

**UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE QUITO**

**CARRERA:
INGENIERÍA ELECTRÓNICA**

**Trabajo de titulación previo a la obtención del título de
INGENIEROS ELECTRÓNICOS**

**TEMA:
DESARROLLO DE UN PROTOTIPO ELECTRÓNICO QUE GUÍE A
PERSONAS INVIDENTES EN LA RECOLECCIÓN DE OBJETOS
MEDIANTE AUDIO Y VISIÓN ARTIFICIAL**

**AUTORES:
CRISTHIAN ANDRÉS GUTIÉRREZ GAVILÁNEZ
HEDIAN XAVIER HERRERA ORBEA**

**TUTOR:
JOSÉ ALCIDES RUMIPAMBA LÓPEZ**

Quito, septiembre del 2019

CESIÓN DE DERECHOS DE AUTOR

Yo Cristhian Andrés Gutiérrez Gavilánez, con documento de identificación N° 210039805-2 y Hedian Xavier Herrera Orbea, con documento de identificación N° 050273035-1, manifestamos nuestra voluntad y cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos los autores del trabajo de titulación intitulado: “DESARROLLO DE UN PROTOTIPO ELECTRÓNICO QUE GUÍE A PERSONAS INVIDENTES EN LA RECOLECCIÓN DE OBJETOS MEDIANTE AUDIO Y VISIÓN ARTIFICIAL”, mismo que ha sido desarrollado para optar por el título de Ingenieros Electrónicos, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En aplicación a lo determinado en la Ley de Propiedad Intelectual, en nuestra condición de autores nos reservamos los derechos morales de la obra antes citada. En concordancia, se suscribe este documento en el momento de hacer la entrega del trabajo final en formato impreso y digital a la Biblioteca de la Universidad Politécnica Salesiana.

Cristhian Andrés Gutiérrez Gavilánez
C.I. 210039805-2

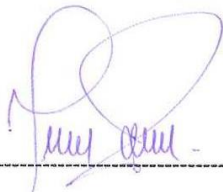
Hedian Xavier Herrera Orbea
C.I 050273035-1

Quito, septiembre del 2019

DECLARATORIA DE COAUTORÍA DEL DOCENTE TUTOR

Yo declaro que bajo mi dirección y asesoría fue desarrollado el Proyecto Técnico, “DESARROLLO DE UN PROTOTIPO ELECTRÓNICO QUE GUÍE A PERSONAS INVIDENTES EN LA RECOLECCIÓN DE OBJETOS MEDIANTE AUDIO Y VISIÓN ARTIFICIAL”, realizado por Cristhian Andrés Gutiérrez Gavilánez y Hedian Xavier Herrera Orbea, obteniendo un producto que cumple con todos los requisitos estipulados por la Universidad Politécnica Salesiana para ser considerado como trabajo final de titulación.

Quito, septiembre del 2019.



José Alcides Rumipamba López

C.I. 180385435-3

DEDICATORIA

Este proyecto está dedicado para toda mi familia y como una muestra de cariño que se puede tener hacia mis padres Abdón Herrera y Olga Orbea personas nobles y sencillas, las cuales me ayudaron a que todo esto se pueda cumplir de la mejor manera y por su incondicional apoyo en todo momento para la culminación de mi carrera. También quiero dedicar este trabajo a la familia **SALESIANA**, para todos ellos tengo una enorme gratitud en el corazón que me impulsa a seguir adelante; también dedico este trabajo a todos los que me conocen como una muestra que con perseverancia se alcanza los objetivos y las cosas buenas no son fáciles y no llegan con brevedad.

Hedian

Este proyecto está dedicado para toda mi familia y como una muestra de cariño que se puede tener hacia a mi madre Abg. Ana María Gavilánez persona noble y sencilla, la cual me ayudó a que todo esto se pueda cumplir de la mejor manera y por su incondicional apoyo en todo momento para la culminación de mi carrera; también quiero dedicar este trabajo a la familia **SALESIANA**, para todos ellos tengo una enorme gratitud en el corazón que me impulsa a seguir adelante; además, dedico este trabajo a todos los que me conocen como una muestra que con perseverancia se alcanza los objetivos y las cosas buenas no son fáciles y no llegan con brevedad.

Cristhian

AGRADECIMIENTOS

Agradezco, primero, a Dios por permitirme llegar a este día y a mis padres Abdón Herrera y Olga Orbea que mediante su sacrificio tanto económico como espiritual han sabido encaminarme a lo largo de mi vida; a mi esposa Alexandra Guachamin, madre de mis hijos que con su paciencia supo muchas veces entender las dificultades que se generan al momento de cursar una carrera universitaria; a mis hijos Hedian Ariel y Divni Daniela por ser los motores que me impulsaron a seguir adelante con mis metas.

Un sincero agradecimiento a mi tutor, el Ingeniero José Rumipamba M.Sc. por la colaboración dada en el desarrollo de este proyecto, a mis compañeros de la Universidad con quienes fue más agradable adquirir, recibir y compartir el conocimiento en las aulas.

Por último, y no menos importante, a todos los miembros de mi familia que nunca dejaron que me retire de esta meta, por hoy, cumplida; a mis hermanos y demás familiares.

Hedian

Ante todo, me gustaría expresar mi sincero agradecimiento a mi tutor, el Ingeniero José Rumipamba MSc. por el continuo apoyo de mi estudio e investigación en el proyecto de tesis, por su paciencia, motivación, entusiasmo y gran conocimiento. Su orientación me ayudó en todo el tiempo de investigación y creación de este proyecto.

Agradezco a mis amigos de Universidad por las estimulantes discusiones, por las noches de insomnio en las que trabajamos juntos y por todos los aprendizajes mutuos conseguidos. Además, las grandes amistades que han iluminado y hecho más llevadera mi vida en todo momento, como un faro atravesando cualquier tormenta.

Por último y más importante, me gustaría agradecer a mi familia: a mi madre Abogada Ana María Gavilánez por darme a luz, en primer lugar, y apoyarme espiritualmente a lo largo de mi vida; y a mi hermano Ingeniero Luis Fernando Gutiérrez, por su consejo constante.

Cristhian

ÍNDICE DE CONTENIDO

CESIÓN DE DERECHOS DE AUTOR.....	i
DECLARATORIA DE COAUTORÍA DEL DOCENTE TUTOR	ii
DEDICATORIA.....	iii
AGRADECIMIENTOS	iv
ÍNDICE DE CONTENIDO.....	vi
RESUMEN.....	xi
ABSTRACT	xii
INTRODUCCIÓN.....	1
CAPÍTULO 1	1
ANTECEDENTES	1
1.3 Tema	1
1.4 Planteamiento del Problema.....	1
1.5 Justificación.....	1
1.6 Objetivo General.....	2
1.7 Objetivos Específicos.....	2
CAPÍTULO 2	4
MARCO TEÓRICO	4
2.1 Estado del arte	4
2.2 Sensor de distancia	6
2.3 Microcomputadora.....	6
2.4 Visión artificial.....	7
2.5 Búsqueda de objetos utilizando visión artificial.....	7
2.6 Raspberry PI.....	8

2.6.1	Características	9
2.6.2	Periféricos	9
2.6.3	Lenguajes de programación que se pueden usar para programar	11
2.7	Python	12
2.7.1	Definición.....	12
2.7.2	Características	12
2.7.3	Programación básica con el lenguaje Python.....	12
2.7.4	Como utilizarlo en Raspberry Pi	13
2.7.5	Generación de sonidos en Python.....	13
2.8	OpenCV	15
2.8.1	Definición.....	15
2.8.2	Como instalar OpenCV en Raspberry Pi	15
2.8.3	Como utilizar Python en Raspberry PI	16
2.8.4	Formato RGB	17
2.8.5	Formato HSV	18
2.8.6	Máscara	19
2.8.7	Momentos.....	20
2.8.8	Ejemplos de programación en Raspberry	20
2.9	Cámara WEB.....	22
2.9.1	Definición.....	22
2.9.2	Características	22
2.9.3	Resolución.....	23
2.9.4	Resolución mínima para utilizarla en visión por computadora.....	23
2.9.5	Cámara Web Havit Hvn5082 640x480 16mpx	23
2.10	Impresión 3D.....	23
2.10.1	Definición.....	23
2.10.2	Características	24

2.10.3	Como modelar piezas en inventor	24
2.10.4	Como imprimir piezas modeladas con inventor.....	25
2.11	Triangulación con dos cámaras	26
2.11.1	Definición de técnica	26
2.11.2	Regresión polinómica en Excel	27
CAPÍTULO 3	28
DISEÑO E IMPLEMENTACIÓN DEL SISTEMA	28
3.1	Arquitectura del sistema.....	29
3.2	Estructura del sistema	30
3.3	Algoritmo creado	32
3.4	Cálculo del ángulo de desviación	35
3.5	Ecuación para calcular la distancia entre usuario y objeto	36
CAPÍTULO 4	39
PRUEBAS Y RESULTADOS	39
4.1	Pruebas de corrección del ángulo de desviación	39
4.2	Pruebas de cálculo de la distancia hasta el objeto	42
4.3	Pruebas en la recolección del objeto.....	44
4.4	Resumen de los resultados obtenidos	45
CONCLUSIONES	48
RECOMENDACIONES	50
BIBLIOGRAFÍA	51

ÍNDICE DE FIGURAS

Figura 2.1. Sistema propuesto por Minh Nguyen, Huy Le, Wei Qi Yan, Arpita Dawda -----	5
Figura 2.2. Esquema Raspberry Pi. -----	8
Figura 2.3. Ejemplo de la reproducción de un sonido -----	14
Figura 2.4. Interfaz de usuario -----	14
Figura 2.5. Comandos para instalar OpenCV -----	16
Figura 2.6. Comando para instalar paquetes y librerías -----	16
Figura 2.7. Comando para instalar libopencv -----	16
Figura 2.8. Comando mkvirtualenv -----	17
Figura 2.9. Entrada como usuario -----	17
Figura 2.10. Código de compilación -----	17
Figura 2.11. Código 1 para detección de movimiento -----	20
Figura 2.12. Código 2 para detección de movimiento -----	21
Figura 2.13. Código 3 para detección de movimiento. -----	21
Figura 3.1. Prototipo creado en este trabajo -----	28
Figura 3.2. Arquitectura del sistema -----	29
Figura 3.3. Estructura para las cámaras de vídeo del prototipo -----	30
Figura 3.4. Estructura para los audífonos, Raspberry Pi y batería del prototipo -----	31
Figura 3.5. Soportes para las cámaras web -----	31
Figura 3.6. Soportes para batería y Raspberry Pi -----	32
Figura 3.7. Lógica utilizada en el prototipo -----	32
Figura 3.8. Reconocimiento de un objeto de color verde. -----	34
Figura 3.9. Mesa experimental para obtener la distancia hacia el objeto -----	36
Figura 3.10. Mesa experimental para obtener la distancia hacia el objeto -----	36
Figura 3.11. Función calculada con EXCEL -----	38
Figura 4.1. Fotografías del momento de evaluación del prototipo -----	39

ÍNDICE DE TABLAS

Tabla 3.1. Datos experimentales obtenidos para calcular la distancia	37
Tabla 4.1. Pruebas de corrección del ángulo de desviación	40
Tabla 4.2. Pruebas de corrección del ángulo de desviación	41
Tabla 4.3. Pruebas de medición de la distancia hacia el objeto.	42
Tabla 4.4. Pruebas de medición de la distancia hacia el objeto.	43
Tabla 4.5. Pruebas de recolección del objeto.....	44
Tabla 4.6. Resumen de los resultados obtenidos	45

RESUMEN

En este documento se expone el trabajo realizado para desarrollar un prototipo que guíe a personas discapacitadas visualmente mediante audio y visión artificial. Para tal efecto se implementó un sistema basado en una tarjeta electrónica Raspberry Pi 3 modelo B+ que adquiere imágenes de dos cámaras web conectadas a sus entradas USB y procesa las capturas para reconocer un objeto de color verde marcando su centro con un recuadro rojo.

Se utiliza el centro dado en coordenadas cartesianas (x) para calcular el ángulo de desviación del objeto con respecto a las cámaras web. Además, se aprovecha la visión estéreo del prototipo para calcular la distancia del objeto con respecto a las dos cámaras de video. Para este último caso se realizaron mediciones experimentales para determinar la ecuación que calculará la distancia.

Con la ayuda de señales de audio se guía a las personas para corregir el ángulo de desviación del objeto con respecto a las cámaras, y con la distancia calculada se generan mensajes de voz que le informan al usuario sobre la distancia que tiene hasta el objeto.

Además, se presentan los resultados en las pruebas realizadas con el prototipo, los cuales fueron asertivos en un 70% para que la persona logre tomar un objeto, empleando un tiempo promedio de 4.8 segundos entre los usuarios y con un error relativo de 3.7%. Mediante estos resultados satisfactorios se presentan las conclusiones y recomendaciones de los autores con respecto al funcionamiento del prototipo.

ABSTRACT

In this document we present the work done to develop a prototype that guides visually impaired people through audio and artificial vision. For this purpose, a system was implemented based on a Raspberry Pi 3 model B + electronic card that acquires images of two webcams connected to its USB inputs and processes the captures to recognize a green object by marking its center with a red box.

The given center is used in Cartesian coordinates (x, y) to calculate the angle of deviation of the object with respect to the web cameras. In addition, the stereo vision of the prototype is used to calculate the distance of the object with respect to the two video cameras. For this last case, experimental measurements were made to determine the equation that will calculate the distance.

With the help of audio signals, people are guided to correct the angle of deviation of the object with respect to the cameras, and with the calculated distance, voice messages are generated that inform the user about the distance to the object.

In addition, the results are presented in the tests carried out with the prototype, which were 70% assertive so that the person manages to take an object, using an average time of 4.8 seconds among the users and with a relative error of 3.7%. Through these satisfactory results, the authors' conclusions and recommendations regarding the operation of the prototype are presented.

INTRODUCCIÓN

Las personas con discapacidad visual tienen una dependencia total o parcial de familiares y terceras personas, lo que genera una carga económica alta. Debido a su carencia en funciones visuales, las actividades que realizan son muy limitadas, en el Ecuador la constitución consagra derechos laborales y permite la inclusión laboral a personas discapacitadas, pero para un desarrollo óptimo de sus tareas se necesitan dispositivos adicionales que les permitan interpretar su entorno, pero en nuestro medio son de difícil adquisición ya sea por la ausencia del mismo en el mercado nacional como los costos elevados de estos en mercados internacionales.

El presente trabajo está ordenado de la siguiente manera:

En el capítulo 1 se exponen los antecedentes de los problemas habituales que viven las personas no videntes y la respectiva justificación de lo que se busca resolver con el dispositivo electrónico creado, así como el objetivo general y los objetivos específicos para llegar a una solución que facilite la recolección de objetos para ayudar a este sector de la sociedad.

En el capítulo 2 se presenta todo el marco teórico revisado de todas las herramientas necesarias para el desarrollo del software y hardware utilizado en el presente trabajo.

