Identificación y Detección de la Distancia de la Persona mas Próxima a un Drone

Valentina Hernández
Felipe Gutierrez
Manuel Rojas
Facultad de Ingeniería
Universidad Nacional de Colombia

25 de abril de 2023

1 Introducción

1.1 Contextualización del problema

Los aviones pilotados a distancia, o RPAS (por sus siglas en inglés), son aeronaves que son controladas de forma remota por un operador en tierra[13]. Estas aeronaves han evolucionado considerablemente desde sus inicios en la década de 1917, cuando los primeros aviones a control remoto fueron utilizados para prácticas de entrenamiento militar. A lo largo del tiempo, se han desarrollado RPAS más avanzados que han sido utilizados para una amplia variedad de aplicaciones, desde misiones militares y de reconocimiento hasta la entrega de paquetes y la vigilancia de la vida silvestre[2]. Hoy en día, los RPAS son una tecnología importante en numerosos campos, y su uso continúa expandiéndose. Sin embargo, para este tipo de vehículos los errores humanos de los pilotos es una de las falencias que pone en peligro el crecimiento de esta tecnología y su introducción en diferentes campos de la vida cotidiana[32]. La migración de concepto de producir RPA a UAV (unmanned aerial vehicle) es la posibilidad de otorgar una autonomía a los vehículos aéreos programados u orientados a control por autorización puntual o seguimiento de órdenes sin la necesidad de conocer acerca de la conducción de estos dispositivos[34].

Las controladoras de vuelo actuales son elementos sofisticados capaces de mantener en el aire un drone bajo una serie de instrucciones programadas. La posibilidad de otorgar autonomía se puede realizar a través de la identificación de escenarios en los que se requiera que el sistema tome decisiones basadas en una programación establecida. Dentro de estos posibles escenarios está la identificación de personas permite la inclusión de los drones en diversas aplicaciones para evitar colisiones, o accidentes en los que se vea perjudicada la integridad de un humano.

La gama de aplicaciones en las que puede ser requerido incluyen localización de personas perdidas desde vistas aéreas[5], vigilancia e inspección de lugares[14], identificación de posición de personas en desastres naturales o incendios[1], además todas las aplicaciones internas como el transporte de paquetes o alimentos entre oficinas, inspecciones militares o de seguridad al interior de complejos de difícil acceso o confinados[9].

El reconocimiento de personas es una utilidad de la revolucionaria inteligencia artificial, esta herramienta es una utilidad de la rama de Object Detection[20]. Esto consiste en la localización de la posición de una persona en una imagen. De manera puntual la detección funciona gracias a un nivel alto de tareas de Machine

Learning. Sin embargo, la integración de esta tecnología con vehículos aéreos no tripulados ha estado enfocada al ámbito militar mientras que en tareas cotidianas dado su alto costo de procesamiento en vuelo y la eficiencia en términos de peso que se sacrifica ha limitado la detección a sensores de proximidad con diferentes principios de funcionamiento[29].

Para alcanzar el reto de hacer control mediante inteligencia artificial en drones y que estos dispositivos estén en la capacidad de tomar decisiones a partir de la detección de personas es necesario hacer proyectos por etapas que permitan alcanzarlo. De tal manera que este proyecto desarrolla la etapa inicial de detección de personas a través de un método de visión de máquina y la identificación de la distancia de la persona más próxima a un UAV que porta el dispositivo de detección.

1.2 Soluciones existentes

Inicialmente hay una gran cantidad de empresas en todo el mundo que se encargan de hacer desarrollos de soluciones para este tipo de problemáticas. Por ejemplo, gradiant ofrece un servicio llamado Sense UAV[28]. Este consiste en la adición de una capa de seguridad sobre una flota de drones para alcanzar una detección automática de objetos y tráfico, una solicitud en específico puede ser la detección de seres humanos sin embargo la solución en específico se queda corta al no entregar el dato de la distancia que detecta y hacer que sea específicamente unos requerimientos de diseño los parámetros que definen la programación interna del algoritmo.

En operaciones de rescate Liu C. y Szirányi T. (2021)[26] realizaron un trabajo con UAV's para la detección de personas en zonas boscosas o tropicales. Lo que busca el desarrollo es que a través de gestos corporales se establezca una comunicación de solicitud de rescate. Para soluciones de características corporales para establecer comunicaciones biométricas, como YOLO3-tiny para la detección humana. La diferencia en el trabajo de Liu es que posterior al reconocimiento de un humano el sistema se acerca y pasa a una fase detección de gestos, mientras que este proyecto se debe pasar a una etapa de identificación de distancia. Una vez el sistema de rescate alcanza una precisión del 99,8% en la prueba de datos recogidos de los gestos corporales y una precisión del 94,71% en la prueba de datos en el conjunto de datos de gestos de la mano mediante el uso del método de aprendizaje profundo.

Así como el anterior otros estudios como el de Sambolek S. e Ivasic-Kos M.[24] o el de Akshatha K.R. y otros[17], tratan en general la detección de humanos y objetos como fin del UAV sin embargo no se realiza una observación adicional con esta extracción ya que están enfocados en determinar la efectividad del algoritmo de detección basándose en una serie de datasets con los que se entrenó y probó el determinado algoritmo.

