



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



ETSI Aeroespacial y Diseño Industrial

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería Aeroespacial
y Diseño Industrial

Hacia un nuevo ratón de mirada de
bajo coste para comunicación
aumentativa y alternativa

Trabajo Fin de Grado

Grado en Ingeniería Electrónica Industrial y Automática

AUTOR/A: López Almansa, Sergio

Tutor/a: García Sánchez, Tania María

Cotutor/a externo: Castillo García, Fernando Jose

CURSO ACADÉMICO: 2023/2024

TRABAJO FIN DE GRADO

Hacia un nuevo ratón de mirada de bajo coste para comunicación aumentativa y alternativa

Realizado por
Sergio López Almansa

Para la obtención del título de
Grado en Ingeniería Electrónica Industrial y Automática

Dirigido por
D. Fernando José Castillo García
Dña. Tania María García Sanchez

En el departamento de
Automática y Robótica

Convocatoria de septiembre, curso 2023/24

Agradecimientos

En primer lugar, gracias a mis directores Dña. Tania María García Sanchez y D. Fernando José Castillo García, por darme la oportunidad de trabajar en este proyecto, por enseñarme tanto y por su ayuda.

También me gustaría mostrar mis agradecimientos a D. Juan Antonio Monsoriu Serra, director de mi escuela, por darme la oportunidad de trabajar en este proyecto y por la formación ofrecida durante mi etapa universitaria.

Gracias a todos y cada uno de los profesores que durante mi vida me han ayudado y animado a continuar formándome. Un enorme gracias a todos los compañeros y amigos que han dado inmensas alegrías en cada día de clase.

Por último, y no menos importante, gracias a mi familia por apoyarme y soportarme en todos los momentos, y por ayudarme a tomar todas las decisiones difíciles con sus sabios consejos.

Resumen

Los tratados internacionales, incluido el Pacto Internacional de Derechos Civiles y Políticos, reconocen la libertad de comunicación como un derecho humano. El Comité de Derechos Humanos de la ONU, en 2011, subraya la importancia de garantizar el acceso a los nuevos medios de comunicación.

Este Trabajo Fin de Grado busca promover este derecho para personas con dificultades de movilidad y comunicación verbal, con el respaldo de la Fundación Gema Canales, dedicada a mejorar la calidad de vida de las personas con parálisis cerebral.

En consecuencia, el proyecto se enfoca en la concepción de una aplicación en MATLAB destinada a integrarse en un dispositivo de seguimiento ocular. Su propósito radica en proporcionar a individuos con impedimentos motores una herramienta que les facilite la comunicación, al cuantificar la eficacia del movimiento ocular como medio de expresión.

La configuración del Trabajo Fin de Grado se desglosa en dos componentes principales: la aplicación implementada en MATLAB y una interfaz gráfica diseñada para rastrear con la máxima precisión el movimiento ocular del usuario que emplea el dispositivo de seguimiento ocular. Este desarrollo se fundamenta en los conocimientos obtenidos durante el grado en informática, visión artificial y procesamiento de imágenes.

El propósito del proyecto se centra en el desarrollo de un algoritmo en MATLAB que, mediante el empleo de una o dos cámaras, pueda equipararse con los dispositivos de seguimiento ocular disponibles en el mercado. Se enfatiza especialmente en la detección y análisis del parpadeo, así como en la captura y seguimiento de la posición y movimiento del globo ocular del usuario.

El objetivo principal de este Trabajo Fin de Grado es crear una primera versión de un dispositivo de seguimiento ocular de código abierto, que pueda ser mejorado gradualmente hasta cumplir con los estándares de rendimiento necesarios. Se busca adaptarlo para su uso en entornos de comunicación aumentativa y luego cederlo a la Fundación Gema Canales para que lo promueva y lo haga accesible a más personas.

Palabras clave: Ratón de mirada, movilidad reducida, código abierto, parálisis cerebral, interfaz de usuario, comunicación aumentativa y alternativa.

Abstrac

Several international treaties in force define freedom of communication as a human right, including the International Covenant on Civil and Political Rights, adopted by the United Nations General Assembly in 1966 and part of the Universal Declaration of Human Rights.

More recently, in 2011, the Human Rights Committee of the United Nations complements this treaty, highlighting that the different states must take the necessary measures to ensure access to individuals of the new ICT media.

This Final Degree Project is born with the idea of contributing to the fulfillment of this right for people with mobility challenges who present problems in their verbal communication. In particular, the main promoter is the Gema Canales Foundation, which was born in 2023 with the aim of improving the quality of life of people with cerebral palsy.

Many of these people require an augmentative and alternative communication system for their communication. Within this system, an essential element is the gaze mouse that allows, by means of the user's eye tracking, to act as a mouse to interact with the augmentative communication system.

At present, although they are scarce, there are some commercial models of gaze mice, which are used by users of augmentative communication systems. However, none of them is open and its price is too high to be acquired by the families of potential users.

The main objective of this Final Degree Project is to develop a first version of an open source gaze mouse that can be continuously improved in future actions until it reaches the required performance standards and makes it suitable for augmentative communication environment. This gaze mouse will be given to the Gema Canales Foundation so that it can act as a promoter of the same and make it accessible to as many people as possible.

Keywords: Gaze mouse, reduced mobility, cerebral palsy, augmentative and alternative communication.

Resum

Diversos tractats internacionals vigents definixen la llibertat de comunicació com un dret humà, entre ells, el Parc Internacional de Drets Civils i Polítics, adoptat per l'Assemblea General de les Nacions Unides en 1966 i que forma part de la Declaració Universal dels Dret Humans.

Més recentment, en 2011, el Comitè de Drets Humans de les Nacions Unitats complementa este tractat, ressaltant que els diferents estats han de prendre les mesures necessàries per a assegurar l'accés als particulars dels nous mitjans de comunicació TIC.

Este Treball fi de grau naix amb la idea de contribuir al compliment d'eixe dret per a persones amb reptes de mobilitat que presenten problemes en la seua comunicació verbal. En particular, el principal entre promotor és la Fundació Gema Canales, que va nàixer en 2023 amb l'objectiu de millorar la qualitat de vida de les persones amb paràlisi cerebral.

Moltes d'estes persones, requerixen d'un sistema de comunicació augmentativa i alternativa per a la seua comunicació. Dins d'este sistema un element primordial és el ratolí de mirada que permet, mitjançant el seguiment ocular de l'usuari, fer de ratolí per a interactuar amb el sistema de comunicació augmentativa.

En l'actualitat, encara que són escassos, existixen alguns models comercials de ratolins de mirada, que són emprats pels usuaris de sistema de comunicació augmentativa. No obstant això, cap d'ells és obert i el seu preu és molt elevat per a ser adquirit per les famílies dels potencials usuaris.

L'objectiu principal d'este Treball fi de grau és desenvolupar una primera versió de ratolí de mirada amb codi obert que puga ser millorat de manera contínua en futures actuacions fins a aconseguir els estàndards requerits de funcionament i fer-ho idoni per a entorn de comunicació augmentativa. Este ratolí de mirada serà cedit a la Fundació Gema Canales perquè puga actuar com a ens promotor del mateix i fer-ho accessible al major nombre de persones possibles.

Paraules clau: Ratolí de mirada, mobilitat reduïda, paràlisi cerebral, comunicació augmentativa i alternativa.

Índice general

1	Introducción	1
1.1.	Planteamiento del problema	1
1.2.	Objetivos del proyecto	1
1.3.	Metodología	2
1.4.	Medios empleados	2
2	Estado del arte	5
2.1.	Principios de la tecnología de seguimiento de mirada	5
2.2.	Historia y evolución del seguimiento de la mirada	11
2.3.	Aplicaciones del seguimiento de la mirada en diversos campos	13
2.4.	Requisitos y restricciones	16
3	Sistemas de medición ocular	17
3.1.	Electro-Oculografía(EOG)	19
3.2.	Electro-Oculografía(EOG)	19
3.3.	Videoculografía (VOG)	20
3.4.	Calibración de la cámara	33
4	Estudio de necesidades	36
4.1.	Identificación de necesidades	36
4.2.	Recursos disponibles	36
4.3.	Limitaciones y condicionantes	37
5	Desarrollo del Código	38
5.1.	Definición de criterios y fases	38
5.2.	Diseño, implementación y resultado de los algoritmos	39
5.2.1.	Primer script:	39
5.2.2.	Segundo script:	45
5.2.3.	Tercer script:	52
6	Pruebas y Resultados	59
6.1.	Estudio de los resultados obtenidos en ambos scripts	59
6.2.	Evaluación de mejoras y ventajas del nuevo seguidor de mirada	68
7	Conclusiones	69
7.1.	Conclusiones del proyecto	69
8	Relación e impacto con los ODS	70

9 Anexos	73
9.1. Anexo al Trabajo de Fin de Grado: Relación del trabajo con los Objetivos de Desarrollo Sostenible de la agenda 2030	73
9.2. Código fuente	73
Bibliografía	79

Índice de figuras

2.1. Líderes en el desarrollo de la tecnología de seguimiento visual. [1] . . .	5
2.2. Emile Javal, 1879 [2]	6
2.3. Edmund Burke Delabarre [3]	7
2.4. Foto-cronógrafo que opera basado en la reflexión corneal.	8
2.5. Aparato utilizado por Buswell, 1935 [4]	9
2.6. Un seguidor de mirada diseñado por H. Hartridge y L.C. Thompson [1]	10
2.7. Seguidor de mirada (Alfred L. Yarbus, 1967) [1]	10
2.8. Evolución seguidores de mirada.[5]	11
2.9. Movimiento ocular. [6]	12
2.10. Metodo de reflexión en la córnea.[7]	14
3.1. Enfoques ejemplares que utilizan técnicas de vídeo y escaneado de imágenes en oculometría.[8]	17
3.2. Ecuaciones momentos [8]	18
3.3. Representación de la posición de los electrodos[9]	19
3.4. Registro EOG de 20 sostenimientos excéntricos de la mirada. [10]	20
3.5. Metodo de Purkinje [11]	22
3.6. Aproximación 2D [11]	23
3.7. Principio basico de video oculografía [10]	24
3.8. Plantilla elíptica [11]	25
3.9. Modelo del ojo [11]	27
3.10. Modelo del ojo de Le Grand [11]	29
3.11. Modelo del ojo para la imagen de Purkinje [12]	31
3.12. Modelo de T. Nagamatsu, J. Kamahara y N. Tanaka [13]	33
3.13. Modelo de escena 3D [14]	34
5.1. Diagrama de bloques 1	40
5.2. Generación de la elipse	42
5.3. Cuenta Regresiva	43
5.4. Ubicación 1	44
5.5. Ubicación 2	44
5.6. Ubicación 3	45
5.7. Frame del vídeo generado	45
5.8. Diagrama de bloques 2. Primera parte	47
5.9. Diagrama de bloques 2. Segunda parte	48
5.10. Frame 81/100. Script 2. Primer usuario	49
5.11. Estadística de fallo en 100 frames usuario 1. Script 2	49
5.12. Frame 60/100. Script 2. Cuarto usuario	50
5.13. Estadística de fallo en 100 frames usuario 4. Script 2	51
5.14. Diagrama de bloques 3. Primera parte	53
5.15. Diagrama de bloques 3. Segunda parte	54
5.16. Frame 34 del vídeo. Script 3. Usuario 1	57

5.17. Estadísticas de fallo en 100 frames usuario 1. Script 3	58
6.1. Ejemplo vídeo Sergio. Script 2.	59
6.2. Gráfica de estadísticas ojo izquierdo Sergio	61
6.3. Ejemplo vídeo voluntaria. Script 2.	62
6.4. Gráfica ojo derecho voluntaria. Script 2.	62
6.5. Extracto de detección ocular voluntaria. Ojo izquierdo.	63
6.6. Ejemplo vídeo Sergio. Script 3.	64
6.7. Ejemplo de extracto Sergio. Ojo derecho. Script 3.	65
6.8. Gráfica ojo derecho Sergio. Script 3	65
6.9. Ejemplo vídeo voluntaria. Script 3.	66
6.10. Ejemplo de extracto voluntaria.Ojo derecho. Script 3.	67
6.11. Gráfica ojo derecho voluntaria. Script 3	67
8.1. Objetivos de Desarrollo Sostenible [15]	70

Índice de tablas

9.1. Grado de relación del trabajo con los Objetivos de Desarrollo Sostenible (ODS).	73
--	----

Índice de extractos de código

5.1. Configuración de la cámara y variables	39
5.2. Configuración y creación de la ventana	41
5.3. Configuración y generación de la elipse	41
5.4. Configuración de la cuenta regresiva	42
5.5. Configuración de la trayectoria del círculo rojo	43
5.6. Inicialización de objetos y variables	46
5.7. Configuración del bucle y detección de caras	46
5.8. Detección de ojos	46
5.9. Ojo izquierdo	49
5.10. Ojo izquierdo	50
5.11. Ojo derecho	51
5.12. Anotaciones y cálculo de fallos	52
5.13. Inicialización de objetos y variables	55
5.14. Configuración del bucle y detección de caras	55
5.15. Detección y procesamiento de los ojos	55
5.16. Visualización de subgráficos	57
5.17. Actualización del título y pausa	58
6.1. Estadísticas Sergio Vídeo. Script 2.	60
6.2. Estadísticas voluntaria	61
6.3. Estadísticas Sergio vídeo. Script 3.	64
6.4. Estadísticas voluntaria	66
9.1. Código Completo Primer Script	73
9.2. Código Completo Segundo Script	75
9.3. Código Completo Tercer Script	77

1. Introducción

1.1. Planteamiento del problema

En el ámbito de la ingeniería, en especial la biomédica, y la accesibilidad tecnológica hay distintos tipos de desafíos, pero entre los más significativos se encuentra el diseño de sistemas que mitiguen las dificultades enfrentadas diariamente por los ciudadanos con enfermedades psicomotrices. Estos trastornos, entre los cuales se incluyen la parálisis cerebral o enfermedades como el ELA, afectan a la capacidad de una persona para coordinar movimientos de forma voluntaria, es decir, reducen tanto las posibilidades para comunicarse como para el uso de tecnologías convencionales. Además, existen casos en los cuales se incorporan discapacidades intelectuales asociadas complicando aún más la capacidad de procesamiento y producción del lenguaje.

Este tipo de deficiencias, tanto visuales como auditivas, impiden la interacción con dispositivos electrónicos y software no adaptados, así como teclados, ratones, pantallas táctiles o interfaces de usuario no accesibles, ya que muchas aplicaciones y sistemas operativos se han diseñado sin considerar a las personas que sufren de las enfermedades psicomotrices. Una solución innovadora y prometedora es el uso de dispositivos de seguimiento ocular, comúnmente denominados ratones de mirada". Estos dispositivos permiten a los usuarios controlar una computadora mediante el seguimiento del movimiento de sus ojos. La implementación de un ratón de mirada no solo democratiza el acceso a la tecnología, sino que también amplía las posibilidades de comunicación y participación social para este porcentaje de la población.

1.2. Objetivos del proyecto

En consecuencia, de este significativo problema, el principal objetivo del proyecto no es otro que ayudar a los afectados por todas las enfermedades que dificultan las acciones cotidianas además de mejorar su día a día con el dispositivo llamado ratón de mirada promoviendo su inclusión social y digital. Entre los dos objetivos principales del proyecto, se encuentra por un lado un código capaz de cuantificar como de precisos y óptimos son los seguidores oculares del mercado en la actualidad. Por otro lado, se trata de un código en el cual se crea un ratón de mirada propio y desde cero con el objetivo de mejorar los actuales. Este proyecto, a su vez, se dejará como un código abierto al finalizar el Trabajo Fin de Grado, para que pueda ser mejorado.

No obstante, el principal objetivo es dar la oportunidad a todas las familias con integrantes con cualquier tipo de enfermedad psicomotriz de una forma mas económica y voluntaria, ya que este proyecto se realiza sin animo de lucro y con la

intención de donarlo a la Fundación Gema Canales, la cual es la promotora de este ratón de mirada.

1.3. Metodología

Se ha diseñado una interfaz de usuario para que sea intuitiva y fácil de calibrar para usarla diariamente, es crucial que el sistema sea adaptable a las necesidades individuales de cada usuario, permitiendo ajustes personalizados. Además, se ha tenido en cuenta la compatibilidad con los sistemas operativos comunes como Windows, MacOS o Linux, incluso con distintas aplicaciones de comunicación aumentativa y alternativa.

Todo el Trabajo Fin de Grado se ha realizado en la aplicación de MATLAB, la cual se ha estudiado a lo largo del grado, con varios ToolBox que han sido necesarios para poder emplear funciones necesarias para el rastreo de un rostro o de una pupila, también ha sido necesario la aplicación de MATLAB llamada App Designer para crear el código que cuantifica los ratones de mirada que existen.

Evidentemente, para el objetivo de cuantificar un ratón de mirada, se necesita el propio dispositivo en físico para poder hacer un estudio y análisis óptimo. En este caso, se escoge el seguidor ocular empleado en la Fundación Gema Canales para estimar la calidad de estos dispositivos. Se realizó una formulación y evaluación integral del proyecto para confirmar su viabilidad económica y comercial, un análisis detallado que abarcó el estudio de las fuerzas competitivas. Adicionalmente, se llevaron a cabo un estudio de mercado y un análisis técnico.

Posteriormente, se investigaron productos similares existentes en el mercado, identificando las deficiencias de estos dispositivos como soluciones para la falta de autonomía en personas con enfermedades críticas como la ELA. Se exploraron las razones por las cuales estos dispositivos no son adecuados para tratar enfermedades que provocan parálisis total. A partir de este estudio, se establecieron las especificaciones objetivo que debe cumplir el prototipo propuesto, incorporando elementos diferenciadores respecto a otros dispositivos de accesibilidad disponibles en el mercado.

1.4. Medios empleados

El principal recurso utilizado es el lenguaje de programación seleccionado para cumplir con el objetivo establecido. El lenguaje de MathWorks es una plataforma de programación y cálculo numérico ampliamente utilizada por millones de ingenieros y científicos para el análisis de datos, el desarrollo de algoritmos y la creación de modelos.

MATLAB proporciona un entorno de escritorio refinado para el análisis iterativo y los procesos de diseño, junto con un lenguaje de programación que facilita la expresión directa de matemáticas de matrices y arrays. Este software

incluye funcionalidades exclusivas, como la creación de aplicaciones web y de escritorio, la conexión de MATLAB con hardware para diversos tipos de diseño, la visualización y exploración de datos, y la integración con otros lenguajes como Python, C/C++, Fortran y Java. Es importante señalar que, aunque esta última capacidad simplifica el mantenimiento del código en un solo lenguaje, aún no se ha implementado. Además, dado el objetivo de que este proyecto se convierta en código abierto en el futuro, la utilización de múltiples programas podría ser un desafío.

Dentro de la interfaz de MATLAB, es posible acceder a diversas aplicaciones propias que permiten ver cómo funcionan diferentes algoritmos con los datos. Esto facilita la iteración hasta obtener los resultados deseados y la generación automática de un programa en MATLAB para reproducir o automatizar el trabajo.

Finalmente, las toolboxes de MATLAB son desarrolladas profesionalmente, sometidas a rigurosas pruebas y completamente documentadas. Estas herramientas han permitido desarrollar el código de manera más óptima, eficiente y sencilla. Entre las toolboxes disponibles se encuentran algunas como:

En el proceso de realización de este Trabajo Final de Grado son obligatorios una serie de componentes:

Lenguaje de programación:

Se ha empleado MATLAB para la programación de ambos códigos, por lo tanto son necesarios una serie de requisitos de software. Por ejemplo, la licencia del lenguaje ha sido proporcionada por la Universidad Politécnica de Valencia, una vez instalado son clave para el funcionamiento varios Add-ons, es decir, una serie de ToolBox que permiten el uso de ciertas funciones importantes para completar el objetivo.

Se han empleado gran variedad de ToolBox para este Trabajo Fin de Grado, a continuación se detalla información acerca de ellos en orden de importancia.

1. MATLAB Support Package for USB Webcams

Permite traer a MATLAB imágenes en directo desde cualquier Webcam compatible con la clase de vídeo USB (UVC). Esto incluye Webcams que pueden estar integradas en portátiles u otros dispositivos, así como las que se conectan a su ordenador a través de un puerto USB. Conéctese a su Webcam desde el escritorio de MATLAB o a través de un navegador web con MATLAB Online™. Si utiliza MATLAB Online, el paquete de soporte ya está instalado.

2. Image Processing Toolbox

Proporciona un completo conjunto de algoritmos estándar de referencia y aplicaciones de flujo de trabajo para el procesamiento de imágenes, el análisis, la visualización y el desarrollo de algoritmos. Puede realizar segmentación de imágenes, mejora de imágenes, reducción de ruido, transformaciones geométricas y registro de imágenes utilizando técnicas de aprendizaje profundo y de procesamiento de imágenes tradicionales. La caja

de herramientas admite el procesamiento de imágenes 2D, 3D y de tamaño arbitrario. Las aplicaciones de **Image Processing Toolbox** le permiten automatizar flujos de trabajo comunes de procesamiento de imágenes. Puede segmentar datos de imágenes de forma interactiva, comparar técnicas de registro de imágenes y procesar por lotes grandes conjuntos de datos. Las funciones y aplicaciones de visualización permiten explorar imágenes, volúmenes 3D y vídeos, ajustar el contraste, crear histogramas y manipular regiones de interés (ROI). Puede acelerar sus algoritmos ejecutándolos en procesadores multinúcleo y GPU. Muchas funciones de la caja de herramientas admiten la generación de código C/C++ para la creación de prototipos de escritorio y la implantación de sistemas de visión integrados.

3. **Image Acquisition Toolbox**

Proporciona funciones y bloques para conectar cámaras a MATLAB y Simulink. Incluye una aplicación MATLAB que permite detectar y configurar interactivamente las propiedades del hardware. La caja de herramientas permite modos de obtención como el procesamiento en bucle, el disparo por hardware, la adquisición en segundo plano y la sincronización de la adquisición en varios dispositivos. **Image Acquisition Toolbox** es compatible con los principales estándares y proveedores de hardware, incluidos USB3 Vision, GigE Vision y GenICam GenTL. Puede conectarse a cámaras de visión artificial y placas de captura de imágenes, así como a dispositivos científicos e industriales de gama alta.

Además de estos Toolbox, existen muchos otros que ayudan al proceso de programación del *eyetracker*, incluso se han añadido Toolbox que han sido diseñados por usuarios de los cuales hay apenas información pero con sus funciones facilitan acciones que han de llevar a cabo en el proceso. También existen otros que no participan directamente en el seguimiento de pupila o reconocimiento ocular pero son necesarios debido a sus funciones de optimización y la estabilidad que aportan a la simulación. [16]

2. Estado del arte

2.1. Principios de la tecnología de seguimiento de mirada

El seguimiento ocular ha experimentado una notable evolución en el transcurso de los últimos 100 años, desde los primeros dispositivos mecánicos rudimentarios hasta la sofisticada tecnología actual.

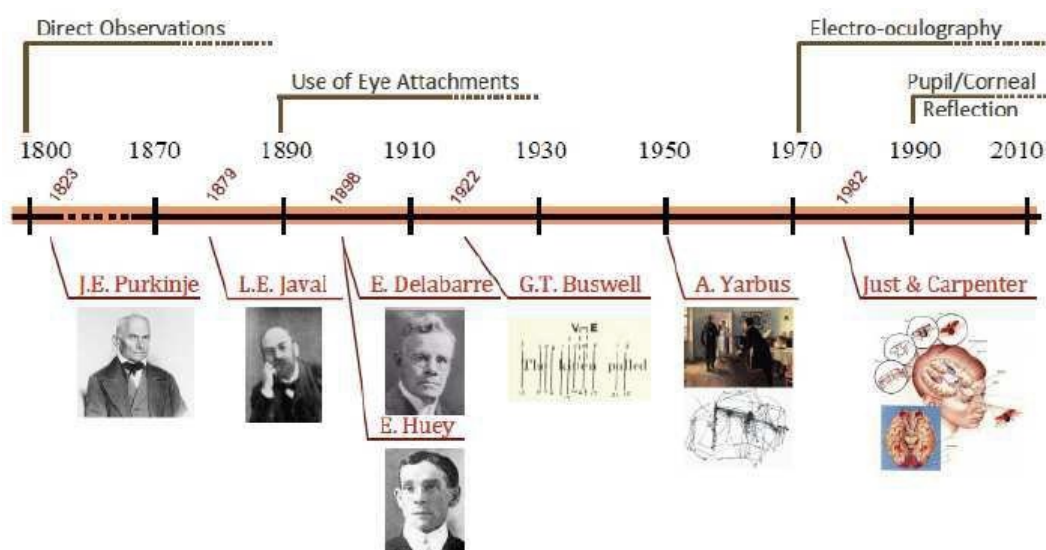


Figura 2.1: Líderes en el desarrollo de la tecnología de seguimiento visual. [1]

Sus orígenes se remontan al siglo XIX, con los estudios iniciales sobre el proceso de lectura. En 1792, se realizaron observaciones visuales simples, conocidas como imágenes fantasma, que son percepciones visuales que persisten después de retirar la vista del objeto original. Al principio, los investigadores analizaban los movimientos oculares sin herramientas de medición, basándose únicamente en observaciones directas. Un método consistía en colocar un espejo en las páginas del libro que leía el sujeto, permitiendo al experimentador, situado detrás del sujeto, observar los movimientos oculares reflejados en el espejo. Aunque esta técnica carecía de precisión, proporcionó valiosas conclusiones sobre la percepción visual durante la lectura.

Estas observaciones fueron utilizadas para describir los movimientos oculares por el científico británico William Charles Wells a finales del siglo XVIII y principios del XIX. Wells logró inferir y describir los movimientos oculares, estableciendo las bases para futuras investigaciones en el campo del seguimiento ocular.

La evolución del seguimiento ocular continuó con los estudios del psicólogo francés Louis Emile Javal en 1878 y posteriormente con Lamare en 1892. Sin

embargo, estos métodos iniciales eran invasivos, ya que requerían contacto mecánico directo con la córnea, lo que podía causar incomodidad o daño a los sujetos de estudio. En uno de los estudios, se utilizó una goma elástica para conectar mecánicamente los ojos con las orejas, amplificando el sonido del movimiento ocular. [1]

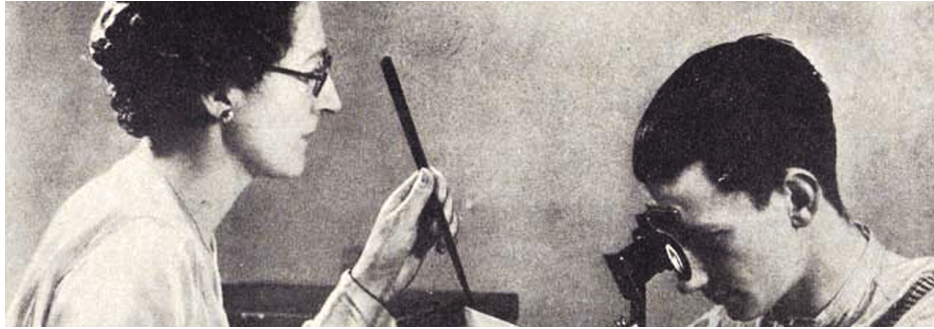


Figura 2.2: Emile Javal, 1879 [2]

En 1879, Louis Emile Javal avanzó en el estudio del seguimiento ocular utilizando espejos para observar los movimientos oculares de los sujetos mientras leían. Javal identificó que estos movimientos consistían en una serie de "sacudidas." sacadas. Para contar las sacadas, se colocaba un micrófono en el párpado cerrado del sujeto, permitiendo así registrar los movimientos monoculares. Se descubrió que, contrariamente a la creencia previa de que los ojos se desplazaban continuamente a lo largo de una línea de texto, la lectura implicaba movimientos cortos y rápidos, conocidos como sacadas, intercalados con pausas o fijaciones en puntos específicos del texto.

Las sacadas se contabilizaban cada vez que una protuberancia de la córnea en movimiento impactaba con el micrófono. La ubicación de las fijaciones se estimaba induciendo una imagen posterior en el ojo del sujeto mediante una luz brillante. Durante la lectura, se solicitaba a los sujetos que indicaran la ubicación de esta imagen posterior. Sin embargo, este método presentaba limitaciones significativas debido a la precisión y la memoria del observador, lo que subrayaba la necesidad de un registro más objetivo y preciso de los movimientos oculares.

Para abordar estas limitaciones, se buscó desarrollar técnicas que permitieran un seguimiento ocular más exacto y menos dependiente de la interpretación subjetiva. La evolución tecnológica subsecuente introdujo métodos más avanzados, incorporando herramientas electrónicas y computarizadas que mejoraron significativamente la precisión y fiabilidad de los datos obtenidos en estudios de seguimiento ocular.

En los años 1897 y 1898, se realizaron los primeros diagnósticos de seguimiento ocular por parte de científicos pioneros en el campo. Uno de los primeros en emplear técnicas mecánicas para registrar movimientos oculares fue Ahrens (1891), aunque se dispone de poca información detallada sobre sus métodos. [1]

Edmund Burke Delabarre, en 1897, desarrolló un método que consistía en fijar una palanca a una capa de yeso de París adherida a la córnea. Posteriormente, en 1898, el psicólogo y educador Edmund Burke Huey, conocido por su investigación

en la lectura y la psicología del aprendizaje, perfeccionó este enfoque. Huey registró los movimientos oculares transfiriendo el movimiento de la palanca a una superficie cubierta de hollín en un tambor giratorio.

Por su parte, Delabarre empleó una pequeña tapa moldeada con yeso de París, que se adhería firmemente a un ojo anestesiado con cocaína, utilizada como anestésico local. Esta tapa se conectaba mediante un cable a una palanca, capaz de registrar los movimientos horizontales del ojo sobre la superficie ahumada de un cilindro kimográfico. Este dispositivo, un tambor recubierto con papel ahumado que gira a velocidad constante, permitía registrar con precisión los movimientos oculares.



Figura 2.3: Edmund Burke Delabarre [3]

Estas técnicas iniciales, aunque rudimentarias y a menudo invasivas, sentaron las bases para el desarrollo de métodos más sofisticados y menos intrusivos en el estudio del seguimiento ocular. La innovación en la instrumentación y el refinamiento de los métodos de registro fueron fundamentales para avanzar en la comprensión de los movimientos oculares y su aplicación en diversas disciplinas científicas.

En los experimentos iniciales, el sujeto leía el texto a través de un orificio en una tapa. Sin embargo, este procedimiento planteaba preocupaciones, ya que Edmund Burke Huey descubrió que el número de sacadas dependía del contenido del texto y no del arco descrito por la rotación del ojo. A pesar de no contar con medios para medir la velocidad del movimiento ocular, Huey dedujo que los datos no se registraban durante las sacadas, sino solo durante las pausas oculares. No obstante, este método mecánico fue criticado debido a la restricción de movimientos y las manchas oculares causadas por el contacto con el equipo experimental.

A principios del siglo XX, Raymond Dodge y Thomas Smith Cline introdujeron avances significativos en el estudio de los movimientos oculares. Estos pioneros desarrollaron una técnica precisa y no invasiva que utilizaba el reflejo de la luz en la córnea para registrar los movimientos oculares. Este método se basaba en una placa fotográfica que caía para registrar únicamente la posición horizontal del ojo, requiriendo que el sujeto mantuviera la cabeza inmóvil.

El primer dispositivo de Dodge proyectaba una línea vertical de luz que se reflejaba en la córnea y pasaba por una rendija horizontal. La placa fotográfica se movía verticalmente detrás de la rendija, regulada por el escape de aire de un cilindro, permitiendo así registrar tanto el movimiento horizontal del ojo (eje X) como el tiempo (eje Y). Esta innovación permitió un registro más preciso y menos intrusivo de los movimientos oculares, marcando un hito en el desarrollo de la tecnología de seguimiento ocular.