En el capítulo 3 se muestra el diseño e implementación del software y hardware en cada una de las etapas que fueron necesarias para el desarrollo del prototipo, además del código y la lógica que se utilizó en el mismo.

En el capítulo 4 se refleja la tabulación de los resultados experimentales obtenidos con cada usuario no vidente, al igual que el resumen de los mismos donde se observa la eficacia del prototipo.

Por último, se presentan las conclusiones y recomendaciones que los autores obtuvieron en el desarrollo del presente trabajo.

CAPÍTULO 1

ANTECEDENTES

1.3 Tema

Desarrollo de un prototipo electrónico que guíe a personas invidentes en la recolección de objetos mediante audio y visión artificial.

1.4 Planteamiento del Problema

La inclusión de personas discapacitadas en la actualidad se genera mediante la creación de políticas y actividades que no son suficientes al no abarcar todas las necesidades para ciertos grupos, como el no vidente, los cuales necesitan en muchos casos una dependencia alta de otras personas, mediante el diseño de este prototipo se busca dar un paso hacia la eliminación de esta necesidad en las personas discapacitadas en la recolección de objetos sobre un espacio controlado, es decir generar independencia en este caso.

Las personas no videntes se encuentran expuestas a una serie de problemas debido a su discapacidad, los más comunes en sus actividades cotidianas son tumbar objetos o derramar líquidos de forma involuntaria, además el tiempo que tardan en ubicar un elemento sobre una mesa.

En el Ecuador la existencia de poca tecnología también es parte de las limitaciones que enfrenta este sector de la sociedad. Este prototipo generaría el desarrollo de más procesos para este sector y se lograría impulsar a futuro la creación de ambientes dedicados en empresas, con costos accesibles y que permitan el desarrollo de actividades cotidianas para personas no videntes, lo cual aportaría en eliminar la exclusión laboral.

1.5 Justificación

Las personas con discapacidad visual tienen una dependencia total o parcial de familiares y terceras personas, lo que genera una carga económica alta. Debido a su carencia en funciones visuales, las actividades que realizan son muy limitadas, en el Ecuador la constitución consagra derechos laborales y permite la inclusión laboral a personas discapacitadas, pero para un desarrollo óptimo de sus tareas se necesita

dispositivos adicionales que les permitan interpretar su entorno, pero en el medio actual son de difícil adquisición ya sea por la ausencia del mismo en el mercado nacional como los costos elevados de estos en mercados internacionales.

El estudio y desarrollo de un prototipo que ayude en la recolección de objetos en un ambiente controlado facilitará la ejecución de ciertas actividades para este grupo de la población.

Además, generaría una preocupación en el desarrollo de tecnología en este campo de investigación orientado a personas con discapacidades visuales de bajos recursos económicos y su impacto fundamental está orientado a mejorar el desenvolvimiento de dicha persona en su entorno de trabajo y en su vida cotidiana.

1.6 Objetivo General

Desarrollar un prototipo electrónico para la orientación de personas discapacitadas visualmente en la recolección de objetos específicos apoyado con señales sonoras y visión artificial en un espacio controlado.

1.7 Objetivos Específicos

- Analizar el estado del arte de sistemas de guiado a personas discapacitadas para la obtención de las variables que intervienen en el proceso.
- Diseñar un algoritmo para el reconocimiento y cálculo del ángulo de desviación de los objetos marcados con respecto al usuario mediante visión artificial.
- Programar un algoritmo que determine la distancia del objeto con respecto al usuario utilizando dos cámaras en modo estéreo para la localización del objeto en la mesa de trabajo.
- Implementar el prototipo electrónico que guíe mediante audio para la recolección de objetos utilizando el ángulo y la distancia encontrados anteriormente.

- Realizar pruebas con el prototipo en el guiado y recolección de objetos con personas discapacitadas visualmente para la validación de su funcionamiento.

CAPÍTULO 2

MARCO TEÓRICO

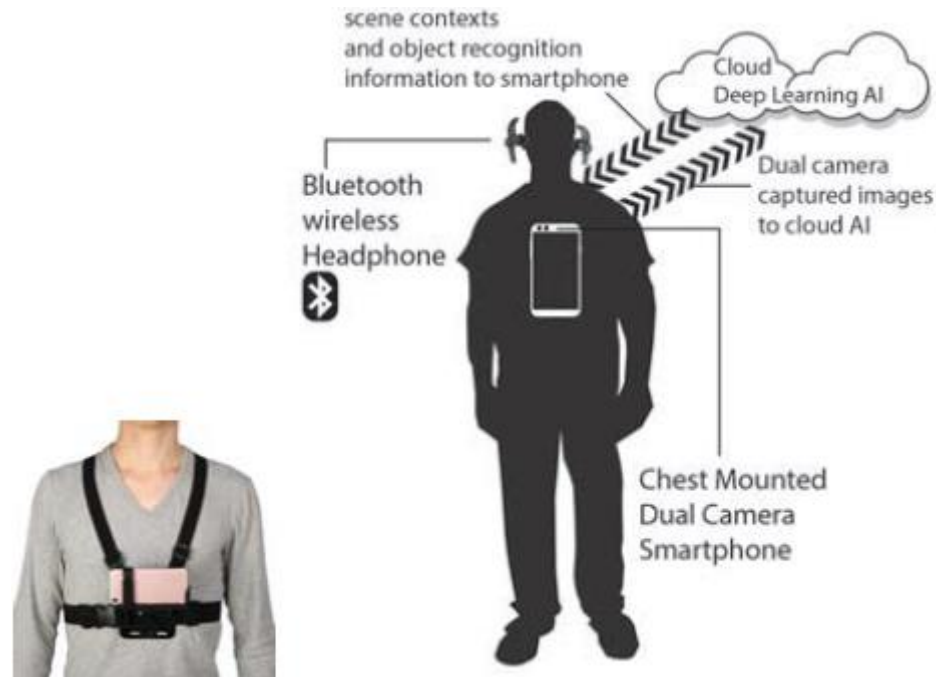
Mediante el diseño de un prototipo electrónico de orientación para personas invidentes se busca ayudarles en el proceso de recolección de objetos en una mesa de pruebas que este provista de un color neutral que evite las lecturas de los objetos en análisis, en la que mediante el uso de una tarjeta embebida se desarrollará algoritmos para el procesamiento de imágenes obtenidas con dos cámaras en modo estéreo, con los cuales se obtendrán los valores de ángulo y profundidad del objeto respecto al usuario. De esta manera se expondrá la ubicación exacta del objeto para su recolección y los individuos puedan ser guiados mediante señales sonoras, ejecutando un mapa auditivo.

2.1 Estado del arte

Se han encontrado algunos trabajos referentes al tema que abarca esta investigación, como por ejemplo el trabajo de Shen-En Shih y Wen-Hsiang Tsai que proponen un sistema de visión estéreo con dos “Omni-Camaras” que cuenta con capacidad de “Adaptación Automática” para aplicaciones de visión 3D. El autor menciona que su sistema se puede utilizar en video vigilancia, modelado de ambientes y juegos de televisión, no se menciona que esté destinado al apoyo de personas con discapacidad visual, pero es un sistema que se podría utilizar para ese fin.

Otras investigaciones como la de Minh Nguyen, Huy Le, Wei Qi Yan, Arpita Dawda aprovechan las ventajas que ofrecen las nuevas tecnologías de teléfonos inteligentes Apple iPhone 7 and 8 Plus, iPhone X, Samsung Galaxy S9, LG V30, Huawei Mate 10 con sus cámaras dobles que permiten obtener la profundidad de los objetos. En este trabajo el autor ocupa las dos cámaras del teléfono inteligente para reconocer y reconstruir de forma audible el escenario que la persona observa. El teléfono inteligente se coloca en el centro del pecho y se comunica con la persona a través de audífonos con conexión Bluetooth como consta en la Figura 2.1.

Figura 2.1. Sistema propuesto por *Minh Nguyen, Huy Le, Wei Qi Yan, Arpita Dawda*



Posición del teléfono inteligente del sistema. Fuente: *Minh Nguyen, Huy Le, Wei Qi Yan, Arpita Dawda*

Por otro lado, Vaddi Chandra Sekhar y otros utilizan dos cámaras para aprovechar su visión estéreo y propone un algoritmo de estimación de disparidad en tiempo real. El autor analiza los algoritmos de coincidencia basados en ventanas locales conocidos como suma de diferencias absolutas (SAD) y SAD de media cero (ZSAD) para la estimación de la disparidad e implementa arquitecturas de hardware eficientes para esos algoritmos en FPGA.

Paul Herghelegiu, Adrian Burlacu and Simona Caraiman también utilizan dos cámaras en modo estéreo para detectar el piso de los escenarios utilizando visión por computador. El autor menciona que detectar el piso es lo principal en su investigación ya que con ello puede detectar obstáculos que se interponen en la trayectoria del usuario. Para detectar el piso con las cámaras el autor utiliza los ángulos de inclinación “Pitch” y “Roll”, y los mapas de disparidad de cada “Frame” de la imagen.

Jaroslawn Bulat y Andrzej Glowacz proponen un sistema de ayuda para personas discapacitadas visualmente que consiste en dos cámaras y un teléfono inteligente

Samsung S3. El autor aprovecha las ventajas de la visión por computador mediante OPENCV y la visión estero de las dos cámaras para reconstruir un escenario y notificar la presencia de obstáculos al frente del usuario.

2.2 Sensor de distancia

Los sensores y transductores de distancia están diseñados premeditadamente para realizar el cálculo de la distancia o desplazamiento lineal, de una manera automática, ya que proporcionan un estímulo eléctrico según se desarrolle la variación física, en este caso la distancia. (Rea Escalante, 2014).

Los rangos de medida disponibles son muy diversos, según el tipo o sensibilidad del sensor de distancia usado. (Alfonso Hernandez, 2016). Como el sensor ultrasónico que proporciona un método muy fácil para medir distancia gracias a su precisión, utilización de pocos recursos y amplio rango de medición sea en movimiento o estática. (Rea Escalante, 2014).

2.3 Microcomputadora

Es un mini PC completa, pero en escala reducida, diseñada para un usuario a la vez. Un término obsoleto sería un dispositivo basado en un microprocesador de un solo chip. Las microcomputadoras incluyen a PC portátiles, de escritorio, algunas calculadoras, estaciones de trabajo y sistemas integrados. (Marco Felipe Lopez, 2017).

En una sociedad moderna donde las computadoras se han apoderado del mundo, o al menos de muchas profesiones, En los primeros días de la computadora personal cualquier persona que usaba un ordenador debía tener habilidades técnicas para poder utilizarla. (Aguiar, 2016)

Las computadoras de hoy en día, tienen menos problemas, están interconectadas y sin duda alguna son máquinas mucho más complejas, Es claro y evidente que la computadora evoluciono desde un artículo novedoso a herramienta esencial para la ciencia, para luego convertirse en objeto cotidiano en un corto tiempo. (Aguiar, 2016)

2.4 Visión artificial

La visión artificial es una disciplina científica que incluye métodos para adquirir, procesar y analizar imágenes del mundo real con el fin de producir información que pueda ser tratada por una máquina. (Jiménes Calle, 2012).

Una manera simple de comprender este sistema es basarse en los propios sentidos. Los humanos usan sus ojos para comprender el mundo que los rodea, y la visión artificial trata de producir ese mismo efecto en máquinas. Éstas podrán percibir y entender una imagen o secuencia de imágenes y actuar según convenga en una determinada situación. La comprensión en los dispositivos se consigue gracias a una descomposición de la imagen en pequeños fragmentos (píxeles) y en su posterior estudio (Jiménes Calle, 2012).

Visión artificial es un campo de la inteligencia artificial que comprende unas técnicas a través de las cuales se busca modelar de una forma basada en procedimientos matemáticos los procesos de percepción visual propios de los seres vivos y trasladar estas capacidades a programas y aplicaciones por computadora. La visión por computadora permitiría el análisis de propiedades y entorno de un espacio dinámico tridimensional a partir de la captura de imágenes desde uno o varios puntos para construir una escena. (Garcia Posada, 2015)

La visión artificial tiene como objetivo adquirir información visual del entorno físico y extraer características de relevancia o útiles. (Garcia Posada, 2015)

2.5 Búsqueda de objetos utilizando visión artificial

La visión artificial es un campo de la informática que funciona para permitir que las computadoras vean, identifiquen y procesen de la misma manera que lo hace la visión humana. Luego proporciona una salida adecuada para ser tratado en un proceso deseado. (Alejandro, 2009)

En la era de la digitalización, la capacidad de IA para rastrear objetos en movimiento y analizarlos juega un papel crucial. Cuantos más objetos pueda rastrear y analizar la

IA moderna, más oportunidades podremos descubrir. Es por eso que cosas como el seguimiento de objetos es tan importante. (Alejandro, 2009)

2.6 Raspberry PI

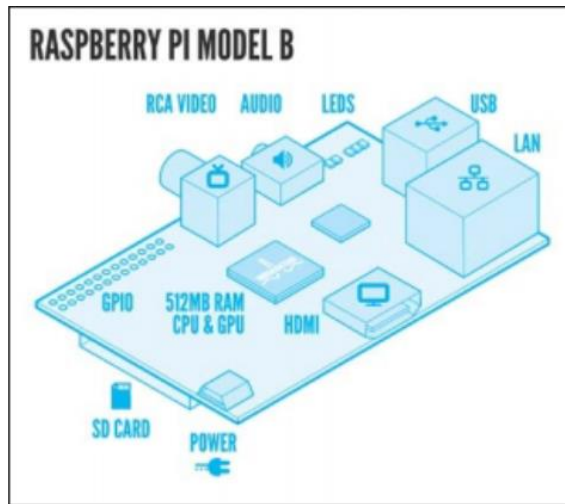
El Raspberry pi es una microcomputadora de bajo costo y tamaño compacto, que permite al usuario aprender programación en lenguajes como Python basado en código C; utiliza un teclado y un mouse estándar, el arranque de este sistema operativo es de una tarjeta micro SD. (Golden, 2015)

Raspberry es un dispositivo accesible y económico, ayuda al desarrollo de software a personas interesadas en crear nuevas rutinas de programación y sus costos relativamente bajos la convierten en una de las mejores opciones para programadores de proyectos profesionales (Pérez, 2013)

El Raspberry Pi es suministrado por una conexión del tipo micro USB a 5V y utiliza una memoria externa SD en el cual opera su sistema operativo, controlando así muchas aplicaciones útiles, se recomienda el uso de una tarjeta de capacidad mayor a 4 gigas para un funcionamiento idóneo ver Figura 2.2. (Golden, 2015)

Los dos puertos USB donde se puede conectar y poner en funcionamiento dispositivos de potencia reducida o baja como un mouse, adaptadores Bluetooth, teclados; para instalar librerías u otros componentes como un disco externo, entre otros módulos que consumen mayor potencia. (Golden, 2015)

Figura 2.2. Esquema Raspberry Pi.



La imagen anterior detalla el diagrama esquemático y las principales partes de una Raspberry Pi (Golden, 2015)

2.6.1 Características

Una de las características principales del Raspberry Pi es su accesible precio, gracias a esto se pueden desarrollar varios proyectos, ya que en su interior tiene funcionalidades de una computadora en tamaño reducido, a esta placa electrónica se le instala un sistema operativo basado en Linux, llamado Raspbian.

