

E.T.S. de Ingeniería Industrial,  
Informática y de Telecomunicación



# Comparativa de métodos de ajuste a elipse para detección de pupilas en un Evaluador de Visión Automatizado (EVA)

Máster en Ingeniería Biomédica  
Trabajo Fin de Máster

**Autor:** *Ainhoa Plazaola Ormazabal*

**Tutores:** *Arantxa Villanueva Larre*

*Rafael Cabeza Laguna*

*Pamplona, 27 de Abril de 2017*

**upna**  
Universidad  
Pública de Navarra  
Nafarroako  
Unibertsitate Publikoa



# RESUMEN

Este proyecto es un trabajo conjunto entre la UPNA y la empresa Davalor, la cual construye sistemas de medición de funciones visuales a través del seguimiento de la mirada basado en video-oculografía para clínicas ópticas. Para llevar a cabo dichos sistemas de medición es necesario una estimación muy precisa de la posición de la mirada del paciente. Esta estimación se realiza a través del contorno de la pupila. Dado que en las aplicaciones de visión artificial las imágenes circulares se proyectan como elipses, en el análisis de objetos circulares el primer paso a dar es el ajuste a elipse. Por lo tanto, el objetivo de este proyecto es probar diferentes algoritmos de ajuste a elipse referenciados en la literatura así como otros algoritmos para detectar outliers y omitirlos en el ajuste y así obtener una estimación muy precisa de la posición de la mirada.

Palabras clave: video-oculografía, ajuste a elipse, punto de mira

# ABSTRACT

This thesis is a collaborative work between UPNA and the Davalor Company. This company builds eye function measure systems through eye tracking based on video-oculography. To perform these measure systems it is essential to estimate with high accuracy the point of regard. Moreover, to make this possible it is necessary to estimate the contour of the pupil. Given that circular objects are projected on ellipses in images, so ellipse fitting is a first step for analysis of circular objects in computer vision applications. Therefore, the aim of this project is to compare different algorithms of ellipse fitting and outlier removal to obtain higher accuracy in terms of point of regard estimation.

Key words: video-oculography, ellipse-fitting, point of regard



# ÍNDICE DE CONTENIDO

<b>INTRODUCCIÓN.....</b>	<b>1</b>
1.1 ESTADO DEL ARTE.....	2
1.1.1 Visión general del seguimiento de la mirada.....	2
1.1.1.1 Clasificación de las técnicas de eye-tracking.....	3
1.1.1.2 Sistemas video-oculográficos (VOG).....	5
1.1.2 Detección precisa de la pupila.....	9
1.1.2.1 Método StarBurst.....	9
1.1.2.2 Métodos de ajuste a elipse.....	14
1.2 OBJETIVOS.....	14
1.3 ORGANIZACIÓN DEL TFM.....	15
<b>MARCO DE TRABAJO.....</b>	<b>17</b>
2.1 BASE DE DATOS.....	17
2.2 MÉTODO SUPER STARBURST.....	18
2.3 VICOMTECH.....	20
<b>ALGORITMOS PROPUESTOS.....</b>	<b>23</b>
3.1 AJUSTE ALGEBRAICO.....	24
3.1.1 Least Square.....	24
3.1.2 Iterative reweight.....	27
3.1.3 Renormalization.....	28
3.1.4 Taubin.....	28
3.1.5 Hyper-renormalization.....	29
3.1.6 HyperLS.....	30
3.2 AJUSTE GEOMÉTRICO.....	31
3.2.1 FNS.....	32
3.2.2 Geometric Distance Minimization.....	32
3.2.3 Hyperaccurate correction.....	34
3.3 DETECCIÓN DE OUTLIERS.....	35
3.3.1 Ellipse - RANSAC.....	35
3.3.2 Random ellipse-RANSAC.....	36
3.4 AJUSTE ESPECÍFICO A ELIPSE.....	37
3.4.1 Fitzgibbon.....	37
3.4.2 Random Sampling.....	38
3.5 EXPERIMENTOS DE AJUSTE A ELIPSE.....	39
3.5.1 Ejemplo de ajuste a elipse.....	39
3.5.2 Métodos específicos de ajuste a elipse.....	40

<b>RESULTADOS.....</b>	<b>41</b>
4.1 CARACTERÍSTICAS DE ELIPSES .....	41
4.2 COMPARACIÓN CENTROS DE PUPILA-REFLEJOS .....	43
4.2.1 Diferencia de centros respecto a Vicomtech.....	43
4.2.2 Diferencia de centros respecto a SuperStarBurst.....	45
4.2.3 Error entre centros de pupila y reflejo .....	47
4.2.4 Análisis del error sistemático.....	49
4.3 CÁLCULO DEL PoR .....	50
<b>CONCLUSIONES Y LÍNEAS FUTURAS .....</b>	<b>73</b>
5.1 CONCLUSIONES .....	73
5.2 LÍNEAS FUTURAS .....	74
<b>BIBLIOGRAFÍA.....</b>	<b>75</b>

# ÍNDICE DE FIGURAS

Figura 1.1: Evaluador de Visión Automatizado (EVA.) .....	1
Figura 1.2: Mapa de calor y representación de la ruta sacádica.....	2
Figura 1.3: Posición de electrodos y ubicación relativa del sujeto para el registro EOG.....	3
Figura 1.4: Eye tracker lente de contacto. ....	4
Figura 1.5: Relación entre el centro de la pupila y el reflejo corneal cuando el usuario se fija en diferentes puntos de la pantalla. ....	5
Figura 1.6: Imágenes de Purkinje. ....	6
Figura 1.7: Técnica de pupila oscura (izda) y técnica de pupila brillante (dcha).....	7
Figura 1.8: Sistema de eye-tracking montado en la cabeza.....	8
Figura 1.9: Esquema de dos de las tres posibles configuraciones. ....	8
Figura 1.10: (a) la imagen original. (b) la imagen con reducción de ruido.....	9
Figura 1.11: (a) imagen con reducción del ruido. (b) Imagen con el reflejo corneal eliminado después de la reducción de ruido. ....	10
Figura 1.12: (a) Candidatos para el borde la pupila. (b) Para cada candidato un grupo de rayos de vuelta (c). Rayos de vuelta de un candidato que no está en el contorno de la pupila. ....	11
Figura 1.13: (a-c) Punto inicial fuera del contorno de la pupila. ....	12
Figura 1.14: (a) Todos los candidatos y su posición media (círculo rojo). (b) Segunda iteración. (c) Convergencia.....	12
Figura 1.15: Resumen algoritmo de detección de puntos del borde.....	13
Figura 1.16: (a) Conjunto de puntos característicos con 2 outliers. (b) Ajuste least-square. (c) Inliers (verde) outliers (rojo) diferenciado por RANSAC. (d) con más outliers. (e) ajuste least-square con solo inliers.....	13
Figura 2.1: Significado del nombre de las imágenes de la base de datos.....	17
Figura 2.2: Imagen obtenida por EVA con 12 reflejos.....	18
Figura 2.3: Oclusión del párpado. ....	19
Figura 3.1: Ajustando la elipse a una secuencia de puntos con ruido [17].....	25
Figura 3.2: Ajuste a elipse minimizando la distancia geométrica [17]. ....	31
Figura 3.3: (a) 30 puntos en una elipse. (b), (c) ajuste a elipse para diferentes instancias de ruido de $\sigma = 0.5$ [17]. ....	39
Figura 3.4: Ejemplos de ajuste cuando hyper- renormalization devuelve hipérbola. Nivel de ruido relativo (a) 0.169 (b) 0.151 (c) 0.092 (d) 0.087 [17]. ....	40
Figura 4.1: Parámetros de la elipse. ....	41
Figura 4.2: Histograma acumulado diferencia centros pupila respecto a Vicomtech. ....	43
Figura 4.3: Histograma acumulado diferencia centros pupila respecto a Vicomtech de 0 a 4 pixeles.....	43
Figura 4.4: Histograma acumulado diferencia centro reflejo respecto a Vicomtech. ....	44
Figura 4.5: Histograma acumulado diferencia centro reflejo respecto a Vicomtech de 0 a 5 pixeles.....	44
Figura 4.6: Histograma acumulado diferencia centros pupila respecto a SuperStarBurst. ....	45
Figura 4.7: Histograma acumulado diferencia centros pupila respecto a SuperStarBurst de 0 a 4 pixeles.....	45
Figura 4.8: Histograma acumulado diferencia centro reflejo respecto a SuperStarBurst. ....	46
Figura 4.9: Histograma acumulado diferencia centro reflejo respecto a SuperStarBurst de 0 a 2 pixeles.....	46
Figura 4.10: Histograma acumulado error entre centros de pupila. ....	47

Figura 4.11: Ajuste mediante Ellipse-RANSAC (izda) y Random Ellipse RANSAC (dcha.).....	48
Figura 4.12: Histograma acumulado error entre centros de pupilas de 0 a 30. ....	48
Figura 4.13: Histograma acumulado error entre centros de reflejos. ....	48
Figura 4.14: Histograma acumulado error entre centros de reflejos de 0 a 3 pixeles. ....	49
Figura 4.15: Distintos patrones de calibración y validación.....	50
Figura 4.16: Posiciones del estímulo.....	51
Figura 4.17: PoR ojo izquierdo y ojo derecho Least Square.....	52
Figura 4.18: PoR ojo izquierdo y ojo derecho Iterative Reweight.....	53
Figura 4.19: PoR ojo izquierdo y ojo derecho Taubin. ....	53
Figura 4.20: PoR ojo izquierdo y ojo derecho Renormalziation.....	53
Figura 4.21: PoR ojo izquierdo y ojo derecho HyperLS. ....	54
Figura 4.22: PoR ojo izquierdo y ojo derecho Hyper –renormalization.....	54
Figura 4.23: PoR ojo izquierdo y ojo derecho FNS. ....	54
Figura 4.24: PoR ojo izquierdo y ojo derecho Hyper accurate correction. ....	55
Figura 4.25: PoR ojo izquierdo y ojo derecho Geometric Distance Minimization. ....	55
Figura 4.26: PoR ojo izquierdo y ojo derecho Fitzgibbon.....	55
Figura 4.27: PoR ojo izquierdo y ojo derecho Random Sampling. ....	56
Figura 4.28: PoR ojo izquierdo y ojo derecho Ellipse-RANSAC.....	56
Figura 4.29: PoR ojo izquierdo y ojo derecho SuperStarBurst. ....	56
Figura 4.30: PoR Random Ellipse-RANSAC con diferentes umbrales.....	57
Figura 4.31: PoR Taubin y Renormalization. ....	57
Figura 4.32: PoR Least Square e Iterative Reweight. ....	57
Figura 4.33: PoR HyperLS y Hyper-renormalization.....	58
Figura 4.34: PoR FNS y Hyperaccurate correction. ....	58
Figura 4.35: PoR Geometric Distance Minimization y Fitzgibbon.....	58
Figura 4.36: PoR Random Sampling y SuperStarBurst. ....	59
Figura 4.37: PoR Ellipse-RANSAC y Random Ellipse RANSAC. ....	59
Figura 4.38: Boxplot error PoR-Target ojo izquierdo. ....	63
Figura 4.39: Boxplot por fijaciones para los métodos Random Sampling y Fitzgibbon ojo izq. .	63
Figura 4.40: Boxplot error PoR-Target ojo derecho. ....	64
Figura 4.41: Distribución PoR Least Square. ....	64
Figura 4.42: Distribución PoR Iterative reweight. ....	65
Figura 4.43: Distribución PoR Taubin. ....	65
Figura 4.44: Distribución PoR Renormalization. ....	65
Figura 4.45: Distribución PoR HyperLS.....	66
Figura 4.46: Distribución PoR Hyper-renormalization. ....	66
Figura 4.47: Distribución PoR FNS.....	66
Figura 4.48: Distribución PoR Hyper accurate correction.....	67
Figura 4.49: Distribución PoR Geometric Distance Minimization.....	67
Figura 4.50: Distribución PoR Fitzgibbon. ....	67
Figura 4.51: Distribución PoR Random Sampling.....	68
Figura 4.52: Distribución PoR Ellipse RANSAC.....	68
Figura 4.53: Distribución PoR Random Ellipse-RANSAC.....	68
Figura 4.54: Distribución PoR SuperStarBurst.....	69
Figura 4.55: Boxplot de todos los métodos para el ojo izquierdo. ....	70
Figura 4.56: Boxplot de todos los métodos para el ojo derecho. ....	71







# Capítulo 1

## Introducción

Este trabajo fin de máster (TFM) se ha realizado en la Universidad Pública de Navarra (UPNA) con la colaboración de la empresa Davalor. En los últimos años, dicha empresa ha desarrollado un Evaluador de la Visión Automatizada (EVA), una máquina con una tecnología basada en los videojuegos interactivos en 3D que monitoriza la interacción entre videojuegos y paciente obteniendo de ellos los distintos parámetros de la función visual que permite un diagnóstico preciso a los problemas de visión. Dentro de los más de 75 parámetros que evalúa se encuentran las áreas de agudeza, el error refractivo, la motilidad ocular o la visión binocular entre otros. La mayor ventaja o característica de esta máquina es el tiempo de adquisición, ya que en 5 minutos calcula dichos parámetros. En cambio una exploración clásica necesita 60 minutos. Tras la sesión, ofrecen recomendaciones ajustadas a las necesidades de cada paciente, entre los que se encuentran tratamientos para antes y después de una cirugía ocular, entrenamientos de la función visual, prescripción de gafas, hábitos de higiene visual, etc.



*Figura 1.1: Evaluador de Visión Automatizado (EVA.)*

Los sistemas de medición de funciones visuales se realizan a través del seguimiento de la mirada basado en la video-oculografía. Para su adecuado funcionamiento es necesario realizar una estimación muy precisa de la posición de la mirada del paciente. Para ello es necesario detectar con mucha exactitud el contorno de la pupila y esto lleva a analizar distintos algoritmos referenciados en la literatura y a elegir el más preciso.

## 1.1 ESTADO DEL ARTE

### 1.1.1 Visión general del seguimiento de la mirada

Entre los años 60 - 70 empezaron a desarrollarse los sistemas de seguimiento de la mirada o “eye-tracking” con el objetivo de utilizar el movimiento del ojo como instrumento de comunicación y así mejorar el día a día de las personas con movilidad muy limitada, como podían ser pacientes que hubieran padecido infartos cerebrales, accidentes medulares o enfermedades degenerativas que causarían la inmovilidad [1].

El dispositivo de medida que generalmente se usa para el movimiento de ojo se conoce como “eye tracker” o seguidor de movimiento ocular. Esta técnica tiene una amplia variedad de aplicaciones como puede ser dentro del campo del marketing, la psicología o la neurología que nacen de las investigaciones realizadas dentro del estudio de la visión. Generalmente, los diferentes usos del eye tracking se pueden dividir en dos tipos: diagnósticas e interactivas.

El primero de los casos, el de análisis y diagnóstico tiene un papel pasivo, es decir, el seguidor de movimiento ocular (eye tracker) es el que proporciona la evidencia objetiva y cuantitativa de los procesos relacionados con la visión y la percepción. Esta técnica es muy utilizada en el ámbito del marketing o diseños de páginas web, ya que mediante mapas de calor realzan las zonas más observadas y mediante la representación de la ruta sacádica obtienen el orden en el que las zonas de la página han ido llamando la atención [2] [3].

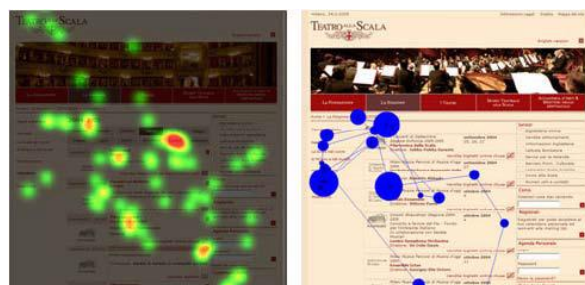


Figura 1.2: Mapa de calor y representación de la ruta sacádica.

Los interactivos, tienen un papel activo con lo que permite al usuario controlar la interfaz mediante el movimiento de los ojos [4]. Por lo tanto, se espera que la interfaz interactúe, responda o reaccione a la mirada del usuario. Esta técnica es especialmente beneficiosa en situaciones donde es necesaria la utilización de manos y el usuario tenga discapacidad motriz como en el caso de la esclerosis múltiple. Además, a la hora de seleccionar objetos, la mirada puede resultar más rápida que la selección con el ratón.

### 1.1.1.1 Clasificación de las técnicas de eye-tracking

Los sistemas de eye-tracking dependiendo de la técnica de medición de movimientos oculares que utilicen se dividen en tres categorías: electro-oculograma (EOG), lentes de contacto/bobinas de búsqueda, foto-oculografía (POG) o video-oculografía (VOG).

Una de los principales requisitos o mejoras de los sistemas de seguimiento de mirada es la comodidad del usuario con lo que es esencial el carácter no invasivo de los sistemas. Dicho esto, las técnicas de eye-tracking se podrían agrupar en sistemas invasivos y no invasivos.

#### (a) Sistemas invasivos

Entre los sistemas invasivos se encuentran los sistemas de electro-oculografía (EOG) y las lentes de contacto/ bobinas de búsqueda.

- **Electro - oculograma (EOG)**

La electro-oculografía, fue la técnica de registro de movimiento de ojos más utilizada hace 40 años (y todavía está en uso). Consiste en medir la diferencia del potencial eléctrico obtenido por los electrodos colocados alrededor de los ojos [5]. Esta técnica mide los movimientos del ojo respecto a la posición de la cabeza, con lo que no es adecuado para la medición de la posición de la mirada excepto si la posición de la cabeza es previamente medida (usando head tracker).



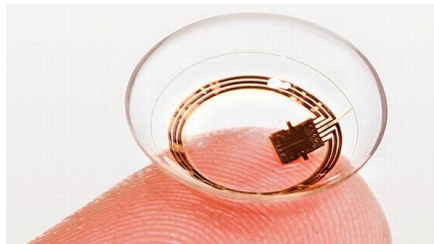
Figura 1.3: Posición de electrodos y ubicación relativa del sujeto para el registro EOG.

El movimiento de ojo es el que produce la diferencia de potencial pero no suelen ser constantes y sus variaciones complican el uso de EOG a la hora de medir movimientos oculares lentos y detectar la dirección de la mirada. Sin embargo, para los movimientos oculares sacádicos asociados con cambios de mirada y detección de parpadeos es una técnica muy sólida. Añadir que esta técnica no necesita una potencia de cálculo muy alta e incluso se utiliza en ausencia de iluminación.

- **Sistemas basados en lentes de contacto:**

Es una de las técnicas más precisas para la medición del movimiento del ojo pero implica el contacto físico con el ojo a través de las lentes de contacto, con lo que resulta muy incómodo para los usuarios.

Para llevar la función del seguimiento de la mirada, necesita un objeto de referencia mecánico u óptico montado en la lente y es puesta directamente en el ojo. Este método trabaja con bobinas de búsqueda y el movimiento se mide a través de un campo electromagnético. Aunque la bobina de búsqueda es el método más preciso, con una precisión de 5-10 arco segundos en un rango de 5° es el método más invasivo [6].



*Figura 1.4: Eye tracker lente de contacto.*

(b) **Sistemas no invasivos**

Dentro de los sistemas no invasivos se encuentran los que se basan en técnicas de video o foto-oculografía. El hecho de que sean no invasivos hace que sean los más utilizados. En este apartado se introducirán los sistemas de video-oculografía. Además, al ser el sistema utilizado para llevar a cabo este trabajo, en el siguiente apartado se desarrolla con más detalle.

- **Foto-oculografía (POG) y video-oculografía (VOG)**

La técnica de video y foto-oculografía hacen uso de cámaras o dispositivos de adquisición de imágenes, ya que gracias a estas imágenes determinan el movimiento del ojo [7]. Para ello, se analiza el comportamiento de las principales características del ojo humano como son el centro

de la pupila, el contorno de la pupila o reflexiones de la córnea y se intenta establecer una relación entre estas y el punto donde se mira (PoR).

Para llevar a cabo la captura de imágenes se requiere una iluminación adecuada del usuario. A pesar de que algunos de estos sistemas cuenten con una o varias fuentes de luz otros solamente utilizan luz visible.

En el caso de que el sistema use iluminadores, la luz emitida suele ser infrarroja. Esta se refleja en los ojos y se captura por la cámara de vídeo. La información capturada es analizada y finalmente se obtiene la rotación de los ojos en función de los cambios de posición de las reflexiones.

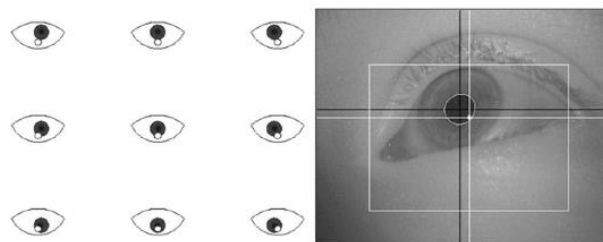
Estos sistemas de seguimiento de la mirada podrían dividirse en función de su objetivo:

- Técnicas para medir los movimientos del ojo
- Técnicas para determinar el punto o posición donde se fija la mirada, o cálculo del *PoR*.

La diferencia de estos dos tipos es que la primera técnica determina la rotación del ojo, es decir, la orientación del ojo respecto a la cabeza y en la segunda técnica se calcula la posición donde está mirando el sujeto. Para ello, es necesario el seguimiento independiente de la posición de la cabeza y la rotación del ojo.

### 1.1.1.2 Sistemas video-oculográficos (VOG)

Como ya se ha comentado los sistemas de eye-tracking basados en video-oculografía están compuestos por las fuentes de iluminación y el sistema de adquisición de imágenes. A pesar de que en los últimos años se han desarrollado diferentes algoritmos para el uso de sistemas de video-oculografía, todos ellos tienen la misma base. La mejora de estos algoritmos hace que el sistema admita variaciones en algunos de los aspectos como puede ser el número de cámaras, el tipo de iluminación, el número de iluminadores, características del ojo a detectar (pupila, iris, esclerótica, reflejos especulares), accesorios...



*Figura 1.5: Relación entre el centro de la pupila y el reflejo corneal cuando el usuario se fija en diferentes puntos de la pantalla.*

En este sistema de seguimiento de mirada el tipo de iluminación puede ser de dos tipos: luz visible o luz infrarroja. Las imágenes obtenidas con luz visible tienden a que el centro de la pupila sea imprecisa y sensible al movimiento de la cabeza. Por ello se suele añadir un punto de referencia llamado reflejo corneal o “*glint*”.

Existen tres métodos de sistemas VOG para obtener información del ojo y así calcular el punto donde fija la mirada el usuario.

#### (a) Imágenes de Purkinje

Es un método que se basa en las reflexiones oculares producidas por una fuente de luz infrarroja. Existen cuatro imágenes Purkinje:

- La primera imagen Purkinje es la de mayor tamaño y formada en la superficie de la córnea, es la más clara y brillante de las cuatro.
- La segunda se refleja en la cara interna de la córnea. Al no haber distancia con la primera, generalmente suele estar superpuesta.
- La tercera se produce en la cara anterior del cristalino.
- La cuarta es la más pequeña de las imágenes, brillante y se forma en la cara posterior del cristalino, con lo que es una imagen invertida.

El método algunas veces se basa en la diferencia de posiciones entre la primera y cuarta imagen Purkinje (generada en la superficie cóncava del cristalino). Otras solo tienen en cuenta la primera imagen Purkinje y otra característica ocular para el cálculo de la posición de la mirada [8].

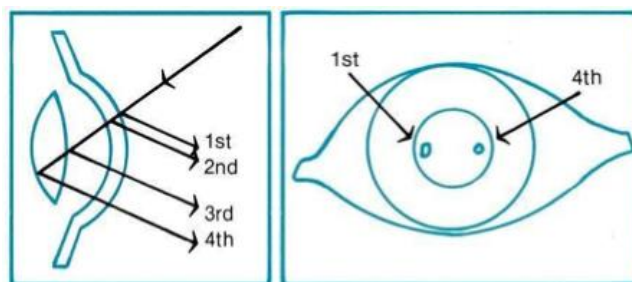


Figura 1.6: Imágenes de Purkinje.

#### (b) Pupila brillante y pupila oscura

El seguimiento de mirada mediante la fuente de luz infrarroja (IR) permite dos tipos de técnicas para la detección de pupila: técnica de pupila brillante y de pupila oscura. Aunque también puede utilizarse la combinación de las dos técnicas para obtener una de mayor precisión [9].



La técnica de pupila brillante requiere una fuente IR cerca o en el eje de la cámara con lo que la iluminación aumenta tanto que la pupila queda claramente delimitada como una zona brillante debido a la naturaleza de la parte reflectiva. En cambio, la técnica de pupila oscura ilumina el ojo con una fuente IR desde fuera del eje de la cámara siendo la pupila la parte más oscura de la imagen. Para detectar la dirección de la mirada se suele hacer uso del vector del centro de la pupila al reflejo de la córnea.



Figura 1.7: Técnica de pupila oscura (izda) y técnica de pupila brillante (dcha).

La detección de pupila tiene varias ventajas respecto al borde entre la esclerótica (membrana de la parte externa del globo ocular) y el iris: el contorno de la pupila es pequeño, definido y más difícil de ocluir por los párpados.

#### (c) Vector PCCR

El vector entre la primera imagen de Purkinje y el centro de la pupila, es conocido como PCCR (Pupil Center Cornela Reflection), en lugar del centro de la pupila como única indicación para detectar la dirección de la mirada. Dentro de las ventajas de esta técnica es que las características son fáciles de extraer y que el vector es más robusto respecto a los movimientos de la cabeza. Es un método muy utilizado siendo ejemplo las siguientes publicaciones [10] [11] [12].

#### (d) Configuraciones posibles

Respecto al diseño de los sistemas video-oculográficos se pueden agrupar en dos grupos: eye tracking montados en la cabeza y sistemas remotos. El primer grupo suele ser preferible en el caso en el que las aplicaciones requieran grandes y rápidos movimientos de cabeza por lo que no es aconsejable para los casos que requieren una continua vigilancia o seguimiento durante largos periodos de tiempo (por ejemplo, el control del cursor para las personas con discapacidad motriz, o niños).



Figura 1.8: Sistema de eye-tracking montado en la cabeza.

Los últimos años, el grado de invasión de los seguidores de mirada se ha reducido considerablemente gracias a la miniaturización del sistema. Como es de esperar los sistemas remotos reducen el grado de invasión pero pierden precisión.

Entre los sistemas remotos existen tres tipos de configuraciones. El primero, la configuración más sencilla, consta de una cámara y una fuente de luz. No es suficiente para calcular la dirección de la mirada con alta precisión y por lo tanto hay que mantener la cabeza fija respecto al sistema y estimar la distancia entre ojo y cámara. Por lo general, suelen calcular la dirección de la mirada con el vector PCCR o disponer de medios para medir la posición del ojo con ayuda de espejos.

La segunda configuración permite la estimación del PoR a partir del centro de la pupila y reflejos. Consiste en una cámara y dos iluminadores uno a cada lado de la pantalla (ver fig. 1.9a). Esta configuración no implica ninguna restricción en cuanto a movimientos de cabeza y no requiere la utilización de ningún dispositivo adicional para la estimación del punto donde fija la mirada.

El tercer y último sistema consiste en más de una cámara y en más de un iluminador fuera del eje de la cámara (ver fig.1.9b), con lo que la detección de la pupila se realiza a través de la técnica de pupila oscura. A este sistema se le denomina multi-cámara y cada una de ellas puede estimar la mirada de forma independiente con lo que permite grandes movimientos de cabeza.

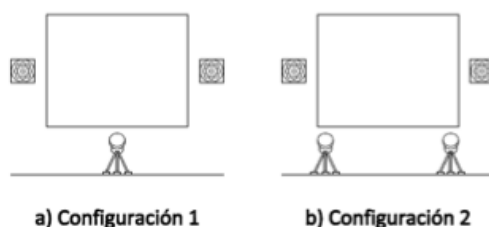


Figura 1.9: Esquema de dos de las tres posibles configuraciones.

