Universidad Tecnológica Nacional

Proyecto Final

Sistema de seguimiento autónomo para cámaras

Autores:

- *Martinez, Francisco Andrés*

Director:

- *Ing. Klimovsky, Ernesto*

Proyecto final presentado para cumplimentar los requisitos académicos

para acceder al título de Ingeniero en electrónica

en la

Facultad Regional Paraná

Fecha: agosto de 2022

Declaración de autoría

Yo declaro que el Proyecto Final “Sistema de seguimiento autónomo para cámaras” y el trabajo realizado son propios.

Declaro:

- Este trabajo fue realizado en su totalidad, o principalmente, para acceder al título de grado de Ingeniero en electrónica, en la Universidad Tecnológica Nacional, Regional Paraná.
- Se establece claramente que el desarrollo realizado y el informe que lo acompaña no han sido previamente utilizados para acceder a otro título de grado o pre-grado.
- Siempre que se ha utilizado trabajo de otros autores, el mismo ha sido correctamente citado. El resto del trabajo es de autoría propia.
- Se ha indicado y agradecido correctamente a todos aquellos que han colaborado con el presente trabajo.
- Cuando el trabajo forma parte de un trabajo de mayores dimensiones donde han participado otras personas, se ha indicado claramente el alcance del trabajo realizado.

Firmas:

-
-

Fecha: de agosto de 2022

Agradecimientos

Al ing. Ernesto Klimovsky y Sergio Burgos por haberme brindado su tiempo y conocimiento además de su predisposición en todo momento para la realización de este proyecto. Estoy profundamente agradecido a mi familia y amigos que me brindaron su ayuda, apoyo y conocimientos a lo largo del desarrollo de este proyecto. También le agradezco a la Universidad Tecnológica Nacional Facultad Regional Paraná y a los docentes de la misma, por enseñarme y apoyarme todos estos años.

Francisco Martinez

Universidad Tecnológica Nacional

Abstract

Facultad Regional Paraná

Ingeniero en Electrónica

Sistema de seguimiento autónomo para cámaras

Francisco Martinez

Abstract

A system capable of detecting and following people autonomously was designed, in order to keep a given subject within the frame of a camera. The device uses OpenCV libraries to perform image processing, and uses ROS to coordinate all robot functions. Regarding the hardware, a Raspberry Pi 4 development board with an ARM Cortex-A72 microprocessor, a Nema 17 motor and a Sony IMX219 8 megapixel camera was used. Finally, a device capable of detecting and following people autonomously was obtained, but which presents certain problems in the presence of multiple people on stage.

Keywords

Robot Operating System (ROS), OpenCV, Haar Cascade Classifier, Face Detection, Raspberry Pi

Resumen

Se diseñó un sistema capaz de detectar y seguir personas de forma autónoma, para de esta manera mantener a un sujeto determinado dentro del encuadre de una cámara. El dispositivo utiliza las librerías de OpenCV, para realizar el procesamiento de imágenes, y utiliza ROS, para coordinar todas las funciones del robot. En lo referido al hardware, se utilizó una placa de desarrollo Raspberry Pi 4 con un microprocesador ARM Cortex-A72, un motor Nema 17 y una cámara Sony IMX219 de 8 megapíxeles. Finalmente se obtuvo un dispositivo capaz de detectar y seguir personas de forma autónoma, pero que presenta ciertos problemas ante la presencia de múltiples sujetos en la escena.

Palabras clave:

Robot Operating System (ROS), OpenCV, Clasificador de Cascada Haar, Detección Facial, Raspberry Pi

Índice

Lista de Figuras

Lista de Tablas

Lista de Abreviaciones y Símbolos

ROS: Robot Operating System

PCB: printed circuit board o plaqueta de circuito impreso

OpenCV: Biblioteca de visión artificial de código abierto

Dedicado a

Todas las personas que me apoyaron y ayudaron durante la realización de esta carrera y de toda mi vida académica en general. Especialmente a todos los representantes académicos que me inspiraron y ayudaron en el camino.

Capítulo 1: Introducción

En los tiempos actuales la producción de contenido audiovisual ha tenido un incremento exponencial. Por efecto de la popularización de redes sociales o plataformas web, como Youtube, Instagram, Netflix, Facebook, etc. Haciendo que los medios más tradicionales como la televisión, radio o diarios tengan que adaptarse, y de esta forma añadir transmisiones online en vivo, diarios digitales o contenidos web para no quedarse atrás.

Esto ha generado que se busquen nuevas formas de producir estos contenidos, facilitando su realización, optimizando sus costos, incentivando las producciones independientes o reemplazando trabajos repetitivos y poco especializados por sistemas automatizados.

Teniendo en cuenta esta idea de simplificar tareas repetitivasy poco especializadas, es que se realizó este proyecto. Como una opción de calidad y bajo coste para la función principal que realizaría un camarógrafo, es decir, mantener dentro de cuadro a una persona determinada.

Esto puede ser útil en contextos muy variados, como:

- Los contenidos de entretenimiento audiovisual que realiza un único individuo.
- Clases virtuales donde un profesor debe realizar anotaciones sobre un pizarrón.
- Reportajes periodísticos en entornos riesgosos o peligrosos.
- Documentales hechos en lugares remotos o de difícil acceso.

1.1 Público objetivo

Nuestro público objetivo está enfocado en las compañías de medios de comunicación, y los creadores de contenidos audiovisuales, con fines académicos, artísticos, periodísticos o de entretenimiento.

En un principio apuntaría a vender el producto en mercados nacionales, realizando múltiples mejoras según el feedback y necesidades de los clientes. Y una vez obtenido un producto de calidad superior, comenzar a

venderlo de forma internacional. Apuntando sobre todo a mercados enfocados en la producción de contenido para público joven, ya que estos últimos son los que más contenido audiovisual consumen.

1.2 Competencia

En el mercado nacional existen diferentes marcas competidoras, que ofrecen productos muy similares.

Figura 1 - trípode autónomo de Arioky

Estos están destinados al mercado de los celulares y a la producción de videos con fines recreativos. Algunas características que limitan su utilización para tareas más profesionales son su dependencia a utilizar el poder de procesamiento del celular para realizar el seguimiento, a la utilización de pilas descartables para la alimentación del dispositivo o su imposibilidad de adaptarlos a un trípode o base auxiliar.

1.3 Ideas iniciales

1.3.1 Método de seguimiento

En un principio al dispositivo se lo ideó para que realizara la función de seguimiento a partir del desfase de una señal inalámbrica recibida por dos antenas. Esta señal inalámbrica sería emitida por un dispositivo que la persona objetivo llevaría consigo. Aunque debido a la complejidad y rapidez que requeriría el hardware, se descartó la idea. Por lo que finalmente se determinó que la mejor opción sería la utilización de una cámara y el reconocimiento de imágenes.

1.3.2 Motor

Inicialmente se pensó en utilizar un servo para realizar el movimiento. Las ventajas que presentaba eran el control exacto de la posición, un torque elevado y simplicidad de uso. Pero se terminó descartando esta idea, por la limitación que estos presentan en su ángulo de rotación (usualmente 180°). Y se optó por utilizar un motor paso a paso, en su lugar.

Figura 2 - Servomotor MG946R

1.3.3 Sensor de posición inicial

En un principio se intentó utilizar un sensor de final de carrera o micro switch, para marcar la posición inicial a la cual debía volver el dispositivo cada vez que se encendiera. Pero se descartó, porque limitaba el ángulo de giro a 360° como máximo y tenía mayores posibilidades de dañarse con el tiempo, por el desgaste mecánico que se generaba al utilizarlo. Por lo que finalmente se decidio por utilizar un sensor de efecto Hall junto con un imán.

Figura 3 - Pulsador SS-5GL

Capítulo 2: Desarrollo

El sistema se puede resumir en dos diagramas de bloques diferentes. Por un lado, tenemos el diagrama relacionado con el Hardware, y por el otro el relacionado con el Software.