Para 1905, Charles Judd, junto con McAllister y Steel, hicieron contribuciones significativas al registro del movimiento ocular. Aunque se dispone de poca información sobre McAllister y Steel, se sabe que colaboraron con Charles Hubbard Judd, un destacado psicólogo y educador. Juntos, lograron introducir la fotografía en movimiento para registrar el movimiento ocular en dos dimensiones. En lugar de utilizar reflejos para capturar la trayectoria del ojo, insertaron una pequeña mota blanca en el ojo de una participante, permitiendo así un registro más preciso del movimiento ocular. Este avance permitió el análisis del movimiento ocular fotograma a fotograma.

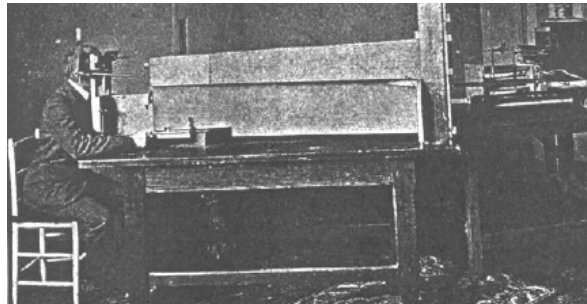


Figura 2.4: Foto-cronógrafo que opera basado en la reflexión corneal.

Para 1920, se logró el primer registro bidimensional del movimiento ocular utilizando la posición de la cabeza y el reflejo de la luz en unas gafas. En 1935, los avances tecnológicos mejoraron aún más esta metodología. Un haz de luz reflejado monocularmente podía dividirse para medir y recombinar su componente vertical, creando un punto de fijación que se grababa en una película. Esto permitía medir la parte del haz que se movía en dirección vertical y luego recombinar los datos para analizar el movimiento ocular vertical. [1]

Este avance fue crucial para investigadores como Charles E. Buswell, un psicólogo estadounidense, que produjo algunos de los primeros estudios sobre trayectorias de exploración bidimensional mientras los sujetos inspeccionaban imágenes. Estos estudios profundizaron en la relación entre el movimiento ocular y los procesos cognitivos, así como en los hábitos de lectura y las diferencias individuales en estos procesos.

Los continuos avances en la tecnología de seguimiento ocular y la metodología de análisis han permitido una comprensión más detallada de cómo los movimientos oculares se relacionan con la cognición y otros comportamientos humanos, consolidando así el seguimiento ocular como una herramienta invaluable en múltiples áreas de investigación.

A finales de la década de 1940, la técnica fotocronográfica de Dodge fue desarrollada y perfeccionada por Guy Thomas Buswell, quien también contribuyó al desarrollo del seguimiento de la cabeza. Este avance marcó un hito significativo en la tecnología de seguimiento ocular.

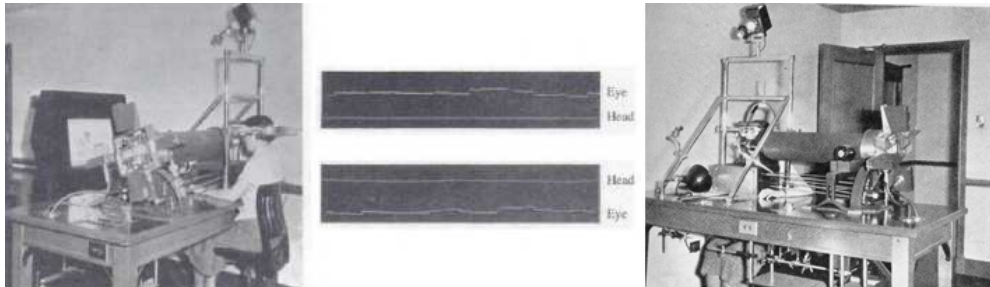


Figura 2.5: Aparato utilizado por Buswell, 1935 [4]

El procesamiento en tiempo real de los datos de la mirada comenzó a vislumbrarse con el método de electrooculografía (EOG) desarrollado por Heinz Otto Jung en 1939. Jung, un científico alemán destacado en psicofisiología y neurociencia, avanzó las mediciones oculares al registrar simultáneamente las direcciones horizontal y vertical. Su técnica involucraba la aplicación de electrodos en la piel cercana al ojo del participante, permitiendo una captura más precisa de los movimientos oculares. [4]

La ingeniería de la usabilidad vio su primera aplicación del seguimiento ocular en 1947, cuando Paul Fitts y sus colegas utilizaron cámaras cinematográficas para estudiar el movimiento ocular de los pilotos en la cabina durante el aterrizaje de un avión. Este estudio pionero proporcionó valiosos insights sobre la interacción entre los pilotos y los instrumentos de vuelo.

En 1948, Hartridge y Thompson inventaron el primer rastreador ocular montado en la cabeza, un avance significativo que permitió a los participantes moverse con mayor libertad, aunque el dispositivo seguía siendo rudimentario en comparación con las tecnologías actuales. Apenas una década más tarde, la técnica evolucionó hacia dispositivos menos intrusivos, facilitando aún más el movimiento de la cabeza de los participantes.

Estos desarrollos en la tecnología de seguimiento ocular han sido fundamentales para liberar a los sujetos de las restricciones físicas impuestas por los primeros equipos, mejorando así la precisión y la comodidad de los estudios. Con el tiempo, estos avances han permitido una aplicación más amplia y efectiva del seguimiento ocular en diversas áreas de investigación y práctica.

En 1967, el psicólogo ruso Alfred L. Yarbus utilizó un sistema basado en una cámara que consistía en un pequeño espejo plano fijado a la superficie de una lente de contacto. Este espejo, adherido mediante succión, permitía registrar el reflejo corneal del ojo bajo una iluminación específica. A diferencia de los métodos anteriores que utilizaban yeso de París, el dispositivo de Yarbus se sujetaba al ojo creando un vacío parcial con una diminuta válvula para extraer líquido de debajo de la lente de contacto. El espejo estaba montado sobre tallos que sobresalían del ojo, evitando que la película lagrimal enturbiara la lente. Estos tallos no solo

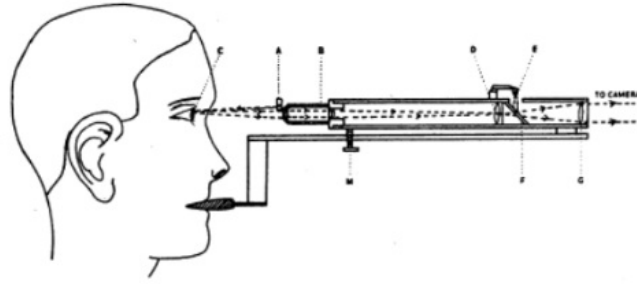


Figura 2.6: Un seguidor de mirada diseñado por H. Hartridge y L.C. Thompson [1]

producían reflejos sino que también podían generar su propia fuente de luz mediante pequeñas lámparas o tritio radiactivo, un isótopo de hidrógeno.

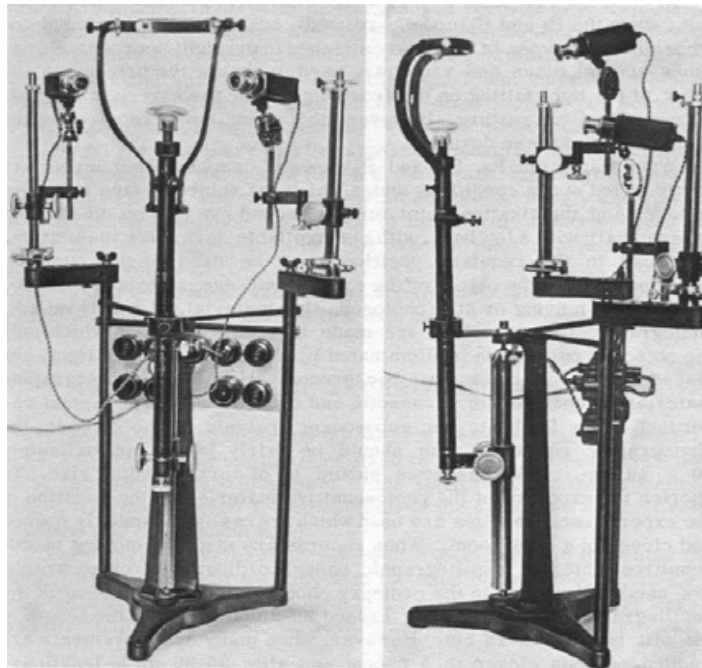


Figura 2.7: Seguidor de mirada (Alfred L. Yarbus, 1967) [1]

Los métodos previos registraban el movimiento ocular en relación con la posición de la cabeza, lo que requería restringir el movimiento de la cabeza para asegurar su inmovilidad. Sin embargo, los avances de la década de 1970 mejoraron tanto la tecnología como la comprensión psicológica del seguimiento ocular. Este período vio el desarrollo de teorías como la hipótesis ojo-mente propuesta por Just y Carpenter, que postulaba que no hay diferencia entre lo que se fija y lo que se procesa visualmente.

La década de 1970 también se caracterizó por un período de inactividad en las actividades de seguimiento ocular, atribuido a los esfuerzos en la recopilación de datos y la resolución de deficiencias tecnológicas y de análisis de datos. Este trabajo preparó el terreno para futuras mejoras y aplicaciones del seguimiento ocular en diversas áreas de investigación.

2.2. Historia y evolución del seguimiento de la mirada

La evolución constante de los dispositivos capaces de rastrear la trayectoria ocular humana ha continuado ininterrumpidamente hasta la actualidad. Los últimos 30 años han sido de particular importancia no solo para el desarrollo de esta tecnología, sino también para la proliferación de avances significativos en diversas áreas tecnológicas. Este periodo marcó la aparición del Internet y la World Wide Web, así como la popularización de dispositivos móviles. Empresas líderes como Apple y lanzamientos fundamentales como Windows 95 contribuyeron significativamente a la integración de estas innovaciones en el ámbito del seguimiento ocular.



Figura 2.8: Evolución seguidores de mirada.[5]

Durante las décadas de 1980 y 1990, los dispositivos de seguimiento ocular experimentaron mejoras significativas impulsadas por los avances en electrónica e informática, siendo aplicados inicialmente en la interacción persona-computadora (HCI). En 1994, Land y Lee desarrollaron un dispositivo portátil de seguimiento ocular montado en la cabeza, equipado con una cámara que permitía registrar tanto los movimientos oculares como la escena frente al sujeto de manera simultánea. Esta innovación permitió la superposición del punto de mira sobre la imagen capturada por la cámara, facilitando su uso en escenarios de la vida real.

A partir del año 2000, los rastreadores ocular portátiles, ligeros, inalámbricos y de tiempo real se hicieron cada vez más accesibles en el mercado, siendo empleados extensivamente en estudios de HCI para entornos de realidad virtual y aumentada.

En 1998, Rayner delineó la evolución de la investigación del movimiento ocular en tres eras distintas: la primera, que abarca hasta aproximadamente 1920, se centró en el descubrimiento de los fundamentos del movimiento ocular; la segunda, desde la década de 1920 hasta mediados de la década de 1970, estuvo dominada por la investigación en psicología experimental. A partir de mediados de la década de 1970, la tercera era presenció mejoras significativas que hicieron el seguimiento ocular más preciso y fácil de registrar.

En 2002, Duchowski identificó el inicio de una cuarta era en la investigación del seguimiento ocular, caracterizada por el aumento de las aplicaciones

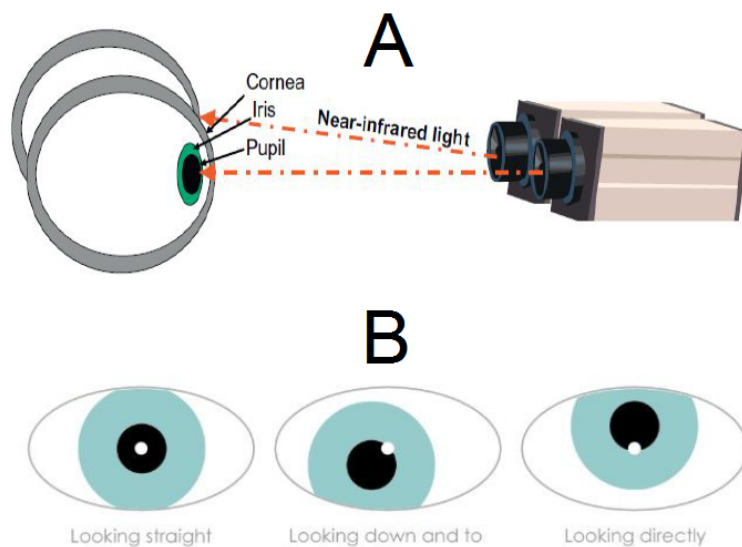


Figura 2.9: Movimiento ocular. [6]

interactivas. Duchowski clasificó los sistemas de seguimiento ocular en diagnósticos e interactivos. Los dispositivos diagnósticos están diseñados para medir procesos visuales y de atención, mientras que los interactivos utilizan el movimiento y la medición de la mirada para interactuar con el usuario.

Dentro de los dispositivos interactivos, se distinguen los selectivos y los contingentes a la mirada. Los selectivos utilizan la mirada como un indicador para la interacción, mientras que los contingentes adaptan su comportamiento según la información proporcionada por la mirada del usuario.

Actualmente, nos encontramos en la cuarta era del seguimiento ocular, impulsada por avances tecnológicos en telecomunicaciones, miniaturización electrónica y potencia computacional. Los dispositivos de seguimiento ocular han evolucionado considerablemente, volviéndose más precisos, accesibles y fáciles de usar. Ahora están disponibles dispositivos ligeros montados en la cabeza que permiten el seguimiento en escenarios del mundo real.

Hasta la fecha, la última revisión exhaustiva de los dispositivos de seguimiento ocular fue realizada por la red Communication by GAZE INteraction (COGAIN) en 2005. En esta revisión, se proporcionó un análisis detallado del progreso técnico en el campo del seguimiento ocular en ese momento. Los autores destacaron la evolución técnica de los dispositivos, incluyendo métodos de seguimiento ocular, interfaces físicas, técnicas de calibración, métodos de archivo de datos, tipos de muestras de archivo, métodos de transmisión de datos y operación de la interfaz de programación de aplicaciones (API).

Más de una década después de la encuesta COGAIN, este resumen subraya los avances recientes en el campo del seguimiento ocular. Este trabajo proporciona a los investigadores expertos una actualización sobre las características y capacidades de los dispositivos más modernos, que continúan mejorando, y permite a los investigadores menos experimentados seleccionar fácilmente la

configuración de adquisición más adecuada para sus estudios.

2.3. Aplicaciones del seguimiento de la mirada en diversos campos

Todo la historia y desarrollo del seguimiento ocular ha dado lugar a una mejora significantes en muchos ambitos de distintas características. Las aplicaciones del eye-tracking se pueden separar en dos categorías: interacción y diagnóstico.

En la función de diagnóstico, el seguimiento ocular proporciona pruebas objetivas y cuantitativas sobre los procesos de atención visual de un usuario. En este contexto, el interés principal radica en los movimientos oculares registrados para determinar patrones de atención sobre un estímulo específico, minimizando la relevancia del punto de mirada (eye-gaze). Es deseable que el rastreador ocular sea discreto e incluso camuflado para evitar que los usuarios sean conscientes de su presencia. Además, en la modalidad de diagnóstico, generalmente no es necesario evaluar los movimientos oculares en tiempo real; se puede realizar una grabación y analizar los resultados después de la fase de captura.

En la modalidad de interacción, el eye-gaze es el mecanismo más utilizado. Este método actúa como un poderoso dispositivo de entrada para diversas aplicaciones con interfaz gráfica. El sistema detecta la mirada del usuario y reacciona en consecuencia, interactuando con él. Un ejemplo destacado es el uso de un teclado proyectado en una pantalla que se controla únicamente con la mirada. Mediante el seguimiento ocular, calibraciones precisas y cálculos específicos, se puede estimar el punto exacto de la pantalla al que el usuario está mirando, permitiendo una interacción eficaz y precisa.

La clasificación inicial de las técnicas de seguimiento ocular se puede hacer en función del seguimiento de un solo ojo o de ambos ojos. Dependiendo de la aplicación y los equipos disponibles, se puede realizar seguimiento ocular monocular o binocular. Para estudios psicológicos y fisiológicos, especialmente en la investigación de posibles enfermedades, es esencial el seguimiento de ambos ojos. Sin embargo, para rastrear el punto de observación de un usuario sano, el seguimiento de un solo ojo puede ser suficiente. No obstante, el seguimiento binocular siempre será más fiable y preciso que el monocular.

Existen tecnologías avanzadas para detectar el movimiento ocular que requieren contacto directo con el usuario, como lentes de contacto especiales, sensores de campo magnético o electrodos que miden el potencial eléctrico para detectar el movimiento. Sin embargo, este proyecto se centra en metodologías no invasivas, por lo que no se profundizará en estas técnicas invasivas.

La tecnología más utilizada para el seguimiento de la mirada es a través de una o varias cámaras de rayos infrarrojos, que permiten detectar las pupilas con facilidad. Esta técnica ha sido ampliamente empleada en la investigación del sistema visual, la psicología cognitiva y el diseño de publicidad para productos. Además, se implementa en gafas y cascos especiales.

Otro tipo de seguimiento ocular es la denominada reflexión en la córnea. Esta técnica utiliza una luz para iluminar los ojos, causando una reflexión en la córnea que es detectada por una cámara de alta resolución. La imagen capturada se utiliza para identificar la reflexión de la luz en la córnea y la pupila, y mediante algoritmos avanzados de procesamiento de imágenes, se estima el punto que se está contemplando.

Por último, existen dispositivos de seguimiento ocular (eye-trackers) que emplean videocámaras o webcams estándar, complementadas con software avanzado capaz de estimar el movimiento del ojo. Aunque algunos algoritmos pueden proporcionar estimaciones razonablemente precisas utilizando una sola cámara, lo habitual es utilizar múltiples cámaras de alta definición para lograr un seguimiento de la mirada más fiable y preciso.

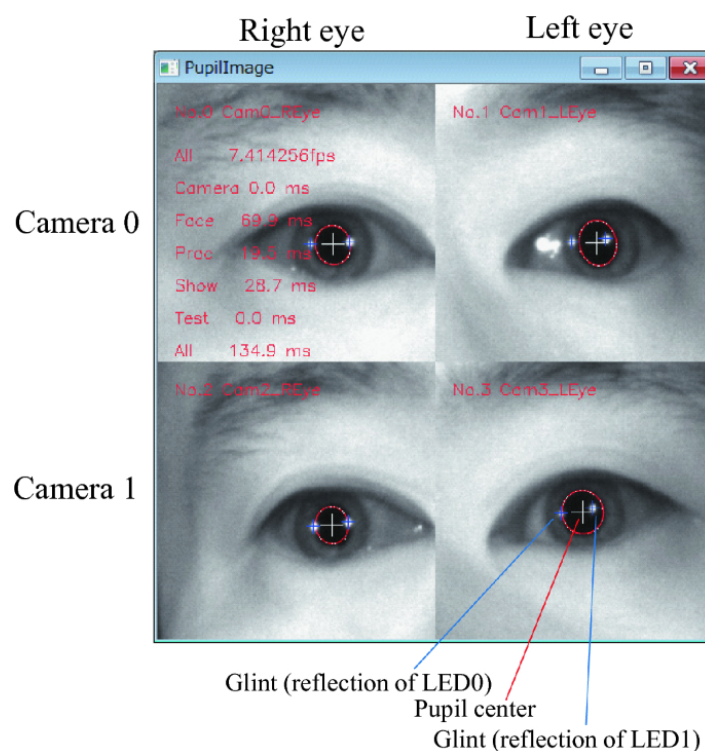


Figura 2.10: Metodo de reflexión en la córnea.[7]

El seguimiento ocular ha encontrado aplicaciones diversas y significativas en varios campos científicos y prácticos, como ocurre en investigación científica, en la que muchas universidades europeas y norteamericanas han utilizado al movimiento ocular para sus avances y proyectos. Por ejemplo, el ámbito de la Psicología y Neurociencia, estudios como el publicado en el Journal of Cognitive Neuroscience han utilizado tecnología de seguimiento de la mirada para investigar patrones de atención en niños con trastornos del neurodesarrollo, proporcionando nuevas perspectivas sobre los mecanismos cognitivos subyacentes y posibles intervenciones terapéuticas, bajo la dirección del Dr. John Smith en la Universidad de Harvard.

La tecnología de seguimiento ocular también ha encontrado aplicaciones con gran impacto en la pedagogía, ergonomía, rendimiento deportivo, salud,

videojuegos, y la interacción con dispositivos móviles y ordenadores.

En el ámbito de la salud y el rendimiento deportivo, el seguimiento ocular es fundamental. En medicina, se emplea en diagnósticos clínicos y corrección de defectos visuales, siendo particularmente útil en operaciones con láser para corregir problemas como la miopía, hipermetropía o astigmatismo. En deportes, como el baloncesto, se utiliza para entrenar lanzamientos a canasta, ayudando a los jugadores a enfocarse en áreas específicas del tablero para mejorar su técnica de tiro. También se analiza el movimiento ocular para calcular tiempos de reacción y estudiar los reflejos de atletas profesionales.

La ergonomía y la interacción con dispositivos móviles se benefician de esta tecnología. Empresas de aeronáutica y fabricantes de automóviles utilizan el seguimiento ocular para optimizar la disposición de los paneles de mando, mejorando la seguridad y la usabilidad en situaciones de emergencia. En dispositivos móviles, software avanzado permite a los terminales con cámaras frontales realizar un seguimiento básico de la mirada. Por ejemplo, algunos dispositivos Android permiten a los usuarios desplazar documentos o páginas web simplemente moviendo la mirada hacia abajo, facilitando una interacción más intuitiva y eficiente.

En el ámbito de la pedagogía, el seguimiento ocular permite identificar dificultades en la lectura de textos en lenguajes no familiares, detectando patrones repetitivos o fijaciones prolongadas que pueden indicar discapacidades cognitivas. Además, la detección de variaciones en la pupila abre la posibilidad de evaluar la carga de trabajo mental de un individuo.

La interacción con ordenadores y la ergonomía se entrelazan en aplicaciones que mejoran la accesibilidad y la usabilidad. La tecnología de seguimiento ocular ha revolucionado la interacción humano-computadora, especialmente para personas con discapacidades físicas. Software y hardware especializados permiten a individuos con discapacidades en las extremidades superiores, como la esclerosis múltiple, manejar ordenadores mediante el seguimiento ocular, facilitando así su acceso y uso de la tecnología de manera autónoma y eficiente.

Finalmente, en el ámbito de los videojuegos, existen cascos con sistemas de seguimiento ocular que permiten a los jugadores interactuar con los juegos mediante la mirada, complementando los controles tradicionales y ofreciendo una experiencia más inmersiva. Esta tecnología no solo enriquece la experiencia de juego, sino que también abre nuevas posibilidades para el diseño y la implementación de interfaces de usuario innovadoras.

La continua evolución de la tecnología de seguimiento ocular sigue abriendo nuevas fronteras en múltiples disciplinas, demostrando su versatilidad y potencial para transformar diversas áreas de la vida y la industria.

2.4. Requisitos y restricciones

El estudio del seguimiento ocular y la tecnología de eye-tracking revela diversas problemáticas y restricciones en sus aplicaciones. Una de las principales dificultades es el coste elevado de los dispositivos, lo cual limita su accesibilidad para muchas familias que podrían beneficiarse de esta tecnología. Este Trabajo Fin de Grado tiene como objetivo abordar este problema económico mediante el desarrollo de un eye-tracker más asequible y mejorado, capaz de identificar con precisión el punto de mirada del usuario, comparándolo con los dispositivos existentes en el mercado.

Otra limitación significativa es la resolución de las cámaras web. Las cámaras de baja resolución no permiten el uso de técnicas avanzadas como la reflexión en la córnea, lo que es crucial para ciertos estudios. Además, estas cámaras presentan dificultades en el calibrado de iluminación y contraste, lo que afecta la eficacia del algoritmo de detección ocular en condiciones de iluminación variable. La baja resolución y la variabilidad de la iluminación reducen la calidad de la detección de pupilas, afectando negativamente la precisión y confiabilidad del seguimiento ocular.

El entorno de desarrollo seleccionado también presenta desafíos. En este proyecto se ha utilizado MATLAB debido a sus amplias herramientas y toolbox de visión, detección de rostros y ojos, que facilitan el proceso de grabación con una cámara web de manera sencilla. Sin embargo, la detección ocular o de pupila podría beneficiarse de programas externos que ofrecen soluciones más avanzadas y específicas. El uso de MATLAB impone la restricción de no poder realizar un seguimiento ocular en tiempo real, limitándose a grabaciones previas o imágenes fijas, lo cual representa un obstáculo significativo para el objetivo de este proyecto.

Este trabajo busca superar estas limitaciones proponiendo soluciones innovadoras y prácticas, con el fin de hacer más accesible y eficiente la tecnología de eye-tracking para un mayor número de usuarios.

3. Sistemas de medición ocular

Los movimientos oculares tienen diferentes significados para distintas personas. En consecuencia, la forma en que se registran y la precisión con la que se analizan pueden variar considerablemente entre disciplinas. La importancia de la medición de los movimientos oculares se manifiesta en aplicaciones que abarcan desde la psicología social del contacto visual, pasando por los procesos cognitivos que implican el examen de las trayectorias de exploración visual, hasta la investigación de la función oculomotora en enfermedades neurológicas y los estudios neurofisiológicos más precisos del comportamiento de las sacadas. Estos ejemplos ilustran la diversidad de enfoques en esta área de investigación.

Dado los requisitos dispares de la investigación sobre movimientos oculares, es previsible que se mantenga la necesidad de utilizar una variedad de técnicas de medición. Esta contribución se centra en la evaluación de los movimientos oculares basada en imágenes, un enfoque que está ganando relevancia debido a los avances continuos en sensores de imagen y en el hardware y software de procesamiento. [8]

Unos 25 años más tarde, con el establecimiento de técnicas electrónicas para el escaneo y la reproducción de imágenes, se introdujo lo que podría denominarse el "enfoque de procesamiento de imágenes" para las mediciones oculométricas, como lo propusieron investigadores como Lowenstein & Lowenfeld (1957).

Lowenstein & Lowenfeld, 1957	Electronic pupillography
Stark & Sandberg ,1961	A simple instrument for measuring eye movements.
Green & Maaseidvaag, 1967	Closed circuit television pupillometer
Ishikawa et al, 1970	A new videopupillography
Merchant et al, 1974	Remote measurement of eye direction allowing subject motion over one cubic foot.
Barbur et al,1987	A new system for the simultaneous measurement of pupil size and two-dimensional eye movements.

Figura 3.1: Enfoques ejemplares que utilizan técnicas de vídeo y escaneo de imágenes en oculometría.[8]

Estos autores reportaron el uso de escaneo de imágenes mecano-óptico, adaptado de los principios clásicos de las cámaras de televisión. Este informe seminal fue seguido por una serie de refinamientos, como los realizados por Asano et al. (1962), quienes emplearon técnicas puramente electrónicas, todas basadas en el principio de escaneo de imágenes de televisión. Desarrollos posteriores, como los de Green & Maaseidvaag (1967) y Oggneill & Stark (1968), emplearon una explotación más sofisticada de los tubos de imagen vidición y las señales de video resultantes. Esta evolución histórica se ilustra en las publicaciones ejemplares

enumeradas en la figura 3.1

Un enfoque particularmente innovador para la medición de la posición ocular mediante técnicas de video fue publicado por Merchant (1974). Este método permitió evaluar la posición horizontal y vertical del ojo en la cabeza. El dispositivo era único porque no solo medía la posición del ojo, sino que lo hacía de manera remota, permitiendo al sujeto de prueba una libertad completa de movimiento de la cabeza dentro de un espacio de un pie cúbico, sin ninguna invasión. Se informó que la precisión de medición de este sistema era de un grado. Versiones refinadas de esta técnica se emplean en los equipos comerciales de seguimiento ocular disponibles actualmente de diversos fabricantes.

Además de estos sistemas remotos, existe una variedad de video-rastreadores oculares de última generación, con la cabeza fija, que realizan mediciones de la posición horizontal y vertical del ojo con precisiones que varían entre 1,0 grados y 0,1 grados, típicamente en un rango de +/- 2 grados. Las frecuencias de muestreo discreto varían desde la frecuencia estándar de cuadros de video de 25 Hz/30 Hz, hasta múltiplos de 200/240 Hz, aunque las tasas de muestreo más altas suelen implicar una correspondiente pérdida en la resolución espacial.

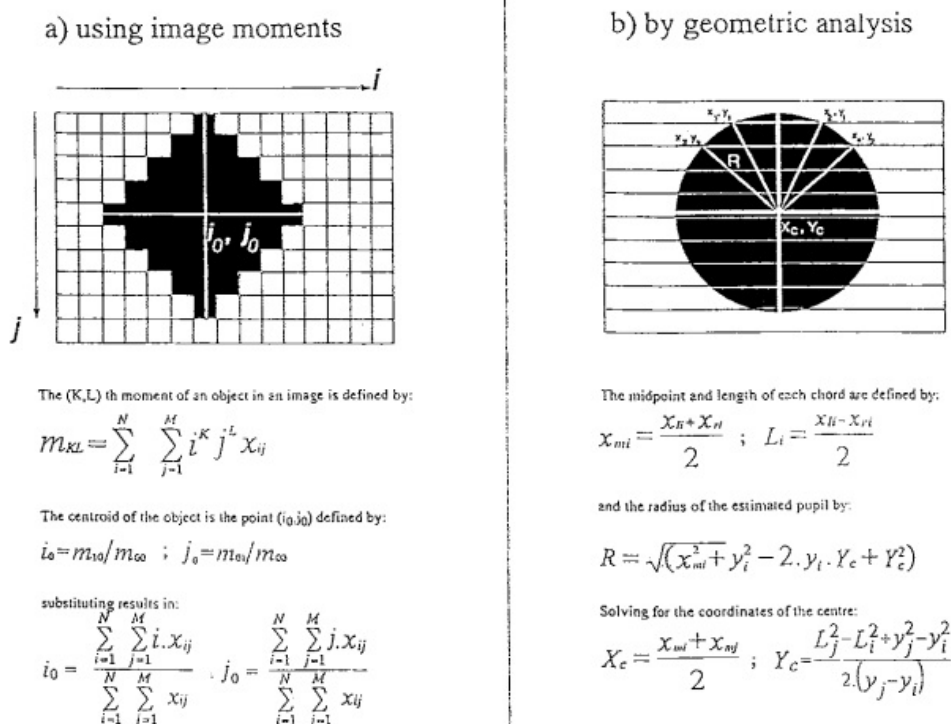


Figura 3.2: Ecuaciones momentos [8]

En la imagen 3.2, apartado a, se observa que la estimación del centro de la pupila se realiza a partir de los momentos de bajo orden.

Por otro lado, el centro y el radio del círculo de mejor ajuste se estiman mediante análisis geométrico, utilizando pares de cuerdas en la imagen 3.2, apartado b. Aunque ambos métodos permiten mejorar la estimación mediante interpolación subpíxel.

3.1. Electro-Oculografía(EOG)

La técnica de electrooculografía (EOG) se basa en la diferencia de potencial entre la córnea y la membrana de Bruch, una capa interna de la coroides. Este potencial, originado en el epitelio pigmentario de la retina, es de aproximadamente 0,4-0,5 mV y se puede modelar como un dipolo, donde la córnea actúa como el extremo positivo y la retina como el extremo negativo, alineado con el eje posterior del globo ocular.