## 1.1.2 Detección precisa de la pupila

Para llevar a cabo la detección precisa del contorno de la pupila es necesario obtener los puntos que se sitúan en el borde de la pupila. Para ello una de las técnicas más conocidas es el método Starburst que se explica en el apartado 1.1.2.1. Una vez obtenidos los puntos que pueden corresponder al contorno de la pupila, una de las maneras más precisas para detectar el centro de la pupila consiste en calcular la elipse formada por el contorno. Por ello, en el apartado 1.1.2.2. se nombran los distintos algoritmos de ajuste a elipse referenciados en la literatura.

### 1.1.2.1 Método StarBurst

Este método [13] es implementado para la técnica de pupilas oscuras. Sin embargo, se puede aplicar a pupilas brillantes realizando pequeños cambios. El método Starburst consiste en el siguiente proceso:

- Detectar, localizar, eliminar el reflejo corneal
- Detección iterativa de los puntos del borde
- Realizar el ajuste a elipse y aplicar RANSAC

El objetivo de este algoritmo es extraer el centro de la pupila y del reflejo corneal para relacionar el vector diferencia de estos dos centros a las coordenadas en la imagen. El algoritmo empieza localizando y eliminando el reflejo corneal de la imagen. Después, los puntos del borde de la pupila son localizados usando una técnica iterativa basada en características.

#### (a) Reducción del ruido

En primer lugar realiza la reducción de ruido ya que en las imágenes aparecen dos tipos de ruido: ruido disparo y ruido de línea. El ruido disparo se elimina aplicando un filtro Gaussiano 5x5 con una desviación estándar de 2 píxeles. El ruido de línea en cambio, es un ruido sistemático y se aplica un factor de normalización línea a línea para cambiar la intensidad media de la línea a la media actual la cual se obtiene de los *frames* anteriores. El efecto de la reducción de ruido se puede apreciar en la Figura 1.10:

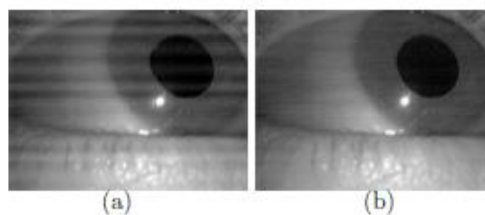
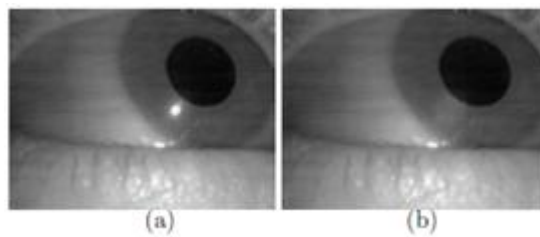


Figura 1.10: (a) la imagen original. (b) la imagen con reducción de ruido.

**(b) Detección, localización, eliminación del reflejo corneal**

Después, para detectar el reflejo corneal se sabe que el reflejo corresponde a una de las regiones más brillantes de la imagen con lo que el reflejo se puede obtener mediante una umbralización. Sin embargo, un umbral fijo no es lo óptimo con lo que hace uso de una técnica adaptativa de umbralización en cada fotograma para localizar el reflejo. Para empezar, se utiliza el umbral máximo para crear una imagen binaria en la cual los únicos valores que estén por encima de ese umbral se elijan como candidatos a reflejos corneales. Es esperable que la región candidata más amplia se atribuya al reflejo corneal ya que cualquier reflejo parásito tiende a ser más pequeño y situado fuera de la córnea. La proporción entre la región candidata más amplia y el área medio de otras regiones se calcula disminuyendo el valor del umbral. Al principio, la proporción puede aumentar ya que el reflejo corneal crece en tamaño con mayor rapidez que otras áreas. Apuntar que la intensidad del reflejo corneal disminuye monótonamente hacia los bordes, explicando el crecimiento. Un umbral más bajo puede inducir a mayor número de falsos candidatos. La proporción puede empezar a bajar cuando los falsos candidatos se vuelven más prominentes y la región del reflejo corneal aumenta. Finalmente, se elige el umbral que genere la proporción más alta y óptima. La localización del reflejo corneal se obtiene por el centro geométrico  $(x_c, y_c)$  de la región más amplia de la imagen usando una umbralización adaptativa determinada.

Para eliminar el reflejo corneal se hace uso de la interpolación radial. Primero, el pixel central de la región del reflejo corneal identificado se ajusta a la media de las intensidades del contorno de la región. Después, para cada pixel entre el centro y el contorno, la intensidad del pixel se determina a través de una interpolación lineal. Un ejemplo de este proceso se puede ver en la siguiente figura.



*Figura 1.11: (a) imagen con reducción del ruido. (b) Imagen con el reflejo corneal eliminado después de la reducción de ruido.*

**(c) Detección del contorno de la pupila**

No todos los puntos del borde de la pupila son necesarios para estimar el contorno por ello este método detecta bordes a lo largo de un número limitado de rayos que se extienden desde el centro estimado de la pupila. Los rayos se pueden ver en la Figura 1.12a. Para mejorar la robustez

respecto de la inexactitud del punto de partida, los bordes también se detectan a lo largo de un número limitado de rayos que se extienden desde el conjunto inicial de puntos detectados hasta el punto de partida. Estos rayos de retorno se pueden ver en la Figura 1.12 a y c.

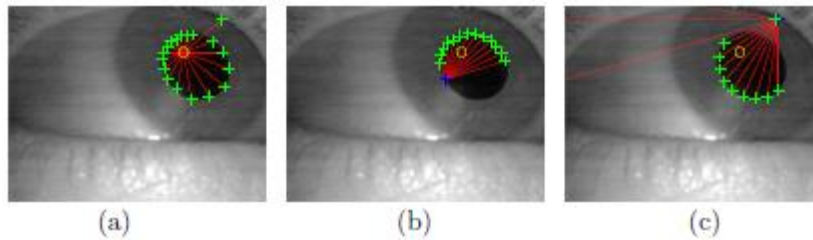


Figura 1.12: (a) Candidatos para el borde la pupila. (b) Para cada candidato un grupo de rayos de vuelta (c). Rayos de vuelta de un candidato que no está en el contorno de la pupila.

Para cada fotograma se elige un punto que represente la estimación del centro de la pupila. En el primer *frame* esto se puede determinar manualmente o eligiendo el centro de la imagen. En los siguientes *frames*, se utiliza la localización del centro de la pupila del fotograma anterior. A continuación, las derivadas ( $\Delta$ ) a lo largo de los  $N$  rayos, que se extienden radialmente alejándose de este punto de partida, se evalúan independientemente pixel a pixel hasta que se supera un umbral ( $\phi$ ). Dado que se está utilizando la técnica de la pupila oscura, sólo se consideran las derivadas positivas (intensidad creciente a medida que el rayo se extiende). Cuando se excede del umbral se define en ese punto un punto característico y se detiene el proceso de búsqueda en ese rayo. Si el rayo se extiende hasta el borde de la imagen no se define el punto característico.

Para cada uno de los puntos candidatos a borde, se repite el proceso descrito anteriormente. Sin embargo, los rayos están limitados a  $\gamma = \pm 50$  grados alrededor del rayo que originalmente generó el punto característico. La motivación para limitar los rayos de retorno de esta manera es que, si el punto candidato de contorno está realmente en el contorno de la pupila, los rayos que retornan generarán puntos característicos adicionales en el lado opuesto de la pupila de tal manera que todos son consistentes con una única elipse. Sin embargo, si el punto candidato no se encuentra en la pupila, este proceso generará puntos candidatos de contorno que no serán consistentes con una elipse. Por lo tanto, este procedimiento tiende a aumentar la proporción de cantidad de puntos característicos en el contorno de la pupila sobre el número de puntos que no están en el contorno de la pupila. Además, debido a que los puntos característicos definidos con una derivada alta tienden a ser puntos situados en el contorno de la pupila, el número de rayos de retorno es variable y se ajusta a  $5\Delta/\phi$ . El número mínimo de rayos es 5, puesto que por definición un punto característico se determina por  $\Delta \geq \phi$ .

El proceso de dos fases para detección de características mejora la robustez del método en los casos en que la estimación del punto inicial no sea la adecuada. Esto se puede apreciar en la Figura 1.13a ya que el conjunto inicial de rayos solamente detecta tres puntos del contorno de la pupila y los rayos de retorno de estos tres puntos detectan muchos más que pertenecen al contorno (Figura 1.13b).

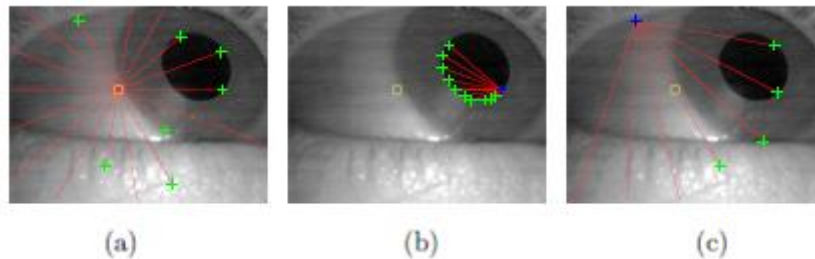


Figura 1.13: (a-c) Punto inicial fuera del contorno de la pupila.

Todos los puntos que correspondan al conjunto de puntos característicos se representan en la Figura 1.14a y el número de puntos que se encuentran en el contorno es mayor a los que se encuentran fuera. El círculo rojo corresponde al punto medio de los puntos característicos el cual va a ser el punto inicial de la segunda iteración. En este punto se podría realizar un ajuste a elipse sin embargo, los puntos que se encuentran fuera del contorno producirán un error significativo al ajuste. Para eliminar esos puntos se realiza el proceso iterativo de dos fases explicado arriba. La Figura 1.14b corresponde a los puntos candidatos detectados en la segunda iteración. En la Figura 1.14c se puede ver como la localización central converge rápidamente en el centro de la pupila. Las iteraciones se detienen cuando el centro de los puntos característicos detectados varía menos de 10 píxeles. Finalmente saber que cuando la estimación del punto inicial es buena solo se requiere de una iteración. En cambio, cuando la estimación inicial no es buena se realizan unas pocas iteraciones.

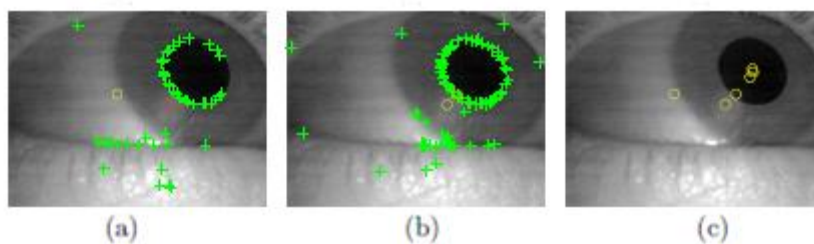


Figura 1.14: (a) Todos los candidatos y su posición media (círculo rojo). (b) Segunda iteración. (c) Convergencia.

El proceso de dos fases de detección de características mejora la robustez del método en el caso que los puntos iniciales son mal elegidos. El resumen del proceso sería el siguiente:

**Entrada:** Imagen con el reflejo corneal eliminado  
 Mejor estimación del centro de la pupila  
**Salida:** Conjunto de puntos característicos  
**Proceso:**  
 Iteración  
 1ª Fase:  
 Seguir los rayos que se extienden desde el punto de partida  
 Calcular la derivada de la intensidad en cada punto  
 Si derivada > umbral entonces  
     Marcar punto característico del borde  
     Detener el proceso a lo largo del rayo  
 2ª Fase:  
 Para cada punto característico detectado en la Fase 1  
     Seguir la marcha de retroceso del rayo hasta el punto de inicio  
     Calcular la derivada de la intensidad en cada punto  
     Si derivada > umbral entonces  
         Marcar punto característico del borde  
         Detener proceso a lo largo del rayo  
 El punto de inicio = centro geométrico de los puntos característicos  
 Hasta que el punto de inicio converja

Figura 1.15: Resumen algoritmo de detección de puntos del borde.

(d) Ajuste a elipse

Dado un conjunto de puntos característicos, el siguiente paso es encontrar el algoritmo que mejor se ajuste al contorno de la pupila. Para ello se hace uso del método Least Square (explicado en el apartado 3.1.1) y se le aplica Random Sample Consensus (RANSAC) al modelo ajustado. RANSAC es una técnica muy efectiva en el caso de que haya puntos que no correspondan al contorno, ya que los omite en el ajuste.

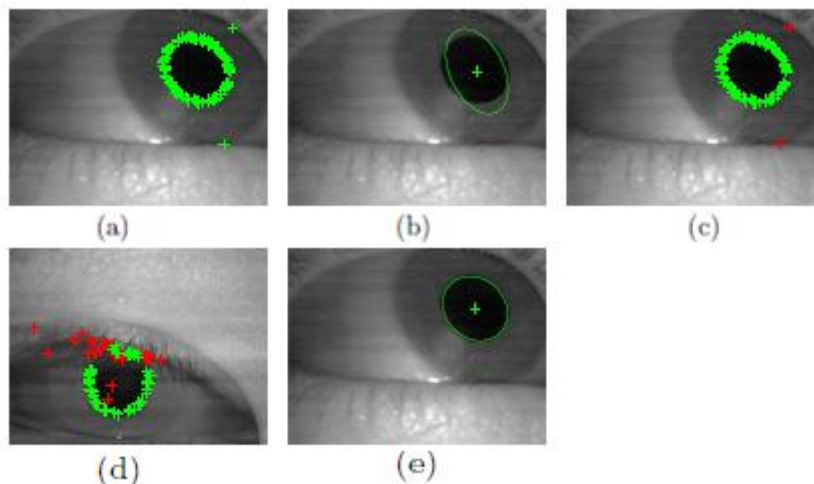


Figura 1.16: (a) Conjunto de puntos característicos con 2 outliers. (b) Ajuste least-square. (c) Inliers (verde) outliers (rojo) diferenciado por RANSAC. (d) con más outliers. (e) ajuste least-square con solo inliers.

### 1.1.2.2 Métodos de ajuste a elipse

En imágenes, los objetos circulares, los cuales son muy comunes en las escenas diarias, se proyectan como elipses, por ello el primer paso para el análisis de objetos circulares en 3D, en aplicaciones de visión artificial, es el ajuste a elipse [14]. Por esta razón, el estudio de ajuste a elipse comienza tan pronto como el uso de los ordenadores para el análisis de imágenes, es decir, en 1970. En 1990, los investigadores introdujeron la idea de computación óptima basada en las propiedades estadísticas del ruido.

Después, se dividieron dependiendo del enfoque que se le diera. El enfoque algebraico incluye el método Least Square, Iterative reweight (1982), Taubin (1991), Renormalization (1993), HyperLS (2009), y Hyper-renormalization (2012). El geométrico en cambio, incluye FNS (2000), Hyperaccurate correction (2006) y Geometric Distance Minimization (2012). Además de estas técnicas también se encuentran los métodos específicos de Fitzgibbon et al (1999) y la técnica de Random Sampling (2014) para evitar hipérbolas, que suele darse en los casos en que la información de entrada es insuficiente. Para terminar se encuentra el método Ellipse-RANSAC que es un método robusto puesto que consiste en detectar outliers, en este caso se trataría de los puntos que se encuentran fuera del contorno de la pupila y así omitirlos en el ajuste.

## 1.2 OBJETIVOS

Para llevar a cabo las funciones visuales de los sistemas video-oculográficos, este trabajo tiene como objetivo probar diferentes algoritmos de ajuste a elipse aplicados a las detecciones de pupila nombradas y referenciadas en la literatura y elegir el más preciso.

Para llevar a cabo el objetivo, se realizará un post procesado estadístico y se calculará la posición de la mirada (PoR) a través del vector que se crea entre el centro de la pupila y el centro de los reflejos.



## 1.3 ORGANIZACIÓN DEL TFM

Además del Capítulo 1 o Introducción, en el cual se ha puesto al lector en contexto, el TFM consta de 5 capítulos más que a continuación se describen brevemente:

### **Capítulo 2:** Marco de trabajo

Este capítulo contiene la explicación del algoritmo *SuperStarburst*, ya que es el algoritmo empleado por la empresa Davalor. Además, se explicará cómo están guardados los datos. Por otro lado, se hablará de los datos de la empresa Vicomtech puesto que es la empresa que creó el eye-tracker para EVA y con quien han comparado los resultados obtenidos.

### **Capítulo 3:** Algoritmos propuestos

Procedimiento de los diferentes algoritmos referenciados en la literatura y sus mayores diferencias a la hora de hacer el ajuste a elipse para detectar el contorno de la pupila. Estos algoritmos se dividen en cuatro tipos: los que se basan en el ajuste algebraico, en el ajuste geométrico, en la detección de outliers y los que siempre ajustan una elipse.

### **Capítulo 4:** Resultados

Primero se calcula las diferencias entre los centros obtenidos por los algoritmos propuestos y los del *SuperStarBurst* y *Vicomtech* y se representan mediante histogramas acumulados. Después, se obtienen las imágenes problemáticas y se les marca el centro para obtener el error o precisión del método de ajuste. Finalmente, la bondad de los métodos se calcula mediante la posición de la mirada (PoR).

### **Capítulo 5:** Conclusión y líneas futuras

En este apartado se explican las conclusiones obtenidas analizando los resultados y los puntos a mejorar en siguientes trabajos para obtener mayores diferencias entre métodos de ajuste.



# Capítulo 2

## Marco de trabajo

### 2.1 BASE DE DATOS

Las imágenes para llevar a cabo este proyecto se han obtenido mediante el Evaluador de Visión Automatizado (EVA) de la empresa Davalor. La base de datos consta de 7 usuarios, 4 de ellos hombres y 3 mujeres con edades de 20 a 30 años. Dentro de la carpeta de cada usuario hay imágenes obtenidas durante la calibración y la validación. Además, las imágenes del ojo derecho (RE) y del ojo izquierdo (LE) están separadas en sus respectivas carpetas. Estas imágenes tienen una resolución de 640 x 480 píxeles.

El nombre de las imágenes tanto de la calibración como de la validación tiene el siguiente significado:

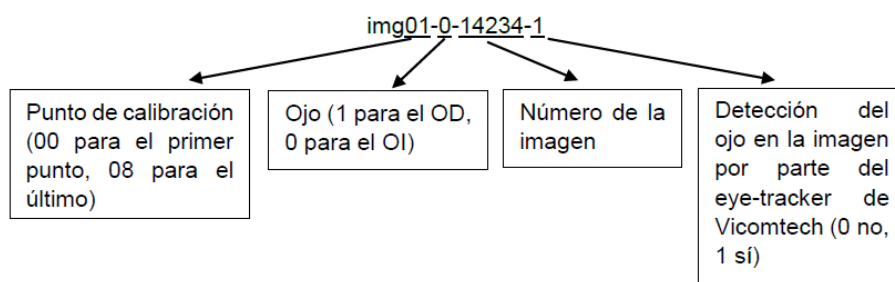


Figura 2.1: Significado del nombre de las imágenes de la base de datos.

El último número es la detección del eye-tracker de Vicomtech, ya que la empresa ha comparado el método utilizado por ellos *StarBurst* con el cual obtienen los centros de la pupila, el de los reflejos y la precisión de detección, con el algoritmo utilizado en el eye-tracker de Vicomtech.

## 2.2 MÉTODO SUPER STARBURST

La empresa Davalor para obtener el centro de la pupila y de los reflejos ha hecho uso del algoritmo *SuperStarburst* que es una extensión del método *Starburst*, el cual se ha explicado en la sección [1.1.2.1](#), para adaptarlo a las características de las imágenes con múltiples reflejos corneales y superar algunas de las limitaciones del algoritmo original, ya que el algoritmo *Starburst* solo cuenta con un reflejo.

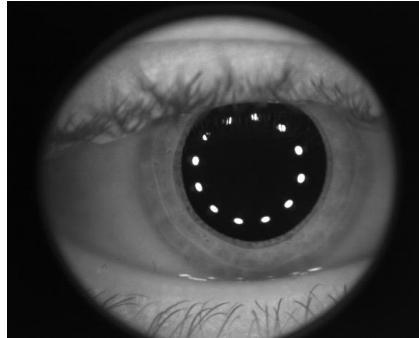


Figura 2.2: Imagen obtenida por EVA con 12 reflejos.

El primer paso consiste en localizar los reflejos corneales o *glints*. El método utilizado en esta sección se basa en el mismo principio que el algoritmo *Starburst* original: binarización de la imagen con un umbral adaptativo que maximice el ratio entre el área de la zona blanca más grande y el área de las otras zonas brillantes. Siguiendo este mismo criterio, el algoritmo se modificó ligeramente para adaptarlo a las imágenes con 12 *glints*. Por otro lado, las zonas identificadas como *glints* deben cumplir unos requerimientos de área y de forma, es decir, deben ser de un tamaño específico y deben tener una forma elíptica con una determinada elongación. En el caso que se identifiquen menos de 5 *glints*, la imagen no se procesa más, ya que no se podrá ajustar una elipse y se pasa al siguiente *glint*.

Una vez localizados los reflejos, se ajusta una elipse a sus centroides con un método directo de mínimos cuadrados [15]. La elipse resultante también tiene que cumplir unos requisitos en términos de radio y elongación. En caso que no cumpla, se vuelve a ajustar una elipse con el método *Ellipse-RANSAC* (Random Sample Consensus) [16], el cual se explica en el apartado [3.3.1](#), para poder detectar la presencia de puntos erróneos (*outliers*). En el caso que la elipse siga sin cumplir los requisitos de dimensiones y forma significa que hay reflejos parásitos que se confunden con los verdaderos reflejos, por lo que no se procesa más la imagen y se pasa al siguiente *frame*.

Mientras que el algoritmo *Starburst* original elimina el reflejo corneal interpolando los niveles de gris, en este caso se diseña una máscara binaria evitando que los *glints* interfieran en el proceso de detección de pupila.

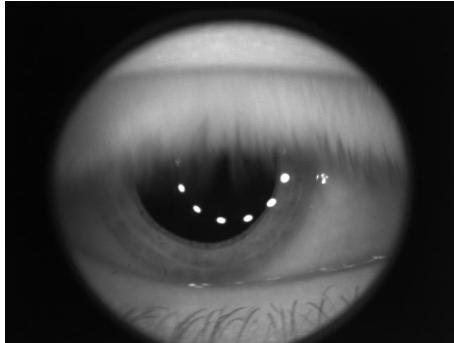


Figura 2.3: Oclusión del párpado.

El proceso de detección del contorno de la pupila consiste en trazar un número determinado de rayos desde un centro pre-estimado de la pupila. Como estimación se utiliza la posición del centro de la pupila del *frame* anterior. En caso de no disponer los datos del *frame* anterior, se utiliza el centro de la elipse ajustada previamente a los *glints*. Se trazan 70 rayos con una largura de 130 píxeles. Si en la imagen se han detectado los 12 *glints*, se trazan los rayos desde el centro pre-estimado en todas direcciones ( $360^\circ$ ). Sin embargo, como método de detección de la posición del párpado, si no se detecta alguno de los *glints* se asume que hay oclusión del párpado en esa dirección con lo que no se trazan los rayos en aquella dirección, ya que se entiende que tampoco se encontrará la pupila.

Después, se calcula el gradiente (derivada) a lo largo de cada rayo. Los puntos del borde de la pupila se consideran que son el punto de cada rayo en que el gradiente es máximo, siempre y cuando no pertenezca a la máscara de los *glints*, su gradiente esté dentro de unos límites predeterminados y tenga un nivel de gris mayor a un umbral predeterminado. El proceso de detección del contorno de la pupila es iterativo hasta que el centro de los puntos detectados converge (igual que en el *Starburst* original).

Finalmente, se aplica el algoritmo *Ellipse-RANSAC* que es un método iterativo que selecciona aleatoriamente pequeños subconjuntos de puntos para detectar *outliers* y después omitirlos en el ajuste. El método original de *Ellipse-RANSAC* elige aleatoriamente subconjuntos de 5 puntos en cada iteración. Sin embargo, se utilizan subconjuntos de 6 puntos.

Los datos obtenidos con este algoritmo son ofrecidos en un fichero .mat con lo que se puede abrir fácilmente en el entorno de Matlab. Este fichero se organiza de la siguiente manera:

- La primera columna corresponde al índice del punto de calibración, del 0 al 8 que son el número de estímulos.
- La segunda columna contiene un valor booleano 1 o 0. 1 ojo derecho y 0 ojo izquierdo.
- La tercera columna representa el índice de la imagen.
- La cuarta columna es un indicador para saber si el algoritmo de Vicomtech fue capaz de detectar algo en esa imagen (para poder comparar la robustez entre algoritmos).
- La columna 5 y 6 son las coordenadas de la pupila detectada (*Ellipse-RANSAC*). 7 y 8 las coordenadas del centro de los reflejos (realiza el ajuste a elipse con 12 reflejos).
- La columna 9 es la proporción entre el número de *inliers* del borde de la pupila respecto a todos los puntos detectados.
- La columna 10 corresponde al número de iteraciones que necesita para el ajuste.
- De la columna 11 a la 15 son los coeficientes para calcular la elipse ajustada de los reflejos en la forma estándar.
- La columna 16 es el semieje mayor de la pupila en pixeles.
- De la columna 17 a la 86 coordenadas horizontales de los puntos del borde de la pupila.
- Del 87 a la 156 coordenadas verticales de los puntos del borde de la pupila detectados.
- De la 157 a la 176 coordenadas horizontales de los centros de los reflejos corneales.
- De la 177 a la 196 coordenadas verticales de los centros de los reflejos corneales.

## 2.3 VICOMTECH

La empresa Vicomtech creó el eye-tracker para EVA. Además, proporcionaron a Davalor unos ficheros .xdl de los cuales obtuvieron la información necesaria para poder comparar con los resultados obtenidos por el método *SuperStarburst*. En estos ficheros se observa que el algoritmo de Vicomtech ofrece los datos o características de las imágenes de calibración y validación:

Dentro de la carpeta ‘calibration’ están las subcarpetas correspondientes a cada ojo. Junto con las imágenes está el fichero calib.xdl. Este fichero contiene las posiciones del estímulo (‘coords’), el vector pupila reflejo (VD), si Vicomtech ha podido detectar el ojo (Y/N), el ojo que es (‘eye’, 1 OD /0 OI), el nombre de la imagen correspondiente, las coordenadas del centro de la elipse ajustada a los *glints* o reflejos (‘g’), el radio de dicha elipse (‘gd’) y las coordenadas del centro de la pupila (‘p’) según el algoritmo de Vicomtech.

Dentro de la carpeta 'validation' están las subcarpetas correspondientes a cada ojo. En este caso, el fichero 'valid.xdl' es común para ambos ojos. Contiene esencialmente la misma información que el de calibración.





# Capítulo 3

## Algoritmos propuestos

Los algoritmos que se van a proponer en este capítulo se han obtenido del libro “Ellipse Fitting for Computer Vision” [17].

Para llevar a cabo el ajuste a elipse, a continuación se presentarán los algoritmos propuestos. Comentar que en estos algoritmos las matrices se denominan en mayúsculas y negrita y los vectores en minúscula y negrita.

Ajuste a elipse significa ajustar la ecuación de la elipse Ec. (3.1) a puntos extraídos de la imagen.

$$Ax^2 + 2Bxy + Cy^2 + 2f_0(Dx + Fy) + f_0^2G = 0 \quad (3.1)$$

Donde  $f_0$  es una constante de ajuste de escala. Por lo tanto, dependiendo de la región en la que se encuentre la elipse se establecerá el valor de  $f_0$ ; es decir, en el caso de que se supiera que se encuentra en una región de 600 x 600 píxeles, se establece  $f_0 = 600$ . Es muy importante recalcar que la Ec. (3.1) tiene una escala indeterminada, por ejemplo, si se multiplican los valores A, B, C, D, F y G por una constante que no sea cero se representa la misma elipse. Por ello, se necesita algún tipo de normalización. En los siguientes algoritmos se utilizará la siguiente ecuación de normalización:

$$A^2 + B^2 + C^2 + D^2 + F^2 + G^2 = 1 \quad (3.2)$$

Si se definen los vectores 6-D,

$$\xi = \begin{pmatrix} x^2 \\ 2xy \\ y^2 \\ 2f_0x \\ 2f_0y \\ f_0^2 \end{pmatrix} \quad \theta = \begin{pmatrix} A \\ B \\ C \\ D \\ F \\ G \end{pmatrix} \quad (3.3)$$

La Ec. (3.1) se puede escribir como,

$$(\xi, \theta) = 0 \quad (3.4)$$

donde el producto escalar de los vectores  $\xi$  y  $\theta$  se denota como  $(\xi, \theta)$ .

