

UNIVERSIDAD AUTÓNOMA DE MADRID

ESCUELA POLITÉCNICA SUPERIOR



**Grado en Ingeniería de Tecnologías y Servicios de
Telecomunicación**

TRABAJO FIN DE GRADO

**Detección del punto de observación del usuario en la pantalla
mediante una webcam**

**Paula Delgado de Santos
Tutor: Jesús Bescós Cano**

Julio 2018

Detección del punto de observación del usuario en la pantalla mediante una webcam

AUTOR: Paula Delgado de Santos
TUTOR: Jesús Bescós Cano



Video Processing and Understanding Lab
Departamento de Tecnología Electrónica y de las Comunicaciones
Escuela Politécnica Superior
Universidad Autónoma de Madrid
Junio 2017

Resumen

En este Trabajo Fin de Grado se estudian diferentes rastreadores oculares que están en el mercado, puntualizando cuáles de ellos pueden ser de utilidad. De igual manera, este trabajo trata de conseguir implementar un rastreador ocular que identifique el punto de observación de un usuario, mediante una webcam, en la pantalla del ordenador.

Para ello, se comienza investigando los diferentes tipos de rastreadores oculares, estudiando cuales son de mayor utilidad para este trabajo. Una vez descubiertos aquellos que pueden ser de ayuda, se contemplan distintos software con licencia libre de uso y que utilicen una cámara web para captar las imágenes. Tras este estudio se ha centrado la atención en tres software distintos, realizando un análisis a fondo de su funcionamiento.

Tras no conseguir resultados suficientes, se ha implementado un nuevo programa que, a partir de un conjunto de puntos característicos y asumiendo una serie de premisas, detecta con cierta fiabilidad el punto de observación del usuario en la pantalla del ordenador.

Para conseguir el punto de observación, es necesaria una extracción de puntos característicos y una calibración previa donde se detecta la distancia que se encuentra entre el usuario y la pantalla. Una vez realizado esto, se estudia el problema separando la posición horizontal y vertical de la pupila del usuario. Para estudiar la posición horizontal se calculan las proyecciones de las pupilas, mientras que la posición vertical se realiza por medio de un serie de cálculos trigonométricos. El programa trata cada uno de los dos ojos por separado.

Tras este estudio se realiza una serie de pruebas para comprobar la fiabilidad del programa, dejando como trabajo futuro aquellos puntos que se han encontrado deficientes en la estimación.

Palabras clave

Punto de observación, *Eye-tracking*, rastreador, puntos característicos, seguimiento, rostro, ojo, pupila.

Abstract

The aim of this Thesis is to study different eye trackers that are available on the market, specifying which of them can be useful. Similarly, this final project aims to implement an eye tracker that identifies a user's observation point in a computer screen through a webcam.

To achieve this, we start by researching the different types of eye trackers, analyzing which one of them is the best suitable for this project. Once we find out the best eye trackers for the project, different software with free license and that use a webcam to capture the images are considered. After this study, the attention has been focused on three softwares, performing a deeper analysis of their operation.

After not getting enough results, a new program has been implemented. From a set of characteristic points and assuming a series of premises, this program detects the user's observation point on the computer screen.

To obtain the observation point, it is necessary to extract characteristic points and a previous calibration. In this calibration the distance between the user and the computer screen is detected. Once this is done, the problem is studied by separating the horizontal and vertical positions of the user's pupil. In order to study the horizontal position, the projections of the pupils are calculated. To get the vertical position, the program uses a series of trigonometric computations. This software treats each eye separately.

After this study, a series of tests was carried out in order to be able to verify the reliability of the software. This leaves as future work the points that were found in the estimate.

Keywords

Observation point, eye-tracking, tracker, characteristic points, monitoring, face, eye, pupil.

Agradecimientos

Para empezar, me gustaría agradecer este trabajo a mi tutor, Jesús Bescós, por ayudarme y animarme en este último año.

También me gustaría dar las gracias a todos aquellos que han estado a mi lado durante estos cuatro años, aquellos que desde el primer momento confiaron en mí, y me han animado a seguir siempre adelante.

Gracias a toda mi familia, que siempre ha estado en los buenos y malos momentos, dándome las fuerzas suficientes para seguir.

Gracias papá y mamá, gracias por apoyarme a la hora de hacer esta carrera y en todas las decisiones que siempre he escogido, gracias por el apoyo incondicional, por estar ahí disfrutando de los buenos momentos conmigo y dándome ánimos en los peores, gracias por haberme dado tan buenos valores y haber conseguido que haya llegado donde estoy, sois lo mejor que tengo.

Gracias a mi hermano, que aunque discutamos mucho, sabe que le quiero mucho.

Gracias a todos esos amigos que siempre están, esos que me ha dado la carrera o esos que vienen de toda la vida, todos esos telecos, Casus, mi gente de Ayllón, mis Manflas... os quiero amigos.

Gracias Marta, hemos comprobado que ni la distancia ni nada va a poder jamás con nosotras, somos invencibles. Te quiero mucho hermanita.

Gracias a Lucía, Marina, Silvia, Basarte, Julia, todos sabemos que mi trabajo sin vosotros no hubiese sido posible, gracias por el tiempo empleado en mi desinteresadamente, sois verdaderos amigos.

Y en especial, quería dar las gracias a aquella persona que ha estado desde el primer momento de esta carrera hasta el último junto a mí, aquella que cuando yo estaba mal ella tiraba para arriba, aquella que ha conseguido sacar lo mejor de mí, y, bueno, aquella que merece la mitad de mi carrera. Gracias por ser como eres, nos queda mucho recorrido juntas, Ali, te quiero.

ÍNDICE DE CONTENIDOS

1. Introducción	1
1.1. Motivación.....	1
1.2. Objetivos.....	1
1.3. Organización de la memoria.....	2
2. Estado del arte	3
2.1. El 'Eye-Tracking'	3
2.2. Rastreadores oculares	5
3. Primeras pruebas	8
3.1. OpenGazer	8
3.2. OpenEyes.....	9
3.3. PyGaze.....	10
4. Puesta en marcha del sistema	14
4.1. Extracción de puntos característicos.....	14
4.1.1 Pruebas iniciales	15
4.1.2 Replanteamiento de objetivos.....	16
4.2. Calibración del sistema.....	17
4.3. Estimación del punto de observación a partir de la posición de la pupila.....	19
4.3.1. Posición horizontal ojo izquierdo	22
4.3.2. Posición horizontal ojo derecho	25
4.3.3. Posición vertical de ambos ojos.....	26
5. Pruebas	29
5.1. Descripción de las pruebas	29
5.1.1 Fase de calibración	30
5.1.3 Fase de operación	31
6. Conclusiones y trabajo futuro	36
6.1. Conclusiones.....	36
6.2. Trabajo futuro	36
Referencias	37

ÍNDICE DE FIGURAS

FIGURA 2-1: TOBII EYE TRACKER	4
FIGURA 3-1: CARA Y OJOS POTENCIALES.....	11
FIGURA 3-2: CARA Y OJOS POTENCIALES (ALGUNOS ERRÓNEOS).....	11
FIGURA 3-3: ESTIMACIÓN POSICIÓN DE LA PUPILA	12
FIGURA 4-1: IMAGEN COMPROBACIÓN <i>PyGAZE</i>	15
FIGURA 4-2: CALIBRACIÓN DEL SISTEMA	17
FIGURA 4-3: IMAGEN ESCRITORIO.....	18
FIGURA 4-4: IMAGEN USUARIO.....	19
FIGURA 4-5: DISTANCIAS OJOS.....	20
FIGURA 4-6: DISTANCIAS HORIZONTALES.....	21
FIGURA 4-7: POSICIONES HORIZONTALES OJO IZQUIERDO.....	22
FIGURA 4-8: POSICIONES HORIZONTALES OJO DERECHO	23
FIGURA 4-9: DISTANCIAS VERTICALES	24
FIGURA 4-10: POSICIÓN VERTICAL SUPERIOR DE AMBOS OJOS.....	25
FIGURA 4-11: POSICIÓN VERTICAL INFERIOR DE AMBOS OJOS.....	26
FIGURA 5-1: ESQUINAS DEL ORDENADOR Y PUNTOS ESTIMADOS USUARIO 1	30
FIGURA 5-2: ESQUINAS DEL ORDENADOR Y PUNTOS ESTIMADOS USUARIO 2.....	31
FIGURA 5-3: ESQUINAS DEL ORDENADOR Y PUNTOS ESTIMADOS USUARIO 3.....	31
FIGURA 5-4: CUADRANTES DE LA PANTALLA	31
FIGURA 5-5: PUNTO DE OBSERVACIÓN USUARIO 1 A LOS CUADRANTES 2 Y 3.....	32
FIGURA 5-6: PUNTO DE OBSERVACIÓN USUARIO 2 A LOS CUADRANTES 2 Y 3.....	32
FIGURA 5-7: PUNTO DE OBSERVACIÓN USUARIO 3 A LOS CUADRANTES 2 Y 3.....	33

1 Introducción

1.1 Motivación

Debido al auge que las nuevas tecnologías están experimentando en la actualidad, cada vez es más común que un usuario consuma una gran cantidad de contenido audiovisual. Además, también está aumentando el número de cámaras que pueden capturar a las personas y sus acciones en la vida diaria, desde las cámaras de los ordenadores portátiles y los teléfonos móviles hasta cámaras de videovigilancia. El acceso a toda esta información crea un nuevo campo de posibilidades, que permiten utilizar esta nueva información de diversas formas. Por ejemplo, podemos tener datos sobre la forma en la que una persona se mueve, su estado de ánimo analizando sus expresiones o hacia donde dirige su atención basándonos en su mirada. Este trabajo de fin de grado se basará en seguir la línea de investigación de este último ejemplo: la existencia de un dispositivo o aplicación que pueda detectar la zona que un usuario está observando en una pantalla nos puede dar información muy relevante de hacia dónde está dirigiendo su atención.

Por otro lado, cada vez es mayor el número de universidades y centros de enseñanza que ofrecen a sus estudiantes la posibilidad de realizar sus estudios a distancia. Uno de los problemas principales que se encuentran estas instituciones en este tipo de enseñanza es que resulta muy complicado verificar que el estudiante se encuentre presente y realizando las tareas requeridas. El desarrollo de sistemas biométricos ofrece soluciones a esta problemática, garantizando mediante diversas técnicas complementarias (reconocimiento facial, análisis de patrones de uso del teclado y del ratón, etc.) la identidad del estudiante que realiza tareas remotamente. Este trabajo buscará ofrecer nuevas soluciones desarrollando una herramienta de bajo coste que permita comprobar si el estudiante está o no prestando atención a la tarea que realiza. Esto se implementará basándose en lo anteriormente comentado: la zona que un usuario está observando está altamente relacionada con hacia donde dirige su atención.

El primer paso será realizar un análisis exhaustivo de los sistemas y aproximaciones existentes. A continuación, se valorará cuál es el modo más adecuado de dar solución al problema, ya sea adaptando sistemas existentes o desarrollando una solución propia.

1.2 Objetivos

El objetivo principal de este trabajo es desarrollar una herramienta que sea capaz de realizar un rastreo o *tracking* de la mirada de un usuario de la forma más precisa posible. Para conseguir este objetivo se seguirá la siguiente ruta:

- Estudiar los diferentes rastreadores oculares que están en el mercado, puntualizando cuáles de ellos pueden servir de utilidad para este trabajo y aportando los resultados que muestran.

- Desarrollar un programa que, a partir de los resultados extraídos por alguno de los rastreadores estudiados anteriormente, permita estimar el punto de observación del usuario.

1.3 Organización de la memoria

La memoria consta de los siguientes capítulos:

- En el segundo capítulo se presenta el estado del arte en el que se realiza un estudio sobre el rastreo de la mirada, en qué consiste y los pasos de los que suele constar. También los tipos de rastreadores que hay, ya sean simplemente con la cámara web del ordenador o con otro tipo de dispositivo que sirva de ayuda.
- En el tercer capítulo se realizan las primeras pruebas con algunos rastreadores seleccionados de internet. Estos se han elegido buscando que se ciñan al objetivo de este trabajo. Después se hace una exposición de los resultados obtenidos.
- En el cuarto capítulo se explica la puesta en marcha del sistema. Se centra en pruebas iniciales, a partir de las cuales se plantearon cambios de objetivos en el trabajo. Además, se explica la forma en la que se ha realizado la calibración del sistema y la estimación del punto de observación a partir de la posición de la pupila.
- En el quinto capítulo se muestran los experimentos más relevantes realizados y se explica el significado de los resultados obtenidos.
- Para concluir, en el sexto capítulo se explican las conclusiones extraídas tras realizar este trabajo y los posibles trabajos futuros.

2 Estado del arte

2.1 El 'Eye-Tracking'

El término '*Eye-Tracking*' se refiere al concepto 'rastreo de la mirada', el cual hace referencia a conocer el movimiento del ojo con respecto de la cabeza (1). Este rastreo no es una técnica que provenga de la neurociencia, sino un tipo de medición biométrica que puede ayudar a entender movimientos que realiza el usuario de manera inconsciente (2). Al inicio, este término fue utilizado en campos como la biología o la medicina centrándose en el ámbito de la investigación sobre los sistemas visuales, pero hoy en día, el estudio del '*Eye-Tracking*' se utiliza fundamentalmente para definir la forma de interacción de un usuario con el ordenador. (3)

Con el objetivo de desarrollar un nuevo sistema capaz de rastrear la zona de observación de un usuario en una pantalla, se han estudiado distintos videos donde es sencillo extraer la posición del ojo. Algunos de los estudios (4, 5) más recientes indican que la localización y el seguimiento del ojo son de gran utilidad para diversas aplicaciones, desde juegos interactivos hasta poder controlar algunas funciones del propio ordenador mediante la mirada.

A la hora de estudiar el seguimiento de la mirada hay que tener en cuenta que es necesario hacer uso de un rastreador que, con una velocidad de muestreo específica y una precisión determinada, ayude a conseguir un seguimiento efectivo. Estos rastreadores suelen centrarse en analizar dos movimientos oculares posibles. Por un lado, se basan en analizar aquellos lugares donde se mantiene fija la mirada durante más tiempo, ya que estas serán las zonas que más interés tengan para el usuario. Por otro lado, estudian los movimientos involuntarios que realiza el propio ojo. (6)

La gran mayoría de rastreadores estudiados realizan la tarea basándose en los siguientes pasos (7) :

- Detección: consta de capturar la información visual mediante un sensor, siendo en este caso la cámara web de un ordenador. A partir de ahí se procede a realizar el reconocimiento y rastreo del usuario.
- Conciencia: intentar conseguir que el ordenador sea capaz de reconocer a dicha persona a partir de una serie de rasgos.
- Control: estimar aquellos gestos que van a facilitar una información útil, es decir, aquellos sobre los cuales se va a proceder a realizar un posterior estudio.
- Interfaz de aplicación: proporcionar el contexto específico sobre el que se va a mover el estudio, indicando todos los requisitos necesarios para el posterior procesamiento.