2.1 Hardware

A continuación, se describen las características de cada una de las partes del hardware utilizados en la etapa de pruebas del proyecto.

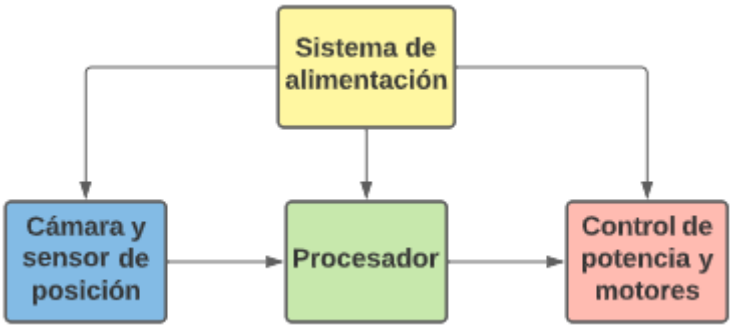

Figura 4 – Diagrama de bloques de hardware

2.1.1 Procesador

Este bloque es el encargado de realizar el control y configuración de todos los sensores y actuadores del dispositivo. Aunque su principal tarea es el procesamiento de las imágenes, y por lo tanto el reconocimiento y seguimiento de la persona objetivo.

Para realizar esta función se decidió utilizar una Raspberry Pi 4 de 8 GB de memoria RAM, que cuenta con un microprocesador ARM Cortex A72. La decisión de utilizar este hardware se fundamenta bajo el punto de que esta placa de desarrollo tiene un amplio soporte por parte de la comunidad open source, una razonable relación potencia/tamaño y un módulo de cámara pensado únicamente para estas placas de desarrollo, por lo que podemos garantizar su compatibilidad.

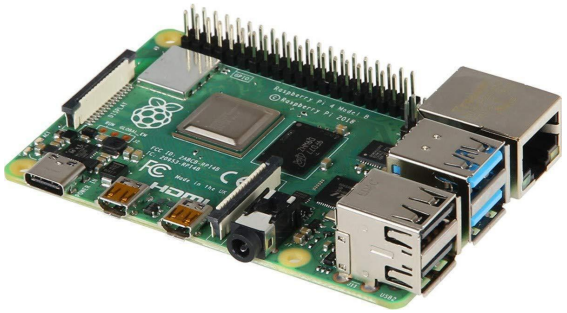

Figura 5 – Raspberry Pi 4

2.1.2 Cámara y sensor de posición

Este bloque se puede describir como el encargado de establecer la posición inicial y realizar la captura de las imágenes, que luego se utilizarán para guiar al dispositivo.

En este caso la cámara utilizada fue el módulo de cámara de Raspberry Pi V2.1, que cuenta con un sensor Sony IMX219 de 8 megapíxeles. Se eligió este módulo, ya que al estar pensado para integrarse con los sistemas Raspberry Pi, podría asegurar la compatibilidad desde un principio. Otra razón fue su tamaño reducido y buena calidad comparado con cámaras similares.

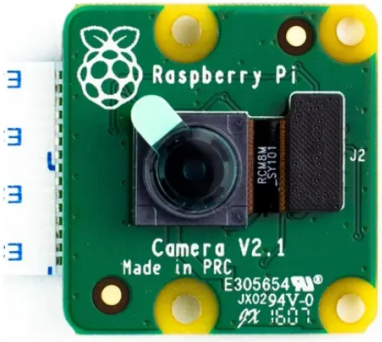

Figura 6 – Raspberry Pi cámara V2.1

Para cumplir con la función de sensor de posición se utilizó un sensor de efecto Hall KY-033 44E junto con un imán de neodimio. Se usaron estos elementos debido a que se necesitaba una forma de determinar la posición inicial del dispositivo, sin que esto generará un desgaste mecánico sobre el sensor o dificultará el diseño de los soportes internos. Por lo que se terminó llegando a esta solución, en donde el sensor nunca hará contacto directo con el imán.

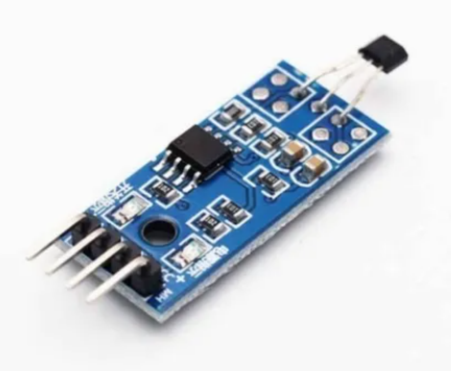

Figura 7 - sensor de efecto Hall

2.1.3 Control de potencia y motores

Este bloque está constituido por dos componentes principales, por un lado está el driver de motor paso a paso DRV8255, y por otro el motor paso a paso Nema 17.

Por un lado se decidió por usar en driver DRV8255, por su bajo costo, gran capacidad de realizar micropasos, tamaño reducido y fácil acceso. Mientras que el motor Nema 17 se eligió por sus características de torque, su posibilidad de realizar movimientos suaves y silenciosos y su fácil acceso al momento de comprarlo.

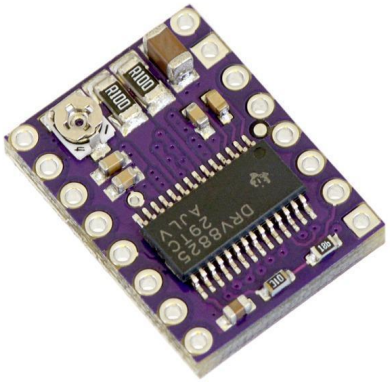

Figura 8 – Driver de motor DRV8255

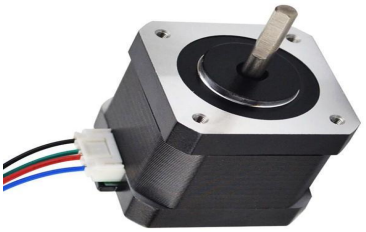

Figura 9 – Motor Nema 17

2.1.4 Sistema de alimentación

Para cumplir con la función de alimentación de todo el sistema se optó por utilizar una PCB diseñada específicamente para este proyecto. En la misma se tienen una fuente de alimentación de 5V-5A y otra de 12V-3A, además de un cargador de baterías de 4 celdas y con las correspondientes protecciones contra sobrecargas y sobredescargas.

Para las fuentes de 5V y 12V se utilizaron los integrados XL4005 y LM2596 correspondientemente, y para el cargador de baterías y la protección de las mismas se utilizaron los integrados DW01C, A2SHB, 06N03LA y HY2213-BB3A. Esta decisión se tomó debido al fácil acceso a los mismos, a su bajo coste y a su alta eficiencia.

Por otro lado las 4 baterías modelo 18650 utilizadas fueron de 3.7V y 2200 mA, puestas en una configuración en serie.

Este proyecto requiere una alimentación de 5V para alimentar toda la etapa de control. Esta incluye sensor de efecto Hall, cooler, cámara y Raspberry Pi. Mientras que la etapa de 12V se utiliza para alimentar el motor paso a paso.