Como el vector  $\theta$  tiene una escala indeterminada hay que normalizarlo:

$$(\theta, N\theta) = 1 \quad (3.5)$$

El uso de la Ec. (3.2) corresponde a  $N = \mathbf{I}$  (identidad) matriz de normalización. Respecto a la Ec. (3.5)  $\|\theta\| = 1$ , por lo tanto se normaliza a vector unitario.

## 3.1 AJUSTE ALGEBRAICO

Todos los métodos de ajuste algebraico resuelven el problema de los autovalores de la misma manera. Para ello calculan la  $\theta$  que satisfaga  $M\theta = \lambda N\theta$  donde  $M$  y  $N$  se definen en términos de la incógnita  $\theta$  y  $\xi_\alpha$ , siendo  $\alpha$  cada punto de la secuencia que corresponde a la elipse.

### 3.1.1 Least Square

Es uno de los primeros métodos que se creó para ajustar la elipse en forma de Ec. (3.1) a una secuencia de puntos  $(x_1, y_1), \dots, (x_n, y_n)$  con presencia de ruido (Fig.3.1). Consiste en encontrar los valores A, B, C, D, F y G para ello

$$Ax_\alpha^2 + 2Bx_\alpha y_\alpha + Cy_\alpha^2 + 2f_0(Dx_\alpha + Fy_\alpha) + f_0^2 G \approx 0, \quad \alpha = 1, \dots, N. \quad (3.6)$$

se reemplazan los valores  $x$ ,  $y$  por  $x_\alpha, y_\alpha$  en el vector 6-D y se obtiene  $\xi_\alpha$  con lo que,

$$(\xi_\alpha, \theta) \approx 0, \quad \alpha = 1, \dots, N. \quad (3.7)$$

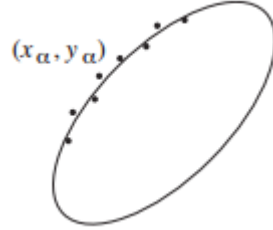


Figura 3.1: Ajustando la elipse a una secuencia de puntos con ruido [17]

Procedimiento del algoritmo:

1. Calcular la matriz 6x6:

$$M = \frac{1}{N} \sum_{\alpha=1}^N \xi_\alpha \xi_\alpha^T \quad N = I \quad (3.8)$$

2. Resolver el problema de los autovalores y devolver el autovector unitario  $\theta$  para el menor autovalor  $\lambda$ .

$$M\theta = \lambda\theta \quad (3.9)$$

Es importante recalcar que cuando los puntos de entrada de la elipse cubren solamente una pequeña parte de la circunferencia de la elipse, este algoritmo produce a menudo una elipse plana y pequeña la cual es muy diferente a la forma real. De hecho es un algoritmo de poca precisión y esto se debe a que las propiedades del ruido en las imágenes no se tienen en cuenta en el diseño del método; para llevar a cabo un ajuste preciso es necesario tener en consideración las propiedades estadísticas del ruido. Suponer que los valores  $x_\alpha, y_\alpha$  son perturbados de sus valores verdaderos  $\bar{x}_\alpha, \bar{y}_\alpha$  por  $\Delta x_\alpha, \Delta y_\alpha$ .

$$x_\alpha = \bar{x}_\alpha + \Delta x_\alpha \quad y_\alpha = \bar{y}_\alpha + \Delta y_\alpha \quad (3.10)$$

Sustituyendo esto por  $\xi_\alpha$  se puede escribir como,

$$\xi_\alpha = \bar{\xi}_\alpha + \Delta_1 \xi_\alpha + \Delta_2 \xi_\alpha \quad (3.11)$$

Donde  $\bar{\xi}_\alpha$  es el valor obtenido de  $\xi_\alpha$  reemplazando  $x_\alpha, y_\alpha$  por sus valores verdaderos  $\bar{x}_\alpha, \bar{y}_\alpha$  respectivamente, mientras  $\Delta_1 \xi_\alpha, \Delta_2 \xi_\alpha$  son los términos de ruido de primer y segundo orden respectivamente. De la Ec. (3.3) se obtienen las siguientes expresiones:

$$\Delta_1 \xi_\alpha = \begin{pmatrix} 2\bar{x}_\alpha \Delta x_\alpha \\ 2\Delta x_\alpha \bar{y}_\alpha + 2\bar{x}_\alpha \Delta y_\alpha \\ 2\bar{y}_\alpha \Delta y_\alpha \\ 2f_0 \Delta x_\alpha \\ 2f_0 \Delta y_\alpha \\ 0 \end{pmatrix} \quad \Delta_2 \xi_\alpha = \begin{pmatrix} \Delta x_\alpha^2 \\ 2\Delta x_\alpha \Delta y_\alpha \\ \Delta y_\alpha^2 \\ 0 \\ 0 \\ 0 \end{pmatrix} \quad (3.12)$$

Se consideran los términos del ruido como variables aleatorias y se define la matriz de covarianza  $V[\xi_\alpha]$  por

$$V[\xi_\alpha] = E[\Delta_1 \xi_\alpha \Delta_1 \xi_\alpha^T] \quad (3.13)$$

donde  $E$  denota el valor esperado sobre la distribución del ruido. Si se asume que  $\Delta x_\alpha, \Delta y_\alpha$  son muestreados por distribuciones Gaussianas independientes de media 0 y desviación estándar  $\sigma$ , se obtiene

$$E[\Delta x_\alpha] = E[\Delta y_\alpha] = 0, \quad E[\Delta x_\alpha^2] = E[\Delta y_\alpha^2] = \sigma^2, \quad E[\Delta x_\alpha \Delta y_\alpha] = 0 \quad (3.14)$$

Sustituyendo en la Ec. (3.12) y utilizando la relación anterior, se obtiene la matriz de covarianza de la siguiente manera:

$$V[\xi_\alpha] = \sigma^2 V_0[\xi_\alpha], \quad V_0[\xi_\alpha] = 4 \begin{pmatrix} \bar{x}_\alpha^2 & \bar{x}_\alpha \bar{y}_\alpha & 0 & f_0 \bar{x}_\alpha & 0 & 0 \\ \bar{x}_\alpha \bar{y}_\alpha & \bar{x}_\alpha^2 + \bar{y}_\alpha^2 & \bar{x}_\alpha \bar{y}_\alpha & f_0 \bar{y}_\alpha & f_0 \bar{x}_\alpha & 0 \\ 0 & \bar{x}_\alpha \bar{y}_\alpha & \bar{y}_\alpha^2 & 0 & f_0 \bar{y}_\alpha & 0 \\ f_0 \bar{x}_\alpha & f_0 \bar{y}_\alpha & 0 & f_0^2 & 0 & 0 \\ 0 & f_0 \bar{x}_\alpha & f_0 \bar{y}_\alpha & 0 & f_0^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad (3.15)$$

Ya que los elementos de  $V[\xi_\alpha]$  tienen el múltiplo  $\sigma^2$ , se saca el factor común fuera y se denomina  $V_0[\xi_\alpha]$  matriz de covarianza normalizada. A la desviación estándar  $\sigma$  también se le puede llamar nivel de ruido. Los elementos de la diagonal de la matriz de covarianza  $V[\xi_\alpha]$  indican la susceptibilidad al ruido de cada componente de  $\xi_\alpha$  y los elementos que están fuera de la diagonal miden la correlación por pares.

La matriz de covarianza se define solamente en términos de primer orden del ruido  $\Delta_1 \xi_\alpha$ , ya que la incorporación del segundo orden tiene poca influencia en el resultado final. Esto se debe a que  $\Delta_2 \xi_\alpha$  es muy pequeño en comparación a  $\Delta_1 \xi_\alpha$ . Además, es interesante fijarse que los elementos

de  $\mathbf{V}_0[\xi_\alpha]$  contienen los valores verdaderos en vez de los observados, aunque prácticamente no tiene ningún efecto en los resultados finales.

En algunos de los siguientes algoritmos se utilizará esta matriz de covarianza así calculada, con el objetivo de mejorar la robustez del ajuste frente al ruido de los puntos de entrada.

### 3.1.2 Iterative reweight

Este método consiste en minimizar la suma ponderada de los cuadrados  $\sum_{\alpha=1}^N W_\alpha (\xi_\alpha, \boldsymbol{\theta})^2$ .

El procedimiento es el siguiente:

1. Dejar  $\boldsymbol{\theta}_0 = \mathbf{0}$  y  $W_\alpha = 1$ ,  $\alpha = 1, \dots, N$ .
2. Calcular la matriz 6x6:

$$\mathbf{M} = \frac{1}{N} \sum_{\alpha=1}^N W_\alpha \xi_\alpha \xi_\alpha^T \quad N = I \quad (3.16)$$

donde  $\xi_\alpha$  es el vector definido en la Ec. (3.3) para el punto  $\alpha$ .

3. Resolver el problema del autovalor y calcular el autovector unitario  $\boldsymbol{\theta}$  para el menor autovalor  $\lambda$ .

$$\mathbf{M}\boldsymbol{\theta} = \lambda\boldsymbol{\theta} \quad (3.17)$$

4. Si  $\boldsymbol{\theta} \approx \boldsymbol{\theta}_0$  dependiendo del signo, devolver  $\boldsymbol{\theta}$  y parar el algoritmo. Si no, actualizar y volver al paso 2.

$$W_\alpha \leftarrow \frac{1}{(\boldsymbol{\theta}, \mathbf{V}_0[\xi_\alpha]\boldsymbol{\theta})}, \quad \boldsymbol{\theta}_0 \leftarrow -\boldsymbol{\theta} \quad (3.18)$$

En la primera iteración se tiene que  $W_\alpha = 1$ , lo que implica que se estaba minimizando  $\sum_{\alpha=1}^N (\xi_\alpha, \boldsymbol{\theta})^2$ . En otras palabras, la solución inicial es la solución de *Least Square*, a partir del cual empiezan las iteraciones.

En el cuarto punto la frase dependiendo del signo refleja que los autovectores tienen signos indeterminados. Por lo tanto, se alinean  $\boldsymbol{\theta}$  y  $\boldsymbol{\theta}_0$  haciendo la inversión de signo  $\boldsymbol{\theta} \leftarrow -\boldsymbol{\theta}$  y nunca  $(\boldsymbol{\theta}, \boldsymbol{\theta}_0) < 0$ .

### 3.1.3 Renormalization

El método de *Iterative reweight* tiene grandes sesgos, sobre todo cuando el arco elíptico de entrada es corto. Como resultado se suele obtener una elipse menor a la esperada. Para remediar este problema se realizó el siguiente algoritmo:

Procedimiento:

1. Dejar  $\theta_0 = \mathbf{0}$  y  $W_\alpha = 1$ ,  $\alpha = 1, \dots, N$ .

Calcular las matrices 6x6:

$$\mathbf{M} = \frac{1}{N} \sum_{\alpha=1}^N W_\alpha \xi_\alpha \xi_\alpha^T \quad \mathbf{N} = \frac{1}{N} \sum_{\alpha=1}^N W_\alpha \mathbf{V}_0[\xi_\alpha] \quad (3.19)$$

2. Resolver el problema de los autovalores y calcular el autovector unitario  $\theta$  para el menor autovalor  $\lambda$ .

$$\mathbf{M}\theta = \lambda \mathbf{N}\theta \quad (3.20)$$

3. Si  $\theta \approx \theta_0$  dependiendo del signo, devolver  $\theta$  y parar el algoritmo. Si no, actualizar y volver al paso 2.

$$W_\alpha \leftarrow \frac{1}{(\theta, \mathbf{V}_0[\xi_\alpha]\theta)}, \quad \theta_0 \leftarrow \theta \quad (3.21)$$

### 3.1.4 Taubin

En algebra lineal resolver la Ec. (3.20) es equivalente a minimizar la forma cuadrática de  $(\theta, \mathbf{M}\theta)$  a la restricción de  $(\theta, \mathbf{N}\theta) = constante$ . En este método se minimiza  $\sum_{\alpha=1}^N (\xi_\alpha, \theta)^2$  sujeto a  $(\theta, (\sum_{\alpha=1}^N \mathbf{V}_0[\xi_\alpha]\theta)) = constante$ .

Procedimiento:

1. Calcular las matrices 6x6:

$$\mathbf{M} = \frac{1}{N} \sum_{\alpha=1}^N \xi_\alpha \xi_\alpha^T \quad \mathbf{N} = \frac{1}{N} \sum_{\alpha=1}^N \mathbf{V}_0[\xi_\alpha] \quad (3.22)$$

2. Resolver el problema de los autovalores y calcular el autovector unitario  $\theta$  para el menor autovalor  $\lambda$ .

$$\mathbf{M}\theta = \lambda\mathbf{N}\theta \quad (3.23)$$

### 3.1.5 Hyper-renormalization

Este método se creó para mejorar el método *Renormalization* modificando la matriz  $\mathbf{N}$  en la Ec. (3.20) para que la solución  $\theta$  tuviera una mayor exactitud.

Procedimiento:

1. Dejar  $\theta_0 = \mathbf{0}$  y  $W_\alpha = 1$ ,  $\alpha = 1, \dots, N$ .
2. Calcular las matrices 6x6:

$$\begin{aligned} \mathbf{M} &= \frac{1}{N} \sum_{\alpha=1}^N W_\alpha \xi_\alpha \xi_\alpha^T \\ \mathbf{N} &= \frac{1}{N} \sum_{\alpha=1}^N W_\alpha (\mathbf{V}_0[\xi_\alpha] + 2\mathbf{S}[\xi_\alpha \mathbf{e}^T]) \\ &\quad - \frac{1}{N^2} \sum_{\alpha=1}^N W_\alpha^2 ((\xi_\alpha, \mathbf{M}_5^- \xi_\alpha) \mathbf{V}_0[\xi_\alpha] + 2\mathbf{S}[\mathbf{V}_0[\xi_\alpha] \mathbf{M}_5^- \xi_\alpha \xi_\alpha^T]) \end{aligned} \quad (3.24)$$

donde  $\mathbf{S}[\cdot]$  es el operador de simetrización ( $\mathbf{S}[\mathbf{A}] = (\mathbf{A} + \mathbf{A}^T)/2$ ), y  $\mathbf{e}$  es el vector

$$\mathbf{e} = (1, 0, 1, 0, 0, 0)^T. \quad (3.25)$$

La matriz  $\mathbf{M}_5^-$  es la pseudoinversa de  $\mathbf{M}$  pero de dimensiones 5x5.

3. Resolver el problema de los autovalores y calcular el autovector unitario  $\theta$  para el menor autovalor  $\lambda$ .

$$\mathbf{M}\theta = \lambda\mathbf{N}\theta \quad (3.26)$$

4. Si  $\theta \approx \theta_0$  dependiendo del signo, devolver  $\theta$  y parar el algoritmo. Si no, actualizar y volver al paso 2.

$$W_\alpha \leftarrow \frac{1}{(\theta, \mathbf{V}_0[\xi_\alpha] \theta)}, \quad \theta_0 \leftarrow \theta \quad (3.27)$$

El vector  $\mathbf{e}$  de la Ec. (3.25) está definido de manera que se ocupa del ruido de segundo orden  $\Delta_2 \xi_\alpha$ .

$$\mathbf{E}[\Delta_2 \xi_\alpha] = \sigma^2 \mathbf{e} \quad (3.28)$$

El  $\mathbf{M}_5^-$  se calcula para los autovalores  $\mu_1 \geq \dots \geq \mu_6$  de  $\mathbf{M}$  y su correspondiente autovector unitario de  $\mathbf{M}$  de la siguiente manera:

$$\mathbf{M}_5^- = \frac{1}{\mu_1} \boldsymbol{\theta}_1 \boldsymbol{\theta}_1^T + \dots + \frac{1}{\mu_5} \boldsymbol{\theta}_5 \boldsymbol{\theta}_5^T \quad (3.29)$$

con lo que los términos  $\mu_6$  y  $\boldsymbol{\theta}_6$  se han eliminado.

### 3.1.6 HyperLS

El método *HyperLS* minimiza la solución inicial del *Hyper-renormalization*  $\sum_{\alpha=1}^N (\xi_\alpha, \boldsymbol{\theta})^2$  sujeto a  $(\boldsymbol{\theta}, \mathbf{N}\boldsymbol{\theta}) = \text{constante}$ .

Procedimiento:

1. Calcular las matrices 6x6:

$$\begin{aligned} \mathbf{M} &= \frac{1}{N} \sum_{\alpha=1}^N \xi_\alpha \xi_\alpha^T \\ N &= \frac{1}{N} \sum_{\alpha=1}^N (\mathbf{V}_0[\xi_\alpha] + 2\mathcal{S}[\xi_\alpha \mathbf{e}^T]) \\ &\quad - \frac{1}{N^2} \sum_{\alpha=1}^N ((\xi_\alpha, \mathbf{M}_5^- \xi_\alpha) \mathbf{V}_0[\xi_\alpha] + 2\mathcal{S}[\mathbf{V}_0[\xi_\alpha] \mathbf{M}_5^- \xi_\alpha \xi_\alpha^T]) \end{aligned} \quad (3.30)$$

2. Resolver el problema de los autovalores y calcular el autovector unitario  $\boldsymbol{\theta}$  para el menor autovalor  $\lambda$ .

$$\mathbf{M}\boldsymbol{\theta} = \lambda \mathbf{N}\boldsymbol{\theta} \quad (3.31)$$



## 3.2 AJUSTE GEOMÉTRICO

El ajuste geométrico consiste en calcular la elipse que minimice la suma de los cuadrados de las distancias  $d_\alpha$  de los puntos observados  $(x_\alpha, y_\alpha)$  de la elipse. Se denotará  $(\bar{x}_\alpha, \bar{y}_\alpha)$  al punto de la elipse cercano a  $(x_\alpha, y_\alpha)$ . Se calcula la elipse que minimice:

$$S = \frac{1}{N} \sum_{\alpha=1}^N ((x_\alpha - \bar{x}_\alpha)^2 + (y_\alpha - \bar{y}_\alpha)^2) = \frac{1}{N} \sum_{\alpha=1}^N d_\alpha^2 \quad (3.32)$$

El cual se conoce como distancia geométrica. Hablando estrictamente, tendría que llamarse *distancia geométrica cuadrada*, porque tiene las dimensiones de cuadrado. Sin embargo, por simplicidad se suele usar el término *distancia geométrica*.

Aquí las notaciones de  $\bar{x}_\alpha, \bar{y}_\alpha$  se usan de manera diferente al ajuste algebraico. En el contexto de ajuste geométrico, los valores  $(x_\alpha, y_\alpha)$  son variables de estimación en vez de valores reales. Se usa este convenio para evitar introducir nuevos símbolos. La Ec. (3.32) también se podría escribir como  $S = \frac{1}{N} \sum_{\alpha=1}^N (\Delta x_\alpha^2 + \Delta y_\alpha^2)$ .

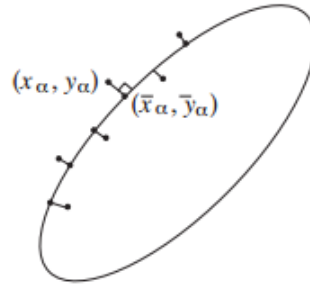


Figura 3.2: Ajuste a elipse minimizando la distancia geométrica [17].

Si el punto  $(x_\alpha, y_\alpha)$  se encuentra cerca de la elipse,  $d_\alpha^2$  se escribe como  $\Delta x_\alpha$  y  $\Delta y_\alpha$  excepto para términos de orden alto:

$$d_\alpha^2 = (x_\alpha - \bar{x}_\alpha)^2 + (y_\alpha - \bar{y}_\alpha)^2 \approx \frac{(\xi_\alpha, \theta)^2}{\theta, \mathbf{V}_0[\hat{\xi}_\alpha]\theta} \quad (3.33)$$

Por lo tanto, la distancia geométrica en la Ec. (3.32) se puede aproximar por lo que se conoce como error Sampson:

$$J = \frac{1}{N} \frac{(\xi_\alpha, \theta)^2}{\theta, \mathbf{V}_0[\hat{\xi}_\alpha]\theta} \quad (3.34)$$

### 3.2.1 FNS

Este algoritmo es un método para minimizar el error Sampson de la Ec. (3.34).

Procedimiento:

1. Dejar  $\boldsymbol{\theta} = \boldsymbol{\theta}_0$  y  $W_\alpha = 1$ ,  $\alpha = 1, \dots, N$ .
2. Calcular la matriz 6x6:

$$\mathbf{M} = \frac{1}{N} \sum_{\alpha=1}^N W_\alpha \boldsymbol{\xi}_\alpha \boldsymbol{\xi}_\alpha^T \quad \mathbf{L} = \frac{1}{N} \sum_{\alpha=1}^N W_\alpha^2 (\boldsymbol{\xi}_\alpha, \boldsymbol{\theta})^2 \mathbf{V}_0[\boldsymbol{\xi}_\alpha] \quad (3.35)$$

3. Calcular la matriz 6x6

$$\mathbf{X} = \mathbf{M} - \mathbf{L} \quad (3.36)$$

4. Resolver el problema de los autovalores y calcular el autovector unitario  $\boldsymbol{\theta}$  para el autovalor  $\lambda$  más pequeño

$$\mathbf{X}\boldsymbol{\theta} = \lambda\boldsymbol{\theta} \quad (3.37)$$

5. Si  $\boldsymbol{\theta} \approx \boldsymbol{\theta}_0$  dependiendo del signo, devolver  $\boldsymbol{\theta}$  y parar el algoritmo. Si no, actualizar y volver al paso 2.

$$W_\alpha \leftarrow \frac{1}{(\boldsymbol{\theta}, \mathbf{V}_0[\boldsymbol{\xi}_\alpha]\boldsymbol{\theta})}, \quad \boldsymbol{\theta}_0 \leftarrow \boldsymbol{\theta} \quad (3.38)$$

### 3.2.2 Geometric Distance Minimization

Una vez calculado  $\boldsymbol{\theta}$  por *FNS*, se puede modificar iterativamente con lo que minimiza la distancia geométrica de la Ec. (3.32). Para ser específicos, se modifica el valor  $\boldsymbol{\xi}_\alpha$ , usando la solución actual  $\boldsymbol{\theta}$ . Después de unas iteraciones, el error Sampson coincide con la distancia geométrica. El proceso es el siguiente:

1. Dejar  $J_0 = \infty$  (un número suficientemente largo)

$$\hat{x}_\alpha = x_\alpha, \hat{y}_\alpha = y_\alpha, \check{x}_\alpha = \check{y}_\alpha = 0, \quad \alpha = 1, \dots, N.$$

2. Calcular la matriz de covarianza normalizada  $\mathbf{V}_0[\hat{\xi}_\alpha]$  reemplazando  $\bar{x}_\alpha, \bar{y}_\alpha$  en la definición de  $\mathbf{V}_0[\xi_\alpha]$  por  $\hat{x}_\alpha, \hat{y}_\alpha$  respectivamente.
3. Calculando el siguiente valor modificado  $\xi_\alpha^*$ :

$$\xi_\alpha^* = \begin{pmatrix} \hat{x}_\alpha^2 + 2\hat{x}_\alpha\check{x}_\alpha \\ 2(\hat{x}_\alpha\hat{y}_\alpha + \hat{y}_\alpha\check{x}_\alpha + \check{x}_\alpha\check{y}_\alpha) \\ \hat{y}_\alpha^2 + 2\hat{y}_\alpha\check{y}_\alpha \\ 2f_0(\hat{x}_\alpha + \check{x}_\alpha) \\ 2f_0(\hat{y}_\alpha + \check{y}_\alpha) \\ f_0^2 \end{pmatrix} \quad (3.39)$$

4. Calcular el valor  $\theta$  que minimice el error Sampson modificado:

$$J^* = \frac{1}{N} \sum_{\alpha=1}^N \frac{(\xi_\alpha^*)^2}{(\theta, \mathbf{V}_0[\hat{\xi}_\alpha]\theta)} \quad (3.40)$$

5. Actualizar  $\check{x}_\alpha, \check{y}_\alpha$  de la siguiente manera:

$$\begin{pmatrix} \check{x}_\alpha \\ \check{y}_\alpha \end{pmatrix} \leftarrow \frac{2(\xi_\alpha^*\theta)}{(\theta, \mathbf{V}_0[\hat{\xi}_\alpha]\theta)} \begin{pmatrix} \theta_1 & \theta_2 & \theta_4 \\ \theta_2 & \theta_3 & \theta_5 \end{pmatrix} \begin{pmatrix} \hat{x}_\alpha \\ \hat{y}_\alpha \\ f_0 \end{pmatrix} \quad (3.41)$$

6. Actualizar  $\hat{x}_\alpha, \hat{y}_\alpha$ :

$$\hat{x}_\alpha \leftarrow x_\alpha - \check{x}_\alpha, \quad \hat{y}_\alpha \leftarrow y_\alpha - \check{y}_\alpha \quad (3.42)$$

7. Calcular

$$J^* = \frac{1}{N} \sum_{\alpha=1}^N (\check{x}_\alpha^2 + \check{y}_\alpha^2) \quad (3.43)$$

Si  $J^* \approx J_0$ , devolver  $\theta$  y parar. Sino, dejar  $J_0 \leftarrow J^*$ , y volver al paso 2.

### 3.2.3 Hyperaccurate correction

Este método fue realizado para mejorar la precisión.

Procedimiento:

1. Calcular  $\boldsymbol{\theta}$  por *FNS*
2. Estimar  $\sigma^2$  de la forma

$$\hat{\sigma}^2 = \frac{(\boldsymbol{\theta}, \mathbf{M}\boldsymbol{\theta})}{1 - 5/N} \quad (3.44)$$

Donde  $\mathbf{M}$  es el valor de la matriz  $\mathbf{M}$  en la Ec. (3.38) después de que las iteraciones en *FNS* converjan.

3. Calcular el termino de corrección

$$\Delta_c \boldsymbol{\theta} = -\frac{\hat{\sigma}^2}{N} \mathbf{M}_5^- \sum_{\alpha=1}^N W_{\alpha}(\mathbf{e}, \boldsymbol{\theta}) \boldsymbol{\xi}_{\alpha} + \frac{\hat{\sigma}^2}{N^2} \mathbf{M}_5^- \sum_{\alpha=1}^N W_{\alpha}^2(\boldsymbol{\xi}_{\alpha}, \mathbf{M}_5^- \mathbf{V}_0[\boldsymbol{\xi}_{\alpha}] \boldsymbol{\theta}) \boldsymbol{\xi}_{\alpha} \quad (3.45)$$

4. Corregir  $\boldsymbol{\theta}$  en

$$\boldsymbol{\theta} \leftarrow \mathcal{N}[\boldsymbol{\theta} - \Delta_c \boldsymbol{\theta}] \quad (3.46)$$

Donde  $\mathcal{N}[\cdot]$  indica normalización con el criterio unitario ( $\mathcal{N}[\mathbf{a}] = \mathbf{a}/\|\mathbf{a}\|$ ).

## 3.3 DETECCIÓN DE OUTLIERS

Consiste en detectar los puntos que se encuentran fuera del contorno y así poder omitirlos en el ajuste. Gracias a esto se aumentará la robustez del sistema y tendrá mayor precisión. El método que lleva esto a cabo se denomina *Ellipse-RANSAC* (Random Sample Consensus).

### 3.3.1 Ellipse - RANSAC

Procedimiento:

1. Elegir aleatoriamente cinco puntos de la secuencia de puntos de entrada y dejar  $\xi_1, \xi_2, \dots, \xi_5$  como vectores correspondientes a la Ec. (3.3).
2. Calcular el autovector unitario  $\theta$  de la siguiente matriz para el autovalor más pequeño  $\lambda$ , y guardarlo si es candidato.

$$M_5 = \sum_{\alpha=1}^5 \xi_{\alpha} \xi_{\alpha}^T \quad (3.47)$$

3. Siendo n el número de puntos de la secuencia de entrada que satisfaga

$$\frac{(\xi, \theta)^2}{(\theta, V_0[\xi]\theta)} < d^2 \quad (3.48)$$

Donde  $\xi$  es el vector representativo de la secuencia de puntos en la forma de la Ec. (3.3).

