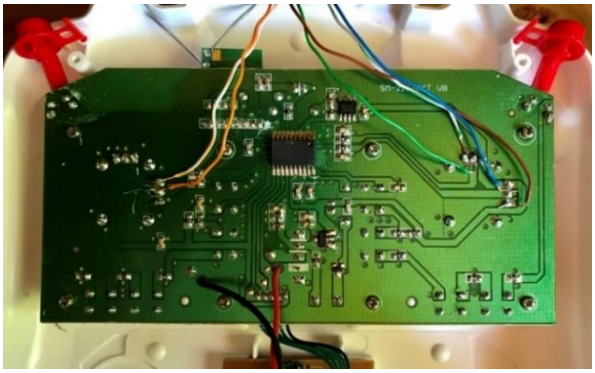

Fig. 2.9 Conexionado de los pines del análogo hacia el Arduino.

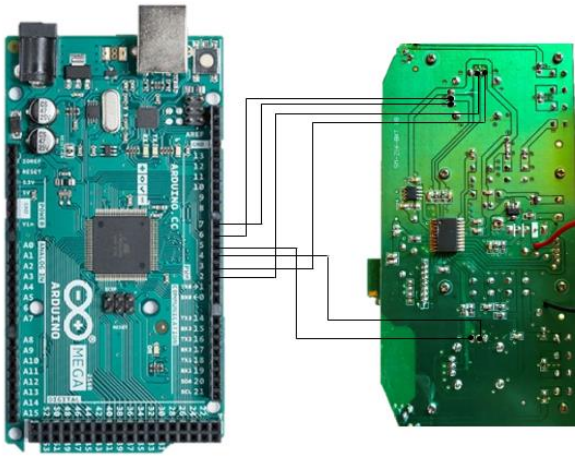

Fig. 2.10 Diagrama de conexionado de los pines del análogo del control hacia el Arduino.

Un análogo del mando son 2 potenciómetros que se utilizan para los movimientos necesarios del Drone. En la identificación del voltaje máximo que pueden recibir los potenciómetros sin dañar la placa, se realiza análisis con el multímetro (con el control

encendido) obteniendo un valor de 4.5 [V], que es cuando el análogo está al máximo de su movimiento. Lo que permite poder conectar a los pines del PWM que su valor máximo de salida es de 5 [V] (Fig. 2.10). Para detectar los movimientos que son necesarios (arriba, abajo, avanzar, retroceder, giro derecho y giro izquierdo), se procede a realizar una conexión con 2 cables (uno alimentado por 3.3 [V] y el otro a GND) hacia los pines del potenciómetro, para testear las direcciones que se envía el voltaje.

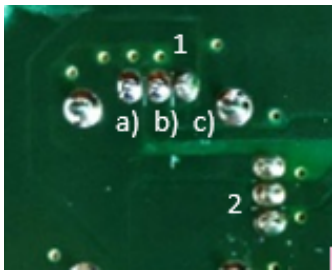

Fig. 2.11 Potenciómetros del análogo derecho: 1) potenciómetro que desplaza hacia derecha e izquierda a) fijo b) variable c) fijo. 2)potenciómetro para avanzar o retroceder.

Fig. 2.12 Potenciómetros del análogo izquierdo:3) giro izquierda o derecha y 4) arriba o abajo.

Cada potenciómetro (Fig. 2.11 y Fig. 2.12) tiene 2 funciones, de cual (puede ser el de subir o bajar, izquierda o derecha) dependerá netamente si la alimentación va por un fijo o por el variable, como ejemplo se toma el potenciómetro 4, teniendo el voltaje en un fijo y el GND en el variable, este bajará la velocidad de las hélices del Drone, permitiendo que vaya hacia abajo. Pero si es viceversa este subirá la velocidad de las hélices del Drone, permitiendo que se eleve.

En el montaje del transmisor y de la cámara FPV en el Drone, se reajusta la estructura diseñada para una cámara GO pro, lo que logra dejar fijos estos elementos y no tenga

problemas en la toma de imágenes. Estos necesitan una alimentación externa que se adjunta al soporte de la batería del Drone, visto en la sección de componentes.

Teniendo en conocimiento de los *softwares* que se utilizan y la implementación puesta en el proyecto. Para los siguientes capítulos queda explicar detalladamente el desarrollo del algoritmo del procesamiento de imágenes (que se verá a continuación) y el algoritmo de control de la posición del Drone. Luego se detalla los resultados experimentales realizados en campo abierto donde se verifica la comunicación de ambos algoritmos y del procesamiento de datos que se genera para poder efectuar los movimientos del Drone.

Capítulo 3 Procesamiento de imagen

3.1 Introducción

En este capítulo se explica el algoritmo de procesamiento de imágenes que se desarrolló en el proyecto cuyo objetivo es tomar la imagen de la cámara y procesarla, con ello, logra calcular el área y el centro de la figura captada, que serán enviados al algoritmo de control con la finalidad de que realice las comparaciones y logre mover al Drone según sea requerido. En primer lugar, se inicia explicando el tipo de lenguaje utilizado y las librerías que la complementan de este proyecto. Así mismo, en este capítulo se muestra el diagrama de flujo general, donde se explica los bloques más importantes de él, para luego mostrar la forma del cálculo del centro de la figura procesada. También se describe la calibración de la cámara que es requerido para obtener el centro óptico, que se utiliza como comparador en el algoritmo de control desarrollado.

3.2 Lenguaje y librerías utilizadas

El lenguaje utilizado para el procesamiento de imágenes es Python, apoyado por las bibliotecas de OpenCV y Numpy, que serán definidas a continuación. Se utiliza Python por ser un lenguaje de libre acceso, de rápido entendimiento y por poseer una amplia selección de librerías que facilitan la creación de algoritmos, para las distintas áreas de trabajo.

En conjunto a Python, se trabaja con OpenCV, que es una librería software *open-source* de visión artificial y *machine learning* [45]. Su librería permite identificar objetos, caras, clasificar acciones humanas en vídeo, hacer *tracking* de movimientos de objetos, extraer modelos 3D, encontrar imágenes similares, eliminar ojos rojos, seguir el movimiento de los ojos.

La otra biblioteca utilizada para procesar los datos obtenidos, definir los contornos, calcular el área y el centro de la figura donde se encuentra el color detectado es Numpy. Numpy es una biblioteca para Python que facilita el trabajo con *arrays* (vectores y matrices), un tipo de dato estructurado muy utilizado en análisis de datos, en informática científica y en el área del aprendizaje automático (*machine learning*). Numpy permite declarar *arrays* con distintas dimensiones capaces de albergar gran cantidad de datos del mismo tipo y relacionados entre sí. Además, provee numerosos métodos para manipular los *arrays* y para acceder a la información y procesarla de forma muy eficiente [46].

El algoritmo implementado para el procesamiento de imágenes se visualiza en el diagrama de flujo de la Fig. 3.1, donde se explica los procesos más relevantes a la hora de poner en marcha el proyecto.

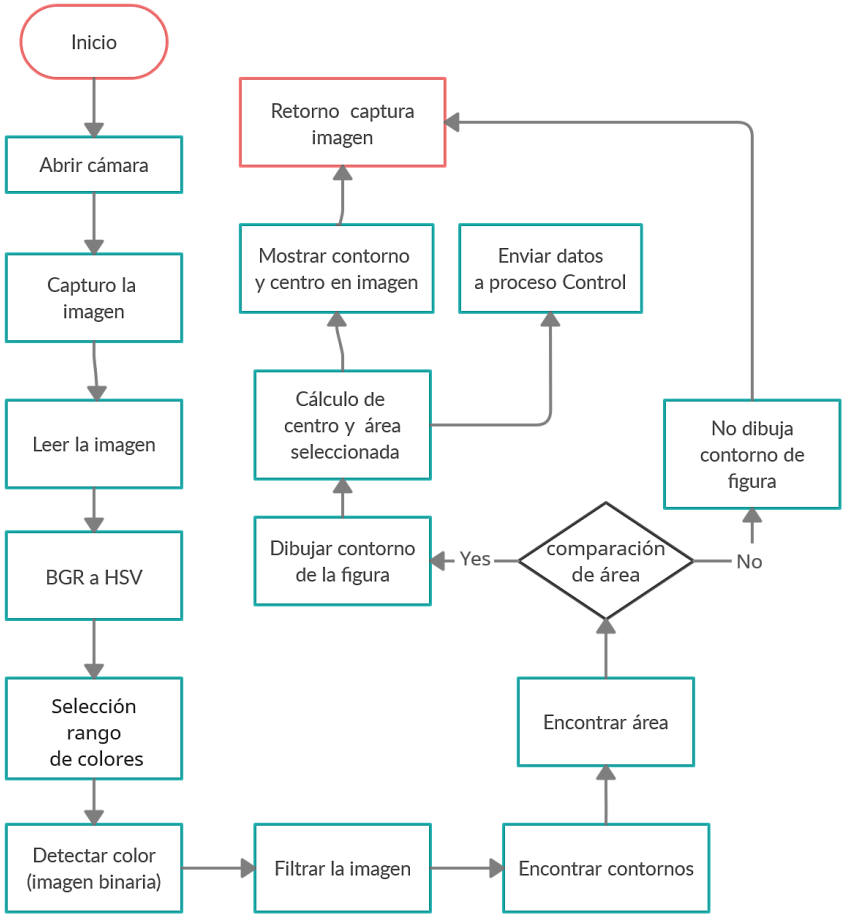

Fig. 3.1 Diagrama de flujo de procesamiento de imagen completo.

3.3 Imagen por procesar

El sensor de imagen será la Webcam del computador HP ProBook 430 g5, siendo esta cámara web HD [38] cuyo propósito mostrar la secuencia de imágenes que serán procesadas más adelante (Fig. 3.2). En esta sección siempre se deja abierta la cámara, pero con el comando *cap.read()*, se realiza la lectura y capta lo que tenga en visión, logrando con esto trabajar con la imagen y así obtener los datos necesitados. Los tiempos de adquisición de imagen de la cámara es respecto a cuanto se demora en avanzar en el algoritmo de procesamiento hasta volver al inicio del programa, siendo valores de milisegundos.

Fig. 3.2 Notebook HP ProBook 430 g5 con zoom en cámara web.

3.4 Detección de colores

La detección de color es la capacidad de un sensor o algoritmo de distinguir colores a partir de la extracción de información de la luz. La manera básica de detectar el color consiste en captar la luz incidente en un sensor. Este, mediante un conjunto de celdas de fotones que forman una matriz de puntos, uno por cada píxel, es capaz de medir la cantidad de luz llegada a cada uno de estos, produciendo una corriente eléctrica que varía en función de la intensidad de luz recibida. Una vez se ha medido la cantidad de luz se procede a la detección de colores. El reconocimiento de color para determinar qué colores aparecen en una imagen, captada o que se está captando, se realiza mediante los modelos de color, ya que estos hacen posible la representación de los colores de forma numérica. Los modelos de color (por ejemplo, RGB, YUV o HSV) permiten crear filtros para discriminar colores [47].

El espacio de color HSV (Hue, Saturation, Value / Matiz, Saturación, Brillo), posee 3 componentes (Fig. 3.3), similar al espacio de color RGB. Para determinar un color se centra en el componente H que corresponde al matiz [48].

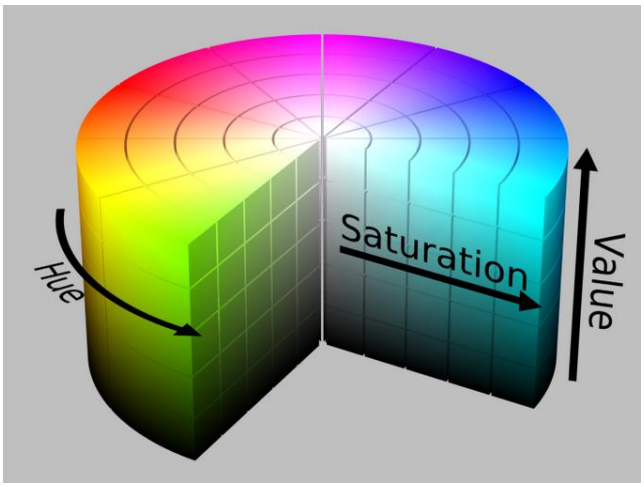

Fig. 3.3 Espacio de color HSV [49].

Los códigos utilizados para este programa son *cv2.cvtColor, cv2.COLOR_BGR2HSV, np.array, cv2.inRange, cv2.inRange, cv2.add, cv2.imshow*.

Una representación desde que se abre la cámara, hasta que genera los datos necesitados para enviar al proceso de control se muestra en la Fig. 3.1 que es el diagrama de flujo de todo el procesamiento de imagen generado en Python. Los procesos son siempre secuenciales, hasta el punto donde llega a la sección de "comparación de área", ya que aquí se trabaja con la imagen binaria para dejar solamente la figura del objeto o persona analizada y no áreas más pequeñas a esta.

3.5 Transformar de BGR a HSV

Por defecto OpenCV lee a las imágenes o fotogramas en BGR, por ello es necesario transformarlas al espacio de color HSV. Para ello se utilizan funciones que nos proporciona la biblioteca de OpenCV.