Para medir los movimientos oculares horizontales o verticales, se colocan electrodos cerca de los bordes de las órbitas, conectados a un amplificador de señal. Un electrodo de referencia se sitúa en la frente o el lóbulo de la oreja. Cuando los ojos están centrados, se registra un potencial de reposo entre los electrodos. La magnitud y polaridad del potencial registrado dependen del ángulo del dipolo con respecto a los electrodos.

Durante las rotaciones oculares, hacia la derecha, izquierda, arriba o abajo, el potencial registrado por los electrodos varía diferencialmente. Típicamente, las rotaciones laterales producen un cambio de potencial de entre 10 y 40 μ V. Estos registros son útiles tanto en laboratorios de optometría como en neurooftalmología para evaluar movimientos oculares de versión y vergencia.

Aunque el principio básico de la EOG ha permanecido constante, los avances en la tecnología de electrodos y amplificadores han mejorado su precisión y facilidad de uso en aplicaciones clínicas y de investigación. La disposición y número de electrodos pueden variar según el método y la necesidad de medición, permitiendo un registro preciso del potencial producido por el dipolo ocular.

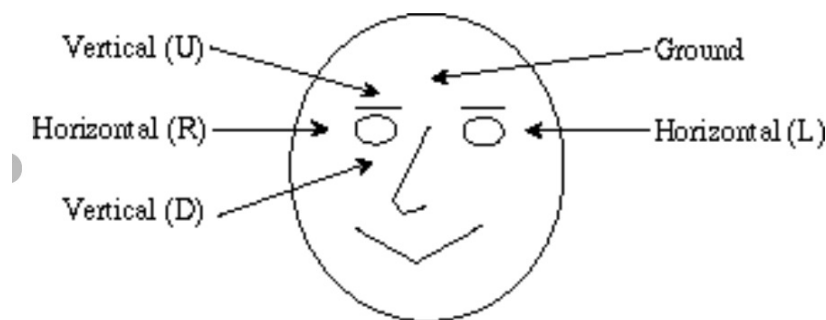


Figura 3.3: Representación de la posición de los electrodos[9]

3.2. Electro-Oculografía(EOG)

Los movimientos oculares horizontales (EOGh) y verticales (EOGv) se registran a través de electrodos, y estas señales se procesan utilizando algoritmos especializados. El algoritmo Continuous Wavelet Transform-Saccade Detection (CWT-SD) se encarga de segmentar estas señales en movimientos sacádicos y fijaciones, permitiendo calcular la amplitud y dirección de los movimientos sacádicos.

Para detectar parpadeos, se emplea el algoritmo Continuous Wavelet Transform-Blink Detection (CWT-BD), que identifica estos eventos utilizando un umbral de 500 ms entre los dos picos principales en la señal EOGv.

En este contexto, el concepto de dispersión (D) es crucial para identificar fijaciones, ya que asume que la mirada se mantiene estable durante estos periodos. Las fijaciones se determinan aplicando un umbral sobre la dispersión de los puntos de vista en la escena visual, descartando aquellos segmentos que superan un umbral de dispersión o tienen una duración inferior a 200 ms.

En la práctica clínica, se utilizan amplificadores con un ancho de banda de 30 Hz para procesar estas señales. Sin embargo, cuando se requiere una mayor precisión en la medición de los movimientos oculares, se pueden emplear anchos de banda entre 80 y 100 Hz, aunque esto incrementa la susceptibilidad de la señal EOG a artefactos [10]. Aunque las técnicas de filtrado sofisticadas pueden mitigar estos artefactos, otras tecnologías como las bobinas de búsqueda escleral, VOG y electro-infrarrojas son preferidas en aplicaciones más avanzadas. Históricamente, cuando era difícil controlar la deriva electrónica en los amplificadores, se usaban filtros de paso alto con una constante de tiempo de 5-10 segundos para estabilizar la señal.

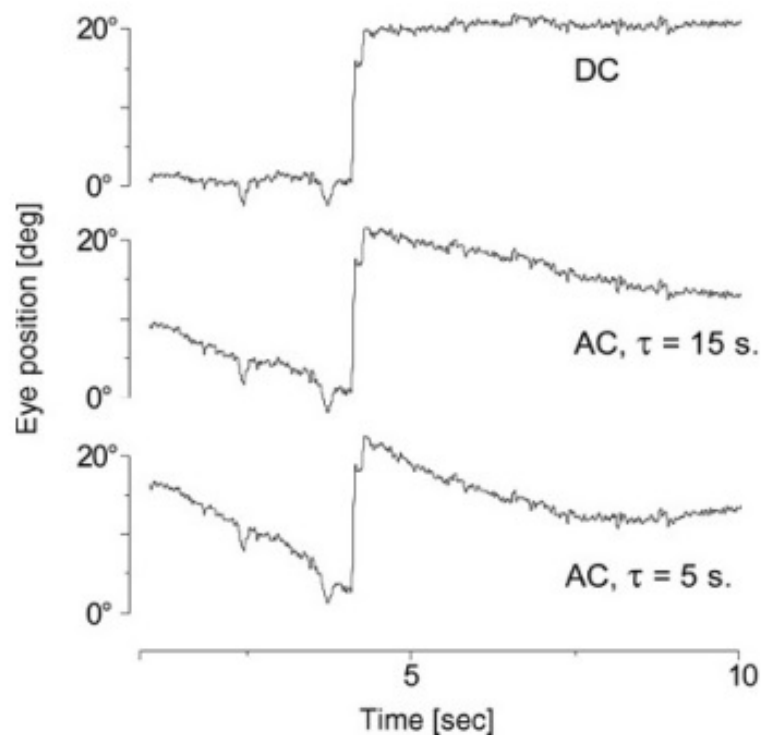


Figura 3.4: Registro EOG de 20 sostenimientos excéntricos de la mirada. [10]

3.3. Videooculografía (VOG)

En los últimos años, ha crecido la conciencia y el interés en la importancia de medir los tres grados de libertad del movimiento ocular: los componentes

horizontal, vertical y ciclorrotacional o de torsión. Este enfoque integral es crucial para un análisis completo del comportamiento ocular, especialmente en situaciones donde los movimientos torsionales desempeñan un papel significativo.

En cuanto a las técnicas de procesamiento de imágenes, un intento pionero en el registro de los tres grados de libertad de los movimientos oculares en respuesta a la estimulación visual-vestibular fue realizado por Melvill-Jones en 1963. Este investigador se centró en los movimientos oculares de los pilotos durante maniobras de giro en aviones. Para llevar a cabo su estudio, utilizó una cámara cinematográfica de 16 mm modificada específicamente para esta tarea. El material filmado se analizaba manualmente, fotograma a fotograma, sobre una superficie de proyección calibrada, permitiendo una evaluación detallada de los movimientos oculares en las tres dimensiones.

Este enfoque se basa en la utilización de cámaras o dispositivos de adquisición de imágenes para determinar el movimiento ocular a partir de las imágenes capturadas. Aunque rudimentario en comparación con las tecnologías modernas, el trabajo de Melvill-Jones representó un avance significativo en la comprensión de los movimientos oculares y sentó las bases para el desarrollo de técnicas más sofisticadas en el estudio de la respuesta ocular a estímulos complejos.

El sistema se organiza en las siguientes fases:

1. Captura de video:

Se colocan cámaras frente a los ojos de la persona. Estas cámaras graban en alta velocidad, capturando cada movimiento ocular.

2. Detección de la pupila:

El software analiza las imágenes de video para detectar la posición de la pupila en cada fotograma.

3. Registro de movimientos:

A partir de la posición de la pupila en las imágenes, el sistema puede determinar hacia dónde está mirando la persona y cómo se están moviendo sus ojos (por ejemplo, si está mirando hacia arriba, abajo, a los lados, etc.).

4. Análisis de datos:

Los datos recopilados se pueden utilizar para diversos fines, como evaluar problemas de visión, investigar el comportamiento visual, o mejorar interfaces en tecnología de realidad aumentada o virtual.

Se distinguen varios sistemas VOG con los que obtener información del ojo para calcular el punto de vista del usuario.

1. **Imágenes de Purkinje**

Se basa en un método de reflexiones producidas por fuentes IR en distintos puntos del globo ocular. Según el ojo humano. Estas reflexiones se forman en cuatro ubicaciones específicas del ojo: la primera imagen de Purkinje, que es la más clara y brillante, se genera en la superficie anterior de la córnea; la

segunda imagen se produce en la superficie posterior de la córnea y suele estar casi siempre muy cerca o superpuesta a la primera, debido a la mínima distancia entre ambas superficies corneales; la tercera imagen se forma en la superficie anterior del cristalino; y la cuarta imagen, que es invertida, se genera en la superficie interna del cristalino. Este método se basa en la comparación de la posición entre la primera y la cuarta imagen para determinar el movimiento ocular. Sin embargo, también existen otros enfoques que utilizan únicamente la primera imagen de Purkinje en combinación con otras características oculares para calcular la dirección de la mirada.(3.5)

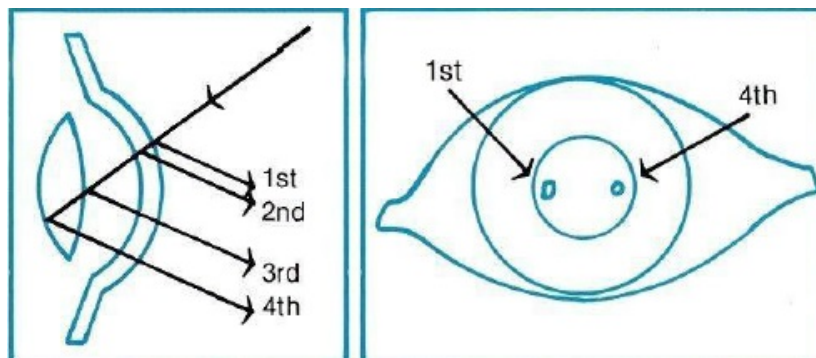


Figura 3.5: Metodo de Purkinje [11]

2. Pupila brillante y pupila oscura

En la Video-Oculografía (VOG), la luz infrarroja (IR) es preferida sobre la luz visible para evitar reflexiones indeseadas y lograr una iluminación uniforme de los ojos. Existen dos métodos principales que utilizan luz IR: la técnica de pupila brillante, que ilumina el ojo cerca del eje de la cámara, resultando en una pupila brillante, y la técnica de pupila oscura, donde la fuente de luz está fuera del eje, haciendo que la pupila aparezca más oscura. La detección de la pupila es ventajosa debido a su contorno pequeño y definido, aunque la luz IR no se puede usar en exteriores. La técnica de pupila brillante mejora el contraste entre el iris y la pupila, facilitando un seguimiento ocular más preciso.

3. Vector PCCR

El vector entre el centro de la pupila y la primera imagen de Purkinje, conocido como vector PCCR, es ampliamente utilizado para detectar la dirección de la mirada en lugar de basarse únicamente en el centro de la pupila. Este método presenta varias ventajas, como la facilidad para extraer las características necesarias y su robustez ante movimientos de cabeza. Debido a estas ventajas, el vector PCCR es un método comúnmente empleado, como se evidencia en múltiples publicaciones.

Dentro de los sistemas basados en Video-Oculografía (VOG), se pueden diferenciar dos tipos de aproximaciones: las aproximaciones 2D y las aproximaciones 3D.

-Aproximaciones 2D:

Las aproximaciones 2D en el seguimiento ocular transforman las coordenadas de las características del ojo o los valores de intensidad en imágenes bidimensionales en coordenadas del punto de vista en el plano del monitor 2D. Estas técnicas son apreciadas por su facilidad de implementación y la ausencia de necesidad de calibrar la cámara o calcular la distancia al monitor. Sin embargo, uno de sus inconvenientes es que suelen capturar la cara completa del sujeto, lo que reduce la resolución de la región ocular. Existen excepciones cuando se utilizan imágenes de alta resolución más propias de las aproximaciones 3D.

Las técnicas de correlación 2D implican la identificación manual y la selección de al menos dos puntos de referencia en el iris, idealmente a 180 grados de separación. La traslación de estos puntos de referencia se estima mediante la función de correlación cruzada normalizada como medida de similitud fotograma a fotograma [8]. La rotación de la imagen se calcula en función de la distancia entre los dos puntos de referencia y sus traslaciones desde el centro de la imagen. Aunque estos métodos son simples, suelen ofrecer una precisión menor en el seguimiento ocular, ya que no utilizan información tridimensional real del ojo y requieren un proceso de calibración dependiente del usuario. Este proceso de calibración, donde el usuario debe fijar la mirada en puntos específicos de la pantalla, puede ser extenso.

Uno de los beneficios de implementar redes neuronales en el seguimiento ocular es la libertad de movimiento de la cabeza del usuario. Para compensar los desplazamientos relativos entre la cámara y el ojo, es necesario localizar el ojo en todo momento. En este sistema, el ojo derecho se identifica mediante la búsqueda del reflejo especular de una fuente de luz normal sobre el ojo, que aparece como un pequeño punto blanco rodeado de una región oscura (3.6). Una vez localizada la posición del ojo, se restringe la zona de búsqueda en el siguiente instante temporal, lo que mejora la precisión y eficiencia del seguimiento [11].

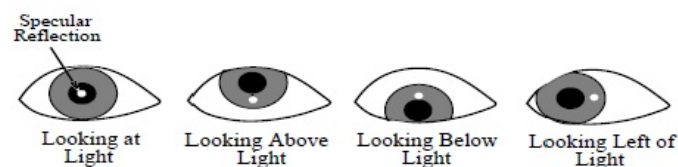


Figura 3.6: Aproximación 2D [11]

Características de las aproximaciones 2D

- **Análisis en un plano:** Se registra cómo se mueven los ojos en dos dimensiones, generalmente en un plano horizontal y vertical. Por ejemplo, si una persona mira una pantalla, las aproximaciones 2D rastrean hacia dónde se dirige la mirada en esa pantalla.
- **Limitación a dos ejes:** Este tipo de análisis se centra solo en los movimientos en dos ejes (X e Y), ignorando la profundidad o el eje Z. Es decir, se ignoran los cambios en la distancia entre el ojo y el objeto observado.
- **Aplicaciones comunes:** Las aproximaciones 2D son muy útiles para tareas como el seguimiento de la mirada en pantallas, diagnóstico de trastornos

oculares, o el control de interfaces de usuario donde solo se necesita saber hacia dónde mira la persona en una superficie plana.

Por un lado, su uso se emplea en mapeo de la mirada, por ejemplo, si se está probando un sitio web, el sistema VOG con aproximaciones 2D puede mostrar exactamente en qué parte de la pantalla está enfocando la mirada la persona. Por otro lado, se emplean en diagnósticos en el ámbito de la medicina, estas aproximaciones ayudan a diagnosticar problemas como el nistagmo (movimientos oculares incontrolables), evaluando cómo se mueven los ojos en respuesta a ciertos estímulos.

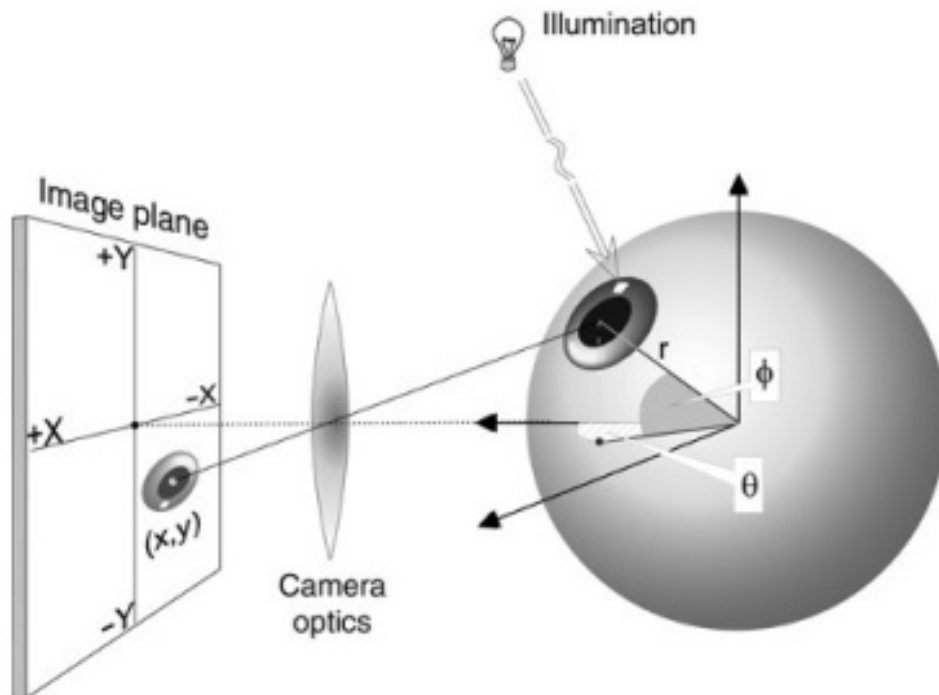


Figura 3.7: Principio básico de video oculografía [10]

En la mayoría de los sistemas VOG, se capturan las coordenadas de la pupila en cada fotograma de video. Estas coordenadas se representan en un sistema de referencia bidimensional (X, Y), donde X es la posición horizontal de la pupila en la imagen, por lo tanto siendo Y la posición vertical.

Para mapear las posiciones pupilares a un plano de observación, como una pantalla, se utiliza una transformación geométrica. Las transformaciones más comunes incluyen:

Transformación Lineal

$$X' = aX + b \quad (3.1)$$

$$Y' = cY + d \quad (3.2)$$

Donde:

- X' y Y' son las coordenadas de la pantalla
- X y Y son las coordenadas de la pupila en la imagen
- $a, b, c,$ y d son parámetros que depende de la calibración del sistema

Transformación Lineal

En casos donde se necesita mayor precisión (por ejemplo, si hay distorsiones debidas a la cámara o la posición de los ojos), se usa una transformación homográfica:

$$X' = \frac{aX + bY + c}{gX + hY + 1} \quad (3.3)$$

$$Y' = \frac{dX + eY + f}{gX + hY + 1} \quad (3.4)$$

Esta transformación considera factores de escala y rotación, y es más flexible para ajustar la relación entre las coordenadas captadas por la cámara y las coordenadas reales en el plano de observación.

Para el metodo de aproximaciones 2D existe y se emplea una plantilla elíptica (3.8) que se ajusta a la forma del ojo en las imágenes capturadas por la cámara.

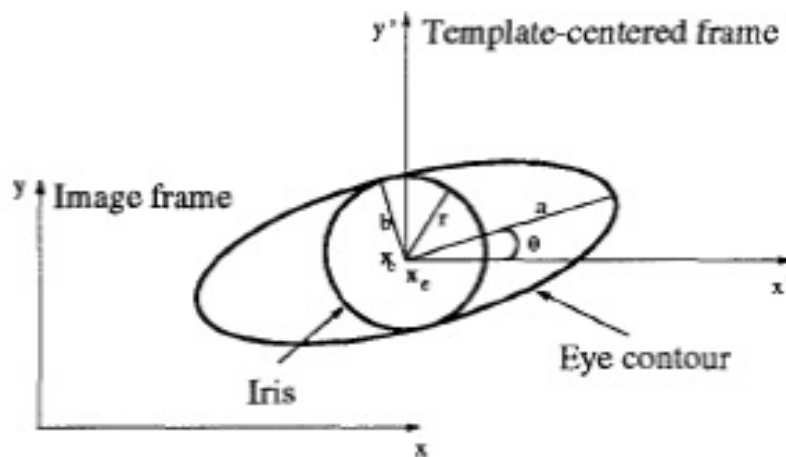


Figura 3.8: Plantilla elíptica [11]

Trás todo el ajuste se busca la posición del iris asumiendo que es el centro de la pupila. A continuación se procede a localizar en la pantalla el punto de mira x_s , para lo cual se utiliza una correspondencia uno a uno con la localización de la pupila en la imagen x_p , mediante un simple modelo de mapeo imagen-pantalla:

$$x_s = A * x_p + t \quad (3.5)$$

en donde t es una traslación y A que es una transformación afín, es decir, una operación matemática que se aplica a una imagen o un conjunto de puntos para modificar su posición, tamaño o forma, manteniendo las líneas rectas y la relación entre puntos. En este caso, la transformación afín puede incluir tres tipos de

cambios, rotación rígida, es un giro de la imagen o del objeto alrededor de un punto fijo, como si lo rotaras sobre un eje. Luego, escalado isotrópico, es un cambio de tamaño que se aplica de manera igual en todas las direcciones. Y por último, el escalado anisotrópico, es un cambio de tamaño que se aplica de manera diferente en distintas direcciones.

Antes de usar el sistema para rastrear la mirada, se necesita calibrarlo. Esto significa que el sistema debe aprender cómo se relacionan las imágenes del ojo con los puntos en la pantalla. Durante la calibración, se muestran una serie de puntos en la pantalla que el usuario debe mirar. Mientras el usuario sigue estos puntos con la mirada, el sistema toma varias imágenes de sus ojos.

Usando las imágenes de los ojos y la información sobre dónde estaba mirando el usuario en la pantalla, el sistema construye el modelo afín. Básicamente, aprende cómo los movimientos de los ojos se traducen en movimientos en la pantalla. Una vez que el sistema ha sido calibrado, sigue funcionando en tiempo real. Esto significa que, mientras el usuario usa el sistema, la calibración puede ajustarse ligeramente para mantener la precisión, especialmente si hay pequeños movimientos de la cabeza u otros cambios.

El método tiene dificultades para manejar movimientos de la cabeza. Si la cabeza se mueve más de 3 píxeles entre dos imágenes consecutivas, la precisión disminuye. El sistema calcula hacia dónde se está mirando, pero como puede equivocarse en la posición de la pupila (con un error promedio de 3 píxeles), esto hace que el punto de mira calculado pueda estar desviado hasta 3.25 cm del lugar real. Estos errores ocurren principalmente porque la cabeza no está completamente inmóvil durante la calibración, lo que introduce pequeñas variaciones que afectan la precisión general del sistema.

-Aproximaciones 3D:

Las aproximaciones 3D en el ámbito del seguimiento ocular constituyen técnicas avanzadas diseñadas para determinar con precisión la posición y orientación de los ojos en un espacio tridimensional. A diferencia de las aproximaciones 2D, que se limitan a considerar coordenadas en un plano bidimensional (como una pantalla), las aproximaciones 3D permiten calcular la dirección de la mirada en un entorno tridimensional, incorporando variables como la profundidad, la posición de la cabeza y la orientación ocular.

En la actualidad, muchos dispositivos de seguimiento ocular emplean cámaras de alta resolución, ya sea mediante zoom o situando la cámara en proximidad cercana al ojo, lo que permite obtener imágenes detalladas y de alta calidad del ojo. Esta alta resolución es fundamental para el análisis tridimensional, aunque limita considerablemente el movimiento de la cabeza del usuario. Para mitigar esta limitación, una alternativa común es el uso de un sistema de dos cámaras: una dedicada al seguimiento de la cabeza y otra al seguimiento ocular.

Las aproximaciones 3D ofrecen varias ventajas significativas en comparación con las 2D. Entre ellas, destaca una mayor precisión en el cálculo del punto de vista, dado que se puede generar un vector de mirada en 3D con respecto al marco de referencia de la cámara. Además, el proceso de calibración dependiente del usuario

es más simple y rápido en el contexto tridimensional.[11]

Sin embargo, estas técnicas también presentan desventajas. La necesidad de una calibración precisa de la cámara y la mayor complejidad en la implementación y en los cálculos asociados son desafíos inherentes a las aproximaciones 3D, lo que puede incrementar tanto los requisitos técnicos como los costos de desarrollo de los sistemas de seguimiento ocular.

Existe un modelo propuesto, 3.9, para el punto de vista, con dos cámaras, una para la posición de la cabeza y otra para el iris de uno de los ojos.

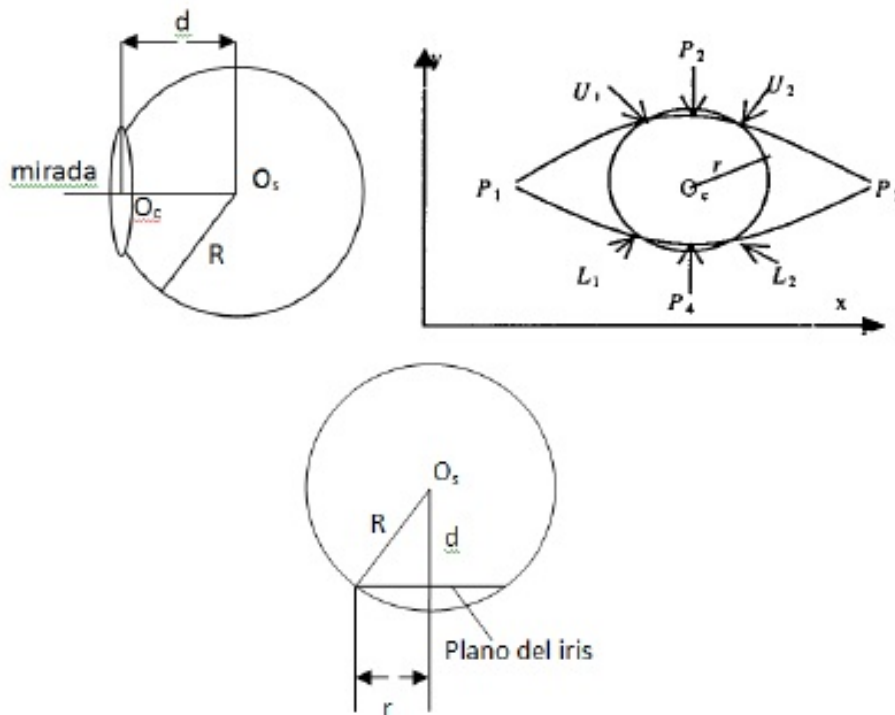


Figura 3.9: Modelo del ojo [11]

El objetivo es encontrar las partes visibles del contorno del iris que no están cubiertas por los párpados, que en el texto se denominan curvas U_1L_1 y U_2L_2 .

Proceso de identificación

- Umbralización: Se aplica un umbral a la imagen para distinguir entre las partes del iris y el fondo.
- Operaciones de apertura: Se realizan operaciones para eliminar pequeñas imperfecciones y mejorar la visibilidad del contorno.
- Extracción de bordes: Se detectan los bordes del iris en la imagen.
- Ajuste de elipse: Se ajusta una elipse a los dos bordes verticales más grandes del iris identificados en el paso anterior.

Por otro lado, se puede determinar analíticamente el plano en el que se encuentra un círculo (como el iris) a partir de su proyección perspectiva, si se conoce el radio del círculo. El algoritmo propuesto presentado en [17] en Osaka,

Japón, en el año 1990, se refiere a la técnica de reconstrucción 3D a partir de trayectorias de imagen de movimiento rotacional. Este algoritmo permite obtener dos posibles posiciones en 3D para el plano del iris a partir del círculo o elipse detectado. Estas posiciones se denominan Os1 y Os2.

Para seleccionar la solución correcta, se debe calcular las distancias entre las posibles posiciones (Os1 y Os2) y ciertos puntos en la imagen (P1 y P3). Tras ello se elige la solución que minimiza la diferencia entre las distancias calculadas, es decir:

- Si la diferencia en las distancias para Os1 es menor o igual a la diferencia para Os2, se selecciona Os1.
- De lo contrario, se selecciona Os2.

Este método descrito se enfrenta a varios problemas y limitaciones. Por un lado el problema con el cálculo del punto de vista, Aunque el método puede determinar el vector de dirección de la mirada, no es capaz de identificar el punto exacto donde se está mirando. Esto significa que, a pesar de saber hacia dónde apunta la mirada, no se puede localizar con precisión el punto en la pantalla o en el entorno que el usuario está observando. Por ello la solución que se propone es asumir que el usuario se encuentra a una distancia fija de 1,3 metros del plano de observación. Esta aproximación permite calcular la posición del punto de vista, pero introduce un error medio de 1,26 cm en la localización del punto exacto.

Por otro lado está el problema con la detección y extracción de características, es decir, que el usuario puede variar su posición, siendo el método sensible a cambios. Las condiciones de luz además de la posibilidad de presencia de gafas, barba, el tipo de peinado, etc. Incluso se caracteriza como problema el ajuste de una elipse al contorno del iris si el contorno visible tiene un número reducido de píxeles. En imágenes donde el contorno es pequeño o poco definido, el ajuste de la elipse puede no ser preciso, afectando la calidad del seguimiento ocular.[11]

Además de este método, existe otro propuesto por S. Shih y J. Liu en el artículo [18] introduce una técnica avanzada para el seguimiento de la mirada utilizando cámaras estereoscópicas.

El método utiliza dos cámaras colocadas en posiciones fijas que capturan imágenes del ojo desde diferentes ángulos. Esto permite obtener información sobre la profundidad y la orientación del ojo en un espacio tridimensional.

Una de las ventajas que presenta son que las imágenes estereoscópicas permiten calcular la profundidad del ojo, lo que es fundamental para un seguimiento de mirada preciso en 3D.

Dicho modelo, se basa en la primera imagen de Purkinje, también conocida como reflejo corneal o imagen de Purkinje-Sanson, es una de las reflexiones de la luz que ocurren en el ojo y es muy importante en el seguimiento ocular. Se emplea para calcular la curvatura de la cornea, y utiliza el modelo del ojo que propone Le Grand, el cual aparece en la 3.10

Yves Le Grand fue un destacado óptico y científico que hizo contribuciones significativas en el campo de la óptica fisiológica. [19]

En su obra "Light, Color and Vision" publicada en 1957, Yves Le Grand realizó contribuciones fundamentales al campo del seguimiento ocular, proporcionando un marco teórico clave para el desarrollo de técnicas avanzadas en esta área. Sus investigaciones se centraron en comprender cómo la luz interactúa con el ojo, desarrollando modelos ópticos que describen el comportamiento de la luz al entrar en el ojo y cómo se refleja en superficies internas como la córnea y el cristalino. Le Grand también profundizó en el estudio de las imágenes de Purkinje, reflexiones críticas para el seguimiento preciso de la dirección de la mirada.

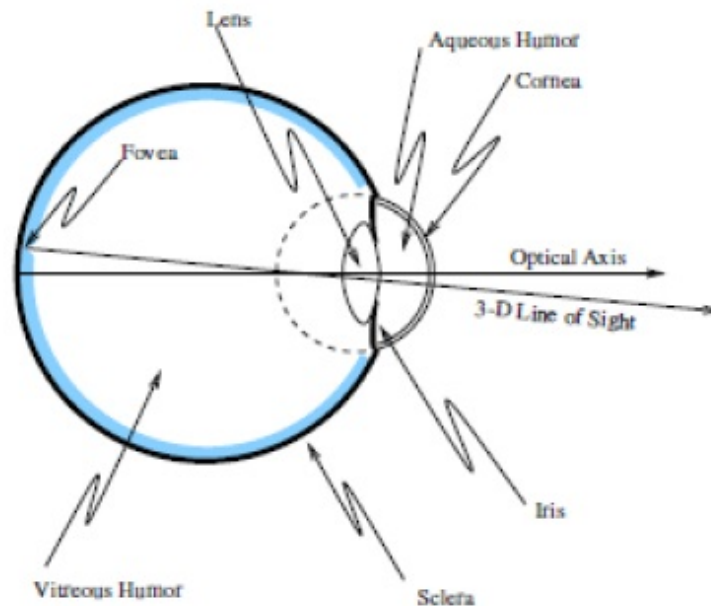


Figura 3.10: Modelo del ojo de Le Grand [11]

Para determinar el número adecuado de cámaras y fuentes de luz necesarias para calcular el centro de la córnea, se han derivado las siguientes conclusiones basadas en la información disponible:

- Si el radio de la córnea, denotado como p , es conocido, se requiere al menos una cámara y dos fuentes de luz puntuales con posiciones conocidas para obtener los datos necesarios.
- Si el radio de la córnea p no es conocido, se necesitan al menos dos cámaras y dos fuentes de luz puntuales. Además, para garantizar una única solución, las fuentes de luz y uno de los centros ópticos de las cámaras no deben estar alineados en una misma línea (no colineales).