La localización de la pupila en imágenes visibles de baja resolución es una tarea complicada de implementar de forma automática ya que en la imagen estará presente tanto

ruido como sombras, oclusiones, cambios de postura, parpadeos... etc. Los rastreadores que han sido estudiados en este trabajo cuentan con técnicas de procesamiento de imagen que sirven para medir la posición de la pupila de forma no invasiva y sin contacto.

La mayoría de los rastreadores oculares existentes, utilizan información extraída del espectro de luz de infrarrojos, ya que se ha demostrado que devuelve resultados más precisos. Sin embargo, estos rastreadores presentan el problema de necesitar un hardware adicional. (8) Por ello, este trabajo está centrado en encontrar un software basado en imágenes para la localización y el seguimiento de ojos en el espectro visible accesible a cualquier usuario. A su vez, es necesario que el método no requiera ningún hardware adicional y que pueda funcionar con cualquier cámara web existente.

Dentro de los rastreadores oculares se pueden encontrar de diversos tipos, dependiendo de su eficiencia y su coste. Según la utilidad que el usuario quiera darle, las especificaciones del rastreador utilizado cambiarán. Cabe resaltar que la mayoría de los rastreadores oculares existentes hacen referencia al uso de cámaras especiales.

Los rastreadores más extendidos en la actualidad son aquellos basados en el uso de luz infrarroja, ya que obtienen resultados muy precisos. El mecanismo consta de un LED infrarrojo el cual provoca un punto de reflexión sobre la esfera ocular. Como dicha esfera es prácticamente redonda, el punto de reflexión es muy preciso. Por otra parte, hay una cámara de vídeo que detecta el punto de reflexión y el centro de la pupila. La dirección de la mirada del ojo puede estimarse a partir de las distancias entre ambos puntos con una simple aplicación lineal. A continuación, se realiza una calibración, por ejemplo, mirando a las cuatro esquinas del ordenador, lo cual permite crear una relación de las medidas que tiene la pantalla dando la posibilidad de realizar diferentes mediciones a partir de todos estos parámetros. Algunos de los rastreadores más presentes en el mercado son el sistema *ERICA* y el sistema *Tobii Eye Tracker 1750*. (9,10)

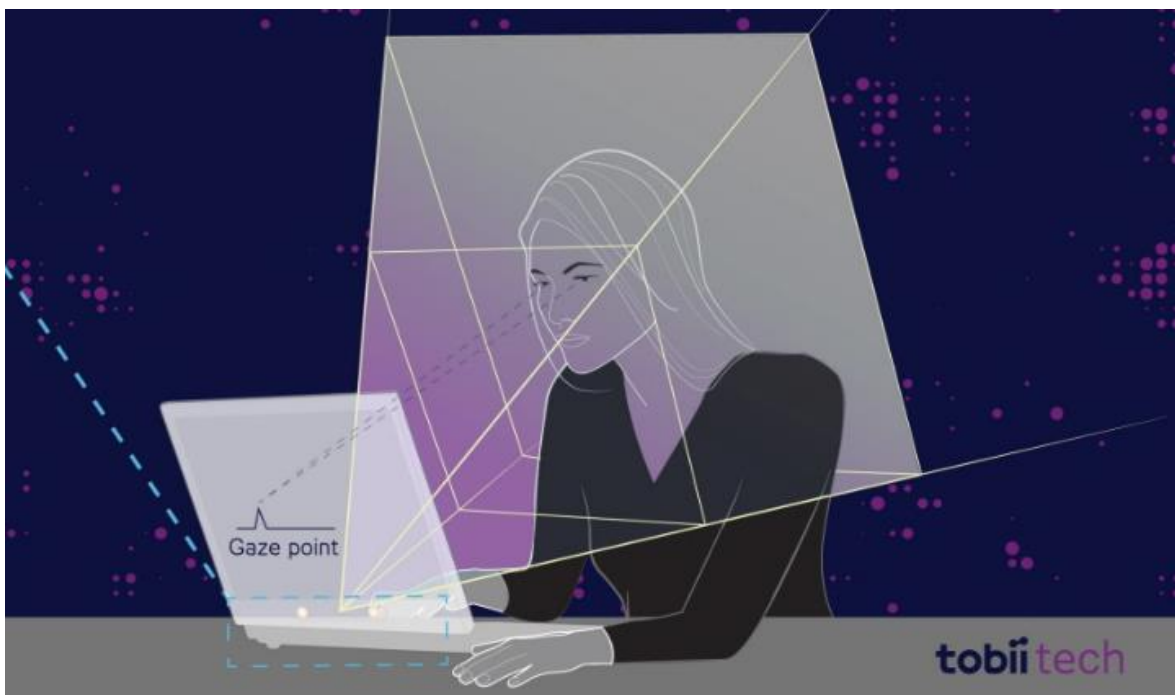


Figura 2-1: Tobii Eye Tracker (11)

La problemática principal de estos dispositivos es la gran maquinaria y software que presentan, lo cual implica un coste elevado difícilmente accesible para el usuario común. Por ello resulta interesante estudiar otras aproximaciones al problema de el rastreo ocular. Gracias a la incorporación de cámaras web en la mayoría de los dispositivos inteligentes, se pueden desarrollar otro tipo de rastreadores que hagan uso de las imágenes captadas de estas. En este caso, la problemática nace de que presentan una menor calidad y exactitud. Sin embargo, la posibilidad de realizar esta tarea con unos costes altamente inferiores hace que esta línea de investigación merezca ser estudiada.

Teniendo en cuenta que este trabajo se va a centrar en un sistema que permita utilizar una cámara web común, se ha realizado un estudio de los rastreadores más populares en este ámbito, indicando: sus principales características, el lenguaje de programación que han utilizado y el tipo de licencia que poseen, centrándose de forma más exhaustiva en aquellos que tienen una licencia que nos permita interactuar con ellos. (12,13)

2.2 Rastreadores oculares

En primer lugar, *Tobii pro*, tiene un software que permite su uso a partir de una cámara web normal, *Sticky*. Es una plataforma biométrica de medición de la emoción y está basado en la nube. Debido a esto se pueden realizar estudios demográficos específicos. También facilita la configuración de paneles en todo el mundo y permite pruebas con las imágenes grabadas a gran escala con un tiempo de respuesta rápido. Sin embargo, no es de código abierto por lo que no es posible estudiar sus códigos, ni conocer el lenguaje de programación en el cual está escrito. (14)

EyeSee es una empresa de investigación del mercado de alta tecnología que desarrolló un software en línea de seguimiento ocular basado en cámaras web y en codificación facial. Su principal finalidad es el marketing. Por un lado, se produce un seguimiento de la mirada mientras el usuario presta atención a los diferentes incidentes que se desarrollan en la pantalla. Tras esto, elabora mapas de calor con los puntos donde el usuario ha centrado más su atención. Por otro lado, cuenta con una codificación facial la cual detecta expresiones faciales y las agrupa en siete grandes grupos: felicidad, sorpresa, confusión, disgusto, miedo, tristeza y neutralidad. No tiene un software libre por lo que no es posible manejarlo, ni saber qué tipo de lenguaje de programación utiliza. (15)

El rastreador *GazeFlow* es un software que incluye un motor de seguimiento mediante una cámara web, analizando el flujo óptico de las imágenes que se introducen mediante dicha cámara. Se beneficia de un alto nivel de precisión. Además, dicho software tiene principalmente tres aplicaciones: *GazeRecorder*, *GazePointer* y *GazeBoard*. *GazeRecorder* es la encargada de efectuar la identificación de caras, ojos, iris y los movimientos 3D que realice el ojo humano. Por ello, se puede decir que esta aplicación ejecuta el rastreo ocular propiamente dicho. Posee un alto nivel de precisión debido a que es inmune al movimiento de las personas, cambios de iluminación o a pequeñas obstrucciones en la cara del usuario a rastrear. *GazePointer* es otra aplicación que, a partir de lo rastreado con *GazeRecorder*, indica la posición del ratón dentro de la pantalla controlándolo con el movimiento ocular. Por último, *GazeBoard* permite la entrada de texto a partir de lo grabado por la cámara web. El lenguaje de programación utilizado por dicho software no viene especificado dentro de la información útil del programa. La única aclaración que aporta es el requisito

de su ejecución en el sistema operativo Windows. Al hablar de la licencia se manifiesta que es de uso libre, exceptuando la utilización del programa para un uso comercial. (16)

Otro rastreador es *WebGaze.js* el cual maneja su software para identificar la ubicación de la mirada en internet mediante una webcam común. El programa tiene auto calibraciones, que se desarrollan a partir de los clics que se efectúan con el ratón en la pantalla y de los movimientos del cursor sobre la misma. Cabe resaltar que el programa solamente permite su ejecución si el usuario autoriza el acceso a la cámara web. Además, utiliza un modelo de regresión para lograr realizar el seguimiento. Este modelo es el llamado modelo de regresión de cresta, el cual asigna píxeles detectados en la pantalla como lugares a los que está observando el usuario. El software está escrito en JavaScript y se ejecuta directamente en el navegador del usuario, realizando un mapa de puntos calientes con las posiciones en las que el usuario enfoca más la mirada. Los navegadores que soportan dicho software son Google Chrome, Microsoft Edge, Mozilla Firefox, Opera y Safari. Tiene una licencia GLPv3, es decir, es de software libre. (17)

XLabs realiza un seguimiento en tiempo real, mediante cualquier cámara web, de la cabeza, los ojos y la mirada. El sistema se desarrolla en el ordenador del usuario, lo que permite una privacidad de los datos. A su vez, éste funciona únicamente cuando el propio usuario consiente el acceso de la cámara al modo de grabación. El programa consta de cinco modos: apagado, cabeza, ratón, aprendizaje y definido por la página. Cabe resaltar que este último modo no puede ser elegido por el usuario, ya que será activado automáticamente cuando la página lo permita.

- El modo cabeza realiza un rastreo de la cabeza, permitiendo a las páginas web acceder a las posiciones X, Y, Z de esta y a los ángulos de balanceo, inclinación y guiñada. Esto lo origina mediante un juego de dos cruces, siendo la cruz verde la que indica la posición de la cabeza y la azul la orientación de la misma.
- El modo ratón permite trabajar simplemente con los movimientos de la cabeza captados con el modo anterior, habilitándose un punto rojo en la pantalla que muestra la posición del cursor. Para ello se requiere una calibración previa. También proporciona la opción de hacer clic descansando la mirada en un punto durante un tiempo.
- El modo aprendizaje es el relevante en este trabajo, efectuando un seguimiento de la mirada. Muestra la posición de la misma mediante con un círculo rojo. En este modo el sistema no requiere calibración explícita, si no que la calibración se realiza a partir de estudiar la mirada y los posiciones donde se haga clic.
- El modo definido por la página, como ya se ha especificado previamente, no se puede establecer, pero sí que se permite apagarlo. Muestra las características del seguimiento de mirada o de cabeza.

El empleo del software se limita a propósitos individuales sin ánimo de lucro, por ello si dicho programa es destinado a fines comerciales se requiere una licencia de ese carácter. Para hacer uso de dicho software y conocer el lenguaje de programación en el que está redactado es necesario ponerse en contacto con la empresa y explicar la finalidad del programa. (18)

TurkerGaze es otro rastreador redactado en *JavaScript*. Obtiene datos del seguimiento ocular mediante una webcam para predecir de donde proviene la mirada del usuario. Además, es un programa de software libre. (19)

OpenEyes contiene diseños tanto de software como de hardware que son útiles para efectuar un seguimiento de la mirada humana. El software que presenta contiene algoritmos que abarcan tanto técnicas para calibrar sistemas de movimiento ocular, como métodos para estudiar oscilaciones oculares dentro de un vídeo previamente grabado. También incluye un software que ayuda al desarrollo de aplicaciones de seguimiento a tiempo real. Además, *OpenEyes* se beneficia del espectro visible de iluminación. Por último, está escrito en el lenguaje de *MatLab* y posee una licencia de libre acceso. (20)

OpenGazer es otro software que investiga la dirección de la mirada. Tiene como principales programas un rastreador de la mirada y un rastreador de cabeza y de cambio de gestos. El rastreador de cabeza y de cambio de gestos se creó para intentar estabilizar los problemas en cuanto al movimiento de la cabeza. Contiene algoritmos de seguimiento de rostro, los cuales evitan la intervención del mismo en el seguimiento de la mirada. Estos algoritmos constan de una reducción a la posición de la cara a partir de un filtro paso bajo autorregresivo. En cuanto al cambio de gesto el programa tiene una pequeña fase de aprendizaje para cada gesto que permite que el gesto se detecte automáticamente. A su vez, contiene algoritmos que detectan los gestos involuntarios de las personas a partir de modelos de fondo. La primera versión de este programa constaba de tres fases: selección del punto característico, calibración del sistema y el rastreo propiamente dicho. Está escrito parte en *C++* y parte en *Python*. Goza de un software libre. (21)

Por último, **PyGaze** realiza un rastreo del rostro a partir de una imagen guardada o una instantánea tomada por la cámara web. Seguidamente rastrea los ojos y la pupila. Es de código abierto y está escrito en *Python*. (22)

3 Primeras pruebas

Una vez hemos analizado la mayoría de los rastreadores oculares existentes, se han seleccionado los tres que se han considerado mejores de cara a solucionar lo que este trabajo propone. Los parámetros estudiados que han decantado la selección de algunos rastreadores frente a otros han sido la licencia que presentan, su accesibilidad y la posibilidad de ser utilizados con cualquier cámara web. Finalmente, tras la deliberación realizada los tres rastreadores oculares a estudiar han sido *OpenGazer*, *OpenEyes* y *PyGaze*.