RGB (Red Green Blue / Rojo Verde Azul), es uno de los espacios de color más utilizado en cuanto a imágenes se trata y el que usa por defecto OpenCV al momento de leer una imagen o dibujar figuras geométricas, por ello es de vital importancia conocer cómo se maneja este espacio de color en esta librería. HSV a diferencia del modelo RGB ampliamente usado en los monitores, televisores, etc., si bien las coordenadas de aquel son euclidianas, el color HSV sigue una representación más parecida a las coordenadas cilíndricas (Fig. 3.3). Además, es una representación más cercana a la forma en que los humanos perciben los colores y sus propiedades, pues se agrupan las tonalidades de color, lo cual es distinto al caso

RGB donde los colores no están necesariamente tan agrupados, es por ello la elección de HSV [50] [51].

3.6 Determinación de los rangos donde se encuentra el color

En este ejemplo realizado extraído de [48] se trabaja con los rangos de colores rojo, lo cual se adquieren los rangos del HSV de la Fig. 3.4.

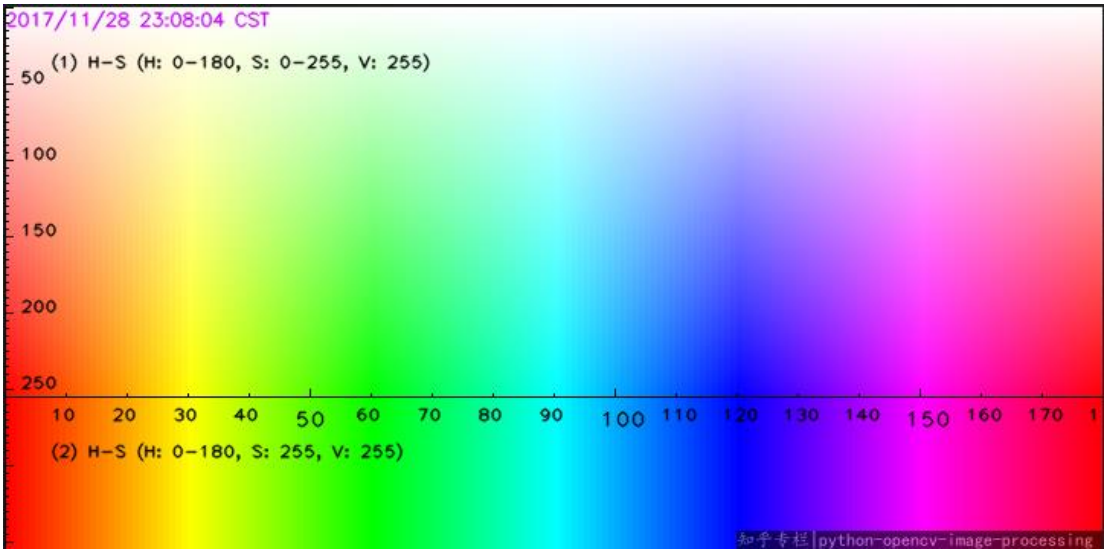

Fig. 3.4 Rango de Colores de HSV [52].

Como se ve en la Fig. 3.4 el componente en H va de 0 a179, y en este pasan colores de rojo, naranja, amarillo, verde, azul, violeta y nuevamente rojo, entonces se determinan 2 rangos para el color rojo que está presente al principio y al final, el primer rango va de 0 a 8, y el segundo de 175 a 179 en H, mientras que para los componentes S de 100 a 255, y V de 20 a 255, para ambos rangos.

3.7 Visualización

En primer lugar, se capta la imagen (Fig. 3.5). Luego se utiliza una detección binaria de la imagen, es decir, si es que detecta o no el color señalado (en este caso rojo). Si se encuentra en el rango de aceptación lo reflejado será de color blanco y mientras que el color negro mostrará el lugar donde no están presentes estos rangos (Para este caso se realiza en 2 secciones de detección, para luego juntarlas y mostrarlas como refiere la Fig. 3.6). Para después realizar la transformación de que el color blanco, llegue a ser al color rojo como muestra la Fig. 3.7.

Fig. 3.5 Objeto a detectar con el rango de color señalado.

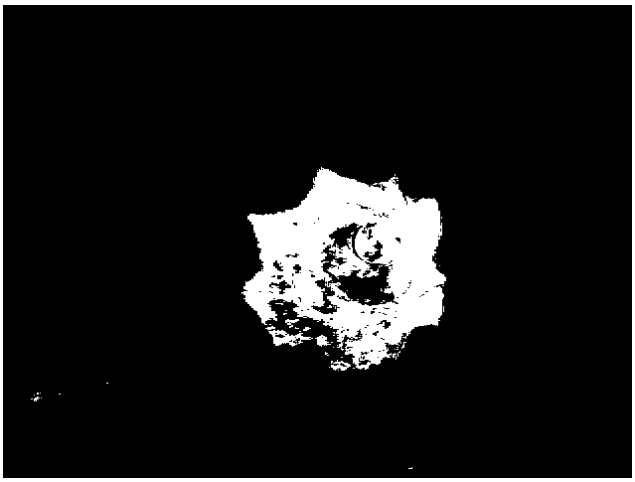

Fig. 3.6 Detección binaria del objeto (detecta o no detecta color).

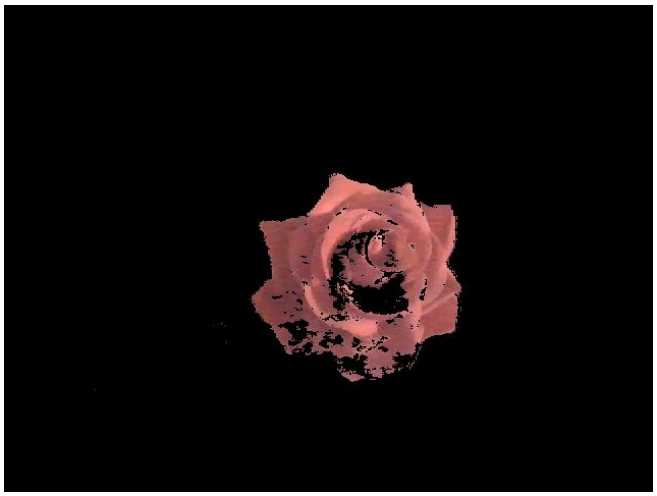

Fig. 3.7 Proceso final con detección de color, mostrando el color analizado.

Para desarrollar este algoritmo, se utiliza el computador y la cámara Web que posee este. Si se observa la Fig. 3.6 y Fig. 3.7, se puede apreciar puntos negros dentro del objeto de estudio, esto debido a la mayor iluminación que refleja esta área, lo que añadiendo un filtro logra disminuir ese ruido existente. Pero también, demuestra que la iluminación debe estar controlada adecuadamente para que la cámara pueda detectar los objetos, como muestra las Fig. 3.8, Fig. 3.9 y Fig. 3.10, que se realiza el mismo análisis, pero con mayor iluminación en el sector.

Fig. 3.8 Detección de objeto con mayor iluminación en el entorno.

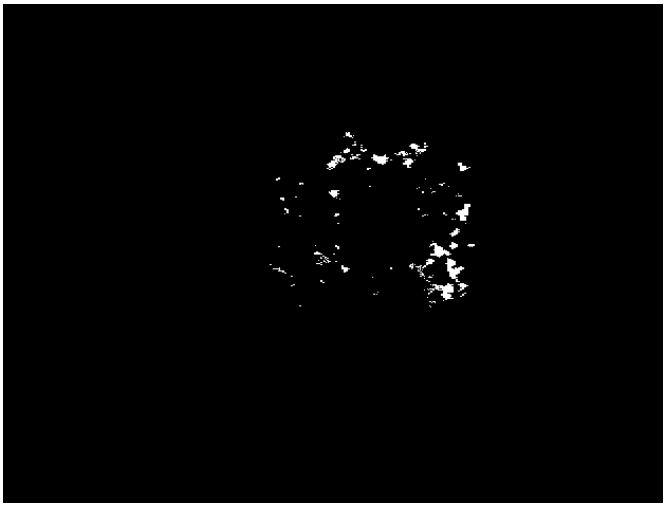

Fig. 3.9 Detección binaria del color con mayor iluminación.

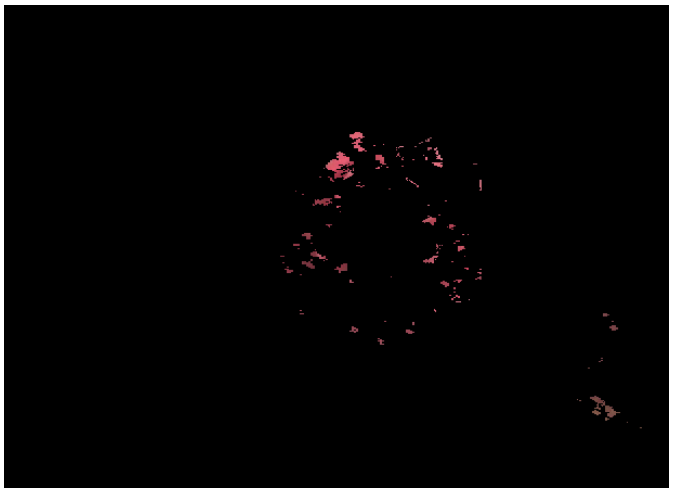

Fig. 3.10 Proceso final mostrando el color analizado en un campo de mayor iluminación.

3.8 Tracking

Considere una secuencia de vídeo tomada por una cámara de mano que representa varios objetos que se mueven dentro y fuera del campo de visión de la cámara. Dado un cuadro delimitador que define el objeto de interés en un solo fotograma, el objetivo es determinar automáticamente el cuadro delimitador del objeto o indicar que el objeto no es visible en cada fotograma que sigue. La secuencia de vídeo se va a procesar a velocidad de fotogramas y el proceso debe ejecutarse indefinidamente. Nos referimos a esta tarea como seguimiento a largo plazo o Tracking [53].

El seguimiento de objetos (Tracking) [54] es el proceso de estimar en el tiempo la ubicación de uno o más objetos móviles mediante el uso de una cámara. La rápida mejora en cuanto a calidad y resolución de los sensores de imagen, juntamente con el dramático incremento en cuanto a la potencia de cálculo en la última década, ha favorecido la creación de nuevos algoritmos y aplicaciones mediante el seguimiento de objetos. Al modelar la relación entre el aspecto del objeto de interés y el valor de los píxeles correspondientes, un seguidor de objetos valora la ubicación de este objeto en el tiempo. La relación entre el objeto y la proyección de su imagen es muy compleja y puede depender de más factores que no sean solamente la posición del objeto, lo que implica que el seguimiento de objetos sea una tarea difícil.

Para esta sección se utiliza la librería de OpenCV en el lenguaje de Python donde se rediseñó un algoritmo que logra hacer el seguimiento dentro del rango de la cámara del

computador (a modo ejemplo). La elección del objeto con el que se trabaja debe ser definido en un comienzo del programa y puede ser escogido por afinidad del usuario.

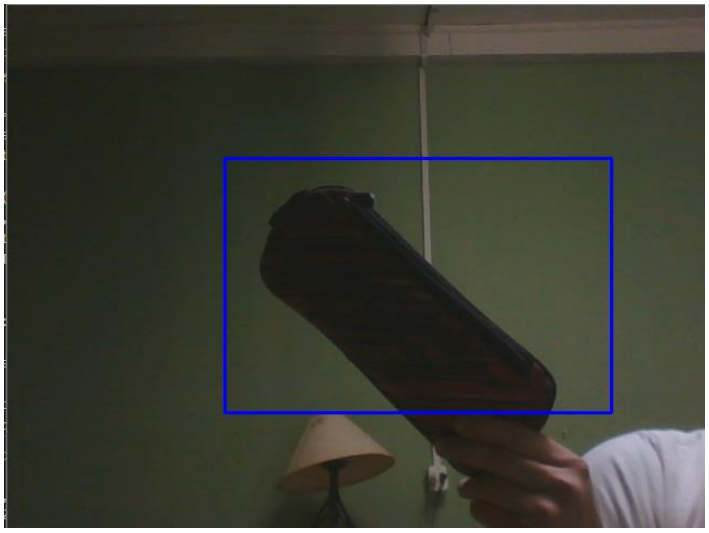

Fig. 3.11 Selección del objeto a seguir.

En primer lugar, al iniciar el algoritmo este toma una fotografía para realizar la selección, el cuadro de color azul es realizado con el ratón del computador donde encierra el objeto como muestra la Fig. 3.11. Después de eso, este cambia de color rosado señalando que está seleccionado y listo para el seguimiento (Fig. 3.12).