En relación con la proyección de la pupila, se tiene en cuenta que los puntos p_i pertenecen al borde del contorno de la imagen bidimensional (2D) de la pupila. Aunque la posición 2D del centro de la pupila p'_i puede ser estimada utilizando estos puntos p_i , esta estimación resulta complicada debido a la no linealidad introducida por la proyección en perspectiva. Sin embargo, dado que el radio de la imagen virtual de la pupila es muy pequeño en comparación con la distancia entre el ojo y la cámara, es factible utilizar una proyección afín para describir la relación entre la posición tridimensional (3D) de la pupila y su imagen 2D proyectada. De

esta manera, la posición virtual de la pupila p'_c se puede determinar con mayor precisión utilizando la proyección afín, como se expresa en la ecuación 3.6, debido a la linealidad inherente a este tipo de proyección.

$$p'_c = \frac{1}{N} \sum_{i=1}^N p_i \quad (3.6)$$

Una vez calibrado, el sistema utiliza la posición proyectada en 2D del centro de la pupila (p'_c) para estimar la orientación del punto de vista en el espacio 3D. Cuando este punto se retroproyecta, el vector resultante (p''_c) es paralelo a la matriz de transformación (T_a), que define la relación entre el sistema de coordenadas auxiliar 3D y el sistema de coordenadas de la cámara.

Si los planos formados por las proyecciones 3D del centro de la pupila (p''_{ci}) y el centro de la córnea (k_{ci}) no coinciden, se puede calcular de manera única tanto el eje óptico del ojo como el punto de vista, ya que estos son coplanarios con la dirección del centro de la pupila en la imagen de la cámara y con la dirección del centro de la córnea respecto a la cámara. [12]

El rendimiento del sistema se evaluó colocando al usuario a una distancia de 45 cm de la pantalla. Tras la calibración, se le pidió al usuario que fijara su mirada en una serie de letras dispuestas en una cuadrícula de 4x4 en la pantalla. Los resultados mostraron errores medios de 0,49 cm en el eje x y de 0,53 cm en el eje y, lo que corresponde a una precisión de menos de un grado ($0,63^\circ$ en x y $0,686^\circ$ en y). Sin embargo, se observó una desviación estándar significativa de 0,135 cm en x y 0,181 cm en y, lo que sugiere que el algoritmo es altamente dependiente de las características intraoculares específicas del usuario.

A pesar de su precisión, el sistema presenta algunas limitaciones. Una de las principales es que el eje x de la cabeza del usuario debe mantenerse paralelo al plano horizontal de la pantalla, lo que restringe la capacidad del usuario para girar su cabeza en el eje y durante el uso del tracker. Además, la necesidad de realizar una calibración dependiente del usuario para determinar el ángulo entre el eje óptico y el eje visual puede complicar su implementación en escenarios de uso más dinámicos. [12]

Se comienza tomando imágenes de alta resolución del ojo mediante una cámara especializada. Estas imágenes, que capturan el ojo en gran tamaño, se reducen a una cuarta parte de su tamaño original para facilitar el procesamiento. La reducción de tamaño ayuda a mejorar la eficiencia en la segmentación de la imagen, donde se identifican regiones con valores de color similares. Cada región segmentada es considerada como una posible candidata para la pupila. Posteriormente, se realiza un ajuste doble de una elipse al contorno de la pupila detectada, para precisar su forma y ubicación.

El siguiente paso consiste en localizar la primera imagen de Purkinje, que es un reflejo especular que aparece cerca de la pupila. A diferencia de otros sistemas que utilizan directamente la posición calculada de la pupila y de la imagen de Purkinje, este método refina estos datos utilizando un modelo del ojo. Dado que la luz

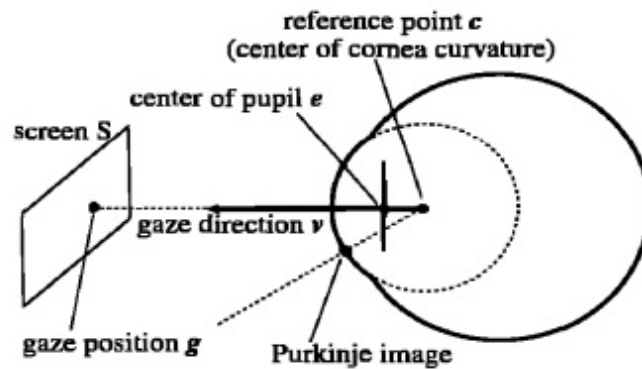


Figura 3.11: Modelo del ojo para la imagen de Purkinje [12]

infrarroja (IR) utilizada para iluminar el ojo no está alineada con el eje de la cámara, se requiere una corrección para calcular la distancia entre la imagen de Purkinje y el eje de la cámara. Además, los puntos del contorno de la pupila deben ser ajustados para compensar la refracción y la curvatura de la córnea.

Una vez ajustada la elipse a los contornos reales, se determina el centro de la elipse, lo que permite calcular el centro de la pupila. Con esta información, el vector de mirada se obtiene al calcular la diferencia entre las coordenadas del centro de la pupila (e) y el centro de curvatura de la córnea (c). La posición de la mirada se determina mediante la intersección de este vector con la pantalla (S).

El sistema está diseñado para permitir un movimiento limitado de la cabeza. Utilizando solo una cámara, el área de movimiento permitida es de aproximadamente 4 cm^2 con el usuario situado a 60 cm de la pantalla. La precisión del sistema varía entre $0,24$ y $0,73 \text{ cm}$ en la coordenada x y entre $0,31$ y $0,94 \text{ cm}$ en la coordenada y . En algunos casos, el error puede superar 1 cm , en cuyo caso se recomienda repetir el proceso de calibración dependiente del usuario. Las principales limitaciones del sistema incluyen la restricción del movimiento de la cabeza y las dificultades para localizar la primera imagen de Purkinje en personas que usan lentillas.

Por otro lado, Park [20] propone el uso de un sistema de seguimiento ocular que emplea cuatro luces infrarrojas (IR) situadas en las esquinas de la pantalla. Estas luces generan cuatro reflejos especulares en la córnea del ojo, conocidos como imágenes de Purkinje. Junto con el centro de la pupila, estos reflejos proporcionan los datos de entrada necesarios para calcular el punto de vista del usuario. La relación espacial entre el centro de la pupila y los cuatro reflejos se utiliza para estimar con precisión hacia dónde está mirando el usuario.

Una característica clave del sistema de Park es la capacidad de permitir el movimiento de la cabeza del usuario. Esto se logra mediante el uso de gafas de visión a través (see-through), sobre las cuales se monta una cámara que graba exclusivamente el ojo. Esta configuración es ventajosa porque mantiene la cámara fija en relación con el ojo, lo que facilita la captación de datos estables incluso cuando la cabeza se mueve.

El sistema propuesto utiliza el Filtro Extendido de Kalman (EKF) para realizar un seguimiento continuo del movimiento 3D del ojo. El EKF es un algoritmo que permite la estimación de estados en sistemas dinámicos no lineales. En este caso, el EKF convierte las mediciones 2D de las posiciones del centro de la pupila y los cuatro reflejos especulares en estimaciones 3D de la traslación y rotación del ojo. Esto se hace utilizando un modelo de aceleración constante, que asume que los movimientos del ojo siguen un patrón predecible basado en su aceleración.

Para implementar el EKF, se define un vector de estado a posteriori $\hat{x}(t)$, que incluye la traslación, rotación, velocidades de traslación y rotación, así como la aceleración de traslación y rotación del ojo. Este enfoque mejora la robustez del sistema, permitiendo que el seguimiento ocular sea preciso incluso cuando se producen movimientos impredecibles de la cabeza o el ojo. Las ecuaciones de estado y estimación se desarrollan para actualizar continuamente este vector, permitiendo un seguimiento preciso y en tiempo real de la mirada del usuario, las cuales son:

$$\hat{x}(t) = \hat{x}(t)^- + K(t)(y(t) - h(\hat{x}(t)^-)) \quad (3.7)$$

$$P(t) = (\phi(\Delta t)P(t-1)\phi(\Delta t)^T + U) * \frac{\partial h}{\partial x(t)} \hat{x}(t) - (I - K(t)) \quad (3.8)$$

en donde $y(t)$ es el vector de medición actual, $K(t)$ es la ganancia de Kalman, $h(\hat{x}(t)^-)$ es la predicción de la medida del vector de estado, $\hat{x}(t)^-$ es el estimador a priori del vector de estado, mientras que $y(t) - h(\hat{x}(t)^-)$ es el residual que indica la discrepancia entre medición actual y la predicción. [20]

El sistema de seguimiento ocular propuesto por Park fue evaluado con 120 usuarios, cada uno proporcionando datos a partir de 23 puntos de vista diferentes. Dos pruebas distintas fueron realizadas para medir la precisión del sistema bajo diferentes condiciones:

1. Cálculo del Error en el Punto de Vista con Movimiento de Cabeza

- En esta prueba, se permitió que los usuarios movieran la cabeza mientras se realizaba la estimación del punto de vista, pero se mantuvo el movimiento de los ojos fijo.
- Los resultados mostraron un error promedio de 0,48 cm, lo que indica una alta precisión en la estimación del punto de vista a pesar del movimiento de la cabeza.

1. Cálculo del Error en el Punto de Vista con Movimiento de Cabeza y Ojos

- En la segunda prueba, tanto la cabeza como los ojos de los usuarios pudieron moverse libremente. Esto proporcionó un escenario más desafiante para el sistema, al tener que compensar ambos tipos de movimiento simultáneamente.

- Aun así, el error promedio fue de 0,51 cm, apenas superior al registrado en la primera prueba.

Los resultados obtenidos demuestran que el movimiento de la cabeza no afecta significativamente la precisión del sistema propuesto. La diferencia mínima de 0,03 cm entre ambas pruebas sugiere que el sistema es robusto ante cambios en la posición de la cabeza, manteniendo un alto grado de exactitud incluso cuando se considera el movimiento combinado de la cabeza y los ojos.

Luego existen muchos otros metodos como el de T. Nagamatsu, J. Kamahara y N. Tanaka [13], el cual presenta un método de seguimiento de la mirada en 3D con una calibración sencilla, utilizando cámaras estéreo para mejorar la comunicación entre robots y humanos. Los autores proponen una técnica que facilita la estimación precisa de la mirada en un entorno tridimensional, empleando cámaras estéreo para capturar la posición de los ojos y la dirección de la mirada. Este enfoque simplifica el proceso de calibración, lo que es crucial para aplicaciones en las que la interacción eficiente y natural entre robots y humanos es esencial.

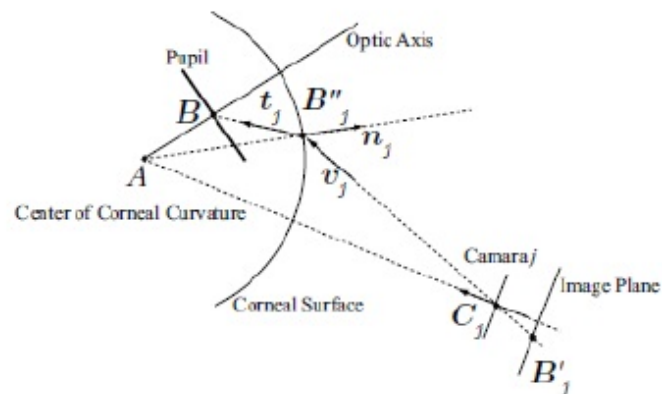


Figura 3.12: Modelo de T. Nagamatsu, J. Kamahara y N. Tanaka [13]

3.4. Calibración de la cámara

La formación de imágenes en el contexto de la visión por computadora puede modelarse matemáticamente como la proyección de una escena tridimensional (3D) sobre un plano bidimensional (2D). Este proceso de proyección perspectiva es fundamental para convertir las coordenadas de un punto en el espacio 3D en coordenadas correspondientes en el plano de la imagen 3.13.

Dentro de este modelo, se definen dos sistemas de coordenadas principales:

- El sistema de coordenadas del mundo, denotado como O
- El sistema de coordenadas de la cámara, representado por C

Un punto $M = [X, Y, Z]^T$ en el espacio 3D, expresado en el sistema de coordenadas euclidiano, es decir, es un marco matemático utilizado para describir

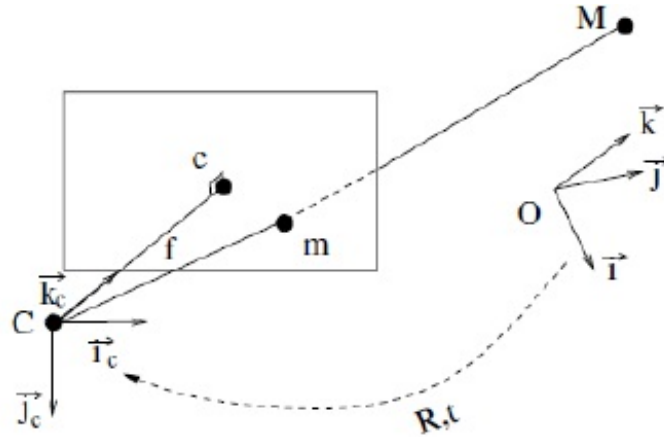


Figura 3.13: Modelo de escena 3D [14]

la posición de puntos en un espacio de cualquier dimensión (aunque comúnmente se refiere al espacio bidimensional y tridimensional).

Se proyecta en un punto $m = [u, v]^T$ en el plano de la imagen.

La transformación matemática entre el punto tridimensional M y su correspondiente proyección bidimensional m se describe mediante la ecuación $s\hat{m} = PM$, donde s es un factor de escala, \hat{m} y M son las coordenadas homogéneas de los puntos en 2D y 3D, respectivamente, y P es la matriz de proyección. Esta matriz de proyección P puede descomponerse en dos componentes:

$$P = K[R|t] \quad (3.9)$$

En esta descomposición, K es la matriz de parámetros intrínsecos de la cámara, que incluye factores como la distancia focal y los parámetros de desplazamiento del centro óptico en la imagen. La matriz $[R|t]$ es una matriz 3×4 que incorpora los parámetros extrínsecos, es decir, la rotación R y la traslación t que transforman las coordenadas del punto desde el sistema de coordenadas del mundo al sistema de coordenadas de la cámara.

Dado P , se puede descomponer en vectores:

$$p_i = [P_1, P_2, P_3]^T \quad (3.10)$$

donde cada P_i representa un vector que contribuye a la proyección del punto 3D en el plano de la imagen. Existen ciertas restricciones inherentes a esta proyección, las cuales se derivan de la naturaleza del modelo de cámara y que se aplican a la determinación precisa de las coordenadas proyectadas, siendo:

$$p_1^T K^{-T} K^{-1} p_2 = 0 \quad (3.11)$$

$$p_1^T K^{-T} K^{-1} p_1 = p_2^T K^{-T} K^{-1} p_2 \quad (3.12)$$

Por otro lado, la matriz K:

$$\begin{pmatrix} \alpha_u & \gamma & u_o \\ 0 & \alpha_v & v_o \\ 0 & 0 & 1 \end{pmatrix} \quad (3.13)$$

4. Estudio de necesidades

4.1. Identificación de necesidades

La identificación de necesidades es un paso crucial en el desarrollo de un seguidor ocular, ya que permite definir con precisión las características y funcionalidades que el dispositivo debe incorporar para ser efectivo y útil. Este proceso se llevó a cabo mediante una combinación de métodos cualitativos y cuantitativos, incluyendo encuestas y entrevistas a usuarios potenciales, tales como individuos con discapacidades motoras y profesionales de la salud. Además, se realizó una revisión exhaustiva de la literatura existente sobre tecnologías de seguimiento ocular para identificar las carencias y desafíos actuales en el mercado. Los principales hallazgos revelaron la necesidad de un sistema que ofrezca alta precisión y velocidad de respuesta, facilidad de uso, confort durante el uso prolongado y compatibilidad con diversas interfaces y dispositivos. Estos requerimientos fueron fundamentales para guiar el diseño y desarrollo del seguidor ocular, asegurando que se ajuste a las expectativas y necesidades reales de los usuarios.

Por lo tanto, es fundamental cuantificar la precisión, la velocidad y la posición de los dispositivos de seguimiento ocular disponibles en el mercado, ya que el desafío principal radica en proporcionar asistencia efectiva a las personas que requieren este tipo de tecnología para mejorar su calidad de vida.

4.2. Recursos disponibles

A continuación, se detallan qué tipo de recursos se incluyen, así como, ejemplos específicos para el proyecto de ratón de mirada.

En primer lugar, los **recursos humanos**, son considerados los mas importantes, pues los tutores y el estudiante han invertido sus capacidades y tiempo para el diseño y programación de ambos objetivos además de las horas invertidas en la investigación del mundo de los *eye trackers*.

En segundo lugar, los **recursos técnicos**, se diferencian en el hardware, el cual se refiere al propio ordenador o equipo de trabajo y el ratón de mirada o *eye tracker*. Por otro lado, el software, que se compone del sistema operativo del equipo y del lenguaje empleado para la programación, es decir, MATLAB.

En tercer lugar, los **recursos financieros**, en este caso han sido gracias a la Fundación Gema Canales, que se ha ofrecido voluntaría para probar ambos códigos con sus dispositivos de seguimiento ocular.

Por ultimo, los **recurso de infraestructura**, el cual es la conexión a internet,

destacar el acceso a la red universitaria que ha permitido obtener información de varios sitios webs.

4.3. Limitaciones y condicionantes

Estas pueden ser de naturaleza técnica, financiera, temporal, organizativa, entre otras. Aquí se detallan las principales limitaciones y condicionantes que pueden influir en el desarrollo y éxito del proyecto.

Por un lado, la disponibilidad del hardware, ya que no es sencillo obtener un ratón de mirada por lo tanto a puesto limitaciones para obtener pruebas y comparaciones. Además, la cámara del equipo utilizado para la simulación y pruebas del código no es de la mejor calidad, por ello se han encontrado problema con la precisión y calibración.

Por otro lado, la compatibilidad del software, en casos como la integración de bibliotecas, es decir, dificultades para integrar múltiples bibliotecas y herramientas de software, debido a que están desarrolladas en diferentes tipos de lenguajes de programación, como por ejemplo, problemas potenciales de compatibilidad entre bibliotecas de Python y C++ y la necesidad de optimizar el rendimiento del software desarrollado. Incluir también problemas de rendimiento al procesar grandes volúmenes de datos de seguimiento ocular en tiempo real.

Limitaciones temporales:

Al tratarse de un proyecto tan complejo y extenso, hay limitaciones en el tiempo disponibilidad del equipo, plazo estricto para la finalización del proyecto, lo que puede restringir la capacidad de realizar pruebas extensivas y refinamientos necesarios. Así como, coordinación de horarios entre los miembros del equipo, especialmente cuando existen otras obligaciones como pueden ser académicas o laborales.

Condicionantes técnicos y operativos:

La curva de aprendizaje no ha sido la mas óptima debido a que es un campo totalmente nuevo para el estudiante, incluso es un tema relativamente nuevo estos años, por lo tanto no ha sido ni es fácil aprender y obtener información sobre el seguimiento de mirada y los *eye trackers*. La necesidad de aprender y familiarizarse con nuevas herramientas, tecnologías y metodologías, lo que requerido tiempo adicional y ha afectado en el cronograma del proyecto.

La fiabilidad de los datos y su calidad, debido a la calidad del hardware ha afectado directamente a los datos obtenidos en el proyecto, tanto en el código empleado para cuantificar como en el propio código del ratón de mirada, ambos se han visto afectados por los factores externos como la iluminación, la posición del usuario y las condiciones del entorno.

5. Desarrollo del Código

En este apartado, se presenta el desarrollo y análisis de un conjunto de códigos en MATLAB destinados a la implementación y evaluación de sistemas de seguimiento ocular o *eye trackers*. La capacidad de monitorear con precisión los movimientos oculares es crucial para obtener datos fiables en dichos campos.

El código desarrollado tiene como objetivo, cuantificar y evaluar la precisión y eficacia de diferentes ratones de mirada disponibles en el mercado. Este análisis es fundamental para determinar qué dispositivos ofrecen las mejores características para el seguimiento ocular, considerando variables como la latencia, la precisión espacial y la robustez frente a diferentes condiciones de uso.

Todo el desarrollo fue llevado a cabo en MATLAB debido a su potencia en la manipulación de datos y señales, así como a su versatilidad para integrar algoritmos de procesamiento de imágenes y análisis estadístico, componentes esenciales en la construcción de sistemas de seguimiento ocular avanzados.

5.1. Definición de criterios y fases

En este proyecto, se ha desarrollado un experimento con el objetivo de evaluar y cuantificar las características técnicas de un dispositivo de seguimiento ocular *eyetracker*, tales como precisión, velocidad de respuesta, y precisión en la detección de la posición ocular. El experimento se fundamenta en la implementación de un código en MATLAB, que genera una trayectoria predefinida que sigue un círculo rojo en movimiento, el cual debe ser rastreado visualmente por los participantes.

Para asegurar la calidad y precisión de las mediciones, se realizaron ajustes preliminares en la cámara del portátil utilizado, enfocándose en la resolución y la exposición. Estos parámetros fueron optimizados para capturar una imagen clara y nítida de los ojos de los usuarios, asegurando que los datos recogidos sean representativos y minimizando la interferencia de factores externos.

El estudio involucró a cinco voluntarios, quienes participaron en la prueba siguiendo la trayectoria del círculo rojo con su mirada. Con el fin de obtener un conjunto de datos robusto y reducir el impacto de posibles errores humanos, cada participante realizó tres grabaciones independientes. Este enfoque permite analizar la consistencia y reproducibilidad de las mediciones del *eyetracker* y proporciona una base estadística sólida para evaluar sus capacidades.

Las grabaciones obtenidas se utilizarán para calcular métricas clave del *eyetracker*, tales como la precisión en la estimación de la posición ocular, la velocidad de seguimiento del movimiento ocular, y la fiabilidad general del dispositivo. Al analizar estos datos, se podrá determinar el rendimiento del

eyetracker bajo condiciones controladas y establecer un perfil detallado de sus capacidades técnicas.

Concretamente, se ha dividido el proceso en 3 códigos o scripts distintos, donde cada uno de ellos completa una función para llegar a resultados lo mas óptimos posibles.

5.2. Diseño, implementación y resultado de los algoritmos

A continuación para una explicación mas clara y precisa, se pueden observar la explicación de los 3 scripts del proceso llevado a cabo mediante un diagrama de bloques, además de la explicación por partes del propio código adjuntando pruebas de lo que se genera durante el proceso.

5.2.1. Primer script:

El código es un script de MATLAB que realiza varias tareas relacionadas con la adquisición de imágenes de una cámara, procesamiento de imágenes en tiempo real y la grabación de un vídeo de una duración aproximadamente de 1 minuto, donde se ha de seguir un círculo rojo que realizar un recorrido determinado. Se observa el diagrama de bloques en la imagen [5.1](#)

En la primera parte se inicializa y configura, es decir:

```
1
2
3 nombre = 'Prueba';
4 ensayo = 1;
5
6 cam = webcam(1);
7 cam.Resolution='1280x720';
8 cam.Exposure=-4;
9 Pix_SS = get(0,'screensize');
10 W = Pix_SS(3); H = Pix_SS(4);
11 M = 100;
12 Pix_SS2 = [M M W-M*2 H-M*2];
13 q = linspace(0,2*pi,100);
14 rH = W/8; rV = H/3;
15 x = W/2+rH*cos(q);
16 y = H/2+rV*sin(q);
```

Extracto de código 5.1: Configuración de la cámara y variables

En esta parte del código, las dos primeras variables **nombre** y **ensayo**, se usan para generar el nombre del archivo de vídeo. Luego se configura la cámara del

Diagrama de flujo

Sergio López | September 3, 2024



Figura 5.1: Diagrama de bloques 1

dispositivo para la resolución y exposición adecuada en función de donde se realice la simulación.

Más adelante, **Pix_SS** obtiene el tamaño de la pantalla, y **'W'** y **'H'** guardan el ancho y alto de la pantalla respectivamente. Con **Pix_SS2** se define un área dentro de la pantalla con un margen **'M'** de 100 píxeles en cada lado, y por último, se trata de la generación de coordenadas para una elipse, con **'q'** se definen los ángulos para un círculo, y con **'rH'** y **'rV'** definen los radios horizontal y vertical de la elipse, por

otro lado, 'x' e 'y' contienen las coordenadas de los puntos que forman la elipse.

La segunda parte del programa cuantificador, es para configurar y crear una ventana de figura.

```
1
2 f1 = 1;
3 figure;
4 set(gca,'XTick',[]); set(gca,'YTick',[]);
5 set(gca,'XColor','None'); set(gca,'YColor','None');
6 axis([0 W 0 H]);
7 set(gcf,'Position',Pix_SS); hold on;
8 set(gcf,'MenuBar','none');
```

Extracto de código 5.2: Configuración y creación de la ventana

Se desactivan los ejes ('XTick', 'YTick') así como los colores de estos ('XColor', 'YColor'). La ventana se ajusta al tamaño de la pantalla con 'set(gcf, 'Position', Pix_CC)' y se deshabilita la barra de menú.

El siguiente paso es generar la elipse con un bucle de 5 segundos donde el usuario deberá colocar la cabeza, para así realizar una captura de la imagen cuando se proceda a seguir la trayectoria con los ojos del círculo rojo que aparece cuando finaliza el bucle.

```
1
2 k = 1;
3 tic
4 t=0;
5 ElapsedTime = 5;
6 while (t < ElapsedTime)
7     clf;
8     set(gcf,'Position',Pix_SS2); hold on;
9     snapshot2 = snapshot(cam);
10    [hi,wi,ci]=size(snapshot2);
11    snapshot1 = imresize(snapshot2,W/wi);
12    snapshot1 = fliplr(snapshot1);
13    imshow(snapshot1); hold on;
14    patch(x,y,[0.8 0.8 0.2],'Facealpha',0.2,'LineWidth',2,'EdgeColor',[1
15    1 0]);
16    msg1='Place your face in the marked area';
17    msg2 = sprintf('Tiempo restante: %4.2f segundos',ElapsedTime-t);
18    text(W/2.5,H/10,msg1,'FontSize',14,'Color',[1 1 0]);
19    text(W/2.45,H/7,msg2,'FontSize',14,'Color',[1 1 0]);
20    k = k + 1;
21    drawnow;
22    t = toc;
end
```

Extracto de código 5.3: Configuración y generación de la elipse

Con `'ElapsedTime'` se define la duración del bucle de la elipse, `'snapshot2'` captura la imagen de la cámara, `'imresize'` ajusta el tamaño de esta y finalmente `'fliplr'` la voltea horizontalmente.

Mas adelante, se muestra la imagen procesada y se dibuja la elipse semitransparente amarilla la cual es el área de referencia para que se coloque el usuario, empleando `'patch'`, como se puede apreciar en la imagen 5.2.

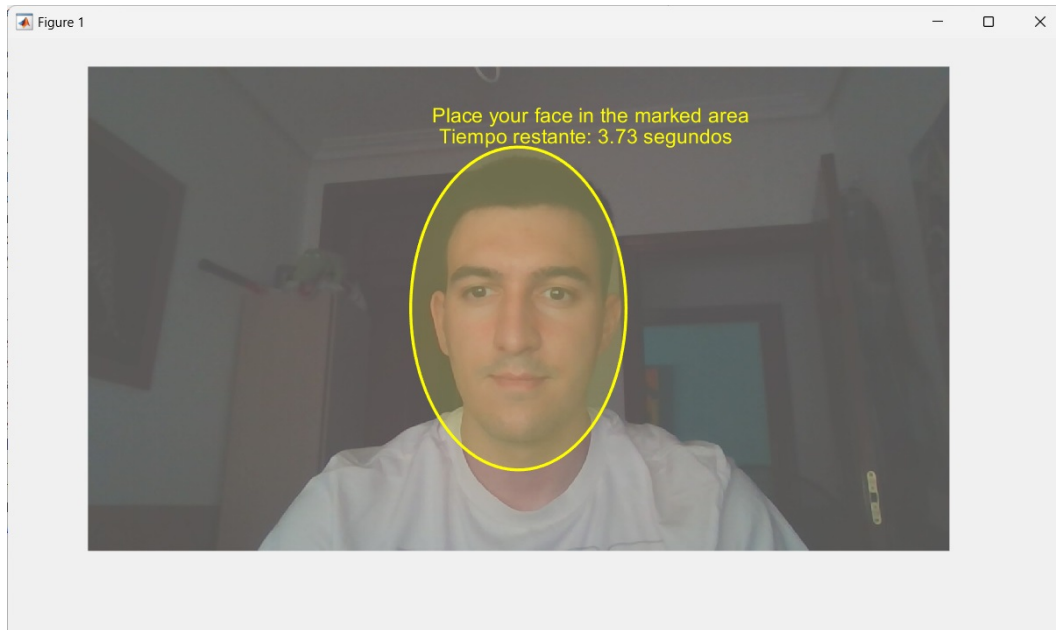


Figura 5.2: Generación de la elipse

Antes de comenzar con el proceso de seguimiento se crea una cuenta regresiva de 3 segundos, indicando que se debe seguir el círculo rojo solamente con los ojos. La pantalla que se observa es la que se ha capturado en la imagen 5.3.

```
1
2     cla;
3 set(gcf,'Position',Pix_SS); hold on;
4 axis([0 W 0 H]);
5
6 FS = 264;
7 for i=1:3
8     cla; axis([0 W 0 H]);
9     text(W/2-FS/2,H/2,num2str(3-i+1),'FontSize',264,'Color',[0 0 0]);
10    msg3 = 'Please, follow the red circle with your eyes';
11    text(W/3.2,3*H/4,msg3,'FontSize',24,'Color',[0 0 0]);
12    pause(1);
13 end
```

Extracto de código 5.4: Configuración de la cuenta regresiva

Por último se trata de configurar como se va a mover dicho círculo. Se han capturado distintos puntos del trayecto del círculo en las imágenes 5.4, 5.5, 5.6.

