UNIVERSIDAD POLITECNICA DE VALENCIA

MIARFID
Máster en Inteligencia Artificial, Reconocimiento de Formas e Imagen Digital

TRABAJO FINAL DE MASTER

TRACKING OCULAR MEDIANTE UN SISTEMA ÓPTICO MONOCULAR CON MARCAS NATURALES

JULIO DE 2012

AUTOR: ASÍS TÁRREGA ARTIEDA

DIRECTOR: CARLOS MONSERRAT ARANDA

CODIRECTOR: MARIO ORTEGA PÉREZ

AGRADECIMIENTOS

Siempre he creído que las personas se hacen, no nacen. Por este motivo, agradezco a mi entorno, a mi casa, a mi familia, ser el hombre en el que me han convertido. Gracias Enana por todos esos momentos de discusión que me han hecho más fuerte, gracias Papá, sin tu interés y tu forma de ver la vida posiblemente no sería como soy, y sobre todo, gracias Mama por tu buena influencia y por tus consejos, por tu amor incondicional hacia mí y por tu apoyo constante. Aunque hayáis pagado muchas de las frustraciones que en momentos de debilidad he sentido, quiero que sepáis, que si hay algo que no quisiera que cambiara nunca en mi vida, sois vosotros.

Siempre he creído que para progresar en la vida, son tan necesarias las oportunidades que le brindan a uno, como el trabajo duro que realiza para aprovecharlas. Estos últimos 8 meses me han dado una oportunidad que espero haber aprovechado. Gracias Mario por tus consejos y tu paciencia, y sobre todo gracias Carlos por tus innumerables correcciones, por la paciencia que has tenido conmigo y por tus constantes palabras de ánimos… mil gracias.

Siempre he creído que el ambiente por el que se mueve una persona, resulta fundamental para encarar los problemas y desafíos de una forma más alegre. Durante la elaboración de este TFM, un gran grupo de personas me ha acogido y me ha ayudado, interesándose por el desarrollo de mi trabajo y haciendo que estos 8 meses pasen en un visto y no visto. Gracias Fran, Miguel, Alejandro, Mª Ángeles, Fer, Eliseo, Juanjo, Sandra, Pablo, Rober, David, Alex, Maja, Valery, Mª José… Gracias Labhuman.

Este TFM ha sido realizado por mí, pero sin vosotros no hubiera sido posible. GRACIAS.

DEFINICIONES Y ACRÓNIMOS

ANN…… Artificial Neural Network o red neuronal artificial.
BCI…….. Brain Computer Interfaces: metodología basada en la adquisición de ondas cerebrales para controlar un ordenador.
Canny…… Operador que utiliza un algoritmo de múltiples etapas para detectar una amplia gama de bordes en imágenes.
Conjuntiva…… Membrana mucosa y transparente que tapiza el globo ocular .
Cornea…….. Estructura hemisférica y transparente que permite el paso de la luz y protege al iris y al cristalino.
Coroides…….. Membrana oscura situada en la pared del ojo que mantiene la temperatura del globo ocular constante.
CWT-BD…….. Continuos Wavelet Transform-Blink Detection: algoritmo que detecta parpadeos.
CWT-SD…….. Continuos Wavelet Transform-Saccade Detection: algoritmo que detecta movimientos sacádicos.
DoG…… Diferencia de gaussianas.
Eje óptico…….. Línea que viene determinada por la unión de los centros de la cornea y el cristalino.
Eje visual…… Línea que une la fóvea con el centro del cristalino.
EKF…….. Extended Kalman Filter: se utiliza para identificar el estado oculto de un sistema dinamico no lineal.
EOG…… Electro-Oculografía: Sistema de medición del movimiento ocular mediante la utilización de electrodos situados alrededor de los ojos.
Esclerótica…… Membrana blanca que da forma al globo ocular y proteger los elementos interiores del mismo
Eye tracking invasivo……. Tipo de eye tracking que utiliza accesorios como gafas, electrodos o lentes de contacto para adquirir señales del movimiento ocular.
Fijaciones…… Movimientos del ojo de entre 200 y 300 ms durante los cuales el ojo permanece estático, para obtener detalle de una zona de la escena.
Fotocronógrafo…….. Instrumento para registrar fotográficamente fracciones de tiempo pequeñas.
Fóvea…… Pequeña depresión en la retina donde se enfocan los rayos luminosos que penetran en el globo ocular.
FPS…….. Frames por segundo.
HCI…….. Interacción entre Humanos y ordenadores
HSV…… Hue, Saturation and Value: espacio de color que separa la imagen en tres canales: matiz, saturación y valor.
HUD…….. Head-Up Display: pantalla transparente que presenta información al usuario de tal forma que éste no debe cambiar su punto de vista para ver dicha información.
IR…….. Infrarroja
Mastoideo…… Hueso situado detrás y debajo de la oreja.
Oftalmoscopia indirecta.. Examen del interior de la parte posterior del ojo mediante un haz de luz y una lente que se sostiene con la mano.
PCCR…….. Vector formado por los puntos correspondientes al centro de la pupila y a la primera imagen de Purkinje.