Por otro lado, el trabajo de Asanka G. y otros (2018) si interviene la posibilidad de extraer información de la distancia que tiene la persona respecto al UAV en este caso con el objetivo de predecir la pose y trayectoria del individuo[8]. Este paper indica que Asanka usa para el entrenamiento un método de Region-based Convolutional Neural Network (R-CNN), cada cuadro de video es sometido a una corrección de perspectiva mediante projective transformation. Al final de este proceso y una serie de procedimientos adicionales el algoritmo alcanza un 98% de precisión para la detección de los individuos.

Otros trabajos tratan el enfoque de la detección de objetos en UAV's desde un punto de vista más especializado hacía la navegación de estos dispositivos, pero en una configuración de enjambre; lo que conlleva ya no solo la evasión de obstáculos presentes a su alrededor sino también de otros drones que acompañan su vuelo. La planeación de la trayectoria es, por lo tanto, su enfoque inicial, y para ello se necesita de una adquisición de

datos de su entorno adecuada para generar un desarrollo satisfactorio y libre de colisiones.

Xin Zhou et al. (2022) [35] plantean un conjunto de drones que, sin GPS, sin captura de movimientos, sin ayuda de cómputo adicional desde una estación de tierra y sin ningún mapeo previo de la zona de vuelo; pueden desplegar una trayectoria en un entorno congestionado sin generar ningún tipo de colisión con objetos de alrededor ni entre sí. Para ello se basan en la utilización de una cámara estéreo RealSense D430 que se encarga de la adquisición de datos visuales y de profundidad, además de la planeación de trayectorias mediante un algoritmo propio, manteniendo una formación inicial predeterminada por el usuario, y que resulta más eficiente, al menos por lo datos mostrados, comparados con otros de una mayor complejidad de cómputo, tales como EGO, MADER, SCP y RBP. Enrica Soria (2022) [30] es otra autora que plantea un desarrollo similar con aplicaciones igualmente interesantes.

1.3 Propuesta de trabajo

A medida que el curso avanza se desean utilizar los conceptos estudiados y aplicarlos conforme al desarrollo de las temáticas, para ello se propone dividir el proyecto de tal forma que se adecue a la metodología y temas vistos y/o información adquirida producto de consultas de fuentes de información externa.Como propuesta se plantea seguir con los pasos expuestos a continuación de manera secuencial, sin querer decir de alguna manera que una vez avanzados no se puedan regresar a ninguno de ellos, de ser el caso, se reevaluará la opción de complementar en alguno de ellos según sea la conveniencia o por indicación del docente.

1. Identificación de dispositivos de video que pueden integrarse a un dron como sistema de evasión de obstáculos.
2. Consulta de información, programas y librerías necesarias para ejecutar funciones de detección y evasión de obstáculos.
3. Selección y adquisición de cámara.
4. Identificación de interfaces, conexiones y comunicación entre computadora y cámara (calibración).
5. Pruebas de código (adquisición de imágenes o datasets, preprocesamiento de imágenes y obtención de características para su clasificación).
6. Uso de características para generar decisiones en el algoritmo del sistema de evasión de obstáculos.

2 Estado del Arte

2.1 Detección de Objetos

Justo de Frías C. presenta en su trabajo una revisión histórica en la que se incluye que el estudio de la visión artificial empezó a mediados del siglo XX, gracias a la idea de permitirle a un ordenador hacer lo que hacían los ojos humanos[33]. En la década de 1960, se lograron buenos resultados en la detección de objetos mediante visión de máquina en experimentos controlados, como la identificación de objetos y su posición en imágenes binarias. Sin embargo, la tecnología disponible en aquel entonces no permitía procesamientos más avanzados y la investigación posterior no tuvo éxito. A pesar de esto, se desarrollaron algoritmos importantes, como los detectores de bordes de Roberts, Prewitt y Sobel, que siguen utilizándose hoy en día. La detección de objetos mediante visión de máquina es un problema complejo, pero la investigación continua en este campo ha llevado a importantes avances y aplicaciones prácticas. Dadas las bajas capacidades durante los 70 hubo una época de pesimismo que disminuye la cantidad de investigaciones al contemplar lo complejo que era lo que hacia el cerebro en el procesamiento de imágenes. En los años 80, la investigación en detección de objetos mediante

visión de máquina resurgió con fuerza, gracias al desarrollo de ordenadores más potentes y sofisticados, y a la aparición de hardware específico para el procesamiento de imágenes. Desde entonces, la investigación en este campo ha avanzado rápidamente y se utiliza en una amplia variedad de aplicaciones prácticas.

La detección de objetos mediante visión de máquina es una técnica utilizada en inteligencia artificial para identificar y localizar objetos específicos en una imagen o video. La detección de objetos es una tarea crítica en aplicaciones de visión de máquina, como la seguridad, el seguimiento de objetos y la robótica. Los algoritmos de detección de objetos se basan en el análisis de características clave en la imagen, como la forma, el color y la textura, para identificar objetos y su ubicación. Hay varios enfoques populares utilizados en la detección de objetos, que incluyen la detección basada en regiones, la detección de puntos de referencia y la detección de objetos a través de redes neuronales. Cada enfoque tiene sus propias fortalezas y debilidades, y la elección del enfoque depende en gran medida de la aplicación específica y las necesidades del usuario. La detección de objetos mediante visión de máquina es una técnica poderosa que se está utilizando cada vez más en una amplia gama de aplicaciones, y se espera que siga siendo una área de investigación y desarrollo importante en los próximos años[21].