$V_0$  es la matriz de covarianza normalizada definido por la Ec. (3.15)

$d$  es un umbral de desviación admisible para el ajuste a elipse, por ejemplo,  $d=2$ (pixeles) representa el criterio para considerar que el punto pertenece a la elipse.

4. Se guarda el valor n
5. Seleccionar un nuevo conjunto de 5 puntos de la secuencia de entrada y rehacer todos los pasos. Repetir este proceso las veces que se establezca y entre las elipses candidatas guardadas devolver la que tenga la n más grande.

### 3.3.2 Random ellipse-RANSAC

A diferencia de este algoritmo los métodos anteriores están referenciados en la literatura. Este método ha sido propuesto como mejora a las debilidades del *Ellipse-RANSAC* que se observarán en el capítulo 4.

Procedimiento:

1. Elegir aleatoriamente cinco puntos de la secuencia de puntos de entrada y dejar  $\xi_1, \xi_2, \dots, \xi_5$  como vectores correspondientes a la Ec. (3.3).
2. Calcular el autovector unitario  $\theta$  de la siguiente matriz para el autovalor más pequeño  $\lambda$ , y guardarlo si es candidato.

$$M_5 = \sum_{\alpha=1}^5 \xi_{\alpha} \xi_{\alpha}^T \quad (3.49)$$

3. Dejar que  $n$  sea el número de puntos de la secuencia de entrada que satisfaga

$$\frac{(\xi, \theta)^2}{(\theta, V_0[\xi]\theta)} < d^2 \quad (3.50)$$

Donde  $\xi$  es el vector representativo de la secuencia de puntos en la forma de la Ec. (3.3).

$V_0$  es la matriz de covarianza normalizada definido por la Ec. (3.15)

$d$  es un umbral de desviación admisible para el ajuste a elipse, es decir,  $d=2$ (pixeles) representa el criterio para considerar que el punto pertenece a la elipse.

Se guarda el valor  $n$  y el índice de cada punto del conjunto  $n$  en un array, es decir, los inliers. Estos valores se modificarán siempre y cuando el número de inliers de la siguiente iteración sea mayor que la guardada.

4. Seleccionar un nuevo conjunto de 5 puntos de la secuencia de entrada y rehacer todos los pasos. Repetir este proceso las veces que se quiera.
5. Calcular la matriz 6x6 con los puntos del array de los inliers:

$$M = \sum_{\alpha=1}^N \xi_{\alpha} \xi_{\alpha}^T \quad (3.51)$$

6. Resolver el problema de los autovalores y devolver el autovector unitario  $\theta$  para el menor autovalor  $\lambda$ .

$$M\theta = \lambda\theta \quad (3.52)$$

## 3.4 AJUSTE ESPECÍFICO A ELIPSE

La Ec. (3.1) no siempre representa una elipse; dependiendo de los coeficientes puede representar una parábola o una hipérbola. Para evitar esto, la Ec. (3.1) tiene que cumplir la siguiente condición:

$$AC - B^2 > 0 \quad (3.53)$$

En el caso de que la secuencia no tenga suficiente información, es decir, que sea muy corta o con mucho ruido, el resultado suele ser una hipérbola; la parábola se da cuando la solución satisface la condición  $AC - B^2 = 0$ .

### 3.4.1 Fitzgibbon

El método de *Fitzgibbon* se creó para que siempre se forzara el ajuste a una elipse.

1. Calcular la matriz 6x6:

$$M = \frac{1}{N} \sum_{\alpha=1}^N \xi_{\alpha} \xi_{\alpha}^T \quad (3.54)$$

$$N = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & -2 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

2. Resolver el problema de los autovalores y calcular el autovector unitario  $\theta$  para el menor autovalor  $\lambda$ .

$$M\theta = \lambda N\theta \quad (3.55)$$

### 3.4.2 Random Sampling

Otro método para ajustar siempre una elipse es aplicar un método estándar como *Hyper-Renormalization* y después modificar los resultados en el caso de que no se ajustara a la elipse para que así lo hiciera. Para ello un método simple y efectivo es el *Random Sampling* del mismo modo que *Ellipse-RANSAC* elige aleatoriamente cinco puntos de la secuencia de entrada, se ajusta una elipse a ellos, y repite el proceso hasta que encuentra la elipse que mejor se ajuste y menor error de Sampson tenga. El proceso es el siguiente:

1. Ajustar una elipse utilizando *Hyper-renormalization* sin considerar la condición de la elipse. Si la solución  $\theta$  satisface la siguiente condición devolver como solución final.

$$\theta_1\theta_3 - \theta_2^2 > 0 \quad (3.56)$$

2. Si no, elegir aleatoriamente cinco puntos de la secuencia de puntos de entrada y dejar  $\xi_1, \xi_2, \dots, \xi_5$  como vectores correspondientes a la Ec. (3.57).

$$\xi = \begin{pmatrix} x^2 \\ 2xy \\ y^2 \\ 2f_0x \\ 2f_0y \\ f_0^2 \end{pmatrix} \quad \theta = \begin{pmatrix} A \\ B \\ C \\ D \\ F \\ G \end{pmatrix} \quad (3.57)$$

3. Calcular el autovector unitario  $\theta$  de la siguiente matriz para el autovalor más pequeño  $\lambda$

$$M_5 = \sum_{\alpha=1}^5 \xi_{\alpha} \xi_{\alpha}^T \quad (3.58)$$

4. Si esta  $\theta$  no satisface a la Ec. (3.56) descartar y seleccionar un nuevo conjunto aleatorio de 5 puntos para ajustar la elipse.
5. Si  $\theta$  satisface a la Ec. (3.56) guardar como candidato. También evaluar el correspondiente error de Sampson  $J$  de la Ec. (3.34) y guardar su valor.

$$J = \frac{1}{N} \sum_{\alpha=1}^N \frac{(\xi_{\alpha}, \theta)^2}{(\theta, V_0[\xi_{\alpha}]\theta)} \quad (3.59)$$



6. Repetir este proceso un número prefijado de iteraciones y devolver el valor de  $\theta$  que tenga el error Sampson más pequeño.

## 3.5 EXPERIMENTOS DE AJUSTE A ELIPSE

En este apartado se presentan algunos de los ejemplos de simulación realizados para los diferentes métodos de ajuste explicados anteriormente. En ellos se evalúa la precisión y cómo afecta el ruido al resultado final.

### 3.5.1 Ejemplo de ajuste a elipse

En este apartado se demuestra que los diferentes métodos pueden producir distintas elipses para la misma secuencia de puntos en presencia de ruido. También se compara el número de iteraciones para los distintos métodos iterativos.

Se definen 30 puntos equidistantes en la elipse de la Fig. 3.3a. Los ejes se han ajustado a 100 y 50 píxeles. Se ha añadido ruido Gaussiano de media 0 y desviación  $\sigma$  (píxeles) para cada punto de las coordenadas x e y. Se ha ajustado una elipse con los métodos *Least Square* (1), *Iterative reweight* (2), *Taubin* (3), *Renormalization* (4), *HyperLS* (5), *Hyper-renormalization* (6), *FNS* (7) y *Hyperaccurate correction* (8).

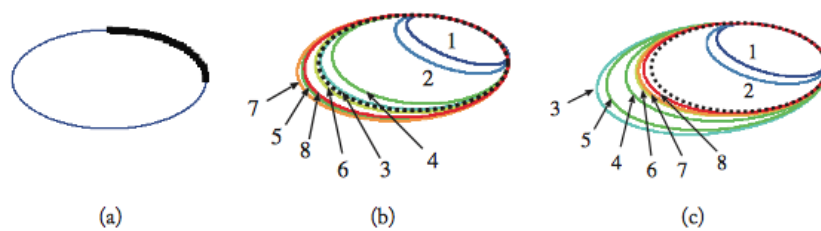


Figura 3.3: (a) 30 puntos en una elipse. (b), (c) ajuste a elipse para diferentes instancias de ruido de  $\sigma = 0.5$  [17].

Como la solución de *Geometric Distance Minimization* es casi idéntico a la solución de *FNS*, no se ha dibujado. En la figura anterior se ven las elipses ajustadas a diferentes instancias de ruido de  $\sigma = 0.5$  (píxeles). A pesar de que el nivel de ruido sea el mismo, las elipses resultantes son diferentes. *Least Square* e *Iterative reweight* tienen grandes sesgos produciendo elipses más pequeñas que la forma original. Las elipses que mejor se ajustan a la esperada son los del método *Hyper-renormalization* Fig.3.3b y *Hyperaccurate correction*. Fig.3.3c. El número de iteraciones de estos métodos son los siguientes:

Método		2	4	6	7/8
Número de iteraciones	( b )	4	4	4	9
	( c )	4	4	4	8

Tabla 3.1: Número de iteraciones de los distintos métodos.

Recordar que *Least Square*, *Taubin*, *HyperLS* y *Hyperaccurate correction* son calculados analíticamente, sin iteraciones, aunque dentro de la técnica de *Hyperaccurate correction* se realiza *FNS* con lo que también se dan iteraciones. Se puede observar que *FNS* y su *Hyperaccurate correction* requieren el doble de iteraciones que *Iterative reweight*, *Renormalization* y *Hyper-renormalization*.

### 3.5.2 Métodos específicos de ajuste a elipse

Se han comparado los métodos que fuerzan a que el ajuste a unos puntos con ruido sea una elipse mientras que los métodos estándares devuelven una hipérbola. Para ello se han comparado tres métodos: *Fitzgibbon* (1), *Hyper-renormalization* (2) y *Random Sampling* (3).

Se han generado cuatro secuencias de puntos Fig. (3.4). Los puntos tienen la misma distancia de separación de arco en cada elipse. Se añade ruido gaussiano aleatorio de media 0 y desviación estándar  $\sigma$  independiente a las coordenadas x e y de cada punto y se ajusta una elipse.

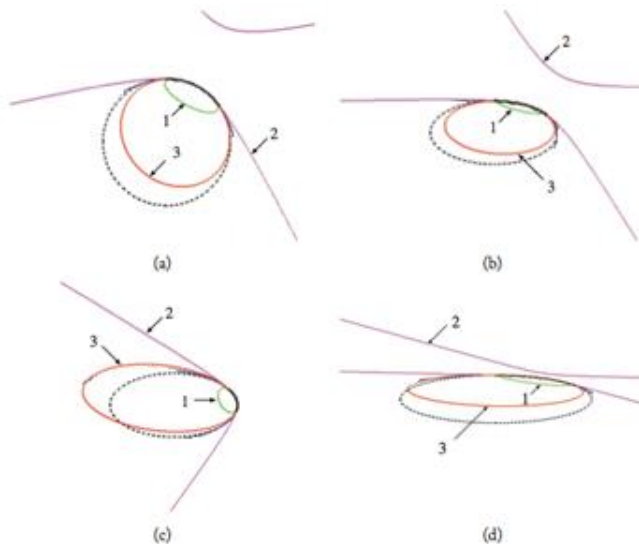


Figura 3.4: Ejemplos de ajuste cuando *hyper-renormalization* devuelve hipérbola. Nivel de ruido relativo (a) 0.169 (b) 0.151 (c) 0.092 (d) 0.087 [17].

La figura 3.4 demuestra ejemplos de ajuste para un ruido particular y *Hyper-renormalization* devuelve una hipérbola. Por otro lado, el método *Fitzgibbon* siempre ajusta una elipse pequeña y plana parecido a *Least Square* e *Iterative reweight* en la sección anterior. Finalmente, *Random Sampling* devuelve una elipse relativamente razonable.

# Capítulo 4

## Resultados

### 4.1 CARACTERÍSTICAS DE ELIPSES

Una elipse se define de la siguiente manera:

$$ax^2 + 2bxy + cy^2 + 2f_0(dx + fy) + f_0^2 g = 0 \quad (4.1)$$

Para simplificar la Ec. (4.1) a la Ec. (4.3) se han realizado las siguientes operaciones:

$$\begin{aligned} f_0 &= 600 \\ d &= f_0 * d \\ f &= f_0 * f \\ g &= f_0^2 * g \end{aligned} \quad (4.2)$$

$f_0$  se ha establecido a 600 ya que la resolución de las imágenes es de 640x480 pixeles por lo tanto en un área de 600 está garantizado que se localice la elipse.

$$ax^2 + 2bxy + cy^2 + 2dx + 2fy + g = 0 \quad (4.3)$$

En la siguiente imagen se puede ver a qué hacen referencia algunas de las letras de la ecuación anterior.

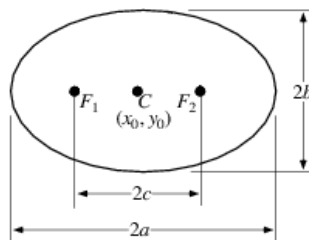


Figura 4.1: Parámetros de la elipse.

Para confirmar que el resultado obtenido corresponde a una elipse se definen los siguientes parámetros y se observa si cumplen las características de la Ec. (4.7):

$$\Delta = \begin{vmatrix} a & b & d \\ b & c & f \\ d & f & g \end{vmatrix} \quad (4.4)$$

$$J = \begin{vmatrix} a & b \\ b & c \end{vmatrix} \quad (4.5)$$

$$I = a + c \quad (4.6)$$

$$\Delta \neq 0, J > 0 \text{ y } \frac{\Delta}{I} < 0. \quad (4.7)$$

Una vez vistos que los resultados corresponden a una elipse se calcula el centro y los ejes:

$$x_0 = \frac{cd - bf}{b^2 - ac} \quad (4.8)$$

$$y_0 = \frac{af - bd}{b^2 - ac}$$

$$ejeA = \frac{\sqrt{2(af^2 + cd^2 + gb^2 - 2bdf - acg)}}{\sqrt{(b^2 - ac) [\sqrt{(a - c)^2 + 4b^2} - (a + c)]}} \quad (4.9)$$

$$ejeB = \frac{\sqrt{2(af^2 + cd^2 + gb^2 - 2bdf - acg)}}{\sqrt{(b^2 - ac) [-\sqrt{(a - c)^2 + 4b^2} - (a + c)]}}$$

Para terminar, se calcula el ángulo de rotación de la elipse de la siguiente manera:

$$\emptyset = \begin{cases} 0 & \text{para } b = 0 \text{ y } a < c \\ \frac{1}{2}\pi & \text{para } b = 0 \text{ y } a > c \\ \frac{1}{2} \cot^{-1}\left(\frac{a-c}{2b}\right) & \text{para } b \neq 0 \text{ y } a < c \\ \frac{\pi}{2} + \frac{1}{2} \cot^{-1}\left(\frac{a-c}{2b}\right) & \text{para } b \neq 0 \text{ y } a > c \end{cases} \quad (4.10)$$

## 4.2 COMPARACIÓN CENTROS DE PUPILA-REFLEJOS

En este apartado se comparan los centros de pupila y reflejos obtenidos con los algoritmos explicados en el capítulo anterior. Añadir que al no tener los centros reales o verdaderos de la pupila y reflejos, se decide comparar los centros obtenidos de los algoritmos del capítulo anterior con los dados por *Vicomtech* y *SuperStarburst*.

### 4.2.1 Diferencia de centros respecto a Vicomtech

Primero se ha calculado la diferencia entre el centro de *Vicomtech* y los algoritmos explicados en el capítulo anterior para el centro de las pupilas como el de los reflejos. Estas diferencias se han representado en histogramas acumulados y cada línea corresponde a un método y representa el porcentaje de imágenes en las cuales la diferencia de pixeles es menor al valor del eje horizontal. Las siguientes imágenes corresponden a diferencias entre centros de pupila del ojo izquierdo y derecho.

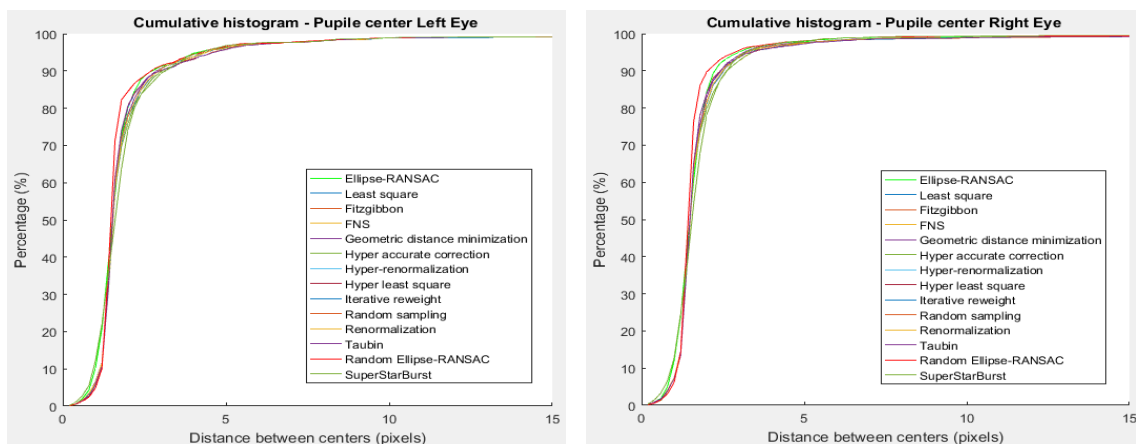


Figura 4.2: Histograma acumulado diferencia centros pupila respecto a Vicomtech.

La siguiente figura corresponde a la misma imagen pero ajustando el eje horizontal entre 0 y 4 para ver las diferencias con mayor claridad. .

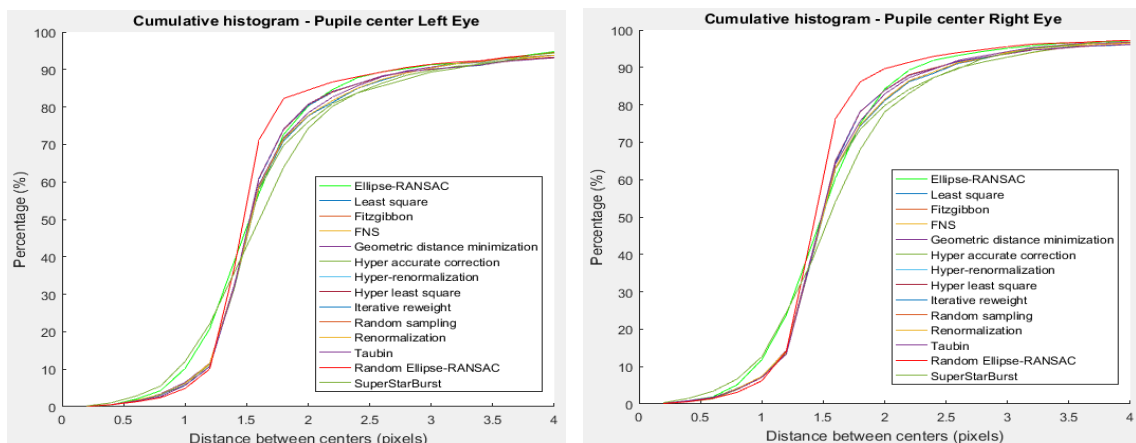


Figura 4.3: Histograma acumulado diferencia centros pupila respecto a Vicomtech de 0 a 4 pixeles.

Las siguientes imágenes corresponden a los histogramas acumulados de diferencia de centros de reflejos. La imagen de la izquierda representa al ojo izquierdo y la de la derecha al ojo derecho.

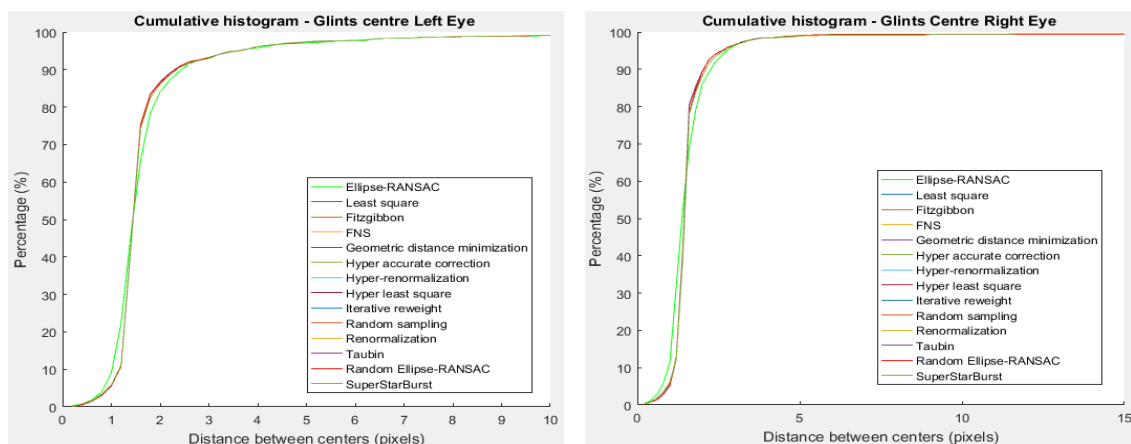


Figura 4.4: Histograma acumulado diferencia centro reflejo respecto a Vicomtech.

La siguiente figura corresponde a la misma imagen pero ajustando el eje horizontal entre 0 y 5, para ver si hay alguna diferencia entre métodos.

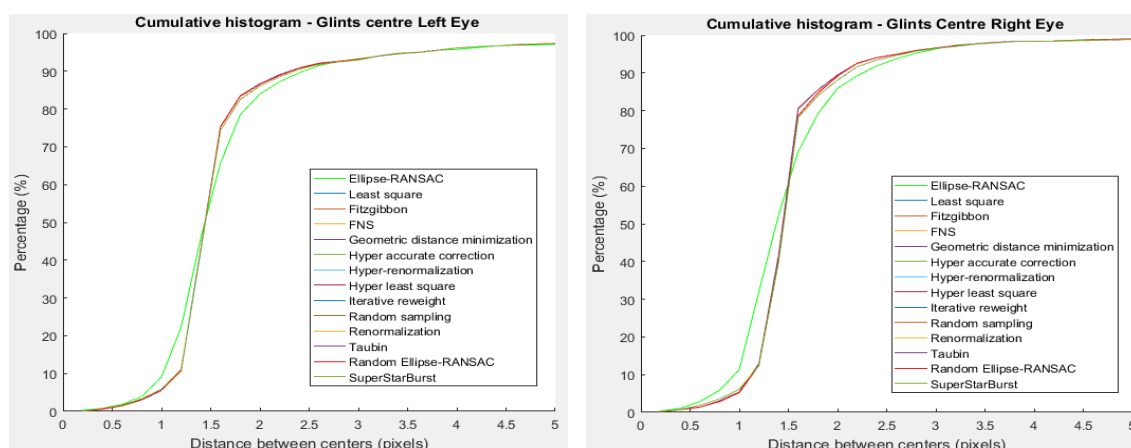


Figura 4.5: Histograma acumulado diferencia centro reflejo respecto a Vicomtech de 0 a 5 pixeles.

Como se puede observar no hay diferencias significativas entre métodos, ya que la distancia entre centros de pupila y de reflejos se asemejan mucho en todos los casos. Generalizando, el 80% de las imágenes tienen una diferencia de hasta 1.5 pixeles respecto al centro de reflejo de *Vicomtech*. En el caso del centro de pupilas, la diferencia de hasta 1.5 pixeles se da en el 60-70% de las imágenes. Por lo tanto, en el caso de los reflejos las diferencias son levemente menores que en el ajuste de pupilas.

### 4.2.2 Diferencia de centros respecto a SuperStarBurst

Después se han comparado los centros obtenidos por los métodos del capítulo anterior con el centro del *SuperStarBurst*. Para ello se ha calculado la diferencia entre el centro de *SuperStarBurst* y los algoritmos explicados en los capítulos anteriores. Esta diferencia de centros se ha obtenido bien para la pupila como para los reflejos.

Como en los anteriores histogramas acumulados, cada línea corresponde a un método y representa el porcentaje de imágenes en las cuales la diferencia de píxeles es menor al valor del eje horizontal. La imagen de la izquierda corresponde al ojo izquierdo y la imagen de la derecha al ojo derecho.

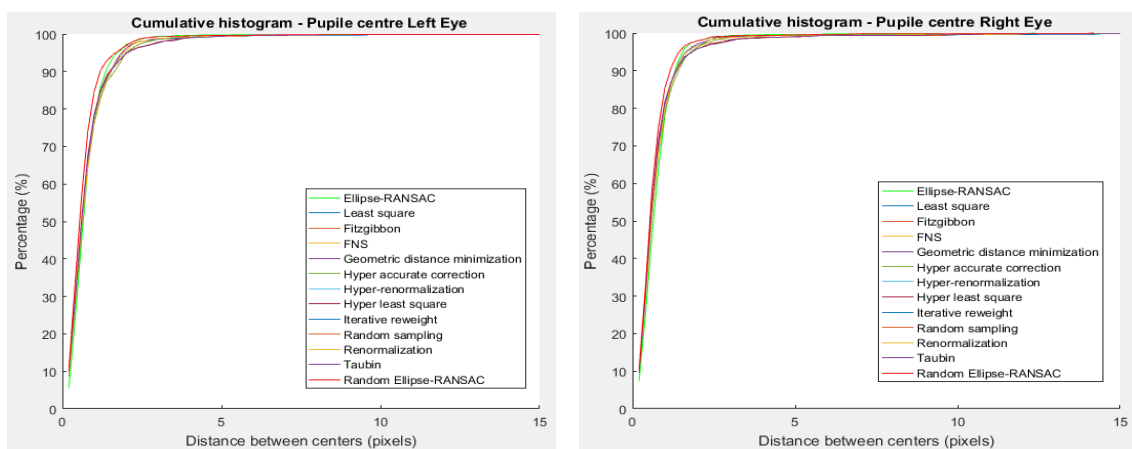


Figura 4.6: Histograma acumulado diferencia centros pupila respecto a SuperStarBurst.

La siguiente figura corresponde a la misma imagen pero ajustando el eje horizontal entre 0 y 4, para ver diferencias entre métodos.

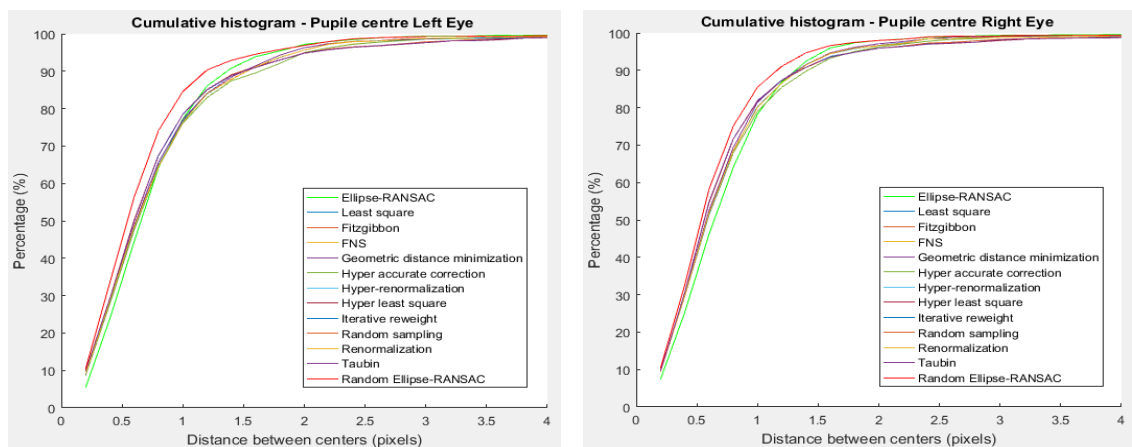


Figura 4.7: Histograma acumulado diferencia centros pupila respecto a SuperStarBurst de 0 a 4 pixeles.

Las siguientes imágenes corresponden a los histogramas acumulados de diferencia de centros de reflejos. La imagen de la izquierda al ojo izquierdo y la de la derecha al ojo derecho.