Retina……	Tejido sensible a la luz, gracias al cual se producen una serie de fenómenos químicos y eléctricos que se traducen en impulsos nerviosos enviados finalmente al cerebro mediante el nervio óptico.
RGB……	Red, Green and Blue: espacio de color que separa la imagen en tres canales, uno rojo, uno verde y otro azul.
RPE……	Epitelio Pigmentario de la Retina: capa de células pigmentadas que aparece en el exterior de la retina que nutre sus células visuales
Sacádicos……	Movimientos rápidos del ojo de una duración de entre 20 y 200 *ms* que barren la escena para crear un mapa mental de la misma.
SIFT……	Scale Invariant Feature Transforms: algoritmo de extracción de puntos invariantes.
SURF……	Speed Up Robust Features: algoritmo de extracción de puntos invariantes.
SVD……	Singular Value Descomposition: técnica de factorización matricial.
TFM……	Trabajo Final de Máster.
Transiluminación……	La transiluminación es el paso de un haz de luz a través de los tejidos del ojo, por medio del cual, se logra iluminar y ver por dentro los tejidos.
VOG……	Video-Oculografía: Sistema de medición del movimiento ocular mediante la utilización de video grabado o en directo de los ojos.

ÍNDICE DE FIGURAS

ÍNDICE DE CONTENIDOS

1. INTRODUCCIÓN

El melanoma de coroides es el tumor maligno primario intraocular más frecuente en el adulto, siendo los varones de raza blanca con una media de edad de 53 años los más afectados por éste. Representa el 5-6% de todos los melanomas diagnosticados y su incidencia se calcula en 6 casos por millón por año.

En el pasado, el único tratamiento posible era la enucleación, basada en la extirpación del globo ocular manteniendo los músculos orbitales, los párpados y la glándula lagrimal. Sin embargo, en la actualidad su tratamiento depende tanto de la localización y del tamaño del tumor como del instrumental con el que se cuente.

Hoy en día, tras comprobarse que la tasa de mortalidad no mejora en casos en los que se aplica tratamiento de braquiterapia frente a la enucleación, se ha eliminado el tratamiento basado en la enucleación, excepto en casos donde el melanoma tiene un gran tamaño. Por tanto, las posibilidades de tratamiento se reducen a la radiación, mucho más eficiente y con un menor impacto psicológico en los pacientes que la enucleación, ya que permite al paciente conservar tanto el ojo como la visión, aunque ésta pueda verse afectada.

Dentro de la radioterapia, la braquiterapia epiescleral es la más utilizada en España. El tratamiento se basa en la exposición del ojo a radioterapia, tras realizar la cirugía necesaria para abrir la conjuntiva y tener acceso a la pared ocular. Tras esto se localiza el tumor mediante transiluminación, se posiciona una placa molde sobre el tumor, se vuelve a realizar la transiluminación para conocer la exactitud de la colocación de la placa, y si la placa está colocada correctamente, se sustituye dicha placa por otra cargada con semillas de yodo-125.

Ésta técnica es muy válida y fiable sobre todo para tumores de tamaño medio o grande y de localización anterior. Cuando son pequeños, poco pigmentados y de localización posterior en el ojo es difícil localizarlos con exactitud, teniendo que recurrir a la exploración del fondo de ojo mediante oftalmoscopia indirecta e recolocación de la placa de prueba para confirmar la correcta colocación de la misma, lo que obliga en ocasiones a rectificar la posición de la placa varias veces.

1.1. Objetivos del TFM

Este TFM tiene como objetivo la validación de un sistema que permita el guiado monocular óptico que facilite el correcto seguimiento del movimiento del ojo para así poder servir de guía en operaciones de braquiterapia.

Uno de los problemas para conseguir esto es la ausencia de elementos claros de seguimiento dentro del ojo, ya que en todos los trabajos de eye tracking, se utilizan características propias de las pupilas, iris… para buscar puntos relevantes para el seguimiento. Este procedimiento es totalmente inviable debido a que, en operaciones de braquiterapia, esta zona ocular no está siempre visible.