La detección basada en regiones es un enfoque que divide la imagen en regiones propuestas, utilizando algoritmos de segmentación y selección de características, y luego utiliza un modelo de clasificación para determinar si cada región contiene un objeto de interés o no. Algunos ejemplos de algoritmos de detección basados en regiones son R-CNN, Fast R-CNN y Faster R-CNN.

El primer método exitoso de detección de objetos basado en regiones fue R-CNN (Red Neuronal Convolucional Basada en Regiones), propuesto en 2014[12]. Extrajo 2000 regiones de la imagen en lugar de un número excesivamente grande de regiones previamente utilizadas, y clasificó cada región con una de las clases conocidas. Aunque el proceso fue largo, sentó las bases para el nuevo modelo Fast R-CNN, que eliminó el proceso de generar regiones y aceleró significativamente los tiempos de entrenamiento y prueba. Posteriormente, se introdujo un modelo aún más mejorado llamado Faster R-CNN, que no utiliza Selección Selectiva y utiliza una Red de Propuestas de Regiones para identificar regiones a alta velocidad, reduciendo aún más los tiempos de prueba. Cuando hay un proceso de solapamiento de regiones se pueden utilizar métodos como la “Non-Maximun-Supression” para alcanzar un resultado final con una sola y posiblemente la mejor región de interés[19].

El enfoque basado en redes neuronales convolucionales (CNN) es uno de los enfoques más efectivos y populares para la detección de objetos. Este enfoque utiliza una red neuronal convolucional para detectar objetos en la imagen. La red neuronal convolucional aprende características de los objetos a partir de los datos de entrenamiento y utiliza esas características para detectar objetos en la imagen. Este enfoque se ha demostrado muy efectivo y rápido en la detección de objetos en imágenes.

Otra poderosa técnica es You Only Look Once (YOLO) es una red que hace una pasada por la red convolucional para detectar los objetos para los que ha sido entrenada para clasificar. Esto permite alcanzar altas velocidades sin ordenadores de alta potencia. Esto permite la detección en video en vivo y su ejecución incluso en dispositivos móviles.

Yolo de manera general define una grilla de 13 x 13 casillas. Sobre este conjunto tratará de detectar objetos valiéndose de anchor fijos en escalas distintas. También utiliza métricas como IoU y técnicas como Non-Max-suppression. Al final tiene asociada una red de regresión.

La magia de YOLO consiste en su red CNN. Ya que utiliza esta misma red de clasificación con una técnica de strided convolution de tal manera que no requiere iterar la grilla si no que la misma red realiza un tipo

de offset que le permite detectar en simultáneo las 169 casillas.

Otra implementación popular de este enfoque es Single Shot MultiBox Detector (SSD). SSD es similar a YOLO en el sentido de que utiliza una red neuronal convolucional para detectar objetos en la imagen. Sin embargo, en lugar de dividir la imagen en una cuadrícula, SSD utiliza varias escalas de características para detectar objetos en la imagen. Esto hace que SSD sea más preciso que YOLO, pero también un poco más lento.

En general, el enfoque basado en redes neuronales convolucionales es uno de los enfoques más efectivos para la detección de objetos en la actualidad. Si estás interesado en aprender más sobre este enfoque, te recomendamos que consultes los recursos que mencionamos anteriormente[19].

El enfoque basado en redes neuronales completamente convolucionales (FCN) para la detección de objetos en imágenes es un método popular en la visión por computadora. A diferencia de los enfoques tradicionales que utilizan regiones de interés o características manuales, FCN utiliza una arquitectura completamente convolucional para detectar objetos en la imagen. Esto significa que la red puede procesar la imagen completa de manera eficiente en lugar de realizar la detección en regiones de interés predefinidas.

En el enfoque FCN, se entrena una red neuronal para asignar a cada píxel de la imagen una etiqueta de clase que indica la presencia o ausencia de un objeto en esa ubicación[6]. La red se compone de varias capas convolucionales seguidas de capas de agrupación y una capa de convolución final que produce una salida de la misma dimensión que la entrada original. Esta salida representa una máscara que indica la ubicación de los objetos en la imagen.

Una ventaja de este enfoque es que es capaz de detectar objetos de diferentes tamaños y formas sin necesidad de definir regiones de interés predefinidas. Además, la arquitectura completamente convolucional de la red significa que puede ser entrenada para detectar múltiples categorías de objetos.

Algunos ejemplos de redes neuronales completamente convolucionales utilizadas para la detección de objetos incluyen SegNet[27], U-Net[31] y DeepLab[4]. Cada una de estas redes utiliza una arquitectura diferente, pero comparten la capacidad de detectar objetos a múltiples escalas sin la necesidad de regiones de interés predefinidas.

Todas estas técnicas representan el estado anterior y actual de las técnicas de detección de objetos, una gran cantidad de conceptos y metodologías extensas y de mucho estudio, ya que aunque se encuentren códigos y tutoriales de uso rápido, la correcta, apropiada y eficiente forma de utilizar cada uno de estos conjuntos de herramientas depende del conocimiento y justificación de cada una de las elecciones en los argumentos y selección de las funciones.

2.2 Camaras

Los drones están equipados con diferentes tipos de tecnologías de cámara para capturar imágenes aéreas y secuencias de vídeo. Estas son algunas de las tecnologías de cámara más utilizadas en drones:

- Cámaras RGB: estos son los tipos de cámaras más comunes que se utilizan en los drones y capturan la luz visible en el espectro rojo, verde y azul. Las cámaras RGB se utilizan para fotografía aérea básica, cartografía y topografía.