Las características principales de la Raspberry Pi son su tamaño, precio, módulos disponibles, haciéndola una mini computadora basada en Linux, lo que permite el desarrollo de una gran variedad de proyectos, en su sistema operativo Raspbian.

Especificaciones Técnicas de Raspberry Pi modelo A y B

<https://hardzone.es/reviews/perifericos/analisis-raspberry-pi-3-modelo-b/>

2.6.2 Periféricos

Existen varios periféricos para la Raspberry pi que se enlistará a continuación

a. Periféricos de comunicación:

- **Adaptador de red USB (Wifi):** Edimax EW-7811UN (b/g/n, 150 Mb/s). (omicron, 2016)
- **Adaptador de red USB (Bluetooth):** ASUS BT400 adaptador USB Bluetooth 4.0, con tecnología BLE para mando PS4, mando Xbox One S,

Auriculares, altavoz, teclado en ordenador de Windows 10, 8, 7, XP, Vista.
(omicron, 2016)

- **Hub USB:** Este periférico es muy buena idea para obtener unos puertos más y es clave en la Raspberry Pi, sobre todo si se quiere añadir muchas utilidades. Este modelo es muy compacto y plano, se puede pegar a la caja y queda integrado. Se recomienda el modelo, Anker Ultra Delgado Hub USB 3.0 4, le da 4 puertos USB adicionales con una velocidad de transferencia de datos de hasta 5 Gbps. (omicron, 2016)
- **Modem 3G USB:** Con este dispositivo puede tener conexión 3G, y así navegar por Internet desde cualquier lugar sin necesidad de red Wifi. Al igual que ocurre en los móviles. Un modelo interesante es el Modem 3G E1750 1750 USB Hsdpa Libre Android compatible teléfono Pincho Huawei, velocidad Hsdpa+ 7,2 Mbps, 2100 mhz. (fnac, 2019)
- **USB GPS:** Compatible con Raspberry Pi A, B, A+, B+, Zero, 2, 3 con su chip GPS L80-39, Se comunica con los satélites gracias a UART o USB CP2102 como puente USB con chip UART, estable y más rápido 66 canales de búsqueda y 22 canales de monitoreo simultaneo. (KUBII, 2019)

b. **Periféricos de Almacenamiento:**

- **Tarjeta microSD:** Sirve para añadir capacidad de almacenamiento a tu Raspberry. En ella se puede instalar el sistema operativo y otros programas que se necesite. Se puede comprar diferentes tipos dependiendo de la cantidad de almacenamiento, pero un ejemplo es la tarjeta Samsung Evo MB-MP32DA/EU (UHS -I Grade 1, Clase 10, con adaptador SD) con 32 GB de almacenamiento. (omicron, 2016)
- **Disco duro:** Tiene la misma utilidad que la tarjeta microSD, dar una capacidad de almacenamiento adicional a tu Raspberry Pi, pero la diferencia es que proporciona una mayor capacidad de almacenamiento. Un modelo recomendado es el WD My Passport Ultra – USB 3.0 súper-rápido portátil

disco duro externo SATA de 2,5 pulgadas, con 1 TB de almacenamiento. (omicron, 2016)

c. Periféricos de entrada:

- **Teclado con touchpad integrado:** Para controlar todo sin necesidad de ocupar espacio el Rii Mini X1, es un teclado inalámbrico ultra compacto que además tiene incorporado el touchpad para controlar el ratón. (omicron, 2016)
- **Mando para juegos:** Si es un fanático de los videojuegos necesita uno de estos periféricos. Para disfrutar de ellos debe incluir unos mandos a la Raspberry, un buen ejemplo es el QUMOX 2X (Controles Mandos de Juegos Clásicos N64 Para USB a PC/MAC). (omicron, 2016)

d. Periféricos de salida:

- **Pantalla:** Tontec es una pantalla táctil de 3,5 pulgadas, con una resolución algo limitada (480x320 pixeles). Para poder trabajar y ver todo lo realizado se necesita que esto se proyecte sobre algo y para eso se recurre a la pantalla. Otra opción es la pantalla oficial de Raspberry Pi, este modelo de 7 pulgadas tiene una resolución de 800x480 pixeles y también es táctil. (omicron, 2016)

2.6.3 Lenguajes de programación que se pueden usar para programar

Raspbian es el sistema operativo que se debe instalar en el Raspberry Pi, se puede programar en varios lenguajes como Javascript, Python, C/C++, aunque los más utilizados normalmente son Python y C porque están implícitos en Raspbian, además que son lenguajes universales y básicos para la programación, los sistemas operativos GNU/Linux pueden ser manejados a través de entornos gráficos y también a través de líneas de comandos que se ingresan por medio de una terminal y que funciona como una interfaz entre el usuario y el sistema operativo. (Becerra, 2016)

El sistema operativo de la Raspberry al ser una versión reducida de Debian tiene un terminal básico y fácil para programar por lo que existen funciones que solo pueden ser utilizadas por medio de la propia terminal. (Becerra, 2016)

2.7 Python

2.7.1 Definición

Python es un lenguaje de programación fácil de aprender y muy útil, cuenta con estructuras de datos eficientes y de alto nivel siendo efectivo en la programación orientada a objetos, la sintaxis y su tipado dinámico lo hacen un lenguaje de programación ideal para realizar proyectos de scripting y desarrollo rápido de aplicaciones en diversas áreas y sobre la mayoría de las plataformas. (Rossum, 2009)

El intérprete de Python puede extenderse fácilmente con nuevas funcionalidades y tipos de datos implementados en C o C++, además puede usarse como un lenguaje de extensiones para aplicaciones personalizables. (Rossum, 2009)

2.7.2 Características

En este lenguaje de programación es imposible escribir código fuente ofuscado, además de soportar objetos y estructuras de datos de alto nivel, contiene múltiples niveles de organización de código, incluye librerías que contienen un sinnúmero de clases de utilidad; la ventaja de Python es que no es necesario declarar las variables y constantes antes de utilizarlas, no requiere compilación ya que tiene un alto desarrollo y buen rendimiento además se puede utilizar en diferentes plataformas, es open source y el propósito general de este lenguaje de programación es hacer en Python todo lo que se puede hacer en Java o C#. (Jimmy Rolando Molina Rios, 2016)

2.7.3 Programación básica con el lenguaje Python

El lenguaje claro, debido a que es la combinación de potentes constructores ortogonales y no está sujeto a compatibilización con versiones anteriores que utilizan operadores “==” o/y “&&” de C; en Python son reemplazados por not, or y and además no es necesario el uso de terminadores de instrucción “;” puesto que al no tener indicador de inicio y fin de bloque (del tipo { } o begin end), el conjunto de instrucciones es identificado por su sangrado por parte del compilador, en Python

todos son punteros y se debe observar una determinada variable como referencia a un valor, no como el código identificador en la pila de memoria, cabe recalcar que se escriben módulos en C o C++. (Peña, 2015)

2.7.4 Como utilizarlo en Raspberry Pi

Sabiendo que Python es un lenguaje de programación multiplataforma se puede ejecutar en el sistema operativo de Windows, Linux o Mac; en este caso se utiliza Raspbian que viene instalado por defecto en la Raspberry, y por defecto viene instalado Python 2 y el IDE de desarrollo que se llama Thonny Python, una vez abierto se observa una pantalla llamada terminal, Shell o consola donde aparece un cursor y es ahí donde se debe empezar a programar. (Abellán, 2017)

Python es un lenguaje de programación de equipado dinámico esto quiere decir que significa que una variable puede tener diferentes tipos de datos, las variables que se utiliza son las ya conocidas en los otros lenguajes de programación como son: booleanos, cadenas de caracteres, números enteros, números decimal, listas entre otros. (Abellán, 2017)

Una condición en programación es una sentencia que comprueba su veracidad, en Python se utiliza las sentencias de control como son el: if, if else, cabe aclarar que el switch case no existe en Python pero se lo puede simular utilizando solamente condiciones de if else. (Abellán, 2017)

Los bucles son sentencias de control que son necesarias para programar iteraciones, en Python como en los demás lenguajes de programación se usa el “For” que en lenguaje común se leería como “para que suceda” y el “While” que se leería como “mientras algo suceda”. (Abellán, 2017)

2.7.5 Generación de sonidos en Python

En Python existe un paquete escrito en C llamado “pyo”, su funcionalidad es principalmente manejar audio y la generación del audio mediante algoritmos ver Figura 2.3., se puede encontrar una variedad de algoritmos para la generación de

sonidos, algunos sencillos y otros complejos, que requieren un conocimiento avanzado en python. (Python, 2014)

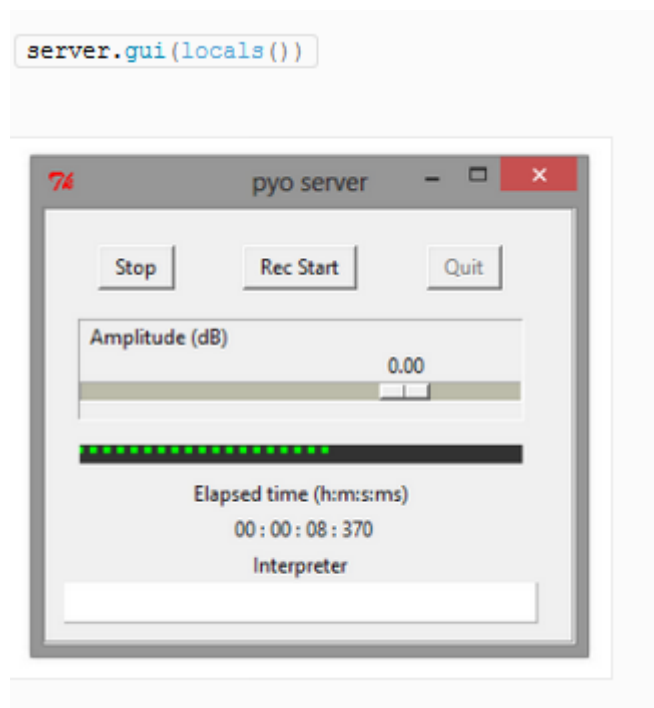
Figura 2.3. Ejemplo de la reproducción de un sonido

```
1.  from time import sleep
2.  from pyo import *
3.
4.  server = Server().boot()
5.  server.start()
6.
7.  sine = Sine(261.63, mul=0.1).out()
8.  sleep(3)
9.
10. server.stop()
```

Código para la reproducción de un audio de 261.61 [Hz], con un retardo de 3 segundos. (Python, 2014)

Para controlar la amplitud o volumen “pyo” tiene una interfaz de usuario para la señal reproducida actualmente ver Figura 2.4. (Python, 2014)

Figura 2.4. Interfaz de usuario



Interfaz de usuario para la reproducción de sonido. (Python, 2014)

Para empezar a generar sonidos en Python es necesario crear una instancia del servidor este a su vez será el encargado de inicializar y configurar el PortAudio seguido de esto se llama a la función `server.start` para que comience el proceso del audio, una de las ventajas es que se pueden editar los atributos en tiempo real lo que produce efectos instantáneos, el módulo integra varias clases para la generación, filtración de audio y aplicar efectos de sonido entre otras. (Python, 2014)

2.8 OpenCV

2.8.1 Definición

OpenCV (Open Source Computer Vision) es una librería de software open source de visión artificial y machine learning que provee una infraestructura para aplicaciones de visión artificial, contiene más de 2500 algoritmos para usar, está escrito en C++, tiene interfaces en C++, C, Python, Java y Matlab además funciona en Windows, Linux, Android y Mac OS. (Gracia, 2013)

La librería OpenCV se enfoca fundamentalmente al trabajo de la visión por computadora en tiempo real, pero primero, se debe entender algunos supuestos y paradigmas básicos cuando se analiza la imagen y el video. En el núcleo, sin embargo, son cuadros estáticos, al igual que las imágenes. Por lo tanto, el reconocimiento de imágenes y el análisis de video utilizan métodos idénticos en su mayor parte. Algunas cosas, como el seguimiento direccional, requerirán una sucesión de imágenes (cuadros), pero se puede hacer algo como la detección facial o el reconocimiento de objetos con casi el mismo código en imágenes y videos. Se debe tomar en cuenta que la visión por computadora es parte esencial de la Inteligencia Artificial (Rosales Romero, 2017)

2.8.2 Como instalar OpenCV en Raspberry Pi

Para instalar OpenCV en una Raspberry Pi con Raspbian, cabe recalcar que Raspbian es una distribución de Linux por lo tanto las instrucciones a seguir deberían ser las mismas para Debian y Ubuntu, se debe tener una Raspberry configurada y preparada; en las siguientes imágenes ver Figura 2.5., se describe los pasos para instalar OpenCV en una Raspberry Pi con Raspbian. (G14r3, ROBOLOGS, 2014)

Figura 2.5. Comandos para instalar OpenCV

```
sudo apt-get update
```

```
sudo apt-get install libopencv-dev python-opencv
```

Abrir el terminal y escribir el código descrito en la imagen (Gl4r3, ROBOLOGS, 2014)

En la Figura 2.6. se manifiesta el código para instalar todos los paquetes y librerías que falten. (Gl4r3, ROBOLOGS, 2014)

Figura 2.6. Comando para instalar paquetes y librerías

```
sudo apt-get -f install
```

Comando para instalar los paquetes y librerías faltantes (Gl4r3, ROBOLOGS, 2014)

Una vez instalado se ingresa el siguiente código ver Figura 2.7.:

Figura 2.7. Comando para instalar libopencv

```
sudo apt-get install libopencv-dev python-opencv
```

Comando para instalar libopencv (Gl4r3, ROBOLOGS, 2014)

Para corroborar que está instalada se debe importar a la consola de Python, si no sale ningún error ya estaría OpenCV lista para trabajar en la Raspberry, al contar con versión para Python, se puede combinar con RPi.GPIO, librería orientada a la robótica. (Gl4r3, ROBOLOGS, 2014)

2.8.3 Como utilizar Python en Raspberry PI

Al ya tener instalado OpenCV en la Raspberry Pi, se empieza por abrir la terminal de programación de Raspbian, al abrirse la terminal o Shell se puede iniciar la programación sin ningún problema ya que todos los paquetes de OpenCV estarán instalados y funcionando.