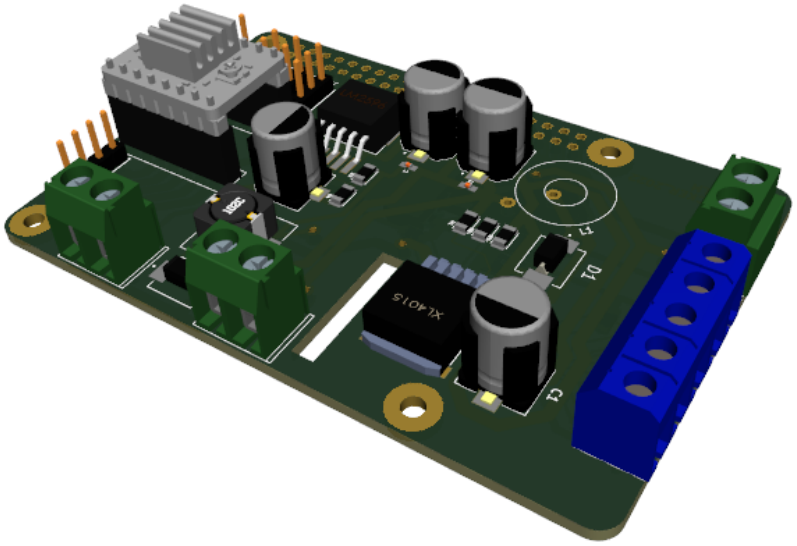

Figura 10 – PCB personalizada

Figura 11 - baterías 18650

2.2 Software

En lo referente al software el dispositivo se basa en el uso de ROS, instalado sobre un sistema operativo Ubuntu 16.04. Siendo ROS un meta sistema operativo de código abierto pensado para robots.

Las ventajas que presenta la utilización de ROS en mi proyecto son la estandarización de mensajes entre las diferentes partes del hardware, pasaje de mensajes entre procesos, manejo de paquetes y la posibilidad de usar y modificar paquetes ya existentes para adaptarlos a mis necesidades.

2.2.1 Conceptos necesarios

A continuación explicaré con detalle algunos conceptos básicos sobre ROS, necesarios para entender por qué el proyecto se realizó de esta manera y como funciona a nivel de software.

2.2.1.1 Robot Operating System (ROS)

ROS provee los servicios 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. El procesamiento toma lugar en los nodos que pueden recibir, mandar y multiplexar mensajes de sensores, control, estados, planificaciones y actuadores, entre otros.

ROS tiene dos partes básicas: la parte del sistema operativo, ros, como se ha descrito anteriormente y ros-pkg, una suite de paquetes aportados por la comunidad de usuarios que implementan diversas funcionalidades.

Nivel del sistema de archivos de ROS

Los conceptos de nivel de sistema de archivos cubren principalmente los recursos de ROS que encuentra en el disco, como:

- **Paquetes:** Los paquetes son la unidad principal para organizar el software en ROS. Un paquete puede contener procesos de tiempo de ejecución de ROS (nodos), una biblioteca dependiente de ROS, conjuntos de datos, archivos de configuración o cualquier otra cosa que se organice de manera útil en conjunto. Los paquetes son el elemento de compilación y el elemento de lanzamiento más atómico en ROS. Lo que significa que lo más granular que puede crear y lanzar es un paquete.

- **Metapaquetes:** Los metapaquetes son paquetes especializados que solo sirven para representar un grupo de otros paquetes relacionados. Por lo general, los metapaquetes se utilizan como un marcador de posición compatible con versiones anteriores de ROS.

- **Manifiestos de paquetes:** Los manifiestos (paquete.xml) proporcionan metadatos sobre un paquete, incluido su nombre, versión, descripción, información de licencia, dependencias y otra metainformación, como paquetes exportados.

- **Repositorios:** Una colección de paquetes que comparten la misma versión de ROS. Los paquetes que comparten una misma versión se pueden lanzar juntos. Los repositorios también pueden contener un solo paquete.

- **Tipos de mensajes (msg):** Las descripciones de mensajes definen las estructuras de datos para los mensajes enviados en ROS.

- **Tipos de servicios (srv):** Las descripciones de los servicios, definen las estructuras de datos de solicitud y respuesta para los servicios en ROS.

Nivel gráfico de cálculo de ROS

La computación a nivel gráfico es la red de ROS que se encarga de procesar todos los datos. Los conceptos básicos son nodos , maestro , mensajes y temas, los cuales proporcionan los datos de diferentes maneras:

- **Nodos:** Los nodos son procesos que realizan cálculos. ROS está diseñado para ser modular en una escala de grano fino; un sistema de control de robot generalmente comprende muchos nodos. Por ejemplo,

en este proyecto un nodo controla la cámara, otro nodo se encarga de controlar el motor paso a paso y un último nodo procesa las imágenes para detectar a las personas en ella.

- **Master:** El Nodo maestro o Master proporciona el registro de nombres y la búsqueda del resto del gráfico de cálculo. Sin el Maestro, los nodos no podrían encontrarse, intercambiar mensajes o invocar servicios.

- **Servidor de Parámetros:** El Servidor de Parámetros permite que los datos sean almacenados por clave en una ubicación central. Actualmente forma parte del Máster.

- **Mensajes:** Los nodos se comunican entre sí pasando mensajes . Un mensaje es simplemente una estructura de datos, que comprende campos escritos. Se admiten los tipos primitivos estándar (entero, punto flotante, booleano, etc.), al igual que las matrices de tipos primitivos. Los mensajes pueden incluir matrices y estructuras anidadas arbitrariamente (al igual que las estructuras C).

- **Temas:** Los mensajes se enrutan a través de un sistema de transporte de tipo publicación/suscripción. Un nodo envía un mensaje publicándolo en un tema determinado . El tema es un nombre que se utiliza para identificar el contenido del mensaje. Un nodo que esté interesado en cierto tipo de datos se suscribirán al tema correspondiente. Puede haber múltiples publicadores y suscriptores simultáneos para un solo tema, y un solo nodo puede publicar y/o suscribirse a múltiples temas. En general, los editores y los suscriptores no conocen la existencia de los demás. La idea es desvincular la producción de información de su consumo. Lógicamente, uno puede pensar en un tema como un bus de mensajes fuertemente tipificado. Cada bus tiene un nombre y cualquiera puede conectarse al bus para enviar o recibir mensajes, siempre que sean del tipo correcto.

- **Servicios:** El modelo de publicación/suscripción es un paradigma de comunicación muy flexible, pero su transporte unidireccional de muchos a muchos no es apropiado para las interacciones de solicitud/respuesta. La solicitud/respuesta se realiza a través de servicios, que están definidos por un par de estructuras de mensajes: una para la solicitud y otra para la respuesta. Un nodo proveedor ofrece un servicio bajo un nombre y un cliente utiliza el servicio enviando el mensaje de solicitud y esperando la respuesta.

- **Bolsas:** Las bolsas son un formato para guardar y reproducir datos de mensajes ROS. Las bolsas son un mecanismo importante para almacenar datos, como datos de sensores, que pueden ser difíciles de recopilar pero que son necesarios para desarrollar y probar algoritmos.

El ROS Master actúa como un servicio de nombres en el gráfico de cálculo de ROS. Almacena información de registro de temas y servicios para nodos ROS . Los nodos se comunican con el Maestro para informar su información de registro. A medida que estos nodos se comunican con el Maestro, pueden recibir información sobre otros nodos registrados y realizar conexiones según corresponda. El maestro también realizará devoluciones de llamada a estos nodos cuando cambie esta información de registro, lo que permite que los nodos creen conexiones dinámicamente a medida que se ejecutan nuevos nodos.

Los nodos se conectan a otros nodos directamente; el maestro solo proporciona información de búsqueda, como un servidor DNS. Los nodos que se suscriben a un tema solicitarán conexiones de los nodos que publican ese tema y establecen esa conexión a través de un protocolo de conexión acordado. El protocolo más común usado en un ROS se llama TCPROS , que usa sockets TCP/IP estándar.

Esta arquitectura permite la operación desacoplada, donde los nombres son el medio principal por el cual se pueden construir sistemas más grandes y complejos.

Por ejemplo, para controlar la cámara del dispositivo, iniciamos el controlador raspicam_node, que se comunica con el sensor y publica mensajes sensor_msgs/Image. Para procesar esos datos, usamos el nodo basics_camera que se suscriba a los mensajes sobre el tema de imagenes. Después de la suscripción, nuestro nodo automáticamente comenzaría a recibir mensajes de la cámara.