- Cámaras térmicas: las cámaras térmicas capturan la radiación infrarroja y se utilizan para detectar firmas de calor. Se utilizan comúnmente para aplicaciones tales como operaciones de búsqueda y rescate, inspecciones de edificios y monitoreo de vida silvestre.
- Cámaras multiespectrales: las cámaras multiespectrales capturan la luz en múltiples bandas del espectro electromagnético, incluida la luz infrarroja, ultravioleta y visible. Se utilizan para aplicaciones tales como agricultura de precisión, silvicultura y monitoreo ambiental.
- Cámaras hiperespectrales: las cámaras hiperespectrales capturan la luz en cientos de bandas espectrales estrechas y se utilizan para aplicaciones como la exploración de minerales, el mapeo geológico y la gestión de cultivos.
- LIDAR: Light Detection and Ranging (LIDAR) utiliza láseres para crear un mapa 3D del terreno debajo del dron. LIDAR se utiliza para aplicaciones tales como mapeo de terreno, silvicultura y construcción.

Teniendo un panorama general de las tecnologías disponibles principales que se usan en los UAV's es importante revisar a profundidad aquellos que están enfocados en las labores de detección de superficies o mapas tridimensionales y que responden al interés principal de este proyecto. En estas categorías se pueden obtener tres tipos de tecnologías en cámaras que resaltan sobre las demás (cámaras RGB-D, cámaras estéreo y cámaras LIDAR). A continuación, se trata de ofrecer un listado de las cámaras más representativas del mercado, cuyo uso está enfocado en el despliegue de drones, para que den una pauta en cuanto a sus características más importantes y como estás pueden ser aplicadas en nuestro proyecto.

2.2.1 Cámaras RGB-D

- La cámara Intel RealSense Depth Camera D435i es una cámara RGB-D que permite capturar imágenes en color y profundidad simultáneamente. Tiene una resolución de 1920x1080 para las cámaras RGB y una resolución de hasta 1280x720 para la cámara de profundidad, lo que le permite capturar imágenes detalladas y precisas. Además, cuenta con un sensor de profundidad infrarroja que permite una captura de profundidad precisa de hasta 2 metros. La cámara también incluye un acelerómetro y un giroscopio, lo que permite la captura de datos de movimiento en tiempo real, y utiliza tecnología de profundidad activa para capturar datos de profundidad incluso en condiciones de baja luz.
- La cámara Stereolabs ZED es una cámara estéreo RGB-D que permite capturar imágenes en color y profundidad simultáneamente. La cámara tiene una resolución de 4 megapíxeles para las cámaras RGB y puede capturar profundidades de hasta 20 metros con una precisión de hasta 1,5 cm. La cámara también cuenta con un procesador integrado para reducir la carga de procesamiento en el dispositivo de host y un modo de baja latencia para aplicaciones en tiempo real.
- La cámara Azure Kinect DK es una cámara RGB-D diseñada para aplicaciones de visión artificial y robótica. Tiene una resolución de 12 MP para las cámaras RGB y una resolución de hasta 1 MP para la cámara de profundidad, lo que permite la captura de imágenes detalladas y precisas. Además, cuenta con un micrófono de matriz espacial de siete canales para la captura de audio en 3D. La cámara utiliza tecnología de profundidad activa para capturar datos de profundidad incluso en condiciones de baja luz y cuenta con una amplia gama de interfaces para la conexión con dispositivos de host y una unidad IMU.

2.2.2 Cámaras Estéreo

- La cámara LI-AR0234CS-STEREO-GMSL2-30 de Leopard Imaging es una cámara estéreo de alta resolución diseñada para aplicaciones en robótica, vehículos autónomos y realidad aumentada. Cuenta con dos sensores de imagen AR0234CS CMOS de alta calidad que permiten la captura de imágenes de hasta 1920x1200 píxeles y una profundidad de hasta 8 metros. La cámara también cuenta con una

interfaz de alta velocidad GMSL2 para la transmisión de datos de imagen y un procesador de señal digital para el procesamiento de la imagen. Además, la cámara es compatible con una amplia gama de software de visión artificial y está diseñada para ser fácilmente integrable en sistemas existentes.

- La cámara Luxonis OAK-D-IoT-40 es una cámara que cuenta con un sensor RGB de 12 MP y un sensor de profundidad con un rango de detección de hasta 6 metros. La cámara también tiene un procesador Intel Myriad X VPU para el procesamiento de imagen en tiempo real y una interfaz USB-C para la conexión con dispositivos de host. Además, la cámara es compatible con una variedad de frameworks de aprendizaje profundo, como TensorFlow y PyTorch, y está diseñada para ser fácilmente integrable en sistemas existentes.
- La cámara IMX219-83 Stereo es una cámara estéreo diseñada con dos sensores de imagen IMX219 de alta calidad que permiten la captura de imágenes de hasta 3280x2464 píxeles. La cámara también cuenta con una interfaz MIPI CSI-2 para la transmisión de datos de imagen y una lente intercambiable de 77 grados para una mayor flexibilidad en la captura de imágenes.