En la Figura 2.8. se pone en manifiesto el comando para crear este entorno, la orden mkvirtualenv dependiendo de la versión de Python que se utiliza. (García, 2019)

Figura 2.8. Comando mkvirtualenv

```
$ mkvirtualenv cv -p python2
```

Comando mkvirtualenv en Python 2.7. (García, 2019)

Para activar el entorno virtual, utiliza la orden workon, tras esto se observa un prompt que muestra el entorno virtual en el que se desarrolla y a partir de este momento va a realizar las operaciones dentro del entorno cv; debe instalar numpy para que pueda ser usado por OpenCV, tras todas estas operaciones se establece la carpeta de usuario ver Figura 2.9., y a continuación la compilación ver Figura 2.10. (García, 2019)

Figura 2.9. Entrada como usuario

```
$ cd ~/opencv-3.2.0/  
$ mkdir build  
$ cd build  
$ cmake -D CMAKE_BUILD_TYPE=RELEASE \  
-D CMAKE_INSTALL_PREFIX=/usr/local \  
-D INSTALL_PYTHON_EXAMPLES=ON \  
-D OPENCV_EXTRA_MODULES_PATH=~/opencv_contrib-3.1.0/modules \  
-D BUILD_EXAMPLES=ON ..
```

Entrada desde nuestra carpeta de usuario. (García, 2019)

Figura 2.10. Código de compilación

```
$ make -j4
```

Código de compilación. (García, 2019)

2.8.4 Formato RGB

El formato RGB es un estándar utilizado para representar los colores de imágenes digitales, el cual está formado por los colores de luz primarios es rojo, verde y azul. Este modelo de color se utiliza en monitores de computadora, televisores y otras

pantallas. Si pone una lupa contra la pantalla del televisor, se puede observar esta cadena de colores. (FotoNostra, s.f.)

El formato RGB es un modelo de color aditivo, en el cual su representación se base en los tres colores primarios, para asignar un valor en el rango de cero a doscientos cincuenta y cinco, dependiendo de la intensidad de cada pixel. (FotoNostra, s.f.)

RGB no proporciona una definición exacta de los tres colores por lo tanto los colores varían mucho de acuerdo al modelo RGB que se va a emplear, en la informática los lenguajes de programación utilizan el modelo RGB para representar los colores, la dinámica de los lenguajes de programación con respecto a este modelo es asignar un valor a los distintos colores, a mayor valor mayor intensidad en la mezcla. (Merino, 2016)

OpenCV incluye operaciones para cambiar el modelo de color de una imagen en el caso del modelo RGB la operación es BGR, para cambiar entre formatos de imagen RGB sirve como un puente ya que para transformar de cualquier formato a otro el primer paso es convertir a RGB. (Merino, 2016)

2.8.5 Formato HSV

HSV (hue, saturation, value); HSV (Matiz, saturación, valor) es un modo de imagen o modelo de colores que están definidos por sus componentes y la variación del grado de propiedad del color y así poder crear nuevos colores. El matiz se representa al color propio de estudio, el mismo que se desarrolla por ángulos y cada ángulo corresponde a un color. La saturación es la viveza del color, que se constituye por el diagrama de cromaticidad del modelo. El valor está representado por la distancia existente entre los ejes blanco y negro. (Lizarazo Zambrano & Ramos Velandia, 2016)

Una imagen en color se puede codificar en formato RGB (redgreenblue) o en el formato HSV (valor de saturación): estos son dos diferentes representaciones de los mismos datos. Algunas tareas que pueden ser difíciles con una representación, pero puede ser fácil con otra. Por ejemplo, la tarea “seleccionar todos los pixeles rojos en la imagen” es más fácil en el formato RGB, mientras que “hacer la imagen menos

saturada” es más simple en el formato HSV. En aprendizaje automático los modelos tratan de encontrar representaciones apropiadas para sus datos de entrada. (Barrero, 2018)

Es bastante usado en las aplicaciones gráficas para que el usuario escoja colores y al mismo tiempo grados de saturación y brillo del mismo, además tiene un código hexadecimal. (Folch, 2010)

2.8.6 Máscara

La máscara es la imagen que debe procesar en un array de cero a uno, cuando se pide a Open CV que muestre una máscara por pantalla, aparecerá una imagen negra con cuadros blancos donde lo más común es realizar una transformación morfológica para depurar la imagen, para esta transformación se necesita la máscara original y un kernel o matriz de convolución (el kernel es un array más pequeño sobre el cual va a realizar una transformación). (Gl4r3, Robologs, 2015)

Cuando Open CV realiza una transformación morfológica lo que hace es sobreponer el kernel encima de cada uno de los pixeles de la máscara y realiza una operación entre pixeles, existen varias transformaciones y a continuación se nombra las más importantes: (Gl4r3, Robologs, 2015)

- **Erosion:** Reduce el tamaño de los objetos blancos de la máscara acotando su límite
- **Dilation:** Hinchla la zona blanca de la máscara.
- **Morphological Gradient:** Es la diferencia entre las dos anteriores, es útil para encontrar la silueta de los objetos.
- **Opening:** Sirve para eliminar el ruido blanco sobre las zonas negras.
- **Closing:** Limpia el ruido negro en las áreas blancas.
- **Top Hat:** Es la diferencia entre la imagen original y su opening.
- **Black hat:** Es la diferencia entre el closing y su imagen original. (Gl4r3, Robologs, 2015)

2.8.7 Momentos

Los momentos son una forma de detección de movimiento que simplemente compara dos imágenes y encuentra la diferencia entre ambas, ya que comparar pixel por pixel no sería viable ni óptimo se realiza otro procedimiento que consiste en capturar una secuencia de tres imágenes, transformarlas a una escala de grises para mejorar el rendimiento del proceso, reducir la resolución de las imágenes previas para mejorar el rendimiento a la hora de comparar ya que mientras más alta calidad tenga la imagen más lento es el procedimiento, seguido de esto se compara las imágenes mediante diferencia de píxeles y por último remarcar las zonas donde exista un cambio relevante. (akirasan, 2016)

2.8.8 Ejemplos de programación en Raspberry

El ejemplo a proponer Figura 2.11. es el de un detector de movimiento con Open CV en Python y lo primero que va a realizar es extraer los objetos que hay en la escena para esto se utiliza el método de substracción de fondo en el cual se toma una imagen de la escena sin movimiento y se procede a restar los sucesivos fotogramas que se obtienen con la cámara. Figura 2.12., el resultado dará un fondo negro y la detección de movimiento de un color diferente. (Hernández, 2016)

Figura 2.11. Código 1 para detección de movimiento

```
1 # Importamos las librerías necesarias
2 import numpy as np
3 import cv2
4 import time
5
6 # Cargamos el vídeo
7 camara = cv2.VideoCapture("detector-movimiento-opencv.mp4")
8
9 # Inicializamos el primer frame a vacío.
10 # Nos servirá para obtener el fondo
11 fondo = None
12
13 # Recorremos todos los frames
14 while True:
15     # Obtenemos el frame
16     (grabbed, frame) = camara.read()
17
18     # Si hemos llegado al final del vídeo salimos
19     if not grabbed:
20         break
21
22     # Convertimos a escala de grises
23     gris = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
24
25     # Aplicamos suavizado para eliminar ruido
26     gris = cv2.GaussianBlur(gris, (21, 21), 0)
27
28     # Si todo se ha hecho obtenemos el fondo. Lo obtenemos
```

Código básico de detección de movimiento. (Hernández, 2016)

Figura 2.12. Código 2 para detección de movimiento

```
28 # Si todavía no hemos obtenido el fondo, lo obtenemos
29 # Será el primer frame que obtengamos
30 if fondo is None:
31     fondo = gris
32     continue
33
34 # Calculo de la diferencia entre el fondo y el frame actual
35 resta = cv2.absdiff(fondo, gris)
36
37 # Aplicamos un umbral
38 umbral = cv2.threshold(resta, 25, 255, cv2.THRESH_BINARY)[1]
39
40 # Dilatamos el umbral para tapar agujeros
41 umbral = cv2.dilate(umbral, None, iterations=2)
42
43 # Copiamos el umbral para detectar los contornos
44 contornosimg = umbral.copy()
45
46 # Buscamos contorno en la imagen
47 im, contornos, hierarchy = cv2.findContours(contornosimg, cv2.RETR_TREE, cv2.CHAIN_AP
48
49 # Recorremos todos los contornos encontrados
50 for c in contornos:
51     # Eliminamos los contornos más pequeños
52     if cv2.contourArea(c) < 500:
53         continue
54
55     # Obtenemos el bounds del contorno, el rectángulo mayor que engloba al contorno
56     (x, y, w, h) = cv2.boundingRect(c)
57     # Dibujamos el rectángulo del bounds
58     cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 255, 0), 2)
```

Código básico de detección de movimiento. (Hernández, 2016)

Figura 2.13. Código 3 para detección de movimiento.

```
59
60 # Mostramos las imágenes de la cámara, el umbral y la resta
61 cv2.imshow("Camara", frame)
62 cv2.imshow("Umbral", umbral)
63 cv2.imshow("Resta", resta)
64 cv2.imshow("Contorno", contornosimg)
65
66 # Capturamos una tecla para salir
67 key = cv2.waitKey(1) & 0xFF
68
69 # Tiempo de espera para que se vea bien
70 time.sleep(0.015)
71
72 # Si ha pulsado la letra s, salimos
73 if key == ord("s"):
74     break
75
76 # Liberamos la cámara y cerramos todas las ventanas
77 camara.release()
78 cv2.destroyAllWindows()
```

Código básico de detección de movimiento. (Hernández, 2016)

Lo primero que se debe hacer para crear el algoritmo que permite la detección de movimiento, es convertir a escala de grises y a su vez eliminar el ruido, después se realiza una operación de sustracción entre el la nueva imagen y la anterior, aplicar un

umbral a la imagen resultado de la resta y detectar los contornos ver Figura 2.13 (Hernández, 2016)

2.9 Cámara WEB

2.9.1 Definición

En la actualidad es imprescindible el uso de una cámara web en todos los dispositivos debido al avance de las tecnologías denominadas el internet de las cosas, para hacer posible la comunicación con el mundo mediante audio e imagen de una manera digital (Informatica, 2019)

La cámara web es un dispositivo de entrada que no tiene ningún almacenamiento incorporado a diferencia de una videocámara o cámara digital; siempre está conectada a una computadora y su almacenamiento se da en el disco duro o en un espacio de la nube, captura imágenes fijas o vídeo de un usuario u objeto, la mayoría tiene conexión USB o se encuentran integradas en el mismo equipo. (Informatica, 2019)

2.9.2 Características

La principal característica de una cámara web es la resolución, dada por los pixeles a lo ancho y a lo largo de su área de captación, la elección de esta dependerá del proceso en la cual se utilice, pues si es algo simple como detectar un color, con un pixelaje bajo bastara, mientras si se necesita clasificar detalles del entorno, la que procese un número mayor de pixeles será la adecuada. (Informatica, 2019)

Las webcams más costosas y avanzadas suelen incluir componentes que aporten en la mejora de la calidad de imagen, como la posibilidad de captar objetos a largas distancias mediante zoom o incluso en ambientes con baja luminosidad mediante visores infrarrojos, adaptándose de esa manera a cualquier ambiente. (Informatica, 2019)

2.9.3 Resolución

La resolución de la cámara es el que definirá la cantidad de datos a procesar, y a su vez generará la calidad de la imagen, la mayoría en el mercado posee configuraciones de 320x420 y 640x480, esta medida indica el máximo valor soportado para la captación de datos por esto, la elección de la misma dependerá del proceso donde será utilizada. (Informatica, 2019)

2.9.4 Resolución mínima para utilizarla en visión por computadora

La resolución mínima para utilizar una cámara web en visión por computadora depende de la aplicación que se vaya a dar a la cámara ya que existen dispositivos de baja y alta definición, cabe recalcar que mientras más resolución tenga la cámara es más lento el manejo de los datos, en los aplicativos para detección de movimiento se demora mucho en procesar si se tiene una cámara de alta resolución. (Informatica, 2019)

2.9.5 Cámara Web Havit Hvn5082 640x480 16mpx

Este tipo de dispositivo es una cámara de alta definición para ordenadores portátiles o monitores LCD, tiene un sensor CMOS, tiene 301600 pixeles con software, su resolución dinámica es de 640x480 con una tasa máxima de 25 fps cuadros, tiene AVI /YUY2 como formato de salida dinámica, además de un puerto USB 2.0 y soporta el sistema Windows vista en adelante. (COMPUDISC, 2013)

2.10 Impresión 3D

2.10.1 Definición

La impresión 3D o manufactura por adición, es la fabricación asistida por computadora (CAM) que crea objetos tridimensionales. Todos los procesos de impresión 3D requiere que el software, el hardware y los materiales trabajen en conjunto, sin embargo, en lugar de imprimir la salida en papel, crea un modelo tridimensional a partir de material personalizado; la principal funcionalidad de estas impresiones es la de crear cosas, desde prototipos y piezas simples hasta órganos artificiales que se producen con capas de células humanas. (Autodesk, Autodesk, 2019)

2.10.2 Características

Quizá el primer formato de modelos digitales en 3D es el .DWG (DraWinG). Se creó en 1982 con el lanzamiento de la primera versión de AutoCAD. Aún hoy es utilizado como el formato de archivo nativo de este programa, aunque a veces represente una dificultad para la migración de un archivo de un lugar a otro. (OMF, 2018)

Uno de los formatos más generalizados en la impresión 3D es el .STL (del inglés Stereo Lithography) que logra cierta precisión en la geometría de los objetos modelados por medio de la creación de un plano cartesiano en el que los objetos se construyen a través de triángulos, pero descarta la información de las propiedades cromáticas, así como también de las texturas, por lo que ya está un poco obsoleto y le han salido al paso otros formatos más decantados. (OMF, 2018)

Por otro lado, un formato que se ha ido estandarizando de a poco es el AMF (en inglés “Additive Manufacturing File”). Está basado en lenguaje XML, permite la lectura de información cromática y el trabajo con múltiples materiales. (OMF, 2018)

Con el formato AMF se puede construir estructuras más complejas, por lo que ofrece muchas más posibilidades. Además, los archivos en este formato pesan menos que en STL. Entre los más conocidos está también el 3FM, creado hace pocos años por un consorcio integrado por empresas como Microsoft y Hewlett Packard. (OMF, 2018)

Una de sus ventajas es que contiene toda la información del producto en un solo archivo luego del diseño en el software de CAD, simplificando el proceso; es también uno de los más completos, por lo que logra una descripción muy acertada del producto modelado. (OMF, 2018)

2.10.3 Como modelar piezas en inventor

Un modelo de pieza de Autodesk Inventor es una colección de operaciones relacionadas de forma geométrica y dimensional que representan un objeto físico. Para modelar una pieza en Inventor, se comienza dibujando bocetos para definir perfiles y rutas de operaciones, a continuación, se procede a utilizar los comandos

necesarios para la geometría y paramétrica, finalizando con la obtención de las piezas tridimensionales requeridas. La mayoría de las operaciones se crean a partir de bocetos mientras que otras mecánicas no las requieren. (Autodesk, Autodesk, 2019)

Todas las operaciones son combinadas para crear piezas complejas, estas utilizan cotas geométricas y restricciones, y si alguna curva de operación se quedara sin acotar, se pueden convertir en operación adaptativa, de esta manera puede cambiar su tamaño cuando la restrinja a la geometría fija de un ensamblaje. (Autodesk, Autodesk, 2019)

2.10.4 Como imprimir piezas modeladas con inventor

Coloque y prepare el modelo para la impresión en 3D, el entorno de impresión 3D proporciona herramientas para la fabricación de aditivos. Se pueden imprimir modelos enteros o parciales en función de la impresora 3D en uso. Este entorno solo está disponible en el documento de pieza. Para imprimir un ensamblaje (no disponible en Inventor LT), guarde el ensamblaje como un archivo STL y envíe el archivo a una impresora o a un distribuidor de bienes y servicios de impresión. (Autodesk, Autodesk, 2019)