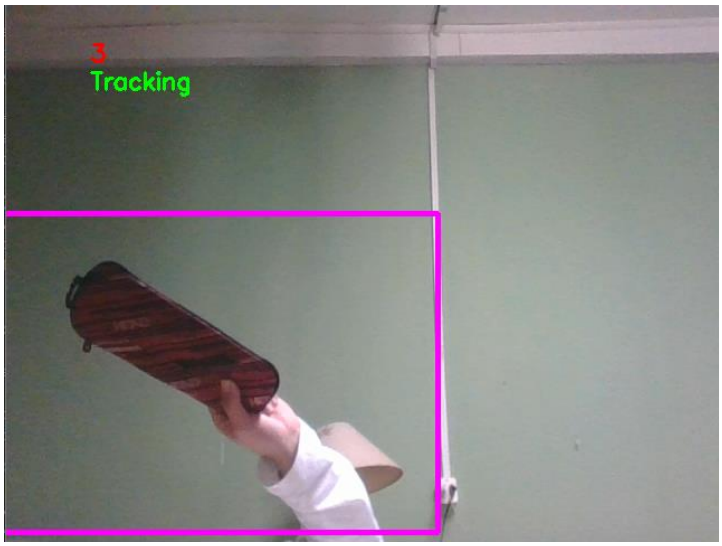

Fig. 3.12 Seguimiento del objeto mediante Tracking.

Fig. 3.13 Acercamiento del objeto en cuestión.

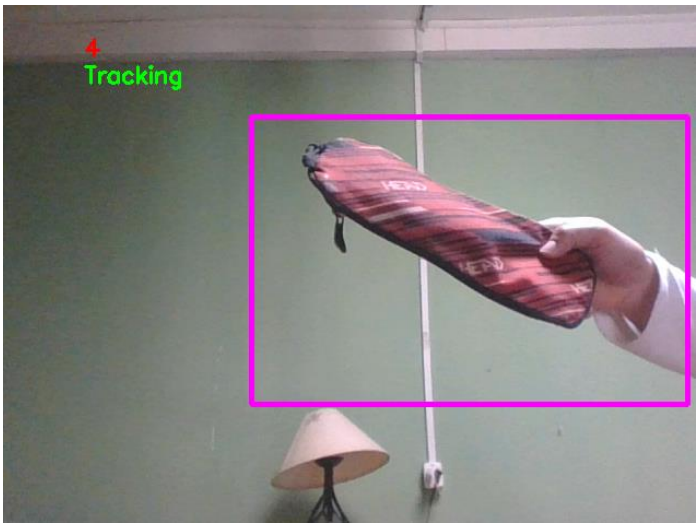

Fig. 3.14 Movimiento del objeto para verificación del seguimiento adecuado.

En las Fig. 3.12, Fig. 3.13 y Fig. 3.14, se muestra a modo ejemplo el modelo del algoritmo rediseñado de Tracking, donde realiza el seguimiento del objeto, independiente del acercamiento o alejamiento de este. Cabe destacar, que cualquier objeto que se tenga frente la cámara, al inicio del programa se puede seleccionar y realizar el seguimiento de su desplazamiento.

3.9 Calibración de la cámara

El modelo de perspectiva es el más utilizado en el análisis de sistemas de formación imágenes. De acuerdo con dicho modelo, para un sistema óptico con una distancia focal f, un punto P del mundo real con coordenadas (X, Y, Z) produce un punto p en el plano imagen con coordenadas (x, y) definidas como [7] muestra la ecuación 3.1.

$$\begin{bmatrix} \lambda x \\ \lambda y \\ \lambda \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{3.1}$$

Aquí, los vectores de posición se representan en coordenadas proyectivas u homogéneas, en donde el factor de escala para los puntos en el mundo real es unitario y λ para los puntos en el plano imagen. En este modelo, tanto P como p están representados en el referencial de la cámara. En la ecuación (3.1) los puntos imagen están expresados en unidades métricas. Pero, como se necesita que estén expresados en píxeles, se requerirá de un factor de escala en cada dirección (k_x, k_y) y, como deben expresar con respecto al punto principal de la imagen y no con respecto al referencial de la cámara, se requerirá de una translación de componentes (C_x, C_y). Lo cual, usando coordenadas proyectivas, se representa como la ecuación 3.2. [55].

$$\begin{bmatrix} \lambda x_p \\ \lambda y_p \\ \lambda \end{bmatrix} = \begin{bmatrix} k_x & 0 & C_x \\ 0 & k_y & C_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \lambda x \\ \lambda y \\ \lambda \end{bmatrix} \tag{3.2}$$

Las constantes k_x, k_y, C_x y C_x se conocen como parámetros intrínsecos (son específicos de una cámara) del sistema de formación de imágenes, llamado también como distancia focal y centros ópticos respectivamente. Ahora bien, a menudo es más útil tener la representación de los puntos 3D del espacio en el referencial del mundo asociado al espacio de trabajo del robot que en el referencial de la cámara, siendo estos los parámetros extrínsecos que corresponden a vectores de rotación y traslación que traducen las coordenadas de un punto 3D a un sistema de coordenadas.

Debido a la distorsión radial, las líneas rectas aparecerán curvadas. Su efecto es mayor a medida que se aleja del centro de la imagen. Todas las líneas rectas esperadas están abultadas y viéndose más en las cámaras de menos gamma del mercado. Para las distintas aplicaciones, estas distorsiones necesitan ser corregidas primero. Para encontrar todos estos parámetros, lo que se debe hacer es proporcionar algunas imágenes de muestra de un patrón bien definido (por ejemplo, un tablero de ajedrez), así para encontrar algunos puntos específicos en él

(esquinas cuadradas del tablero) [56]. Para la calibración de la cámara, se puede mostrar desde la Fig. 3.15 a la 3.20, el tablero de ajedrez de 7x10 cuadrados de 2 cm. por lado, utilizado para la calibración de la cámara en distintas posiciones como se recomienda [56].

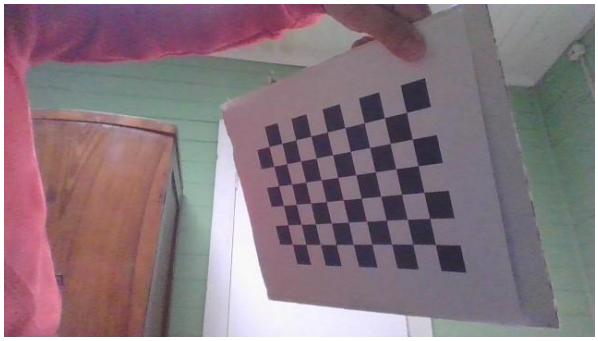

Fig. 3.15 Imagen del tablero de ajedrez y una de las imágenes utilizadas para la calibración de la cámara.

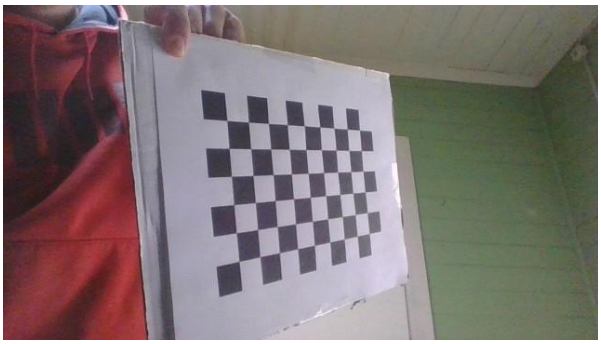

Fig. 3.16 Imagen del tablero de ajedrez y una de las imágenes utilizadas para la calibración de la cámara.

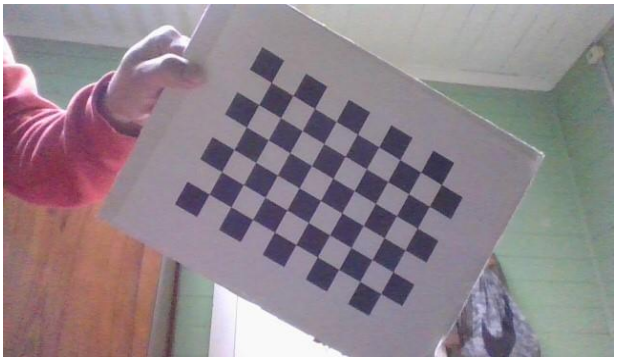

Fig. 3.17 Imagen del tablero de ajedrez y una de las imágenes utilizadas para la calibración de la cámara.

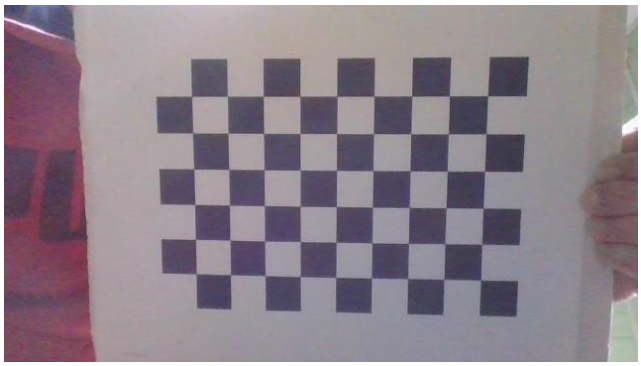

Fig. 3.18 Imagen del tablero de ajedrez y una de las imágenes utilizadas para la calibración de la cámara.

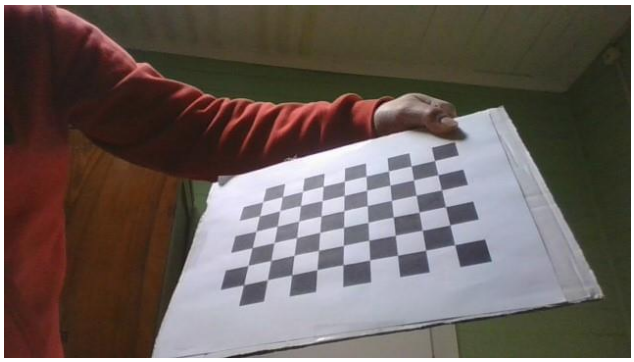

Fig. 3.19 Imagen del tablero de ajedrez y una de las imágenes utilizadas para la calibración de la cámara.

Fig. 3.20 Imagen del tablero de ajedrez y una de las imágenes utilizadas para la calibración de la cámara.

Para realizar la calibración se utilizan 14 imágenes tomadas por la misma cámara, ubicando al tablero de ajedrez en distintas posiciones, esto para obtener los valores donde se encuentras los puntos o esquinas del tablero de ajedrez y comenzar con la calibración. Así

que, teniendo los puntos de objeto y puntos de imagen listos para ser calibrados, se utiliza la función cv2.calibrateCamera () que devuelve la matriz de la cámara como muestra la ecuación 3.3.

$$mtx=\begin{bmatrix} 463.39235359 & 0 & 317.26712898 \\ 0 & 466.40985366 & 160.7222758 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.3)$$

Podemos refinar la matriz de la cámara basada en un parámetro de escalado libre usando cv2.getOptimalNewCameraMatrix ().

$$Newmtx=\begin{bmatrix} 605.92956543 & 0 & 324.29180584 \\ 0 & 466.40985366 & 159.15975046 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.4)$$

Los valores de la columna (3) fila (1) y (2) de la ecuación 3.4, corresponderían a los valores del centro óptico de la cámara (C_x y C_y respectivamente), los cuales serán los valores que se utilizan para poder realizar la comparación con los centros de la persona u objeto en estudio. Luego, se obtiene una nueva imagen (Fig. 3.21) con los parámetros corregidos, menos nítida que la original, pero con una disminución en el error de distorsión de la cámara. El error de re-proyección da una buena estimación de cuán exactos son los parámetros encontrados. Esto debería estar lo más cerca posible de cero. Dadas las matrices intrínsecas, de distorsión, rotación y traslación, primero transformamos el punto objeto punto a punto de imagen (los puntos 3D se denominan puntos de objeto y los puntos 2D se denominan puntos de imagen.) usando cv2.projectPoints (). Luego se calcula la norma absoluta entre lo que obtuvo con la transformación y el algoritmo de búsqueda de esquinas.

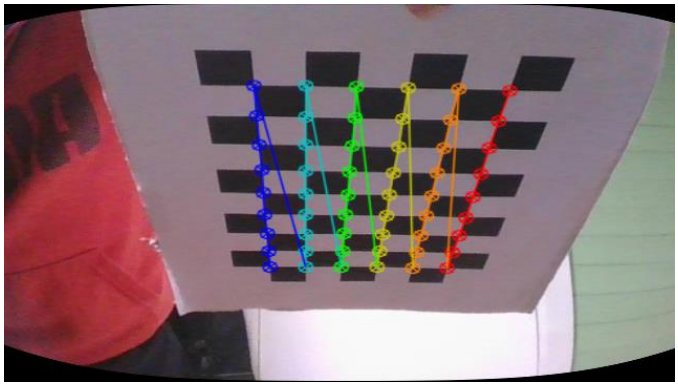

Fig. 3.21 Imagen del tablero de ajedrez ya calibrado.

Para encontrar el error medio se calcula la media aritmética de los errores calculados para todas las imágenes de calibración. El error debe ser un valor entre 0 a 1. Si es más cercano al valor 0 tendrá una calibración más precisa [56]. Teniendo esto en consideración, al calcular el valor del error, nos entrega un valor de 0.16105891982404366. Lo que permite considerar una buena precisión en la calibración realizada para obtener el centro óptico de la cámara.

3.10 Algoritmo de trabajo final

Para generar el procesamiento de imagen adecuado se trabajó con distintas librerías dispuestas para Python, como lo son cv2, Numpy, serial.