2.2.3 Cámaras LIDAR

- La cámara Velodyne VLP-16 es una cámara LiDAR que cuenta con 16 canales de láser y un campo de visión de 360 grados, lo que permite una cobertura completa de la zona circundante en tiempo real. La cámara tiene una tasa de muestreo de hasta 300.000 puntos por segundo y una resolución de hasta 0,2 grados, lo que permite una detección precisa de objetos y una navegación segura. Además, la cámara es resistente al agua y al polvo, lo que la hace adecuada para su uso en ambientes adversos.
- La cámara Livox Mid-40 es una cámara LiDAR diseñada para aplicaciones en robótica, mapeo y exploración. Cuenta con un escáner de alta velocidad y precisión que permite una tasa de muestreo de hasta 100.000 puntos por segundo y un rango de detección de hasta 260 metros. La cámara tiene un campo de visión de 38,4 grados y una resolución de hasta 0,1 grados, lo que permite una detección precisa de objetos y una navegación segura. Además, la cámara es compacta, liviana y consume menos energía que las cámaras LiDAR convencionales, lo que la hace adecuada para su uso en drones y otros sistemas móviles.

2.3 Propuestas de Solución

En general las propuestas de solución consisten en trabajar con imágenes tomadas del video en tiempo real cada 2 segundos con 8 tonos de gris. Sin embargo estas características están sujetas a cambios según las necesidades que se evidencien en el proceso como la posibilidad de utilizar el video en tiempo real o la necesidad de mas o menos todos de gris o la utilización de los canales de color. Posterior a este procedimiento aplicar filtros derivativos que permitan hacer un realce de los bordes y alcanzar una mejor detección de las posibles figuras del cuerpo humano. Aquí surgen varias posibles opciones de trabajo:

- Hacer un proceso de esqueletización binaria para alcanzar luego hacer un trabajo de identificación con redes neuronales y luego obtener las coordenadas de las posiciones de individuos en la imagen para solicitar la distancia a la cámara estéreo y mediante una serie de condicionales obtener la posición de la persona más cercana.
- Otra posibilidad es utilizar YOLO para la detección de las personas y hacer un entrenamiento con uno o más Datasets de Kaggle que contengan figuras de humanos en diferentes entornos y con diferentes poses y posteriormente realizar de forma idéntica a la alternativa anterior la detección de el humano más proximo[23].

- Una posibilidad adicional es un método de machine learning basado en una función en cascada[22] propuesta por Paul Viola y Michael Jones en el que se entrena mediante una serie de imágenes en positivo y negativo[3] Inicialmente, el algoritmo necesita muchas imágenes positivas (imágenes con el objeto a detectar) e imágenes negativas (imágenes sin el objeto a detectar) para entrenar al clasificador. posteriormente se extraen los kernel convolucionales. Esto usa luego este entrenamiento para identificar en nuestro caso a los humanos en diferentes entornos. Para proceder a extraer coordenadas y extraer la distancia con el sujeto más cercano.
- Otra posible solución es una vez pre-procesadas las imágenes hacer uso del framework MediaPipe[7] un toolkit construido en machine learning usado para procesar datos de series temporales que en su versión open source ofrece una serie de modelos de TensorFlow pre-entrenados para la detección de entre otros la pose de seres humanos[10] Una vez detectadas estas regiones se procede a realizar la identificación de la distancia de cada una de las personas identificadas en video y la determinación de la distancia más corta a un individuo detectado[25]
- Las cámaras estereográficas pueden ser usadas para distintas funcionalidades, una de estas utiliza la segmentación semántica para la identificación de regiones. Estas regiones se pueden clasificar según se convenga y tienen la ventaja de que pueden detectar objetos o regiones de objetos desconocidos, es decir, aquellos que la red neuronal no haya sido entrenada para identificar y tomarlas en consideración. Este principio es especialmente útil en aplicaciones en donde la navegación con un sistema de evasión de obstáculos es fundamental [18].
- Otra aproximación hace referencia a la utilización de un red neuronal para la detección de objetos combinada con un mapeo de profundidad de la imagen. Es decir que en esencia, se podrían identificar regiones de interés previamente definidas para que sean resaltadas y a las cuales se les va a realizar una estimación de la profundidad a la cual se encuentran con respecto al dispositivo de adquisición de imágenes con el fin de acotar la cantidad de elementos a ser tenidos en cuenta dentro de la navegación [15].

3 Materiales y métodos

3.1 Computador

- Procesador Intel 11th Gen. I5 11300H
- 16 Gb RAM
- Sistema operativo 64 bits Windows 11
- RTX 3050 Laptop

3.2 Camara Stereografica Luxonis OAK-D-IoT-40 [16]

- 12MP, 4K / 60Hz
- 2 cámaras monocromáticas 720P / 120Hz
- Procesador de visión Myriad X MA2485
- Interfase cámara 2x 2-lane MIPI
- Interfase cámara 1x 4-lane MIPI
- Interfase Quad SPI con dos 2 selectores dedicados

- Interfase I²C
- Interfase UART
- Interfase USB2 y USB3
- GPIO (1.8 V y 3.3 V)
- Almacenamiento: Flash NOR integrado opcional y / o EEPROM
- Compatible con OpenVINO
- API de Python y C ++
- Capacidad de procesamiento: 4 TOPS (1.4 TOPS para AI)
- Potencia: entrada de alimentación única de 5 V o 3,3 V
- Dimensiones: 40 x 30 x 6 mm

3.3 Métodos

3.3.1 Librerías

A continuación se listan las librerías que se van a usar y algunas que probablemente tengan utilidad en el desarrollo del proyecto:

- OpenCv
- Numpy
- Matplotlib
- Depthai
- Pytorch
- TensorFlow

4 Avances

Como ya se ha mencionado con anterioridad, se hace uso de la cámara Luxonis OAK-D-IoT-40 como sensor de adquisición de imágenes y vídeo. Esta cámara cuenta con un tipo de visión estereográfica de tipo pasiva en la cual mide la disparidad de un par de pixeles que representan el mismo objeto en la vida real para determinar su profundidad a través de un cálculo matemático, el cual implica conocer la distancia focal, su línea base (distancia entre las dos cámaras) y la disparidad entre píxeles.