3.1 OpenGazer

El primer programa que ha sido estudiado es *OpenGazer*. Se caracteriza por estar escrito en C++ y en Python. Este programa tiene principalmente dos fases, la primera se basa en el seguimiento de la cabeza y la segunda es la visión de conjunto, es decir, el rastreo de la mirada propiamente dicho. (21)

En primer lugar, se estudia el seguimiento de la cabeza cuya finalidad es prevenir que la propia cabeza, al realizar movimientos, interfiera en el rastreo de la mirada. El rastreo de la cabeza puede ser de un vídeo previamente grabado o de un vídeo extraído directamente por la cámara web. Este vídeo se examina buscando una cara que seguir, sabiendo que, si encuentra varias caras, el rastreador se centrará en aquella que aparezca en primer plano. Para este proceso utiliza un filtro paso bajo en las coordenadas que se supone que debería de estar el rostro, centrándose en estas coordenadas para el posterior estudio de la mirada del usuario. (21)

En segundo lugar, el programa se adentra en el rastreo de la mirada. Éste consta de tres grandes fases: selección de puntos característicos, calibración del sistema y por último el *tracking* o rastreo propiamente dicho. (21)

- Dentro de la selección de puntos característicos el usuario es el que, mediante el ratón del ordenador, extrae uno a uno aquellos puntos de mayor relevancia dentro de la imagen. En primer lugar, se seleccionan los extremos de los ojos, los cuales son de vital importancia para cálculos posteriores. Cabe resaltar que a la hora de efectuar esta elección es necesario mantener la cabeza lo más estática posible para no interferir en el cómputo.
- En la calibración del sistema se etiquetan varias zonas en la pantalla mediante una serie de puntos rojos. A continuación, se solicita al usuario que se fije en dichos puntos, extrayendo así instantáneas del rostro del usuario enfocando los ojos en dichos lugares. A partir de dichas posiciones oculares y la posición de los puntos marcados en la pantalla, se efectúa un proceso gaussiano el cual representa un mapeo entre la imagen de un ojo y la posición en la pantalla, estimando así las medidas necesarias para el posterior rastreo. Cabe resaltar que cualquier movimiento de la cabeza necesitaría una recalibración del sistema.
- En último lugar se encuentra el rastreo propiamente dicho. Este paso es posible gracias a una calibración previa. Se desarrolla una distribución predictiva que

ayuda a conseguir saber el punto de enfoque visual del usuario en el monitor a partir de una instantánea del rostro de dicho usuario.

Para poner en marcha el programa es necesario realizarlo desde el sistema operativo Linux, instalando una serie de librerías para su puesta en marcha (*vxI* (C++), *Boost* (C++), *OpenCV*...). Tras configurarlas, se procede a instalar el programa. Al ejecutar el programa con las diversas librerías instaladas, se observaban errores de compilación que, debido a mis conocimientos de programación en este entorno, no supe solventar. Por ello, al estar tres semanas intentando resolver el problema sin solución aparente, decidimos continuar estudiando otros rastreadores.

3.2 OpenEyes

Dentro de este programa se ha estudiado el software *Visible Spectrum Starburst*, recordando que el programa está escrito en el entorno de MatLab. El algoritmo para el seguimiento de ojos sigue un esquema de tres pasos: extraer imágenes del vídeo, calibración y procesar todo el vídeo. (20)

- Para la extracción de imágenes de video se utiliza la función *extract_image_from_video.m*. Es importante resaltar que este sistema no es a tiempo real, es decir, solamente utiliza vídeos previamente grabados. Por tanto, el programa utiliza dos vídeos, uno de la escena y otro de imagen visual del usuario. Extrae las imágenes del ojo y escenas de los vídeos y las deposita en los directorios *./Eye/* y *./Scene/* respectivamente.
- La calibración se realiza a partir de la función *extract_calibration_from_video.m*. El usuario necesita especificar el directorio en el que se encuentran las imágenes previamente grabadas, es decir, *./Eye/* y *./Scene/*. A continuación, se le permite al usuario seleccionar aquella imagen que desee analizar a partir de los botones del ratón. Una vez escogidas tanto la instantánea de la escena como la imagen del rostro que está contemplando dicha escena, el usuario etiqueta la posición donde se encuentran el centro de la pupila y el punto de la escena en el que dicho ojo ha centrado la mirada. El programa resaltará una elipse alrededor del centro de la pupila y el punto de la pantalla donde el usuario está observando. Finalmente, el programa archivará, en el directorio actual, el archivo *Calibration.mat* el cual contiene los datos de calibración y una imagen (*Calibration.jpg*) de las correspondencias de los puntos de calibración.
- Por último, el programa se encarga de procesar todo el vídeo a partir de la función *calculate_pupil_and_gaze.m*. El usuario necesita especificar los directorios *./Eye/* y *./Scene/* y *Calibration.mat*. Además, también es el encargado de excluir aquellas malas correspondencias que ha instaurado el programa. Son eliminadas por medio de un clic en la cruz azul que se muestra en la pantalla la cual indica la ubicación de la reflexión de la cornea. Tras esto, el programa calculará la ubicación de la mirada en la imagen de la escena.

Este programa ha sido desechado por varias razones. En primer lugar, es un programa que no trabaja a tiempo real. Además, al tratar de ejecutar el primer paso, es decir, la extracción de las imágenes del video mediante la función *extract_image_from_video.m* se obtiene un

fallo dentro de otra función secundaria *get_first_or_last_frame_num.m*. Esto ocurre debido a que el programa no es capaz de extraer el número del frame en el que se encuentra al no concordar las dimensiones de las matrices con las que se está trabajando. Estas matrices son la imagen del ojo indicada por el usuario mediante la directriz de la carpeta en la que se encuentra y el nombre que el programa que debe tener dicho archivo. Tras una semana estudiando el programa, decidimos continuar buscando otro rastreador con el que se consiguiese obtener el punto de observación del usuario en la pantalla del ordenador.

3.3 PyGaze

PyGaze es un programa capaz de procesar una instantánea del rostro del usuario procedente de un vídeo grabado o directamente de la cámara web. A partir de ahí, según las características de la imagen, se reconoce la cara y los ojos del usuario, obteniendo la posición donde se encuentran los mismos. Se recuerda que el programa está escrito en Python. (22,23)

Además, el programa tiene dos modos, el modo *DUMMY* y el modo *DEBUG*, siendo el usuario el encargado de escoger el modo a emplear. En el caso de manejar el modo *DUMMY*, el programa efectúa el seguimiento a partir de una imagen guardada que se le pasa directamente, escogiendo si la instantánea se encuentra en blanco y negro, en color, o en alguna de las tres componentes de color (rojo, verde o azul). Si el usuario se decanta por el modo *DEBUG* el programa accederá a la cámara web del ordenador realizando una instantánea y actuando a partir de ella. Este segundo modo conecta la cámara, obtiene la fotografía e, igual que en el modo anterior, actúa según el método que el usuario haya especificado (blanco y negro, todas las componentes de color o simplemente alguna de ellas). Finalmente se cierra la cámara web. (22,23)

A continuación, independientemente de la procedencia de la imagen a analizar, se siguen los mismos pasos. Detección del rostro y de los ojos del usuario a partir de los clasificadores '*Cascade*' de las librerías de OpenCV versión 2. Estos clasificadores realizan la detección de objetos a partir de unas características simples mediante las funciones de '*Haar*'. El clasificador de rostros ha sido entrenado con imágenes en las que aparecen rostros y otras imágenes en las que no, extrayendo una serie de características que indican cuales de aquellas imágenes los poseen centrando la atención en ellas. De igual manera se ha utilizado un algoritmo con funciones '*Haar*' para realizar la búsqueda de los ojos.

Posteriormente recorta la cara y los ojos de la imagen con una función que el mismo programa implementa. A partir de la imagen y de un tamaño mínimo que debe tener la cara, éste devuelve el rectángulo más grande en la imagen en el que la cara puede ser detectada. (22,23) (Figura 3-1)

Por otro lado, utiliza otra función para encontrar los ojos en una imagen de la cara, tratando cada ojo por separado. (Figura 3-1)

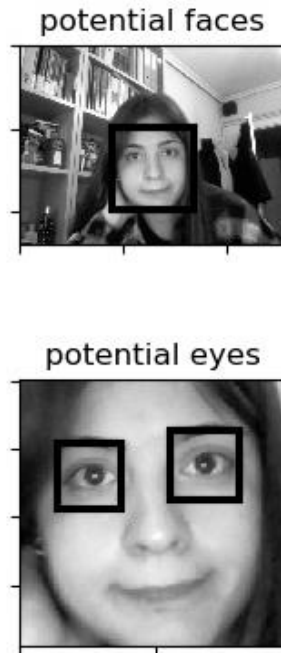


Figura 3-1: cara y ojos potenciales

De igual manera, si en la imagen se localizan varios ojos, el programa operará con aquellos que se ubiquen en la posición más acertada relativa a la cara delimitada anteriormente. Con esto, el programa retorna las posiciones relativas de cada ojo, encuadrándolos en un rectángulo, considerando que el ojo derecho estará en la parte izquierda de la pantalla y el ojo izquierdo en la parte derecha, debido a como ha sido captada la imagen mediante la cámara web. A continuación, el programa cree haber encontrado más posiciones potenciales como ojos, que en este caso son los agujeros nasales. Como se ha mencionado anteriormente, se centra en las posiciones que considera más adecuadas para ubicarse los mismos. (22,23) (Figura 3-2)

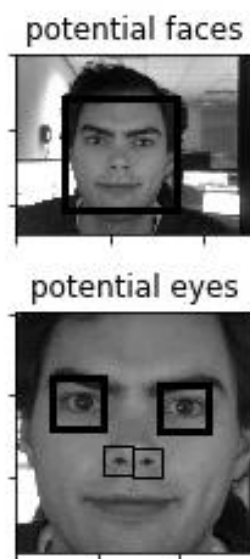


Figura 3-2: Cara y ojos potenciales (algunos erróneos)

A partir de aquí se utiliza otra función implementada en el programa, la cual detecta las pupilas de los ojos. *PyGaze* origina una distinción de si encuentra un destello en el ojo o no, información que define el usuario. Esta función procede a partir de dos vectores que incluyen las posiciones de ambos ojos, un dato que apunta si en el ojo se percibe un destello de luz y el modo que indica como quiere que se devuelva el tamaño de la pupila.

Teniendo en cuenta estos datos, se calcula una matriz que contiene, para cada ojo, las posiciones X e Y y el tamaño de la pupila dentro de la imagen. Este proceso se realiza en otra función para cada ojo por separado, juntándose los resultados en una matriz de la función principal (22,23). Aquí es donde se han localizado los principales errores a la hora de ejecutar dicho programa, ya que tiene problemas al encontrar la pupila dentro del ojo, ya sea con destellos de luz o no. Una de las principales razones por las cuales puede suceder esto es debido a la poca diferencia de nivel de color de la pupila con respecto al resto del ojo al encontrarse con un ojo de color oscuro. Por ello, el programa funciona ocasionalmente y no es posible obtener una cierta fiabilidad. Aquellas veces que consigue encontrar la pupila, dibuja un rectángulo de color verde alrededor de ella identificando el punto en el que se encuentra. (Figura 3-3)

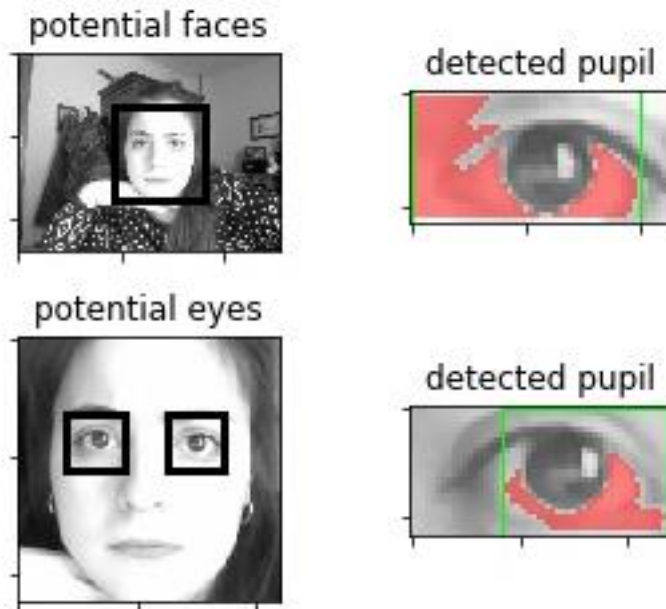


Figura 3-3: Estimación posición de la pupila

El programa finalmente capta la posición dentro de la imagen donde se encuentra la pupila, y solamente funciona cuando tiene unas características muy concretas, como tener luz difusa, captar una imagen clara, entre otras. Por lo que únicamente con este programa no es viable obtener el punto de observación del usuario mediante una cámara web, ya que no hace referencia en ningún momento al lugar que está mirando el usuario en la pantalla del ordenador, sino que simplemente muestra la posición de la mirada dentro de la imagen.

4 Puesta en marcha del sistema

La estimación del punto de observación con el movimiento libre de cabeza es complicada de tratar. Por ello se han adoptado una serie de pasos que han logrado simplificar el problema. La finalidad es averiguar el punto de observación del usuario, siendo necesaria una calibración previa. Tanto para la calibración como para estimar el punto de observación, se extraen los puntos característicos en cada medición a realizar.

La extracción de puntos característicos es transcendental evaluarla desde las características de la imagen y los valores obtenidos en la etapa de medición del ojo, como de algún dato externo más que resulte de gran relevancia. Se ha diseñado un modelo matemático que incorpora varias referencias. La geometría a analizar se fundamenta en la cara del sujeto con importantes puntos de interés; la cámara y la pantalla del ordenador. Para una primera estimación se ha asumido que la cabeza se encuentra en paralelo con la pantalla del ordenador, permaneciendo en una misma línea recta los extremos de los ojos y el centro entre ellos, ubicándose este último a su vez en perpendicular con el centro de la pantalla del ordenador. De la misma manera, se ha estimado que el ojo es una esfera perfectamente redonda, manteniendo una semejanza entre la profundidad y la longitud que guarda éste de extremo a extremo. Por último, también se ha asumido que tanto la cámara web como la lente del propio ojo no producen ninguna distorsión. La extracción es necesaria cada vez que se realice un análisis, tanto para calibrar el sistema como para estimar el punto de observación.

Dentro de la calibración del sistema, tras realizar la extracción de puntos característicos, se asume que el usuario no va a desplazar la cabeza y que ésta va a permanecer centrada durante todo el proceso, estableciendo esto como trabajo futuro. Una vez extraídos los puntos característicos en base a todas las restricciones, es necesaria una calibración del sistema para estimar la distancia a la que se encuentra el usuario de la pantalla del ordenador y a partir de ahí poder estimar el punto de observación.

Por último, se encuentra la estimación del punto de observación. Debido a los puntos característicos extraídos y a la calibración del sistema, se reúne la información necesaria para efectuar dicha evaluación. Esta estimación se desarrolla a partir de la posición en la que se localice la pupila en ese mismo instante.