La impresión en 3D se basa en la visibilidad de la operación y de los componentes. Para operaciones que no desee imprimir, cambie el estado de visibilidad o desactive la operación si eso no afecta a otras operaciones. (Autodesk, Autodesk, 2019)

Al entrar en el entorno de impresión en 3D, el modelo está orientado con el eje Z apuntando hacia arriba. Puede: cambiar la orientación y la posición del componente, modificar el componente para la impresión y dividir el componente. Las particiones se pueden anidar en el mismo espacio de generación de la impresora o exportarse para usarlos en otra sesión de impresión. (Autodesk, Autodesk, 2019)

Muchas de las herramientas de modelado de piezas normalizadas y las herramientas de boceto están disponibles en el entorno de impresión en 3D, de forma que el modelo de impresión se puede modificar. Las modificaciones realizadas mientras se trabajan en la impresión en 3D solo afectan a la pieza impresa, no al modelo, cuando

se utiliza el entorno de impresión en 3D, las opciones Enviar a servicio de impresión en 3D y Vista preliminar en 3D están desactivadas. (Autodesk, Autodesk, 2019)

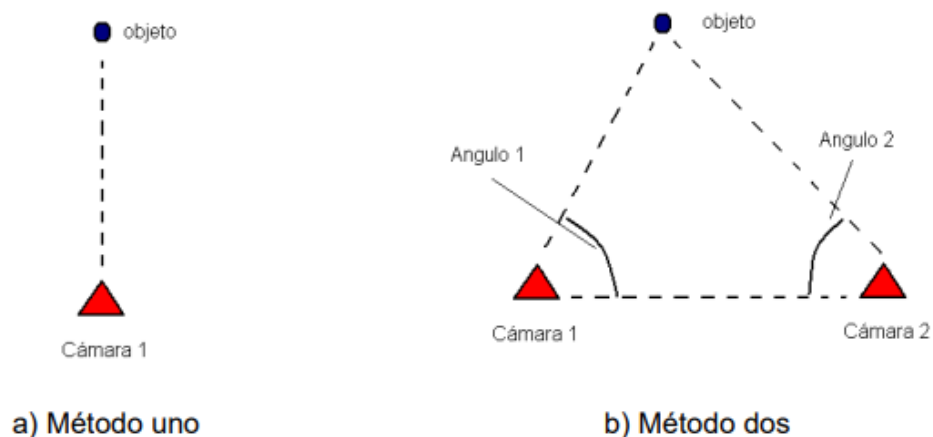
2.11 Triangulación con dos cámaras

2.11.1 Definición de técnica

Cuando un objeto es detectado por dos cámaras y además se conoce la posición de ambas entonces es posible medir la distancia entre éstas y el objeto, o la distancia correspondiente a la altura del triángulo que se forma con los tres vértices, además, medir los ángulos de giro que hay entre esta línea y el objeto no es una tarea complicada. (Camacho, 2009)

De esta manera justamente se conoce uno de los lados del triángulo y los dos ángulos correspondientes a los extremos como se muestra en la Figura 2.14., de este lado pudiendo ciertamente aplicar las fórmulas de la triangulación para encontrar cualquier distancia restante. De esta manera, sólo resta elegir desde cuál de las cámaras se desea saber la distancia al objeto ya que puede ser de cualquiera. O en su defecto, si se desea saber la distancia correspondiente a la altura del triángulo. (Camacho, 2009)

Figura 2.14. Métodos para calcular la distancia



Código básico de detección de movimiento. (Camacho, 2009)

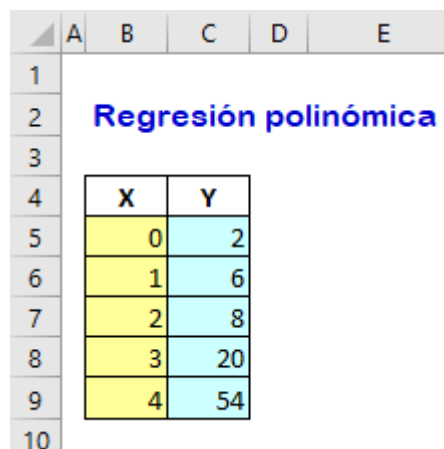
2.11.2 Regresión polinómica en Excel

Se puede realizar un arreglo un conjunto de datos representados como puntos mediante una recta, ésta sería la típica regresión lineal $y=a+bx$. A su vez se puede realizar el mismo arreglo para obtener una curva característica del tipo polinomio de grado dos, una parábola $y=ax^2+bx+c$. Y a su vez se puede ir subiendo el grado del polinomio para ver si el ajuste mejora o tiene una mejor tendencia. (Aparicio, 2018)

En Excel se dispone de la función **ESTIMACIÓN LINEAL** que permite obtener los coeficientes de estos polinomios y el coeficiente de determinación R^2 , así como otras magnitudes estadísticas de errores y desviaciones propias del ajuste realizado. Para obtener el coeficiente de correlación R se debe aplicar la radicación a R^2 , el signo del coeficiente de correlación indica si la recta en el caso de ser lineal o curva de ser polinómica sea creciente (signo positivo) o es decreciente (signo negativo). (Aparicio, 2018)

Se cuenta con un conjunto de datos formado por dos variables X e Y. La variable X es la variable independiente y la variable Y es la variable dependiente de X. En la Figura 2.15., se expresa un ejemplo de datos a calcular mediante Excel para su posterior cálculo de regresión lineal. (Aparicio, 2018)

Figura 2.15. Datos para calcular la regresión



	A	B	C	D	E
1					
2		Regresión polinómica			
3					
4		X	Y		
5		0	2		
6		1	6		
7		2	8		
8		3	20		
9		4	54		
10					

Ejemplo de datos para cálculo de una regresión polinómica haciendo uso de Excel. (Aparicio, 2018)

CAPÍTULO 3

DISEÑO E IMPLEMENTACIÓN DEL SISTEMA

En este capítulo se detalla el desarrollo del prototipo en cada una de sus etapas y se muestra las metodologías utilizadas en cada fase. Se empieza mostrando la arquitectura del sistema realizado y el diseño de las piezas del prototipo en Inventor. Luego se detalla la lógica del algoritmo creado con Python sobre una Raspberry Pi para conseguir el resultado final que se presenta en la Figura 3.1.

Figura 3.1. Prototipo creado en este trabajo



Prototipo finalizado y usado en las pruebas. Elaborado por: Cristhian Gutiérrez y Hedian Herrera

3.1 Arquitectura del sistema

Para crear el prototipo se utilizó una tarjeta Raspberry Pi, dos cámaras de video con conexión USB y dos audífonos para proporcionarles mensajes de audio a los usuarios. En la Figura 3.2 se presenta la interconexión de los elementos que componen el sistema.

Figura 3.2. Arquitectura del sistema



Interconexión de los elementos del sistema. Elaborado por: Cristhian Gutiérrez y Hedian Herrera

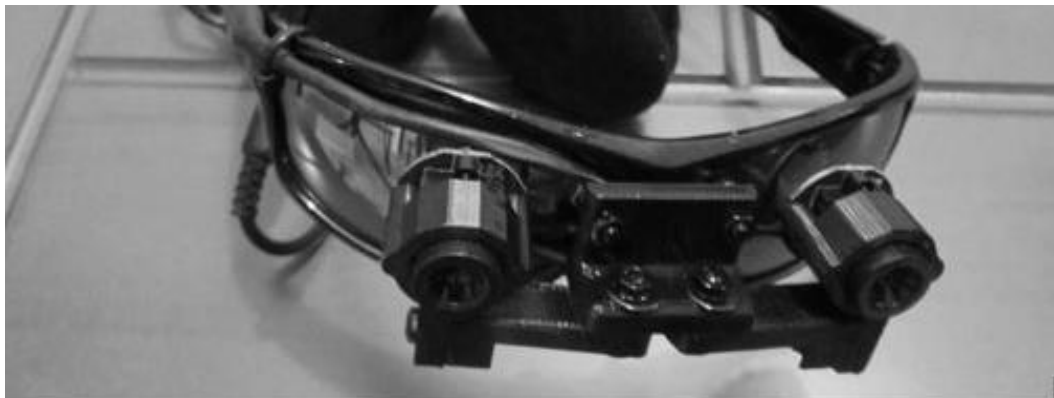
En la Figura 3.2 se puede observar que el controlador del sistema es la tarjeta electrónica Raspberry Pi 3 modelo B+. Para alimentar el sistema se utilizó una batería portable “iLuv” de 2500 amperios hora y se conecta al sistema por medio de un cable USB. Los audífonos son reciclados y se utilizaron como estructura principal de sujeción para el prototipo. Las cámaras de vídeo que se utilizaron son de marca “Havit” y tienen una resolución de 16 Mega pixeles.

3.2 Estructura del sistema

Para la creación del prototipo se pensó en una manera de colocar dos cámaras de vídeo a la altura de los ojos de los usuarios para que se muevan de acuerdo con su cabeza utilizando unas gafas de sol. Además, se observó la necesidad de añadirle audio con dos audífonos para guiar con sonido a las personas que utilicen el prototipo.

También, como el objetivo fue controlarlo en base a una tarjeta Raspberry Pi y que el prototipo sea portable se colocó una batería con soportes diseñados en una impresora 3D en el marco de los audífonos. En la Figura 3.3 se presenta la estructura que se utilizó para sujetar las cámaras de vídeo y en la Figura 3.4 se observa la estructura que se utilizó para sujetar la Raspberry Pi y la fuente portable del sistema.

Figura 3.3. Estructura para las cámaras de vídeo del prototipo



Gafas de sol donde se colocan las cámaras con la ayuda de unos acoples diseñados en una impresora 3D. Elaborado por: Cristhian Gutiérrez y Hedian Herrera

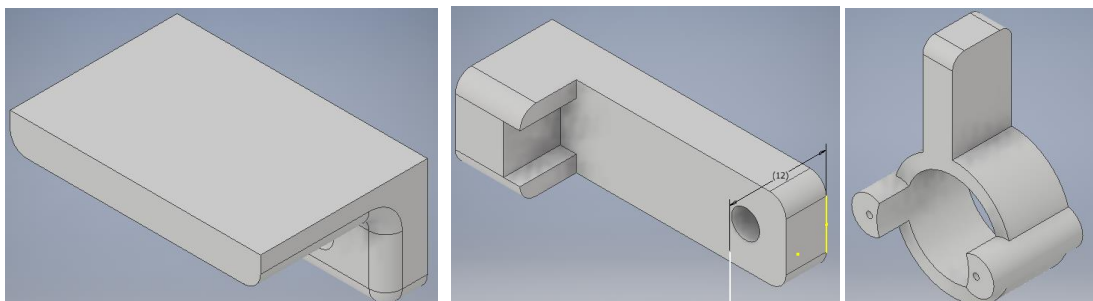
Figura 3.4. Estructura para los audífonos, Raspberry Pi y batería del prototipo



Estructura donde se colocan los audífonos, Raspberry Pi y batería. Elaborado por: Cristhian Gutiérrez y Hedian Herrera

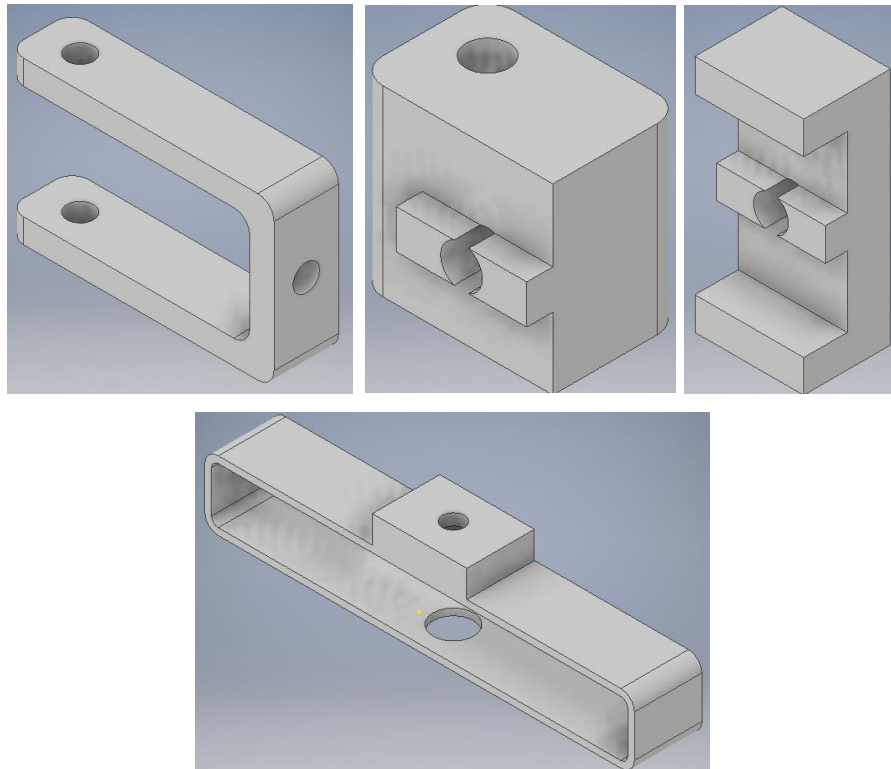
Para sujetar la Raspberry Pi, la batería portable y las cámaras web se diseñaron varios soportes en Inventor y se imprimieron en una impresora 3D. Dichos soportes se presentan en las Figura 3.5 y 3.6 respectivamente.