Vale la pena mencionar que la detección de profundidad en cámaras estereográficas pasivas tiene una limitante cuando se enfrenta a superficies planas o con entradas donde la luz sea demasiado intensa. Esto se debe a que uno de los principios que se utiliza para calcular la profundidad se ve afectado; para medida de la disparidad se necesita identificar el mismo punto en la imagen izquierda como en la imagen derecha, sino existen puntos característicos que se puedan relacionar puede ocasionar que la lectura sea errónea.

Teniendo en cuenta cómo funciona el dispositivo se procede a realizar una calibración del par de cámaras, que hasta el momento se consideran independientes, para que esta funcione como un solo sensor de adquisición de imágenes con capacidades de detección de profundidad.

El procedimiento para la calibración consiste en utilizar la intersección entre los píxeles que se correlacionan entre sí de las dos imágenes obtenidas por las cámaras, para hallar la orientación y la distancia a la tabla que se muestra en la imagen. El éxito de este paso depende de que la imagen que observan las cámaras sea clara y que se encuentre sobre una superficie plana para que no existan curvaturas y ondas que afecten la medición.

Para está familia de cámaras de la empresa Luxonis se proporciona un repositorio en GitHub el cual contiene los códigos necesarios para realizar estos primeros pasos. Inicialmente se pide que se clone el repositorio de forma local y se ejecute un script el cual instala en el equipo una serie de librerías y de programas los cuales son necesarios para el correcto funcionamiento de todas las funcionalidades del dispositivo.

Uno de estos códigos corresponde al algoritmo de calibración. Al ejecutarlo se pide que se tomen una serie de imágenes que corresponden a capturas de la tabla de charuco en distintas posiciones y orientaciones. En total se piden 13 imágenes para realizar un correcto proceso de calibración y con el cual, al ser finalizado, permite el uso de la cámara en condiciones adecuadas.

El API de DepthAI permite configurar la comunicación con la cámara disponible. Para ello, se basa en la generación de un flujo llamado "pipeline". Este pipeline conecta las tres cámaras con el host y permite la obtención de imágenes, tal y como se observa en la figura 1 .

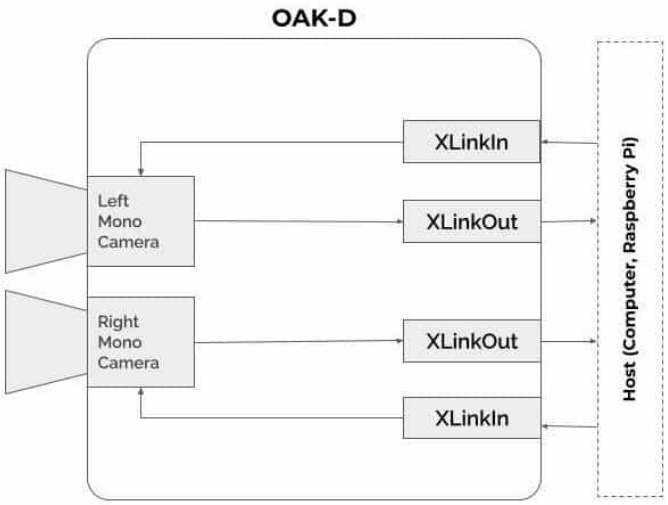

Figura 1: Pipeline DepthAI [11]

Por ello, con el fin de poder capturar las imágenes en vivo para la detección, se realizó la configuración de la cámara para poder capturar imágenes a color y capturar las imágenes monocromáticas a través de cada una de las cámaras estereográficas. Para esto, se creó una clase con la cual se pudiera instanciar cualquiera de las dos cámaras:

```
class Camera():

    def __init__(self, usb2mode = True):
        self.usb2mode = usb2mode
```

El argumento `usb2mode` se usa para indicar el protocolo de comunicación para la cámara, el cual depende del host y que por defecto es USB3. En este caso, el computador usa USB2 en lugar de USB3.

A continuación, se define un método para configurar las cámaras monocromáticas y poder obtener la imagen de cada una.

```
    def getMonoCamera(self, pipeline, isLeft):

        mono = pipeline.createMonoCamera()
        mono.setResolution(dai.MonoCameraProperties.SensorResolution.THE_800_P)

        if isLeft:
            mono.setBoardSocket(dai.CameraBoardSocket.LEFT)
        else:
            mono.setBoardSocket(dai.CameraBoardSocket.RIGHT)

        return mono
```