4.1 Extracción de puntos característicos

4.1.1 Pruebas iniciales

Para las pruebas iniciales se ha manejado el programa *PyGaze* el cual aporta la posición de las pupilas dentro de una instantánea del rostro tanto en horizontal como en vertical. Este programa goza de dos opciones, en la primera la imagen es aportada por el usuario al programa y en la segunda la instantánea es captada mediante la cámara web del dispositivo. El primer propósito de este trabajo es averiguar si la posición que obtiene *PyGaze* referente a las pupilas es fiable. Para comprobarlo, se va a examinar dicho programa a partir de la opción en la que el usuario es el que aporta la imagen.

Al ejecutar el programa *PyGaze* con la imagen de la Figura 3-2 se obtienen las siguientes posiciones para las pupilas, (30, 13) para el ojo izquierdo y (38, 15) para el ojo derecho. Estas posiciones están escritas en píxeles siendo el primer número dentro del paréntesis la posición horizontal en X y el segundo la posición vertical en Y. Cabe resaltar el empleo de píxeles para la comprobación de dicho ejercicio debido a que en ambos programas se está comparando la misma imagen y, por consiguiente, tienen las mismas dimensiones.

Para verificar el funcionamiento de dicho programa se ha ejecutado un pequeño código en el entorno de *MatLab* en el cual a partir de una imagen es posible extraer a mano las posiciones de las pupilas. Mediante este programa se obtiene la posición de la pupila del ojo izquierdo, (217, 231'6), y la posición de la pupila del ojo derecho, (339, 237) (Figura 3-2). Según la comparativa de estos dos resultados se contempla que no son semejantes.

Debido a la obtención de la posición de las pupilas mediante *MatLab* manualmente, se considera este resultado como apto mientras que la solución alcanzada por *PyGaze* se valora como errónea. Otra manera de verificar que el resultado aportado por *MatLab* es el correcto es contrastando las dimensiones en píxeles de la imagen, teniendo ésta una superficie de (480,640) (Figura 3-2). Fundamentándose en esto y comprendiendo que la posición de las pupilas está debidamente centrada en la imagen, se puede comprobar que el resultado aportado por el programa *PyGaze* es incorrecto.

Para verificar esto se ha comprobado con más imágenes. El programa *PyGaze* ha determinado que la pupila del ojo izquierdo está en la posición (36,12) mientras que la del ojo derecho está en (21,11) (Figura 4-1). Por otro lado, comparando la posición con el código creado en *MatLab*, es decir, extrayendo la posición de las pupilas a mano se comprueba como la verdadera posición para la pupila del ojo izquierdo es (623,710) y para la pupila del derecho es (280,285) (Figura 4-1). Tras este estudio se han analizado los datos y se ha deducido que *PyGaze* extrae la posición de la pupila con respecto a la imagen seccionada de cada ojo. A su vez no es un sistema preciso ya que en esta segunda imagen ha detectado como pupila una pestaña de dicho ojo.

Por último, cabe resaltar que en esta imagen *PyGaze* ha sido capaz de extraer las posiciones debido a la desigualdad de color entre el iris y la pupila. Se ha tratado de estimar la posición de las pupilas con apenas diferencia de color siendo *PyGaze* incapaz de establecer un resultado.



Figura 4-1: Imagen comprobación *PyGaze*

Tras este estudio se verifica como las posiciones aportadas por el programa *PyGaze* son erróneas y no es viable proseguir a partir de ellas, siendo indispensable una estimación de la posición de las pupilas mediante otro método. Por ello, es inevitable desechar los resultados mostrados por el programa *PyGaze*. Se ha determinado continuar mediante *MatLab*, extrayendo estas posiciones manualmente, de la misma manera que se ha efectuado para comprobar la viabilidad de *PyGaze*.

4.1.2 Replanteamiento de objetivos

Al no conseguir la extracción de dichos puntos de forma automática, permanece esta detección como trabajo futuro que afrontar y, por tanto, se decidió reorientar el proyecto. Adquiriendo una solución lógica de la posición de la pupila y considerando todas las restricciones relatadas, independientemente del mecanismo de extracción de dichas posiciones, es ineludible ligar la posición correcta de las pupilas con la posición de la pantalla que está siendo observada. Esto se desarrollará mediante un sistema de proyecciones de la pupila en la pantalla del ordenador.

Para esta siguiente etapa es necesaria una calibración la cual posibilite estimar la distancia a la que se ubica el usuario de la pantalla, para ello se va a desarrollar un programa en *MatLab* basado en cálculos matemáticos con la posición de las pupilas tanto en horizontal como en vertical.

4.2 Calibración del sistema

El objetivo final de la calibración del sistema es calcular las longitudes necesarias para estimar el punto de observación. En primer lugar, esta calibración va a concretar la distancia a la que se localiza el usuario de la pantalla del ordenador, D , obviando como todas las suposiciones relatadas son admisibles; la distancia entre los extremos de los ojos, la longitud de cada ojo y la distancia entre la pupila y el extremo del ojo. De igual manera, se asumen conocidas las coordenadas del origen de referencias que va a utilizarse para este proceso. Todos los parámetros asignados arbitrariamente quedan como trabajo futuro y se plantea en una primera aproximación calcular únicamente D .

La distancia a la que se encuentra el usuario con respecto a la pantalla del ordenador, D , es un dato difícil de adquirir con precisión a simple vista, por ello el programa requiere una serie de imágenes previas para poder estimar dicha separación.

La primera consideración en este apartado es que el usuario, como se ha recalado anteriormente, tiene la cabeza estática durante todo el proceso de calibración y pruebas, variando únicamente la posición de las pupilas.

El objetivo final de la calibración es obtener la distancia D para beneficiarse de ella en el momento de realizar pruebas estimando el punto de observación del usuario. Para dicha calibración solo es necesario estudiar el punto de observación horizontal de ambos ojos, ya que es de ahí de donde se extrae la información relativa a la distancia. Al haber asumido varias premisas, es imposible calcular la distancia D con exactitud. Por ello, para poder ajustarla lo máximo posible, se va a obtener a partir de cuatro puntos de observación diferentes, correspondiendo cada uno de dichos puntos a las esquinas de la pantalla del ordenador, y promediando el resultado que se aporte en cada caso. Indirectamente de estos cálculos también se extrae el valor máximo que pueden tener las posiciones de las pupilas cuando miran dentro de la pantalla del ordenador, ya que se toman los bordes de la misma.

Como se ha mencionado anteriormente, para la estimación de la distancia entre la pantalla y el usuario se van a comprender una serie de cálculos. Para comenzar, saber que este estudio se va a realizar sobre el eje horizontal por lo que solo es necesario tomar los cálculos referentes a este. Al moverse únicamente por este eje la pantalla puede ser representada mediante una línea totalmente vertical. Al ser un estudio realizado en dos dimensiones, esta línea vertical se puede calcular a partir de un punto de la misma. Es posible calcular este punto basándose en las proyecciones de las pupilas en la pantalla del ordenador. Las proyecciones son calculadas a partir de la posición de los ojos y de sus respectivas pupilas dentro de la instantánea previamente tomada. Obteniendo la posición del punto más profundo de cada ojo y la posición de la pupila respectiva, se puede formar una recta que representa la proyección de cada ojo. El punto donde cortan ambas rectas será el lugar donde se encuentra la pantalla.

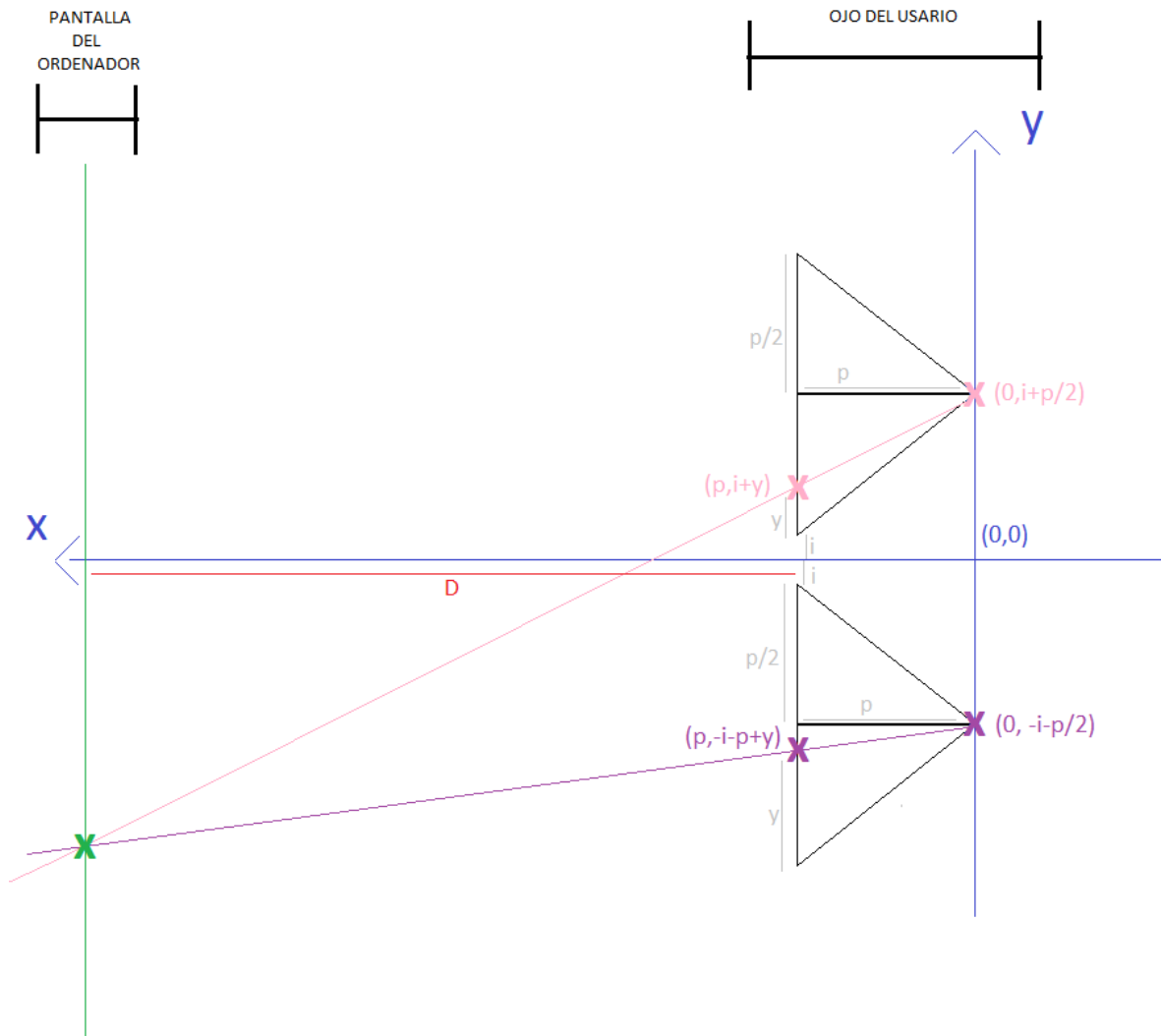


Figura 4-2: Calibración del sistema

En la Figura 4-2 se representan los dos ojos mediante dos triángulos negros. En dichos triángulos se pueden observar dos puntos representativos. Estos puntos son extraídos por el usuario manualmente para poder realizar la calibración. El primero es el punto más profundo del ojo. Este ha sido hallado a partir de la premisa de que el ojo es una esfera perfectamente redonda, por lo que tiene la misma longitud de extremo a extremo que de profundo. El segundo es la posición que presenta la pupila en el eje horizontal. La figura 4-2 muestra como para el ojo derecho los puntos son $(0, i+(p/2))$ y $(p, i+y)$ respectivamente, mostrados mediante dos cruces rosas. De igual manera, los puntos para el ojo izquierdo son $(0, -i-(p/2))$ y $(p, -i-p+y)$ respectivamente, representados con dos cruces moradas.

Al tener estos cuatro puntos se puede estimar la distancia a la cual se encuentra la pantalla. Se realiza una recta con los dos puntos marcados para cada ojo, indicando la proyección de cada pupila. En la Figura 4-2 se puede observar una recta rosa para el ojo derecho y otra morada para el ojo izquierdo. La posición a la que se cortan ambas rectas representa uno de los puntos de la pantalla del ordenador. Este punto está dibujado en la Figura 4-2 de color verde. Recordando que la pantalla es una recta vertical, la posición X de este punto mostrará la distancia a la que se encuentra el usuario de la pantalla del ordenador.

Tras hallar la posición de la pantalla se sabe que está separada una distancia $D+p$ del centro de coordenadas por lo que para hallar la distancia D simplemente es necesario restarle p a dicho valor. Todas las distancias tomadas se pasan a milímetros para poder operar con ellas, y finalmente la distancia D viene dada también en esta unidad de medida.

Debido a la imprecisión de este programa, se plantea el mismo en cuatro ocasiones diferentes, cada una referida a una esquina de la pantalla del ordenador. Finalmente, la distancia D será la media de todas ellas.

4.3 Estimación del punto de observación a partir de la posición de la pupila

A continuación, se va a realizar una estimación del punto de observación mediante un modelo de proyección. Este modelo obtiene las proyecciones de ambas pupilas en la pantalla del ordenador. El foco (punto de partida) de dicho sistema se encuentra detrás de los ojos. Para este programa es necesario tener en cuenta una serie de circunstancias que han sido asumidas: los extremos de los ojos están todos situados en una misma línea horizontal y ésta, a su vez, se encuentra completamente paralela al centro de la pantalla del ordenador, el ojo es una esfera redonda teniendo igual longitud de ancho que de profundo, y, por último, tanto la cámara web como el propio ojo no realiza ningún tipo de distorsión.

El programa requiere dos imágenes, una del rostro del usuario y otra de la pantalla a la cual está observando. Al asumirse que los extremos de los ojos y el centro entre ellos se sitúan justo en paralelo con la línea que marca de manera horizontal la mitad de la pantalla, lo primero que se ha establecido es que el centro de la imagen del usuario corresponda con el centro que se encuentra entre sus ojos. A partir de aquí se le pide al usuario extraer los puntos característicos. Dentro de la imagen de la pantalla del ordenador el centro de la misma (Figura 4-3), y en la imagen del usuario: cada uno de los extremos de los ojos, el centro entre los mismos y la posición de cada una de las pupilas (Figura 4-4).