Cv2 es la librería que OpenCV es una librería de inteligencia artificial, nos permite manipular, leer y/o mostrar las imágenes en general. Numpy es una librería para realizar cálculo numérico en Python. La usaremos principalmente porque nos permite crear y modificar matrices, y hacer operaciones sobre ellas con facilidad [57]. Y la librería serial permite comunicarse con los otros dispositivos que utilicemos en nuestro proyecto, en este caso, *visual studio code* permite arrancar y desarrollar el procesamiento de imagen en lenguaje Python, pero el control del Drone se desarrolla en otro programa, lo que con la librería serial permite transmitir los datos necesarios para su control.

Algunos de los códigos expuestos en el algoritmo trabajado, mostrado en el diagrama de flujo (Fig. 3.1) se muestran en la tabla 3.1.

Tabla 2.1 Códigos más importantes utilizados en el algoritmo de procesamiento de imagen.

Códigos	Función
Serial.serial	Inicia la comunicación con el Arduino
cv2.VideoCapture	Inicia la comunicación entre la cámara utilizada y Python
cap.read	Lee lo que la cámara señala
cv2.cvtColor	Convierte al espacio de colores al que se desea utilizar
cv2.inRange	Detecta el color y transforma la imagen en binaria
cv2.medianBlur	Filtro para eliminar el ruido que entrega la cámara
cv2.findContours	Encuentra los puntos del contorno del objeto trabajado
cv2.contourArea	Calcula el área que se encerró con el código anterior
cv2.drawContours	Dibuja los contornos señalados

cv2.moments	Calcula el momento
cv2.imshow	Muestra la imagen en la pantalla
Arduino.write	Envía los datos obtenidos al Arduino

Al tener la imagen binaria (se refiere donde detecta el color es un valor uno o blanco y donde no está, es 0 o negro) tiene mucho ruido por efecto de la iluminación en el escenario de prueba y por la baja calidad de las cámaras, por ende, para eliminar la gran parte de ese ruido que afecta el procesamiento se aplica un filtro. En la figura 3.22 muestra una toma del inicio del algoritmo de trabajo, previo a pasar por la detección de los colores. Para este proyecto se decide trabajar con el rojo y, por lo que se vio con anterioridad en la Fig. 3.4 del rango de colores en HSV, el color rojo está en 2 sectores de la tabla por lo que la detección de rango se realiza en 2 partes para luego juntarlo en una imagen como se ve en las Fig. 3.22, 3.23, 3.24 y 3.25.

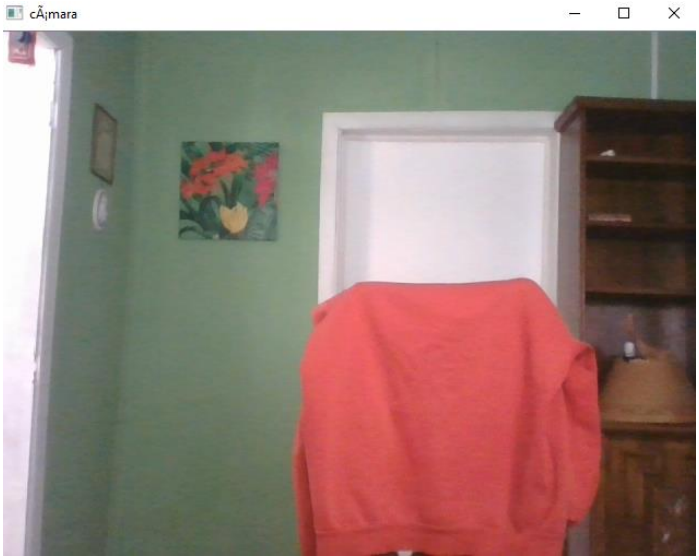

Fig. 3.22 Imagen captada del algoritmo antes de ser procesada.

En la Fig. 3.22 se muestra inicialmente lo que la cámara capta antes de iniciar con la detección del color rojo. Consiguiente, se realiza con la detección binaria del color, como bien se ha dicho, se tienen dos rangos de color rojo en HSV, lo cual se seccionan en maskred1 (Fig. 3.23) y maskred2 (Fig. 3.24).

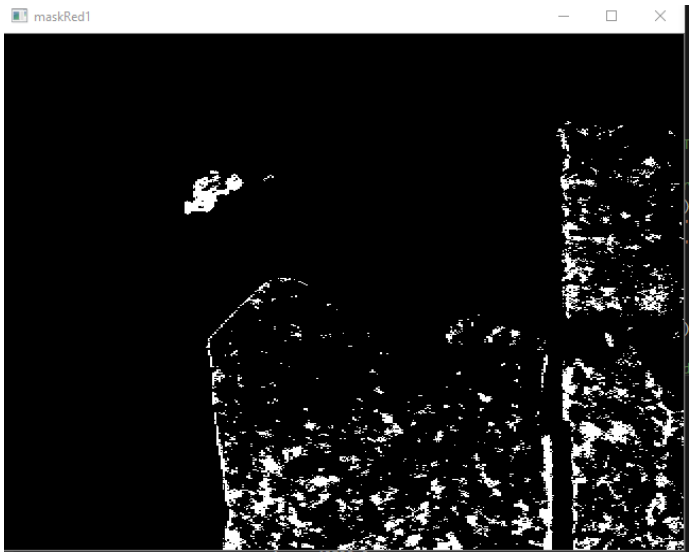

Fig. 3.23 Imagen binaria detección del Rango 1 del rojo en HSV.

Fig. 3.24 Imagen binaria detección del Rango 2 del rojo en HSV.

Lo que luego, se montan ambas imágenes binarias para trabajar con una sola (Fig. 3.25). Como se aprecia, la imagen tiene muchos puntos blancos por todo el cuadro, esto es porque se detecta rangos del color rojo en esos sectores, lo cual llegan a ser un problema al momento de calcular los valores necesitados. De igual manera, en los rangos detectados del color, se encuentran puntos negros, que por efecto de la luminosidad varía el color, quedando fuera del rango de detección. Por lo que se necesita un filtro para poder disminuir la perturbación hallada.

Fig. 3.25 Imagen que detecta el Rango 2 del rojo en HSV sin filtro.

Finalmente, al tener acoplados en una imagen ambos rangos del color a detectar, se procede a filtrar para eliminar en su mayoría el ruido como se observa en la Fig. 3.26.

Fig. 3.26 Imagen que detecta el Rango 2 del rojo en HSV después del filtro.

Se aprecia que el ruido no se va del todo de la imagen de la Fig. 3.26, por lo que más adelante, en donde se genera el contorno de la figura, se realiza una comparación de áreas para solo dibujar el contorno de un área mayor y así ese ruido no afecte del todo como se muestra en el diagrama de flujo de la Fig. 3.1. Ya teniendo esta comparación, se procede a calcular el centro de la figura contorneada, Para ello usaremos los momentos de cada

contorno [58] de la imagen [59]. Los momentos son una medida particular que indica la dispersión de una nube de puntos, y matemáticamente se definen como se muestra en la ecuación 3.5.

$$M_{ij} = \sum_x \sum_y x^i y^j I(x,y) \tag{3.5}$$

Donde x, y son las coordenadas de un píxel y la función $I(x,y)$ indica su intensidad. Estamos trabajando sobre una máscara binaria, por tanto, la función $I(x,y)$ sólo puede valer 0 o 1. Hay tres momentos que nos interesan: M00, M01 y M10. Fijémonos que, si se calcula M00, la ecuación anterior queda como la ecuación 3.6.

$$M_{00} = \sum_x \sum_y x^0 y^0 I(x,y) = \sum_x \sum_y I(x,y) \tag{3.6}$$

Para trabajar con esta ecuación de momento, se recuerda que $a^0 = 1$, acá se considera que $0^0 = 1$. Como $I(x,y)$ sólo puede dar 0 o 1, la fórmula de M_{00} es equivalente al número de píxeles cuyo valor es 1. Por lo tanto, M_{00} es el área en píxeles de la región blanca. Para M_{10}, la fórmula original se transforma en la ecuación 3.7.

$$M_{10} = \sum_x \sum_y x^1 y^0 I(x,y) = \sum_x \sum_y x\, I(x,y) \tag{3.7}$$

Esto es igual a la suma de las coordenadas x de los píxeles cuya intensidad sea 1. Por tanto, dividiendo M_{10} por M_{00} obtendríamos el centroide para la componente x, como muestra la ecuación 3.8.

$$\frac{M_{10}}{M_{00}} = \frac{\sum_x \sum_y x\, I(x,y)}{\sum_x \sum_y I(x,y)} = \frac{x_0 + x_1 + x_2 + \cdots + x_n}{n} = c_x \tag{3.8}$$

Lo mismo con M_{01} que queda representado por la ecuación 3.9.

$$\frac{M_{01}}{M_{00}} = \frac{\sum_x \sum_y y\, I(x,y)}{\sum_x \sum_y I(x,y)} = \frac{y_0 + y_1 + y_2 + \cdots + y_n}{n} = c_y \tag{3.9}$$

Con la ecuación 3.9 se tiene el centroide de la componente y.

Teniendo los valores calculados del centro de la figura captada y de su área, se procede a enviar estos datos, por medio del código *Arduino.write*, a la tarjeta del Arduino donde se encuentra el algoritmo de control.

Capítulo 4 Control

4.1 Introducción

En este capítulo, se detalla el algoritmo de control que permite ejecutar los movimientos del Drone, según los valores calculados y enviados por el algoritmo de procesamiento de imágenes anteriormente visto. Inicialmente, explicamos cuales son los objetivos del algoritmo y cuáles son las variables que se utilizarán para su desarrollo. Para luego explicar los procesos que conlleva para lograr mantener siempre en visión a la persona u objeto que se esté estudiando a una distancia constante.

4.2 Transmisión de los datos calculados a placa Arduino

Como se menciona en el capítulo 3, luego de calcular los datos en el algoritmo de procesamiento de imágenes, se transmiten los valores del centro de la figura contorneada y su área hacia al Arduino por medio de la comunicación serial (Fig.4.1). Los valores calculados se transmiten por medio de la función *write()*. La función utilizada *write* realiza la transmisión en bytes [59], por lo que es necesario convertir los datos que están en valores de enteros a código ASCII por el código *encode('ascii')*.

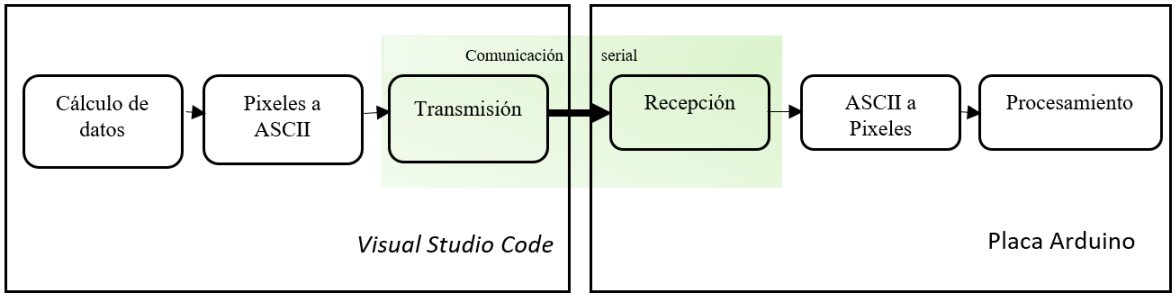

Fig. 4.1 Representación gráfica de transmisión y recepción de datos.

En Arduino, por efecto de sencillez y mejor entendimiento del trabajo, se convierte nuevamente los valores que se adquieren con el código *nombredeldato.toInt()*, de esta manera los datos están en la unidad de enteros para trabajarlos en el algoritmo de control.

4.3 Desarrollo del algoritmo de control

El objetivo de este algoritmo es mantener siempre en visual a la persona u objeto y mantener una distancia constante con él, es decir, si existe movimiento el Drone lo seguirá sin perderlo de la visión de la cámara, rotando, acercándose o alejándose según sea requerido. En primer lugar se necesitan 2 parámetros que se obtienen del procesamiento realizado en

VS Code ($c_x, Area$), donde c_x será comparado con valores cercanos al valor horizontal del centro focal (C_x) que se obtiene de la calibración de la cámara realizada (esto para poder tener un rango de estabilidad del Drone sin movimiento), con estos datos se genera la señal para mantenerlo siempre cercano al centro focal de la cámara como muestra las Fig. 4.2, 4.3 y 4.4 que hacen una referencia gráfica a los movimientos rotatorios para realizar esta acción, donde el cuadro que enmarca a la persona es el sensor de imagen del proyecto. Por otra parte, para estimar la distancia que se tiene entre el Drone y la persona u objeto que se esté analizando, se utiliza el área calculada ($Area$), el cual se compara con un área inicial ($Areai$) que es ingresada antes que comenzar con el vuelo (con esta área inicial se tiene la estimación aproximada de la distancia que se mantendrá), y según sea el resultado de la comparación, el Drone se acercará o alejará como se ve en las Fig. 4.5, 4.6 y 4.7.