La función anterior es una función de ayuda para crear un nodo para cada cámara monocromática. Con la cual, dependiendo de un parámetro definido como `isLeft`, se asigna la cámara de la derecha o de la izquierda según se necesite. También se configura la resolución de la cámara a 1280x800p. Por último se retorna el nodo configurado.

```
    def createStereoCamera(self):
        pipeline = dai.Pipeline()
        monoR = self.getMonoCamera(pipeline, isLeft = False)
        monoL = self.getMonoCamera(pipeline, isLeft = True)

        xoutL = pipeline.createXLinkOut()
        xoutL.setStreamName("left")

        xoutR = pipeline.createXLinkOut()
        xoutR.setStreamName("right")

        monoL.out.link(xoutL.input)
        monoR.out.link(xoutR.input)

        return dai.Device(pipeline, usb2Mode=self.usb2mode)
```

Posterior a la creación de los nodos, se crea cada una de las cámaras nombradas como `monoR` y `monoL`. Para cada una se asigna y se crea el `xLinkOut`, el cual representa la salida de la cámara. El método retorna un ”device”, que en este caso es una cámara configurada para usar las dos cámaras monocromáticas y el cual está listo para comunicarse con el host.

```
camera = Camera()
device = camera.createStereoCamera()

leftQueue = device.getOutputQueue(name="left", maxSize=1)
rightQueue = device.getOutputQueue(name="right", maxSize=1)

cv.namedWindow("Stereo Pair")

sideBySide = False
images = 0
```

Dentro del programa principal, creamos la cámara y obtenemos un ”device”, el cual representa la cámara estereográfica. Creamos una cola de petición para cada una de las cámaras con el fin de poder solicitar el frame o imagen correspondiente. Además, se configura el nombre de la ventana donde aparecerán las imágenes.

```
while True:

```

```
    leftFrame = camera.getFrame(leftQueue)
    rightFrame = camera.getFrame(rightQueue)

    if sideBySide:
        imOut = np.hstack((leftFrame, rightFrame))
    else :
        imOut = np.uint8(leftFrame/2 + rightFrame/2)

    cv.imshow("Stereo Pair", imOut)
    cv.imwrite(f"./images/{images}.jpg", imOut)

    key = cv.waitKey(1)
    if key == ord('q'):
        break
    elif key == ord('t'):
        sideBySide = not sideBySide
```

Por último, se crea un ciclo para tomar las imágenes. Se obtiene cada una de las tomas correspondientes a la cámara derecha y a la izuqierda, y se muestran, dependiendo de la elección del usuario, una al lado de la otra, o sobrepuestas. Por último, las imágenes se guardan en una carpeta para poder usarlas después.

En las figuras 2 y 3 se muestran las imágenes obtenidas con las cámaras monocromáticas.

Figura 2: Imágenes obtenidas por las cámaras izquierda y derecha una al lado de la otra

Figura 3: Imágenes obtenidas por las cámaras izquierda y derecha sobrepuestas

El proceso para configurar la cámara RGB es similar:

```
    def createColorCamera(self, previewSize = (500,500)):
        pipeline = dai.Pipeline()
        cam_rgb = pipeline.create(dai.node.ColorCamera)
        cam_rgb.setPreviewSize(previewSize)
        cam_rgb.setInterleaved(False)

        xout_rgb = pipeline.create(dai.node.XLinkOut)
        xout_rgb.setStreamName("rgb")
        cam_rgb.preview.link(xout_rgb.input)

        return dai.Device(pipeline, usb2Mode=self.usb2mode)
```

En este caso, solo se debe configurar un nodo correspondiente a la cámara a color. Nuevamente se retorna un ”device” listo para ser usado por el host con el cual se obtienen las imágenes a color, tal y como se muestra en la figura 4. Igual que en el caso anterior, las imágenes se guardan en una carpeta.

Figura 4: Imagen a color

Finalmente, a las imágenes obtenidas se les pueden aplicar filtros para obtener algunas características, como por ejemplo bordes. Estos filtros pueden ser aplicados a cada imagen después de tomarlas o mientras la cámara las captura, tal como se ve en las figuras 5 y 6.

Figura 5: Imagen con filtro Sobel aplicado

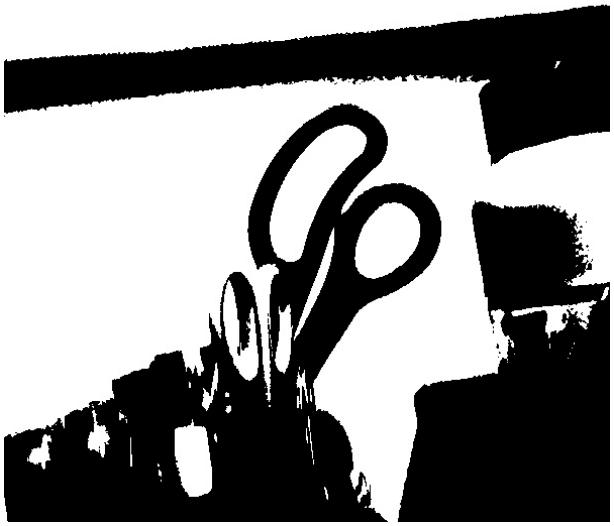

Figura 6: Imagen umbralizada

Referencias

[1] *¿Sabes qué pueden hacer los drones por nosotros?* URL: https://www.iberdrola.com/innovacion/drones-usos-tipos.

[2] *A Brief History of Drones*. URL: https://www.iwm.org.uk/history/a-brief-history-of-drones.

[3] *Cascade Classifier*. URL: https://docs.opencv.org/3.4/db/d28/tutorial_cascade_classifier.html.

[4] *DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs*. URL: https://arxiv.org/abs/1606.00915#.