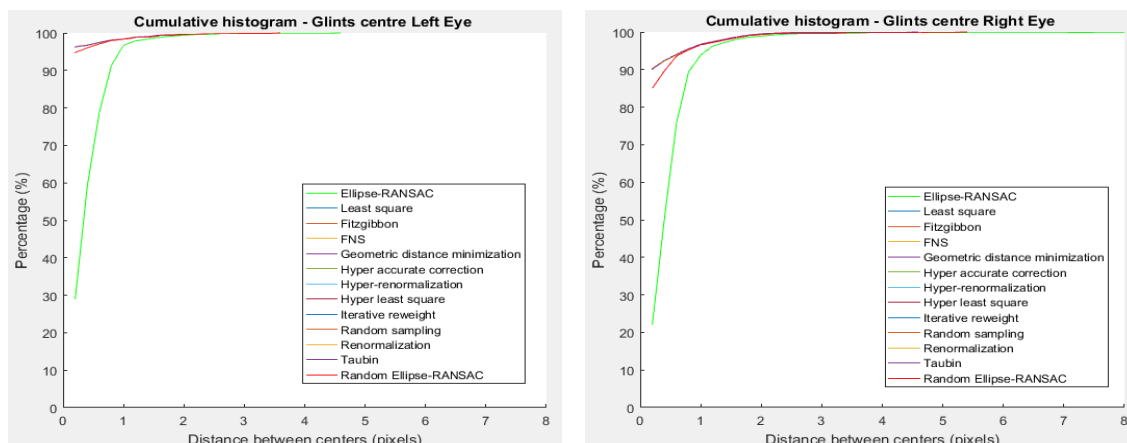


Figura 4.8: Histograma acumulado diferencia centro reflejo respecto a SuperStarBurst.

La misma imagen con un mayor zoom, es decir, el eje horizontal de 0 a 2, puesto que la diferencia entre métodos se observa entre 0 y 1 píxeles.

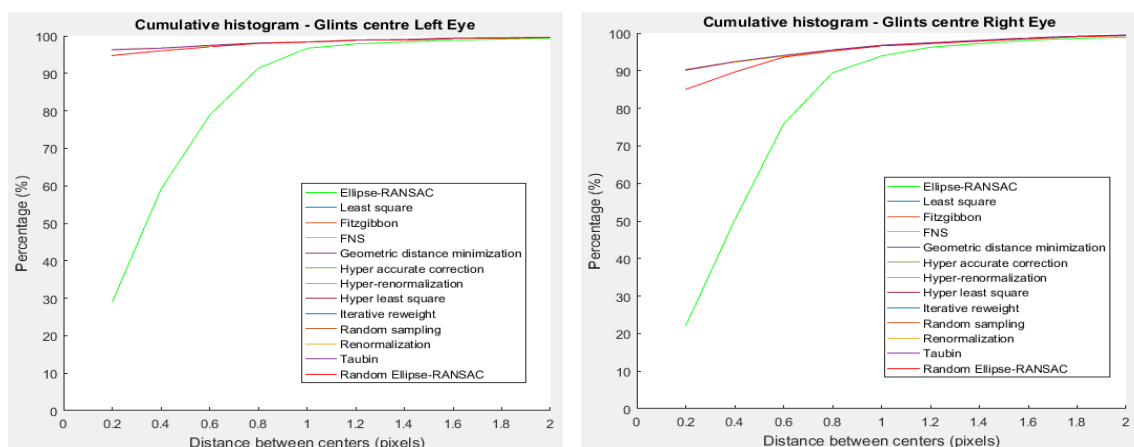


Figura 4.9: Histograma acumulado diferencia centro reflejo respecto a SuperStarBurst de 0 a 2 píxeles.

En este caso los centros de los reflejos obtenidos por los distintos métodos se asemejan más al del *SuperStarburst* que los de las pupilas, ya que en el 95% de las imágenes no hay diferencia entre métodos y la diferencia entre centros es de 0.2 píxeles. También se puede apreciar que en el caso de los reflejos el método de *Ellipse-RANSAC* es el que peor funciona o el que mayor diferencia tiene respecto al *SuperStarBurst*. Para solventar estas diferencias se propuso el algoritmo *Random Ellipse-RANSAC* explicado en la sección 3.2.2. Añadir que en este último método se usa una semilla aleatoria con lo que el ajuste se realiza diez veces y por lo tanto se obtienen diez diferencias de centros para cada imagen con los que se realiza el promedio.



Como conclusión general, los centros obtenidos por los algoritmos explicados en el capítulo anterior se asemejan más a los centros del *SuperStarburst* que al de *Vicomtech*. Además, los centros de los reflejos tienen menor diferencia que el de las pupilas.

### 4.2.3 Error entre centros de pupila y reflejo

Visto que no se puede saber cuál es el método que mejor ajusta, ya que todos ajustan de manera similar, se decidió seleccionar las imágenes problemáticas, es decir, aquellas imágenes en las cuales la diferencia entre centros de pupila del método *Ellipse RANSAC* y el *SuperStarburst* era mayor que 2 píxeles y entre centros de reflejos mayor que 1.5. En estas imágenes se marcó manualmente el centro real para poder analizar el error de cada método. Este subconjunto de imágenes con diferencia de centros de pupilas mayores a 2 píxeles consta de 20 imágenes para el ojo izquierdo y 11 para el derecho. En el caso de los reflejos, 16 para el ojo izquierdo y 28 para el derecho.

Los siguientes histogramas acumulados corresponden al error entre centro de pupilas del ojo izquierdo y derecho. Como se puede apreciar el ojo izquierdo tiene algún caso en el cual el *Ellipse-RANSAC* no lo ajusta bien.

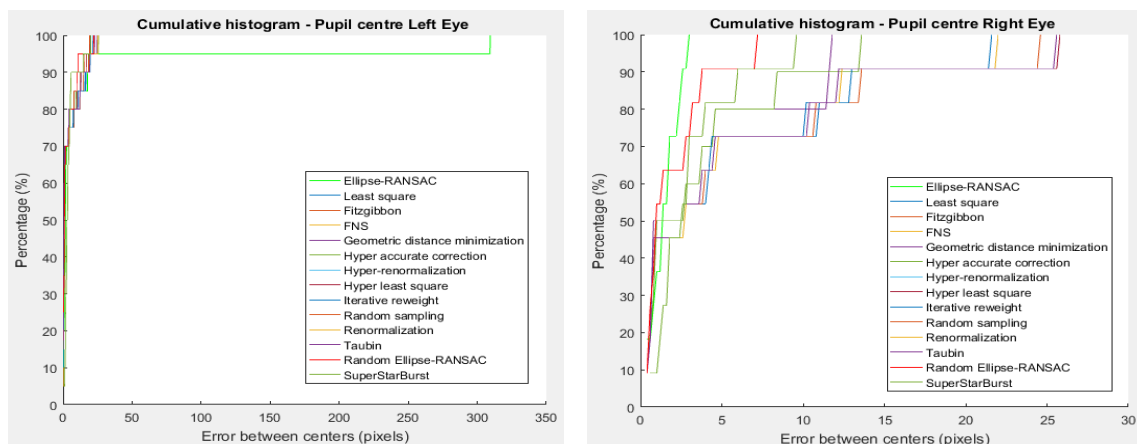


Figura 4.10: Histograma acumulado error entre centros de pupila.

Uno de los casos que el método *Ellipse-RANSAC* no ajusta bien corresponde al usuario cinco. En este caso se puede apreciar como el método *Random Ellipse-RANSAC* mejora la solución.

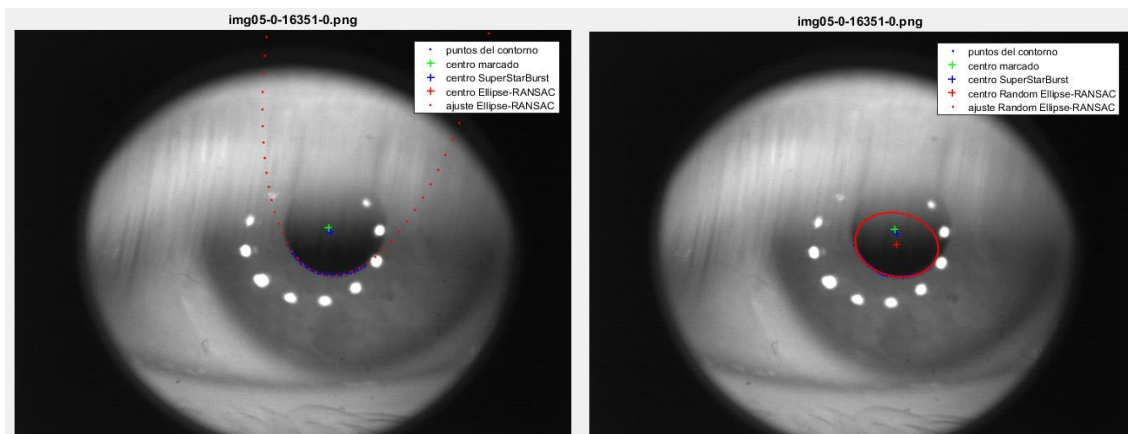


Figura 4.11: Ajuste mediante Ellipse-RANSAC (izda) y Random Ellipse RANSAC (dcha.).

En el histograma del ojo izquierdo se ha dejado fuera la imagen anterior.

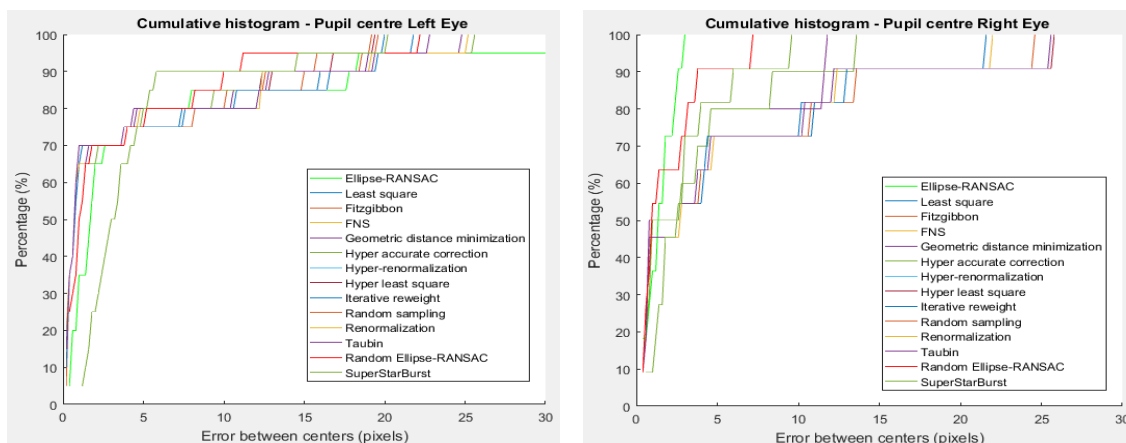


Figura 4.12: Histograma acumulado error entre centros de pupilas de 0 a 30.

En la figura anterior se observa que algunos métodos obtienen el centro de la pupila del ojo derecho con menor error que en el izquierdo aunque no se sabe cuál es el método más preciso.

Después, se ha realizado lo mismo para los centros de los reflejos. Las siguientes imágenes corresponden a los histogramas acumulados de diferencia de centros de reflejos.

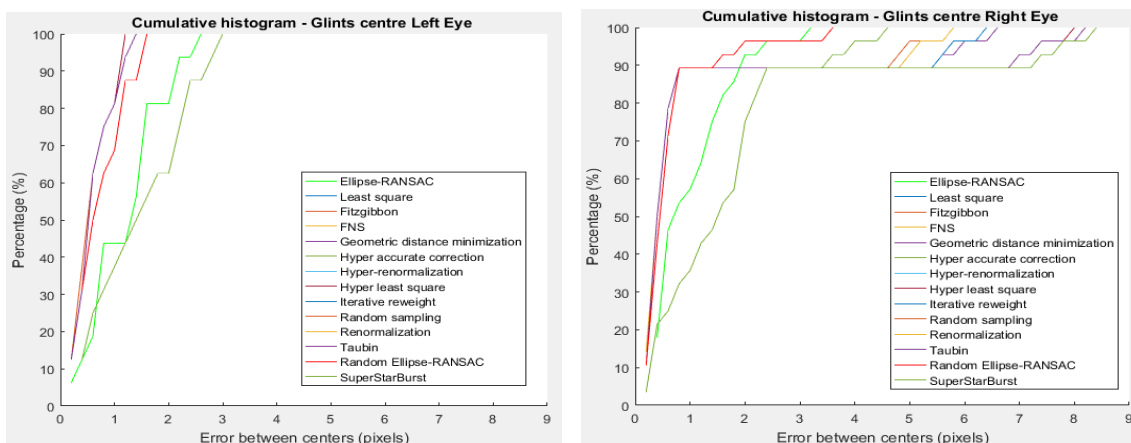


Figura 4.13: Histograma acumulado error entre centros de reflejos.

La siguiente figura muestra el error entre centro de reflejos focalizando el eje horizontal entre 0 y 3 pixeles de error.

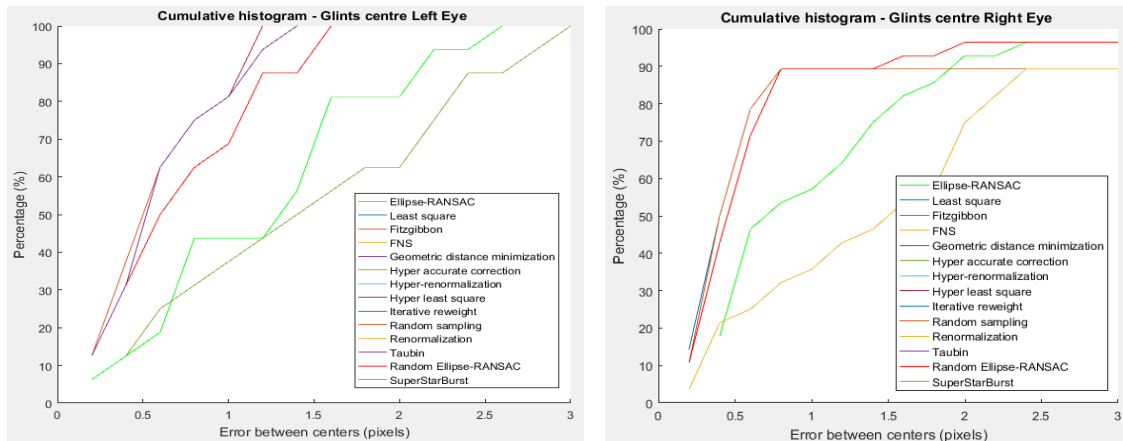


Figura 4.14: Histograma acumulado error entre centros de reflejos de 0 a 3 pixeles.

Se puede observar que al igual que en la comparación con el *SuperStarBurst* el ajuste del centro de los reflejos del ojo izquierdo funciona mejor que el derecho. También, se observa que para el ajuste del centro los reflejos el método *Random Ellipse-RANSAC* funciona mejor que el *Ellipse-RANSAC*. Finalmente, el error de ajuste del centro del reflejo es menor que el de la pupila, con lo que se ajusta mejor al centro del reflejo.

#### 4.2.4 Análisis del error sistemático

El siguiente paso fue analizar que no existía algún tipo de error sistemático y ver si había algún usuario que funcionara peor. Para ello, se utilizó la siguiente ecuación:

$$\bar{v} = \frac{\sum v_i}{N} = 0 \quad (4.11)$$

$v$  es la distancia entre centros, es decir, el vector obtenido desde el centro del *SuperStarBurst* al centro de uno de los métodos explicados anteriormente. Este cálculo se realizó para cada uno de los usuarios diferenciándolos por ojos y por vector de centro de pupilas y de reflejos. Sin embargo, al tener tantos métodos la forma más clara de representar estos valores fue haciendo la media del módulo de todos los usuarios para cada método.

MÉTODOS	Valor medio del módulo del vector centro pupilas		Valor medio del módulo del vector centro reflejos	
	LE	RE	LE	RE
Hyper accurate correction	0,187 ± 0,25	0,236 ± 0,16	0,01 ± 0,09	0,017 ± 0,13
FNS	0,187 ± 0,25	0,236 ± 0,16	0,01 ± 0,09	0,017 ± 0,13
Hyper-renormalization	0,152 ± 0,21	0,206 ± 0,16	0,01 ± 0,09	0,014 ± 0,13
Geometric distance minimization	0,152 ± 0,22	0,176 ± 0,13	0,01 ± 0,09	0,012 ± 0,12
Iterative reweight	0,168 ± 0,20	0,204 ± 0,16	0,01 ± 0,09	0,014 ± 0,13
Random sampling	0,152 ± 0,21	0,206 ± 0,16	0,01 ± 0,09	0,014 ± 0,13
Renormalization	0,158 ± 0,22	0,205 ± 0,16	0,01 ± 0,09	0,014 ± 0,13
HyperLS	0,135 ± 0,24	0,178 ± 0,15	0,008 ± 0,09	0,009 ± 0,12
Least square	0,110 ± 0,23	0,172 ± 0,15	0,008 ± 0,09	0,011 ± 0,12
Ellipse-RANSAC	0,107 ± 0,22	0,076 ± 0,08	0,167 ± 0,09	0,166 ± 0,15
Fitzgibbon	0,105 ± 0,22	0,175 ± 0,16	0,008 ± 0,09	0,01 ± 0,12
Taubin	0,141 ± 0,25	0,178 ± 0,15	0,008 ± 0,09	0,01 ± 0,12

Tabla 4.1: Análisis del error sistemático para los distintos métodos.

Se puede observar que hay un consenso, con lo que no hay diferencias significativas importantes entre métodos y que no hay ningún error sistemático. Por lo tanto, se decidió calcular la bondad de cada método de ajuste en términos del PoR.

### 4.3 CÁLCULO DEL PoR

Una sesión típica de tracking consiste en una fase de calibración y una fase de validación. Dependiendo del experimento que se quiera realizar se define un número u otro de puntos en el panel de calibración y validación.

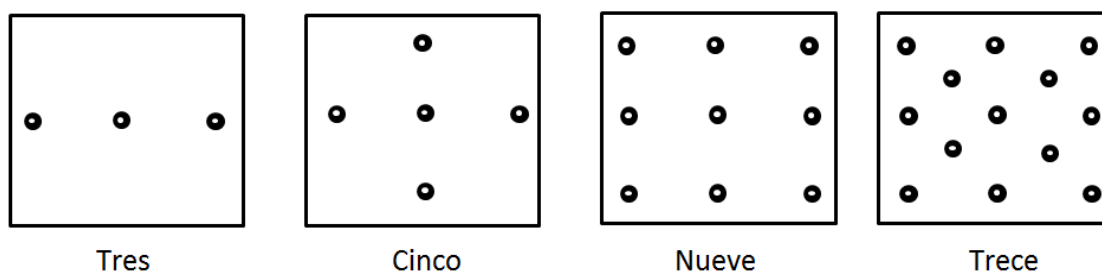


Figura 4.15: Distintos patrones de calibración y validación.

El número de puntos de calibración que se use depende en el tipo de experimento que se esté realizando. En el caso de estudios de lectura de una sola línea se hace uso de tres puntos porque solo desean grabar en el eje x. Los seguidores visuales y los estudios de percepción tienden a utilizar nueve puntos ya que desean registrar lo que los usuarios están viendo en cualquier punto de la pantalla. Finalmente, los estudios de lectura de múltiple línea que consisten en pasajes de

texto, pueden usar nueve puntos, aunque suelen emplear trece para asegurarse de que la calibración en cada línea es suficientemente precisa.

En este caso, al ser un sistema de seguimiento de mirada las calibraciones se realizan a través de nueve puntos. Por lo tanto, para llevar a cabo el cálculo del Point of Regard (PoR) primero se definen las posiciones del estímulo o el patrón de calibración.

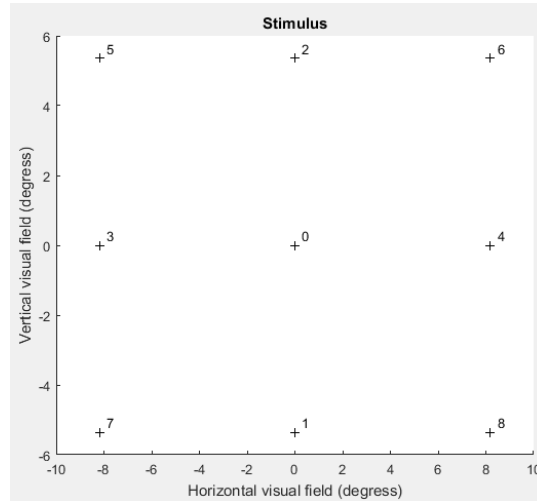


Figura 4.16: Posiciones del estímulo.

Después se obtienen los coeficientes de calibración. Para ello, primero se calcula el vector medio entre pupila ( $p$ ) y reflejo ( $g$ ) para cada imagen de calibración.

$$\begin{aligned} v_x &= p_x - g_x \\ v_y &= p_y - g_y \end{aligned} \quad (4.12)$$

El vector medio se obtiene mediante la función “trimmean”. Para ello se fija un porcentaje y se ordenan los vectores en  $x$  e  $y$ . Después, dependiendo del porcentaje establecido se elimina un cierto número de valores de ambos extremos del vector de cada fijación o punto de calibración que se consideran posibles outliers. Por ejemplo, si el porcentaje se fija en un 50% y el número de vectores medios por fijación es de 20 se ordena el vector de menor a mayor, se eliminan 5 números del extremo inferior, 5 del superior y se calcula el promedio con los valores restantes.

Con un valor medio para cada fijación y utilizando la ecuación de interpolación de segundo orden, se obtiene la siguiente matriz de calibración:

$$VD_{cal} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ x_0^2 & x_1^2 & x_2^2 & x_3^2 & x_4^2 & x_5^2 & x_6^2 & x_7^2 & x_8^2 \\ y_0^2 & y_1^2 & y_2^2 & y_3^2 & y_4^2 & y_5^2 & y_6^2 & y_7^2 & y_8^2 \\ x_0 y_0 & x_1 y_1 & x_2 y_2 & x_3 y_3 & x_4 y_4 & x_5 y_5 & x_6 y_6 & x_7 y_7 & x_8 y_8 \\ y_0 & y_1 & y_2 & y_3 & y_4 & y_5 & y_6 & y_7 & y_8 \\ x_0 & x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 & x_8 \end{pmatrix} \quad (4.13)$$

Cada columna corresponde a una fijación. La primera columna se calcula mediante el vector medio del primer estímulo (el centro), la segunda al vector medio del segundo estímulo (en el centro abajo) y así sucesivamente.

Después se calculan los coeficientes de calibración, multiplicando las coordenadas de las fijaciones por la inversa de la matriz de calibración.

$$C = target * pinv(VD\_cal) \quad (4.14)$$

Una vez obtenidos los coeficientes de calibración se empieza con la validación. Para ello se calculan los vectores entre el centro de la pupila y los reflejos en x e y de las imágenes de validación. Después, se obtiene la matriz de validación para cada fijación, es decir, no se realiza el trimmean porque a la hora de evaluar un método no sería adecuado quedarnos con un solo valor del PoR por cada fijación. Finalmente, multiplicando las matrices de validación obtenidas de cada fijación por el coeficiente de calibración se calcula el PoR de las imágenes de cada fijación.

$$PoR = C * VD\_val \quad (4.15)$$

Una vez obtenidos los PoR de las imágenes y dibujados encima de cada estímulo se decidió realizar el PCA a través de la matriz de covarianza y así dibujar las elipses que representan la desviación y media de los PoR por fijación. En primer lugar, se realizó para cada usuario con los distintos métodos pero al tener muchos algoritmos para comparar se decidió realizar para los distintos métodos separando el ojo izquierdo y derecho y tomando a los 7 usuarios como un único usuario. A continuación, se representan las figuras que muestran la distribución del PoR por cada fijación y el valor medio:

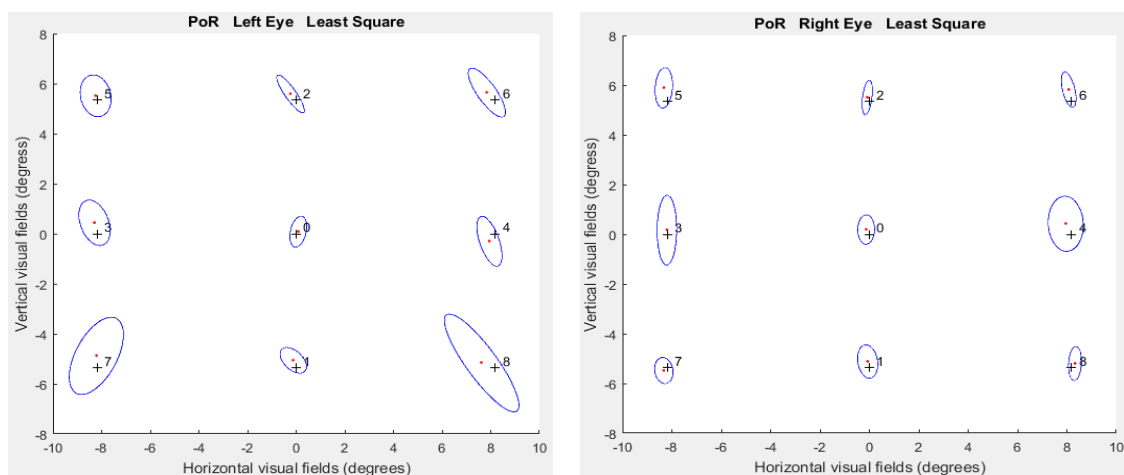


Figura 4.17: PoR ojo izquierdo y ojo derecho Least Square.

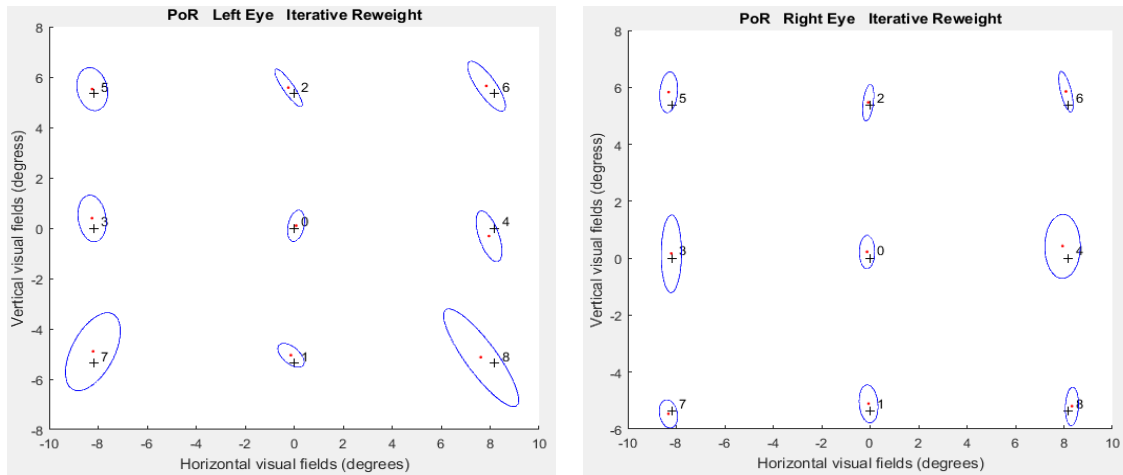


Figura 4.18: PoR ojo izquierdo y ojo derecho Iterative Reweight.

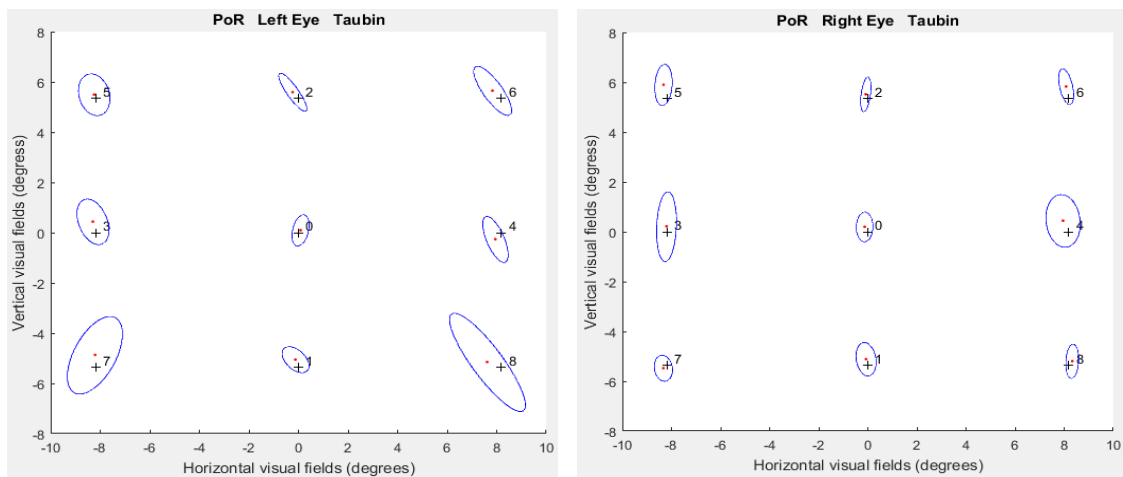


Figura 4.19: PoR ojo izquierdo y ojo derecho Taubin.