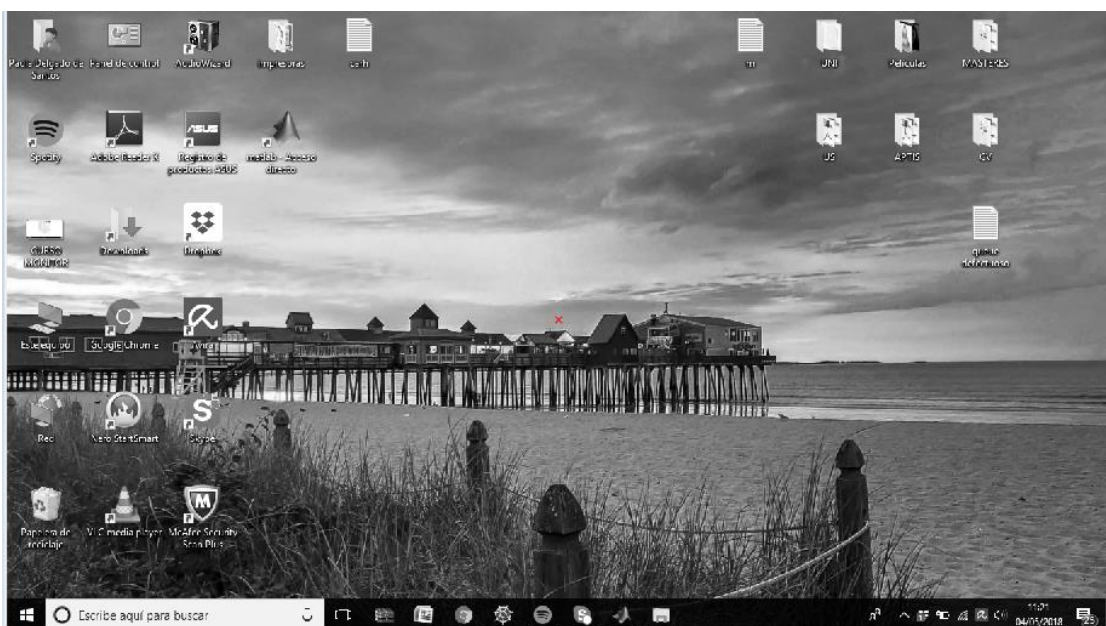


Figura 4-3: Imagen escritorio

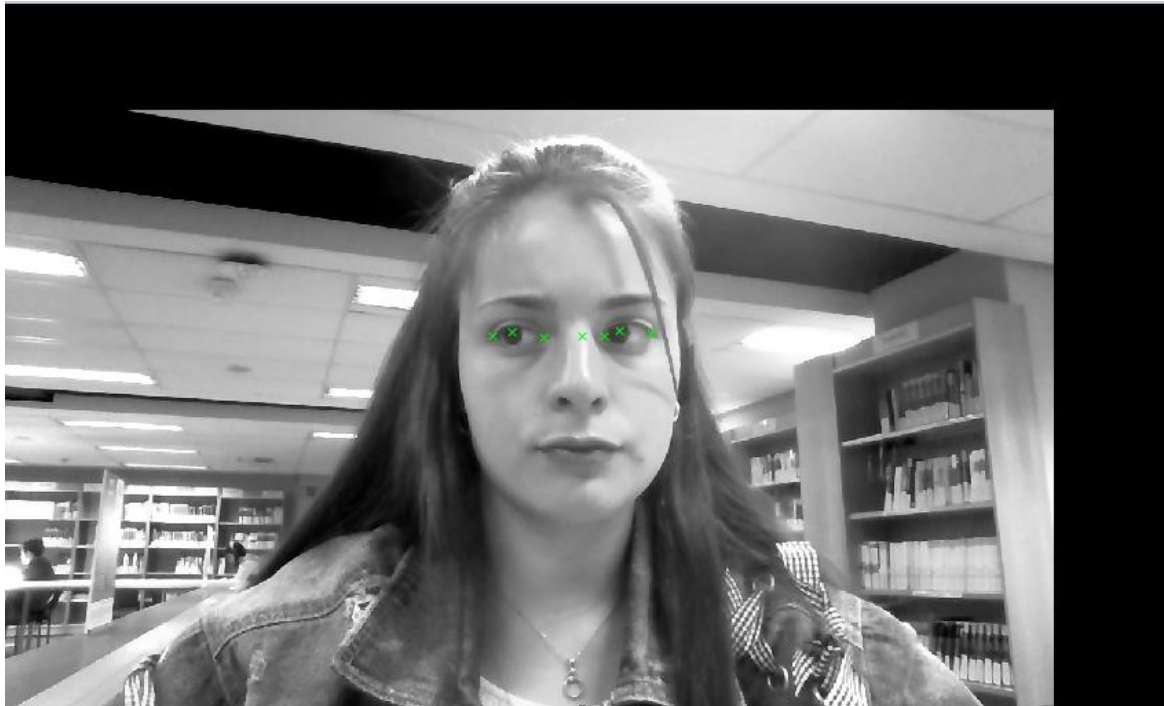


Figura 4-4: Imagen usuario

Al haber realizado una calibración previa, también es conocida la distancia a la que se encuentra el usuario de la pantalla.

Centrando la atención en los ojos del usuario, es necesario saber que la pupila puede oscilar tanto horizontal como verticalmente. Para simplificar los cálculos, se va a realizar un análisis vertical y otro horizontal de la proyección de cada pupila en el ordenador.

Antes de proceder a la estimación propiamente dicha, es necesario aclarar una serie de aspectos. Las imágenes no tienen la misma resolución, es necesario pasar los píxeles a otra unidad de medida, como en este caso a milímetros. En el caso de la imagen de la pantalla se ha calculado midiendo el ancho la pantalla y comparándolo con la distancia que da la anchura en píxeles de la imagen. Para la imagen del usuario se ha medido la distancia entre ambas pupilas equiparándola con la distancia en píxeles de la imagen. A su vez, la imagen del usuario está captada con la cámara web por lo que se debe voltear a partir de un eje vertical situado en el centro de la pantalla. De igual manera, como se ha asumido que el centro entre los ojos está en paralelo con el centro de la imagen de la pantalla, se permite al usuario definir dicho punto para centrar la imagen de la cara con respecto a la imagen de la pantalla.

Se es consciente de que, al efectuar dos análisis del punto de observación en dos dimensiones, uno horizontal y otro vertical, es complicado solventar el problema que provoca que los ojos no estén alineados, por ello se considera que sería necesario un solo estudio en tres dimensiones para simplificarlo, dejando dicho estudio como trabajo futuro.

Tras haber realizado los pasos previos, se procede al análisis del punto de observación. Como se ha recalcado anteriormente, se va a realizar un análisis de las proyecciones de manera horizontal y otro de manera vertical.

Con respecto al análisis horizontal es necesaria una distinción para cada ojo ya que no se sitúan en las mismas posiciones como se ha observado en la Figura 4-2. Partiendo del eje de coordenadas dibujado, el ojo derecho se sitúa por encima del origen de éste y el izquierdo por debajo. Teniendo estas posiciones en cuenta, los cálculos no se realizan de igual manera para cada ojo por lo que habrá que separarlos. Lo que se puede calcular, independientemente del ojo que se esté analizando, es cada una de las medidas necesarias (Figura 4-5). La distancia de extremo a extremo de cada ojo tiene un valor ' p '. La distancia entre el extremo interior de cada ojo y el centro entre los mismos le damos un valor ' i '. Se muestra cada una de las pupilas con una línea morada, indicando como la distancia de cada pupila al extremo interior de cada ojo tiene una distancia ' y '. A partir de todas estas distancias, se puede calcular la posición horizontal de cada uno de los ojos.

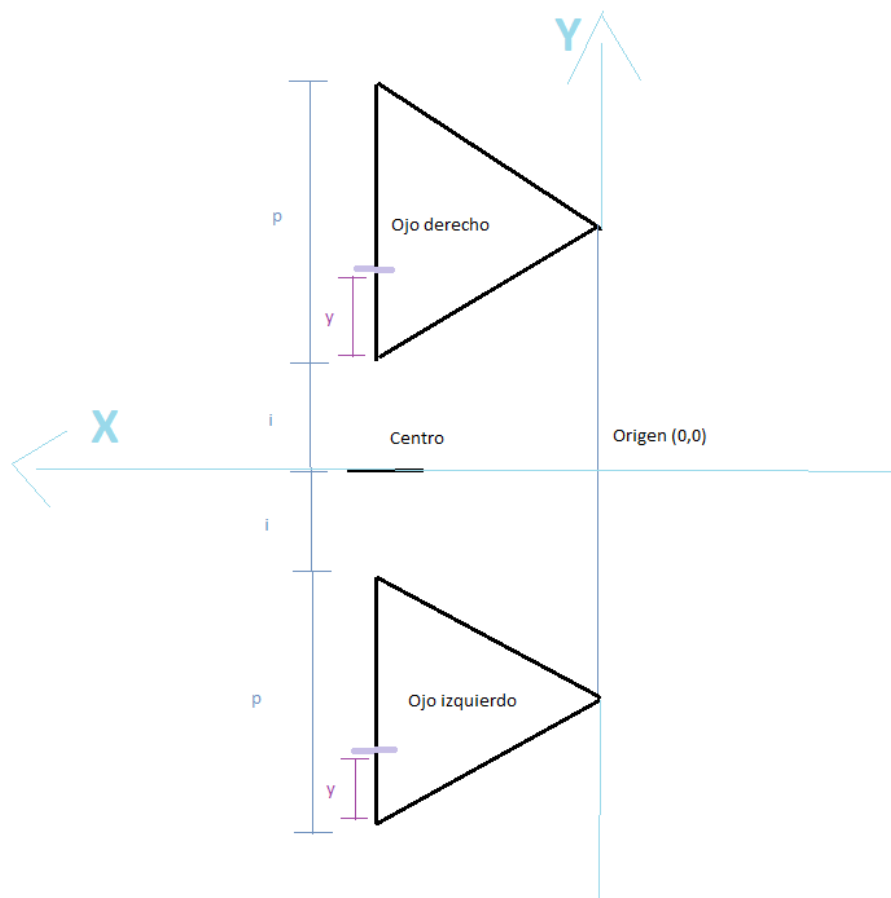


Figura 4-5: Distancias ojos

Al realizar el análisis vertical no es necesario realizar distinción entre ambos ojos. La única distinción que se puede hacer es si el ojo está mirando por encima o por debajo del centro de la pantalla del ordenador.

Aclarado esto, se va a calcular la posición horizontal del ojo izquierdo, la posición horizontal del ojo derecho y la posición vertical de ambos ojos en los siguientes apartados.

4.3.1 Posición horizontal ojo izquierdo

Se va a estudiar la posición del ojo izquierdo por lo que solo hay que centrarse en las distancias relativas a dicho ojo.

Para comenzar se parte de una serie de datos que se han ido extrayendo como son las dimensiones de la pantalla del ordenador ' x_screen ' e ' y_screen ', la distancia a la pantalla del ordenador ' D ', la distancia horizontal del ojo ' p ', y por último la posición de la pupila en horizontal con respecto al extremo externo del ojo ' y ' y en vertical con respecto al centro de los ojos o a los extremos del ojo ' g ' (se recuerda que se ha asumido que tanto los extremos como el centro de los ojos están en el mismo eje horizontal) (Figura 4-6).

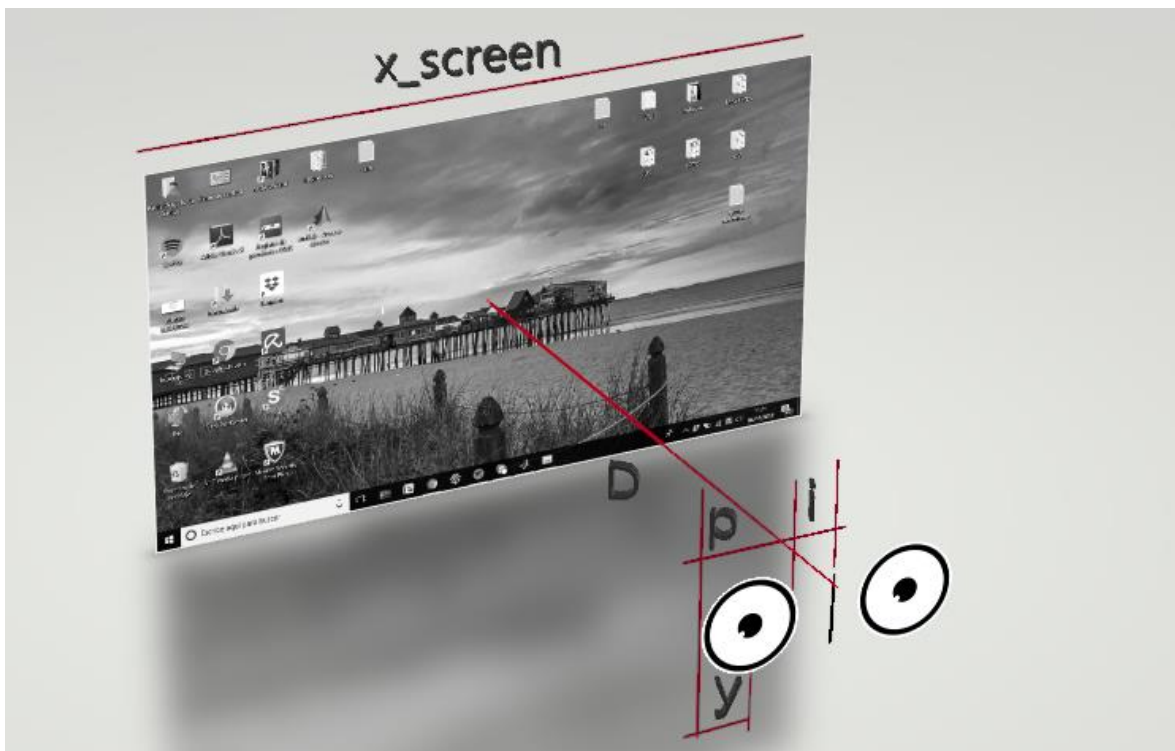


Figura 4-6: Distancias horizontales

A continuación, se muestra una vista aérea del ojo izquierdo y la pantalla analizando el punto de observación del mismo (Figura 4-7).