Todo lo que hace raspicam_node es publicar imágenes, sin saber si alguien está suscrito. Todo lo que hace el nodo procesador de imágenes es suscribirse a las imágenes, sin saber si alguien las está publicando. Los dos nodos se pueden iniciar, eliminar y reiniciar, en cualquier orden, sin inducir ninguna condición de error.

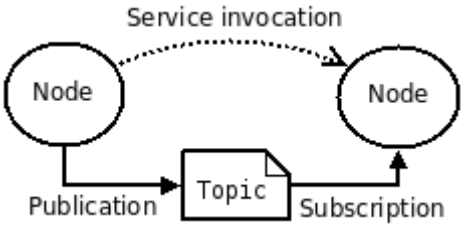

Figura 12 - Diagrama básico de funcionamiento de los nodos

2.2.1.2 Procesamiento digital de imagenes

Clasificador en cascada Haar

Los clasificadores en cascada basados en características de Haar son un método eficaz de detección de objetos. Este se basa en el aprendizaje automático en el que se entrena una función en cascada con muchas imágenes “positivas” y “negativas”. Siendo imágenes “positivas”, aquellas en donde está presente el objeto que se busca detectar; e imágenes “negativas”, aquellas en donde no se encuentra dicho objeto.

Figura 13 - Imágenes positivas de entrenamiento

Figura 14 - Imágenes negativas de entrenamiento

Un clasificador se entrena con muchas imágenes “positivas” y “negativas” de un objeto en particular que se esté buscando detectar, puede ser una cara, un automóvil o cualquier otro objeto. Por ejemplo, mi proyecto se basa en detectar una silueta humana, entonces se debe entrenar el clasificador con imágenes “positivas” y “negativas”, es decir, que contienen o no siluetas humanas respectivamente.

De estas imágenes de entrenamiento se extraen rasgos del objeto a detectar. Para esta tarea se utilizan las características de Haar. Cada característica es un valor individual obtenido restando la suma de píxeles bajo rectángulo blanco de la suma de píxeles bajo rectángulo negro.

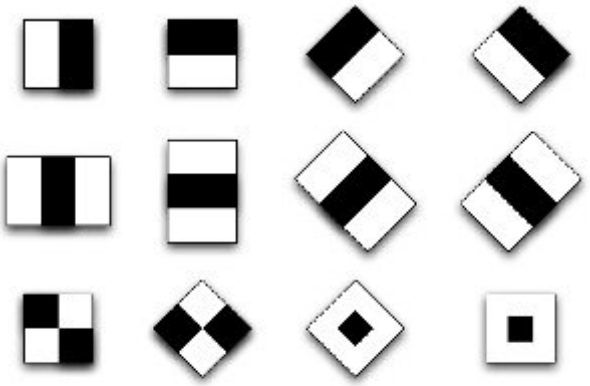

Figura 15 - Patrones de características de Haar

Aplicando todos los tamaños y ubicaciones posibles de cada característica, se calculan los rasgos del objeto deseado.

Pero entre todas estos rasgos calculamos que la mayoría de ellos son irrelevantes. Por ejemplo, como se puede ver en la imagen siguiente, la fila izquierda muestra dos buenos rasgos. El primero parece centrarse en la propiedad de que la región de los ojos es a menudo más oscura que la región de la nariz y las mejillas. La segunda característica seleccionada se basa en la propiedad de que los ojos son más oscuros que el puente de la nariz. Pero las mismas ventanas que se aplican en las mejillas o cualquier otro lugar es irrelevante.

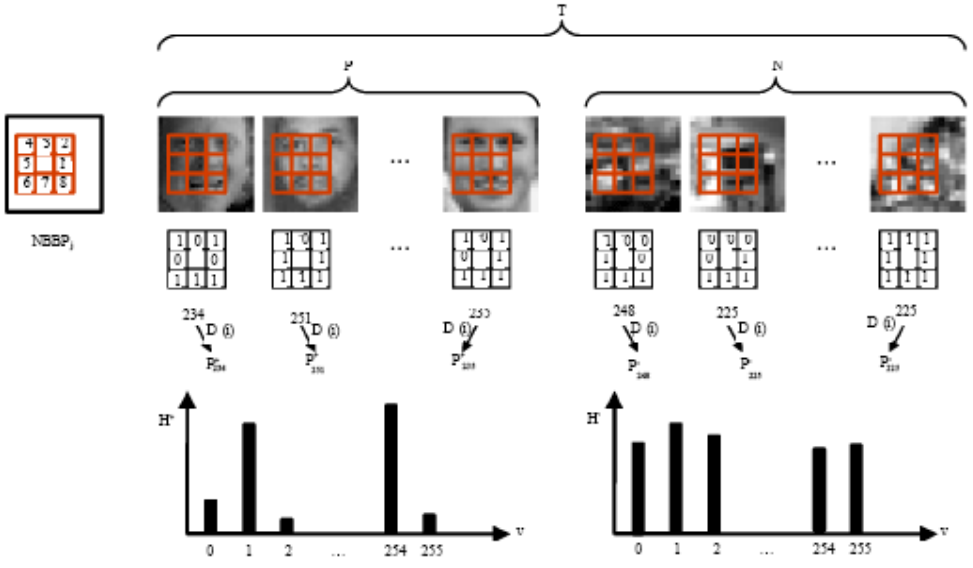

Figura 16 - Aplicación de características de Haar

Para seleccionar las mejores características de todo el conjunto, aplicamos todas y cada una de las características en todas las imágenes de entrenamiento. Usando cada característica, obtenemos el umbral que clasificará las imágenes en positivos y negativos. Pero obviamente, habrá errores o clasificaciones erróneas. De esta forma, seleccionamos las características con una tasa de error mínima, lo que significa que son las características que mejor clasifican las imágenes objetivo.

Finalmente, el clasificador es una suma ponderada de estos clasificadores más simples o débiles. Se llama débil porque solo no puede clasificar la imagen, pero junto con otros forma un fuerte clasificador. Técnicamente 200 características proporcionan una detección con un 95% de precisión.

Un problema que se puede generar luego de obtener este clasificador, es que el tiempo de procesamiento puede extenderse excesivamente, debido a que la mayor parte de una imagen no tiene al objeto buscado. Por lo tanto, se desarrolló un método simple para comprobar si una ventana no es una región buscada, desechandola y evitando que esa región se pase por el resto de clasificadores.

Para ello introdujeron el concepto de Clasificadores de cascada. En lugar de aplicar todas las características en una ventana, se agrupan las características en diferentes etapas de clasificadores y se aplican una por una. Si una ventana falla la primera etapa, se la desecha. Si pasa, se le aplica la segunda etapa de las características y se continúa con el proceso. La ventana que pasa por todas las etapas es una zona que contiene el objetivo buscado.

2.2.2 Funciones y librerías principales del software

El software principal se puede dividir en las siguientes librerías y programas representados en el diagrama de bloques.

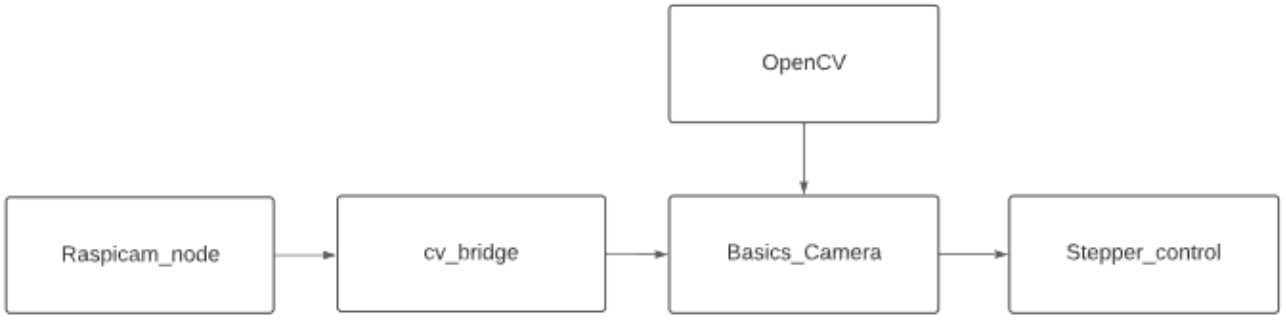

Figura 17 – Funciones y librerías principales del software

2.2.2.1 Raspicam_node

Las tareas principales de esta librería son controlar y configurar la cámara de la Raspberry Pi, además de poner a disposición del resto del programa las imágenes tomadas.