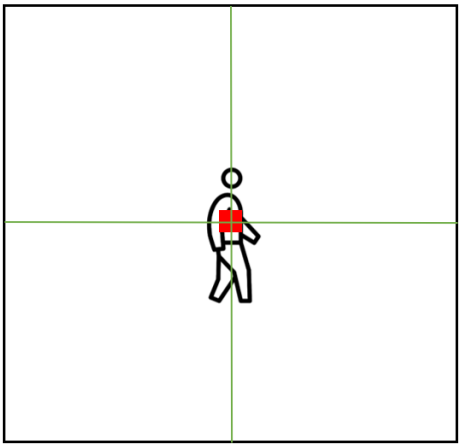

Fig. 4.2 Referencia de cámara en dirección a la persona.

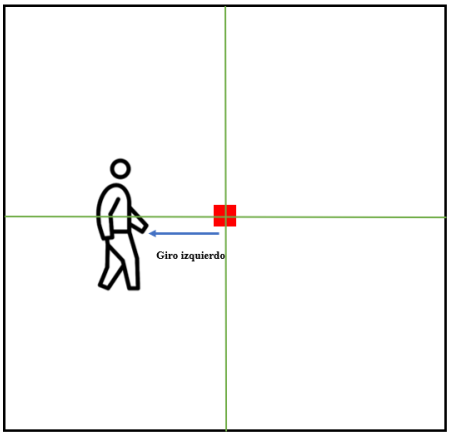

Fig. 4.3 Referencia de movimiento rotatorio izquierdo.

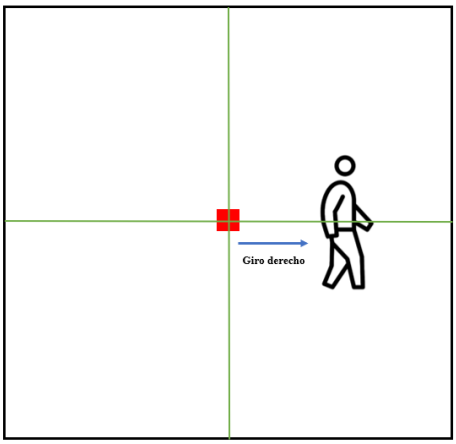

Fig. 4.4 Referencia de movimiento rotatorio derecho.

Fig. 4.5 Representación de distancia inicial para obtener área.

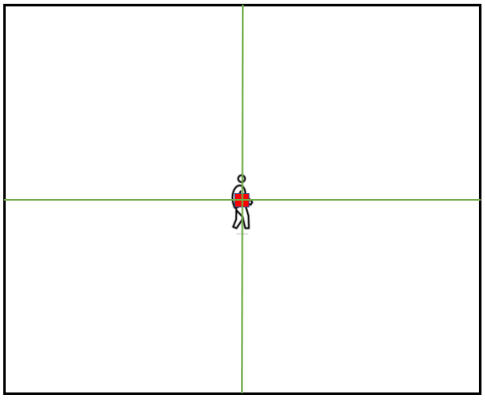

Fig. 4.6 Referencia del acercamiento del Drone.

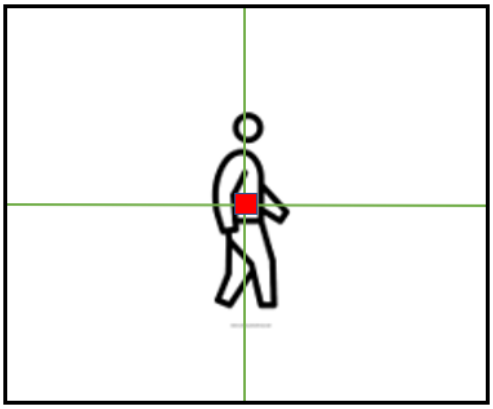

Fig. 4.7 Referencia de distanciamiento del Drone.

Los beneficios del dron Syma X8HG con respecto al vuelo, que viene incorporado con *Hold function*, o función de retención. La función de retención de altitud permite a un Drone mantener automáticamente una altitud. Esto significa que, salvo la entrada del piloto, el Drone volará a la misma altitud, independientemente de los factores externos. Esto libera al piloto para que se centre en asuntos más importantes, como dirigir la dirección en la que el Drone volará o enmarcar fotos o vídeos [60]. Aunque, los movimientos horizontales si se ve afectada por factores externos, por ejemplo, el viento del entorno en el que está volando.

Para los sistemas de paneles fotovoltaicos, existe un algoritmo para obtener el punto de máxima potencia (MPP), denominado Perturbar y Observar (P&O) se trata sobre observar o medir el voltaje y corriente del panel solar en el momento para luego calcular la potencia con el producto de la corriente con el voltaje y luego, comparar los resultados de la mediciones anteriores de potencia y voltaje, y según estas comparaciones perturbar al sistema para aumentar o disminuir el voltaje de referencia de modo de cada vez acercarse más al punto máximo [61]. Del mismo modo se observa y se mide los valores del centro de la figura captada y su área, y con ello, se compara con los valores iniciales cercanos del centro focal y del área muestreada, ingresados antes de iniciar por completo la comunicación serial. Según la posición en que se encuentre la figura captada, se perturba la posición de Drone reajustando de acuerdo la comparación anterior vista, como se ve en el diagrama de flujo de la Fig. 4.8.

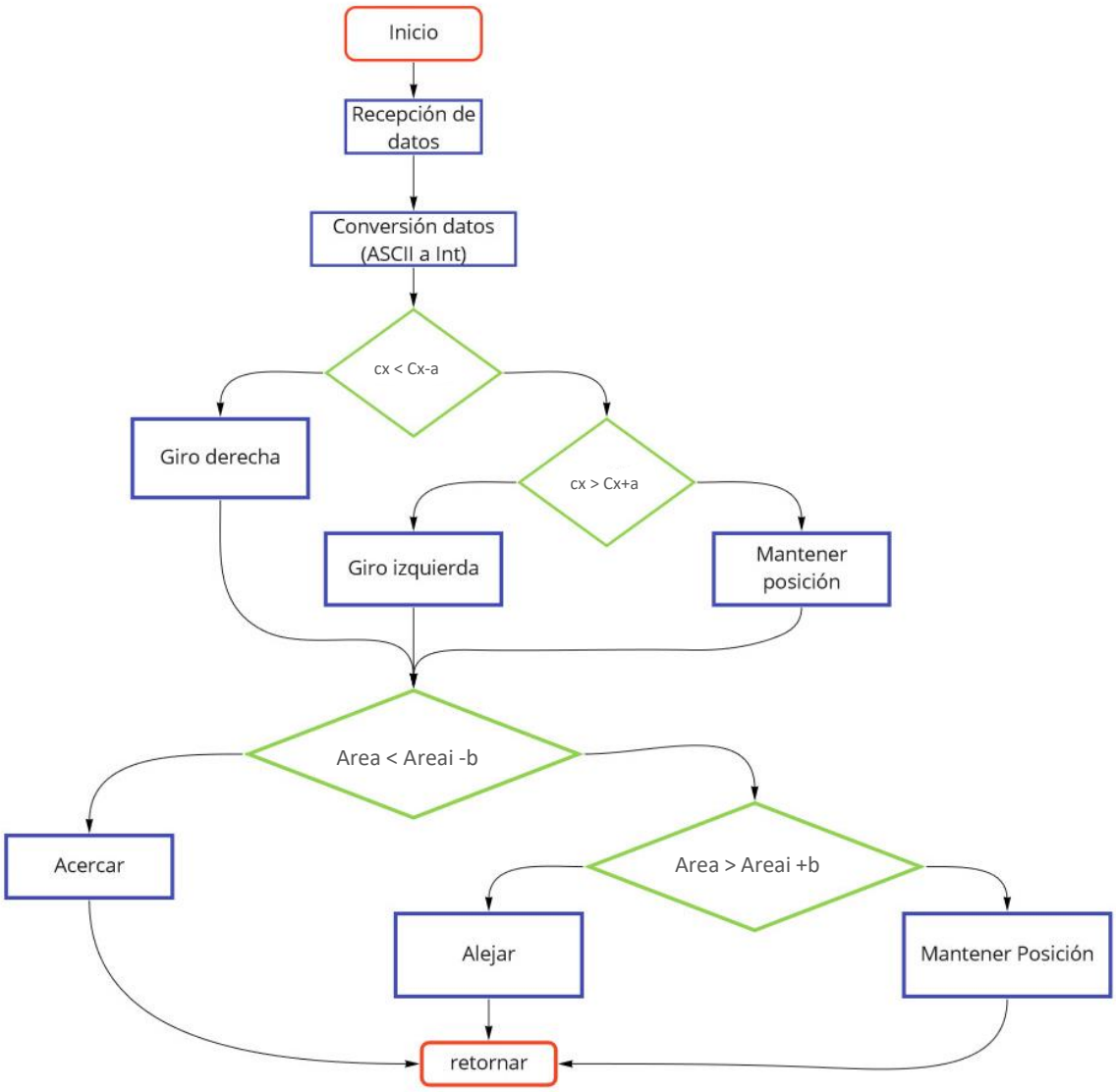

Fig. 4.8 Diagrama de flujo de control.

El diagrama de flujo (Fig. 4.8) inicia adquiriendo los datos desde Python. Luego convertimos de ASCII a enteros para un mejor entendimiento del trabajo que se tiene. En el proceso de perturbar y observar, se subdivide el proceso de mantener el objeto en el centro de la cámara con el de distancia constante para más simplicidad del control, el cual no afecta en el proceso ya que hablamos de velocidades de milisegundos de trabajo entre Python y de Arduino, además que son en distintas coordenadas del plano en las que están trabajando. En el primer proceso comparativo, se observa que el dato c_x se compara con el Cx aumentado y disminuido "a" (valor constante por afinidad del usuario), estos valores son el que contiene el valor horizontal del centro óptico de la cámara (324 calculado en la calibración de la

cámara), esto para tener un rango de tiempo muerto en que no se mueve el Drone y mantenerlo estable hasta que centro del contorno procesado salga de los parámetros establecidos. Si este es menor a ese rango el Drone girará a su derecha, en cambio sí es mayor, este girará a su izquierda (Fig. 4.9).

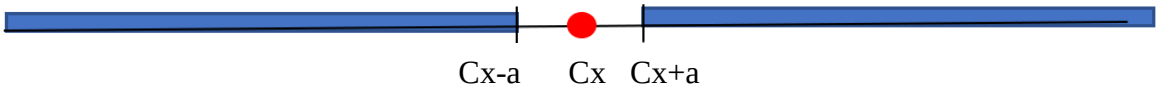

Fig. 4.9 Rango de trabajo del algoritmo de control para mantener en el centro al Drone.

En el segundo proceso comparativo se trabaja con el valor del $Area$ adquirido del algoritmo de procesamiento de imagen, con esto se realiza la estimación de la distancia que mantendría constante. Del mismo modo que la comparación para mantener en el centro focal de la cámara a la persona u objeto en estudio, la distancia que debe mantener el Drone también se trabaja con un rango "más menos b" para mantenerlo estable en el rango cercano al $Area\ inical$. Los Rangos de color azul, tanto para este proceso como el de la Fig. 4.9 y 4.10 son donde se realizan los movimientos para llevarlo a los puntos cercanos a los iniciales (punto rojo).

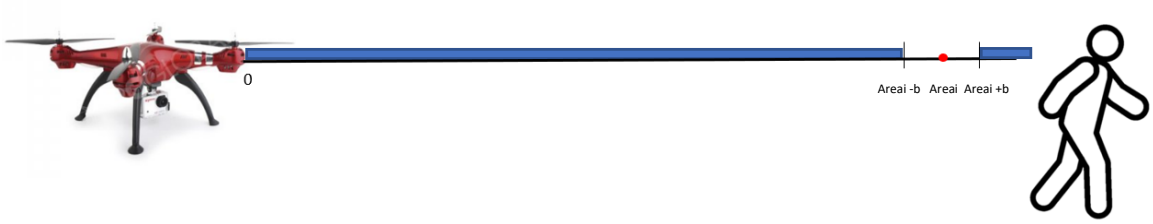

Fig. 4.10 Rango de trabajo del algoritmo de control para mantener la distancia del Drone.

Para el siguiente capítulo, se muestra los resultados experimentales que se realizaron evaluando en conjunto los algoritmos para el desarrollo del proyecto final.

Capítulo 5 Resultados Experimentales

5.1 Introducción

Este capítulo, explica la puesta en marcha del proyecto realizado en campo abierto, el cual se verifica el funcionamiento en conjunto del algoritmo de procesamiento de imagen y el algoritmo de control vistos con anterioridad. Esto, genera los movimientos necesarios del Drone para cumplir con los objetivos puesto en este proyecto (mantener a la persona u objeto evaluado, en un rango cercano al centro focal de la cámara y una distancia constante).

Para las siguientes secciones se mostrarán 2 vídeos en secuencias de imágenes, extraídas de los videos como prueba de los resultados realizados en diferentes días y horarios. El cual en el primer testeo no afecta directamente el viento como perturbación, a diferencia de la segunda secuencia de imágenes en cual, si tiene efecto no permitiendo tener en una posición al Drone. Cabe destacar que se tuvo que realizar una modificación respecto al proyecto inicialmente planteado, interviniendo de manera distinta el proyecto, pero manteniendo los objetivos iniciales.