Sin embargo, la naturaleza ocular hace posible que tras abrir la conjuntiva, la pared ocular tenga ciertos rasgos característicos que no cambian entre frames consecutivos: los pequeños vasos capilares propios del ojo. Éstos hacen posible el seguimiento del ojo, siempre y cuando encontremos frames cercanos en el tiempo que tengan un número mínimo de correspondencias entre ellos.

Para validar el sistema que se propone en este TFM, se realiza el seguimiento de una esfera, con figuras arbitrarias dibujadas en su superficie, de la que se obtienen video en tiempo real gracias a una videocámara. Este flujo de video se analiza mediante la librería OpenCV, con la que se manejan las imágenes y se extraen los datos necesarios para el análisis del movimiento de la esfera.

1.2. Objetivos específicos

Para conseguir el objetivo propuesto para la tesina, es necesario dividir el problema en sub problemas u objetivos específicos. Como resultado del análisis de problema a resolver, aparecen tres objetivos de obligado cumplimiento para la consecución del éxito del TFM:

- Calibración de la cámara: localización de los parámetros intrínsecos de la cámara y de la distorsión producida por la lente de la misma, para más tarde revertir los efectos de esta distorsión.
- Localización de la esfera dentro de la imagen.
- Seguimiento del movimiento de la esfera una vez localizada dentro de la escena.

1.3. Hipótesis de partida

Como producto de los objetivos de este TFM y de los objetivos específicos de la misma, aparecen un conjunto de características o hipótesis que el sistema debe de cumplir para su correcto funcionamiento. Estas hipótesis son las siguientes:

1. **Es posible realizar la localización y el seguimiento de una esfera con un sistema óptico monocular.**

   Dado que el material con el que se cuenta en la mayoría de quirófanos de este tipo es un visor que aumenta el tamaño de la proyección del ojo captada con una sola cámara, es necesario adecuar el sistema propuesto al material del que se dispone. De esta forma, no es posible utilizar un sistema de visión estereoscópico para detectar con total exactitud las coordenadas 3D de los puntos característicos que se utilizan para el cálculo de la rotación.

2. **Se puede calcular la posición y la rotación de la esfera con un error inferior a los 2 *mm*.**

   Dado el ámbito médico en el que se mueve este TFM, es necesario ser todo lo exacto posible, motivo por el cual se establece una cota de error máximo de 2 *mm* en cuanto la

localización y la rotación que la esfera realiza. Esta cota de error es el error considerado como aceptable por los cirujanos al colocar la placa sobre el tumor.

3. **Es posible la utilización de puntos invariantes para el cálculo de la rotación.**

   Debido a que no es posible asegurar la presencia de la pupila y el iris durante todo el proceso de cálculo, es necesario utilizar otras características para realizar el seguimiento del ojo y calcular correctamente la rotación. Por este motivo se pretende seguir los vasos capilares propios de la pared ocular. Como aproximación a esta metodología, se pretende calcular la rotación sufrida por la esfera mediante una serie de marcas aleatorias realizadas sobre la superficie de la esfera.

4. **Se puede aplicar todo lo desarrollado para el seguimiento del ojo en tiempo real.**

   Se define un sistema en tiempo real como aquel que no depende únicamente del resultado o precisión que devuelve, sino también del tiempo en el que se produce ese resultado. Además, este tipo de sistema debe interactuar con el mundo real, tomando éste como estimulo de entrada. En definitiva, debe de interactuar con el mundo real, emitir respuestas correctas y cumplir restricciones temporales [48], de forma que el programa tenga un aspecto fluido además de un comportamiento preciso.

1.4. Organización del TFM

Además del Capítulo 1 o introducción, el presente TFM consta de 6 capítulos más, que se describen brevemente a continuación:

Capítulo 2 Estado del arte del seguimiento ocular

Tradicionalmente, el termino seguimiento ocular se relaciona con el cálculo del punto donde un sujeto fija su vista. En este capítulo se describen los dos principales enfoques para el cálculo de dicho punto de vista: electro-oculografía y video-oculografía. Se realiza pues, un estudio sobre los trabajos publicados hasta la fecha en estas dos vertientes del seguimiento ocular.

Capítulo 3 Calibración de la cámara

Este capítulo contiene la descripción del modelo de la cámara y la explicación del funcionamiento del método de Zhang utilizado para calibrarla.