2.2.2.2 OpenCV

OpenCV es una biblioteca de software de visión artificial de código abierto, pensada para proporcionar una infraestructura común para las aplicaciones de visión por computadora.

2.2.2.3 Cv_bridge

Esta librería es la encargada de transformar las imágenes provenientes de la cámara en un formato que sea compatible con la librería OpenCV.

2.2.2.4 Basics_Camera

Este programa se encarga de tomar las imágenes provenientes de la cámara, luego de ser convertidas por la librería cv_bridge, para aplicar distintos filtros y transformaciones, por medio de la utilización de OpenCV, y de esta forma puedan utilizarse para que un clasificador de cascada Haar se encargue de diferenciar las personas que reconozca en cada imagen. Una vez que el programa reconoce una persona en las imágenes, el software

interpreta la posición de la misma y lo establece como coordenadas x-y. Estas coordenadas luego serán enviadas al programa final.

Este programa a su vez está compuesto de diferentes funciones:

- SubAndPub: Función encargada de iniciar el procesamiento de las imágenes obtenidas de la cámara, de enviar la posición establecida de la persona objetivo y de establecer los parámetros relacionados con la procedencia de las imágenes a procesar, clasificador de cascada a utilizar, tamaño máximo de la imagen, etc.

- imageCb: Esta función se encarga principalmente de recibir las imágenes del tópico y convertirlas a un formato compatible con OpenCV. Además, de asegurarse de la correcta carga del clasificador de cascada Haar.

- Detection: En esta función se realizan todas las transformaciones necesarias sobre la imagen original, como una conversión de RGB a escala de grises, un redimensionamiento con interpolación bilineal y una ecualización de histograma para mejorar su contraste y brillo. Luego de estos pasos, se la pasa por el clasificador de cascada Haar y se determinan las coordenadas de la persona objetivo en la imagen

2.2.2.5 Stepper_control

Este programa es el encargado de mantener a la persona siempre dentro del cuadro, mediante el control de un motor paso a paso. Esto se realiza comparando las coordenadas determinadas en el paso anterior, con unos límites que establecen si es necesario rotar el dispositivo hacia la derecha, izquierda o mantenerlo estático.

Este programa está compuesto por las siguientes funciones:

- ZeroPos: Esta función se encarga de realizar una rotación continua del dispositivo hasta que se encuentra en la “posición inicial”. Esto se realiza por medio de un sensor de efecto Hall, que se activa al

encontrarse con un imán situado en la base. Con esta simple función nos aseguramos de que sin importar la posición en la que se desconecte el dispositivo, siempre volverá a la posición inicial cuando se lo vuelva a utilizar.

- track_face: Esta función se encarga de determinar si es necesario rotar el dispositivo, tomando el valor obtenido por la función basics_camera, y comparándolo con diferentes límites, que generan diferentes movimiento con diferentes velocidades.

La lógica seguida por el programa, a grandes rasgos, se puede representar mediante el siguiente diagrama de flujo.

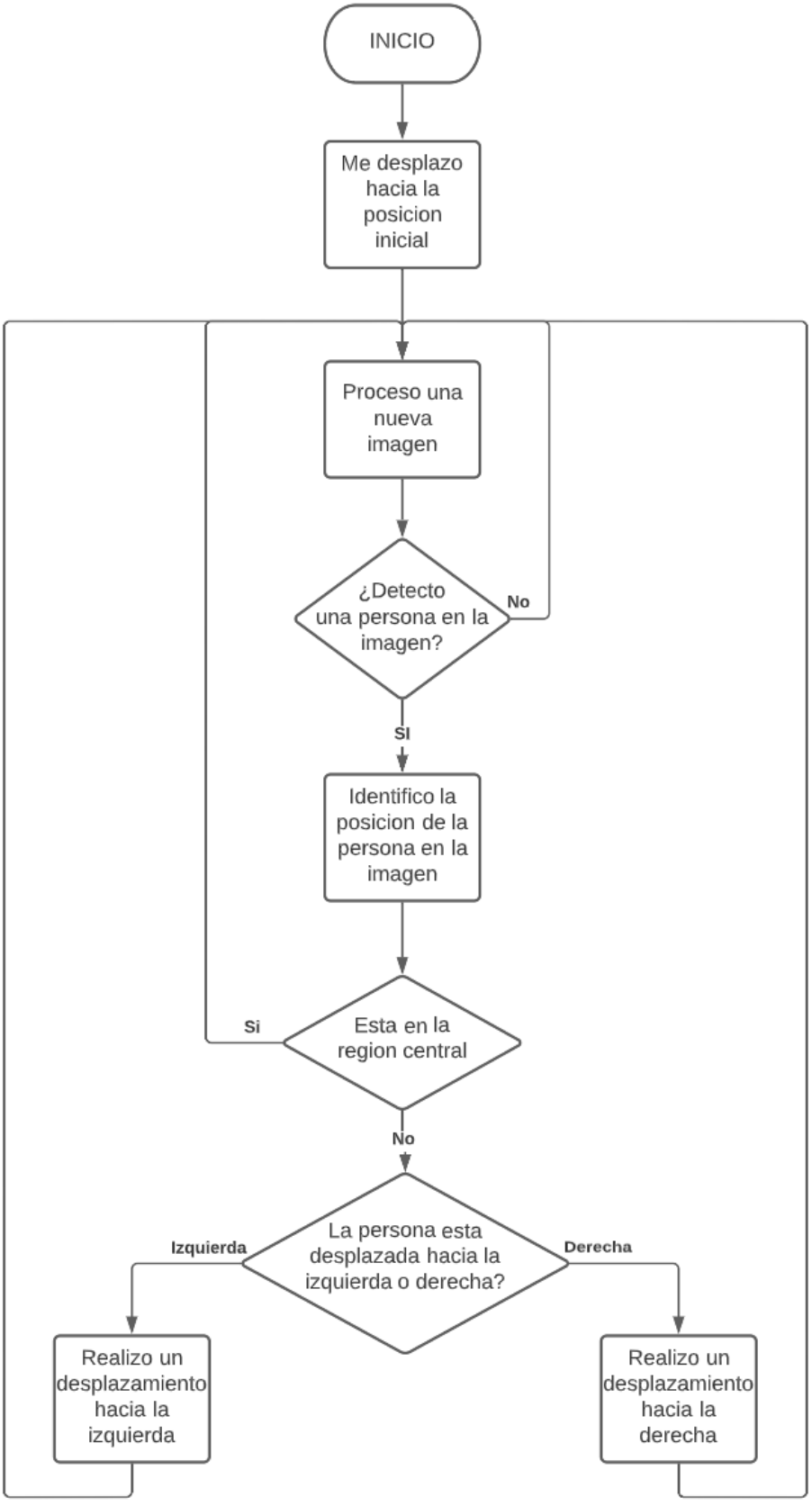

Figura 18 - Diagrama de flujo del programa principal

2.3 Diseño completo

El diseño final del sistema consiste de un cilindro superior, donde se encuentran todos los sistemas principales, y un cilindro que se usa como base.