Figura 3.5. Soportes para las cámaras web



Soportes para las cámaras web diseñados con Inventor. Elaborado por: Cristhian Gutiérrez y Hedian Herrera

Figura 3.6. Soportes para batería y Raspberry Pi

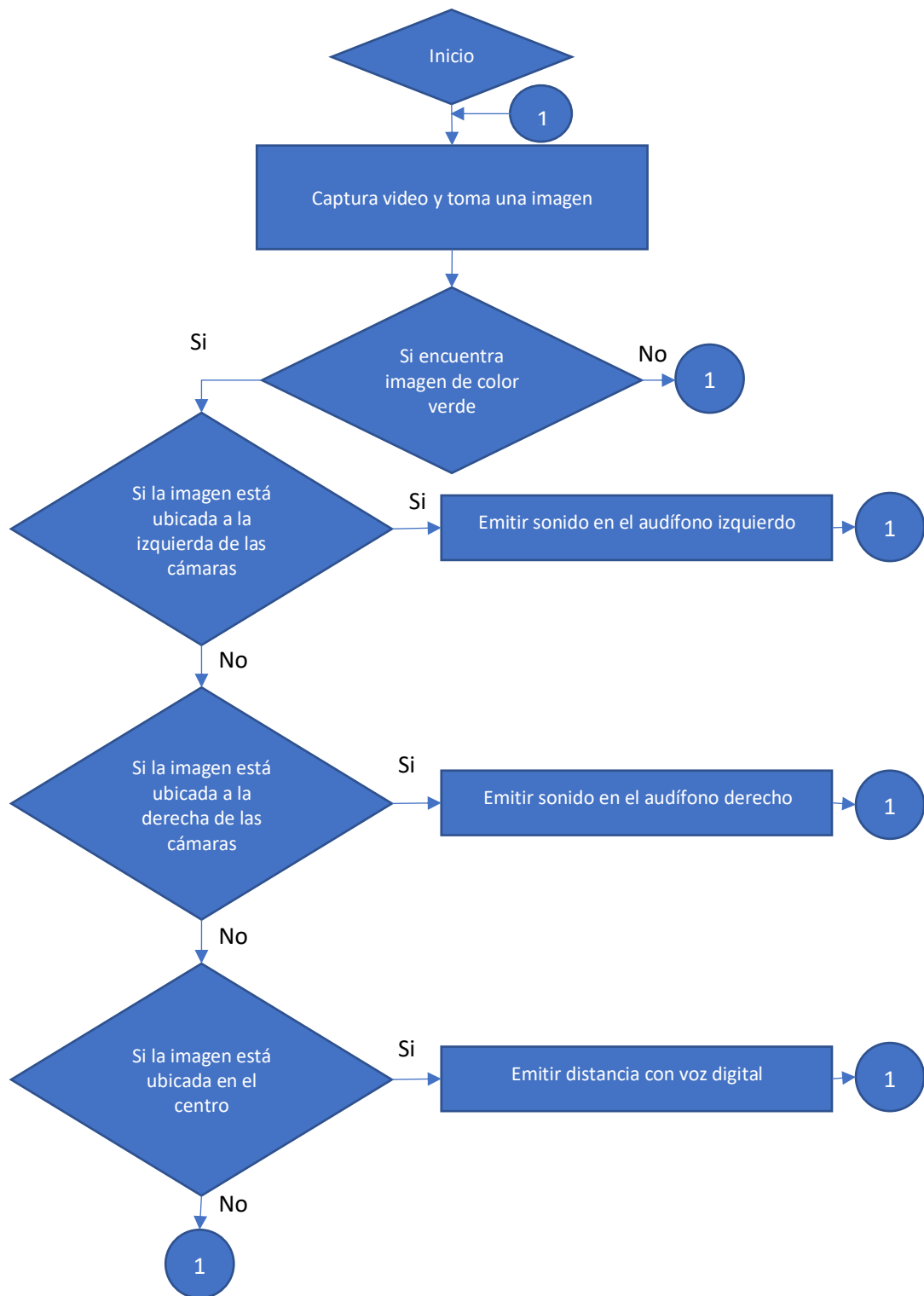


Soportes para batería y Raspberry Pi diseñados con Inventor. Elaborado por: Cristhian Gutiérrez y Hedian Herrera

3.3 Algoritmo creado

Para buscar un objeto marcado sobre una mesa se utilizó visión artificial y se creó un algoritmo utilizando Python y OpenCV en una tarjeta Raspberry Pi. Además, se usaron dos audífonos en la salida de audio de la tarjeta Raspberry Pi para emitir sonidos audibles que fueron controlados con la librería “pygame.mixer” en Python.

Figura 3.7. Lógica utilizada en el prototipo



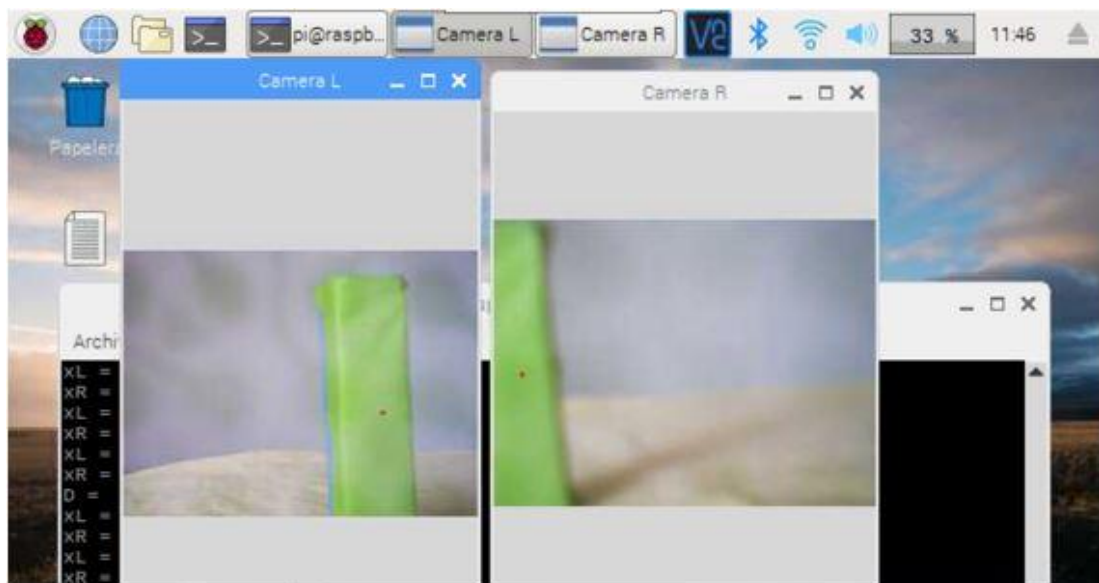
Lógica utilizada para crear el algoritmo en Python y OpenCV. Elaborado por: Cristhian Gutiérrez y Hedian Herrera

Para generar los sonidos se utilizó el comando “`pygame.mixer.init()`” que permite ajustar una frecuencia y el volumen con que se emiten las señales sonoras, además, se utilizó el comando “`sndR = pygame.mixer.Sound("beep-07.wav")`” para cargar un

sonido con extensión WAV en el código y ejecutarlo luego mediante los comandos “SR.set_volume(0.5,0)” y “SR.play(sndR)” que controlan el volumen y el parlante donde suena, es decir, en el audífono derecho o en el audífono izquierdo. La lógica seguida en el algoritmo se presenta en la Figura 3.7.

El algoritmo utilizado para buscar el objeto sobre la mesa empieza por adquirir una imagen del vídeo captado con las cámaras web colocadas en las gafas de sol con la sentencia “captura = cv2.VideoCapture(0)”. La imagen capturada es tratada para reconocer color verde mediante “verde_bajos=np.array([47,95,160],dtype=np.uint8)”, encontrar un área de un tamaño determinado para descartar ruidos y obtener el área y el centro del objeto encontrado mediante “area=moments['m00]”, “x=int(moments['m10']/moments['m00'])” y “y=int(moments['m01']/moments['m00'])”.

Figura 3.8. Reconocimiento de un objeto de color verde.



Reconocimiento y determinación del centro de un objeto de color verde con Python y OpenCV.
Elaborado por: Cristhian Gutiérrez y Hedian Herrera

Una vez que se obtiene el centro del objeto con coordenadas (x) se calcula la distancia de desviación del usuario con respecto al objeto en el eje “x”. Para corregir la desviación del usuario al objeto se genera un sonido de alarma en intervalos de

tiempo en el audífono del lado a donde debe girar la cabeza la persona. En la Figura 3.8 se puede observar cómo se ha detectado el objeto de color verde y las coordenadas que se escriben en la pantalla de comandos señalando el centro del objeto.

Al final, si se coloca el objeto en frente del usuario después de realizar un barrido con el movimiento horizontal de la cabeza, el algoritmo dicta de forma audible la distancia que hay hasta dicho objeto. Para generar la distancia el algoritmo lo redondea en múltiplos de cinco, es decir que si se calcula una distancia de 33 centímetros el código lo redondea a 35 centímetros. Con la distancia conocida se espera guiar de cierta manera al usuario para alcanzar el mismo sobre la mesa.

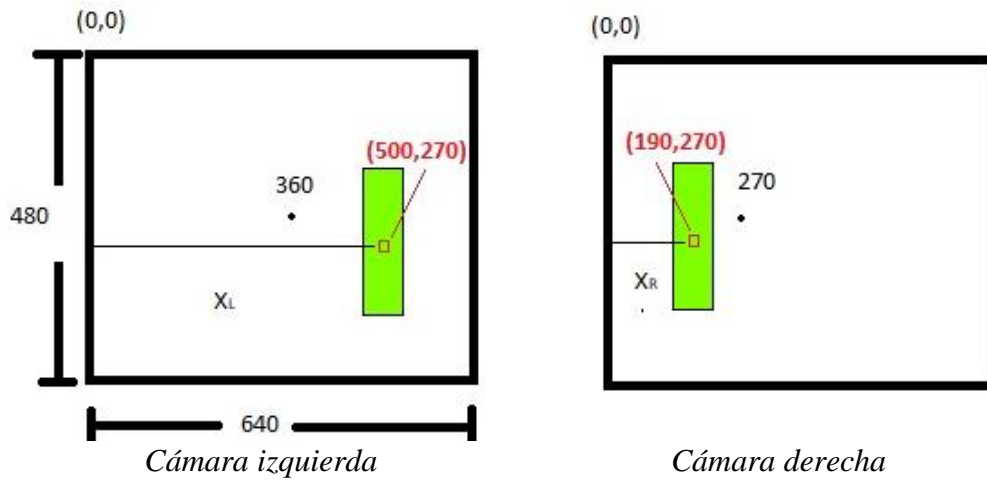
3.4 Cálculo del ángulo de desviación

Para encontrar el ángulo de desviación del usuario con respecto al objeto se utilizó las coordenadas del centro objeto de color verde encontrado en la sección 3.3. Se observó que las cámaras tanto izquierda como derecha veían de distintas posiciones el objeto y si se ubicaba el mismo en dos posiciones distintas dependiendo de la cámara. En la Figura 3.9 se puede observar las distintas posiciones del objeto visto de ambas cámaras.

Si se obtiene el ángulo de desviación del objeto con respecto a una de las cámaras este valor sería diferente al calculado desde la otra cámara, por lo que se tendrían dos ángulos distintos para el mismo objeto observado en modo estéreo. Por esta razón se omitió el cálculo del ángulo de desviación y se colocó un intervalo de coordenadas en el eje “X” que permitirían colocar el objeto frente a las dos cámaras.

Experimentalmente se observó que el intervalo adecuado de coordenadas en el eje “X” de la cámara izquierda es de 360 a 640 y de la cámara derecha es de 0 a 270. Cabe recalcar que si el objeto se sale de estos intervalos significará que el objeto se encuentra descentrado del centro de las cámaras y que habrá que centrarlo corrigiendo el ángulo de desviación.

Figura 3.9 Mesa experimental para obtener la distancia hacia el objeto

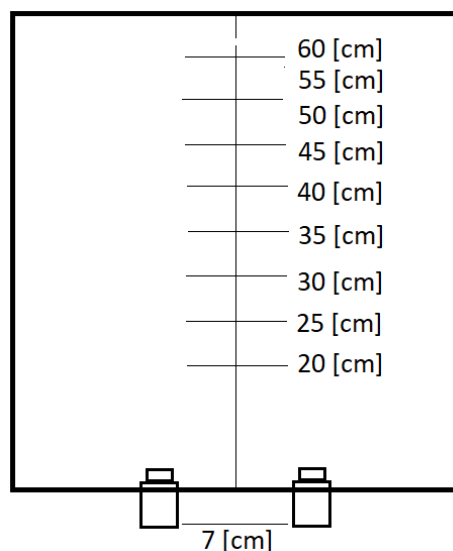


Mesa experimental usada para obtener la distancia hacia el objeto. Elaborado por: Cristhian

3.5 Ecuación para calcular la distancia entre usuario y objeto

El análisis para encontrar la ecuación que proporciona la distancia desde el usuario hasta el objeto se realizó de forma experimental, para ello se implementó una mesa de pruebas donde se fijó las dos cámaras sobre un fondo blanco que tenía dibujado medidas en centímetros en la mitad de la distancia de separación de las cámaras, como se observa en la Figura 3.9.

Figura 3.10. Mesa experimental para obtener la distancia hacia el objeto



Mesa experimental usada para obtener la distancia hacia el objeto. Elaborado por: Cristhian Gutiérrez y Hedian Herrera

Con el algoritmo creado en el apartado anterior se obtuvo las coordenadas en el eje “x” de la cámara derecha y de la cámara izquierda. Con estos datos se obtuvo una distancia de separación entre las coordenadas del eje “x” de la izquierda y de la derecha cada diez centímetros de longitud en dirección hacia el objeto empezando desde 20 centímetros como se observa en la Figura 3.9. Con la medida entre las coordenadas de la cámara izquierda y derecha se construyó una tabla con los valores promedios a las distintas distancias de la mesa experimental, y se obtuvieron los datos que se muestran en la Tabla 3.1.

Tabla 3.1. Datos experimentales obtenidos para calcular la distancia

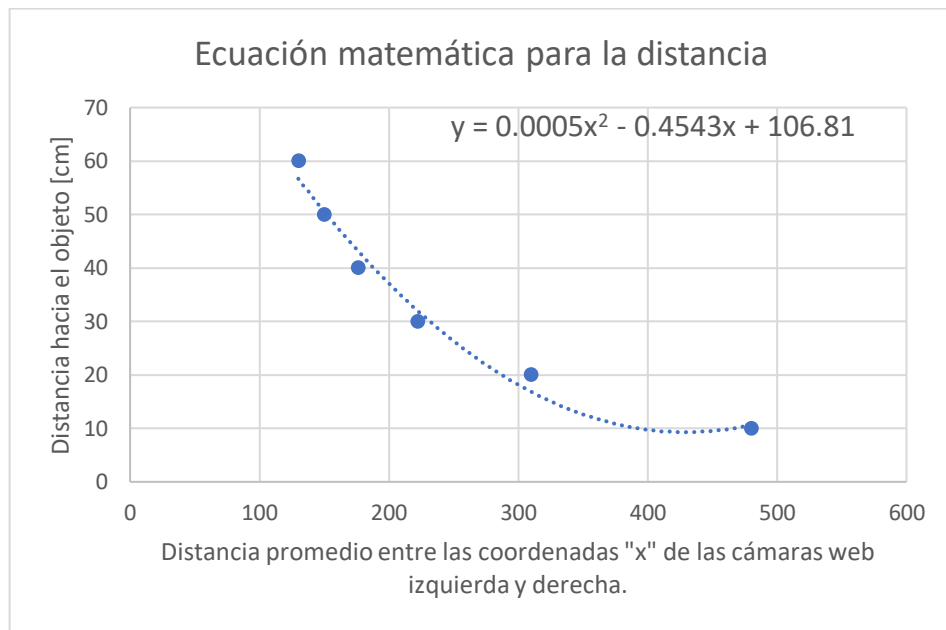
Distancia hacia el objeto [cm]	Distancia promedio entre cámaras web [píxeles]
10	480
20	310
30	222
40	176
50	150
60	130

Distancias en píxeles obtenidas a partir de la coordenada en el eje “x” de la cámara derecha e izquierda y la longitud de la mesa. Elaborado por: Cristhian Gutiérrez y Hedian Herrera