5.2 Resultados experimentales

Como se menciona en el capítulo 3, el color seleccionado a detectar fue rojo. Al detectar este color, y por medio del proceso de tracking encierra el objeto que posee el color, realizando este seguimiento a medida que se va moviendo. Con ello, calcula el centro de la figura encerrada y el área, que luego lleva estos datos al Arduino gracias a la comunicación serial, para poder modificar la posición del Drone por medio de las señales de voltaje que se le envían al mando modificado su posición.

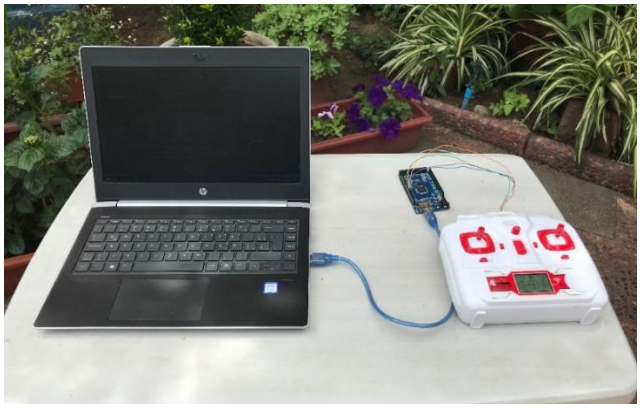

Fig. 5.1 Puesto de trabajo para adquisición de la imagen, procesarlo y control del Drone.

5.2.1 Ajustes del proyecto en pruebas experimentales

La modificación del proyecto para las pruebas experimentales se generó por un no envío a tiempo de los componentes, por parte de los proveedores, que lograban transmitir la imagen capturada de la cámara en el Drone hacia el computador. Por lo que, se realiza una simulación con la webcam del computador semejando la cámara montada en el Drone, lo que lleva a que la persona de prueba se ubique frente al computador, para interpretar su posición y de esto, se envían las señales correspondientes al Drone para modificar su posición según la comparación realizada entre los datos calculados de la persona en prueba y los valores inicialmente puestos.

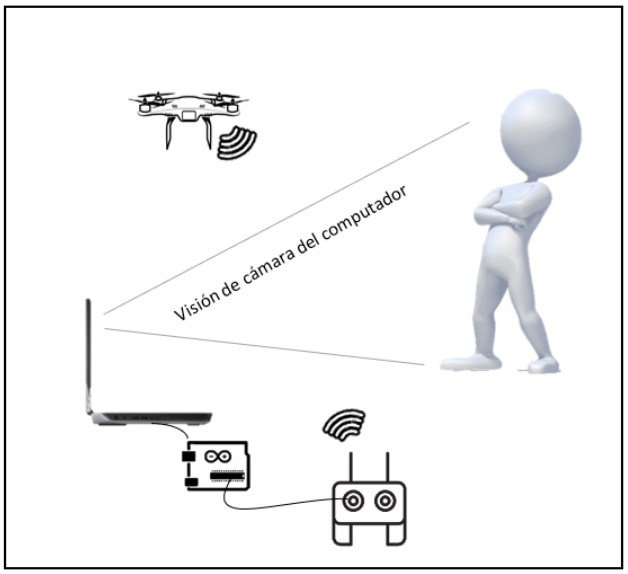

Fig. 5.2 Referencia gráfica de cómo sería el montaje para pruebas experimentales.

5.2.2 Secuencia de imágenes realizada del día uno de pruebas

Gráficamente, los movimientos registrados en la primera prueba experimental se muestran en la secuencia de la Fig. 5.3, donde el Drone en un inicio al iniciar el vuelo realiza un giro en sentido horario hasta luego estabilizarse, prontamente este vuelve a girar, para que finalmente realiza un movimiento de traslación. Todo en búsqueda de mantener a la persona en inspección en el rango generado para el centro focal y a una distancia constante.

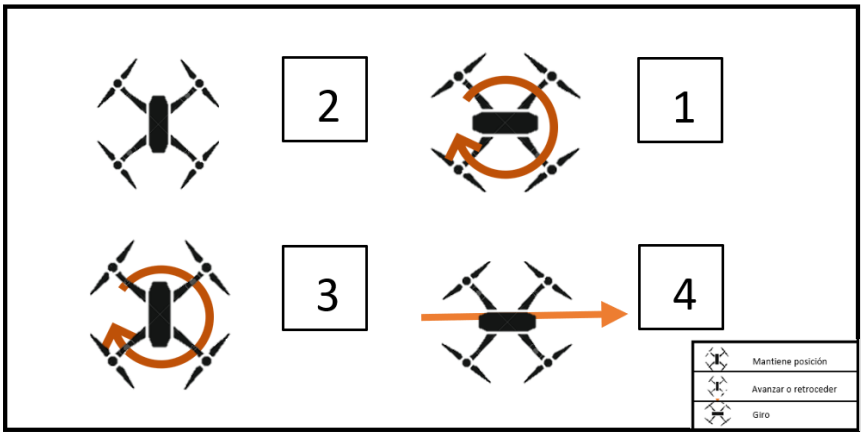

Fig. 5.3 Referencia de los movimientos secuenciales realizados en la prueba experimental 1.

En el primer dia de testeo realizado, se toma la secuencia de imágenes de uno de los videos obtenidos, el cual el viento era de baja intensidad y en un horario cercano al atardecer, permitiendo estabilizar al Drone en una posición y realizar el procesamiento de imagen adecuadamente. Se inicia (Fig. 5.4.1) con el Drone en el suelo esperando recibir la señal para poder elevarse y realizar el seguimiento continuo a la persona de testeo que se ubica frente la cámara del computador.

Fig. 5.4 Secuencia de elevación del Drone.

Una vez recibida la señal, este se comienza a elevar (Fig. 5.4.2 y Fig. 5.4.3) hasta que se acabe el tiempo dado de elevación. Luego comienza el proceso de seguimiento que tiene el algoritmo inscrito. Para los rango de trabajo de comparación inicial, tanto del área como del centro de la figura captadas están dentro lo establecido en esta prueba, pór lo que permite mantener al Drone de manera estable en su vuelo como se ve en la Fig. 5.6.1, 5.6.2 y 5.6.3.

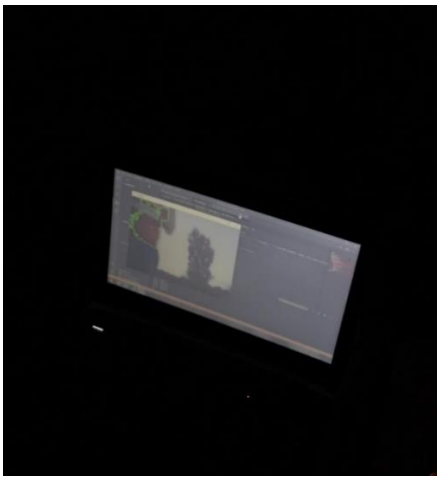

Fig. 5.5 Posicionamiento incial en rango cercano al origen de la camara y área en rango del área inicial.

Los rangos de comparación para mantener en visión a la persona en estudio, como primera instancia se deja en un valor cercano al origen del sistema de coordenadas de la cámara, como se observa en la Fig. 5.5, esto es porque al momento de arrancar el programa no se realiza la modificación inicial del valor de los rangos de c_x , por lo que se tiene que ubicar en esta zona antes mencionada una vez elevado el Drone e iniciada la grabación.

Fig. 5.6 Secuencia de estabilización del Drone en la altura.

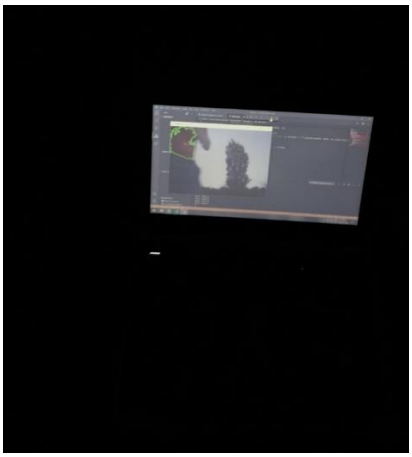

Fig. 5.7 Modificación de posición de la persona frente la cámara.

Fig. 5.8 Movimiento rotacional del Drone.

Luego se modifica la posición de la persona en prueba para luego volver al rango establecido del tiempo muerto (esto para generar una rotación en el Drone), además de disminuir el área de trabajo del algoritmo del procesamiento de imagen (Fig. 5.7) lo que genera que el Drone rote hasta cierto punto y avance para encontrar el valor del rango de área inicial, siendo este el final del video de esta prueba como se ve en las Fig. 5.8, y la secuencia de la Fig. 5.9.1, 5.9.2 y 5.9.3.

Fig. 5.9 Avance del Drone hasta poder estar en el rango del área inicial.

5.2.3 Secuencia de imágenes realizada del día dos de prueba

En la siguiente sección, se explica la secuencia de imágenes extraídas de uno de los videos realizados para evidenciar el correcto funcionamiento del proyecto de la segunda jornada de pruebas. Estas se realizan en campo abierto, alrededor de las 17:00 horas. Donde se tiene como perturbación el viento del ambiente y la iluminación del entorno. Con la Fig. 5.10 se muestra gráficamente los movimientos del Drone que realiza en el video de la segunda experiencia.

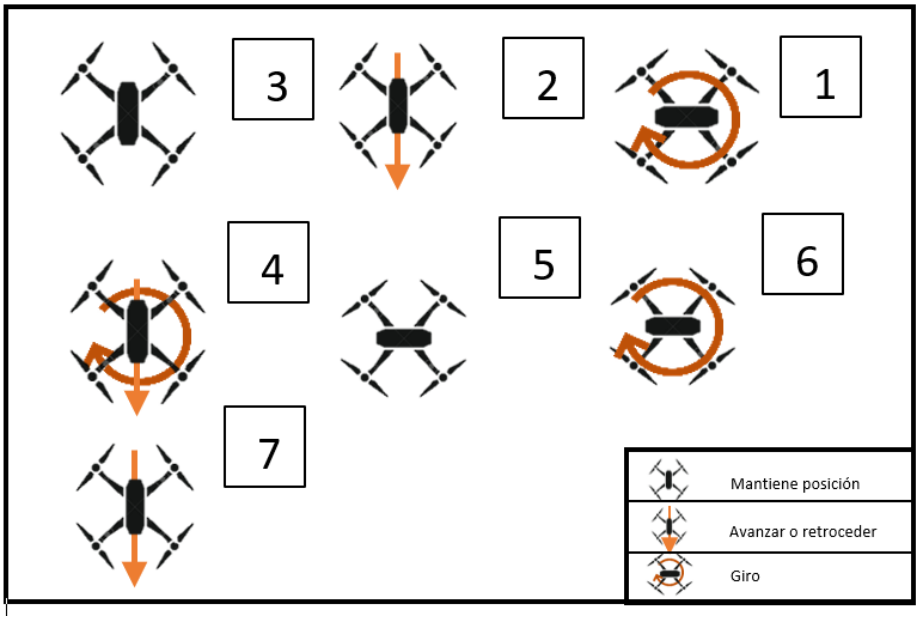

Fig. 5.10 Referencia de los movimientos secuenciales realizados en la prueba experimental 2.

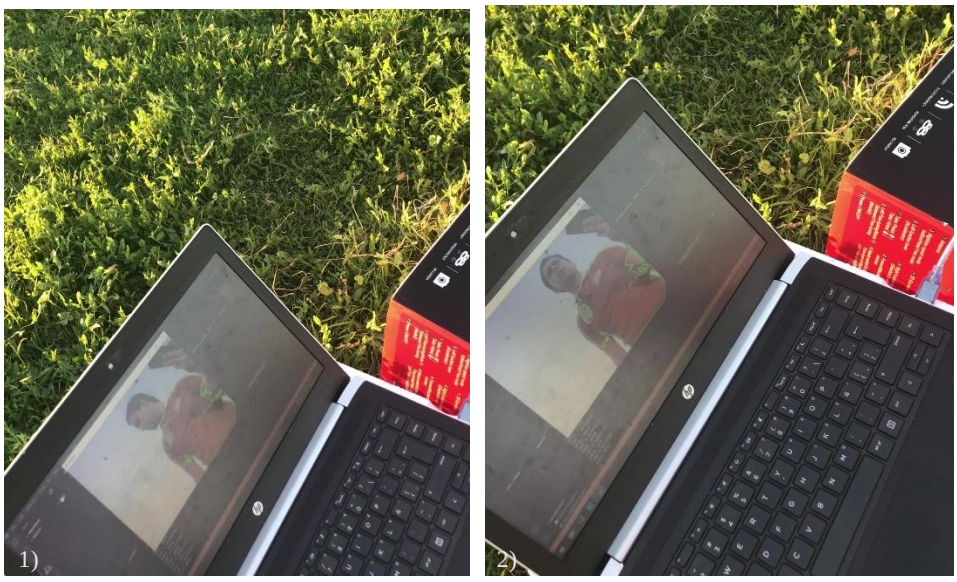

Fig. 5.11 Inicio de video y a la secuencia de prueba.