Capítulo 4 Localización de la esfera

Tras una necesaria etapa de preprocesado, se realiza un informe sobre tres técnicas distintas para realizar la localización de la esfera. Una vez comentadas y analizadas empíricamente, se elige una de ellas como la base para el cálculo de la posición de la esfera.

Finalmente se detallan una serie de correcciones necesarias para la correcta localización de la esfera. El motivo de estas correcciones tiene naturaleza geométrica, y provoca un desplazamiento en el centro localizado de la esfera.

Capítulo 5 Seguimiento del movimiento de la esfera

En este capítulo se realiza la detección de movimiento de la esfera. Una vez detectado el movimiento, se procede a calcular los puntos invariantes o característicos sobre la superficie de la esfera en las imágenes anterior y posterior al movimiento para, tras la realización de una serie de correspondencias entre los puntos de ambas imágenes, se calcule la rotación sufrida por la esfera.

Capítulo 6 Conclusiones.

Recapitulación sobre la precisión alcanzada por el método que se presenta en este TFM, argumentando su posible adaptación a imágenes reales de ojos durante operaciones quirúrgicas.

Capítulo 7 Contribuciones del TFM y trabajos futuros.

En este capítulo se hace un repaso sobre las características novedosas que se presentan en este TFM respecto al resto de métodos de eye tracking. Así mismo, se nombran los puntos a mejorar en siguientes trabajos para solucionar los problemas que surgen con el desarrollo del método propuesto.

2. ESTADO DEL ARTE

Los fines que tiene el eye tracking son muy diversos, van desde la interacción entre humanos y ordenadores, más conocida como HCI, hasta estudios de alteraciones en las capacidades cognitivas, pasando por estudios sobre el impacto de la publicidad.

En este apartado, se realiza una breve descripción histórica de la evolución del eye tracking y un resumen de los métodos actuales que trabajan en esta técnica.

2.1. Historia del eye tracking

A finales del siglo XIX surgieron los primeros trabajos interesados en la medición directa del movimiento ocular. El oftalmólogo Louis Emile Javal explicó en 1879 que el movimiento ocular que tiene lugar durante la lectura no es un suave barrido de los ojos a lo largo del texto, tal y como se presuponía, sino una serie de paradas cortas (fijaciones) y movimientos rápidos [24]

Hoy en día se conoce el comportamiento de los ojos, que se encargan de realizar pasadas rápidas buscando partes interesantes en la escena, para así crear un mapa mental que represente dicha escena. Estos movimientos rápidos se conocen como sacadas o movimientos sacádicos.

Más de 20 años después, en 1900, Edmund Huey [22] creó el primer prototipo de eye tracking utilizando una lente de contacto con un agujero. Esta lente se conectaba a un puntero de aluminio que se movía en respuesta al movimiento del ojo. De esta forma, surgía el primer eye tracking invasivo, mientras que el primer eye tracking no invasivo surgiría poco después: en 1901, Dodge y Cline [13] desarrollaron su propio sistema de eye tracking: un fotocronógrafo para grabar los movimientos oculares producidos durante la lectura y durante movimientos rotacionales.

Este aparato emitía líneas de luz a los ojos y grababa su reflejo mediante fotografía, por lo que fue el pionero en uso de las propiedades de reflexión propias de la cornea, propiedad que todavía se usa en la actualidad.

Unos años más tarde, Charles H. Judd [25] creó una cámara que grababa el movimiento ocular permitiendo el estudio detallado de éste mediante la observación de frames individuales. Gracias a este sistema, en los años 30 Guy Thomas Buswell, pionero en la psicología experimental, analizó los movimientos oculares en función de la edad y del nivel escolar de los sujetos que probaron el sistema [9] [8]. Así fue como encontraron variaciones en las fijaciones y sacadas entre distintas edades y distintos niveles de educación, lo que sirvió para dar lugar a avances en los campos de la educación y la alfabetización.

Entre 1950 y 1960, el psicólogo ruso Alfred L. Yarbus realizó una serie de experimentos sobre los patrones de movimiento ocular que cristalizaron en 1967 [53]. Para estudiar estos patrones, Yarbus usó un dispositivo óptico de su propia invención, que consistía en un disco de goma con un orificio diminuto fijado directamente al ojo. Este disco era considerablemente

más pesado y voluminoso que una lente de contacto normal, por lo que se hacía muy incómoda.

En el borde del disco, se colocaba un espejo, de forma que se dirigía un rayo de luz hacia este espejo, que se movía a la vez que el ojo. Cuando la luz se reflejaba en el espejo, el movimiento del ojo se registraba en un papel fotosensible, obteniendo así un mapa de la visión del usuario.