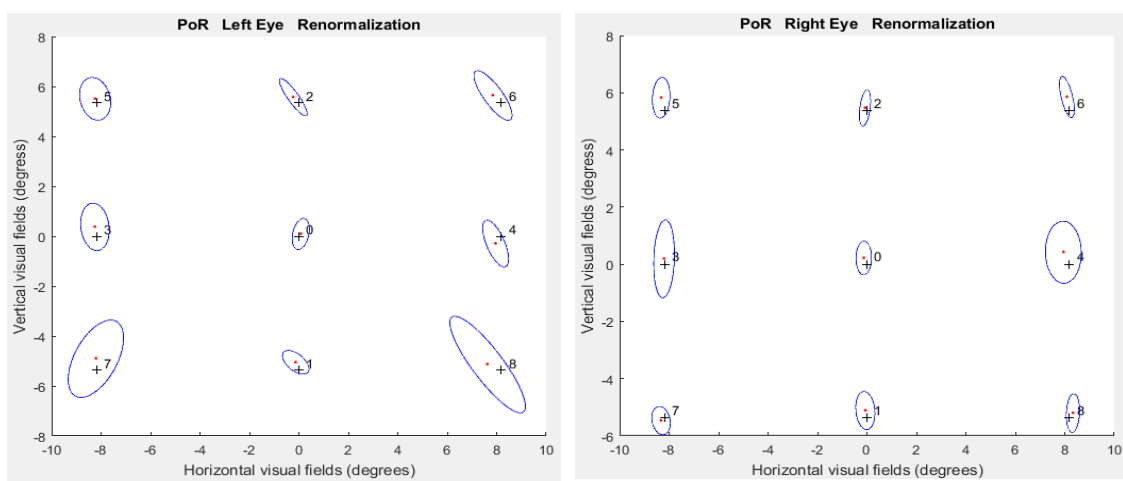


Figura 4.20: PoR ojo izquierdo y ojo derecho Renormalization.

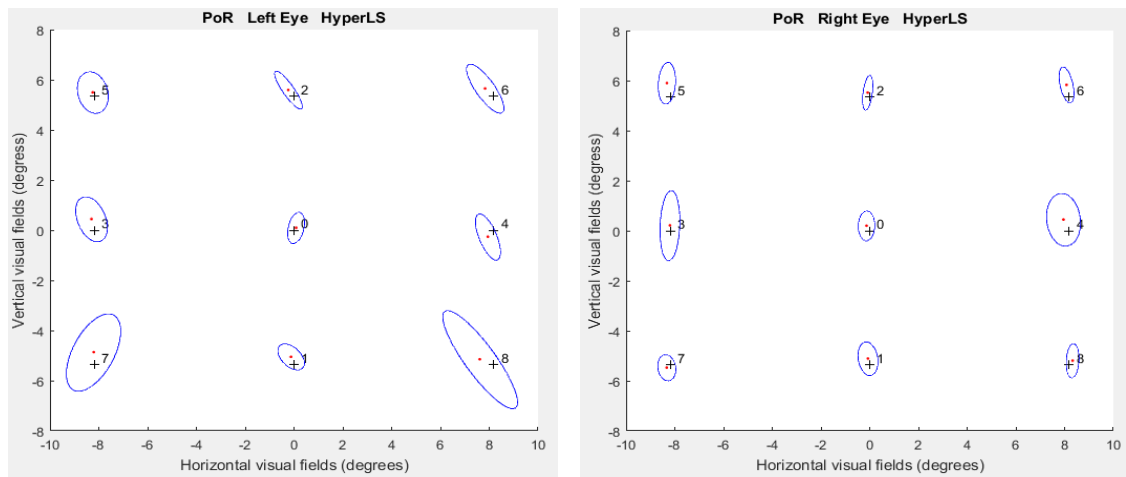


Figura 4.21: PoR ojo izquierdo y ojo derecho HyperLS.

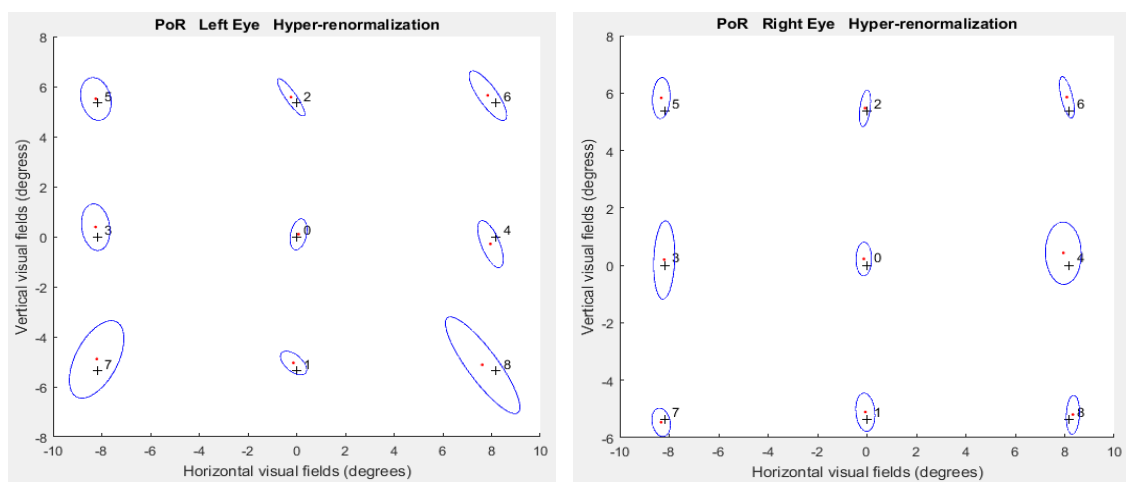


Figura 4.22: PoR ojo izquierdo y ojo derecho Hyper-renormalization.

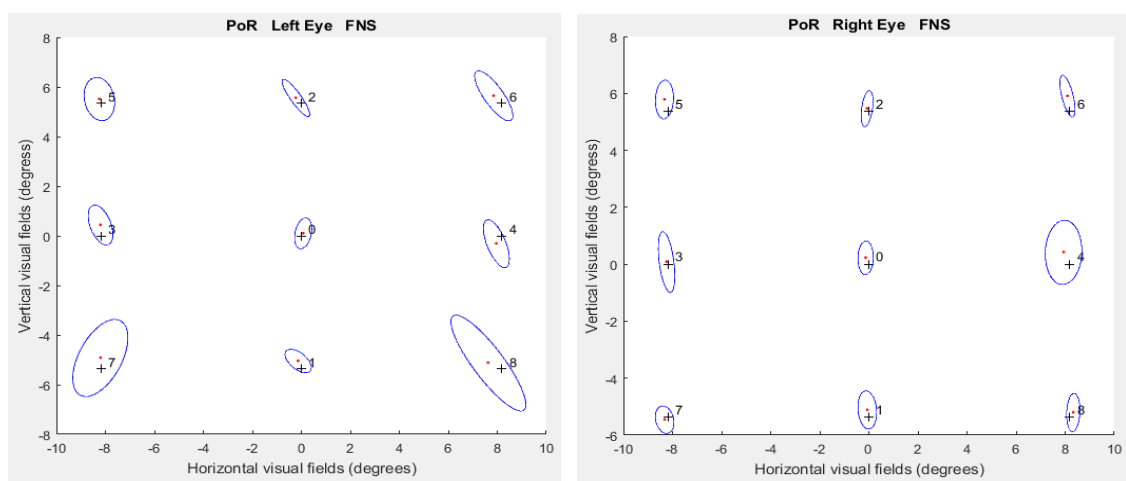


Figura 4.23: PoR ojo izquierdo y ojo derecho FNS.



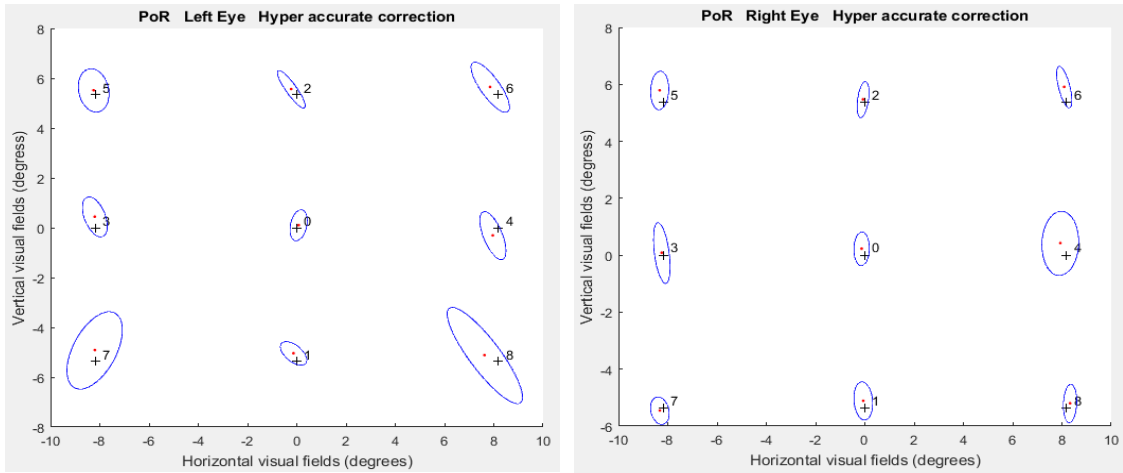


Figura 4.24: PoR ojo izquierdo y ojo derecho Hyper accurate correction.

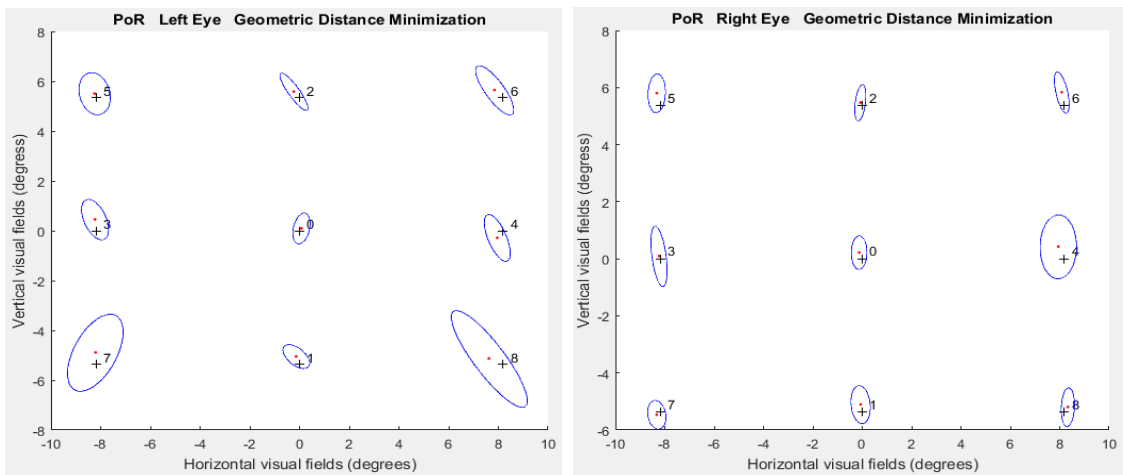


Figura 4.25: PoR ojo izquierdo y ojo derecho Geometric Distance Minimization.

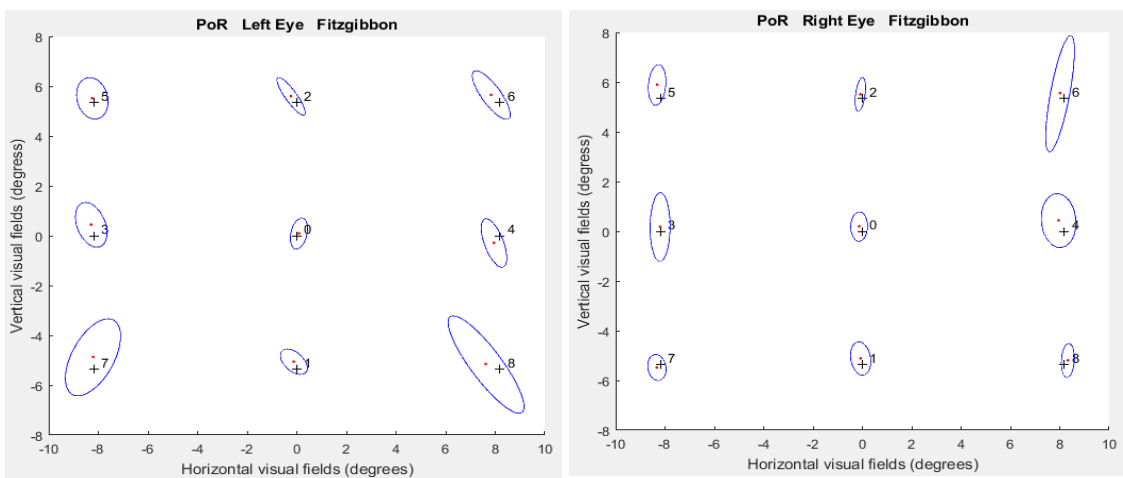


Figura 4.26: PoR ojo izquierdo y ojo derecho Fitzgibbon.

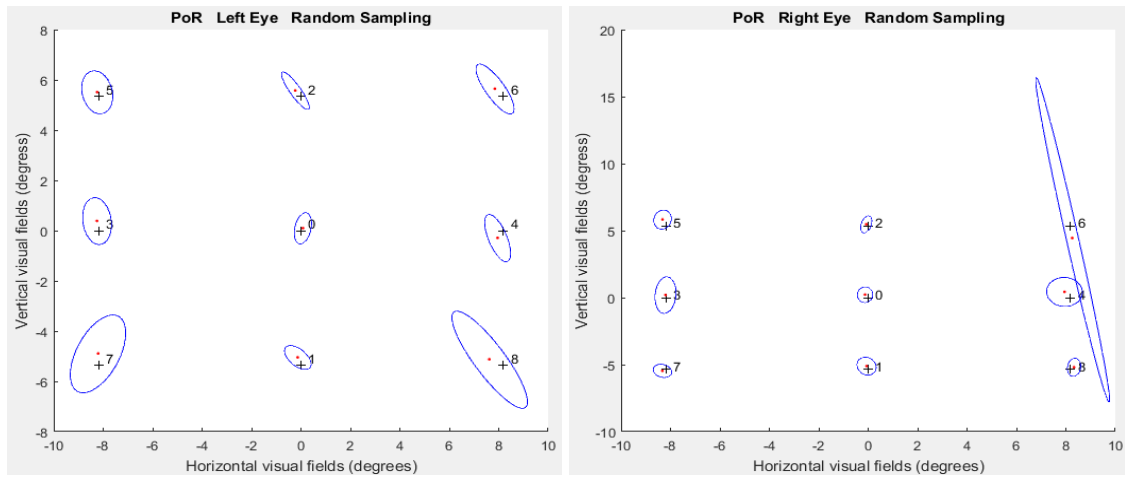


Figura 4.27: PoR ojo izquierdo y ojo derecho Random Sampling.

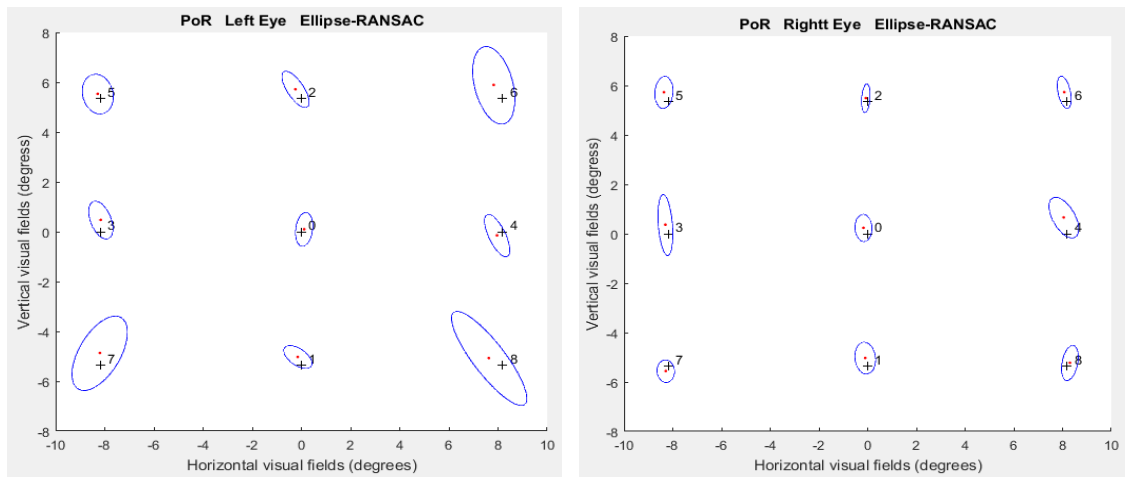


Figura 4.28: PoR ojo izquierdo y ojo derecho Ellipse-RANSAC.

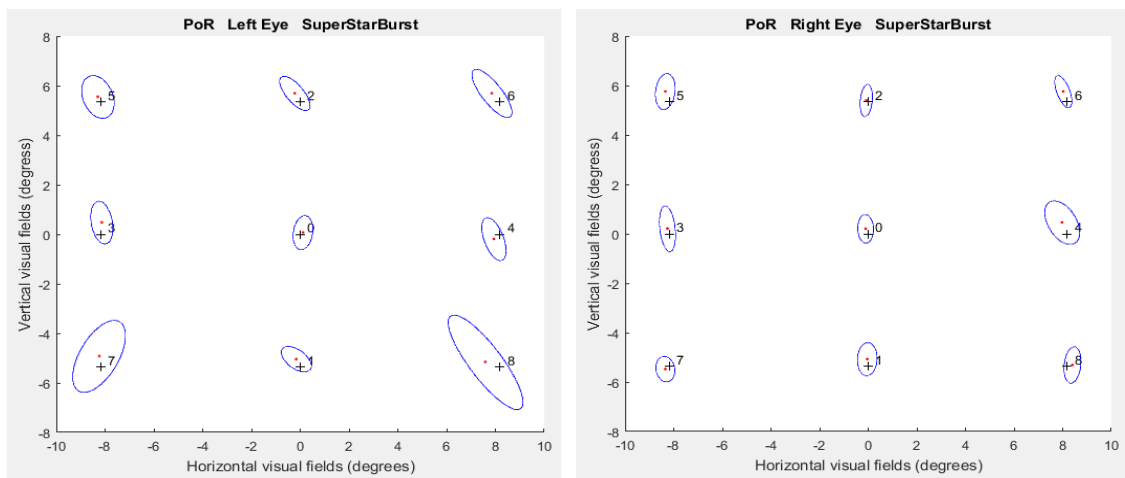


Figura 4.29: PoR ojo izquierdo y ojo derecho SuperStarBurst.

Finalmente se agrupó en una misma gráfica el método *Random Ellipse-RANSAC* con los cambios de umbral para ver qué umbral se ajustaba mejor.

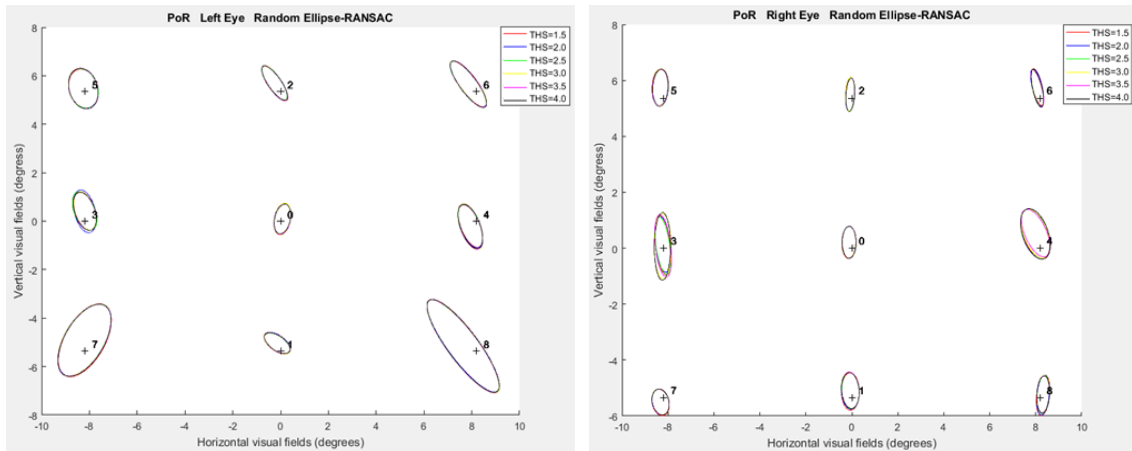


Figura 4.30: PoR Random Ellipse-RANSAC con diferentes umbrales.

Después, se decidió unificar los dos ojos y dibujar la distribución de los PoR.

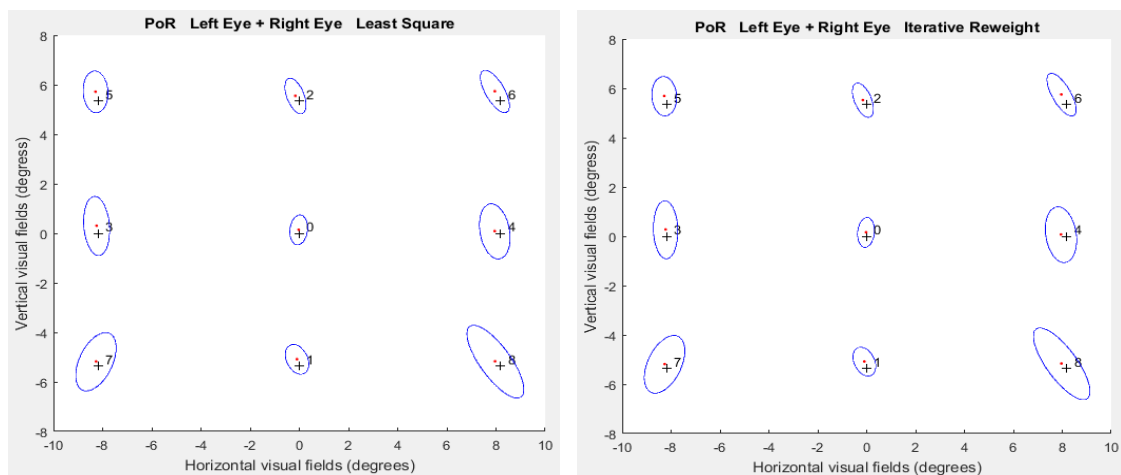


Figura 4.32: PoR Least Square e Iterative Reweight.

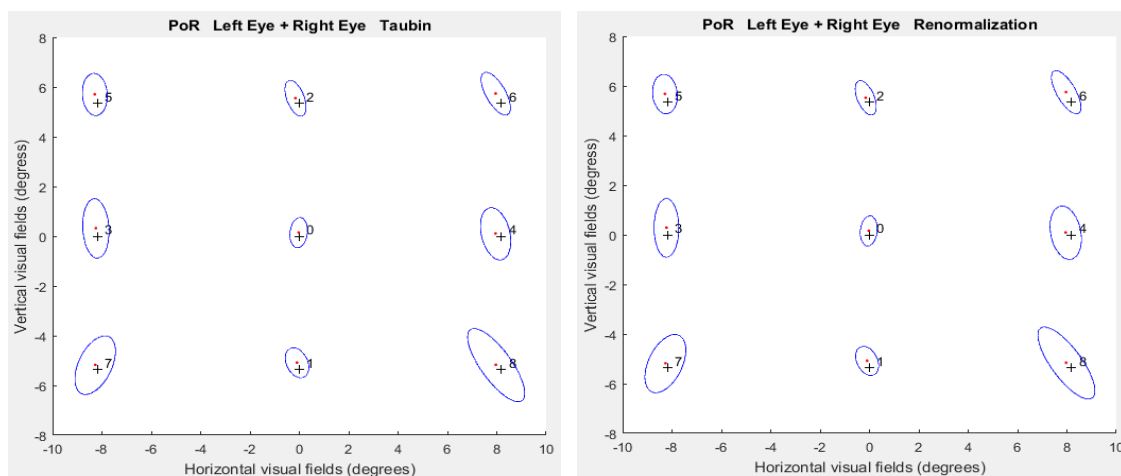


Figura 4.31: PoR Taubin y Renormalization.

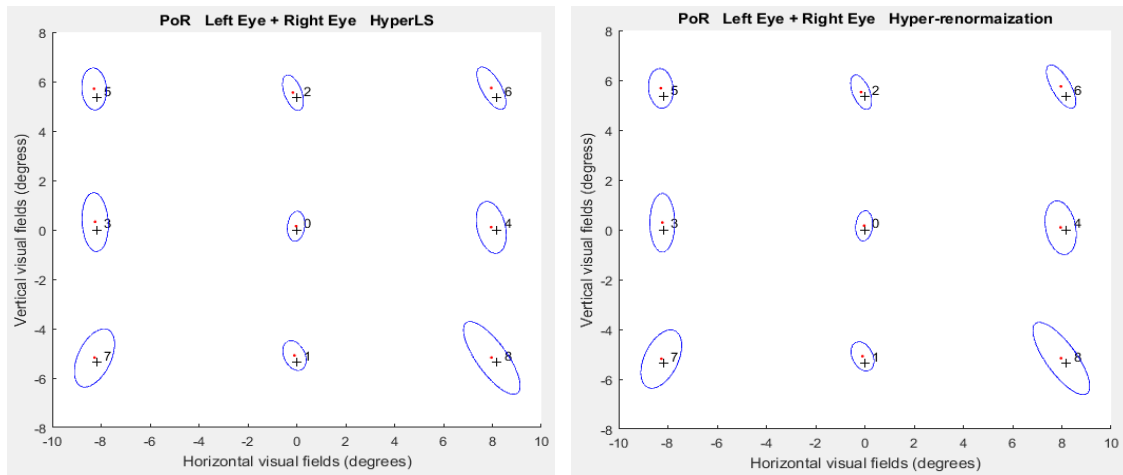


Figura 4.33: PoR HyperLS y Hyper-renormalization.

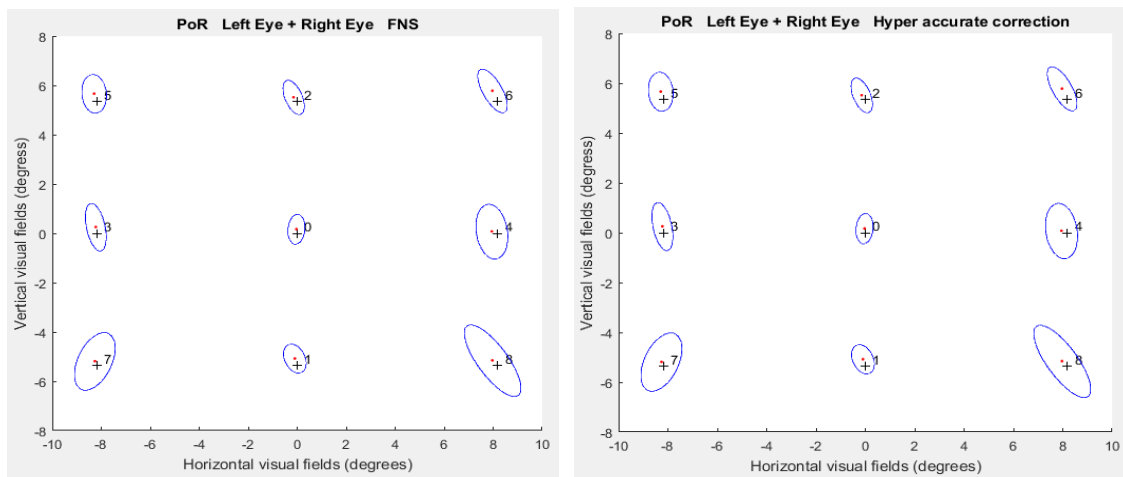


Figura 4.34: PoR FNS y Hyperaccurate correction.