Esta prueba comienza (Fig. 5.11.1) explicando lo que debiese ocurrir al momento de iniciar el vuelo del Drone (Fig. 5.11.2). Como el área encerrada de color rojo (que es el color que se trabaja) es menor al rango del área inicial y la posición del sujeto está a un lado izquierdo del computador, la orden que se le da al Drone es que gire en sentido horario (giro derecho) como muestra la Fig. 5.12.1 y Fig. 5.12.2, y avance hasta estar a la distancia que es proporcional al área establecida en un inicio (Fig. 5.13.1 y Fig. 5.13.2).

Fig. 5.12 Inicio del vuelo y rotación del Drone.

Fig. 5.13 Secuencia de avance del Drone.

Luego, el Drone comienza intenta mantener la posición (Fig. 5.14) pero rotando en el mismo sentido en el que estaba (Fig. 5.15.1 y Fig. 5.15.2), ya que se había aumentado el área hasta llegar hasta el rango de estabilización mencionado en el capítulo 4, y la rotación se debía ya que la persona en estudio se ubicaba en el lado izquierdo del computador inicialmente visto.

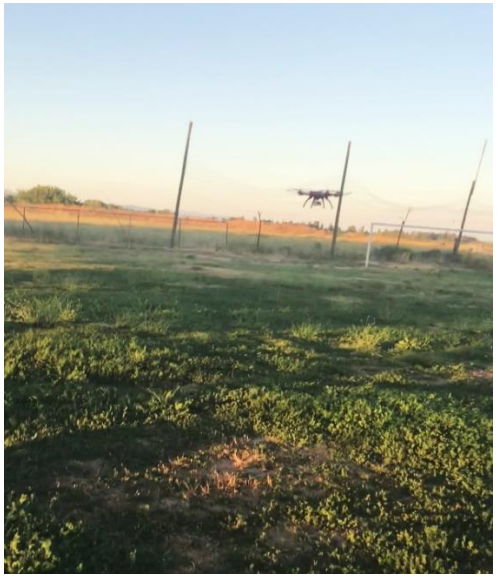

Fig. 5.14 Rotación y avance del Drone por mantener ubicación.

Fig. 5.15 Estabilización de Drone por obtener misma área inicial.

El viento es un problema al momento de mantener en la misma posición al Drone, aunque no afecta en su altura, si en las otras coordenadas en las que trabaja el robot aéreo, mostrándose en los videos de prueba del proyecto (véase en Anexos, pag 74)

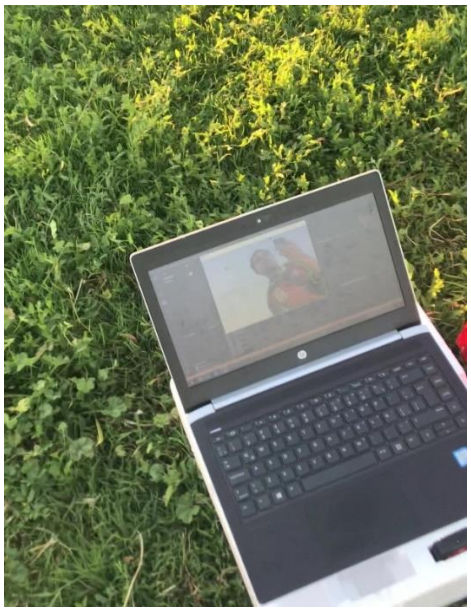

Fig. 5.16 Verificación de la posición de figura capturada (izquierdo del computador).

Por un movimiento del brazo con el que se filma el video, genera que se disminuya un poco el área (Fig. 5.16) lo que con el movimiento rotacional que tenía, genera un pequeño vaivén en su posición, lo que al momento de sacar el brazo se vuelve al área para poder

mantener al Drone posicionado con esa rotación que tenía como se muestra en las Fig 5.17.1, 5.17.2 y 5.17.3.

Fig. 5.17 Rotación del Drone por posición frente la cámara.

La prueba de este video finaliza disminuyendo el área de observación de la cámara web del computador (Fig. 5.18), lo que genera es que el Drone se mueva hasta que el algoritmo de control calcule estar en el rango de espacio muerto donde no genera ninguna señal de movimiento como se demuestra en las Fig. 5.19.1 y 5.19.2.

Fig. 5.18 Nueva posición mas alejada de la cámara.

Fig. 5.19 Cambio de posición por nueva área.

Capítulo 6 Conclusiones

6.1 Sumario

En el capítulo 1 infiere al estado del arte que rodea al proyecto en particular, con ello se especifican los objetivos, alcances y la metodología de trabajo necesitada en la integración de tecnologías en relación de la adquisición y análisis de datos para el seguimiento continuo de objetos o personas que se requiera.

El capítulo 2 hace referencia a los implementos (*hardware* y *softwares*) utilizados en todo el proyecto, es decir, los requeridos en la realización y ejecución del algoritmo de procesamiento de imágenes y del control del Drone, además de los componentes para la transmisión de datos y señales con el objetivo de cumplir lo propuesto en el proyecto.

Sobre el capítulo 3 se describe el desarrollo del algoritmo de procesamiento de imágenes realizado, cuyo objetivo es calcular y adquirir los valores del área y el centro de la figura captada por medio de la cámara utilizada que luego se transmiten hacia el algoritmo de control. Además, se describen los pasos utilizados para adquirir la imagen, definiendo la elección de los *softwares*, el tipo de lenguaje y también como se calcula los datos necesitados.

Consiguientemente el capítulo 4 habla sobre el algoritmo de control generado a partir de los datos adquiridos del proceso anterior, el cual se explica el método en que se basa para mantener a la persona u objeto de estudio siempre cercano al centro focal de la cámara además de una distancia constante.

Para finalizar, el capítulo 5 se describe las pruebas experimentales realizadas en campo abierto, donde se muestra el trabajo de todos los procesos anteriormente descritos, desde la adquisición de la imagen, la transmisión de datos por medio de la comunicación serial y transmisión de las señales desde la placa Arduino que contiene el algoritmo de control hacia el mando del Drone permitiendo realizar los movimientos según se ha calculado.

6.2 Conclusiones

El desarrollo de este proyecto es una base firme para distintas áreas de trabajo dentro de la robótica, una de las áreas que se puede enfocar es el estudio termográfico de deportistas para prevenir una lesión que pueda perjudicar su futuro, o seguimiento de vehículos en un radio más amplio del que se puede actualmente con los dispositivos trabajados, entre otras aplicaciones en las que se pueda desarrollar.

El objetivo de este proyecto fue realizar el seguimiento a objetos mediante la detección de color y proceso tracking, integrando tecnología y estudios actuales sobre el procesamiento de imagen y control de robots que permitiría realizar el seguimiento, pero con un presupuesto menor al de algunos equipos en el mercado actual, el cual se logra de manera concreta desarrollar como se ve en este reporte. Se simula con ayuda de la cámara web del computador, que la imagen captada provenía de una cámara montada en el Drone, esto para corroborar que los datos calculados por el procesamiento de imagen eran correctos para poder realizar el control, sin necesidad de utilizar el mando manual usualmente visto. Se optó por esta solución ya que con los proveedores se tuvo un percance con los elementos pedidos, pero se logra implementar correctamente y alcanzar los objetivos puestos en un inicio del desarrollo del proyecto.

El lenguaje utilizado para procesar las imágenes captadas fue Python, con este, se logra calcular el centro de la figura junto con su área, para poder transmitir hacia el algoritmo de control realizado en Arduino. Se logra apreciar en los resultados experimentales, que la iluminación afecta directamente al momento de detectar el color deseado, necesitando filtros y caracterización para solo evaluar a la persona u objeto puesto en escena.

El algoritmo de control utilizado se basa en un perturbar y observar (P&O). Se utiliza este método ya que no es necesario trabajar con las ecuaciones dinámicas del robot aéreo para los movimientos de rotación y traslación del Drone, más bien solo se necesitan los datos calculados en el algoritmo de procesamiento de imágenes para modificar su posición según corresponda.

El sensor de imagen fue la cámara web del computador, se puede modificar utilizando una cámara de mejor gama (sobre 720p), ya que, con una de menor calidad la imagen captada genera mucho ruido provocando un mayor error en el procesamiento del proyecto. Esta cámara puede ir montada directamente en el Drone, así tendría un sistema completamente autónomo sin preocuparse de que la persona u objeto en estudio se salga de la visión de la cámara y que ellos puedan realizar las actividades con normalidad. Así, el Drone es quien se mueve para mantener en un mismo punto el centro de la figura contorneada con la del centro de la cámara.

El viento es un problema a la hora de mantener estable en una posición al Drone, se podría incorporar un anemómetro que envíe la señal de la velocidad del viento y con esto

modificar las señales de salida que se dirigen al Drone para poder trabajar adecuadamente con él.

El seguimiento de imagen y la detección del cuerpo está hecho para trabajar de forma unitaria, por tal que, si llegase alguien a entrar en la visión del sensor de imagen del mismo color y área, afecta directamente al trabajo con la persona u objeto que se evalúa.

Teniendo en cuenta lo analizado en este documento en conjunto a las pruebas realizadas, en el marco de los alcances y limitaciones puestos, se tiene evidencia que si es posible crear un sistema de seguimiento continuo a objetos por medio de un Drone utilizando los lenguajes de Python y Arduino para el procesamiento de imagen y control de vuelo.

6.3 Trabajo a futuro

- Implementar el procesamiento de imagen y control en un robot terrestre o acuático.
- Utilizar ROS como desarrollo de ambos algoritmos en vez de 2 *softwares* distintos
- Trabajar en lenguaje C++ o Matlab el procesamiento de imagen verificando si es más rápido o no.
- En algoritmo de procesamiento de imágenes se puede utilizar uno más robusto como SURF, SIFT, ASIFT en vez de P&O.
- Intervenir de otra forma el Drone para controlar sus movimientos (modificando su radio frecuencia, o realizar la construcción del Drone de cero).
- Trabajar con las ecuaciones dinámicas del sistema para tener un mejor control respecto perturbaciones.
- Implementar este proyecto en algún desarrollo termográfico que permita realizar análisis de temperatura en distintos objetos mientras se mueve.
- Realizar la detección múltiple para el seguimiento de distintas personas u objetos.

Referencias

[1] elEconomistaAmérica.com | Chile, «eleconomistaamérica,» 27 02 2020. [En línea]. Available: https://www.eleconomistaamerica.cl/economia-eAm-chile/noticias/10383013/02/20/Cinco-sectores-donde-la-robotica-esta-transformando-la-economia.html. [Último acceso: 12 12 2020].

[2] «Toptal,» Francesco Castellano, 2016. [En línea]. Available: https://www.toptal.com/finance/market-research-analysts/los-drones-comerciales-estan-revolucionando-las-operaciones-comerciales#:~:text=En%20un%20informe%20de%202016,d%C3%B3lares%20entre%202016%20y%202020.. [Último acceso: 18 09 2020].

[3] MBA egresado del IESA – Venezuela, «world energy trade,» 01 03 2019. [En línea]. Available: https://www.worldenergytrade.com/articulos-tecnicos/logistica-at/drones-tecnologia-de-alto-vuelo. [Último acceso: 12 12 2020].

[4] E. R. Guerrero, «uniandes,» 16 12 2015. [En línea]. Available: https://repositorio.uniandes.edu.co/bitstream/handle/1992/18783/u722265.pdf?sequence=1. [Último acceso: 2020 09 17].

[5] P. D. González, «CUADRICOPTERO PARA SEGUIMIENTODE OBJETOS,» Julio 2016. [En línea]. Available: https://repositorio.comillas.edu/xmlui/bitstream/handle/11531/14379/TFG000963.pdf?sequence=1&isAllowed=y. [Último acceso: 13 Junio 2020].

[6] F. Ciudad Fernández, «ebuah,» 2017. [En línea]. Available: https://ebuah.uah.es/dspace/handle/10017/32638. [Último acceso: 17 09 2020].

[7] E. d. J. C. Juan Manuel IBARRA ZANNATHA, «mecamex,» [En línea]. Available: http://www.mecamex.net/anterior/cong02/papers/art14.pdf. [Último acceso: 11 10 2020].

[8] J. &. T. R. &. M. J. &. M. F. &. G. P. omares, «researchgate,» 01 01 2002. [En línea]. Available: https://www.researchgate.net/publication/39437612_Seguimiento_de_trayectorias_3D_mediante_control_visual_basado_en_imagen. [Último acceso: 16 09 2020].

[9] E. C. Laso, «robolab.unex,» [En línea]. Available: https://robolab.unex.es/wp-content/plugins/papercite/pdf/tracking-automatico-secuencias.pdf. [Último acceso: 22 09 2020].

[10] S.-H. Y. BUSTAMANTE, «opac.pucv,» 03 2014. [En línea]. Available: http://opac.pucv.cl/pucv_txt/txt-4500/UCE4968_01.pdf. [Último acceso: 12 12 2020].