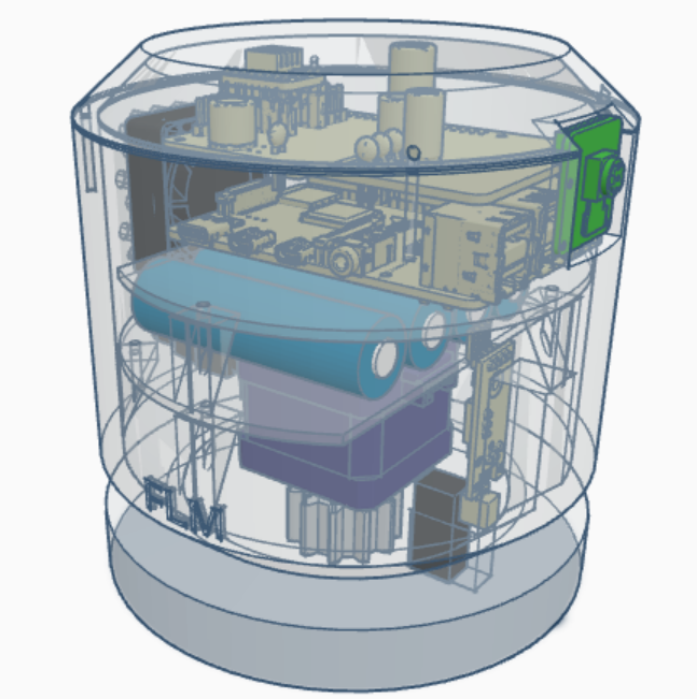

Figura 19 - Diseño 3D de la carcasa del dispositivo

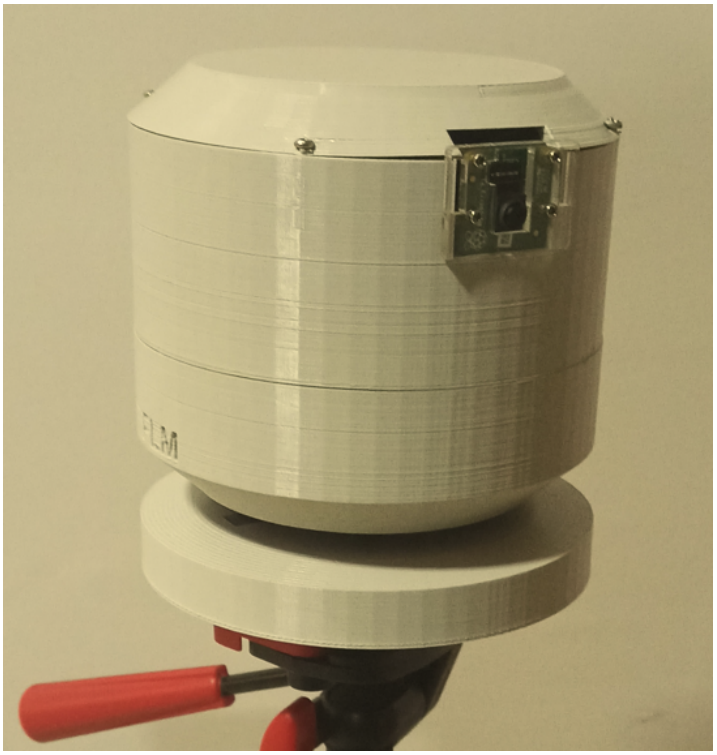

Figura 20 - Dispositivo final

Dentro del cilindro principal se pueden hallar todos los sistemas necesarios para el funcionamiento del robot, divididos en 3 niveles. En el nivel superior se pueden encontrar el sistema de procesamiento, las fuentes de alimentación, la cámara, y el cargador de baterías. Luego en el nivel medio se puede encontrar el arreglo de baterías. En el último nivel, se encuentra el motor, el conector de alimentación y el sensor de efecto Hall. Finalmente en la base, solo se puede hallar el imán que será captado por el sensor de efecto Hall. Además, se puede destacar que entre los primeros dos niveles, se encuentra un cooler de 40 mm que se encarga de refrigerar los componentes internos.

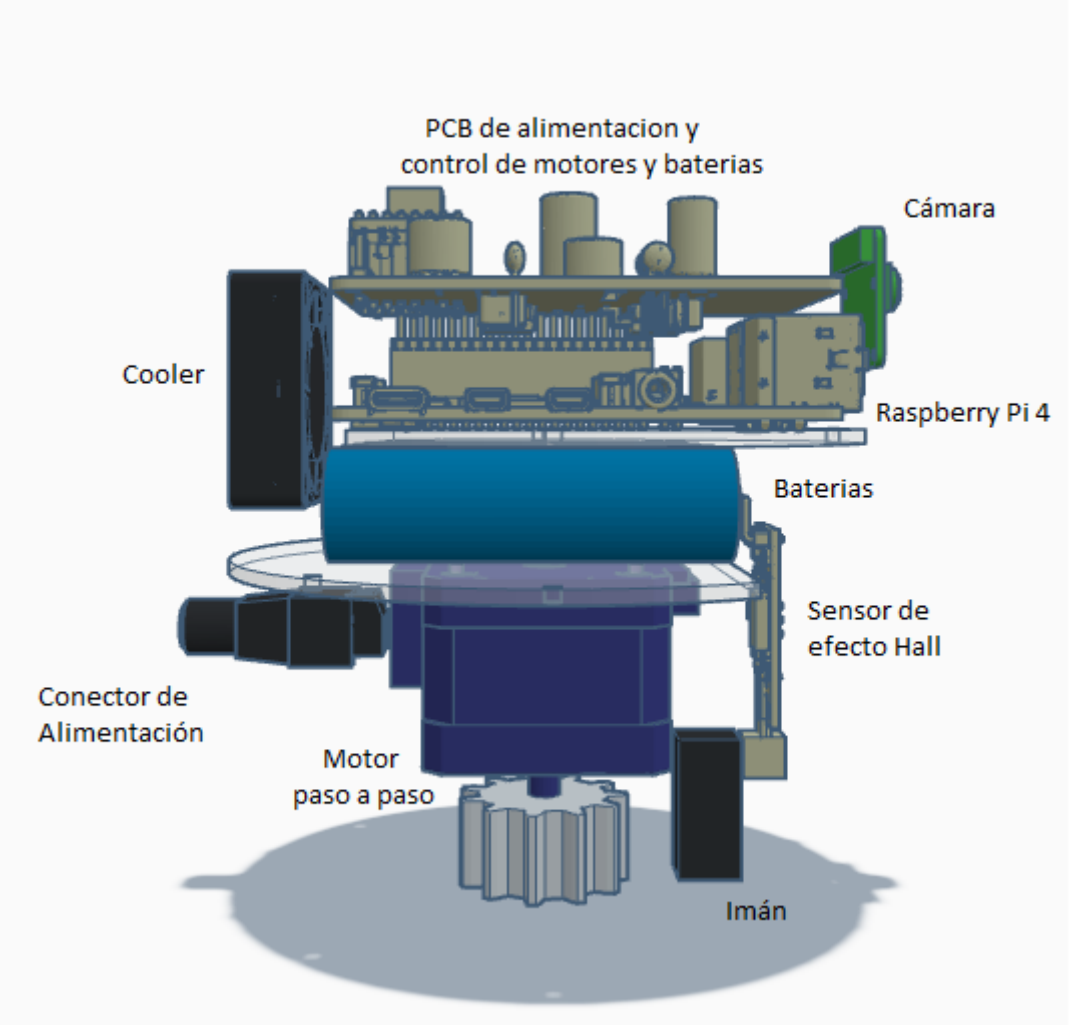

Figura 21 - Diagrama de los componentes del dispositivo

La interfaz de usuario consiste en simplemente un pulsador que se utiliza para iniciar o detener la función de seguimiento.