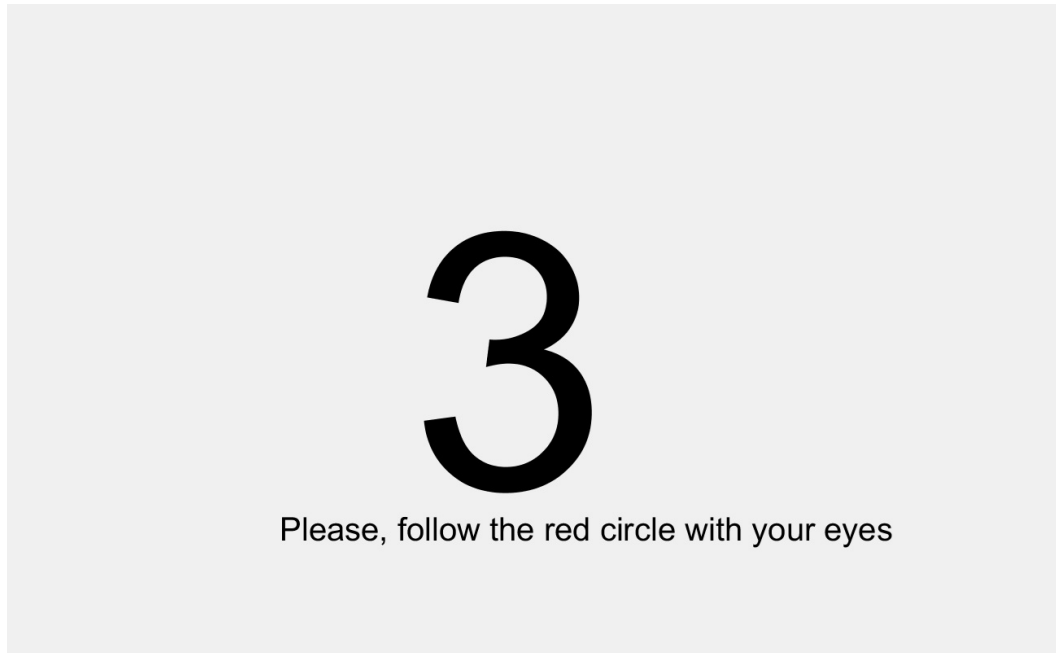


Figura 5.3: Cuenta Regresiva

```

1
2     Rc = 100; N = 100; velInv = 5;
3 XPoints1=[W/2.0 0+Rc 0+Rc W-Rc W-Rc 0+Rc W/2.0];
4 YPoints1=[H/2.0 H-Rc 0+Rc 0+Rc H-Rc H-Rc H/2.0];
5
6 XPoints2=[W/2.0 W/4+Rc W/4+Rc W-W/4-Rc W-W/4-Rc W/4+Rc W/2.0];
7 YPoints2=[H/2.0 H-H/4-Rc H/4+Rc H/4+Rc H-H/4-Rc H-H/4-Rc H/2.0];
8
9 XPoints = [XPoints1 XPoints2];
10 YPoints = [YPoints1 YPoints2];
11
12 vname = sprintf(' %s_ %d.mp4', nombre, ensayo);
13
14 v = VideoWriter(vname, 'MPEG-4');
15 open(v);
16 for i=1:length(XPoints)-1
17     dX = abs(XPoints(i+1)-XPoints(i));
18     dY = abs(YPoints(i+1)-YPoints(i));
19     d = ceil(max([dX dY])/velInv);
20     x = linspace(XPoints(i),XPoints(i+1),d);
21     y = linspace(YPoints(i),YPoints(i+1),d);
22     for j=1:d
23         cla; axis([0 W 0 H]);
24         drawcircle('Center',[x(j) y(j)],'Radius',Rc,'Color',[1 0 0]);
25         plot(x(j),y(j),'ro','MarkerSize',12,'LineWidth',3);
26         I = snapshot(cam);
27         writeVideo(v,I);

```

```
28     drawnow;
29     end
30 end
31 close(v);
32 close all;
```

Extracto de código 5.5: Configuración de la trayectoria del círculo rojo

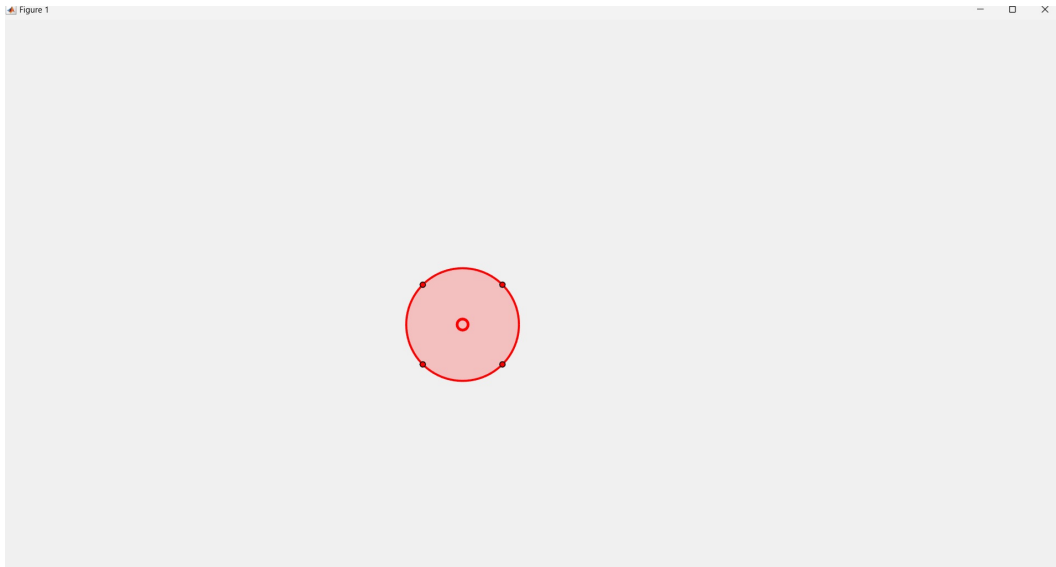


Figura 5.4: Ubicación 1

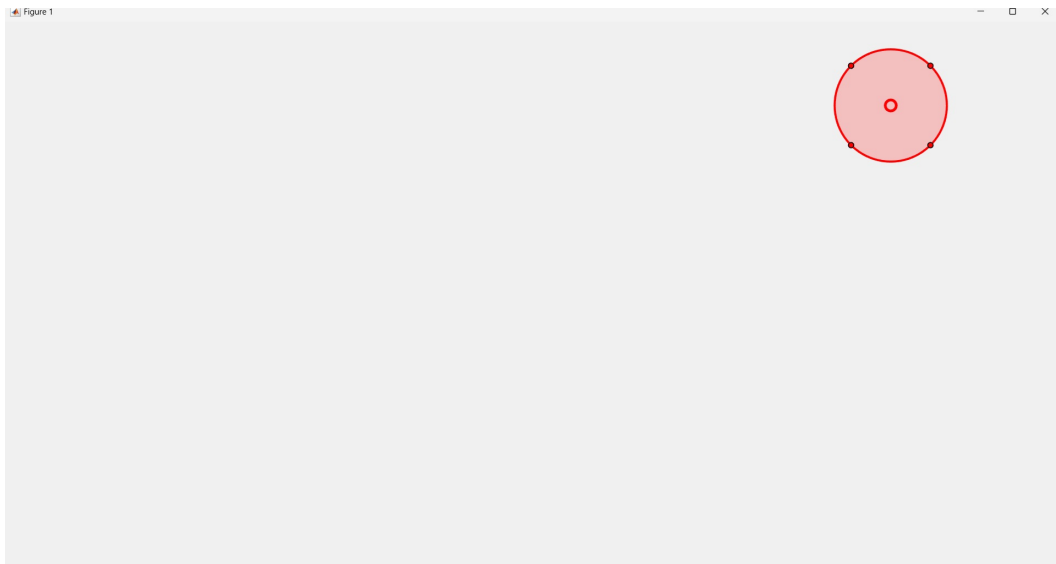


Figura 5.5: Ubicación 2

Se definen las trayectorias que seguirá el círculo con `'XPoints'` y `'YPoints'`, se crea un objeto de vídeo y se abre para la escritura en `'VideoWriter'`. Luego, dentro de un bucle, se calcula la trayectoria entre dos puntos sucesivos y se mueve el círculo rojo a lo largo de esta trayectoria. Cada vez que el círculo cambie de posición, `'snapshot(cam)'` toma una captura y la guarda en el vídeo generado. Finalmente se cierra el objeto de vídeo y la figura al acabar todo el recorrido.

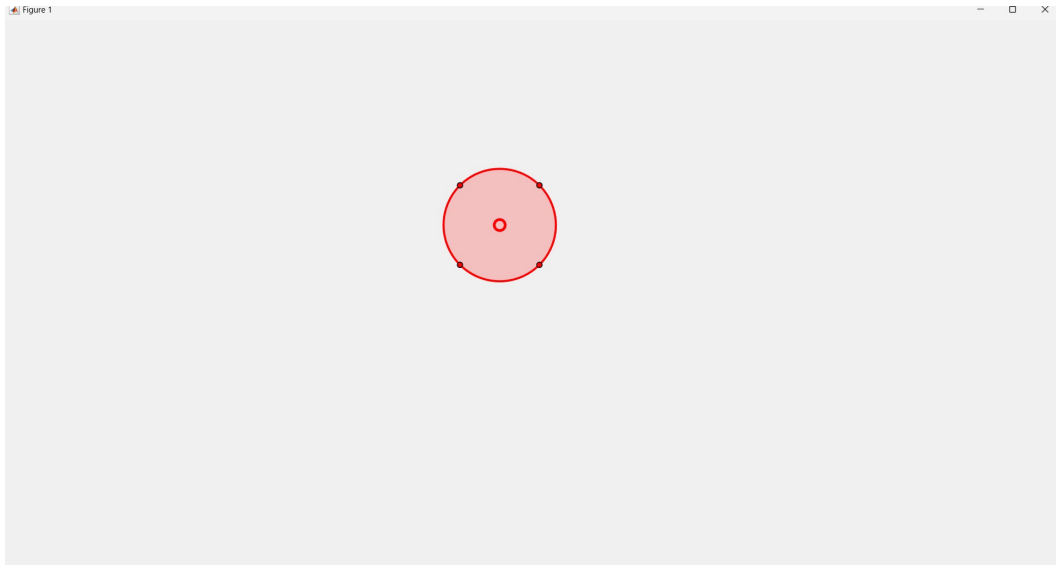


Figura 5.6: Ubicación 3

Quedando como resultado un vídeo que se empleará para cuantificar las características del *eye tracker*, se puede observar un frame del vídeo en la imagen 5.7. Este proceso se ha llevado a cabo con 5 usuarios, generando cada uno 3 vídeos ya que siempre existe el error humano al seguir la trayectoria del círculo.

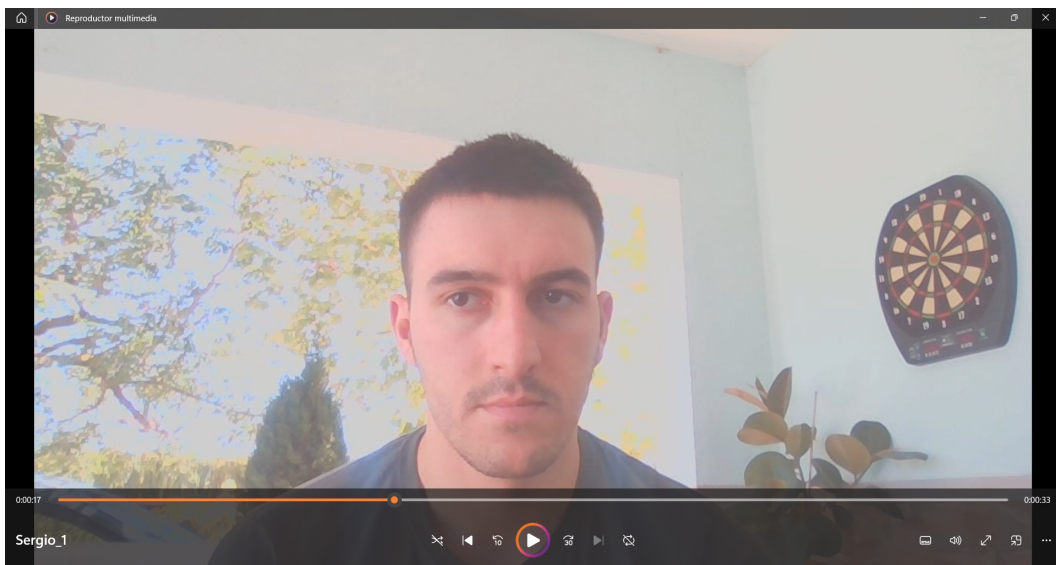


Figura 5.7: Frame del vídeo generado

5.2.2. Segundo script:

De nuevo en la imagen 5.8 y 5.9 se aprecia el diagrama de bloques, el cual se ha dividido en dos debido al tamaño, además se procede a la explicación del código con sus respectivos ejemplos de simulación. En este proceso se procesan todos y cada uno de los frames de los vídeos de los usuarios, si se detecta algún rostro en el vídeo se generará un rectángulo amarillo cuyo título será 'cara', mientras que

si se detectan los ojos, se generará un rectángulo azul cuyo título será 'ojos'. En caso contrario, es decir, que no se detecte o bien el rostro del usuario o los ojos, no aparecerá el rectángulo correspondiente. En función de cuantas veces se detecten el rostro y los ojos durante todos los frames de los vídeos se podrá sacar una estadística de la precisión y calidad del seguidor de mirada.

En cuanto a la explicación detallada del código, se desglosa en diferentes partes. En primer lugar, se crea un objeto, '**VideoReader**' para leer el video con el nombre que se haya guardado. Además de inicializarse el detector de caras y ojos, con `'vision.CascadeObjectDetector()'` y `'vision.CascadeObjectDetector('EyePairSmall')'`.

```
1
2 v = VideoReader("Sergio_2.mp4");
3 faceDetector = vision.CascadeObjectDetector();
4 detector1 = vision.CascadeObjectDetector('EyePairSmall');
```

Extracto de código 5.6: Inicialización de objetos y variables

Más adelante se configura el bucle de procesamiento y figura, en el cual se podrán ajustar los frames a procesar. Una vez decididos los frames, comienza la detección de caras. Con '`bboxF = step(faceDetector, A)`' se detectan las caras en el fotograma determinado, en este caso el A, seleccionando la cara mas grande en caso de que haya mas de una con la estructura condicional. Por último, con '`face = imcrop(A, bboxF)`', se recorta la región del fotograma donde se encuentre la cara para optimizar la detección de ojos próximamente.

```
1
2 figure;
3 for i=1:100
4     clf;
5     A = readFrame(v);
6     bboxF = step(faceDetector, A);
7     if size(bboxF,1)>1
8         bboxF = bboxF(size(bboxF,1),:);
9     end
10    face = imcrop(A,bboxF);
```

Extracto de código 5.7: Configuración del bucle y detección de caras

El siguiente paso, una vez recortada la región, se comienza con la detección de ojos que funciona de la misma manera que el rostro.

```
1
2 bboxE = step(detector1, face);
3     if size(bboxE,1)>1
4         bboxE = bboxE(size(bboxE,1),:);
5     end
6     eye = imcrop(face,bboxE);
```

Extracto de código 5.8: Detección de ojos

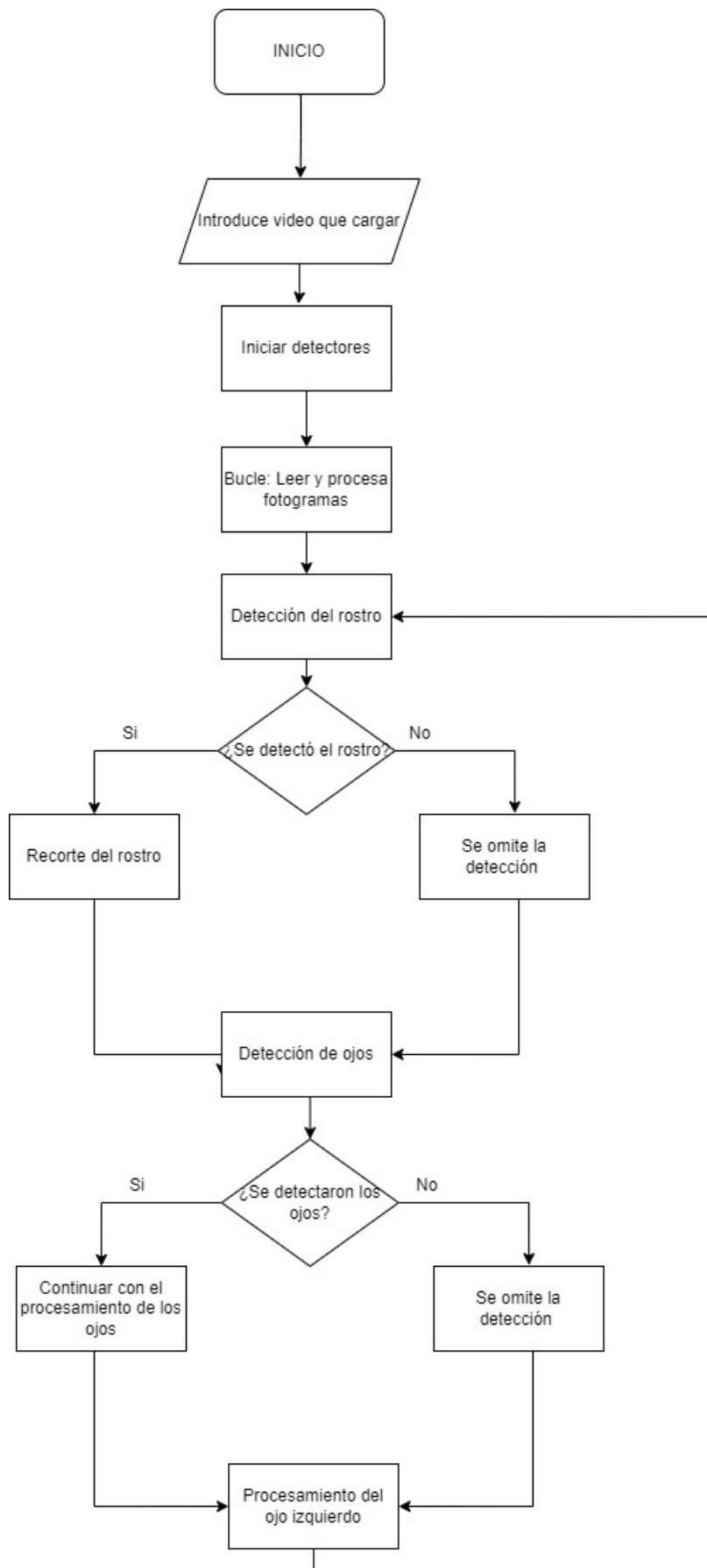


Figura 5.8: Diagrama de bloques 2. Primera parte

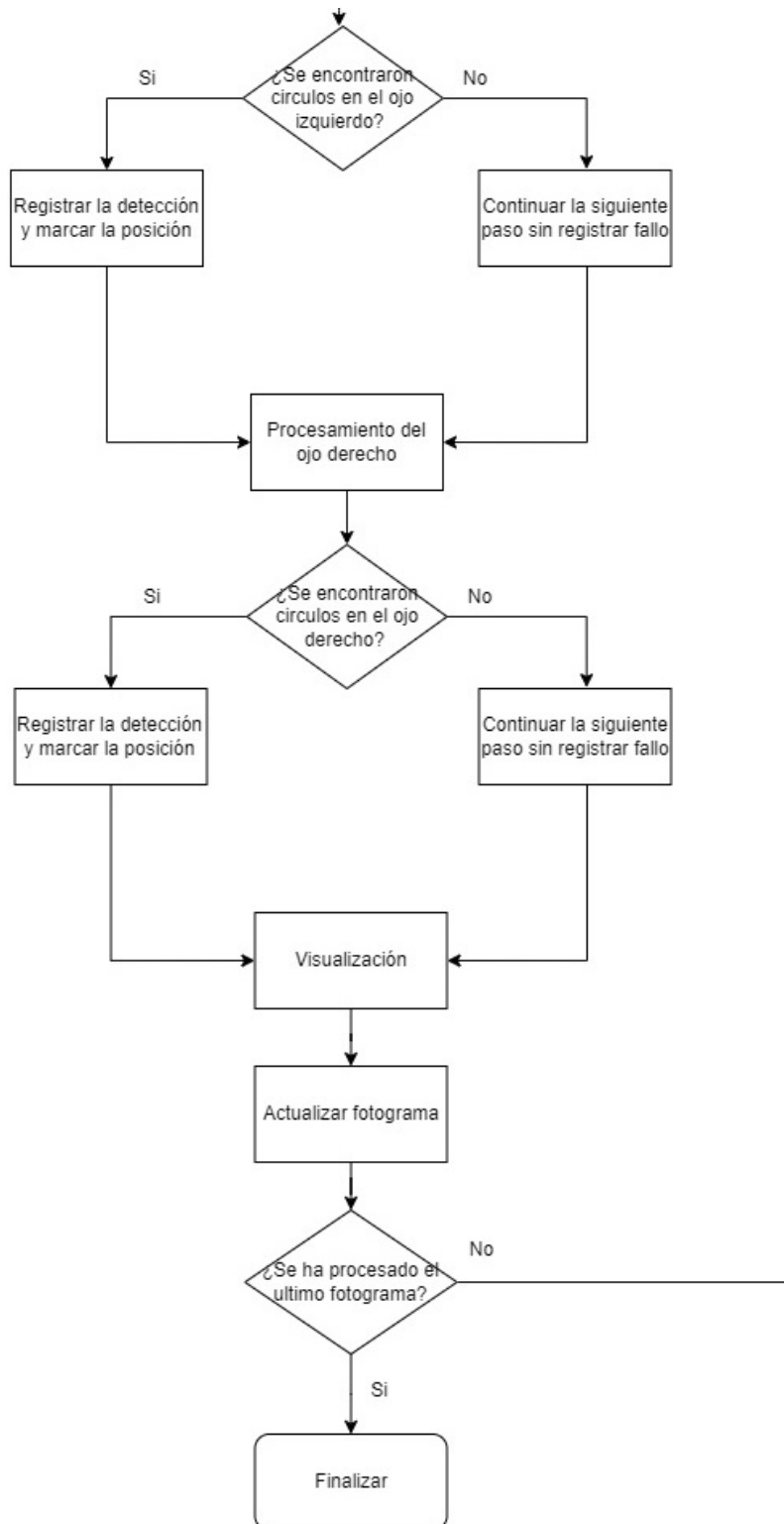


Figura 5.9: Diagrama de bloques 2. Segunda parte

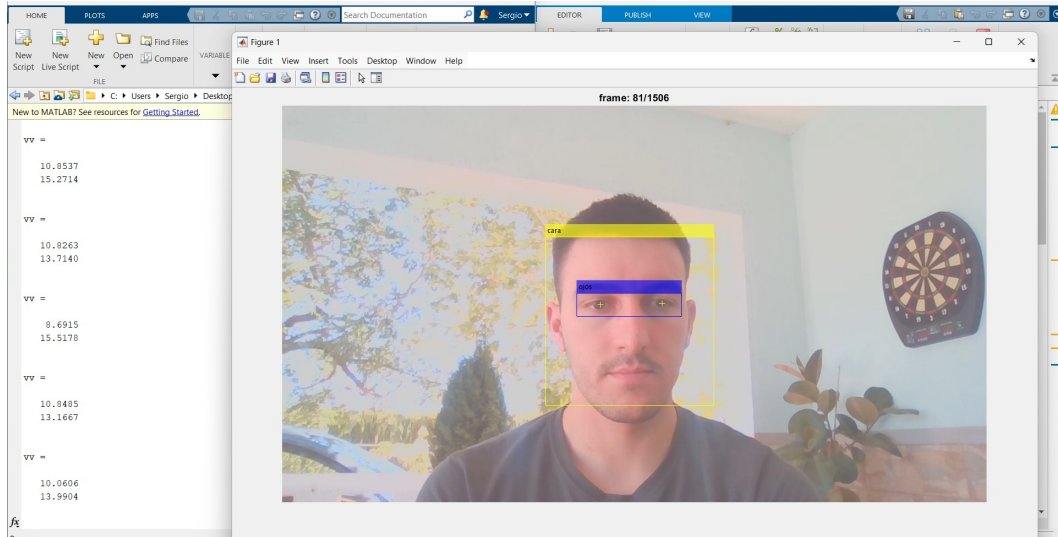


Figura 5.10: Frame 81/100. Script 2. Primer usuario

A continuación se diferencia el ojo izquierdo del derecho, los cuales se procesan por separado. Además se observa en 5.11 una estadística simulada en la cual se escribe un 0 por cada error al detectar la pupila derecha.

```
falloR =

Columns 1 through 15
    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1

Columns 16 through 30
    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1

Columns 31 through 45
    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1

Columns 46 through 60
    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1

Columns 61 through 75
    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1

Columns 76 through 90
    1    1    1    1    1    1    1    1    1    1    1    0    1    1    1

Columns 91 through 100
    1    1    1    1    1    1    1    1    1    1
```

Figura 5.11: Estadística de fallo en 100 frames usuario 1. Script 2

```
1
2 eyeL = eye(:,1:floor(size(eye,2)/2),:);
3   RL = eyeL(:,1);   GL = eyeL(:,2);   BL = eyeL(:,3);
4   eyeLg = im2gray(eyeL);
```

```

5 eyeLg2 = scale255(eyeLg);
6 [cL, rL] = imfindcircles(eyeLg2,[8 20], 'ObjectPolarity', 'dark', '
Sensitivity', 0.90);
7 if (size(cL,1)>1)
8     meanSi = size(eyeLg2,2)/2;
9     vv = abs(cL(:,1)-meanSi);
10    [Cent,iCent] = min(vv);
11    cL = cL(iCent,:);
12    rL = rL(iCent);
13 end

```

Extracto de código 5.9: Ojo izquierdo

Se separa la mitad izquierda del rostro con `'eyeL = eye(:,1:floor(size(eye,2)/2),:);'`. Luego se extren los canales de color RGB (rojo, verde y azul), se pasa a una escalas de grises y con una función auxiliar se escala la imagen, `'eyeLg2 = scale255(eyeLg);'`, para una mejora de los colores detectados.

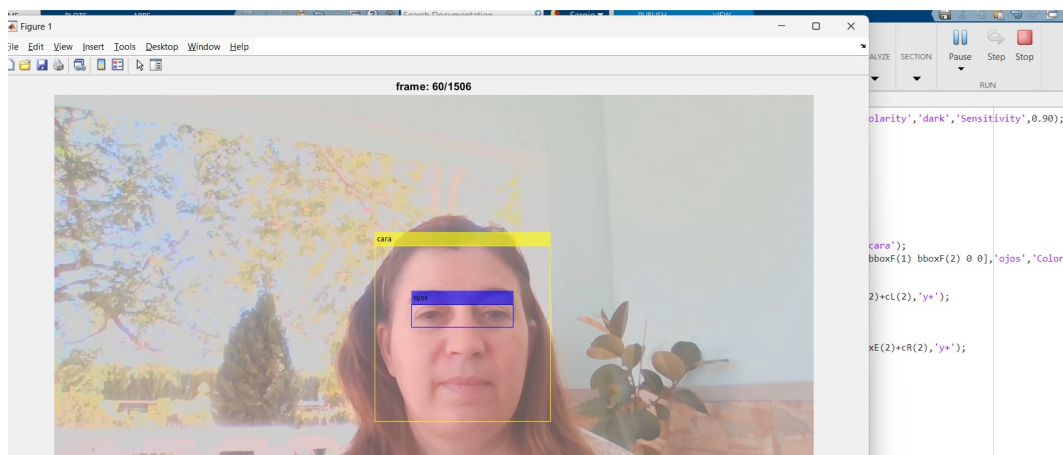


Figura 5.12: Frame 60/100. Script 2. Cuarto usuario

```

1
2 function [Ic]=scale255(I)
3
4 M = max(max(I)); m = min(min(I));
5 Ia = I-m;
6 Ib = double(Ia)/double(M-m)*255;
7 Ic = uint8(Ib);
8
9 end

```

Extracto de código 5.10: Ojo izquierdo

Los círculos, que serían las pupilas del usuario en el video se detectan en escala de grises mediante `[cL, rL] = imfindcircles(eyeLg2,[8 20], 'ObjectPolarity', 'dark', 'Sensitivity', 0.90);`, ajustando la sensibilidad en función de lo óptima que sea la simulación, eligiendo siempre el círculo más cercano al centro del ojo con la función `if (size(cL,1)>1)`. El ojo derecho se realiza de forma idéntica, pero con la mitad derecha de la cara.

```

1
2 Eo= ceil(size(eye,2)/2);
3 eyeR = eye(:,Eo:end,:);
4 eyeRg = im2gray(eyeR);
5 eyeRg2 = scale255(eyeRg);
6 [cR, rR] = imfindcircles(eyeRg2,[8 20], 'ObjectPolarity', 'dark', '
Sensitivity', 0.90);
7 if (size(cR,1)>1)
8     meanSi = size(eyeRg2,2)/2;
9     vv = abs(cR(:,1)-meanSi);
10    [Cent,iCent] = min(vv);
11    cR = cR(iCent,:);
12    rR = rR(iCent);
13 end

```

Extracto de código 5.11: Ojo derecho

En la imagen 5.13 se puede observar un ejemplo de como falla en gran porcentaje la detección del ojo izquierdo.

```

falloL =
Columns 1 through 17
    0    0    0    1    1    0    0    0    0    0    0    0    1    1    0    1    0
Columns 18 through 34
    1    1    0    0    0    1    0    1    1    0    0    0    1    0    0    0    1
Columns 35 through 51
    0    1    0    0    0    0    1    1    1    0    0    0    0    0    0    1    0
Columns 52 through 68
    1    1    0    1    1    1    1    1    0    1    0    1    1    1    1    0    1
Columns 69 through 85
    1    1    1    1    1    0    1    1    1    1    1    1    1    1    1    1    0
Columns 86 through 100
    1    0    1    0    1    1    1    1    1    1    1    1    1    1    1

```

Figura 5.13: Estadística de fallo en 100 frames usuario 4. Script 2

Por último, se definen distintas anotaciones cuando se detecta la cara, los ojos, así como la función para mostrar el fotograma. Además, se añaden condiciones para

mayor claridad y las dos funciones que permiten obtener una estadística de las veces que se ha detectado el ojo izquierdo y derecho de forma independiente.

```
1
2 A = insertObjectAnnotation(A,"rectangle",bboxF,'cara');
3 A = insertObjectAnnotation(A,"rectangle",bboxE+[bboxF(1) bboxF(2) 0 0
4 ],'ojos','Color','blue');
5 imshow(A); hold on;
6 if (size(cL,1)>0)
7     plot(bboxF(1)+bboxE(1)+cL(1),bboxF(2)+bboxE(2)+cL(2),'y+');
8 end
9 if (size(cR,1)>0)
10    plot(bboxF(1)+bboxE(1)+Eo+cR(1),bboxF(2)+bboxE(2)+cR(2),'y+');
11 end
12 if (size(cL,1)>0)
13    falloL(i)=1;
14 end
15 if (size(cR,1)>0)
16    falloR(i)=1;
17 end
18 msg = sprintf('frame: %d/%d',i,v.NumFrames); title(msg);
19 drawnow;
```

Extracto de código 5.12: Anotaciones y calculo de fallos

5.2.3. Tercer script:

Por último, se trata el tercer script, cuyo diagrama se puede observar en 5.14 y 5.15 se aprecia el diagrama de bloques, el cual se ha dividido en dos debido al tamaño, además se procede a la explicación del código con sus respectivos ejemplos de simulación. En este proceso, se trata de detectar la pupila en los vídeos generados en el primer script durante todos los frames para obtener una estadística de cuando se detectan, en función de la luz o posición del usuario. Si la pupila es detectada sin problema, se obtiene como resultado un 1, además de generarse un círculo rojo en ella en la ventana de la simulación. En cambio si no se detecta el resultado será un 0 y ningún tipo de círculo. La imagen resultante una vez simulados los frames sera de ambos ojos por separado con una escala de grises en 3 tonos distintos, la cual se ha ido ajustando para optimizar el resultado de identificar la pupila del usuario en el vídeo.

El código del tercer script es similar al anterior, consiguiendo como resultado 3 imágenes de ambos ojos en las cuales se genera un círculo rojo en cada frame del vídeo si esta se detecta de forma lo suficientemente clara. Primero, se crea un objeto, **'VideoReader'** para leer el con el nombre que se haya guardado. Además de inicializarse el detector de caras y ojos, con **'vision.CascadeObjectDetector()'** y **'vision.CascadeObjectDetector('EyePairSmall')'**. Observando este código en el extracto 5.13 tras los diagramas de bloques.