[11] D. R. W. I. J. M. M. I. P. Tristan, «Facultad de ciencias de la universidad nacional del centro de la provincia de Buenos Aires,» 2011. [En línea]. Available: http://www.exa.unicen.edu.ar/catedras/pdi/FILES/TE/CP1.pdf. [Último acceso: 22 09 2020].

[12] n. señala, «bibliotecadigital.ilce.edu,» [En línea]. Available: http://bibliotecadigital.ilce.edu.mx/sites/ciencia/volumen2/ciencia3/084/htm/sec_9.htm. [Último acceso: 12 12 2020].

[13] J. &. J. R. &. A. L. &. V. B. Chicala, «researchgate,» 05 01 2006. [En línea]. Available: https://www.researchgate.net/publication/28791943_Reconocimiento_Y_Seguimiento_De_Objetos_Moviles_En_Un_Sistema_De_Futbol_Robotico#pf22. [Último acceso: 16 09 2020].

[14] S. y. A. d. d. I. p. e. d. u. A. M. P. C. L. Z. M. Estudio y Selección de las TécnicasSIFT, «SADIO,» 2012. [En línea]. Available: http://41jaiio.sadio.org.ar/sites/default/files/6_EST_2012.pdf. [Último acceso: 22 12 2020].

[15] A. D. Mairale, «International Research Journal of Engineering and Technology,» 10 2016. [En línea]. Available:https://www.irjet.net/archives/V3/i10/IRJET-V3I1053.pdf. [Último acceso: 22 12 2020].

[16] C. B. M. M. M. A. G. Miguel Ángel Abellán, «programoergosum,» 2020. [En línea]. Available: https://www.programoergosum.com/cursos-online/raspberry-pi/244-iniciacion-a-python-en-raspberry-pi/que-es-python. [Último acceso: 22 11 2020].

[17] ecured, «Ecured,» 12 9 2010. [En línea]. Available: https://www.ecured.cu/index.php?title=C%2B%2B&action=history. [Último acceso: 21 12 2020].

[18] c. d. a. Matlab, «Matlab,» Matlab, [En línea]. Available: https://la.mathworks.com/help/matlab/learn_matlab/product-description.html. [Último acceso: 21 12 2020].

[19] S. Halide, «Halide,» Halide, 2012. [En línea]. Available: https://halide-lang.org/. [Último acceso: 21 12 2020].

[20] ROS, «ROS,» [En línea]. Available: https://www.ros.org/about-ros/. [Último acceso: 21 12 2020].

[21] M. G. B. E. E. S. Andrea Tapia Arenas, «Procesamiento de imágenes apartir de vehículos aéreos no tripulados utilizando software libre,» 2009. [En línea]. Available: https://drive.google.com/file/d/1e4DiRChJDmfFfb0F2sNaLsKeVWlcQMh4/view?fbclid=IwAR0EPtPpcB8ysyhNV-fTMCqxz5bXOW7RFgXwLkIHnAnJ-sYjg7DLaZdz0Ho. [Último acceso: 13 12 2020].

[22] E. E. S. Cruz, «Universidad Autónoma de Barcelona,» 2017. [En línea]. Available: https://www.tdx.cat/bitstream/handle/10803/456309/eesc1de1.pdf?sequence=1&isAllowed=y. [Último acceso: 13 12 2020].

[23] J. Rejón, «rentadrone,» 30 05 2018. [En línea]. Available: https://rentadrone.cl/multi-rotor-o-ala-fija-aprende-a-elegir/. [Último acceso: 13 12 2020].

[24] Tienda de venta de Drones, «drones baratos ya,» 2020. [En línea]. Available: https://www.dronesbaratosya.com/mjx-x600-un-hexacoptero-de-iniciacion/. [Último acceso: 13 12 2020].

[25] RDN, «the rdn,» 18 09 2012. [En línea]. Available: https://the-rdn.com/2012/09/83984/. [Último acceso: 13 12 2020].

[26] F. C. Fernández, «Diseño y construcción de un dron para el seguimiento de un objeto de forma autónoma mediante visión artificial.,» Universidad de Alcalá. Escuela Politécnica Superior, Madrid, 2017.

[27] recdron, « recdron,» 17 10 2017. [En línea]. Available: https://www.recdron.com/es/blog/el-uso-de-drones-en-la-practica-deportiva. [Último acceso: 18 09 2020].

[28] F. O. S. HUAYTA, «UNIVERSIDAD NACIONAL DE SAN AGUST´IN DE AREQUIPA,» 2019. [En línea]. Available: http://repositorio.unsa.edu.pe/bitstream/handle/UNSA/10621/CCsuhufo%20%281%29.pdf?sequence=3&isAllowed=y. [Último acceso: 18 09 2020].

[29] P. N. Juan Cortabitarte, «Semantic Scholar,» 2004. [En línea]. Available: https://www.semanticscholar.org/paper/Seguimiento-y-estimaci%C3%B3n-de-trayectorias-Cortabitarte-Novarini/aaa541b3f40110e9379f14c5c49159873af94d0e#related-papers. [Último acceso: 17 09 2020].

[30] R. G. J. M. L. F. T. P. G. J. Pomares, «Comité español de automatica (CEA),» 2005. [En línea]. Available: https://intranet.ceautomatica.es/old/actividades/jornadas/XXIII/documentos/ja02_061.pdf. [Último acceso: 17 09 2020].

[31] S. A. P. C.-W. R. C. Lote, «Universidad Militar Nueva Granada,» [En línea]. Available: https://repository.unimilitar.edu.co/bitstream/handle/10654/17513/ParraCabreraSusanaAndrea2018%20%282%29.pdf?sequence=2&isAllowed=y. [Último acceso: 18 09 2020].

[32] M. A. &. C. C. D. P. Albornoz Chicango, «Repositorio Digital Institucional de la Escuela Politécnica Nacional,» 2016. [En línea]. Available: https://bibdigital.epn.edu.ec/bitstream/15000/16978/1/CD-7555.pdf. [Último acceso: 18 09 2020].

[33] R. M. J. Cristian, «Universidad catolica de san pablo,» 09 2019. [En línea]. Available: http://repositorio.ucsp.edu.pe/bitstream/UCSP/16086/1/ROSAS_MENCHOLA_JOS_DRO.pdf. [Último acceso: 17 09 2020].

[34] R. Jiménez Bravo, «Depósito de investigación universidad de sevilla,» 2018. [En línea]. Available: https://idus.us.es/handle/11441/85466. [Último acceso: 18 09 2020].

[35] L. C. G. D. l. P. V. B. Monroy Anieva Jesús Arturo, «Asociación Mexicana de Mecatrónica A.C,» 2017. [En línea]. Available: www.researchgate.net/profile/Martin_Villa_Bracamontes/publication/323399150_Modelado_y_Control_de_Sistemas_Mecatronicos_ISBN_978-607-9394-10-3/links/5a941db7a6fdccecff063fee/Modelado-y-Control-de-Sistemas-Mecatronicos-ISBN-978-607-9394-10-3.pdf#page=69. [Último acceso: 18 09 2020].

[36] J. P. Puerta, «Dialnet,» 07 2017. [En línea]. Available: https://dialnet.unirioja.es/servlet/tesis?codigo=156073. [Último acceso: 18 09 2020].

[37] anónimo, «proglobal,» 2020. [En línea]. Available: https://proglobal.cl/productos/dron-syma-x8hg-camara-hd-sistema-estabilizacion-vuelo-2. [Último acceso: 06 11 2020].

[38] s. HP, «HP,» [En línea]. Available: https://support.hp.com/cl-es/document/c05695402. [Último acceso: 21 12 2020].

[39] Anónimo, «tutotial de drones,» no señala. [En línea]. Available: https://www.tutorialdedrones.com/transmisor-video-fpv/. [Último acceso: 06 11 2020].

[40] Ingeniería MCI Ltda, «arduino,» Ingeniería MCI Ltda, [En línea]. Available: https://arduino.cl/arduino-mega-2560/. [Último acceso: 29 11 2020].

[41] S. t. v. s. code, «Visual Studio Code,» Visual Studio, 11 06 2020. [En línea]. Available: https://code.visualstudio.com/docs/supporting/FAQ. [Último acceso: 22 11 2020].

[42] MCI electronics, «Arduino,» MCI electronics, [En línea]. Available: https://arduino.cl/que-es-arduino/. [Último acceso: 25 12 2020].

[43] Equipo de Laboratorio Didáctico, «Tutorial 5: Comunicación Serial,» Proserquisa, San Salvador , 2016.

[44] Anónimo, «blogspot,» 12 11 2013. [En línea]. Available: http://dzlsevilgeniuslair.blogspot.com/2013/11/more-toy-quadcopter-hacking.html. [Último acceso: 26 11 2020].

[45] L. Garcia, «Un poco de java,» 9 10 2013. [En línea]. Available: https://unpocodejava.com/2013/10/09/que-es-opencv/. [Último acceso: 22 11 2020].

[46] A. S. L. y. A. S. Jiménez, «python-para-impacientes,» blogspot, 2014. [En línea]. Available: https://python-para-

impacientes.blogspot.com/2019/10/primeros-pasos-con-numpy.html. [Último acceso: 13 12 2020].

[47] Wikipedia, «wikipedia,» 23 03 2020. [En línea]. Available: https://es.wikipedia.org/wiki/Reconocimiento_de_color. [Último acceso: 03 10 2020].

[48] G. Solano, «omes,» 14 09 2019. [En línea]. Available: https://omes-va.com/deteccion-de-colores/. [Último acceso: 03 10 2020].

[49] G. D. Giuseppe, «Medium,» 29 08 2019. [En línea]. Available: https://medium.com/@gastonace1/detecci%C3%B3n-de-objetos-por-colores-en-im%C3%A1genes-con-python-y-opencv-c8d9b6768ff. [Último acceso: 04 10 2020].

[50] Ecured, «Ecured,» 2012. [En línea]. Available: https://www.ecured.cu/Modelo_HSV. [Último acceso: 13 12 2020].

[51] blogero anónimo, «tecnonautas,» 2020. [En línea]. Available: https://tecnonautas.net/comprension-del-modelo-de-color-hsv/. [Último acceso: 13 12 2020].

[52] U. d. b. anónimo, «https://stackoverflow.com/questions/10948589/choosing-the-correct-upper-and-lower-hsv-boundaries-for-color-detection-withcv?lq=1,» 05 2012. [En línea]. Available: https://stackoverflow.com/questions/10948589/choosing-the-correct-upper-and-lower-hsv-boundaries-for-color-detection-withcv?lq=1. [Último acceso: 04 10 2020].

[53] K. M. y. J. M. Zdenek Kalal, «vision.stanford.,» Enero 2010. [En línea]. Available: http://vision.stanford.edu/teaching/cs231b_spring1415/papers/Kalal-PAMI.pdf. [Último acceso: 05 10 2020].

[54] Anónimo, «Wikipedia,» 12 2006. [En línea]. Available: https://es.wikipedia.org/wiki/Seguimiento_de_objetos. [Último acceso: 05 10 2020].

[55] «unipython,» 2018. [En línea]. Available: https://unipython.com/calibracion-la-camara-opencv/. [Último acceso: 11 10 2020].

[56] anónimo, «Unipython,» 01 2018. [En línea]. Available: https://unipython.com/calibracion-la-camara-opencv/. [Último acceso: 30 11 2020].

[57] fuente anónima, «Github,» no señala. [En línea]. Available: http://facundoq.github.io/courses/images/res/03_numpy.html. [Último acceso: 30 11 2020].

[58] Gl4r3, «robologs,» 24 06 2019. [En línea]. Available: https://robologs.net/2019/06/24/buscar-el-centro-de-un-contorno-con-opencv-python/. [Último acceso: 26 12 2020].

[59] Luis, «luisllamas.es,» blog, 25 01 2016. [En línea]. Available: https://www.luisllamas.es/controlar-arduino-con-python-y-la-librería-pyserial/. [Último acceso: 27 12 2020].

[60] J. Flynt, «3dinsider,» 15 06 2020. [En línea]. Available: https://3dinsider.com/drone-altitude-hold/. [Último acceso: 22 11 2020].

[61] M. A.-F. L.-G. L.-S. Valdebenito, «Comparativa de técnicas de punto de máxima potencia en paneles fotovoltaicos,» Talca, 2020.

Anexos

Primer Dia de pruebas

Link 1: https://alumnosutalca-my.sharepoint.com/:v:/g/personal/josses15_alumnos_utalca_cl/Ea1ErvAiNHhCtu_0VcRLT0oBvJG1hbem0PB-0jB5JP3XQg

Link 2: https://alumnosutalca-my.sharepoint.com/:v:/g/personal/josses15_alumnos_utalca_cl/EeX7wgsuQdlIj0EJOXXb8egBq9JLI4AhcO0P7Z1DWIJCCw?e=rc1etQ

Segundo Dia de pruebas

Link 3: https://alumnosutalca-my.sharepoint.com/:v:/g/personal/josses15_alumnos_utalca_cl/EbYV-gL4lzxFuhqkvH9No8UB_XZm4NcQF8zqsBp80l8O1Q?e=Vldnmb