Con los datos obtenidos experimentalmente en la mesa de pruebas se encontró una Ecuación 3.1 con la ayuda de EXCEL (ver sección 2.10.2) que se acerca al comportamiento de dichos valores de la mejor manera como se puede observar en la Figura 3.10. Con dicha ecuación se pudo encontrar la distancia que existe entre las cámaras web y el objeto en múltiplos de cinco como se explica en la sección anterior.

$$D = 0.0005x^2 - 0.4543x + 106.81 \quad \text{Ec. (3.1)}$$

Figura 3.11. Función calculada con EXCEL



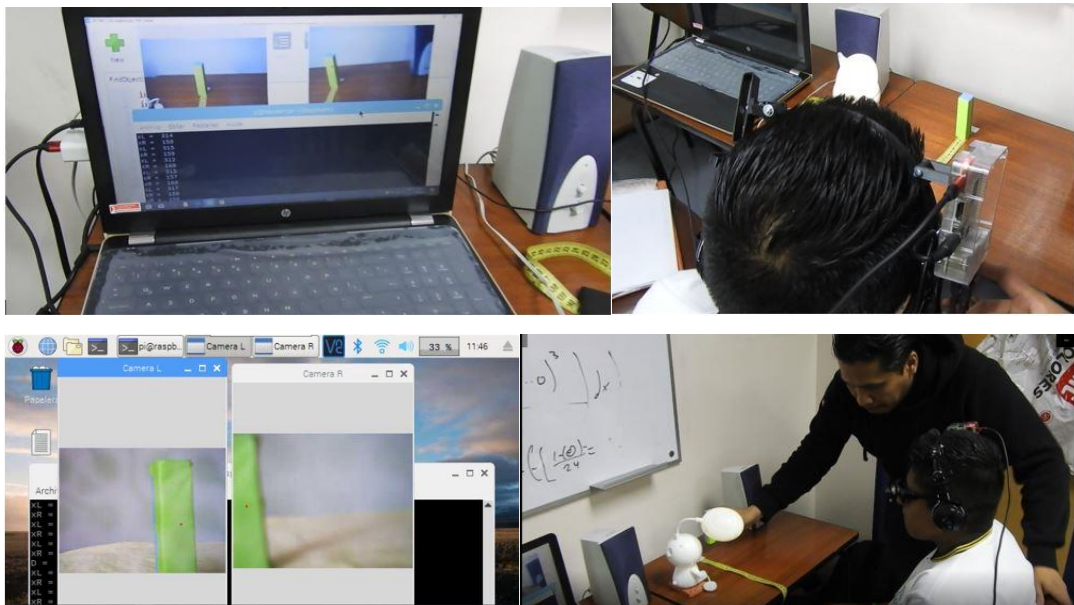
Función matemática para calcular la distancia encontrada con la ayuda de EXCEL. Elaborado por: Cristhian Gutiérrez y Hedian Herrera

CAPÍTULO 4

PRUEBAS Y RESULTADOS

En este capítulo se presentan los resultados obtenidos con el prototipo en pruebas realizadas con estudiantes voluntarios con discapacidad visual del colegio “Leonardo Ponce Pozo”. Se exponen además los comentarios vertidos de los estudiantes acerca de su experiencia con el prototipo y algunas recomendaciones que hicieron para mejorar el mismo. Las pruebas que se mostrarán a continuación arrojaron una idea clara del funcionamiento del prototipo y validaron su trabajo con un nivel de aceptación entre el 70% de acuerdo a los datos obtenidos con los voluntarios que probaron el mismo. En la Figura 4.1 se puede observar uno de los momentos en que se evaluó el funcionamiento del prototipo.

Figura 4.1. Fotografías del momento de evaluación del prototipo



Momento de evaluación del prototipo en el guiado para la recolección de un objeto marcado sobre una mesa con voluntarios con discapacidad visual. Elaborado por: Cristhian Gutiérrez y Hedian Herrera

4.1 Pruebas de corrección del ángulo de desviación

En esta sección se realizaron las primeras pruebas de funcionamiento colocando el objeto a distintos ángulos en la mesa para verificar el algoritmo de guiado en la corrección del ángulo de desviación. Se utilizaron dos audífonos, uno derecho y uno izquierdo para emitir un sonido en intervalos de tiempo de un segundo indicando la

dirección en la que el usuario debe mover la cabeza de forma horizontal al escuchar el audio en uno de los audífonos, es decir, si el sonido se escucha en el audífono izquierdo el usuario debería mover su cabeza de forma horizontal hacia la dirección del audio. Una vez que el usuario centra el objeto justo delante de su rostro se deja de emitir los sonidos en intervalos de tiempo y se informa con una voz digital la distancia que existe entre las cámaras y el objeto. En las pruebas realizadas se observó que era necesario realizar movimientos horizontales con la cabeza de forma lenta para dar tiempo al algoritmo en el reconocimiento de imágenes y en la emisión de las instrucciones sonoras.

Tabla 4.1. Pruebas de corrección del ángulo de desviación

Tiempo de corrección del ángulo de desviación con audífonos			
Pruebas	Voluntario 1 [s]	Voluntario 2 [s]	Voluntario 3 [s]
1	8	10	8
2	5	8	8
3	4	6	7
4	3	4	5
5	4	4	4
6	4	5	4
7	3	3	3
8	3	3	5
9	4	4	6
10	4	2	3

Tiempo de corrección del ángulo de desviación con audífonos. Elaborado por: Cristhian Gutiérrez y Hedian Herrera

Luego de corroborar el trabajo conjunto de las cámaras y los audífonos se procedió a las pruebas con las personas voluntarias con discapacidad visual. Para realizar estas pruebas se utilizó audífonos y parlantes externos de computadora. Se escogió una posición determinada para colocar el objeto y realizar 10 intentos consecutivos de corrección de ángulo. Además, se dio unas breves instrucciones a los participantes para que se familiaricen con el prototipo y conozcan su manera de operar.

Luego de la breve introducción se hicieron las primeras pruebas con audífonos y se tomó el tiempo que los usuarios se demoraban en corregir el ángulo de desviación,

los resultados arrojaron los datos que se muestran en la Tabla 4.1. En dichos resultados se observa que las primeras pruebas presentan un tiempo de retardo más elevado en los primeros intentos al momento de corregir el ángulo de desviación del usuario con respecto al objeto, y se notó que fue debido a que al principio fue difícil adaptarse a la manera de trabajar del prototipo, esto a causa del retardo de tiempo en el procesamiento de imágenes para devolver una respuesta sonora, lo que provocaba que el usuario se pase del punto de centrado. En las pruebas posteriores los tiempos en la corrección fueron menores en comparación con los primeros intentos, esto debido a que tras los primeros intentos los usuarios se adaptaban al funcionamiento del prototipo. Con los datos obtenidos en esta prueba se calculó un tiempo promedio de 4.8 segundos que el usuario se demorará en centrar el objeto justo en frente de las cámaras.

Tabla 4.2. Pruebas de corrección del ángulo de desviación

Tiempo de corrección del ángulo de desviación con parlantes			
Pruebas	Voluntario 1 [s]	Voluntario 2 [s]	Voluntario 3 [s]
1	5	4	5
2	4	5	4
3	3	3	4
4	4	3	3
5	3	3	3
6	3	4	3
7	3	4	4
8	3	3	4
9	4	3	3
10	4	3	3

Tiempo de corrección del ángulo de desviación con parlantes externos. Elaborado por: Cristhian Gutiérrez y Hedian Herrera

Se debe añadir que al momento de las pruebas fue necesario ubicar el objeto dentro del rango de visión de las cámaras para que el sonido empiece a dar las instrucciones del caso, esto último se observó debido a que en varias ocasiones el objeto quedaba fuera del rango de visión de las cámaras y el usuario advertía que no escuchaba ninguna instrucción o sonido que le pueda guiar. Con esto último, se notó que el prototipo necesitaba un mantenimiento correctivo para superar dichos eventos inesperados.

En las pruebas también se observó que fue necesario advertir de forma verbal a la persona que el objeto se encuentra por encima o por debajo de la mitad de la cámara. Detalle que se observó como un aporte para mejorar el prototipo para que pueda advertir de forma audible al usuario el ángulo de desviación de forma vertical.

De la misma manera se hicieron las pruebas con audífonos y se tomó el tiempo que los usuarios se demoraban en corregir el ángulo de desviación, los resultados obtenidos se muestran en la Tabla 4.2. En dichos resultados se observa que las primeras pruebas ya no presentan un tiempo de retardo más elevado en los primeros intentos al momento de corregir el ángulo de desviación del usuario con respecto al objeto, y se observó que fue producto del entrenamiento con los audífonos. Ya no afecto mucho el retardo de tiempo en el procesamiento de imágenes para devolver una respuesta sonora, y el usuario se adaptó a la forma de trabajar del prototipo. Además, los voluntarios que probaron el prototipo mencionaron que los parlantes externos les permitían distinguir de mejor manera el sonido proveniente de derecha o izquierda.

4.2 Pruebas de cálculo de la distancia hasta el objeto

En este apartado se muestran las pruebas realizadas para verificar el funcionamiento de la Ecuación 3.1 la cual fue encontrada en el apartado 3.4. En las pruebas realizadas se tomó medidas en centímetros proporcionados por la mesa adaptada para tal efecto en la sección mencionada en el párrafo anterior. Los resultados de los diez intentos realizados se exponen en la Tabla 4.3.

Tabla 4.3. Pruebas de medición de la distancia hacia el objeto.

Pruebas	Distancia medida [cm]	Distancia calculada [cm]
1	15	19
2	20	21
3	25	26
4	30	30
5	35	35
6	40	40
7	45	45
8	50	51
9	55	56
10	60	58

Medidas tomadas en centímetros para verificar el funcionamiento de la Ecuación 3.1. Elaborado por:
Cristhian Gutiérrez y Hedian Herrera

De acuerdo con los datos obtenidos en la Tabla 4.3 se puede constatar el funcionamiento adecuado de la Ecuación con un error en medida de un centímetro, exceptuando las medidas menores a 20 centímetros y cercanas a 60 centímetros. Debido a que el objeto que se va a buscar en la mesa de pruebas va a estar en una distancia entre 30 centímetros y 45 centímetros de distancia la Ecuación 3.1 es aceptable para las pruebas que se presentan a continuación.

Para realizar las pruebas de distancia con los voluntarios con discapacidad visual se colocó el objeto a una distancia determinada en centímetros y se realizaron 10 pruebas consecutivas. En cada intento se midió la longitud en centímetros desde las gafas hasta el objeto con la ayuda de una cinta métrica. Los resultados obtenidos se pueden observar en la Tabla 4.4.

De acuerdo con los resultados de la Tabla 4.4 se puede observar la fiabilidad que presenta el prototipo a la hora de calcular una distancia utilizando un sistema de visión estéreo. Son pocas las ocasiones en que la distancia diverge bastante de la realidad y esto se debe a varios factores, como por ejemplo los errores cometidos en mediciones de la distancia entre las cámaras y el objeto que debido al movimiento del usuario a veces variaba.

Tabla 4.4. Pruebas de medición de la distancia hacia el objeto

Pruebas	Distancia de prueba [cm]	Voluntario 1		Voluntario 2		Voluntario 3	
		Distancia calculada [cm]	Distancia exacta [cm]	Distancia calculada [cm]	Distancia exacta [cm]	Distancia calculada [cm]	Distancia exacta [cm]
1	30	35	33	30	30	35	34
2	30	30	32	30	32	35	33
3	30	30	28	35	33	30	31
4	35	35	34	40	38	35	35
5	35	35	36	35	37	35	37
6	35	35	35	40	39	40	38
7	35	40	38	35	34	35	35
8	40	40	40	45	44	40	39
9	40	45	43	40	39	40	42
10	40	40	39	40	42	45	43

Medidas tomadas en centímetros para verificar el funcionamiento del prototipo en el cálculo de la distancia hacia el objeto. Elaborado por: Cristhian Gutiérrez y Hedian Herrera

Otros efectos son la luz del medio ambiente que influye al momento de calcular la distancia de separación en el eje “x” de la cámara izquierda con respecto a la derecha. Sin embargo, pese a todos estos factores externos las pruebas se realizaron sin mayor inconveniente arrojando como resultado una fiabilidad del cálculo de la distancia con un error relativo de 3.7% de la Ecuación 4.4

4.3 Pruebas en la recolección del objeto

Para probar si el usuario pudo tomar el objeto con la ayuda del prototipo se anotó el número de aciertos al tomar el objeto en las pruebas mostradas en la Tabla 4.4.

Tabla 4.5. Pruebas de recolección del objeto.

Pruebas	Distancia establecida [cm]	Voluntario 1	Voluntario 2	Voluntario 3
		Tomó el objeto [si/no]	Tomó el objeto [si/no]	Tomó el objeto [si/no]
1	30	No	No	No
2	30	Si	No	Si
3	30	Si	Si	Si
4	35	No	Si	Si
5	35	No	Si	No
6	35	Si	No	Si
7	35	Si	Si	Si
8	40	Si	Si	Si
9	40	Si	Si	Si
10	40	Si	No	Si

Aciertos conseguidos con los voluntarios en la recolección de un objeto marcado sobre una mesa.

Elaborado por: Cristhian Gutiérrez y Hedian Herrera

Con los resultados obtenidos en la Tabla 4.5 se pudo calcular el porcentaje de efectividad que tiene el prototipo para guiar a una persona discapacitada en la recolección de objetos marcados sobre una mesa de trabajo. En promedio, de todos los intentos y de los tres participantes se consiguió un porcentaje de un 70% de efectividad en la recolección de objetos presentados en la Ecuación 4.5.

En los intentos realizados se pudo observar que la longitud proporcionada en base al algoritmo creado en la sección 3.3 que dicta la distancia redondeada a múltiplos de 5 centímetros no fue mayor problema para dar una idea de la distancia que el usuario tenía que estirar la mano para alcanzar el objeto en la mesa

4.4 Resumen de los resultados obtenidos

En este apartado se muestra un resumen de los resultados obtenidos en las pruebas de corrección del ángulo de desviación, cálculo de la distancia desde el objeto hacia las cámaras web y el número de aciertos en la recolección del objeto desde la mesa.

Tabla 4.6. Resumen de los resultados obtenidos

Pruebas	Tiempo [s]	Error Relativo [%]	Aciertos [#]
1	8.7	3,0	0
2	7.0	6,2	2
3	5.7	5,5	3
4	4.0	2,7	2
5	4.0	4,5	1
6	4.3	2,6	2
7	3.0	2,7	3
8	3.7	1,6	3
9	4.7	4,0	3
10	3.0	4,0	2
Promedio general	4.8 [s]	3,7 [%]	70 [%]

Resumen de los promedios obtenidos en las pruebas de corrección del ángulo de desviación, cálculo de distancia y recolección del objeto con los tres voluntarios. Elaborado por: Cristhian Gutiérrez y Hedian Herrera

Para obtener el tiempo promedio de los tres voluntarios en la primera prueba se utilizó la Ecuación 4.1. De la misma manera se encontró los promedios en cada una de las diez pruebas y se colocó los resultados en la columna 1 de la Tabla 4.6.