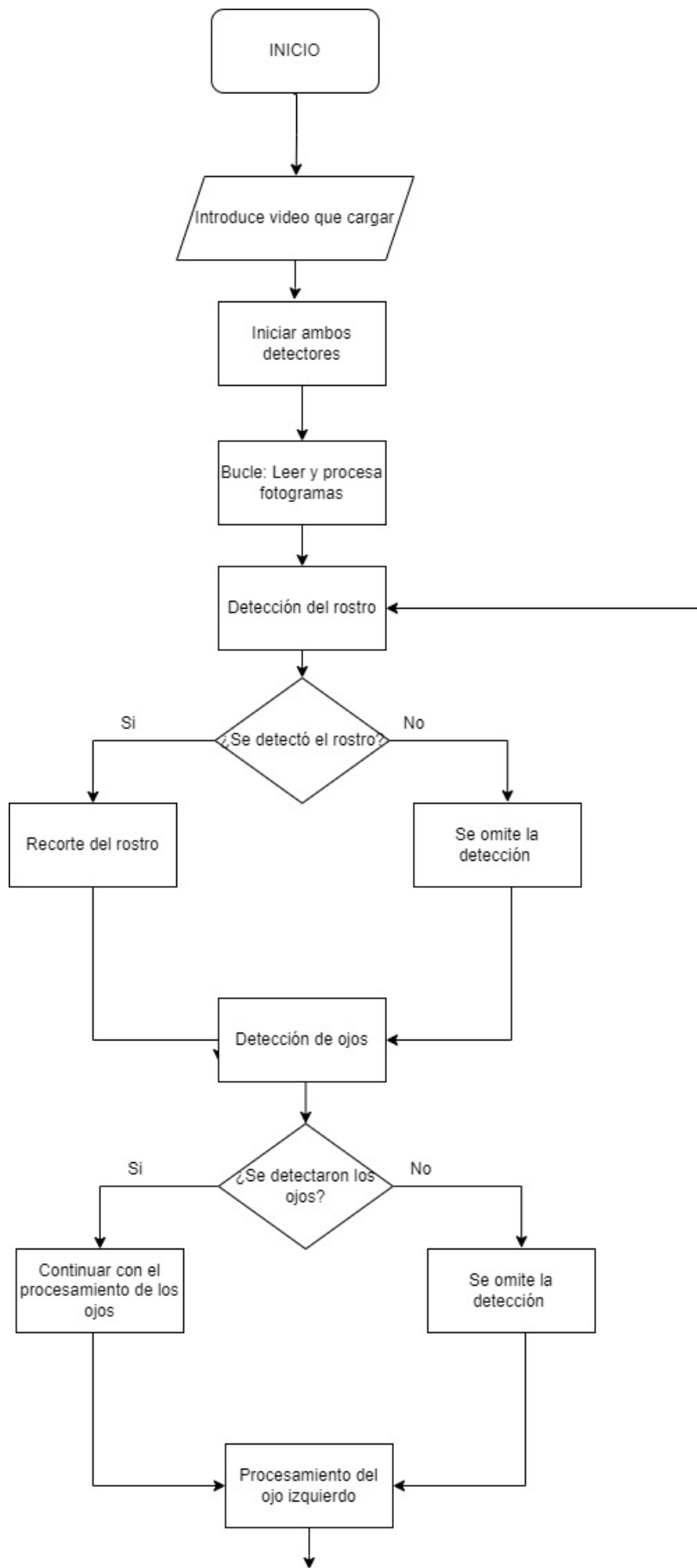


Figura 5.14: Diagrama de bloques 3. Primera parte

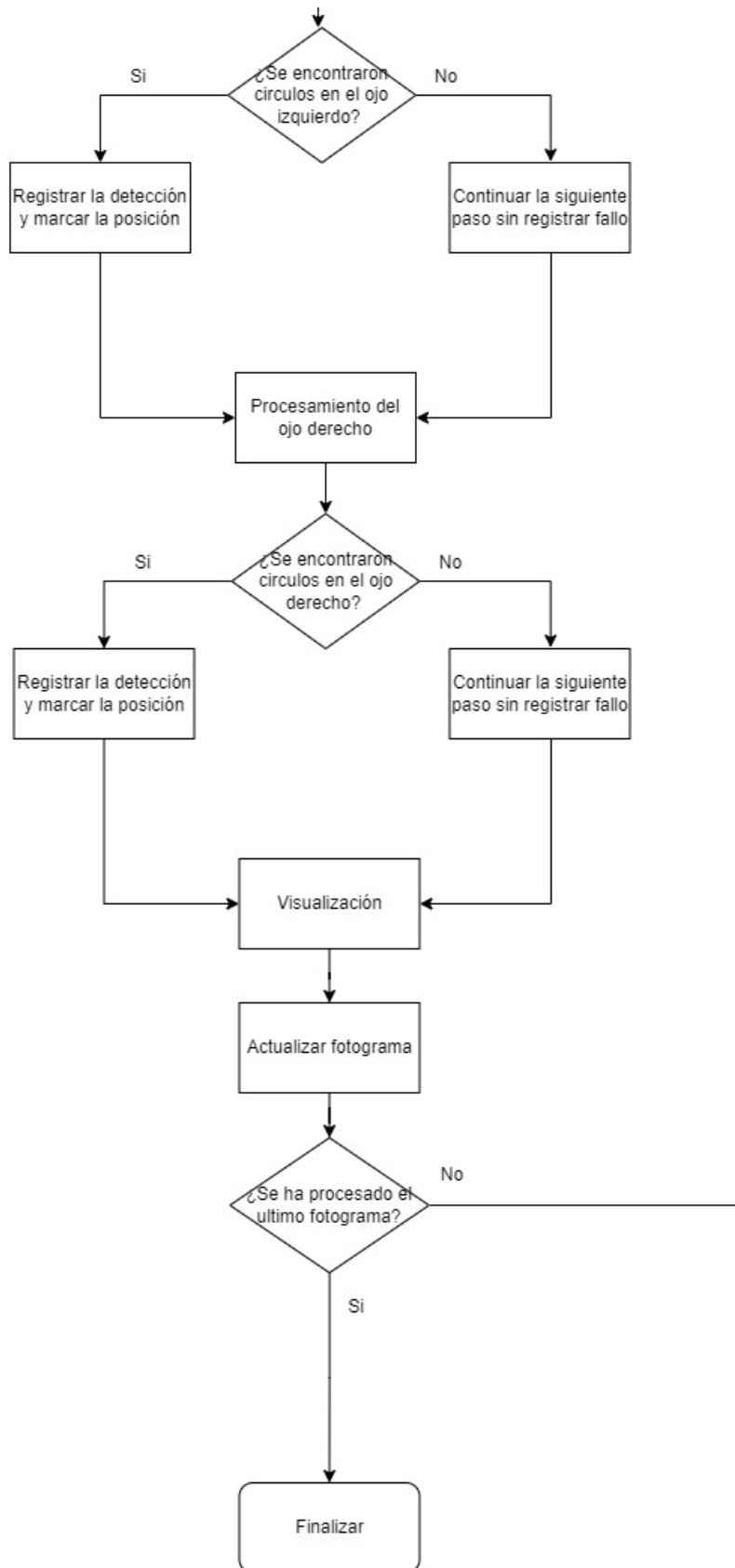


Figura 5.15: Diagrama de bloques 3. Segunda parte

```

1 def hello_world():
2
3 v = VideoReader("Sergio_2.mp4");
4 faceDetector = vision.CascadeObjectDetector();
5 detector1 = vision.CascadeObjectDetector('EyePairSmall');

```

Extracto de código 5.13: Inicialización de objetos y variables

A continuación el proceso sigue siendo como en el anterior script salvo por una pequeña diferencia, se configura el bucle de procesamiento y figura, en el cual se podrán ajustar los frames a procesar, en esta caso, en lugar de 100 frames como antes, se procesan todos los del video, eso es posible con `'for i=1:v.NumFrames'`. Una vez decididos los frames, comienza la detección de caras. Con `'bboxF = step(faceDetector, A)'` se detectan las caras en el fotograma determinado, en este caso el A, seleccionando la cara mas grande en caso de que haya mas de una con la estructura condicional. Por último, con `'face = imcrop(A, bboxF)'`, se recorta la región del fotograma donde se encuentre la cara para optimizar la detección de ojos próximamente.

```

1
2 figure;
3 for i=1:v.NumFrames
4     clf;
5     A = readFrame(v);
6     bboxF = step(faceDetector, A);
7     if size(bboxF,1)>1
8         bboxF = bboxF(size(bboxF,1),:);
9     end
10    face = imcrop(A,bboxF);

```

Extracto de código 5.14: Configuración del bucle y detección de caras

Una vez realizado el recorte y la detección de caras, se inicia la detección de los ojos, está parte del código es idéntica al script anterior por lo tanto no varía nada ni en código ni en explicación. Una vez detectados comienza el procesamiento de los ojos de forma individual.

```

1
2
3 bboxE = step(detector1, face);
4     if size(bboxE,1)>1
5         bboxE = bboxE(size(bboxE,1),:);
6     end
7     eye = imcrop(face,bboxE);
8
9 %ojoizquierdo
10
11 eyeL = eye(:,1:floor(size(eye,2)/2),:);
12     RL = eyeL(:, :, 1);  GL = eyeL(:, :, 2);  BL = eyeL(:, :, 3);

```

```

13     eyeLg = im2gray(eyeL);
14     eyeLg2 = scale255(eyeLg);
15     [cL, rL] = imfindcircles(eyeLg2,[8 20], 'ObjectPolarity', 'dark', '
Sensitivity', 0.90);
16     if (size(cL,1)>1)
17         meanSi = size(eyeLg2,2)/2;
18         vv = abs(cL(:,1)-meanSi);
19         [Cent,iCent] = min(vv);
20         cL = cL(iCent,:);
21         rL = rL(iCent);
22     end
23
24 %ojoderecho
25
26 Eo= ceil(size(eye,2)/2);
27     eyeR = eye(:,Eo:end,:);
28     eyeRg = im2gray(eyeR);
29     eyeRg2 = scale255(eyeRg);
30     [cR, rR] = imfindcircles(eyeRg2,[8 20], 'ObjectPolarity', 'dark', '
Sensitivity', 0.90);
31     if (size(cR,1)>1)
32         meanSi = size(eyeRg2,2)/2;
33         vv = abs(cR(:,1)-meanSi);
34         [Cent,iCent] = min(vv);
35         cR = cR(iCent,:);
36         rR = rR(iCent);
37     end

```

Extracto de código 5.15: Detección y procesamiento de los ojos

En la siguiente parte comienza lo que diferencia al tercer script, se trata de la forma en la que se muestran los resultados a la hora de detectar la pupila en los distintos vídeos de los usuarios, mostrando tanto el ojo izquierdo como el derecho.

Se muestran en 3 tonalidades grisáceas distintas en formato de vídeos mientras avanzan los frames, gracias a este formato se observa claramente cuando se produce el fallo en la detección de la pupila en los distintos vídeos.

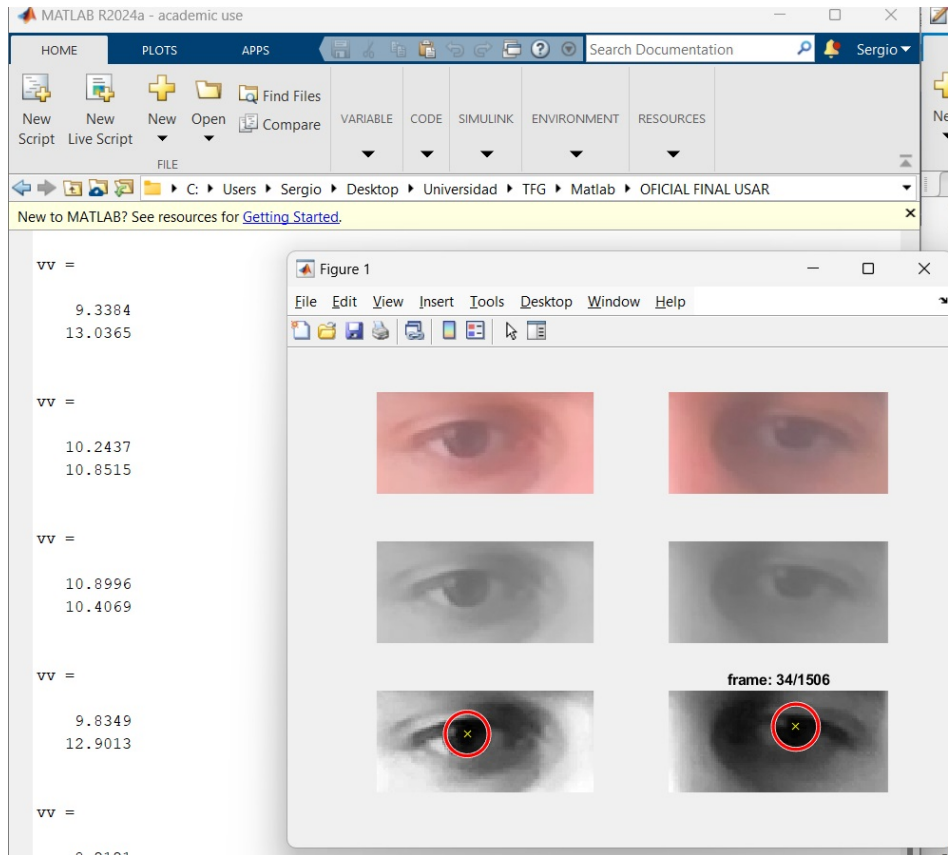


Figura 5.16: Frame 34 del vídeo. Script 3. Usuario 1

```

1
2 subplot(3,2,1);imshow(eyeL); hold on;
3 subplot(3,2,2);imshow(eyeR); hold on;
4 subplot(3,2,3);imshow(eyeLg); hold on;
5 subplot(3,2,4);imshow(eyeRg); hold on;
6 subplot(3,2,5);imshow(eyeLg2); hold on;
7 if (size(cL,1)>0)
8     plot(bboxF(1)+bboxE(1)+cL(1),bboxF(2)+bboxE(2)+cL(2), 'y+');
9 end
10 if (size(cR,1)>0)
11     plot(bboxF(1)+bboxE(1)+Eo+cR(1),bboxF(2)+bboxE(2)+cR(2), 'y+');
12 end
13
14 if (size(cL,1)>0)
15     h = viscircles(cL,rL);
16     plot(cL(1),cL(2), 'xy');
17     falloL(i)=1;
18 end

```

Extracto de código 5.16: Visualización de subgráficos

Por último se muestra la imagen procesada con la escala de grises con el círculo rojo si se ha detectado la pupila y se actualiza el título de los frames conforme avanza el proceso.

```
1  
2 msg = sprintf('frame: %d/%d',i,v.NumFrames); title(msg);  
3 drawnow;  
4 pause
```

Extracto de código 5.17: Actualización del título y pausa

```
falloR =  
  
Columns 1 through 15  
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  
  
Columns 16 through 30  
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  
  
Columns 31 through 45  
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  
  
Columns 46 through 60  
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  
  
Columns 61 through 75  
1 1 1 1 1 1 1 1 1 0 0 0 0 0 0  
  
Columns 76 through 90  
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
  
Columns 91 through 100  
0 0 0 0 0 0 0 0 0 1
```

Figura 5.17: Estadísticas de fallo en 100 frames usuario 1. Script 3

6. Pruebas y Resultados

6.1. Estudio de los resultados obtenidos en ambos scripts

En este apartado se realiza el estudio de los resultados obtenidos en algunos de los casos simulados en el proyecto.

Por ejemplo, se va a analizar los datos obtenidos tanto en el script 2 como en el 3 por los distintos usuarios. aunque hay un total de 15 vídeos de pruebas con diferentes usuarios se han seleccionado dos, siendo el mas preciso y el menos para contrastar los casos extremos a la hora de usar el código desarrollado. Gracias a estas simulaciones, se puede hacer una estimación de los fallos del *eye tracker* afectando en su proceso varios factores.

Se va a observar como en función de la posición, luminosidad e incluso tamaño de los ojos de la persona el programa detecta mas o menos pupilas, añadiendo por lo tanto mas ceros al resultado, ya que un 1 significa que ha sido detectada y un 0 que no lo ha sido.

CÓDIGO 2:

Una vez se ha grabado como el estudiante sigue con la mirada el recorrido del circulo rojo, es decir, el primer script. Obteniendo los siguientes resultados al simularlo:

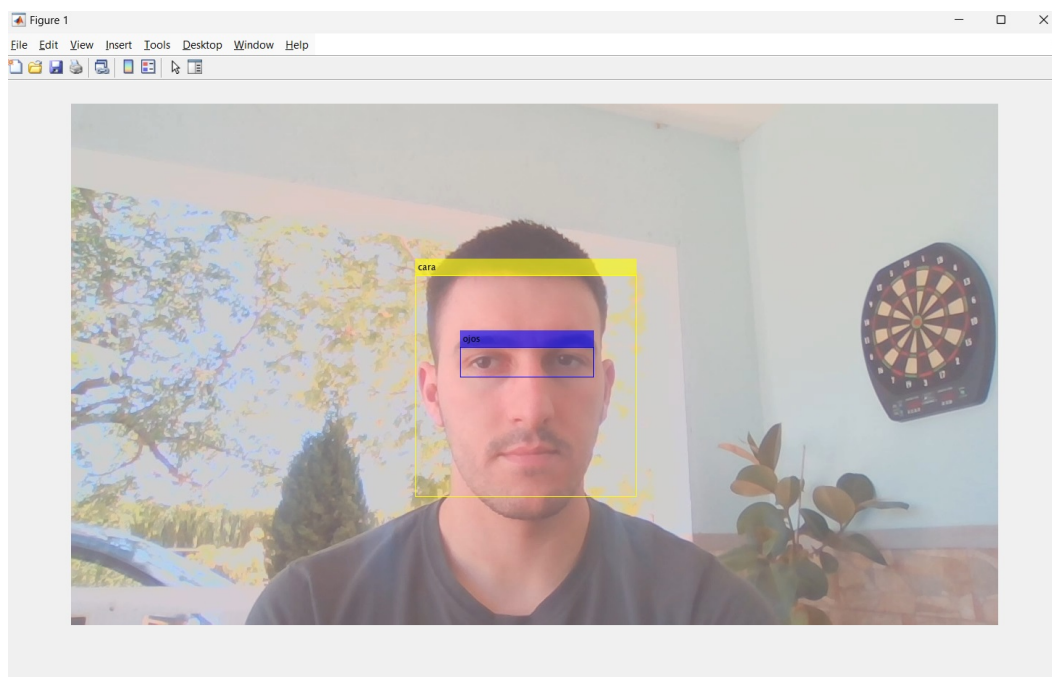


Figura 6.1: Ejemplo vídeo Sergio. Script 2.

En el extracto 6.1 se observa la cantidad de ceros que se han generado en dicho vídeo, dando como resultado un porcentaje de fallo de 1.33 % en el ojo derecho y 24.96 % en el ojo izquierdo. Evidentemente, estos resultados pueden ser mejorados aun más, por ejemplo en el ojo izquierdo son bastante notables, sin embargo, estos resultados superan las expectativas esperadas teniendo en cuenta de los recursos que se disponen durante el proyecto.

```
1
2 %% Contar ceros en los frames totales
3 totalZerosL = sum(falloL == 0); % Contar ceros en falloL (ojo izquierdo)
4 totalZerosR = sum(falloR == 0); % Contar ceros en falloR (ojo derecho)
5
6 %% Mostrar resultados
7 disp('Ceros totales en falloL (ojo izquierdo):');
8 disp(totalZerosL);
9
10 disp('Ceros totales en falloR (ojo derecho):');
11 disp(totalZerosR);
12 Ceros totales en falloL (ojo izquierdo):
13     368
14
15 Ceros totales en falloR (ojo derecho):
16     19
```

Extracto de código 6.1: Estadísticas Sergio Vídeo. Script 2.

La imagen 6.2 muestra una gráfica representativa de donde se falla en la detección de pupila.

Por otro lado, se ha realizado la prueba también otra voluntaria, dando unos resultados completamente distintos, gracias a estos dos voluntarios se tienen dos puntos de vista distintos de la precisión del código desarrollado, ya que en función de la situación en la cual se emplee y las características físicas del usuario los resultados cambian.

Se observará un descenso significativo de aciertos entre ambos casos debido a que simplemente por la luminosidad y la estructura ocular de la hermana del estudiante, el programa es incapaz de detectar durante todo el proceso la pupila de la usuario. El extracto de código 6.2 aporta información de cuantos ceros ha habido en total durante el proceso, con ello es posible sacar un porcentaje de fallo y éxito del programa.

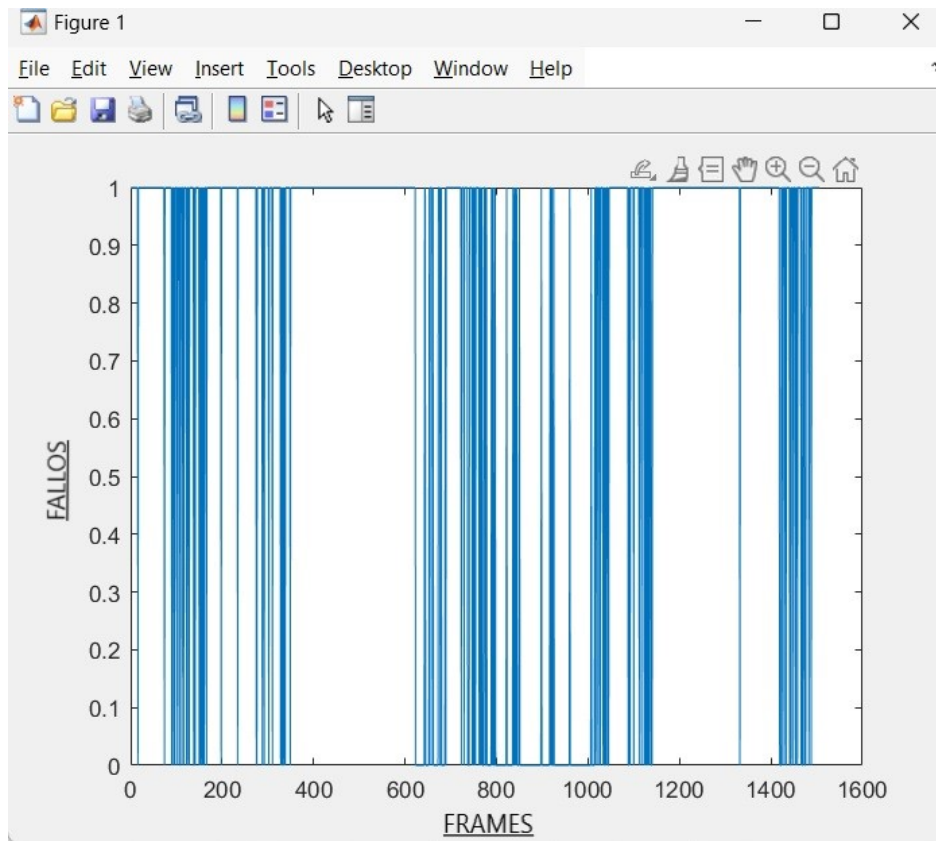


Figura 6.2: Gráfica de estadísticas ojo izquierdo Sergio

```

1
2 %% Contar ceros en los frames totales
3 totalZerosL = sum(falloL == 0); % Contar ceros en falloL (ojo izquierdo)
4 totalZerosR = sum(falloR == 0); % Contar ceros en falloR (ojo derecho)
5
6 %% Mostrar resultados
7 disp('Ceros totales en falloL (ojo izquierdo):');
8 disp(totalZerosL);
9
10 disp('Ceros totales en falloR (ojo derecho):');
11 disp(totalZerosR);
12 Ceros totales en falloL (ojo izquierdo):
13     245
14
15 Ceros totales en falloR (ojo derecho):
16     1260

```

Extracto de código 6.2: Estadísticas voluntaria

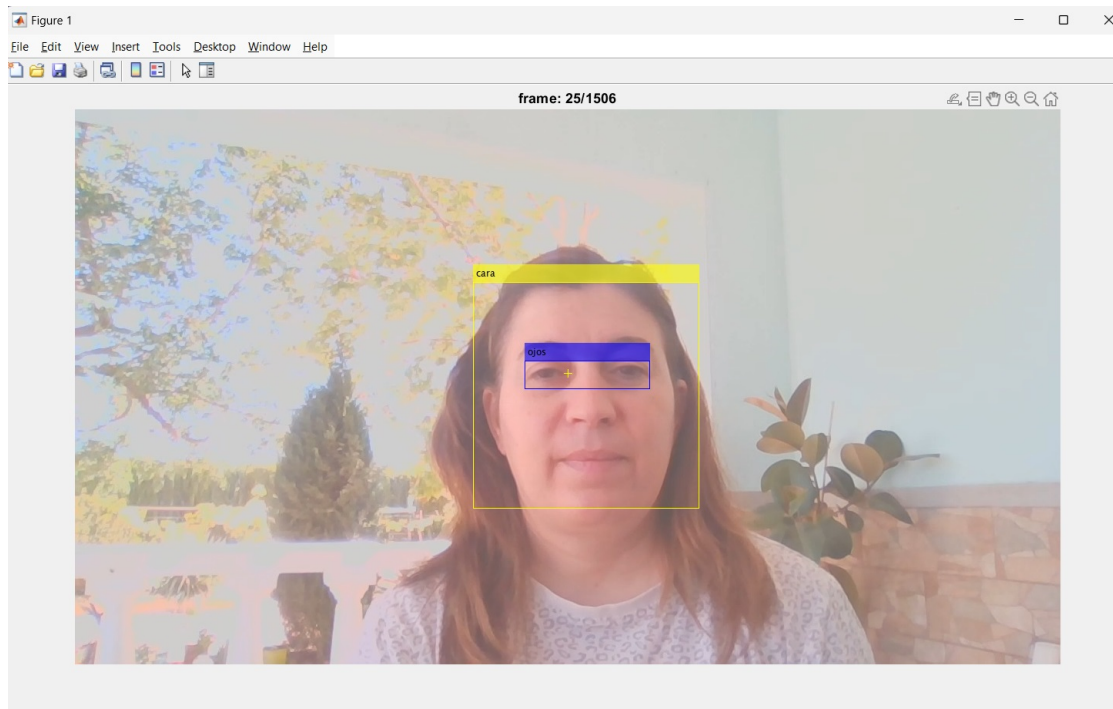


Figura 6.3: Ejemplo vídeo voluntaria. Script 2.

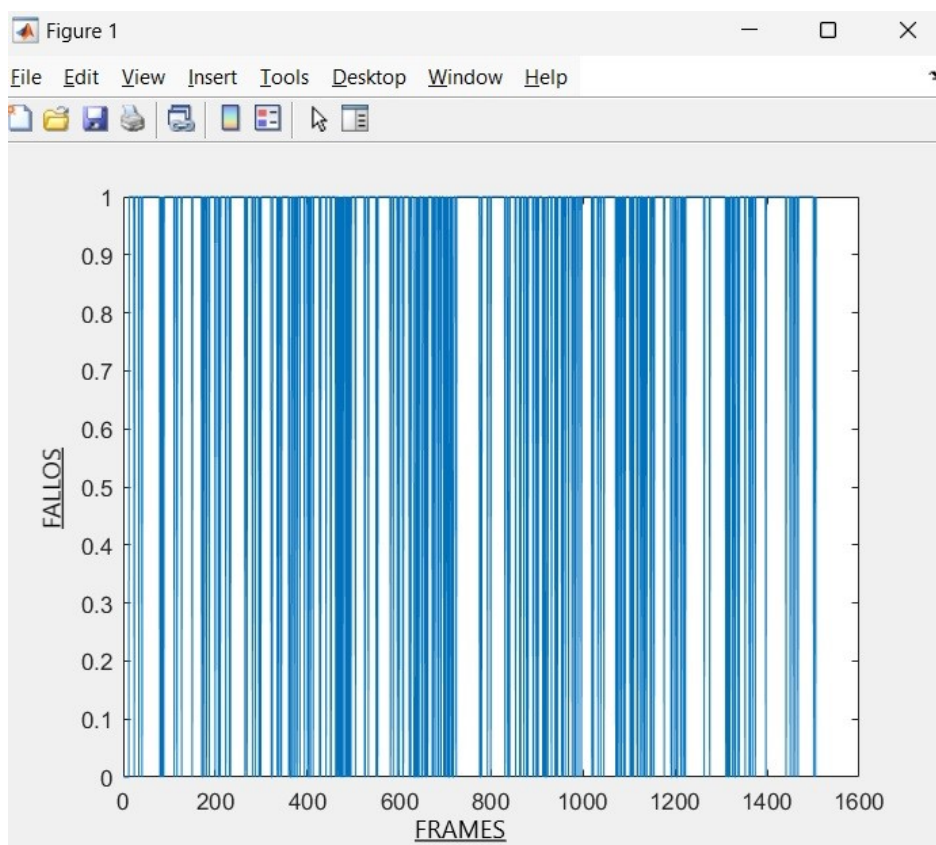


Figura 6.4: Gráfica ojo derecho voluntaria. Script 2.

No obstante, que hay un gran porcentaje de fallo, en concreto 83.66 % para el ojo derecho y 16.27 % para el ojo izquierdo, no es algo negativo, ya que se trata de un

tipo de estudio necesario, obteniendo información de que sin la suficiente apertura de ojos y luminosidad la precisión del programa no será todo lo óptima como podría ser.

```
Columns 781 through 795
  1   1   1   1   1   1   1   1   1   1   1   0   1   1   1
Columns 796 through 810
  1   1   1   0   1   1   1   1   1   1   1   1   1   1   1
Columns 811 through 825
  1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
Columns 826 through 840
  1   1   1   1   0   1   1   1   1   1   0   0   0   1   0
Columns 841 through 855
  1   1   1   1   1   1   1   1   1   1   1   1   0   1   1
Columns 856 through 870
  1   1   1   1   1   1   0   1   0   1   1   1   1   1   1
Columns 871 through 885
```

Figura 6.5: Extracto de detección ocular voluntaria. Ojo izquierdo.

Aún siendo la comparativa demasiado distinta, se deben considerar ambos resultados ya que aportan un tipo de información valiosa para la mejorar continua que existe en el ámbito de seguimiento ocular y visión artificial.

CÓDIGO 3:

Otro tipo de datos realizados son los que se obtienen con el script 3. El procedimiento es el mismo, se trata de simular y cuantificar los vídeos de los voluntarios.

Una vez se llega al frame final, se puede observar un ejemplo en la imagen 6.6, se procede a estudiar las estadísticas.

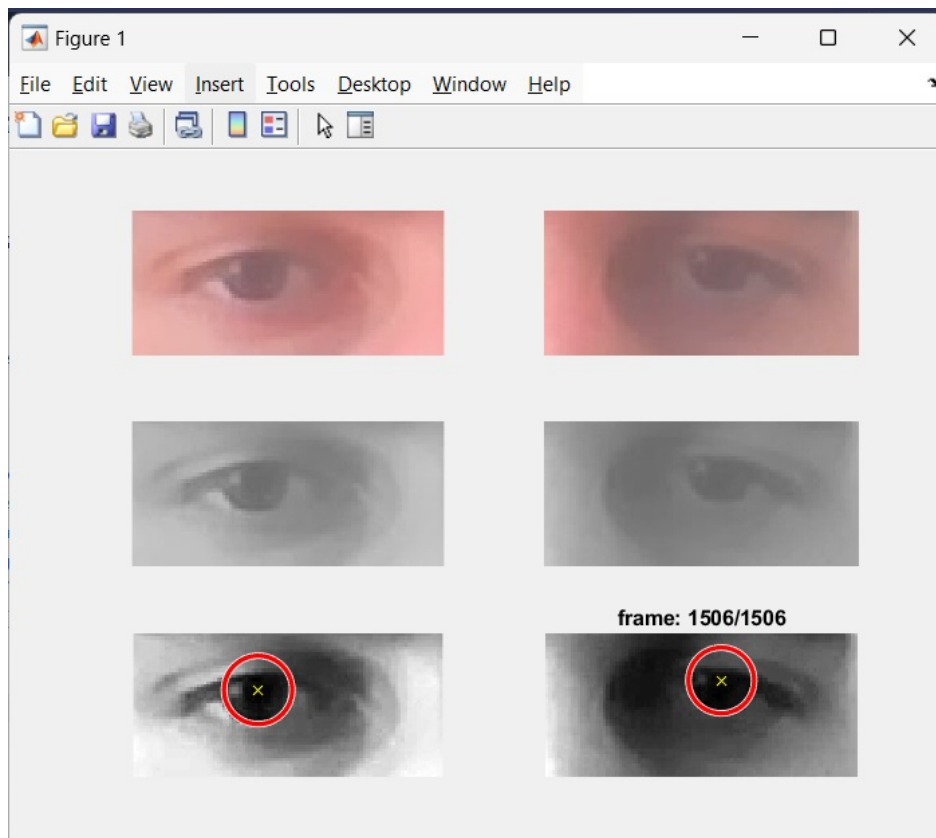


Figura 6.6: Ejemplo vídeo Sergio. Script 3.

```

1
2
3 %% Contar ceros en los frames totales
4 totalZerosL = sum(falloL == 0); % Contar ceros en falloL (ojo izquierdo)
5 totalZerosR = sum(falloR == 0); % Contar ceros en falloR (ojo derecho)
6
7 %% Mostrar resultados
8 disp('Ceros totales en falloL (ojo izquierdo):');
9 disp(totalZerosL);
10
11 disp('Ceros totales en falloR (ojo derecho):');
12 disp(totalZerosR);
13 Ceros totales en falloL (ojo izquierdo):
14     380
15
16 Ceros totales en falloR (ojo derecho):
17     29

```

Extracto de código 6.3: Estadísticas Sergio vídeo. Script 3.