Y el procedimiento para utilizarlo sería el siguiente:

1. Colocar el soporte para el trípode correspondiente en la cara inferior o ,en caso de no usar un trípode, colocarlo sobre una superficie estable.
2. Acoplar la cámara, celular, o dispositivo de grabación en la cara superior.
3. Presionar el botón para que comience en seguimiento autónomo.
4. Una vez finalizada la grabación presionar nuevamente el botón para detenerlo.

2.4 Pruebas realizadas

2.4.1 Pruebas iniciales de funcionamiento de hardware y software

En un principio, se efectuaron pruebas básicas sobre el hardware. Estas se realizaron con el objetivo de elegir el hardware más apropiado para las tareas que buscábamos desarrollar.

2.4.2 Pruebas del software de reconocimiento

Desde el comienzo del desarrollo del sistema de reconocimiento de personas, se realizaron múltiples pruebas con el fin de analizar el desempeño de cada función que compone el programa. Se procedió de esta manera con el objetivo de que si en algún momento surgían problemas, se podrían limitar sus posibles causas.

Por medio de estas pruebas, se logró mejorar la capacidad de detección del software y se detectó el problema de la baja tasa de procesamiento de imágenes que tiene el dispositivo.

2.4.3 Pruebas del software de control de motores

Una vez finalizado el programa de reconocimiento, se comenzó a desarrollar el software encargado de traducir las posiciones x-y a movimientos del motor paso a paso. Durante esta etapa, se realizaron pruebas iniciales básicas para evaluar el control del motor por medio de la placa de desarrollo. Y en etapas posteriores también se probó y calibró el funcionamiento del programa final de control de motor.

2.4.4 Pruebas de campo

Luego de tener el software totalmente funcional, se realizaron pruebas de campo en diferentes entornos con variedad de condiciones ambientales (luz, distancia entre el dispositivo y la persona). Con estas pruebas se logró calibrar la velocidad de movimiento del motor, y se detectó el problema causado por la baja tasa de procesamiento de imágenes. Este inconveniente se origina de la insuficiente capacidad de procesamiento que tiene la placa de desarrollo elegida.

2.4.5 Pruebas con hardware adecuado

Buscando mejorar la tasa de procesamiento de imágenes, se realizó la siguiente prueba.

Ya que la limitación principal es la capacidad de procesamiento del hardware, se optó por conectar el dispositivo, por medio de una red LAN cableada, a una computadora con mejores capacidades de procesamiento. Con el objetivo de que la computadora se encargue del análisis de las imágenes, mientras que el dispositivo solo tendría que encargarse de la obtención y transferencia de las imágenes y del movimiento.

Los resultados obtenidos son una mejora significativa de la tasa de procesamiento de imágenes, pasando de 1 frame por segundo a casi 3 frames. Lo que a su vez implicó un incremento en la precisión del dispositivo a la hora de seguir a la persona objetivo.

Por otro lado, debo aclarar que esta prueba también encuentra algunas limitaciones, relacionadas con la latencia o retraso de los datos y la limitada tasa de transferencia entre los equipos. La primera limitación implica que al tener que transferir los datos entre equipos, se generaría un retraso en la respuesta de los mismos, que podría ser significativa para un dispositivo que requiere seguir a una persona en todo momento. Mientras que la segunda limitación se relaciona con la limitada tasa de transferencia de datos 12Mb/s que se puede tener entre los equipos. Esto genera que aunque releguemos el procesamiento de las imágenes a un equipo con mejores prestaciones, no se pueda incrementar la tasa de procesamiento de las imágenes aún más, debido a que la velocidad de transferencia de los datos es insuficiente.

En definitiva con esta prueba, demostramos que el problema de la tasa de procesamiento se puede resolver mejorando el hardware del propio dispositivo.

Capítulo 3: Resultados

Finalmente tras la realización de este proyecto se obtuvo un producto final capaz de reconocer a una persona en una imagen y seguirla para mantenerla siempre en el centro de la imagen. Por lo que se podría decir que se cumplió con el objetivo buscado.

Aunque existen algunos puntos a mejorar y resolver, como por ejemplo, la tasa de imágenes procesadas por segundo, o la imposibilidad de seguir a la persona objetivo cuando múltiples personas están en el plano.

El primer punto es relevante debido a que el dispositivo cuenta con una tasa de procesamiento de imágenes de aproximadamente 1 frame por segundo. Esto genera que tanto la precisión de seguimiento como la velocidad de reacción del dispositivo se vean afectadas de forma negativa. Sencillamente, al tener una tasa de procesamiento tan baja, el dispositivo puede llegar a perderse en caso de que la persona se mueva muy rápido a su alrededor.

Otro gran inconveniente es la imposibilidad de seguir a una persona determinada, al encontrarse múltiples sujetos en las imágenes.

Una posible solución para este inconveniente sería la utilización del template matching o búsqueda de patrones, como segundo método de análisis de imágenes. El template matching se basa en la búsqueda de un patrón o “etiqueta” distintiva dentro de una imagen de mayor tamaño.

En definitiva, mediante la combinación del template matching y los clasificadores de cascada haar, es posible crear un algoritmo que siga a una persona en particular dentro de un grupo de personas.

Capítulo 4: Análisis de Costos

El costo que se describe a continuación implica la fabricación de equipos con los mismos componentes utilizados en las pruebas.

En el caso de realizar un diseño final para su producción a gran escala podrían reemplazarse algunos componentes por otros con mejores prestaciones y placas más pequeñas que integren la totalidad del dispositivo. Por obvias razones estos factores podrían afectar el coste final de producción.

Para la implementación del dispositivo se suponen los siguientes costos fijos:

Insumos - Costos Fijos	Unidades	Precio unitario en pesos	Subtotal en pesos	Subtotal en dólares
Raspberry Pi 4 de 8GB	1	$79,977.00	$79,977.00	$551.57
Tarjeta SD 32GB	1	$1,020.00	$1,020.00	$7.03
Disipadores para Raspberry Pi	1	$1,800.00	$1,800.00	$12.41
Cooler 40 mm	1	$560.00	$560.00	$3.86
Módulo LM2596	1	$520.00	$520.00	$3.59
Módulo XL4005	1	$675.00	$675.00	$4.66
Cargador y protección de baterías 4S	1	$2,103.00	$2,103.00	$14.50
Baterias 18650 2200 mAh	4	$720.00	$2,880.00	$19.86
Carcasa impresa en 3D	1	$2,148.00	$2,148.00	$14.81
Motor Nema 17	1	$3,200.00	$3,200.00	$22.07
Sensor de efecto Hall	1	$300.00	$300.00	$2.07
Imán de Neodimio	2	$353.00	$706.00	$4.87
Fuente de alimentación de 16V	1	$2,300.00	$2,300.00	$15.86
Cámara para Raspberry Pi V2.1	1	$8,093.00	$8,093.00	$55.81
Acrilico de protección para cámara	1	$1,540.00	$1,540.00	$10.62
Cable USB-C	1	$279.00	$279.00	$1.92

Pines	2	$206.00	$412.00	$2.84
Bornera	5	$209.00	$1,045.00	$7.21
Bornera de alimentación	1	$385.00	$385.00	$2.66
Tornillos M3	15	$8.59	$128.85	$0.89
Tuerca de inserción	4	$50.25	$201.00	$1.39
PCB de diseño	1	$614.00	$614.00	$4.23
		$107,060.84	$110,886.85	$764.74

Tabla 1 - Costos fijos

En las últimas dos columnas se muestran los precios en dólares al valor de cotización oficial del día 31/8/2022, que es de $145.

Por otro lado, los costos variables se detallan a continuación:

Servicios - Costos Variables	Horas	Precio unitario en pesos	Precio total en pesos	Costo total en dólares
Desarrollo de Software	400	$500.00	$200,000.00	$1,379.31
Diseño de carcasa	200	$500.00	$100,000.00	$689.66
Diseño de plaquetas	200	$500.00	$100,000.00	$689.66
			$400,000.00	$2,758.62

Tabla 2 - Costos variables

Se realizaron 800 horas de trabajo en total, que incluyen las horas de planificación, diseño y pruebas de cada módulo o parte del dispositivo. El precio de la hora se fijó en $500 pesos argentinos.