Cálculo del tiempo promedio de los tres voluntarios en la prueba 1:

$$Er = \frac{\sum_{i=1}^3 \text{Tiempo}(i)}{\text{N}^\circ \text{ de voluntarios}} = \frac{8+10+8}{3} = \frac{26}{3} = 8.7 [s] \quad \text{Ec. (4.1)}$$

Utilizando las Ecuaciones 4.2 y 4.3 se encontró el error relativo de la distancia calculada con el voluntario 1 en la primera prueba mediante las Ecuaciones 4.4 y 4.5. De la misma manera se encontró el error relativo en la primera prueba del voluntario 2 y 3. Al final, se sumó los tres errores relativos y se obtuvo un promedio para la prueba 1 de los tres voluntarios mediante la Ecuación 4.6. Con el mismo

procedimiento se encontró los promedios de los errores relativos en las siguientes pruebas y se anotaron en la columna 2 de la Tabla 4.6.

Error absoluto:

$$Ea = |\text{valor medido} - \text{valor exacto}| \quad \text{Ec. (4.2)}$$

Error relativo:

$$Er = \frac{\text{error absoluto}}{\text{valor exacto}} \times 100\% \quad \text{Ec. (4.3)}$$

Cálculo del error relativo del voluntario 1:

$$Ea = |\text{valor medido} - \text{valor exacto}| = |35 - 33| = 2 \text{ [cm]} \quad \text{Ec. (4.4)}$$

$$Er = \frac{\text{error absoluto}}{\text{valor exacto}} \times 100\% = \frac{2}{33} = 6.1 \text{ [%]} \quad \text{Ec. (4.5)}$$

Cálculo del error relativo promedio de los tres voluntarios en la prueba 1:

$$Er = \frac{\sum_{i=1}^3 \text{Error relativo}(i)}{N^{\circ} \text{ de voluntarios}} = \frac{6.1+0.0+2.9}{3} = 3 \text{ [%]} \quad \text{Ec. (4.6)}$$

Finalmente, para conseguir los promedios generales del tiempo en la corrección del ángulo de desviación de la persona con respecto al objeto, del error relativo de la distancia calculada desde el usuario hasta el objeto, y del número de aciertos que se obtuvieron en la recolección del objeto se usaron las Ecuaciones 4.7, 4.8 y 4.9 respectivamente.

Cálculo del promedio de tiempo general:

$$Er = \frac{\sum_{i=1}^{10} \text{Tiempo Promedio}(i)}{N^{\circ} \text{ de pruebas}} = \frac{48 \text{ [s]}}{10} = 4.8 \text{ [s]} \quad \text{Ec. (4.7)}$$

Cálculo del promedio de error relativo general:

$$Er = \frac{\sum_{i=1}^{10} \text{Error relativo promedio}(i)}{N^{\circ} \text{ de pruebas}} = \frac{52 \text{ [%]}}{10} = 5.2 \text{ [%]} \quad \text{Ec. (4.8)}$$

Cálculo del promedio de porcentaje de efectividad general:

$$Er = \frac{\sum_{i=1}^{10} \text{Intentos Exitosos } (i)}{N^{\circ} \text{ total de pruebas}} = \frac{21}{30} = 4.8 [s] \quad \text{Ec. (4.9)}$$

Al final de las pruebas los voluntarios dejaron algunas opiniones sobre el prototipo que se mencionan a continuación y que se tendrá presente en una siguiente fase de construcción con mejoras del prototipo:

- La referencia del objeto ayuda mucho a ubicarlo en la mesa.
- Mencionaron que los audífonos no son tan claros como los parlantes pero que se puede mejorar el guiado cambiando los simples sonidos por voces digitales.
- Los sonidos no parecieron molestarles y no presentaron quejas al respecto.
- Al inicio es difícil saber la distancia a la que se debe estirar el brazo para recoger el objeto.
- En general dieron buenas opiniones del funcionamiento del prototipo.

CONCLUSIONES

Se consiguió guiar a una persona discapacitada visualmente en la recolección de objetos mediante visión artificial utilizando Python, OpenCV, una Raspberry Pi, dos cámaras web, audífonos y una batería de 2500 amperios hora, implementado todo en un prototipo portable que calcula el ángulo de desviación y la distancia del objeto con respecto del usuario. Todo esto sin la necesidad de computadoras de gran tamaño y con procesadores más potentes.

De acuerdo con el estado del arte de la sección 2.1 se observó que las investigaciones que se realizan en el área utilizan dos cámaras para obtener la distancia desde el usuario hacia objetos utilizando diferentes técnicas de visión artificial.

Se observó experimentalmente en la sección 3.4 que se puede corregir el ángulo de desviación sin necesidad de calcularlo, ya que si se proporcionan rangos de coordenadas en el eje "X" se puede ubicar el objeto al frente de las dos cámaras fijas emitiendo sonidos en los audífonos izquierdo o derecho para tal efecto como se explica en la lógica del algoritmo creado de la sección 3.3.

Con el algoritmo elaborado en este trabajo se pudo corregir el ángulo de desviación de las cámaras con respecto al objeto y encontrar la distancia desde el usuario hacia el objeto. El algoritmo permitió realizar pruebas de búsqueda del objeto sobre una mesa de pruebas y no presentaron errores al momento de ponerlo a prueba.

En la sección 4.1 se presentaron las pruebas con tres voluntarios con discapacidad visual en la corrección del ángulo de desviación. Para dichas pruebas se midió el tiempo que tardaban los usuarios en corregir la desviación de las cámaras con respecto al objeto y dejarlo centrado justo al frente de las cámaras web. Con los datos obtenidos es esta prueba se calculó un tiempo promedio de 4.8 segundos que el usuario se demorará en centrar el objeto justo en frente de las cámaras, y este tiempo comparado con el tiempo que tardan utilizando la forma tradicional (sin guía de ningún tipo) para buscar un objeto se considera aceptable.

En las pruebas de la sección 4.2 en el cálculo de la distancia utilizando la Ecuación 3.1 se obtuvieron datos que mostraban una fiabilidad aceptable de 5.2 %. Hecho que

se corroboró con los voluntarios que usaron el prototipo, ya que con una estimación aproximada de la distancia al objeto era más sencillo llegar a él y sujetarlo. Se mostraron algunos errores de sujeción en ciertos intentos, pero esto se debió a que el usuario desenfocaba el objeto provocando que la voz generada con la distancia se pierda. Se considera que este tipo de errores se puede corregir con un entrenamiento adecuado con el prototipo.

Con el primer avance mostrado en este trabajo se consiguió buenos comentarios por parte de las personas voluntarias, y también se tomaron en cuenta algunas de las recomendaciones hechas, pero al final el trabajo llamó bastante a atención de los usuarios que lo probaron.

RECOMENDACIONES

Al momento de realizar las pruebas de búsqueda del objeto se recomienda ubicarlo dentro de una zona de trabajo predefinida, ya que en las pruebas de la sección 4 se observó en varias ocasiones que el objeto quedaba fuera del rango de visión de las cámaras y el usuario no escuchaba ninguna instrucción o sonido que le pueda guiar.

Después de finalizar las pruebas de guiado con los voluntarios se recibió recomendaciones de cambiar el sonido simple de intervalos de tiempo que se utiliza para corregir el ángulo de desviación y reemplazarlo por comandos de voz para que el guiado sea más fácil de seguir por parte de los usuarios. Añadido a esto, se recomendó colocar un mensaje de voz que informe del hecho que no se detecta objetos en las cámaras y que el usuario debe buscarlo moviendo despacio la cabeza de forma continua hacia los costados en forma horizontal y vertical.

De la misma manera, los usuarios recomendaron añadir un mensaje de voz que advierta sobre la desviación del ángulo de forma vertical, para corregirlo y ubicar el objeto justo en frente de las cámaras. Y además comentaron que les gustaría oír una voz que hable despacio y lentamente para una mejor comprensión de las instrucciones emitidas.

Por otro lado, es necesario iluminar de forma adecuada el ambiente donde se utilizará el prototipo, ya que la poca luz influye en la detección de los valores establecidos para el color verde.

BIBLIOGRAFÍA

- Abellán, M. Á. (27 de OCTUBRE de 2017). *PROGRAMO ERGO SUM*. Obtenido de <https://www.programoergosum.com/cursos-online/raspberry-pi/244-iniciacion-a-python-en-raspberry-pi/introduccion>
- Aguiar, E. (2016). *Capa 8 - Seguridad informática para no informáticos*. Mexico: Palibrio.
- akirasan. (30 de junio de 2016). *akirasan*. Obtenido de <http://akirasan.net/detectar-el-movimiento-con-opencv/>
- Alejandro, V. P. (2009). *Sistemas de visión artificial*.
- Alfonso Hernandez. (20 de 11 de 2016). *Sensor de Distancia*. Obtenido de www.sensores-de-medida.es
- Aparicio, A. (29 de Marzo de 2018). *ExcelAvanzado.com*. Obtenido de <https://www.excelavanzado.com/2018/03/regresion-polinomica.html>
- Autodesk. (2019). *Autodesk*. Obtenido de <https://latinoamerica.autodesk.com/solutions/3d-printing>
- Autodesk. (2019). *Autodesk*. Obtenido de <https://knowledge.autodesk.com/es/support/inventor-lt/learn-explore/caas/CloudHelp/cloudhelp/2019/ESP/InventorLT-Help/files/GUID-16DF561D-C44E-438D-B6F4-41AA8D81C0B1-htm.html>
- Autodesk. (2019). *Autodesk*. Obtenido de <https://knowledge.autodesk.com/es/support/inventor-products/learn-explore/caas/CloudHelp/cloudhelp/2019/ESP/Inventor-Help/files/GUID-72511A09-F17C-4A93-8274-B953FA0139B6-htm.html>
- Barrero, G. (2018). *Predicción de la calidad de software desarrollado en IBM RPG usando Deep Learning*. Lima.
- Becerra, J. (28 de 1 de 2016). *HUNABSYS RESEARCH / DEVELOPMENT*. Recuperado el 15 de 06 de 2019, de <http://blog.hunabsys.com/introduccion-a-la-programacion-en-raspberrypi/>
- Camacho, E. J. (11 de mayo de 2009). *Medición de distancias por medio de procesamiento de imágenes y triangulación, haciendo uso de cámaras de video*. Cholula, Puebla, Mexico.
- COMPUDISC. (2013). *COMPUDISC*. Obtenido de <http://www.compudisc.net/camaras-web/camara-web-havit-hv-n5082-detail>

- fnac. (2019). *fnac*. Obtenido de <https://www.fnac.es/mp2532164/Modem-3g-E1750-1750-USB-Hsdpa-Libre-Android-Compatible-Telefono-Pincho-Huawei/w-4>
- Folch, S. (27 de mayo de 2010). *Hablamos de comunicacion multimedia*. Obtenido de <https://www.comunicacion-multimedia.info/2010/05/modos-o-modelos-de-color-hsb-o-hsv-y.html>
- FotoNostra. (s.f.). *FotoNostra*. Obtenido de <https://www.fotonostra.com/grafico/rgb.htm>
- Garcia Posada, M. (2015). *Diseño de prototipo de recogida automatizada de bolos mediante brazo robótico y vision artificial*. Oviedo: Universidad de Oviedo.
- García, V. S. (2019). *Booleanbite*. Obtenido de <https://booleanbite.com/web/opencv-y-raspberry-pi/>
- GI4r3. (25 de Abril de 2014). *ROBOLOGS*. Obtenido de <https://robologs.net/2014/04/25/instalar-opencv-en-raspberry-pi-2/>
- GI4r3. (26 de julio de 2015). *Robologs*. Obtenido de <https://robologs.net/2015/07/26/como-filtrar-el-ruido-de-una-mascara-con-opencv/>
- Golden, R. (2015). *Rasberri Pi Networking Cookbook*. Birmingham: PACKT PUBLISHING.
- Gracia, L. (9 de octubre de 2013). *UN POCO DE JAVA Y +*. Obtenido de <https://unpocodejava.com/2013/10/09/que-es-opencv/>
- Hernández, L. d. (9 de octubre de 2016). *programarfacil*. Obtenido de https://programarfacil.com/blog/vision-artificial/deteccion-de-movimiento-con-opencv-python/#Substraccion_de_fondo
- Informatica, T. &. (22 de Julio de 2019). *La Cámara web. Como comprar una webcam. Aplicaciones*. Obtenido de <https://webcache.googleusercontent.com/search?q=cache:olmTi4qEjuwJ:https://tecnologia-informatica.com/camara-web-comprar-webcam/+&cd=11&hl=es-419&ct=clnk&gl=ec>
- Jiménes Calle, D. V. (1 de 5 de 2012). *Vision Artificial*. Obtenido de <http://dspace.ups.edu.ec/handle/123456789/2944>
- Jimmy Rolando Molina Rios, N. M. (2016). Evaluación de los frameworks en el desarrollo de aplicaciones web con python. *Revista latinoamericana de Ingenieria de Software*, 202.

KUBII. (2019). *KUBII*. Obtenido de <https://www.kubii.es/tarjetas-de-extension/1922-modulo-gps-para-raspberry-pi-kubii-3272496007888.html>

Lizarazo Zambrano, J. A., & Ramos Velandia, M. A. (2016). Visión artificial y comunicación en robots cooperativos omnidireccionales. *Ingeciencia*, 5-11.

Marco Felipe Lopez. (22 de 2 de 2017). *Microcomputadota*. Obtenido de www.alegsa.com.ar

Merino, J. P. (2016). *Definicion* . Obtenido de <https://definicion.de/rgb/>

OMF. (21 de junio de 2018). *custom 3D*. Obtenido de <https://www.custom3d.es/principales-formatos-caracteristicas-impresion-3d>

omicron. (13 de 05 de 2016). *El Español*. Obtenido de Guía de accesorios para Raspberry Pi: Los complementos imprescindibles: <https://omicron.elespanol.com/2016/05/accesorios-para-raspberry-pi/>

Peña, R. (2015). *aenui.net* . Obtenido de <http://aenui.net/ojs/index.php?journal=revision&page=article&op=viewArticle&path%5B%5D=199&path%5B%5D=329>

Pérez, A. F. (2013). *Control de dispositivos con android* . Barcelona.

Python, R. (21 de septiembre de 2014). *recursos Python*. Obtenido de <https://recursospython.com/guias-y-manuales/pyo-procesamiento-generacion-audio/>

Rea Escalante. (5 de 2014). *Sensor ultrasonico*. Obtenido de <http://dspace.ups.edu.ec/handle/123456789/10410>

Rodrigo Alonso, R. V. (s.f.). *HARDZONE*. Obtenido de <https://hardzone.es/reviews/perifericos/analisis-raspberry-pi-3-modelo-b/>

Rosales Romero, C. E. (2017). *Prototipo de detección de expresiones corporales mediante vision artificial para mejorar la comunicacion con niños que tienen parálisis cerebral infantil*. Loja: Universidad Nacional de Loja.

Rossum, G. v. (31 de agosto de 2009). *Tutorial Python version 2.6.2*. Obtenido de <http://ralsina.me/static/tutorial-8.pdf>