Se puede ver en 6.3 que el porcentaje de fallo es similar al anterior simulación, es decir, entre el script 2 y 3 hay una variación mínima, por lo tanto se trata de un proceso exacto aunque su precisión no sea la mas óptima, aún. Reafirmando los

```

Columns 916 through 930
  1   1   1   1   1   1   1   1   1   1   1   1   1   0   1
Columns 931 through 945
  1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
Columns 946 through 960
  1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
Columns 961 through 975
  1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
Columns 976 through 990
  1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
Columns 991 through 1.005
  1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
Columns 1.006 through 1.020
  1   1   1   1   1   1   1   1   1   1   1   1   1   1   1

```

Figura 6.7: Ejemplo de extracto Sergio. Ojo derecho. Script 3.

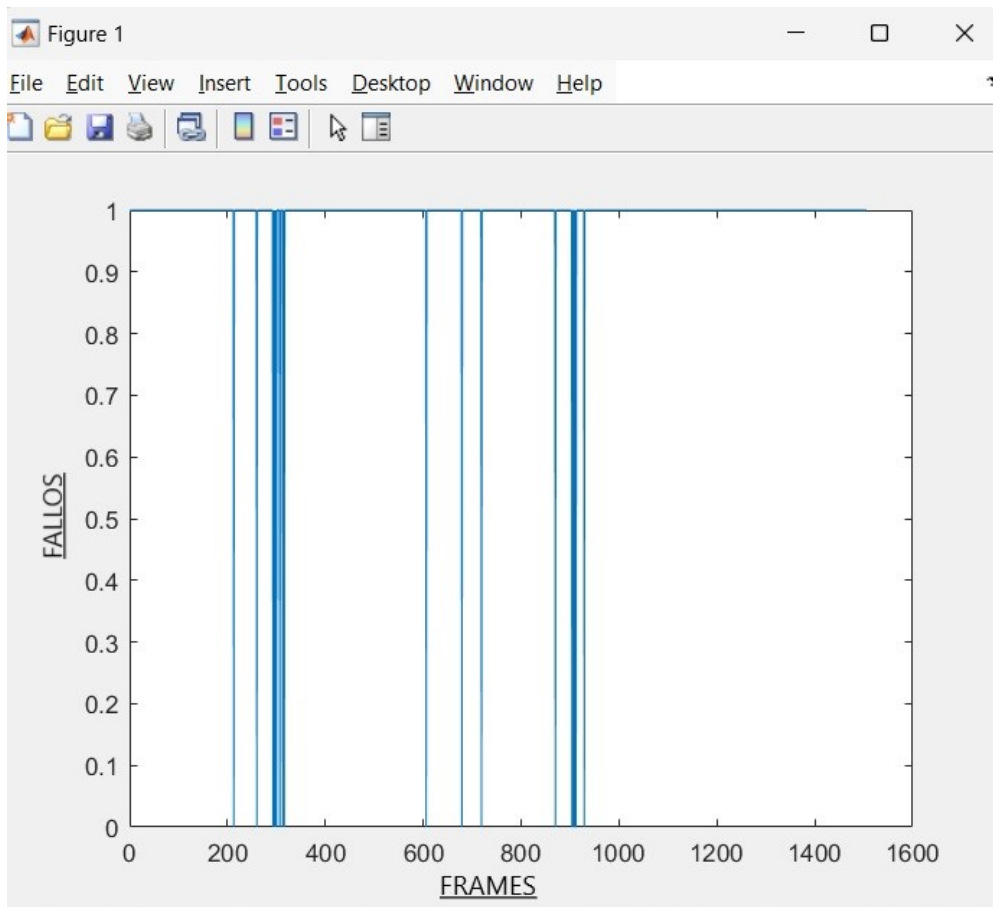


Figura 6.8: Gráfica ojo derecho Sergio. Script 3

resultados anteriores, el usuario se encuentra en un lugar y con unas facciones que hacer el estudio mas sencillo dando lugar a que se generen pocos ceros, por lo tanto, mas detecciones de su pupila durante el uso del seguidor de mirada.

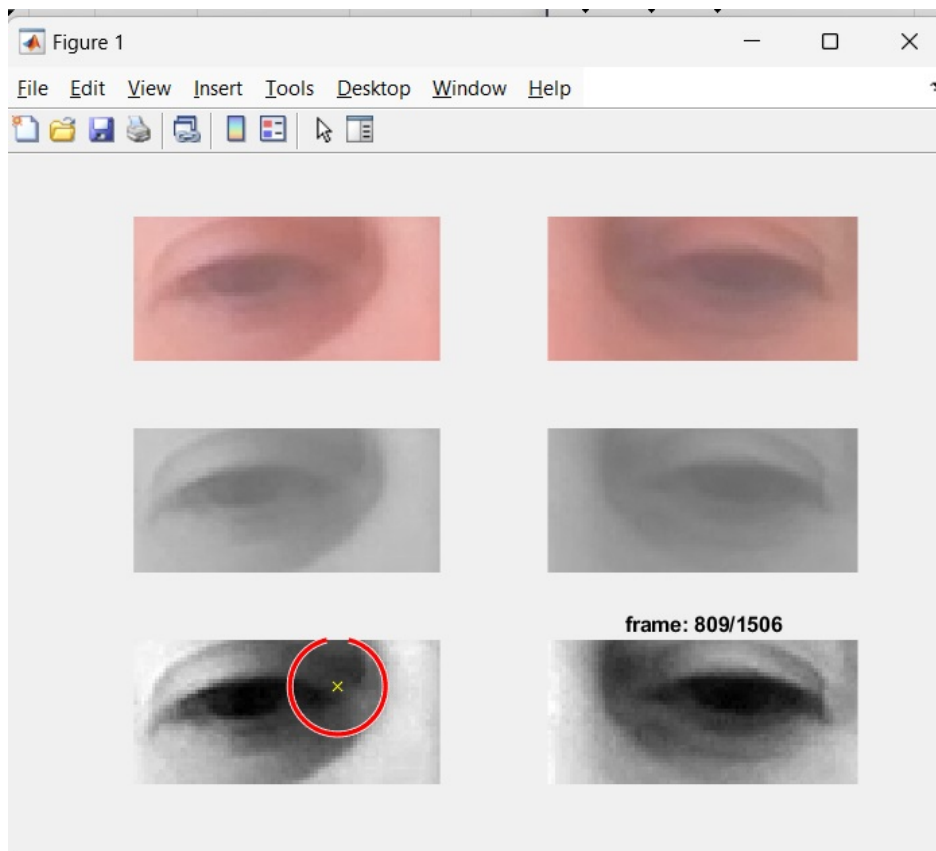


Figura 6.9: Ejemplo vídeo voluntaria. Script 3.

Por último, se muestran los resultados con la voluntaria anterior. Obteniendo de nuevo un número similar de ceros, como se aprecia en el extracto [6.4](#)

```

1
2 %% Contar ceros en los frames totales
3 totalZerosL = sum(falloL == 0); % Contar ceros en falloL (ojo izquierdo)
4 totalZerosR = sum(falloR == 0); % Contar ceros en falloR (ojo derecho)
5
6 %% Mostrar resultados
7 disp('Ceros totales en falloL (ojo izquierdo):');
8 disp(totalZerosL);
9
10 disp('Ceros totales en falloR (ojo derecho):');
11 disp(totalZerosR);
12 Ceros totales en falloL (ojo izquierdo):
13     245
14
15 Ceros totales en falloR (ojo derecho):
16     1261

```

Extracto de código 6.4: Estadísticas voluntaria

```

Columns 166 through 180
    1    1    1    1    1    0    0    1    1    1    0    1    1    1    1

Columns 181 through 195
    0    1    1    1    1    1    0    1    1    1    1    1    1    1    1

Columns 196 through 210
    1    1    1    1    0    1    1    1    1    1    1    0    0    1    1

Columns 211 through 225
    0    1    1    1    1    1    1    1    1    1    1    1    0    1    1

Columns 226 through 240
    1    1    1    1    1    0    1    0    1    1    1    1    1    1    1

Columns 241 through 255
    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1

```

Figura 6.10: Ejemplo de extracto voluntaria.Ojo derecho. Script 3.

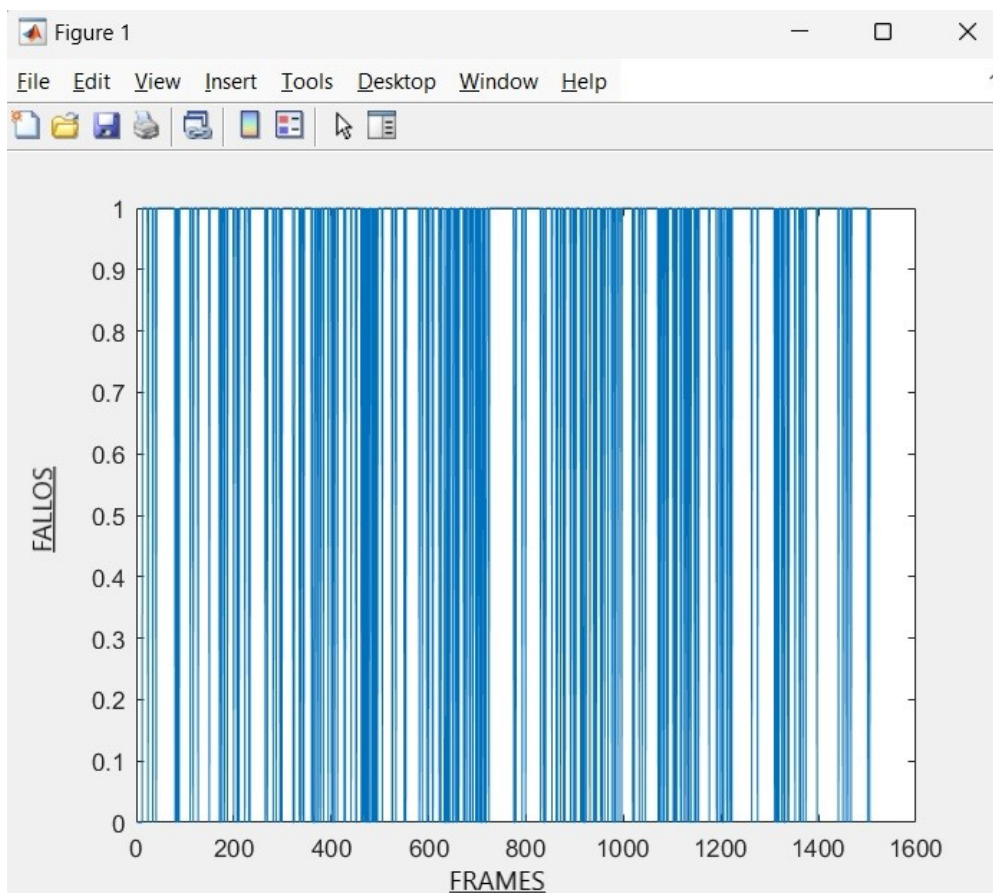


Figura 6.11: Gráfica ojo derecho voluntaria. Script 3

6.2. Evaluación de mejoras y ventajas del nuevo seguidor de mirada

En este proyecto, centrado en el desarrollo de un cuantificador de seguidor de mirada, la investigación ha revelado varios factores que impactan el desempeño del sistema y que, cuando se administran adecuadamente, podrían mejorar en gran medida los resultados logrados.

Uno de los factores más determinantes es la calidad de los recursos utilizados, en particular los videos empleados para las pruebas. Las pupilas del usuario quedan plasmadas de forma clara y adecuada en la imagen, lo que se traduce en una reducción significativa del número de fallos, es decir, una mayor resolución y una buena iluminación permitirían que el algoritmo identifique los contornos y características de la pupila con mayor exactitud.

Además de la calidad del video, la ubicación física del seguidor de mirada es otro factor que afecta los resultados. Si el dispositivo encargado de capturar la mirada del usuario está ubicado en una posición adecuada y estable, la precisión del sistema mejora notablemente. Esto implica que, si el seguidor de mirada se coloca a una distancia y ángulo óptimos respecto al rostro del usuario, se reduce la probabilidad de que ocurran fallos debido a movimientos de la cabeza o ángulos incorrectos de visión. No obstante, a pesar de los avances obtenidos, el proyecto no ha logrado alcanzar el desarrollo de un código completamente óptimo y funcional dentro del plazo establecido. Como resultado, los resultados obtenidos no son los mejores posibles, pero representan un primer paso valioso en el camino hacia un sistema más robusto. Lograr una precisión del 100 % en la detección de la pupila es un reto extremadamente complejo debido a la naturaleza impredecible de los movimientos humanos. El constante movimiento de los ojos y la cabeza, sumado a factores como cambios en la iluminación, representan desafíos significativos que el algoritmo debe afrontar, lo que hace difícil alcanzar un desempeño perfecto en todos los casos.

A pesar de estas limitaciones, el proyecto presenta varias ventajas significativas. El dispositivo de seguimiento ocular es más barato y accesible que otros dispositivos de seguimiento ocular siendo una herramienta versátil que puede ser utilizada por una variedad de usuarios, tanto en la investigación como en la vida cotidiana. Es barato, fácil de conseguir y se puede implementar con facilidad.

Otra gran ventaja es que puedes mejorar este proyecto continuamente, debido a que quedará como código abierto para que sea mejorado por cualquiera que esté interesado en mejorar la vida de las personas que depende de dispositivos como es el seguidor de mirada para su vida diaria. La modularidad del código permite modificarlo y mejorarlo con el tiempo, lo que podría conducir al desarrollo de un sistema avanzado de seguimiento ocular. Utilizando nuevas técnicas de procesamiento de imágenes y aprendizaje automático, el sistema podría superar las limitaciones actuales y lograr una precisión casi total.

7. Conclusiones

7.1. Conclusiones del proyecto

En términos generales, los resultados muestran una notable variabilidad en el rendimiento del algoritmo. En ciertos vídeos, el sistema ha mostrado una alta precisión, con un porcentaje de detección exitosa superior al 98 %, en su simulación más precisas, y con una frecuencia de fallos inferior a 10 por cada 100 fotogramas. Este nivel de precisión sugiere que, en escenarios controlados o con condiciones favorables de iluminación y visibilidad, el algoritmo es capaz de identificar la pupila de forma fiable.

Sin embargo, en otros vídeos, el seguidor de mirada ha experimentado un notable descenso en su efectividad, con una tasa de fallo que supera el 95 %. Estas fallas se han atribuido a factores como cambios en la iluminación, oclusiones parciales del rostro y variaciones en las facciones de los ojos. Este tipo de escenarios ha revelado las limitaciones del sistema, lo que indica la necesidad de optimizar el algoritmo para situaciones más complejas.

En resumen, aunque los resultados iniciales no son perfectos, el proyecto tiene un enorme potencial para seguir mejorando. La mejora en la calidad del vídeo, la optimización de la posición del seguidor de mirada y la evolución del código hacia versiones más robustas ofrecen una perspectiva prometedora para futuras iteraciones. En conjunto, estos factores hacen que este proyecto sea no solo accesible y económico, sino también altamente adaptable, con la capacidad de alcanzar resultados de alta precisión en el seguimiento ocular.

8. Relación e impacto con los ODS

El ratón de mirada o *eyetracker* que refiere al proyecto puede tener un impacto significativo y estar relacionada con varios Objetivos de Desarrollo Sostenible (ODS) de la ONU.



Producido en colaboración con TROLLBÄCK COMPANY | TheGlobalQualityEngineering.com | +1 212 628 1010
Para cualquier duda sobre la utilización, por favor comuníquese con: dpc@compagny.org

Figura 8.1: Objetivos de Desarrollo Sostenible [15]

Se procede a detallar los ODS más significativos del Trabajo Fin de Grado.

3. Salud y Bienestar

En el ámbito de la salud, los seguidores de mirada se utilizan para diagnósticos y tratamientos en diversas áreas, como la oftalmología, la neurología y la psicología. Por ejemplo, pueden ayudar a detectar enfermedades oculares y neurológicas en etapas tempranas, facilitando intervenciones más rápidas y efectivas.

Además, en el contexto de la rehabilitación, los seguidores de mirada son herramientas cruciales para evaluar y mejorar la recuperación de pacientes con lesiones cerebrales o trastornos del espectro autista, permitiendo la personalización de terapias según las necesidades específicas de cada individuo. En la investigación, estos dispositivos contribuyen a un mejor entendimiento de enfermedades neurodegenerativas, como el Alzheimer, al analizar patrones de atención y deterioro cognitivo.

Por otro lado, en el ámbito del bienestar, los seguidores de mirada se utilizan para diseñar entornos más accesibles y seguros, optimizando interfaces y espacios

públicos para personas con discapacidades. Al mejorar la precisión y la eficiencia de los diagnósticos y tratamientos médicos, así como al contribuir a la creación de entornos más inclusivos, los seguidores de mirada promueven directamente el bienestar general y una mejor calidad de vida, apoyando así los objetivos del ODS 3.

9. Industria, Innovación e Infraestructura

Este dispositivo es una herramienta poderosa en la investigación y el desarrollo de tecnologías avanzadas y en la mejora de procesos industriales.

En el ámbito de la industria, los seguidores de mirada se utilizan para optimizar el diseño de productos y procesos, mejorando la ergonomía y la eficiencia. Por ejemplo, en la manufactura, pueden analizar cómo los trabajadores interactúan con máquinas y herramientas, identificando áreas donde se pueden hacer mejoras para aumentar la productividad y reducir el riesgo de lesiones. En la innovación, los eye trackers son esenciales para el desarrollo de interfaces de usuario más intuitivas y efectivas en dispositivos tecnológicos, lo cual es crucial para el avance de la tecnología y la creación de productos innovadores que respondan mejor a las necesidades de los usuarios.

En términos de infraestructura, los seguidores de mirada ayudan en el diseño de espacios más eficientes y accesibles. Por ejemplo, en la planificación urbana y la arquitectura, estos dispositivos pueden evaluar cómo las personas interactúan con diferentes entornos, permitiendo a los diseñadores crear espacios que optimicen el flujo de personas y mejoren la experiencia del usuario en lugares públicos y privados.

10. Reducción de las desigualdades

En el ámbito de la educación, los seguidores de mirada se utilizan para crear materiales didácticos y entornos de aprendizaje que sean inclusivos y accesibles para todos los estudiantes, independientemente de sus habilidades o antecedentes socio económicos. Al analizar cómo los estudiantes interactúan con los contenidos educativos, es posible adaptar estos materiales para satisfacer mejor las necesidades de aquellos con dificultades de aprendizaje o discapacidades, contribuyendo a una educación más equitativa.

En el área de la accesibilidad, los seguidores de mirada son herramientas esenciales para el diseño de tecnologías asistivas y interfaces de usuario que sean más fáciles de usar para personas con discapacidades físicas o cognitivas. Por ejemplo, pueden ayudar a desarrollar sistemas de control ocular para personas con movilidad reducida, permitiéndoles interactuar con computadoras y otros dispositivos de manera independiente, lo que promueve su inclusión social y económica.

Además, en el campo de la salud, los eye trackers pueden mejorar la atención médica para grupos vulnerables. Al proporcionar datos detallados sobre las reacciones y necesidades específicas de estos grupos, los profesionales de la salud pueden ofrecer diagnósticos y tratamientos más personalizados y efectivos, reduciendo las brechas en la calidad de la atención recibida.

También, en la investigación de mercados y el desarrollo de políticas públicas, los seguidores de mirada permiten analizar cómo diferentes grupos perciben y utilizan servicios y productos, lo cual es fundamental para desarrollar estrategias que sean verdaderamente inclusivas y que respondan a las necesidades de las poblaciones marginadas o desfavorecidas.³

En resumen, al promover la accesibilidad y la personalización en la educación, la salud y la tecnología, los seguidores de mirada contribuyen de manera directa a la reducción de las desigualdades, alineándose plenamente con los objetivos del ODS 10 y fomentando una sociedad más justa e inclusiva.

9. Anexos

9.1. Anexo al Trabajo de Fin de Grado: Relación del trabajo con los Objetivos de Desarrollo Sostenible de la agenda 2030

Tabla 9.1: Grado de relación del trabajo con los Objetivos de Desarrollo Sostenible (ODS).

Objetivos de Desarrollo Sostenibles	Alto	Medio	Bajo	No Procede
ODS 1.Fin de la pobreza				X
ODS 2.Hambre cero				X
ODS 3.Salud y bienestar	X			
ODS 4.Educación de calidad			X	
ODS 5.Igualdad de genero				X
ODS 6.Agua limpia y saneamiento				X
ODS 7.Energía asequible y no contaminante				X
ODS 8.Trabajo decente y crecimiento económico				X
ODS 9.Industria, innovación e infraestructuras	X			
ODS 10.Reducción de las desigualdades	X			
ODS 11.Ciudades y comunidades sostenibles				X
ODS 12.Producción y consumo responsables				X
ODS 13.Acción por el clima				X
ODS 14.Vida submarina				X
ODS 15.Vida de ecosistema terrestres				X
ODS 16.Paz, justicia e instituciones sólidas				X
ODS 17.Alianzas para lograr objetivos				X

9.2. Código fuente

```
1
2 nombre = 'Fernando';
3 ensayo = 2;
4
5 cam = webcam(1);
6 Pix_SS = get(0,'screensize');
7 W = Pix_SS(3); H = Pix_SS(4);
8 M = 50;
9 Pix_SS2 = [M M W-M*2 H-M*2];
10 q = linspace(0,2*pi,100);
11 rH = W/8; rV = H/3;
12 x = W/2+rH*cos(q);
```

```

13 y = H/2+rV*sin(q);
14
15 f1 = 1;
16 figure;
17 set(gca,'XTick',[]); set(gca,'YTick',[]);
18 set(gca,'XColor','None'); set(gca,'YColor','None');
19 axis([0 W 0 H]);
20 set(gcf,'Position',Pix_SS); hold on;
21 set(gcf,'MenuBar','none');
22 k = 1;
23 tic
24 t=0;
25 ElapsedTime = 5;
26 while (t < ElapsedTime)
27     cla;
28     snapshot2 = snapshot(cam);
29     snapshot1 = imresize(snapshot2,1.5);
30     snapshot1 = fliplr(snapshot1);
31     imshow(snapshot1); hold on; set(gcf,'Position',Pix_SS2); hold on;
32     patch(x,y,[0.8 0.8 0.2],'Facealpha',0.2,'LineWidth',2,'EdgeColor',[1
33     1 0]);
34     msg1='Place your face in the marked area';
35     msg2 = sprintf('Tiempo restante: %4.2f segundos',ElapsedTime-t);
36     text(W/2.5,H/10,msg1,'FontSize',14,'Color',[1 1 0]);
37     text(W/2.45,H/7,msg2,'FontSize',14,'Color',[1 1 0]);
38     k = k + 1;
39     drawnow;
40     t = toc;
41 end
42 cla;
43 set(gcf,'Position',Pix_SS); hold on;
44 axis([0 W 0 H]);
45
46 FS = 264;
47 for i=1:3
48     cla; axis([0 W 0 H]);
49     text(W/2-FS/2,H/2,num2str(3-i+1),'FontSize',264,'Color',[0 0 0]);
50     msg3 = 'Please, follow the red circle with your eyes';
51     text(W/3.2,3*H/4,msg3,'FontSize',24,'Color',[0 0 0]);
52     pause(1);
53 end
54 Rc = 100; N = 100; velInv = 5;
55 XPoints1=[W/2.0 0+Rc 0+Rc W-Rc W-Rc 0+Rc W/2.0];
56 YPoints1=[H/2.0 H-Rc 0+Rc 0+Rc H-Rc H-Rc H/2.0];
57
58 XPoints2=[W/2.0 W/4+Rc W/4+Rc W-W/4-Rc W-W/4-Rc W/4+Rc W/2.0];

```

```

59 YPoints2=[H/2.0 H-H/4-Rc H/4+Rc H/4+Rc H-H/4-Rc H-H/4-Rc H/2.0];
60
61 XPoints = [XPoints1 XPoints2];
62 YPoints = [YPoints1 YPoints2];
63
64
65 vname = sprintf('%s_%d.mp4', nombre, ensayo);
66
67 v = VideoWriter(vname, 'MPEG-4');
68 open(v);
69 for i=1:length(XPoints)-1
70     dX = abs(XPoints(i+1)-XPoints(i));
71     dY = abs(YPoints(i+1)-YPoints(i));
72     d = ceil(max([dX dY])/velInv);
73     x = linspace(XPoints(i), XPoints(i+1), d);
74     y = linspace(YPoints(i), YPoints(i+1), d);
75     for j=1:d
76         cla; axis([0 W 0 H]);
77         drawcircle('Center', [x(j) y(j)], 'Radius', Rc, 'Color', [1 0 0]);
78         plot(x(j), y(j), 'ro', 'MarkerSize', 12, 'LineWidth', 3);
79         I = snapshot(cam);
80         writeVideo(v, I);
81         drawnow;
82     end
83 end
84 close(v);
85 close all;

```

Extracto de código 9.1: Código Completo Primer Script

```

1
2 v = VideoReader("Sergio_2.mp4");
3 faceDetector = vision.CascadeObjectDetector();
4 detector1 = vision.CascadeObjectDetector('EyePairSmall');
5
6 figure;
7 for i=1:100
8     clf;
9     A = readFrame(v);
10    bboxF = step(faceDetector, A);
11    if size(bboxF, 1) > 1
12        bboxF = bboxF(size(bboxF, 1), :);
13    end
14    face = imcrop(A, bboxF);
15    bboxE = step(detector1, face);
16    if size(bboxE, 1) > 1
17        bboxE = bboxE(size(bboxE, 1), :);
18    end

```

```

19 eye = imcrop(face,bboxE);
20 eyeL = eye(:,1:floor(size(eye,2)/2),:);
21 RL = eyeL(:,:,1); GL = eyeL(:,:,2); BL = eyeL(:,:,3);
22 eyeLg = im2gray(eyeL);
23 eyeLg2 = scale255(eyeLg);
24 [cL, rL] = imfindcircles(eyeLg2,[8 20], 'ObjectPolarity', 'dark', '
Sensitivity', 0.90);
25 if (size(cL,1)>1)
26     meanSi = size(eyeLg2,2)/2;
27     vv = abs(cL(:,1)-meanSi);
28     [Cent,iCent] = min(vv);
29     cL = cL(iCent,:);
30     rL = rL(iCent);
31 end
32 Eo= ceil(size(eye,2)/2);
33 eyeR = eye(:,Eo:end,:);
34 eyeRg = im2gray(eyeR);
35 eyeRg2 = scale255(eyeRg);
36 [cR, rR] = imfindcircles(eyeRg2,[8 20], 'ObjectPolarity', 'dark', '
Sensitivity', 0.90);
37 if (size(cR,1)>1)
38     meanSi = size(eyeRg2,2)/2;
39     vv = abs(cR(:,1)-meanSi)
40     [Cent,iCent] = min(vv);
41     cR = cR(iCent,:);
42     rR = rR(iCent);
43 end
44
45
46 A = insertObjectAnnotation(A,"rectangle",bboxF,'cara');
47 A = insertObjectAnnotation(A,"rectangle",bboxE+[bboxF(1) bboxF(2) 0 0
], 'ojos', 'Color', 'blue');
48 imshow(A); hold on;
49 if (size(cL,1)>0)
50     plot(bboxF(1)+bboxE(1)+cL(1),bboxF(2)+bboxE(2)+cL(2), 'y+');
51
52 end
53 if (size(cR,1)>0)
54     plot(bboxF(1)+bboxE(1)+Eo+cR(1),bboxF(2)+bboxE(2)+cR(2), 'y+');
55 end
56
57 if (size(cL,1)>0)
58     falloL(i)=1;
59 end
60
61 if (size(cR,1)>0)
62     falloR(i)=1;

```



```

63     end
64
65
66     msg = sprintf('frame: %d/%d',i,v.NumFrames); title(msg);
67     drawnow;
68
69
70 end

```

Extracto de código 9.2: Código Completo Segundo Script

```

1
2 v = VideoReader("Sergio_2.mp4");
3 faceDetector = vision.CascadeObjectDetector();
4 detector1 = vision.CascadeObjectDetector('EyePairSmall');
5
6 figure;
7 for i=1:v.NumFrames
8     clf;
9     A = readFrame(v);
10    bboxF = step(faceDetector, A);
11    if size(bboxF,1)>1
12        bboxF = bboxF(size(bboxF,1),:);
13    end
14    face = imcrop(A,bboxF);
15    bboxE = step(detector1, face);
16    if size(bboxE,1)>1
17        bboxE = bboxE(size(bboxE,1),:);
18    end
19    eye = imcrop(face,bboxE);
20    eyeL = eye(:,1:floor(size(eye,2)/2),:);
21    RL = eyeL(:,:,1); GL = eyeL(:,:,2); BL = eyeL(:,:,3);
22    eyeLg = im2gray(eyeL);
23    eyeLg2 = scale255(eyeLg);
24    [cL, rL] = imfindcircles(eyeLg2,[8 20],'ObjectPolarity','dark','
Sensitivity',0.90);
25    if (size(cL,1)>1)
26        meanSi = size(eyeLg2,2)/2;
27        vv = abs(cL(:,1)-meanSi);
28        [Cent,iCent] = min(vv);
29        cL = cL(iCent,:);
30        rL = rL(iCent);
31    end
32    Eo= ceil(size(eye,2)/2);
33    eyeR = eye(:,Eo:end,:);
34    eyeRg = im2gray(eyeR);
35    eyeRg2 = scale255(eyeRg);
36    [cR, rR] = imfindcircles(eyeRg2,[8 20],'ObjectPolarity','dark','

```

```

Sensitivity',0.90);
37 if (size(cR,1)>1)
38     meanSi = size(eyeRg2,2)/2;
39     vv = abs(cR(:,1)-meanSi)
40     [Cent,iCent] = min(vv);
41     cR = cR(iCent,:);
42     rR = rR(iCent);
43 end
44
45 subplot(3,2,1);imshow(eyeL); hold on;
46 subplot(3,2,2);imshow(eyeR); hold on;
47 subplot(3,2,3);imshow(eyeLg); hold on;
48 subplot(3,2,4);imshow(eyeRg); hold on;
49 subplot(3,2,5);imshow(eyeLg2); hold on;
50
51 if (size(cL,1)>0)
52     plot(bboxF(1)+bboxE(1)+cL(1),bboxF(2)+bboxE(2)+cL(2),'y+');
53
54 end
55 if (size(cR,1)>0)
56     plot(bboxF(1)+bboxE(1)+Eo+cR(1),bboxF(2)+bboxE(2)+cR(2),'y+');
57 end
58
59 if (size(cL,1)>0)
60     h = viscircles(cL,rL);
61     plot(cL(1),cL(2),'xy');
62     falloL(i)=1;
63 end
64
65 subplot(3,2,6);imshow(eyeRg2); hold on;
66 if (size(cR,1)>0)
67     h = viscircles(cR,rR);
68     plot(cR(1),cR(2),'xy');
69     falloR(i)=1;
70 end
71
72
73 msg = sprintf('frame: %d/%d',i,v.NumFrames); title(msg);
74 drawnow;
75
76 pause
77 end

```

Extracto de código 9.3: Código Completo Tercer Script

Bibliografía

- [1] Monika Płużyczka. The first hundred years: a history of eye tracking as a research method. https://www.academia.edu/10634700/The_history_of_Eye_tracking, 2018.
- [2] Historiadelamedicina. Estereómetro, 2016. URL <https://historiadelamedicina.wordpress.com/wp-content/uploads/2016/12/estereometro.png>.
- [3] Image titled "history of eye tracking". <https://images.app.goo.gl/b74t6srd3yYJ5PpL8>.
- [4] Monika Płużyczka. The first hundred years: A history of eye tracking as a research method. *Applied Linguistics Papers*, (25/4):101–116, 2018.
- [5] Pramodini A Punde, Mukti E Jadhav, and Ramesh R Manza. A study of eye tracking technology and its applications. In *2017 1st International Conference on Intelligent Systems and Information Management (ICISIM)*, pages 86–90. IEEE, 2017.
- [6] BrainLatam. Como funciona e quais são os tipos de eye tracking, 2023. URL <https://brainlatam.com/blog/como-funciona-e-quais-sao-os-tipos-de-eye-tracking-485>. Accedido: 2024-09-11.
- [7] E.D. Guestrin and M. Eizenman. General theory of remote gaze estimation using the pupil center and corneal reflections. *IEEE Transactions on Biomedical Engineering*, 53(6):1124–1133, 2006. doi: 10.1109/TBME.2005.863952.
- [8] Andrew H Clarke. Current trends in eye movement measurement techniques. In *Advances in Psychology*, volume 116, pages 347–364. Elsevier, 1996.
- [9] ResearchGate. Posición de los electrodos para el análisis de los electrooculogramas. https://www.researchgate.net/figure/Figura-1-Posicion-de-los-electrodos-para-el-analisis-de-los-electrooculogramas_fig3_27557516.
- [10] Thomas Haslwanter and Andrew H Clarke. Eye movement measurement: electro-oculography and video-oculography. *Handbook of Clinical Neurophysiology*, 9:61–79, 2010.
- [11] Author's Name. Title of the document. <http://hdl.handle.net/10251/27241>, Year of Publication.
- [12] T. Ohno, N. Mukawa, and A. Yoshikawa. Freegaze: A gaze tracking system for everyday gaze interaction. In *Proceedings of the Symposium on ETRA 2002: Eye Tracking Research & Applications Symposium*, pages 125–132, New Orleans, USA, 2002.