Resumiendo, los costos fijos y variables comprenden una inversión inicial total de U$S 3524.

Teniendo en cuenta estos factores, se propuso un precio de venta de US$1000. Con este valor, se necesitaría de al menos 12 ventas para generar ganancias y reintegrar la inversión inicial.

Cantidad vendida	Costo fijo	Venta	Margen
0	$0.00	$0.00	-$2,758.62
1	$764.74	$1,000.00	-$2,523.36
2	$764.74	$2,000.00	-$2,288.10
3	$764.74	$3,000.00	-$2,052.84
4	$764.74	$4,000.00	-$1,817.58
5	$764.74	$5,000.00	-$1,582.32
6	$764.74	$6,000.00	-$1,347.06
7	$764.74	$7,000.00	-$1,111.80
8	$764.74	$8,000.00	-$876.54
9	$764.74	$9,000.00	-$641.28
10	$764.74	$10,000.00	-$406.02
11	$764.74	$11,000.00	-$170.76
12	$764.74	$12,000.00	$64.50

Tabla 3 - Análisis de amortización

Ante el incremento continuo de los costos fijos y variables en pesos, se optó por establecer el precio de venta en dólares. De esta manera, se puede simplificar la tarea de actualizar el precio de venta en pesos.

Capítulo 5: Discusión y Conclusión.

5.1 Discusión

Este proyecto inicialmente se pensó con el fin de diseñar un producto, que sea capaz de realizar el seguimiento de una persona determinada. Y por lo tanto, pensado para su uso en la realización de contenidos audiovisuales, desde entrevistas o documentales, hasta vídeos de entretenimiento.

Finalmente se obtuvo un dispositivo capaz de reconocer personas, y seguirlas mediante la utilización de procesamiento de imágenes y clasificadores Haar en cascada. Además de poseer ciertas características atractivas, como:

- La utilización de baterías recargables que brindan una autonomía de varias horas.
- Una construcción basada en componentes de fácil acceso, y en su mayoría de bajo costo.
- Un programa modular capaz de facilitar la actualización y mejora del software y hardware.
- Una interfaz de usuario muy simple.
- La posibilidad de utilizarse con una infinidad de dispositivos de grabación.

Aunque también con ciertas deficiencias, como su movimiento poco fluido o su falta de capacidad de seguir a una persona determinada.

En comparación con productos ya existentes en el mercado, se consiguió tener ciertas características diferenciadoras:

- Posibilidad de utilizarlo con celulares o cámaras: los productos de la competencia solo apuntan a la rama de los celulares, ya que necesitan una app para funcionar. Mientras que nuestro producto funciona de forma independiente al objeto que se le adicione.
- Utilización de baterías recargables : ciertos modelos de la competencia aún funcionan con pilas AAA. Mientras que nuestro dispositivo funciona con baterías recargables de mayor autonomía. Esta característica le

permitirá a los clientes utilizar el producto por más tiempo y sin tener que comprar baterías desechables.

- Método de utilización sencillo y rápido: el proceso que se debe realizar para utilizarlo consiste en acoplarlo con la cámara o celular, y presionar un botón para comenzar el proceso de seguimiento. Esto es mucho más simple a comparación de los productos de la competencia, que necesitan de una aplicación y múltiples configuraciones.
- Capacidad de adaptarse a trípodes genéricos: la totalidad de los productos competidores a escala nacional sólo pueden utilizarse de forma independiente a cualquier trípode o base. Mientras que nuestro producto es capaz de adicionarse a cualquier trípode genérico o base de soporte.

Figura 22 - Dispositivo acoplado a un trípode

Aunque también se tienen ciertas debilidades, con respecto a la competencia, como su elevado costo (U$S1000), su movimiento poco fluido y su gran tamaño.

5.2 Conclusión

Aunque se hayan encontrado múltiples problemas, el aporte más significativo e importante que se puede obtener de este proyecto, es que establece un nuevo enfoque, al estar pensado para que se utilice en un mercado más profesional y variado que sus competidores. Al ser un dispositivo que trabaja de forma individual al resto de complementos que se le añaden (cámaras profesionales, celulares o trípodes), es capaz de realizar su tarea independientemente del ámbito para el que se lo utilice.

Además, al estar realizado con ROS, es fácil de reproducir y mejorar. Debido a que su funcionamiento interno, a nivel de software, es modular. Por lo que se pueden reemplazar o adicionar partes de hardware, sin que esto genere grandes cambios en el programa principal. Además, esta capacidad modular también permite la posibilidad de agregar nuevas funciones con relativa facilidad.

5.3 Posibles mejoras a implementar

A continuación se expondrán las posibles mejoras futuras, tanto de hardware como de software que se podrían realizar sobre el dispositivo.

5.3.1 Control de forma remota desde un celular o pulsera

Esta mejora consiste en la implementación de un control o una aplicación para celular o pulsera inteligente, la cual permita su control de forma remota. Su funcionamiento se basaría seguramente en la utilización de la tecnología bluetooth.

5.3.2 Mejora de autonomía

Esta mejora consiste en el rediseño de la posición del paquete de baterías dentro de la estructura. Con el objetivo de dejarlas en una posición más favorable para su reemplazo en caso de agotarse. De esta forma podríamos obtener una mejora sustancial de la autonomía del producto por medio de una redistribución de sus componentes internos. Una de las posibles desventajas de esta opción es el inminente apagado del dispositivo para realizar el cambio de baterías.

Otra posibilidad sería la adición de un soporte para baterías auxiliares, sin cambiar la distribución interna de los componentes. De esta forma se podrían agregar múltiples paquetes de baterías, incrementando la autonomía, sin generar una interrupción en el funcionamiento del sistema.

5.3.3 Utilización de múltiples métodos de seguimiento

El dispositivo presenta un problema en determinar la persona objetivo si en las imágenes procesadas se encuentran varias personas.

Es por esto que se propone la utilización de múltiples métodos para la determinación de la persona objetivo.

Entre estos métodos están:

- Etiqueta de reconocimiento: Utilización de una etiqueta que establezca a la persona objetivo, frente a todas las demás.
- Dispositivo emisor: Utilización de un dispositivo que lleve la persona objetivo, y que mediante la emisión de una señal, pueda ayudar en la ubicación de la persona.
- Reconocimiento facial: Realizar el reconocimiento facial de la persona objetivo para diferenciarla entre todas las demás. Se debe indicar que este método presenta serios problemas, con respecto a la complejidad del sistema y también a las especificaciones de hardware, por lo que sería muy difícil de implementar.

Capítulo 6: Literatura Citada.

[1] «ROS,» Disponible: https://www.ros.org/ [Último acceso: 26 9 2022].

[2] «ROS,» Disponible: https://wiki.ros.org/ [Último acceso: 26 9 2022].

[3] «OpenCV,» Disponible: https://opencv.org/ [Último acceso: 26 9 2022].

[4] «Clasificadores Haar en cascada,» Disponible: https://docs.opencv.org/3.4/db/d28/tutorial_cascade_classifier.html [Último acceso: 26 9 2022].

[5] «Clasificadores Haar en cascada,» Disponible: https://unipython.com/deteccion-rostros-caras-ojos-haar-cascad/#:~:text=La%20detecci%C3%B3n%20de%20objetos%20mediante,of%20Simple%20Features%E2%80%9D%20en%202001. [Último acceso: 26 9 2022].

[6] «OpenCV,» Disponible: https://opencv.org/ [Último acceso: 26 9 2022].

[7] Lentin Joseph, ROS Robotics Projects, 2017.

[8] Lentin Joseph, Robot Operating System (ROS) for Absolute Beginners, 2018.