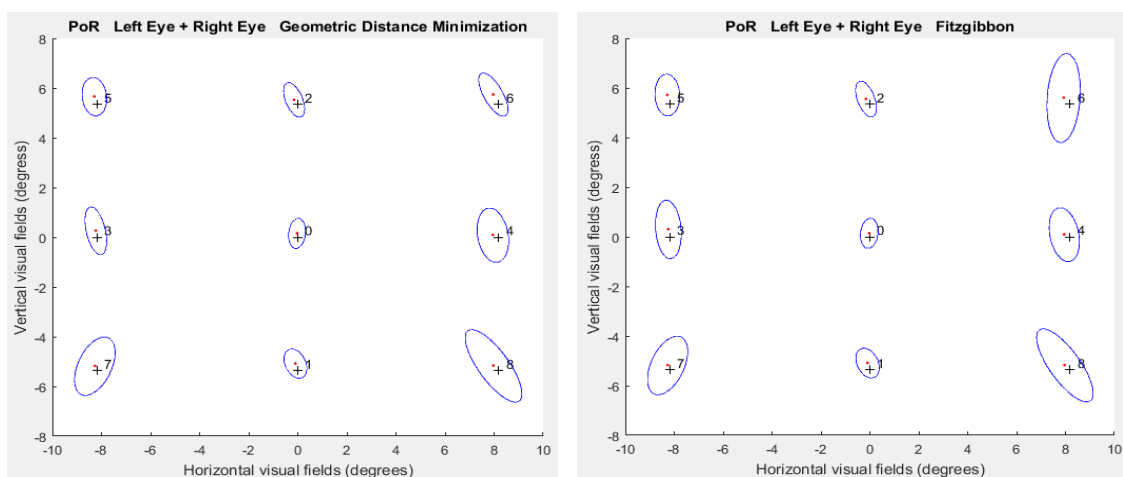


Figura 4.35: PoR Geometric Distance Minimization y Fitzgibbon.

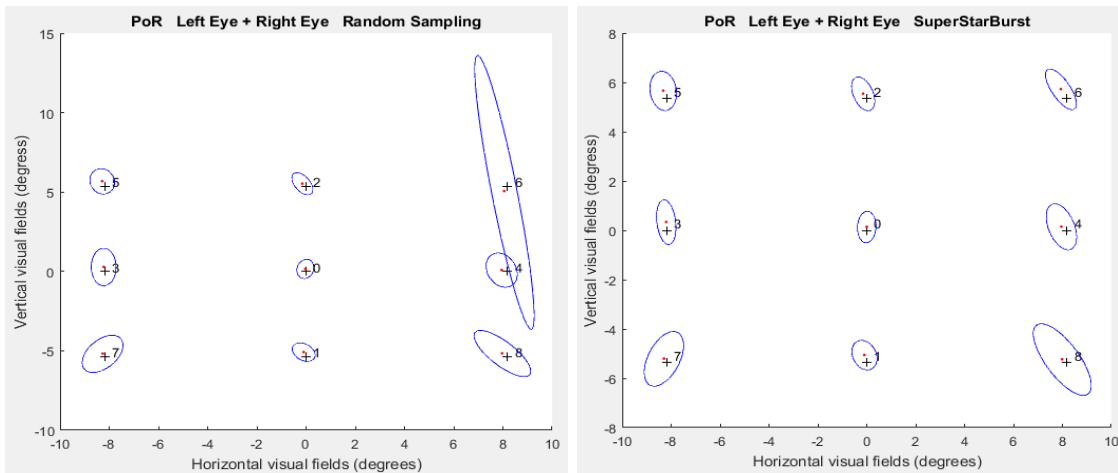


Figura 4.36: PoR Random Sampling y SuperStarBurst.

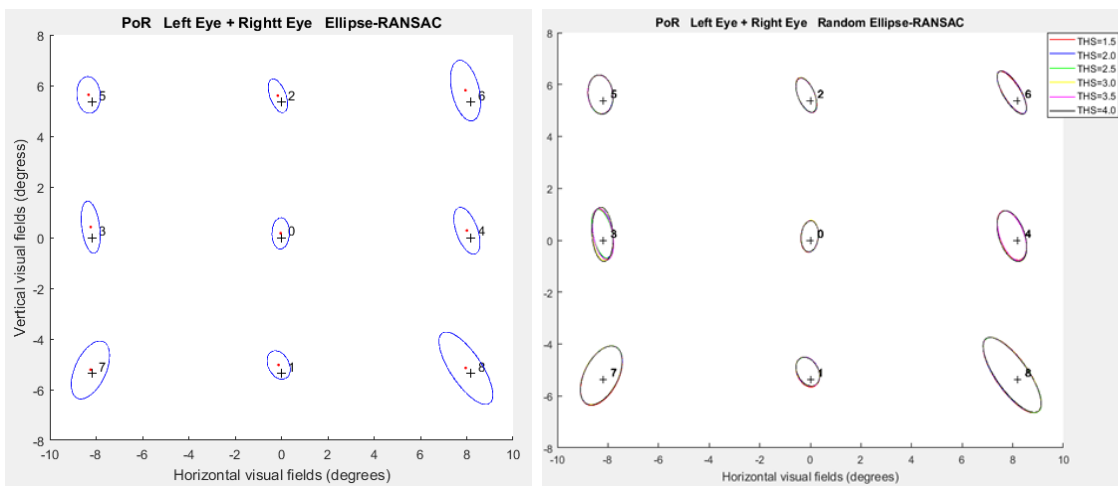


Figura 4.37: PoR Ellipse-RANSAC y Random Ellipse RANSAC.

Observando los resultados del método de *Random Sampling*, se comprobó que en la fijación 6 del ojo derecho había una imagen que su centro no se había detectado correctamente. Para mejorar estos problemas del *Random Sampling* se probó un nuevo algoritmo que consistía en calcular la matriz  $\mathbf{M}$  con todos los inliers en vez de con 5 puntos aleatorios, es decir, con el mismo proceso que se utiliza en el método *Random Ellipse-RANSAC*. Sin embargo, se observó en cuantas imágenes se realizaba el *Random Sampling* y solo fue en una, todas la demás se ajustaban mediante *Hyper-renormalization*. Por lo tanto, en los resultados no se ha llegado a implementar como posible método ya que el tiempo de cálculo también aumentaba.

En la siguiente tabla se resume los resultados obtenidos en las gráficas de ojos separados:

MÉTODOS	Ojo=LE	Ojo=RE
Random Ellipse-RANSAC d=1.5	0,8207± 0,9416	0,7489 ± 0,7867
Random Ellipse-RANSAC d=2.0	0,8201 ± 0,952	0,7458 ± 0,7892
Random Ellipse-RANSAC d=2.5	0,8208 ± 0,9461	0,7448 ± 0,7876
Random Ellipse-RANSAC d=3.0	0,824 ± 0,9436	0,7482 ± 0,7975
<b>Random Ellipse-RANSAC d=3.5</b>	<b>0,8179 ± 0,9397</b>	<b>0,7431 ± 0,7886</b>
Random Ellipse-RANSAC d=4.0	0,8173 ± 0,9382	0,7442 ± 0,7966
SuperStarBurst	0,859 ± 0,9424	0,7855 ± 0,7727
Hyper accurate correction	0,8489± 0,9497	0,774 ± 0,8105
FNS	0,8489± 0,9497	0,7741 ± 0,8105
Hyper-renormalization	0,8486 ± 0,96	0,7797± 0,8326
Geometric Distance Minimization	0,8433 ± 0,9506	0,7708± 0,8067
Iterative reweight	0,8527 ± 0,9671	0,7822± 0,8377
Random sampling	0,8486 ± 0,96	0,8645± 1,6968
Renormalization	0,8488 ± 0,9611	0,7798± 0,8328
HyperLS	0,8537± 0,9613	0,792± 0,8393
Leastsquare	0,8579 ± 0,9698	0,7938 ± 0,8444
Fitzgibbon	0,8565 ± 0,9654	0,8076 ± 0,9423
Taubin	0,8536 ± 0,9612	0,8179 ± 1,0559

Tabla 4.2: Error medio de los PoR para el ojo izquierdo y derecho.

La siguiente tabla corresponde a los gráficos de los ojos conjuntos.

MÉTODOS	Ojo=LE+RE
Random Ellipse-RANSAC d=1.5	0,7489 ± 0,7867
Random Ellipse-RANSAC d=2.0	0,7458± 0,7892
Random Ellipse-RANSAC d=2.5	0,7448± 0,7876
Random Ellipse-RANSAC d=3.0	0,7482± 0,7975
<b>Random Ellipse-RANSAC d=3.5</b>	<b>0,7431± 0,7886</b>
Random Ellipse-RANSAC d=4.0	0,7442± 0,7966
SuperStarBurst	0,7855± 0,7727
Hyper accurate correction	0,774± 0,8105
FNS	0,7741 ± 0,8105
Hyper-renormalization	0,7797 ± 0,8326
Geometric Distance Minimization	0,7708 ± 0,8067
Iterative reweight	0,7822 ± 0,8377
Random sampling	0,8645 ± 1,6968
Renormalization	0,7798 ± 0,8328
Hyperls	0,792 ± 0,8393
Leastsquare	0,7938 ± 0,8444
Fitzgibbon	0,8076 ± 0,9423
Taubin	0,8179 ± 1,0559

Tabla 4.3: Error medio de los PoR unificando los dos ojos.

En las gráficas se puede apreciar que en algunas fijaciones hay mayor error que en otras. Esto se ha representado en la siguiente tabla mostrando el error máximo de cada fijación.

OJO = LE	Max0	Max1	Max2	Max3	Max4	Max5	Max6	Max7	Max8
Random RANSAC d=1.5	4,583	1,870	2,923	2,95	3,583	2,248	3,660	8,516	13,33
Random RANSAC d=2.0	4,592	1,834	2,449	8,373	4,109	2,316	3,609	8,496	13,33
Random RANSAC d=2.5	4,584	1,840	2,671	5,069	4,335	2,298	3,649	8,485	13,34
Random RANSAC d=3.0	4,583	1,851	2,751	2,880	3,974	2,338	3,589	8,482	13,32
Random RANSAC d=3.5	4,376	1,833	2,309	2,691	4,143	2,323	3,6432	8,476	13,33
Random RANSAC d=4.0	4,292	1,838	2,322	2,628	4,585	2,324	3,615	8,495	13,33
SuperStarBurst	4,426	2,061	2,373	2,954	3,056	2,647	3,600	8,198	13,52
Hyper accurate correction	4,13	1,806	3,450	2,448	2,755	2,282	3,560	8,460	13,31
FNS	4,13	1,806	3,45	2,448	2,76	2,282	3,56	8,46	13,31
Hyper-renormalization	4,097	1,804	3,35	3,784	2,527	2,286	3,559	8,460	13,31
Geometric Distance Minimization	4,081	1,803	3,322	2,57	2,494	2,284	3,556	8,457	13,31
Iterative reweight	4,107	1,846	3,414	3,627	3,476	2,288	3,556	8,462	13,31
Random sampling	4,097	1,804	3,35	3,784	2,527	2,286	3,559	8,46	13,31
Renormalization	4,098	1,804	3,35	3,974	2,551	2,287	3,558	8,46	13,31
Hyperls	4,051	1,861	3,304	3,149	2,348	2,294	3,558	8,457	13,31
Least Square	4,048	1,801	3,38	3,092	3,583	2,294	3,556	8.458	13,31
Fitzgibbon	4,065	1,802	3,373	2,994	3,102	2,291	3,551	8.462	13,31
Taubin	4,05	1,846	3,316	3,113	2,351	2,294	3,556	8.457	13,31
Ellipse-RANSAC	4,589	2,138	2,387	2,577	2,804	2,133	9,957	8.543	13,21

Tabla 4.4: Error máximo en cada fijación para el ojo izquierdo.

En este caso se puede apreciar como las fijaciones 7 y 8 son las que más se dispersan. Se ha analizado qué imágenes son los que dan este error tan grande y corresponden al usuario 6 en el momento en que el ojo se está moviendo de la fijación 7 a la 8. Por lo tanto, no significa que el PoR esté mal calculado si no que al usuario no le dio tiempo de moverse o que existe un problema de sincronía entre otras posibles razones.

OJO = RE	Max0	Max1	Max2	Max3	Max4	Max5	Max6	Max7	Max8
Random RANSAC d=1.5	2,030	1,811	1,459	6,554	4,764	1,922	2,289	2,003	2,292
Random RANSAC d=2.0	2,091	2,058	1,718	6,516	4,872	2,005	2,526	2,008	2,292
Random RANSAC d=2.5	2,001	1,806	1,601	6,730	4,758	2,454	2,564	2,011	2,297
Random RANSAC d=3.0	2,104	1,947	1,752	7,360	4,674	1,966	2,640	2,011	2,314
Random RANSAC d=3.5	2,095	1,873	1,586	7,971	4,469	1,891	2,590	2,003	2,302
Random RANSAC d=4.0	2,128	1,992	1,445	7,497	4,69	2,647	2,583	1,989	2,308
SuperStarBurst	1,448	1,736	2,299	3,555	4,356	2,227	2,052	2,117	2,303
Hyper accurate correction	1,949	1,71	2,025	5,811	6,303	1,881	2,613	1,986	2,263
FNS	1,948	1,71	2,025	5,811	6,304	1,881	2,613	1,986	2,263
Hyper-renormalization	1,942	1,710	1,899	7,386	5,824	2,441	2,570	1,983	2,262
Geometric Distance Minimization	1,943	1,711	1,857	5,654	6,321	1,877	2,614	1,983	2,262
Iterative reweight	1,875	1,709	2,064	7,136	6,308	2,602	2,428	1,983	2,263
Random sampling	1,942	1,71	1,899	7,386	5,824	2,441	96,232	1,983	2,262
Renormalization	1,936	1,71	1,907	7,294	5,863	2,409	2,578	1,984	2,262
Hyperls	1,927	1,974	1,715	7,71	5,28	3,816	2,554	1,984	2,26
Least Square	1,86	1,965	1,867	7,498	5,973	3,671	2,406	1,984	2,26
Fitzgibbon	1,85	1,969	1,785	7,323	5,628	3,786	17,735	1,984	2,26
Taubin	1,921	1,973	1,69	7,659	5,323	3,779	2,561	1,984	2,26
Ellipse-RANSAC	1,509	1,992	1,538	7,316	4,389	2,086	1,733	1,65	2,193

Tabla 4.5: Error máxima en cada fijación para el ojo derecho.

En este caso la fijación que mayor dispersión tiene es el número 3. Sin embargo, el error de las demás fijaciones es admisible, es decir, no hay usuarios que miren a otro punto distinto en alguna de las fijaciones.



Estos resultados se han representado gráficamente mediante un diagrama de cajas. Primero se ha calculado para el ojo izquierdo y después para el ojo derecho.

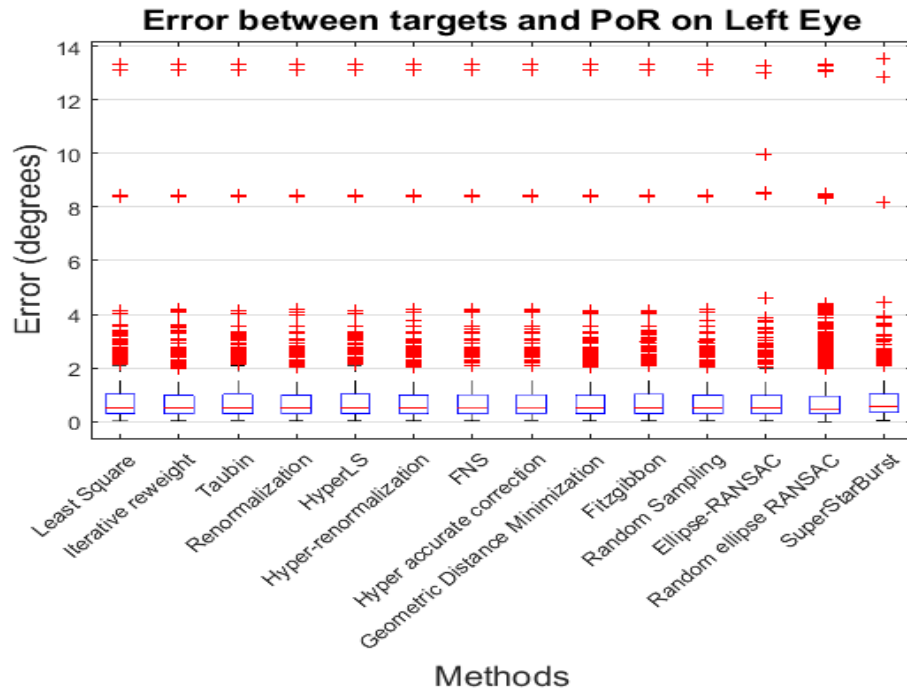


Figura 4.38: Boxplot error PoR-Target ojo izquierdo.

En la gráfica anterior, los outliers de mayor valor corresponden a las imágenes en las cuales la mirada se encuentra en el desplazamiento de la fijación 7 a la 8 como se ha explicado anteriormente. En el siguiente diagrama se demuestra lo dicho con el ejemplo de dos métodos.

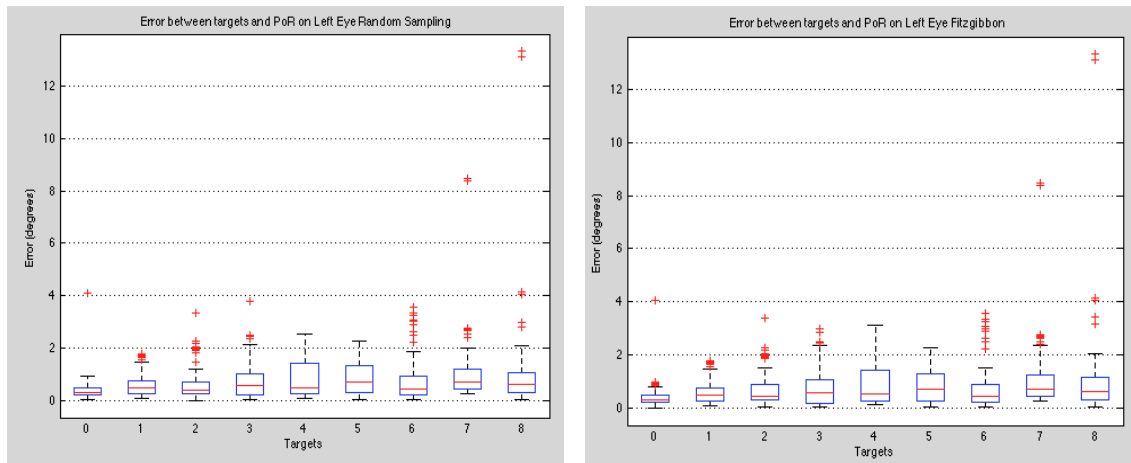


Figura 4.39: Boxplot por fijaciones para los métodos Random Sampling y Fitzgibbon ojo izq.

En la gráfica, se puede observar que hay dos outliers que destacan respecto a los demás.

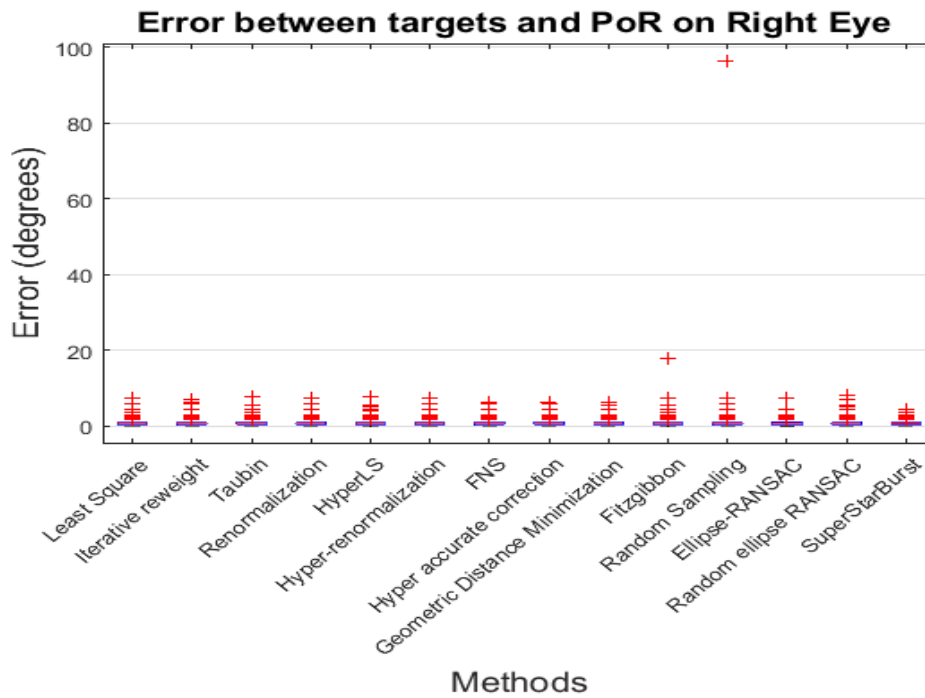


Figura 4.40: Boxplot error PoR-Target ojo derecho.

Visto que con los diagramas anteriores no se han obtenido resultados precisos, se decidió eliminar los outliers entre fijación y PoR que cumplían la siguiente condición, ya que se asume que corresponden a imágenes para las que el usuario no está mirando a la fijación correcta, en vez de un mal cálculo de centros:

$$error > \mu \pm 2\sigma \tag{4.16}$$

Eliminando estos valores, en las siguientes imágenes se puede observar como mejoran estos nuevos valores de PoR respecto a los anteriores:

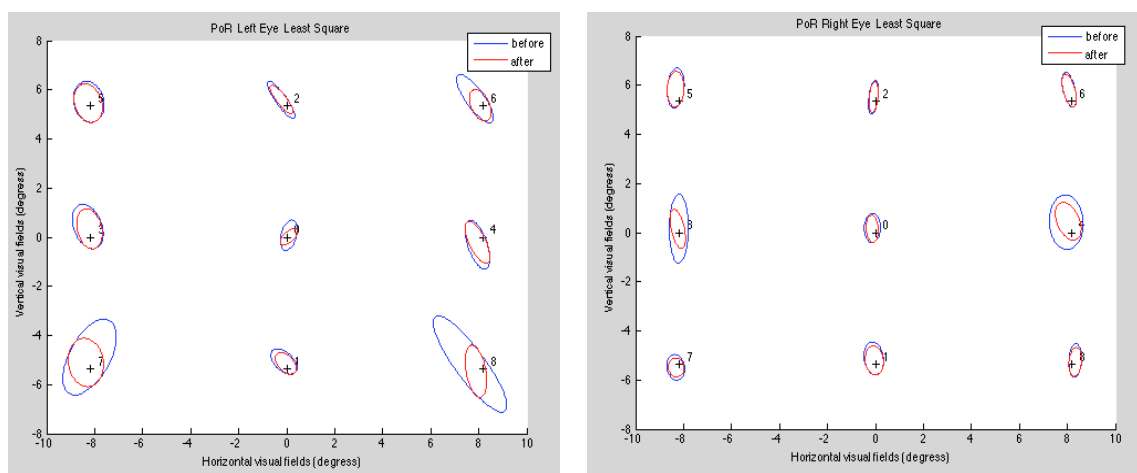


Figura 4.41: Distribución PoR Least Square.

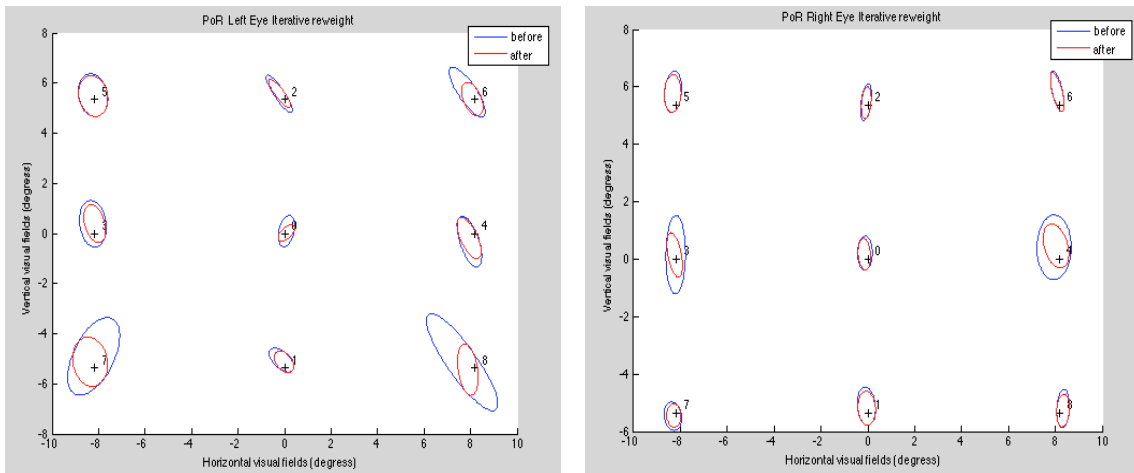


Figura 4.42: Distribución PoR Iterative reweight.

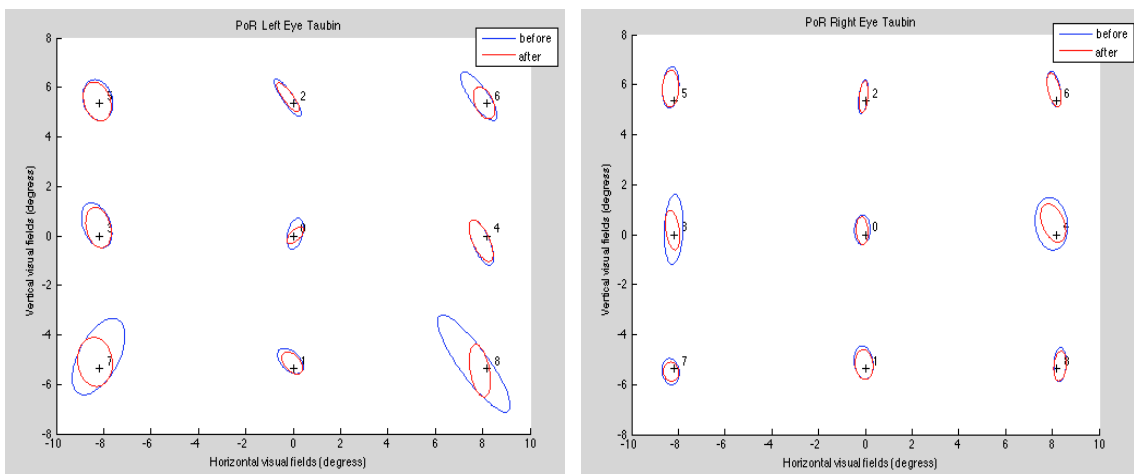


Figura 4.43: Distribución PoR Taubin.

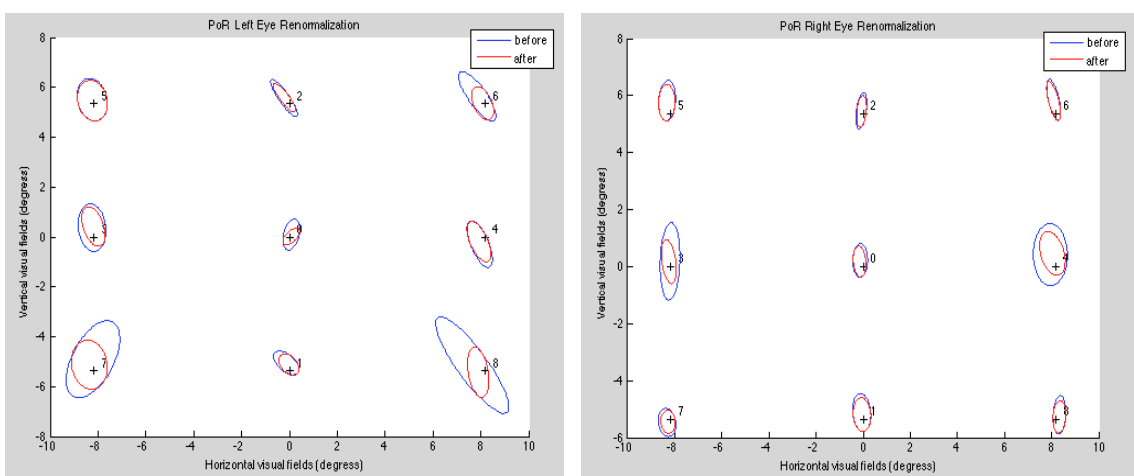


Figura 4.44: Distribución PoR Renormalization.