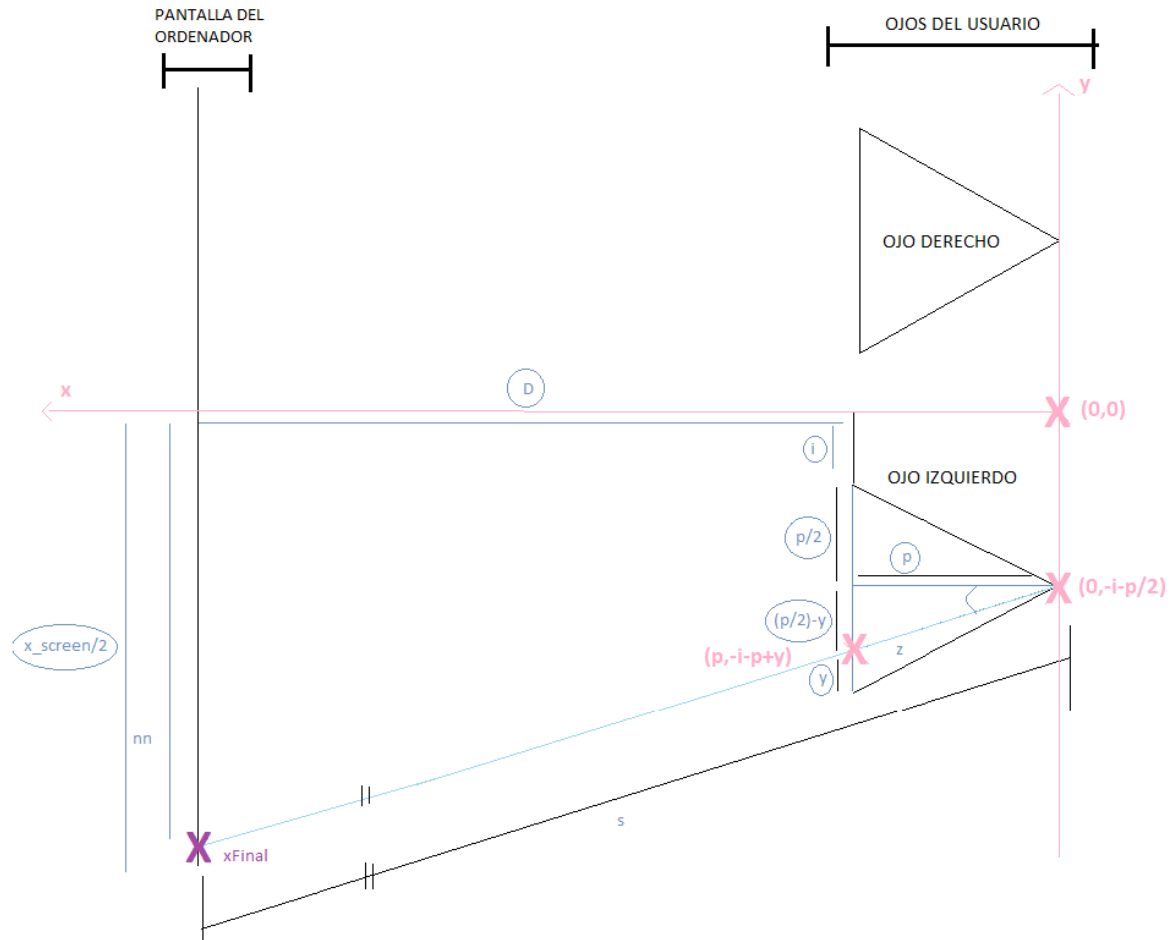


Figura 4-7: Posiciones horizontales ojo izquierdo

La finalidad es llegar a conseguir saber la posición de x_{Final} , para ello se han hecho una serie de reglas trigonométricas, que se van a explicar a continuación. Aquellas distancias rodeadas son conocidas por cálculos previos.

Para realizar el cálculo de la proyección de dicha pupila se ha creado un sistema de coordenadas en dos dimensiones como se puede observar en la anterior figura dibujado de color rosa. Se ha colocado el origen de este eje en la posición central de los ojos, pero a la altura del punto más profundo de ambos, para simplificar los cálculos. A raíz de esto, se han tomado como referencia para calcular x_{Final} dos puntos, los cuales también están dibujados con dos cruces rosas. Estos puntos son el punto más profundo del ojo $(0, -i-p/2)$ y el punto en la pupila del ojo $(p, -i-p+y)$. A partir de aquí se ha trazado una recta con estos dos puntos y, como en este primer cálculo se ha dado por conocida la distancia a la que se encuentra la pantalla, el punto de observación horizontal del ojo izquierdo será el lugar donde corte la recta que ha sido creada a partir de estos dos puntos con la recta que representa la pantalla del ordenador, lo que es representado en la Figura 4-7 como x_{Final} con una cruz morada.

4.3.2 Posición horizontal ojo derecho

Se ha hecho una distinción entre ambos ojos debido a que no ocupan la misma posición y por tanto no tienen las mismas coordenadas los puntos de interés. Cabe recalcar que las referencias y distancias tomadas se calculan de igual manera para ambos ojos, las restricciones a tener en cuenta son las mismas y tanto los ejes de coordenadas como el origen de los mismos se encuentra en el mismo lugar. La única distinción son los puntos tomados para trazar la recta que, al cortarse con la recta que marca la posición de la pantalla del ordenador, se calcula el punto de observación del ojo derecho del usuario en el eje horizontal (Figura 4-8).

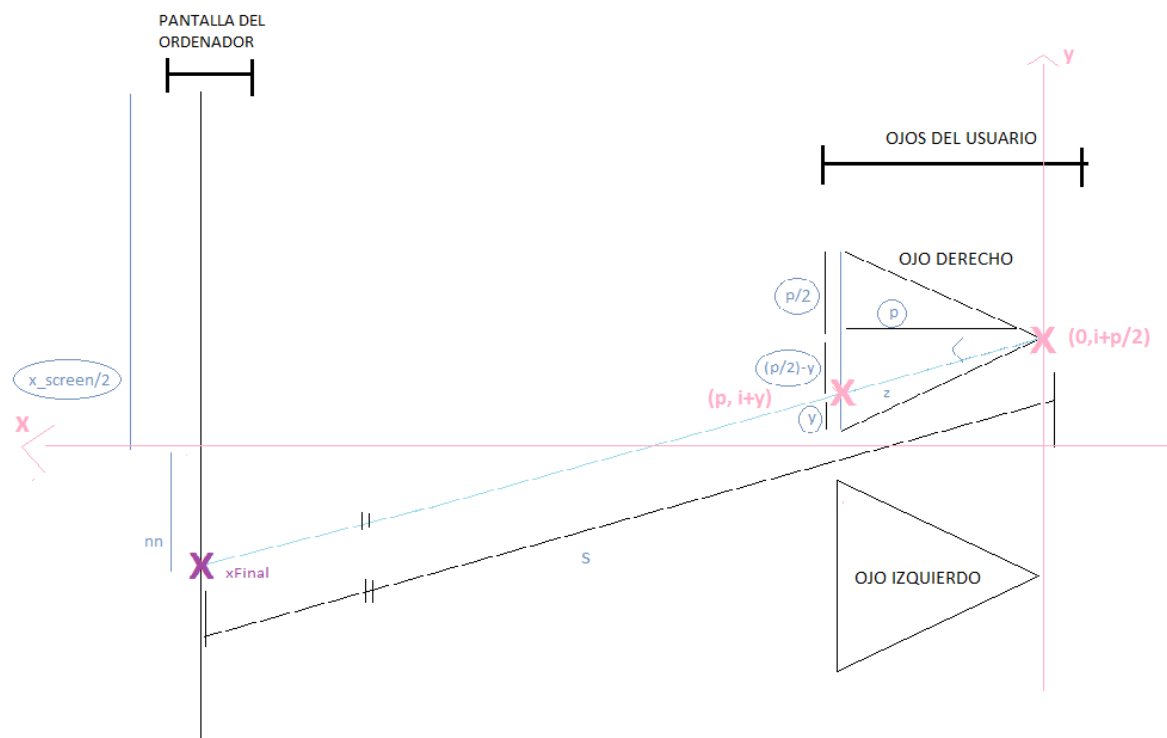


Figura 4-8: Posiciones horizontales ojo derecho

Se puede observar cómo el ojo derecho se encuentra esta vez por encima del eje de ordenadas, lo que implica que las distancias en el eje de abscisas serán tomadas en positivo. Al igual que antes, están marcados con cruces rosas el punto de máxima profundidad del ojo, con la posición $(0, i+p/2)$ y la pupila del ojo que se encuentra en el punto $(p, i+y)$. Como ocurría en el caso anterior, a partir de estos dos puntos se traza una recta la cual al cortar con la recta que representa la pantalla del ordenador muestra el punto de observación del usuario en el eje horizontal (Figura 4-8).

4.3.3 Posición vertical de ambos ojos

Una vez extraída la posición horizontal es necesario extraer la posición vertical. Esta última es igual tanto para el ojo derecho como para el ojo izquierdo ya que ante el eje

horizontal no muestran diferencias entre ambos. La única distinción que se puede hacer es si la pupila está mirando por encima de la mitad de la pantalla o por debajo de la misma.

Para ello hay que tener en cuenta varias distancias. Con respecto a la pantalla, la altura de esta es ' y_screen ', y la anchura es ' x_screen '. También es necesario tener en cuenta la distancia que hay desde los ojos al centro de la pantalla ' D '. El valor ' p ', el cual es tanto el ancho como la profundidad del ojo y ' g ' que es la distancia que hay en vertical de la pupila a los extremos del ojo o al centro entre ambos. A su vez se utiliza el valor ' nn ' mostrado en la figura que representa la distancia que hay desde el centro de la pantalla hasta el punto marcado como ' $xFinal$ ' (Figura 4-9).

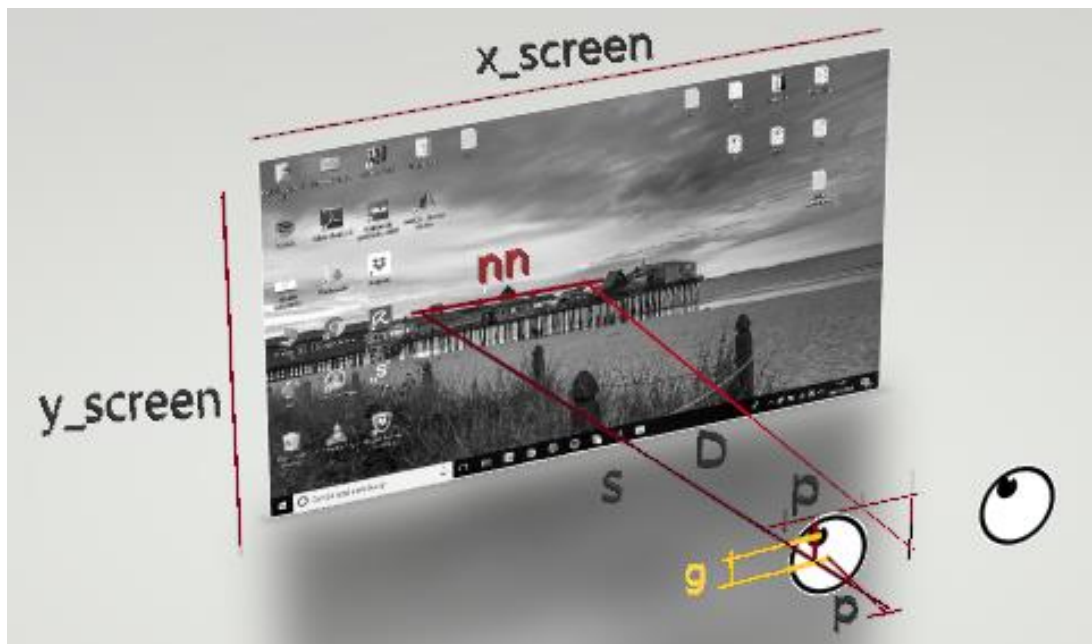


Figura 4-9: Distancias verticales

Tras esta figura se va a estudiar el punto de observación en cuanto al eje vertical, por lo que a continuación se mostrará un dibujo del lateral derecho de cualquiera de los dos ojos mientras está observando a la pantalla del ordenador, indicando los parámetros que hay que tener en cuenta para estimar la coordenada vertical del punto de observación (Figura 4-10).

Hay que tener en mente que, para calcular el punto de observación de manera vertical, es necesario un primer cálculo en el eje horizontal. Por ello se calcula la longitud ' s ' que es aquella que existe entre el punto más profundo del ojo y el punto horizontal de la pantalla del ordenador donde el usuario tiene fijada su mirada. A partir de este punto se va a realizar el análisis vertical (Figura 4-10).

Una vez extraídas las distancias horizontales ' z ' (Figura 4-8) ($z=\sqrt{p^2+((p/2)-y)^2}$) y ' s ' (Figura 4-10) ($s=\sqrt{(p+D)^2+nn^2}$) se puede calcular la posición vertical de manera sencilla. Se aclara que la siguiente figura se trata de la vista vertical derecha de un ojo, por ello distancias como ' $(p+D)$ ' y ' s ' parecen ser iguales, pero es la perspectiva de la imagen (Figura 4-10).

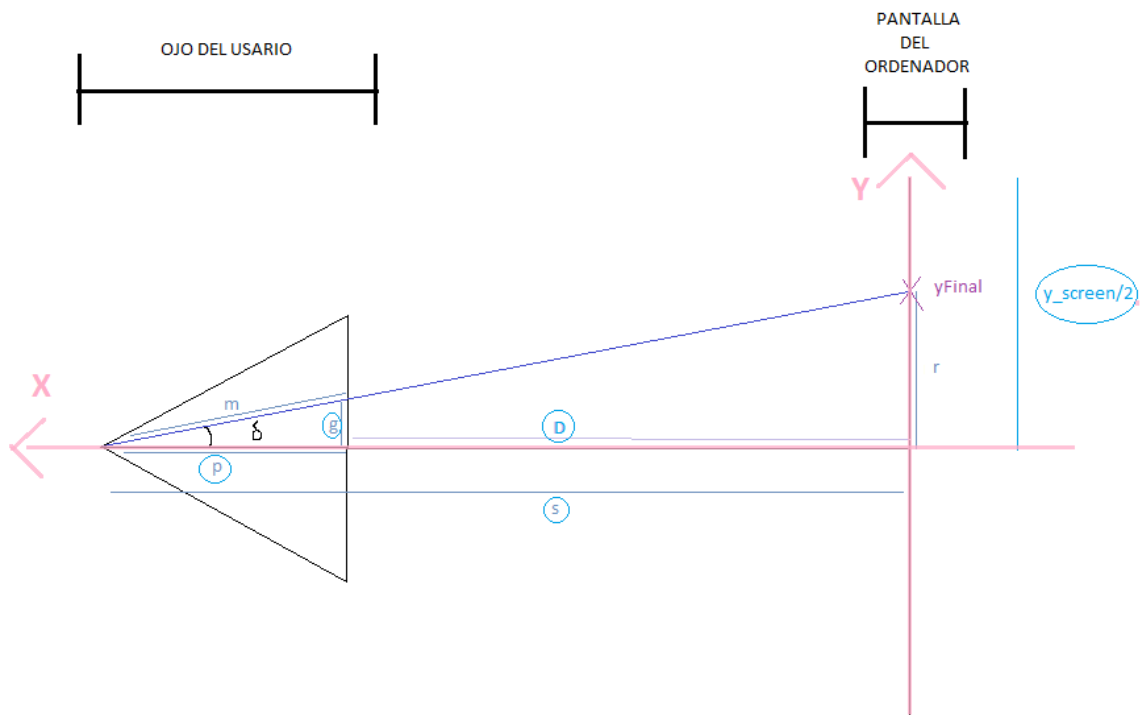


Figura 4-10: Posición vertical superior de ambos ojos

Como se ha explicado previamente, la única distinción a la hora de estudiar las posiciones verticales de las pupilas es si estas están mirando por encima o por debajo del centro de la pantalla.