[5] *Drones aprovechan la IA para localizar a las personas perdidas*. URL: https://www.kionetworks.com/blog/drones-aprovechan-la-ia-para-localizar-a-las-personas-perdidas.

[6] *Fully Convolutional Networks for Semantic Segmentation*. URL: https://arxiv.org/abs/1605.06211.

[7] *human body detection OpenCV python*. URL: https://www.youtube.com/watch?v=x70ALxNzKto&ab_channel=HaiderTech.

[8] *Human Detection and Motion Analysis from a Quadrotor UAV*. URL: https://iopscience.iop.org/article/10.1088/1757-899X/405/1/012003/pdf.

[9] *Indoor inspections and applications*. URL: https://www.medexon.com/our-services/indoor-applications/.

[10] *Introduction to MediaPipe*. URL: https://learnopencv.com/introduction-to-mediapipe/.

[11] *Introduction to OAK-D and DepthAI*. URL: https://learnopencv.com/introduction-to-opencv-ai-kit-and-depthai/.

[12] *Introduction to object detection with deep learning*. URL: https://www.superannotate.com/blog/object-detection-with-deep-learning.

[13] *Introduction to Remotely Piloted Aircraft Systems (RPAS)*. URL: https://www.skybrary.aero/articles/introduction-remotely-piloted-aircraft-systems-rpas.

[14] *La privacidad en la era de los drones y la vigilancia aérea*. URL: https://caseguard.com/es/articles/la-privacidad-en-la-era-de-los-drones-y-la-vigilancia-aerea/.

[15] Luxonis. *Spatial AI*. 2023. URL: https://docs.luxonis.com/en/latest/pages/spatial-ai/#spatialai.

[16] *Luxonis OAK-D-IoT-40 cámara estero visión artificial*. URL: https://dynamoelectronics.com/tienda/luxonis-oak-d-iot-40/.

[17] *Manipal-UAV person detection dataset: A step towards benchmarking dataset and algorithms for small object detection*. URL: https://www.sciencedirect.com/science/article/abs/pii/S0924271622003008.

[18] MathWorks. *Semantic Segmentation*. https://la.mathworks.com/solutions/image-video-processing/semantic-segmentation.html. 2023.

[19] *Modelos de Detección de Objetos*. URL: https://www.aprendemachinelearning.com/modelos-de-deteccion-de-objetos/.

[20] *Object Detection and Person Detection in Computer Vision*. URL: https://www.cameralyze.co/blog/object-detection-and-person-detection-in-computer-vision.

[21] *Object Detection Guide*. URL: https://www.fritz.ai/object-detection/.

[22] *Pedestrian Detection using OpenCV Python | Human Detection | Python | OpenCV*. URL: https://www.youtube.com/watch?v=jTwNerGmc1s&ab_channel=FalconInfomatic.

[23] *People-Detector*. URL: https://www.google.com/url?q=https://github.com/humandecoded/People-Detector&sa=D&source=docs&ust=1681306036074855&usg=AOvVaw1Ms52GlngVBTWEHlxXzZCT.

[24] *Person Detection in Drone Imagery*. URL: https://www.researchgate.net/publication/346687347_Person_Detection_in_Drone_Imagery.

[25] *Pose$_e$stimation*. URL: https://github.com/BakingBrains/Pose_estimation/blob/main/PoseModule.py.

[26] *Real-Time Human Detection and Gesture Recognition for On-Board UAV Rescue*. URL: https://www.mdpi.com/1424-8220/21/6/2180.

[27] *SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation*. URL: https://arxiv.org/abs/1511.00561.

[28] *SENSE UAV*. URL: https://www.gradiant.org/en/portfolio/sense-uav/?gclid=Cj0KCQjw27mhBhC9ARIsAIFsETGc3fQCXLOUJfnTyq53zK4KdJxBM1-CrqCpyqq960yOsQw1ekGQmxkaAtL9EALw_wcB.

[29] *Sensor Proximity Types*. URL: https://robocraze.com/blogs/post/proximity-sensor-types.

[30] Enrica Soria. «Swarms of flying robots in unknown environments». En: *Science Robotics* 7.66 (2022), eabq2215. DOI: 10.1126/scirobotics.abq2215. eprint: https://www.science.org/doi/pdf/10.1126/scirobotics.abq2215. URL: https://www.science.org/doi/abs/10.1126/scirobotics.abq2215.

[31] *U-Net: Convolutional Networks for Biomedical Image Segmentation*. URL: https://arxiv.org/abs/1505.04597.

[32] *UNMANNED AERIAL VEHICLES*. URL: https://web.archive.org/web/20090724015052/http://www.airpower.maxwell.af.mil/airchronicles/apj/apj91/spr91/4spr91.htm.

[33] *Visión artificial aplicada en la identificación de objetos y su parametrización geométrica*. URL: https://core.ac.uk/download/pdf/288502112.pdf.

[34] *Will Autonomous Drones Replace the Drone Pilot?* URL: https://www.galeforcedrone.com/will-autonomous-drones-replace-the-drone-pilot/.

[35] Xin Zhou et al. «Swarm of micro flying robots in the wild». En: *Science Robotics* 7.66 (2022), eabm5954. DOI: 10.1126/scirobotics.abm5954. eprint: https://www.science.org/doi/pdf/10.1126/scirobotics.abm5954. URL: https://www.science.org/doi/abs/10.1126/scirobotics.abm5954.