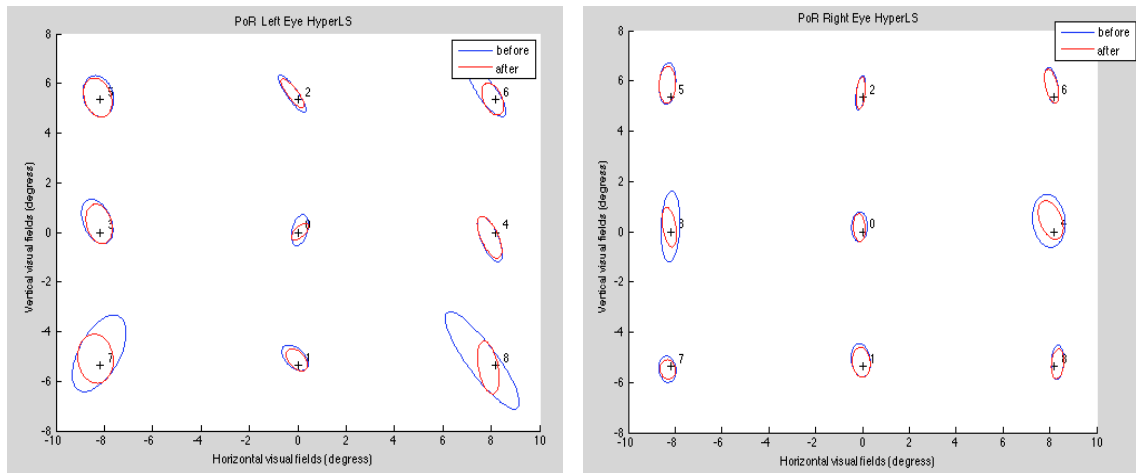


Figura 4.45: Distribución PoR HyperLS.

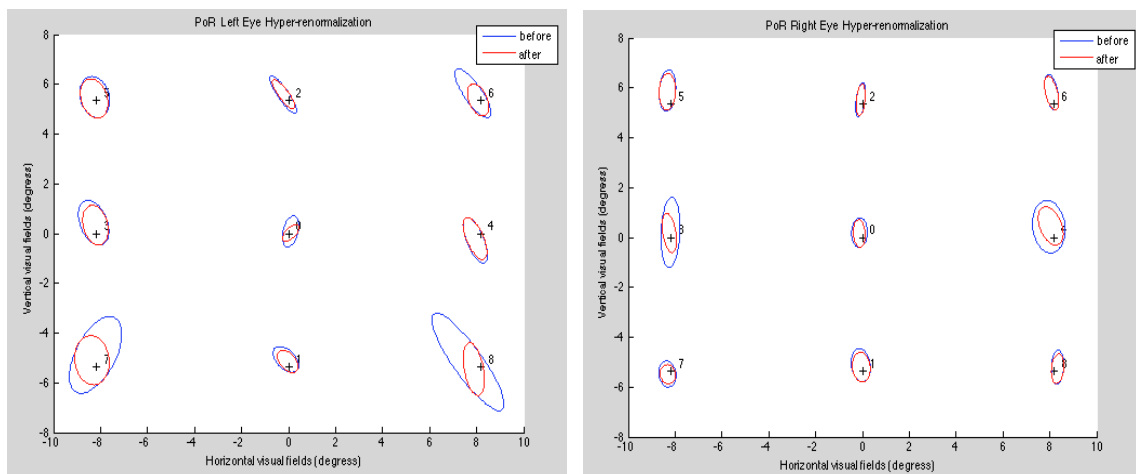


Figura 4.46: Distribución PoR Hyper-renormalization.

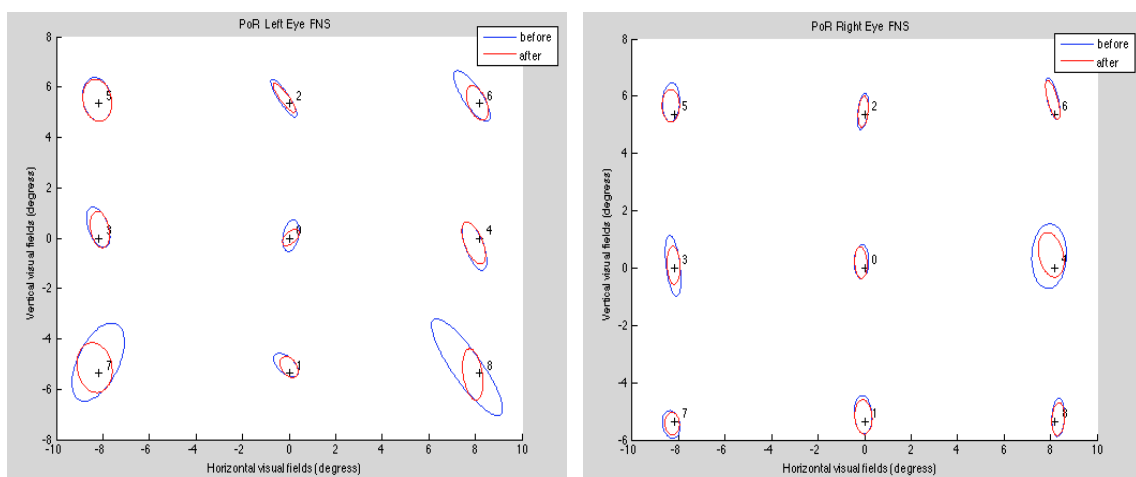


Figura 4.47: Distribución PoR FNS.

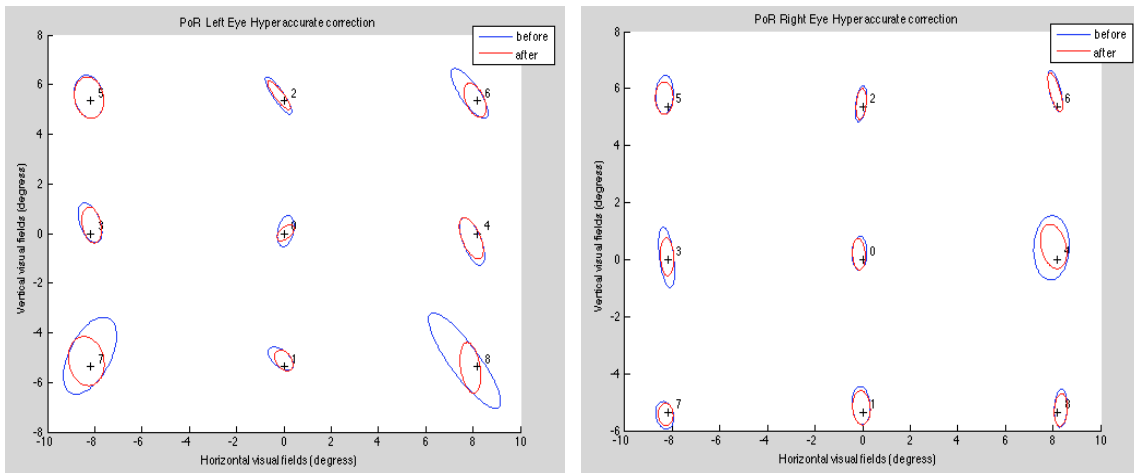


Figura 4.48: Distribución PoR Hyper accurate correction.

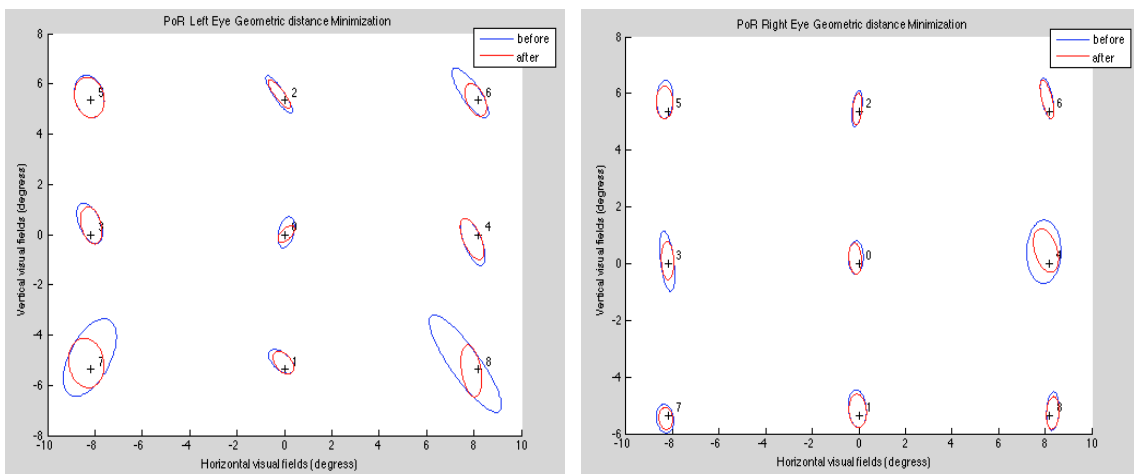


Figura 4.49: Distribución PoR Geometric Distance Minimization.

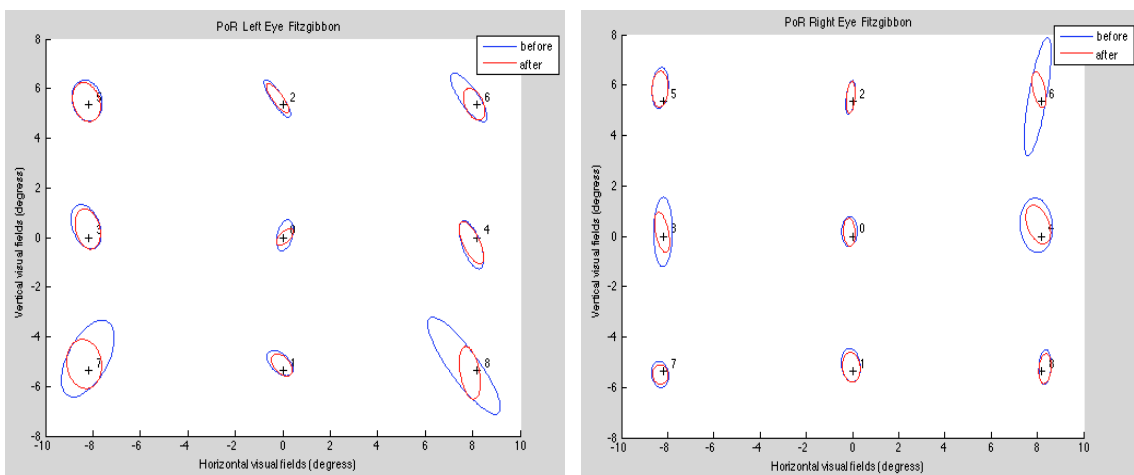


Figura 4.50: Distribución PoR Fitzgibbon.

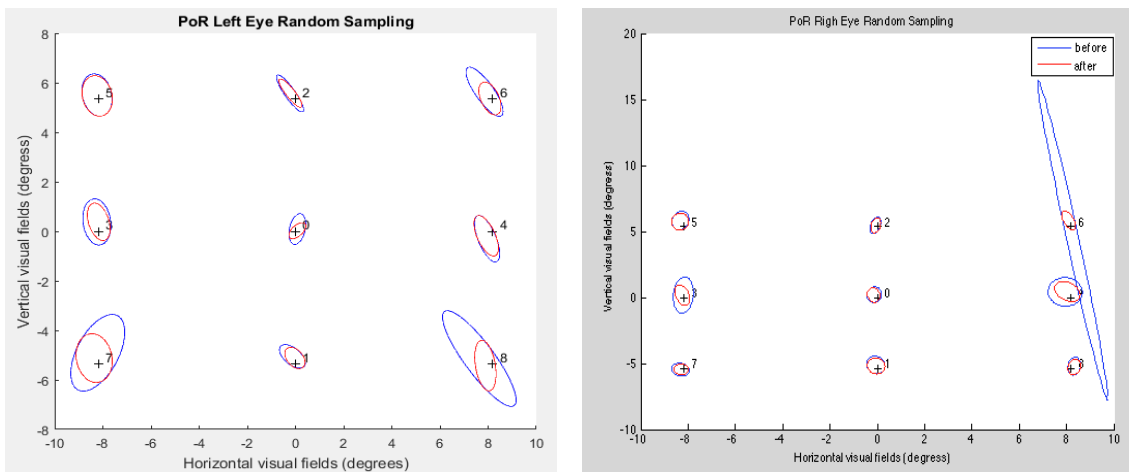


Figura 4.51: Distribución PoR Random Sampling.

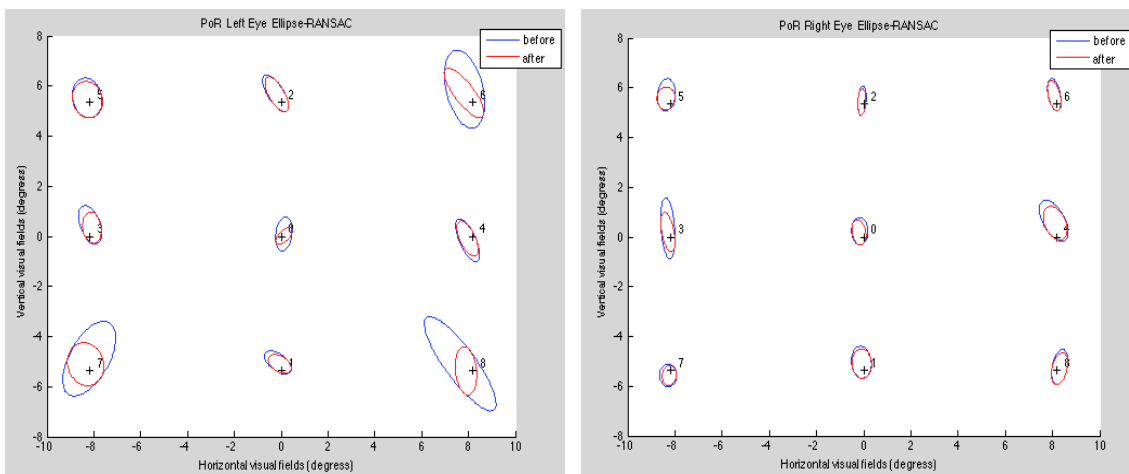


Figura 4.52: Distribución PoR Ellipse RANSAC.

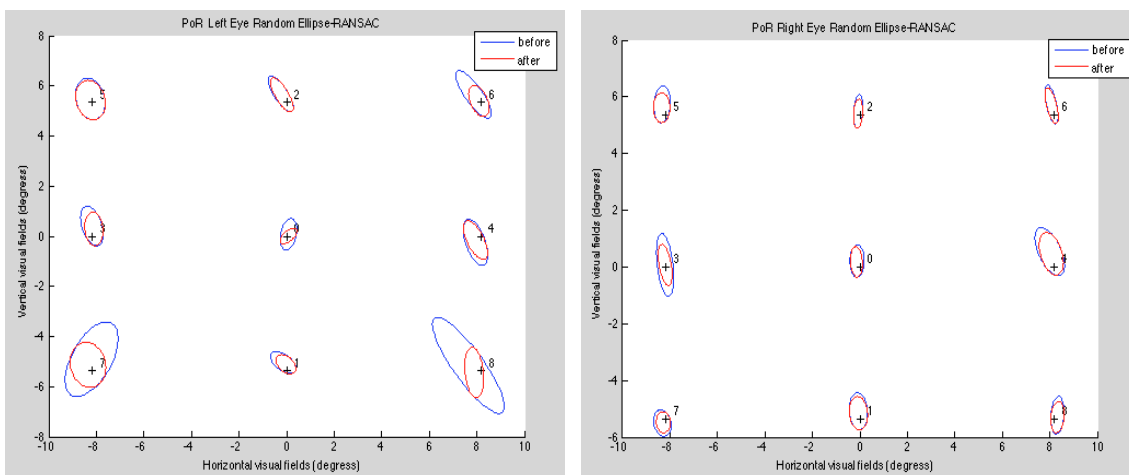


Figura 4.53: Distribución PoR Random Ellipse-RANSAC.

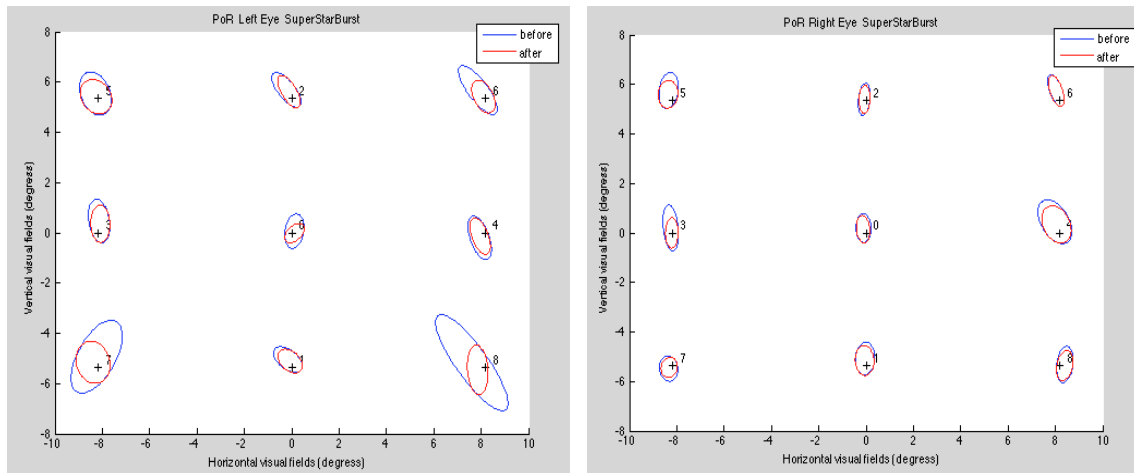


Figura 4.54: Distribución PoR SuperStarBurst.

La siguiente tabla corresponde al resumen de las figuras anteriores.

MÉTODOS	Ojo=LE	Ojo=RE
<b>Random Ellipse-RANSAC d=3.5</b>	<b>0,6692 ± 0,5657</b>	<b>0,5978 ± 0,3719</b>
SuperStarBurst	0,6991 0,5472	0,6368 ± 0,3568
Hyper accurate correction	0,6955 0,5701	0,6241 ± 0,385
Fns	0,6955 0,5701	0,6241 ± 0,385
Hyper-renormalization	0,6957 0,579	0,6352 ± 0,3992
Geometric Distance Minimization	0,6974 0,581	0,6222 ± 0,3751
Iterative reweight	0,695 0,581	0,6376 ± 0,4021
Random sampling	0,6957 0,579	0,6385 ± 0,4052
Renormalization	0,6987 0,582	0,6355 ± 0,3996
Hyperls	0,7023 0,5894	0,6557 ± 0,4103
Leastsquare	0,704 0,5912	0,6549 ± 0,4101
Fitzgibbon	0,7041 0,591	0,6577 ± 0,4146
Taubin	0,7023 0,5897	0,6558 ± 0,4103

Tabla 4.6: Valor medio del error de cada método sin imágenes que no mantienen un consenso.

La siguiente gráfica de boxplot se ha realizado eliminando esos errores y se puede apreciar que el método que menor mediana tiene es el Random ellipse RANSAC.



Figura 4.55: Boxplot de todos los métodos para el ojo izquierdo.

Como los resultados varían con el primer o segundo decimal, en la siguiente tabla se han resumido los valores de la gráfica.

Métodos	Límite inferior	Q1	Q2 mediana	Q3	Límite superior
Least Square	0,0147	0,28	0,47	0,9	1,7645
Iterative reweight	0,0257	0,28	0,49	0,9	1,78
Taubin	0,0184	0,28	0,47	0,904	1,76
Renormalization	0,0238	0,275	0,49	0,903	1,78
HyperLS	0,018	0,28	0,47	0,9	1,76
Hyper-renormalization	0,024	0,27	0,49	0,9	1,78
FNS	0,022	0,278	0,5	0,889	1,78
Hyper accurate correction	0,022	0,278	0,49	0,889	1,78
Geometric Distance Minimization	0,025	0,28	0,47	0,898	1,82
Fitzgibbon	0,0158	0,28	0,49	0,898	1,77
Random Sampling	0,024	0,27	0,49	0,9	1,78
Ellipse-RANSAC	0,024	0,27	0,49	0,91	1,86
<b>Random ellipse-RANSAC</b>	<b>0,0015</b>	<b>0,27</b>	<b>0,45</b>	<b>0,86</b>	<b>1,741</b>
SuperStarBurst	0,037	0,32	0,53	0,88	1,762

Tabla 4.7: Resumen del diagrama de cajas para el ojo izquierdo.



En el caso del ojo derecho, los resultados se encuentran todavía más ajustados, pero numéricamente aunque sea en términos del segundo decimal el método Random ellipse-RANSAC da el mejor resultado.



Figura 4.56: Boxplot de todos los métodos para el ojo derecho.

En la siguiente tabla se han resumido los valores obtenidos por la gráfica del boxplot.

Métodos	Límite inferior	Q1	Q2 mediana	Q3	Límite superior
Least Square	0,0357	0,34	0,538	0,898	1,73
Iterative reweight	0,04	0,327	0,518	0,876	1,68
Taubin	0,0246	0,34	0,53	0,897	1,72
Renormalization	0,04	0,33	0,52	0,87	1,67
HyperLS	0,02	0,34	0,532	0,89	1,72
Hyper-renormalization	0,04	0,33	0,518	0,87	1,67
FNS	0,046	0,313	0,514	0,86	1,68
Hyper accurate correction	0,046	0,313	0,514	0,86	1,68
Geometric Distance Minimization	0,042	0,326	0,514	0,86	1,66
Fitzgibbon	0,038	0,345	0,531	0,899	1,728
Random Sampling	0,40	0,33	0,518	0,87	1,67
Ellipse-RANSAC	0,0021	0,31	0,51	0,875	1,7
<b>Random ellipse-RANSAC</b>	<b>0,007</b>	<b>0,296</b>	<b>0,49</b>	<b>0,815</b>	<b>1,59</b>
SuperStarBurst	0,014	0,33	0,53	0,835	1,58

Tabla 4.8: Resumen del diagrama de cajas para el ojo derecho.



# Capítulo 5

## Conclusiones y líneas futuras

### 5.1 CONCLUSIONES

En la comparación de centros obtenidos por los algoritmos propuestos respecto a los de *Vicomtech* y *SuperStarBurst* se pudo ver que los centros de los algoritmos propuestos se asemejaban más a los centros de *SuperStarBurst*. Además, dicha afirmación cobraba sentido ya que analizando las imágenes se observaba que el centro de *Vicomtech* no era muy preciso.

En los histogramas acumulados que representan las diferencias de centros de los algoritmos propuestos y *SuperStarBurst*, se concluye que en el caso de los reflejos, cualquiera de los métodos excepto los dos algoritmos RANSAC, eran los que más se asemejaban al resultado del *SuperStarBurst*. Sin embargo, a la hora de calcular el error del centro de los reflejos de las imágenes problemáticas se pudo observar que los que mejor resultado daban eran o el *Random Ellipse-RANSAC* o el *Ellipse RANSAC*.

Respecto a la separación o no de los ojos para las distintas pruebas, se ha decidido que es mejor no unificarlos ya que a la hora de obtener las imágenes se realiza una calibración individual para cada ojo. Esto se debe a que puede haber pacientes que tengan problemas en un solo ojo y por lo tanto una sola calibración para los dos ojos no sería lo óptimo.

Respecto a la posición de la mirada (PoR), por lo general el ojo derecho es más preciso que el izquierdo. Además, en el ojo izquierdo a medida que el usuario va pasando de una fijación a otra el error máximo va aumentando con lo que en las fijaciones 7 y 8 es donde se encuentra la mayor

dispersión de puntos PoR y en el centro donde menos. Respecto al ojo derecho no hay grandes diferencias entre las fijaciones del centro y borde.

Finalmente, midiendo la bondad de los métodos mediante el PoR se ha podido observar que el método que ajusta con mayor precisión corresponde a *Random Ellipse-RANSAC* con un umbral de  $d=3.5$  píxeles. Al analizar como variaban los resultados dependiendo del umbral se pudo observar que el cambio solamente se daba en la fijación 3, con lo que la variabilidad era mínima. Respecto a los métodos *Random Sampling* y *Hyper-renormalization* se pudo ver que daban los mismos resultados ya que las características para realizar el método *Random Sampling* solamente se daban en una imagen.

## 5.2 LÍNEAS FUTURAS

Los resultados obtenidos demuestran que la diferencia entre métodos es muy pequeña, por lo que habría que aumentar la base de datos de las imágenes problemáticas, ya que es donde más se diferencian los métodos. Esto conllevaría a aumentar el número de usuarios, puesto que solo hay 7 usuarios.

Otra de las posibles mejoras sería cambiar el patrón de calibración. En este caso se utilizan patrones de nueve puntos pero habría que calcular con patrones de 16, ya que mejoraría la robustez y precisión del sistema.

Para terminar, habría que realizar un post procesado para eliminar las imágenes cuya posición de la mirada no mantengan un consenso respecto a las fijaciones, es decir, eliminar de manera automática aquellas imágenes que estén mirando a otro punto y por lo tanto se conseguirían unos resultados más precisos de los PoR por fijaciones.

# Bibliografía

- [1] A. Duchowski, «A breath-first survey of eye-tracking applications,» *Behavior Research Methods Instruments and Computer* 34, pp. 455-470, 2002.
- [2] D. Parkhurst y E. Niebur, «A feasibility test for perceptually adaptive level of detail rendering on desktop systems,» *ACM Applied Perception in Graphics and Visualization Symposium*, pp. 105-109, 2004.
- [3] E. N. D. Parkhurst, «Variable resolution displays: a theoretical, practical and behavioral evaluation,» *Human Factors*, Vols. 44 de 2611-29, 2002.
- [4] R. Jacob, «The use of eye movements in human-computer interaction techniques: what you look at is what you get,» de *ACM Transactions on Information Systems*, 1991, pp. 152-169.
- [5] A. Duchowski, «Eye Tracking Techniques, Electro-Oculography,» de *Eye Tracking Methodology*, Springer, p. 52.
- [6] A. Duchowski, «Eye Tracking Techniques, Scleral Contact Lens/ Search Coil,» de *Eye Tracking Methodology*, Springer, p. 52.
- [7] A. Duchowski, «Eye Tracking Technology, Photo-Oculography (POG) or Video-Oculography (VOG),» de *Eye Tracking Methodology*, Springer, p. 53.
- [8] A. Józwik, D. Siedlecki y M. Zajac, «Analysis of Purkinje images as an effective method for estimation of intraocular lens implant location in the eyeball».
- [9] D. Koons, C. Morimoto, A. Amir y M. Flickner, «Pupil Detection and Tracking Using Multiple Light Sources».
- [10] M. C.H., D.Koons, A.Amir, F. M. y Z. S., «Keeping and Eye for HCI,» *XII Brazilian Symposium on Computer Graphics and Image Processing*, pp. 171-176, 1999.
- [11] F. L.A. y W. K.P, «Eye-Gaze Word Processing,» *IEEE Transactions on Systems, Man and Cybernetics*, vol. 20, nº 4, pp. 944-950, 1990.
- [12] H. T.E., K. White, W. Martin, K. Reichert y F. L.A., «Human Computer Interaction Using Eye-Gaze Input,» *IEEE transactions on systems, man and cybernetics*, pp. 1527-1534, 1989.

- [13] D. Li., D. J. y P. j., «Starburst: A robust algorithm for video-based eye tracking,» *Human Computer Interaction Program*.
- [14] K. Kenichi, S. Yasuyuki y K. Yasushi, *Ellipse Fitting for Computer Vision*, Morgan & claypool publishers, 2016.
- [15] F. Andrew W., P. Maurizio y R. B.Fisher, «Direct Least Squares Fitting of Ellipses».
- [16] F. Martin A. y B. Robert C., «Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography,» *SRI International*.
- [17] K. Kanatani, Y. Sugaya y K. Yasushi, *Ellipse Fitting for Computer Vision: Implementation and Applications*, Morgan & Claypool, 2016.