Por un lado, se va a estudiar la posición de la pupila cuando esta se encuentra por encima del centro de la pantalla (Figura 4-10). La finalidad es llegar a conseguir saber la posición de y_{Final} ; para ello se han hecho una serie de reglas trigonométricas que se van a explicar a continuación. Aquellas distancias que están rodeadas se debe a que son conocidas por cálculos previos. A continuación, se halla ' m ' ($m = \sqrt{p^2 + g^2}$). A raíz de esto se puede calcular ' δ ' ($\delta = \arcsen(g/m)$). Tras esto es posible sacar el valor de ' r ' ($r = s * tg(\delta)$). Por último, se puede obtener el valor final en Y de la estimación del punto de observación, extrayendo la posición ' y_{Final} ' ($y_{Final} = r$) (Figura 4-10).

Por otra parte, se va a estudiar el caso en el que la pupila se encuentra por debajo del punto medio de la pantalla. En este caso los cálculos serían resueltos de igual manera debido a la similitud de las distancias con las de la Figura 4-10. La única distinción que se puede realizar es el último paso, en el momento que se coloca la posición final, ' y_{Final} ', que en este caso la distancia r será restada del centro de coordenadas (Figura 4-11).

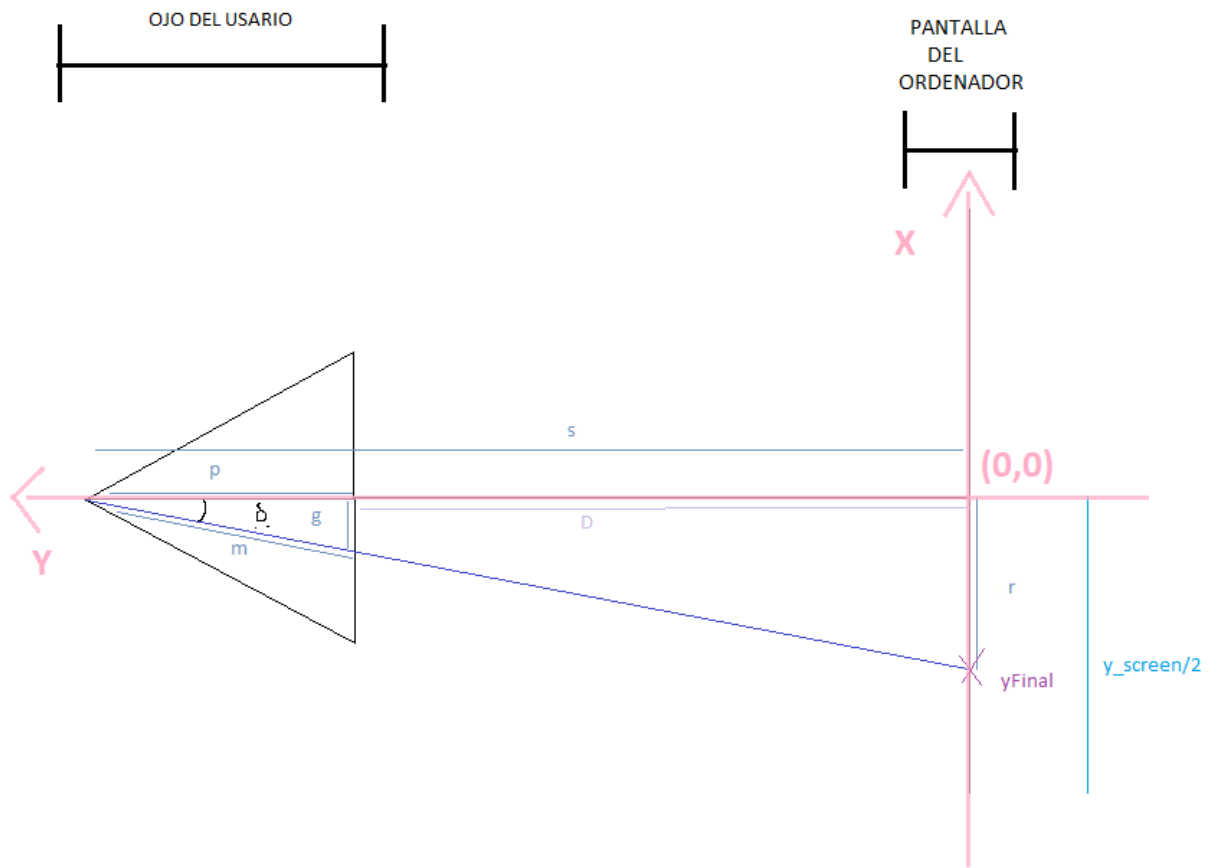


Figura 4-11: Posición vertical inferior de ambos ojos

5 Pruebas

Partiendo del método desarrollado, se han realizado una serie de pruebas para comprobar la fiabilidad del mismo. Como el sistema no ha sido capaz de calcular la posición de las pupilas automáticamente, y es necesario realizarlo a mano, se limita el número de usuarios para realizar las pruebas del programa. Finalmente, se ha decidido realizar pruebas con tres usuarios diferentes.

5.1 Descripción de las pruebas

Se ha determinado realizar el estudio a tres usuarios, debido al alcance de los métodos del programa. Las pruebas que se van a realizar a cada uno de los usuarios están compuestas de dos fases. En primer lugar, se encuentra una fase de calibración y, tras esta, se realiza una fase de operación. Cabe resaltar que es necesaria una extracción de puntos característicos antes de realizar cada una de estas etapas.

Con la extracción de puntos característicos se determinan, dentro de cada una de las imágenes, aquellos puntos que resultan de gran relevancia para nuestro estudio. Se examinan tanto la imagen de la pantalla del ordenador como la imagen del rostro del usuario observando dicho ordenador. En la primera imagen se calcula la posición del centro del ordenador. En la segunda imagen, del rostro del usuario se extrae manualmente las posiciones de ambos extremos de cada uno de los ojos, sus respectivas pupilas y el centro entre ellos. Una vez extraídas dichas posiciones, se asumen una serie de premisas para poder analizar al usuario. En primer lugar, es necesario tener en cuenta las relaciones de las instantáneas tomadas con las medidas de reales de dichos objetos. Para ello, se extrae la distancia del ancho de la pantalla en milímetros. De igual manera se extrae, para cada usuario, la distancia real existente entre sus pupilas.

En la primera fase, tras la extracción de puntos característicos, se realiza una calibración del sistema. Con dicha calibración el programa determina la distancia entre el usuario y la pantalla. Dentro de este apartado se debería extraer todas las distancias necesarias para realizar las pruebas. Como para una primera estimación no han sido obtenidas, se plantea como trabajo futuro aceptando una serie de premisas. Se considera que el centro de la pantalla del ordenador está en perpendicular con el punto medio entre los ojos del usuario. A su vez se considera que tanto los extremos de los ojos, como el punto medio entre ellos, se encuentran paralelos al eje horizontal que determina la mitad de la pantalla del ordenador.

En la segunda fase, tras la extracción de puntos característicos, se realiza una serie de pruebas determinando el punto de observación del usuario. Se asume que, tanto durante todo el proceso de calibración, como durante todo el proceso de pruebas, el usuario permanece con la cabeza inmóvil, variando únicamente la posición de sus pupilas. Al asumir la posición de los extremos de los ojos con respecto de la pantalla, se conoce la imprecisión del programa, ya que el usuario no suele tener la cabeza totalmente colocada según se indica en las premisas. Se comprende que, al realizar los cálculos, es posible que las proyecciones de cada ojo no se dirijan exactamente al mismo punto en la pantalla. Por ello, se propone como solución implementar un sistema en tres dimensiones que trate a los ojos como conjunto y no estudie cada uno de ellos por separado. Basándose en la precisión

del programa, las pruebas a realizar van a consistir en conseguir estimar si el usuario está observando dentro de dos de los cuatro cuadrantes en los que se va a dividir la pantalla del ordenador.

5.2 Fase de calibración

Para realizar los diversos cálculos se han utilizado tres usuarios diferentes.

Lo primero a tener en cuenta es la relación de las medidas de cada imagen. Por un lado, hay que conocer las medidas de la pantalla del ordenador a la que los usuarios están observando, siendo en los tres casos la misma. Por otro lado, se realiza una medición de la distancia entre las pupilas de cada usuario (usuario uno: 58 milímetros, usuario dos: 56 milímetros, usuario tres: 58 milímetros) para poder estimar las relaciones de la imagen del rostro. Gracias a todos estos datos es posible conseguir que todas las distancias estén en la misma unidad de medida, milímetros.

Tras esto, es necesaria una calibración del sistema. Con esta calibración se calcula la distancia a la que el usuario se encuentra de la pantalla. Debido a todas las premisas especificadas anteriormente, se realiza el estudio con cuatro imágenes distintas, calculando finalmente la distancia del usuario al ordenador, D , como el promedio de los resultados de estas cuatro calibraciones. Los cuatro puntos observados por el usuario son las esquinas del ordenador, por lo que de aquí también se obtiene indirectamente la posición máxima que pueden tener las pupilas cuando el usuario está observando dentro de la pantalla.

A continuación, se muestran una serie de tablas que indican el punto que está observando el usuario en la pantalla del ordenador y la distancia estimada a la que se encuentra la pantalla en cada punto. A su vez se muestra el promedio de todos los puntos utilizados para posteriormente realizar los cálculos con dicho valor.

ESQUINA A LA QUE ESTÁ OBSERVANDO EL USUARIO 1	DISTANCIA D EN MILÍMETROS
ESQUINA SUPERIOR IZQUIERDA	511mm
ESQUINA SUPERIOR DERECHA	500 mm
ESQUINA INFERIOR IZQUIERDA	637mm
ESQUINA INFERIOR DERECHA	574 mm
MEDIA TOTAL	555 mm

Tras estas cuatro calibraciones se decide finalmente que la distancia a la que el usuario 1 se encuentra la pantalla es de 555 milímetros.

ESQUINA A LA QUE ESTÁ OBSERVANDO EL USUARIO 2	DISTANCIA D EN MILÍMETROS
ESQUINA SUPERIOR IZQUIERDA	623mm
ESQUINA SUPERIOR DERECHA	524mm
ESQUINA INFERIOR IZQUIERDA	552mm
ESQUINA INFERIOR DERECHA	537mm
MEDIA TOTAL	559mm

El usuario 2 se encuentra a una distancia de 559 milímetros de la pantalla del ordenador.

ESQUINA A LA QUE ESTÁ OBSERVANDO EL USUARIO 3	DISTANCIA D EN MILÍMETROS
ESQUINA SUPERIOR IZQUIERDA	586 mm
ESQUINA SUPERIOR DERECHA	531 mm
ESQUINA INFERIOR IZQUIERDA	536 mm
ESQUINA INFERIOR DERECHA	575 mm
MEDIA TOTAL	557 mm

El usuario 3 se encuentra a 557 milímetros.

5.3 Fase de operación

A partir de aquí se han extraído algunas instantáneas de cada usuario para estimar la precisión del programa. Antes de proceder a la comprobación, estudiando los distintos valores que han sido obtenidos por medio de la calibración, se observa como el programa no va a aportar una precisión muy aguda. Esto es debido a las distorsiones de la lente de la cámara del ordenador, aparte de todas las premisas que han sido asumido previamente, quedando como trabajo futuro.

A partir de todas las posiciones extraídas manual mente y de la distancia D a la que se encuentra el usuario de la pantalla, se va a estimar el punto de observación del usuario tanto vertical como horizontalmente como se ha indicado en el apartado 4.2 de este trabajo.

La primera comprobación que se realiza es estimando el punto de observación de cada usuario en las cuatro esquinas de la pantalla del ordenador. En las siguientes figuras se muestra una representación de la pantalla del ordenador observada cada vez por un usuario diferente. En estas representaciones se encuentran las cuatro esquinas marcadas con círculos verdes. También se incluyen con círculos azules las posiciones estimadas del punto de observación del ojo izquierdo y con círculos rojos las del ojo derecho. Cada círculo azul y cada círculo rojo estiman la posición del punto de observación del usuario relativo a la esquina más próxima (Figura 5-1) (Figura 5-2) (Figura 5-3).