- [13] Takashi Nagamatsu, Junzo Kamahara, and Naoki Tanaka. 3d gaze tracking with easy calibration using stereo cameras for robot and human communication. In *RO-MAN 2008-The 17th IEEE International Symposium on Robot and Human Interactive Communication*, pages 59–64. IEEE, 2008.
- [14] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.
- [15] EcoTerra. Objetivos de desarrollo sostenible, 2015. URL <https://ecoterra.com/objetivos-de-desarrollo-sostenible-ods-y-agenda-2030/>.
- [16] MathWorks. Matlab toolbox access. <https://es.mathworks.com/help/thingspeak/matlab-toolbox-access.html>.
- [17] H. S. Sawhney, J. Oliensis, and A. R. Hanson. Description and reconstruction from image trajectories of rotational motion. In *Proceedings of the Third International Conference on Computer Vision (ICCV)*, pages 494–498, Osaka, Japan, 1990.
- [18] S. Shih and J. Liu. A novel approach to 3-d gaze tracking using stereo cameras. *IEEE Transactions on Systems, Man, and Cybernetics*, 34(1):234–245, February 2004.
- [19] Y. Le Grand. *Light, Color and Vision*. Chapman and Hall, London, England, 1957.
- [20] Kang Ryoung Park. Robust gaze estimation for human computer interaction. In *PRICAI 2006: Trends in Artificial Intelligence: 9th Pacific Rim International Conference on Artificial Intelligence Guilin, China, August 7-11, 2006 Proceedings 9*, pages 1222–1226. Springer, 2006.

Hacia un nuevo ratón de mirada de bajo coste para comunicación aumentativa y alternativa

TRABAJO FIN DE GRADO

DOCUMENTO SEGUNDO

PLIEGO DE CONDICIONES

Para la obtención del título de
Grado en Ingeniería Electrónica Industrial y Automática

Índice general

1	Objeto del pliego	1
2	Condiciones de los materiales	2
2.1.	Normativa que avala los elementos	2
2.2.	Especificaciones técnicas	2
2.2.1.	Descripción general	2
2.2.2.	Características de los componentes	2
2.2.3.	Especificaciones de seguridad del <i>eyetracker</i>	3
2.2.4.	Soluciones aportadas	8
2.2.5.	Resultados finales	10

1. Objeto del pliego

El presente documento ha sido redactado con el propósito principal de definir de forma clara y detallada las especificaciones que orientarán la ejecución del proyecto referido. En él se describen los estándares técnicos necesarios para los materiales que se utilizarán durante su desarrollo.

El proyecto se enfoca en el desarrollo de un programa utilizando el lenguaje de programación MATLAB. Los distintos scripts de MATLAB tienen como objetivo principal de evaluar la efectividad y precisión de los ratones de seguimiento ocular disponibles en el mercado actual, concretamente en velocidad, posición y trayectoria.

2. Condiciones de los materiales

2.1. Normativa que avala los elementos

El dispositivo *eye tracker* está alineado con las regulaciones establecidas por la siguiente norma:

- Norma ISO 14490 Óptica y fotónica. Métodos de prueba para sistemas telescópicos. Esta norma detalla métodos de prueba específicos para sistemas telescópicos en el ámbito de la óptica y la fotónica.

2.2. Especificaciones técnicas

2.2.1. Descripción general

El *eye tracker*, conocido también como dispositivo de seguimiento ocular, es una herramienta tecnológica avanzada diseñada para capturar y registrar con precisión los movimientos oculares de un usuario. Utilizando sensores ópticos o de infrarrojos, este dispositivo detecta y analiza la dirección y posición de la mirada sobre una superficie, como una pantalla de computadora o dispositivo móvil.

Se trata de un producto de comunicación aumentativa alternativa, apropiado para pacientes de todas las edades, con la capacidad de generar voz integrada.

El código desarrollado se diferencia en 3 scripts para cuantificar las habilidades del *eye tracker* ha sido evaluado en la Fundación Gema Canales, una organización dedicada a mejorar la calidad de vida de personas con parálisis cerebral y sus familias, quienes trabajan diariamente con este tipo de enfermedades y dispositivos.

2.2.2. Características de los componentes

En este Trabajo Final de Grado se utilizó MATLAB para la programación, con licencia proporcionada por la Universidad Politécnica de Valencia. Se necesitaron varios Toolbox específicos para cumplir con los objetivos del proyecto:

1. MATLAB Support Package for USB Webcams

Permite la captura de imágenes en tiempo real desde Webcams compatibles con UVC, accesibles desde MATLAB y MATLAB Online™.

2. Image Processing Toolbox

Proporciona herramientas para el procesamiento y análisis de imágenes, incluyendo segmentación, mejora, reducción de ruido y registro de imágenes, con soporte para procesamiento en 2D, 3D y tamaños arbitrarios. Facilita la automatización de flujos de trabajo y permite la aceleración con procesadores multinúcleo y GPU.

3. Image Acquisition Toolbox

Ofrece funciones para conectar cámaras a MATLAB y Simulink, admitiendo diversos modos de obtención y compatibilidad con estándares y hardware de visión artificial y dispositivos industriales de alta gama.

Adicionalmente, se emplearon otros ToolBox, algunos diseñados por usuarios, que facilitaron la programación del eyetracker y optimizaron la estabilidad de las simulaciones.

El dispositivo de trabajo o ordenador debe tener unas características mínimas para el correcto funcionamiento de Mathworks, para la versión R2024 es necesario un sistema operativo Windows 10 o 11, un procesador Intel o ADM de x86 o x64 con 2 o mas cores, normalmente son intel 5, el cual cumple sin problema, por lo tanto no es una restricción. Una RAM de 8 GB aunque se recomienda de 16 GB y un espacio de ROM suficiente para la instalación del programa MATLAB. Pero eso no es lo mas importante ya que las verdaderas limitaciones no están en el software.

Las características a tener en cuenta son las del propio *eye tracker* o ratón de mirada, por ello se ha código un dispositivo del mercado para poder estimar los componentes necesarios.

Funciones estandar: 2 cámaras, 1 micrófono, 2 altavoces, 1 ECU (unidad de control ambiental) con IR, 1 Bluetooth® y 1 WLAN (red de área local inalámbrica)

La media de una pantalla es de 13.3" con una resolución de 1920 × 1080 píxeles, es decir, aproximadamente el tamaño de una tablet cotidiana.

Se expresan en detalle la características de los componentes del dispositivo en la tabla [2.1](#)

Seguidor Ocular

2.2.3. Especificaciones de seguridad del *eyetracker*

Para este tipo de productos, hay algunas advertencias que se deben considerar:

- No se permite realizar ninguna modificación en este equipo.
- Las reparaciones de un dispositivo solo deben ser realizadas por el fabricante o un centro de reparación autorizado y aprobado.

Contraindicaciones: El dispositivo nunca debe ser el único medio de comunicar información importante para el usuario. En caso de avería del dispositivo, el usuario no podrá comunicarse utilizándolo.

Tabla 2.1: Características de los componentes

Componentes del Seguidor Ocular	
Pantalla	13.3"
Resolución de pantalla	1920 × 1080 píxeles (FHD)
Panel táctil	Multitáctil 10 puntos
Pantalla trasera	480 × 128 píxeles
Dimensiones (an. x al. x pr.)	34,3 × 24,1 × 8,1 cm
Peso	2,4 kg
Altavoces	2 altavoces cerrados de 10 W
Micrófono	1 micrófono analógico
Procesador	Intel Core i5-7300U
RAM	16 GB LPDDR4
Sistema Operativo	Windows 10
Unidad de disco duro	256 GB, SSD basado en SATA
Conectores	
Botones	1 de encendido 1 para subir volumen 1 para bajar volumen 2 programables
Cámara	Cámara de usuario (trasera) 8 MP AF (Autoenfoco), con flash Cámara web (delantera) 2 MP FF (Foco fijo)
Vida útil prevista	5 años
Duración de la batería	Hasta 8 h
Tiempo de carga de la batería	Máximo 5.4 h (0-100 %)
Fuente de energía	Adaptador de CA de 65 W

- En caso de ocurrir un incidente grave, comuníquese con el fabricante y las autoridades locales de inmediato. Aviso: Se recomienda instalar el software apropiado. Si tiene alguna pregunta, comuníquese con el fabricante para recibir ayuda. Aviso: Para evitar un acceso no autorizado, se recomienda instalar un software antivirus apropiado o no conectarse a redes externas poco seguras
- El seguidor ocular no debe utilizarse como dispositivo de soporte vital ni tampoco deberá utilizarse en caso de fallo debido a un corte de energía u otro motivo.
- Existe riesgo de atragantamiento si se desmontan las piezas pequeñas del dispositivo.
- El dispositivo no deberá exponerse ni utilizarse en condiciones meteorológicas diferentes a las indicadas en las Especificaciones técnicas.
- El dispositivo solo deberá utilizarse con los accesorios específicos del mismo, que vienen acompañados de instrucciones de montaje.
- El cable de carga podría representar riesgos de estrangulación para niños pequeños. Nunca deje a niños pequeños desatendidos con el cable de carga.

Tabla 2.2: Adaptador de corriente del dispositivo

Elemento	Especificación
Fabricante	FSP Group Inc.
Tipo/Escribir	Adaptador de corriente de conmutación
Modelo	FSP065-DBCM1
Voltaje de entrada	100 a 240 VCA
Corriente de entrada (máx.)	2.0-1.0 A
Frecuencia de entrada	50 a 60 Hz
Corriente de salida	3.43 A
Voltaje de salida nominal	19.0 VCC

- Los niños y aquellas personas que presenten discapacidades cognitivas no deberán tener acceso al dispositivo ni usarlo sin la supervisión de padres o tutores.
- No coloque ningún elemento decorativo, notas adhesivas ni otros objetos similares en el lateral de la pantalla del dispositivo. Esto podría interferir con el Eye Gaze o afectar el rendimiento de la pantalla táctil.
- Nunca fuerce un conector en un puerto. Si el conector y el puerto no se unen con facilidad, es probable que no coincidan. Asegúrese de que el conector coincida con el puerto y de que lo haya colocado correctamente en relación con el puerto. Extreme las precauciones con los conectores USB.

Prevención de daño auditivo

Se puede producir una pérdida permanente de la audición si los audífonos, los auriculares o los altavoces se utilizan a un volumen alto. Para evitar esto, el volumen se debe fijar en un nivel seguro. Con el tiempo, se puede perder la sensibilidad a los niveles altos de sonido, que luego pueden sonar aceptables, pero aun así podrían estar dañando su audición. Si experimenta síntomas, como un zumbido en los oídos, baje el volumen o deje de usar los audífonos/auriculares. Cuanto más alto es el volumen, menos tiempo se requiere para que la audición resulte afectada. Los expertos en audición sugieren las siguientes medidas para proteger la audición:

- Limitar la cantidad de tiempo que usa audífonos o auriculares con el volumen alto.
- Evitar subir el volumen para bloquear los entornos ruidosos.
- Bajar el volumen si no puede oír a las personas que hablan cerca de usted.

Para establecer un nivel de volumen seguro:

- Fije el control de volumen en una configuración baja.
- Aumente lentamente el sonido hasta que pueda escuchar de forma cómoda y clara, sin distorsión.

El dispositivo puede producir sonidos en rangos de decibelios que pueden provocar pérdida de la audición en personas con audición normal, incluso si se

Tabla 2.3: Adaptador de corriente del dispositivo

Especificaciones técnicas	Modelo X
Distancia de funcionamiento	45 — 95 cm
Libertad de movimiento de la cabeza (ancho x altura)	~20 × 20 cm (7.9 × 7.9 in) a 50 cm de la pantalla ~35 × 35 cm (13.8 × 13.8 in) entre 65 y 80 cm de la pantalla
Posicionamiento:	
Distancia (de la pantalla)	45 — 95 cm (20 — 37 in)
Tamaño del cuadro de seguimiento (ancho × alto)	20 × 20 — 35 × 35 cm (7.9 × 7.9 — 13.8 × 13.8 in)
Fondo del cuadro de seguimiento	50 cm (19.7 in)
Velocidad de transferencia de datos de la mirada	33 Hz
Velocidad de muestra de la mirada	133 Hz
Técnica de seguimiento ocular	Seguimiento ocular basado en vídeo y reflejo corneal con modos de iluminación de pupila oscuro y claro.
Uso en exteriores posible	Sí
Calibración de usuario (solidez del seguimiento)	¿98 %
Detectar mirada	
Interacción ¿30Hz	98 % para 95 % de la población
Precisión de la mirada En el 95 % de la población	1.58°
Máx. velocidad de movimiento de la cabeza	40 cm/s (15.7 in/s)
Posición del ojo	10 cm/s (3.9 in/s)
Datos de la mirada	
Flujo de datos y Velocidad de transferencia de datos	17 ms
Latencia de mirada	0 ms
Recuperación de mirada	
Montaje	Integrado
Fuente de energía	Integrado

exponen durante menos de un minuto. El nivel de sonido máximo de la unidad equivale a los niveles de sonido que una persona joven y sana puede producir mientras grita. Debido a que el aparato está diseñado para utilizarse como una prótesis de voz, posee las mismas posibilidades y los mismos riesgos potenciales de afectar la audición. Los intervalos de decibelios más altos se ofrecen para permitir la comunicación en entornos ruidosos y deben usarse con cuidado y solo en dichos entornos ruidosos.

Temperatura

Si se utiliza directamente bajo la luz solar o en otro entorno caluroso, es posible que el ratón de mirada presente superficies calientes. Hay sensores de temperatura

integrados que controlan la temperatura. Si los sensores detectan temperaturas superiores a la temperatura especificada en tabla ??, el dispositivo generará automáticamente un comando de apagado o de hibernación de Windows (según la configuración del botón de inicio de Windows) para prevenir un riesgo de lesión para el usuario. Es posible que tarde un momento antes de que el sistema se pueda reiniciar, ya que la unidad necesita enfriarse. Exciten áreas que puede alcanzar temperaturas que podrían causar lesiones corporales permanentes en caso de contacto prolongado con la piel. Tomar las precauciones necesarias para evitar lesiones.

Tabla 2.4: Adaptador de corriente del dispositivo

Temperatura máxima permitida para la superficies externas metálicas del aparato con las que es probable entrar en contacto	Temperatura máxima °C/°F
Umbral de temperatura:	
Temperatura máxima de las superficies permitida	55/131
Temperatura máxima externa permitida en un area con la que es improbable que entremos en contacto	60/140

Fuente de alimentación y baterías

- El dispositivo utiliza baterías de iones de litio. El rango de temperatura de almacenamiento de estas baterías oscila entre -20 °C/-4 °F y 40 °C/104 °F en un período de 3 meses. Si se encuentra en un entorno caluroso, tenga en cuenta que esto puede afectar el estado de carga de las baterías. La temperatura interna debe situarse entre 0 °C/32 °F y 45 °C/113 °F para la carga de la batería. Si la temperatura interna de las baterías asciende por encima de 45 °C/113 °F, estas no se cargarán.
- Traslade la unidad y las baterías a un lugar más frío para que estas puedan cargarse de manera apropiada.
- Evite exponer las baterías al fuego o a temperaturas superiores a 50 °C/122 °F. Estas condiciones pueden provocar que la batería funcione mal, genere calor, prenda fuego o estalle. Tenga en cuenta que es posible, en el peor de los casos, que las temperaturas sean superiores a las mencionadas anteriormente, por ejemplo, en el maletero de un automóvil en un día caluroso. Por lo tanto, el almacenamiento del dispositivo con las baterías instaladas en el maletero de un automóvil expuesto al calor puede causar un mal funcionamiento.
- No desarme ni dañe la batería. Siga las leyes y regulaciones ambientales que se aplican en su zona al momento de desechar las baterías.

Emergencia

No utilice el dispositivo para realizar llamadas de emergencia ni transacciones bancarias. Recomendamos tener varias maneras de comunicarse en caso de

situaciones de emergencia. Las transacciones bancarias solo se deben realizar con un sistema recomendado por el banco y aprobado según sus normas.

Epilepsia

Algunas personas con epilepsia fotosensible son propensas a sufrir ataques epilépticos o pérdida de conciencia cuando se exponen a ciertas luces parpadeantes o a patrones lumínicos de la vida diaria. Esto puede ocurrir incluso si la persona no tiene un historial médico de epilepsia o si nunca tuvo ataques de epilepsia.

Una persona con epilepsia fotosensible posiblemente también tenga problemas con las pantallas de TV, algunos videojuegos y las bombillas fluorescentes que parpadean. Estas personas pueden tener un ataque epiléptico al ver ciertas imágenes o patrones en un monitor, o inclusive cuando quedan expuestos a las fuentes de luz de un seguidor ocular. Se calcula que aproximadamente entre el 3% y el 5% de las personas con epilepsia tienen este tipo de epilepsia fotosensible. Muchas personas con epilepsia fotosensible ven un “aura” o experimentan sensaciones extrañas antes de sufrir los ataques epilépticos. Si se siente extraño durante el uso, aleje sus ojos del seguidor ocular.

Software

Windows Defender, que viene instalado en Windows, brinda protección contra virus, malware y otras amenazas potenciales. Independientemente de si utiliza Windows Defender o un software antivirus de terceros, le recomendamos que mantenga el software del antivirus actualizado.

Seguridad para los niños

Los *eyetracker* son sistemas de computación avanzados y dispositivos electrónicos. Como tales, se componen de numerosas piezas ensambladas independientes. Es posible que estas piezas se separen del dispositivo en manos de un niño, lo que crea un posible riesgo de ahogo u otro peligro para el niño. Los niños no deben tener acceso al dispositivo ni deben usarlo sin la supervisión de los padres o tutores.

Eye Gaze

Es posible que algunas personas experimenten cierto grado de fatiga (debido a la focalización ocular y el alto nivel de concentración necesarios) o incluso sequedad ocular (debido a una menor frecuencia de parpadeo) mientras se familiarizan con el Eye Gaze. Si se siente fatiga o sequedad en los ojos, empiece poco a poco y limite la duración de sus sesiones con Eye Gaze hasta tener un nivel de comodidad adecuado. Remojarse los ojos con gotas oftalmológicas puede ayudarle a combatir esta sequedad.

2.2.4. Soluciones aportadas

En este proyecto, se ha desarrollado un experimento con el objetivo de evaluar y cuantificar las características técnicas de un dispositivo de seguimiento ocular *eyetracker*, tales como precisión, velocidad de respuesta, y precisión en la detección

de la posición ocular. El experimento se fundamenta en la implementación de un código en MATLAB, que genera una trayectoria predefinida que sigue un círculo rojo en movimiento, el cual debe ser rastreado visualmente por los participantes.

Para asegurar la calidad y precisión de las mediciones, se realizaron ajustes preliminares en la cámara del portátil utilizado, enfocándose en la resolución y la exposición. Estos parámetros fueron optimizados para capturar una imagen clara y nítida de los ojos de los usuarios, asegurando que los datos recogidos sean representativos y minimizando la interferencia de factores externos.

El estudio involucró a cinco voluntarios, quienes participaron en la prueba siguiendo la trayectoria del círculo rojo con su mirada. Con el fin de obtener un conjunto de datos robusto y reducir el impacto de posibles errores humanos, cada participante realizó tres grabaciones independientes. Este enfoque permite analizar la consistencia y reproducibilidad de las mediciones del *eyetracker* y proporciona una base estadística sólida para evaluar sus capacidades.

Las grabaciones obtenidas se utilizarán para calcular métricas clave del *eyetracker*, tales como la precisión en la estimación de la posición ocular, la velocidad del seguimiento ocular, y la fiabilidad general del dispositivo. Al analizar estos datos, se podrá determinar el rendimiento del *eyetracker* bajo condiciones controladas y establecer un perfil detallado de sus capacidades técnicas.

Una vez realizados todos los vídeos con los distintos usuarios, se procede a simular los 2 scripts restantes para estimar los errores de detección de pupila en los vídeos que se han generado. Por un lado, el primer script y estudio se trata de analizar exhaustivamente todos y cada uno de los fotogramas de los vídeos proporcionados por los usuarios. Si se identifica un rostro en algún fotograma, se generará un rectángulo amarillo con la etiqueta "cara". De igual manera, si se detectan los ojos, se visualizará un rectángulo azul con la etiqueta "ojos". En el caso de que no se identifiquen ni el rostro ni los ojos, no se mostrará el rectángulo correspondiente. A partir de la frecuencia con la que se detecten tanto el rostro como los ojos a lo largo de todos los fotogramas de los vídeos, se podrá derivar una estadística que refleje la precisión y la calidad del sistema de seguimiento ocular.

Por otro lado, el tercer y última parte del proceso desarrolla una función similar pero en este caso se busca identificar la pupila en los vídeos generados por el primer script a lo largo de todos los fotogramas, con el fin de obtener una estadística sobre las condiciones de detección, considerando factores como la iluminación y la posición del usuario. Si la pupila es detectada correctamente, el resultado será un valor de 1, además de mostrarse un círculo rojo sobre la pupila en la ventana de simulación. Por el contrario, si no se logra detectar la pupila, el resultado será un valor de 0 y no se visualizará ningún círculo. La imagen resultante, tras la simulación de los fotogramas, mostrará ambos ojos por separado en una escala de grises compuesta por tres tonos distintos, ajustada para optimizar la detección de la pupila en el vídeo.

2.2.5. Resultados finales

Tras el periodo de prueba y ajuste final del código hacia el *eyetracker* o seguidor de mirada, se ha podido emplear en pacientes reales que van a utilizar este dispositivo y proyecto en su día. Todo ello, gracias a la Fundación Gema Canales que ha regalado la oportunidad y ha hecho posible que se ponga en uso el proyecto.

Como se puede apreciar en los vídeos adjuntos el resultado no es otro que optimizar y ayudar en el día a día de los portadores de este tipo de enfermedad tan dura. Destacar de nuevo que el código quedará en abierto para que cualquier persona pueda seguir desarrollando y mejorando el mundo del *eyetracking*.

Hacia un nuevo ratón de mirada de bajo coste para comunicación aumentativa y alternativa

TRABAJO FIN DE GRADO

DOCUMENTO TERCERO

PRESUPUESTO

Para la obtención del título de
Grado en Ingeniería Electrónica Industrial y Automática

Índice general

1	Introducción	1
2	Costes materiales	2
2.1.	Presupuesto hardware	2
2.2.	Presupuesto software	2
2.3.	Presupuesto total de materiales	2
3	Costes mano de obra	3
4	Amortizaciones de equipos y licencias	4
5	Costes totales	6

1. Introducción

El presente documento tiene como objetivo detallar el presupuesto estimado para el diseño de un seguidor de mirada destinada a facilitar la comunicación y acciones cotidianas de personas que sufren enfermedades que afectan a la movilidad del cuerpo humano.

Este dispositivo ha sido diseñado concretamente para habilitar un nuevo ratón de mirada para comunicación aumentativa y alternativa pero de un coste más reducido en comparación con los existentes.

Se tendrán en cuenta los costes del propio *eyetracker* así como los materiales necesarios para la programación del producto además, de la mano de obra, las amortizaciones y licencias necesarias para realizar el proyecto.

2. Costes materiales

A continuación, se detallan los gastos de los materiales adquiridos para el diseño del ratón de mirada.

2.1. Presupuesto hardware

Tabla 2.1: Materiales Hardware

DENOMINACIÓN	PRECIO(€/UD)	CANTIDAD	TOTAL(€)
Ordenador personal	750	1	750
Eye tracker Fundación Gema Canales	12000	1	12000
TOTAL			12750

2.2. Presupuesto software

Tabla 2.2: Materiales Software

DENOMINACIÓN	PRECIO(€/UD)	CANTIDAD	TOTAL(€)
Licencia MATLAB	262	1	262
TOTAL			262

2.3. Presupuesto total de materiales

Tabla 2.3: Presupuesto total de materiales

DENOMINACIÓN	TOTAL(€)
Materiales Hardware	12750
Materiales Software	262
TOTAL	13012

En el caso de este trabajo fin de grado, la licencia de MATLAB ha sido financiada por la UPV, ya que dicha universidad dispone de licencia gratuitas para sus usuarios, además, el ratón de mirada también ha sido financiado por la Fundación Gema Canales, que ha trabajado con pacientes que se han ofrecido como voluntarios para probar ambos códigos desarrollados en el proyecto.

3. Costes mano de obra

La sección de los costes de mano de obra se tratan los costes asociados al trabajo del estudiante encargado de desarrollar el proyecto.

Se tienen en cuenta, las horas de dedicación del alumno a la programación del dispositivo, las horas necesarias en la puesta a prueba, y finalmente, el tiempo invertido en la redacción de los documentos necesarios para completar el proyecto.

Tabla 3.1: Gastos Mano de obra

DENOMINACIÓN	Realizador	PRECIO(€/h)	HORAS	TOTAL(€)
Programación	Alumno	5,1	135	688,5
Puesta en marcha	Alumno	5,1	20	102
Redacción Documentos	Alumno	5,1	190	969
		TOTAL	345	1759,5

4. Amortizaciones de equipos y licencias

En el presente apartado, se detallan los gastos asociados con los equipos y licencias de los lenguajes empleados para la programación del seguidor de mirada.

Para ello, se ha estimado un calculo del precio por hora de cada uno de los componentes.

A continuación, se muestran tablas y periodos para amortizar el desembolso invertido en el equipo de trabajo para la programación del proyecto.

Tabla 4.1: Amortizaciones

	COSTE (€)	AMORTIZACIÓN (AÑOS)
Ordenador Personal	750	5
Licencia Matlab	262	1
Eye Tracker Fundación Gema Canales	12000	10

Una vez conocido el coste del equipo y licencia de matlab así como su periodo de vida útil se pasa al cálculo del precio por hora, tal y como se muestra a continuación.

Ordenador Personal

$$\frac{750\text{€}}{5\text{años}} = \frac{150\text{€}}{1\text{año}} \quad (4.1)$$

$$\frac{150\text{€}}{1\text{año}} \times \frac{1\text{año}}{12\text{meses}} \times \frac{1\text{mes}}{22\text{días laborables}} \times \frac{1\text{ día laborable}}{8\text{ horas}} = 0,0710 \frac{\text{€}}{\text{h}} \quad (4.2)$$

Licencia MATLAB

$$\frac{262\text{€}}{1\text{año}} = \frac{262\text{€}}{1\text{año}} \quad (4.3)$$

$$\frac{262\text{€}}{1\text{año}} \times \frac{1\text{año}}{12\text{meses}} \times \frac{1\text{mes}}{22\text{días laborables}} \times \frac{1\text{ día laborable}}{8\text{ horas}} = 0,124 \frac{\text{€}}{\text{h}} \quad (4.4)$$

Eye Tracker Fundación Gema Canales

$$\frac{12000\text{€}}{10\text{años}} = \frac{1200\text{€}}{1\text{año}} \quad (4.5)$$

$$\frac{1200 \text{ €}}{1 \text{ año}} \times \frac{1 \text{ año}}{12 \text{ meses}} \times \frac{1 \text{ mes}}{22 \text{ días laborables}} \times \frac{1 \text{ día laborable}}{8 \text{ horas}} = 0,568 \frac{\text{€}}{\text{h}} \quad (4.6)$$

Por tanto, ya conocidos el precio por hora que supondrán los equipos y licencias necesarios, se pasa al cálculo del coste total realizando una estimación de las horas consumidas por cada uno de ellos.

Tabla 4.2: Amortizaciones Finales

DENOMINACIÓN	PRECIO(€/h)	CANTIDAD (h)	TOTAL(€)
Ordenador Personal	0.0710	345	24.495
Licencia Matlab	0.124	155	19.22
Eye Tracker Fundación Gema Canales	0.568	20	11.36
TOTAL			55.075

5. Costes totales

Para el cálculo total del presupuesto del proyecto será necesario una suma de todos los gastos que se han ido comentando a los largo de los anteriores apartados, además de costes indirectos lo cuales están relacionados con el uso de los recursos disponibles, gasto de mantenimiento y otros imprevistos posibles, se estima alrededor de un 13 %.

Por último, al presupuesto total habrá que sumarle los impuestos del país, en concreto 21 %, lo cual se trata de impuestos relacionados con el IVA.

Tabla 5.1: Presupuesto Total

PRESUPUESTO TOTAL	%	€
Total Mano de Obra		1759.5
Total Materiales		13012
Total Amortizaciones		55.075
PRESUPUESTO TOTAL NETO		14826.58
Costes Indirectos	13	1927.46
PRESUPUESTO ANTES DE IMPUESTOS		16754.04
IVA	21	3518.35
TOTAL		20272.39