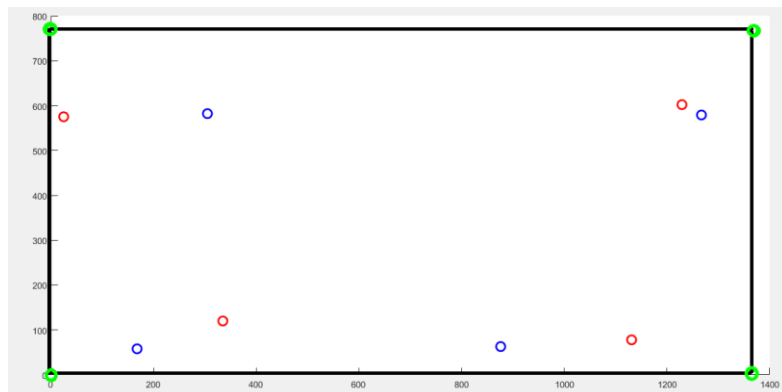


Figura 5-1: Esquinas del ordenador y puntos estimados usuario 1

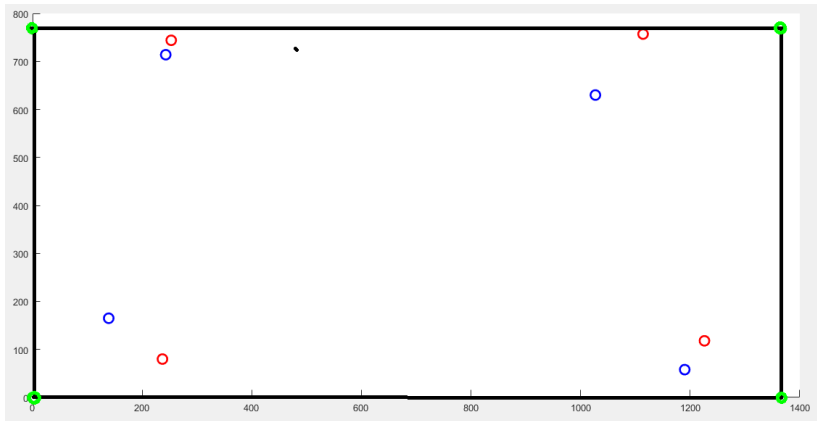


Figura 5-2: Esquinas del ordenador y puntos estimados usuario 2

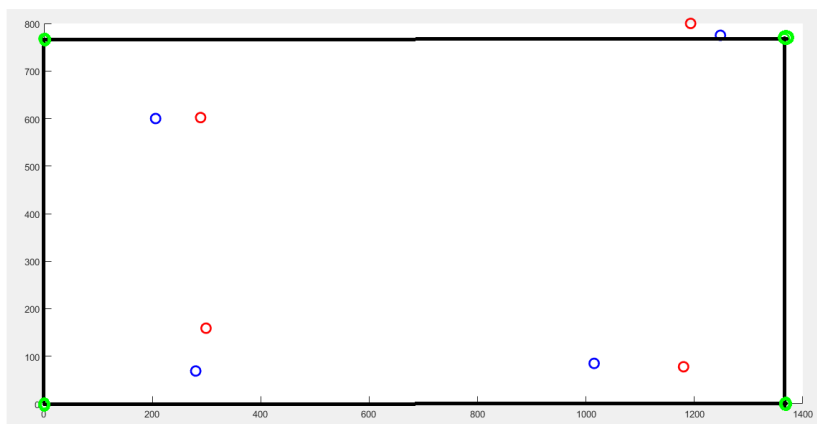


Figura 5-3: Esquinas del ordenador y puntos estimados usuario 3

Tras este estudio se puede observar la escasa precisión del programa. Por ello, los cálculos siguientes se van a realizar de manera cualitativa. Se divide la pantalla en cuatro cuadrantes, acotando la superficie que estudiar (Figura 5-4).

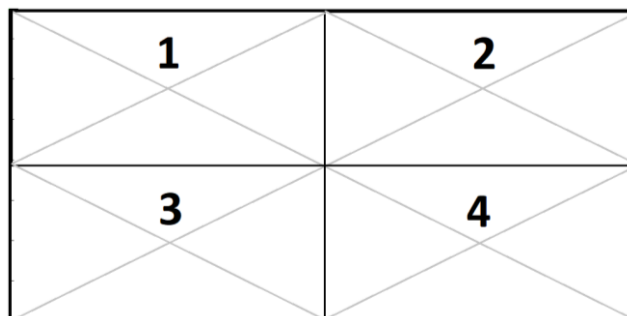


Figura 5-4: Cuadrantes de la pantalla

Se indica a los usuarios que observen el centro de los cuadrantes dos y tres, para poder estimar el punto en el que están observando (Figura 5-4).

Al igual que en las figuras anteriores, el círculo azul indica la proyección del ojo izquierdo mientras que el círculo rojo indica la proyección del ojo derecho. Ambos círculos corresponden al punto de observación del usuario al centro del cuadrante en el que se encuentran (Figura 5-5) (Figura 5-6) (Figura 5-7).

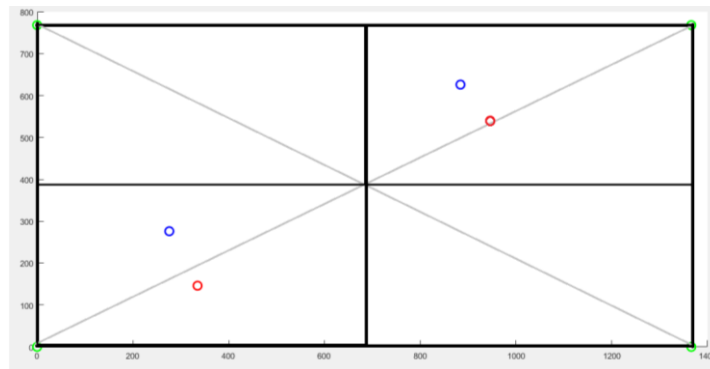


Figura 5-5: Punto de observación usuario 1 a los cuadrantes 2 y 3

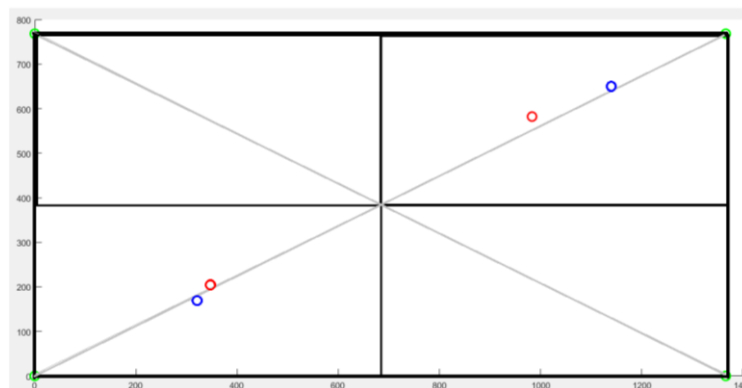


Figura 5-6: Punto de observación usuario 2 a los cuadrantes 2 y 3

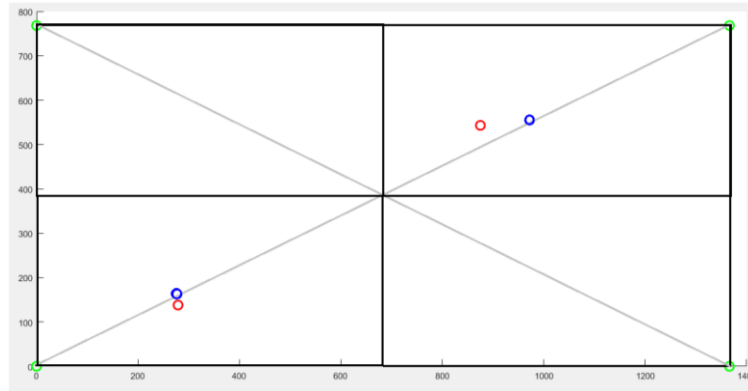


Figura 5-7: Punto de observación usuario 3 a los cuadrantes 2 y 3

Tras analizar el punto de observación de cada usuario dentro de estos cuadrantes se puede comprobar que el programa consigue estimar la proyección de las pupilas dentro de cada uno de esos cuadrantes. A su vez se puede extraer como no presenta una elevada precisión, por lo que estudiar los resultados desde un punto de vista cuantitativo no sería coherente hasta conseguir que el programa obtenga una mayor precisión.

6 Conclusiones y trabajo futuro

6.1 Conclusiones

En primer lugar, se han investigado algunos rastreadores oculares que hay en el mercado, centrando la atención en aquellos que puedan servir de utilidad para este trabajo. Los rastreadores estudiados más a fondo deben cumplir con unos requisitos: tener licencia de libre uso y actuar a partir de imágenes captadas por la cámara web del ordenador. Tras este estudio el trabajo se ha centrado en tres rastreadores: *OpenGazer*, *OpenEyes* y *PyGaze*, especificando la función de cada uno y los errores encontrados tras su ejecución. Al haber profundizado en estos tres rastreadores, y observado el alcance de cada uno, se ha determinado utilizar el rastreador *PyGaze* para el posterior análisis.

PyGaze es un programa que aporta la posición de la pupila dentro de la imagen del usuario que es captada mediante la cámara web del ordenador. Tras realizar un análisis inicial de los resultados de este rastreador, se ha comprobado que no son del todo precisos y que, además, solamente capta la pupila (siendo este el principal punto de interés para su después análisis) si es perfectamente diferenciada del iris, por lo que para ojos oscuros es incapaz de funcionar con éxito el programa.

Tras este pequeño bache, el trabajo ha sido reorientado captando la posición de las pupilas manualmente, junto con la posición del resto del ojo dentro de la pantalla del ordenador. Una vez extraídos los puntos de interés, independientemente si han sido obtenidos mediante un programa automáticamente o manualmente, es necesaria una calibración del sistema para calcular la distancia a la que se encuentra la pantalla del ordenador. Obtenidas todas las distancias y posiciones necesarias, se realiza la estimación del punto de observación a partir de un sistema de proyecciones de la pupila del usuario en la pantalla del ordenador. Para este sistema ha sido necesario depender de una serie de restricciones y hechos que han sido asumidos.

En último lugar, con toda la información recogida, se han realizado una serie de pruebas para estudiar los resultados mostrados por el programa.

6.2 Trabajo futuro

Como trabajo futuro se propone, en primer lugar, conseguir un sistema que capte de forma automática tanto la posición de cada una de las pupilas del individuo, como los extremos de cada uno de los ojos, indicando la posición exacta en la que se encuentre dentro de la imagen que capta la cámara web de la cara del usuario.

De igual manera, todas las restricciones que han sido asumidas se dejan como trabajo futuro. Tratar de estimar el punto de observación independientemente de la posición que se encuentre la cara con respecto de la pantalla del ordenador. Asimismo, tratar de resolver el problema que indica que todos los extremos de los ojos deben estar en una misma línea recta y paralela al centro de la pantalla, es decir, permitir que el usuario pueda girar la cabeza consiguiendo que la estimación sea más precisa. Para ello se propone realizar un

estudio en tres dimensiones en lugar de dos estudios de dos dimensiones. A su vez, se propone tratar los ojos como conjunto en lugar de analizar cada ojo por separado.

Otro trabajo futuro sería estimar la distorsión que produce la lente de la cámara web y el propio ojo.

Por último, quedaría conseguir el funcionamiento del programa en tiempo real.

Referencias

1. Qué es y para qué sirve el eye tracking [Internet]. [cited 2018 Jun 7]. Available from: <https://franciscotorreblanca.es/que-es-y-para-que-sirve-el-eye-tracking/>
2. Eye Tracking – Seguimiento ocular | Neuromarca [Internet]. [cited 2018 Jun 7]. Available from: <http://neuromarca.com/neuromarketing/eye-tracking/>
3. Vision-Based Interaction - Morgan & Claypool Books [Internet]. [cited 2018 Jun 7]. Available from: <https://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6812522&newsearch=true&queryText=history%20of%20gaze%20gestures>
4. Yao Z, Le Bars J-M, Charrier C, Rosenberger C. Literature review of fingerprint quality assessment and its evaluation. *IET Biom.* 2016 Sep 1;5(3):243–51.
5. Vertegaal R, Mamuji A, Sohn C, Cheng D. Media eyepliances: using eye tracking for remote control focus selection of appliances. In ACM Press; 2005 [cited 2018 Jun 7]. p. 1861. Available from: <http://portal.acm.org/citation.cfm?doi=1056808.1057041>
6. Garzás J. Rastreadores oculares para estudiar cómo trabajan los desarrolladores [Internet]. Javier Garzás. 2016 [cited 2018 Jun 7]. Available from: <http://www.javiergarzas.com/2016/06/14068.html>
7. Medioni G, Dickinson S. *Synthesis Lectures on Computer Vision.* :28.
8. George A, Routray A. Fast and accurate algorithm for eye localisation for gaze tracking in low-resolution images. *IET Comput Vis.* 2016 Oct 1;10(7):660–9.
9. Meng C, Zhao X. Webcam-Based Eye Movement Analysis Using CNN. *IEEE Access.* 2017;5:19581–7.
10. Drewes H, Schmidt A. Interacting with the Computer Using Gaze Gestures. In: Baranauskas C, Palanque P, Abascal J, Barbosa SDJ, editors. *Human-Computer Interaction – INTERACT 2007* [Internet]. Berlin, Heidelberg: Springer Berlin Heidelberg; 2007 [cited 2018 Jun 7]. p. 475–88. Available from: http://link.springer.com/10.1007/978-3-540-74800-7_43
11. Tobii Tech - What is eye tracking? [Internet]. [cited 2018 Jun 7]. Available from: <https://www.tobii.com/tech/technology/what-is-eye-tracking/>
12. Jensen OB. Webcam-Based Eye Tracking vs. an Eye Tracker [Pros & Cons] [Internet]. iMotions. 2017 [cited 2018 Jun 11]. Available from: <https://imotions.com/blog/webcam-eye-tracking-vs-an-eye-tracker/>
13. Farnsworth B, Ph.D. 10 Free Eye Tracking Software Programs [Pros and Cons] [Internet]. iMotions. 2017 [cited 2018 Jun 11]. Available from: <https://imotions.com/blog/free-eye-tracking-software/>

14. Inc S. Measure the world visually using eye tracking and emotion analysis. [Internet]. [cited 2018 Jun 7]. Available from: <https://www.sticky.ai>
15. EyeSee Research - The next step in eyetracking technology [Internet]. [cited 2018 Jun 11]. Available from: <http://eyesee-research.com/>
16. GazeRecorder – GazeRecorder [Internet]. [cited 2018 Jun 7]. Available from: <http://gazerecorder.christiaanboersma.com/gazerecorder/>
17. WebGazer.js: Democratizing Webcam Eye Tracking on the Browser [Internet]. [cited 2018 Jun 7]. Available from: <https://webgazer.cs.brown.edu/>
18. xLabs eye, gaze and head tracking via webcam [Internet]. [cited 2018 Jun 7]. Available from: <https://xlabsgaze.com/>
19. TurkerGaze [Internet]. [cited 2018 Jun 7]. Available from: <http://turkergaze.cs.princeton.edu/>
20. openEyes - Eye tracking for the masses [Internet]. [cited 2018 Jun 7]. Available from: <http://thirtysixthspan.com/openEyes/>
21. Opengazer: open-source gaze tracker for ordinary webcams [Internet]. [cited 2018 Jun 7]. Available from: <http://www.inference.org.uk/opengazer/>
22. Dalmaijer E. webcam-eyetracker: Webcam-based pupil-tracker, complete with GUI to perform calibrations [Internet]. 2018 [cited 2018 Jun 7]. Available from: <https://github.com/esdalmaijer/webcam-eyetracker>
23. Webcam Eye Tracker – PyGaze [Internet]. [cited 2018 Jun 7]. Available from: <http://www.pygaze.org/2015/06/webcam-eye-tracker/>