Como resultado de este dispositivo, Yarbus concluyó que el patrón visual que sigue un usuario al observar una escena sin un objetivo en particular, es muy diferente al que sigue cuando se le encomienda una tarea, como por ejemplo "fijarse en la expresión facial de los personajes en la escena".

En los 70, el eye tracking crece rápidamente, gracias a los estudios sobre los movimientos de ojo en la lectura, como los realizados por el psicólogo cognitivo Keith Rayner [38]. Durante esta década comenzó a utilizarse el alto contraste entre el iris y la esclerótica, facilitando la detección del borde del iris.

En esta década también destacaron Stanford Earl Taylor, quien creó un dispositivo mecánico para almacenar permanentemente el movimiento realizado por el ojo [43], y John Merchant [32], quien desarrolló un sistema que contaba con un oculómetro que obtenía el punto de vista mediante la detección de la pupila y los reflejos en la cornea. Este fue el primero en permitir el movimiento de la cabeza del usuario, aunque el área de movimiento permitido fuera pequeño(2,5 cm^3). Esto es posible debido a que según Merchant, los reflejos corneales son invariantes ante movimientos de traslación de la cabeza, pero no lo son ante movimientos de rotación oculares.

En la década de los 80, se comenzó a utilizar el eye tracking en interfaces HCI y el uso de los ordenadores ayudó al desarrollo del eye tracking.

Desde los 90 en adelante, se ha utilizado mucho esta tecnología para investigar el impacto que tiene la publicidad en los usuarios, además de para hacer estudios sobre usabilidad de determinados productos.

2.2. Estado actual

En la actualidad, podemos diferenciar los sistemas de eye tracking en dos grandes paradigmas, dependiendo de la forma que tienen para medir el punto de vista: electro-oculografía (EOG) y video-oculografía (VOG).

2.2.1. Electro-Oculografía(EOG)

Este sistema se basa en la diferencia de potencial existente entre la cornea y la capa mas interna de la coroides, más conocida como membrana de Bruch.

La diferencia de potencial entre estas partes del ojo es de unos 0,4-0,5 *mV*, y su origen se encuentra en el epitelio pigmentario de la retina, el cual permite utilizar una aproximación en la que se considera la presencia de un dipolo, el cual puede ser representado por

un vector cuyo brazo coincide con el eje posterior del globo ocular, donde la córnea corresponde al extremo positivo y la retina al extremo negativo de dicho dipolo.

El potencial producido por este dipolo es susceptible de ser registrado a través de sistemas de registro mediante la colocación de electrodos en zonas de la piel cercanas al ojo. Al medir el potencial producido por un dipolo, la magnitud (voltaje) y polaridad del potencial registrado dependerán, en gran medida, del ángulo del dipolo con respecto a los electrodos pertenecientes a dichos sistemas de registro.

La configuración de este sistema varía de un método a otro, utilizando distintos números de dipolos y situándolos en distintas posiciones.

En [7], Ward *et al.* utilizan 5 electrodos: dos verticales (v), dos horizontales (h) y uno de referencia(r) (Figura 1).

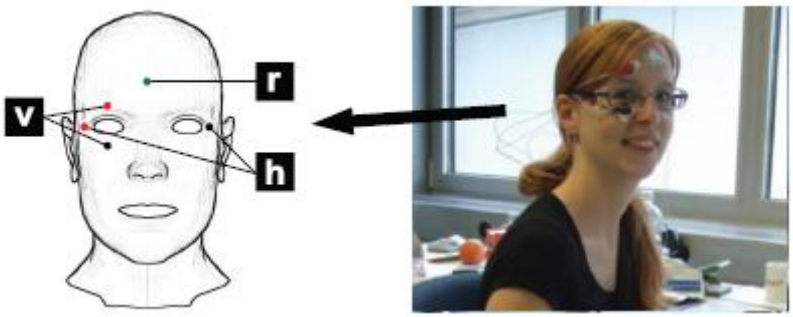


Figura 1. Representación de la posición de los electrodos en [7]

Gracias a estos electrodos, se producen dos señales, EOGh y EOGv, correspondientes al movimiento horizontal y vertical respectivamente, que forman las señales de entrada del algoritmo *Continuos Wavelet Transform- Saccade Detection* (CWT-SD). Este algoritmo divide las señales en segmentos sacádicos y no sacádicos (fijaciones), para calcular más tarde la amplitud y la dirección de los segmentos sacádicos.

Por su parte, los parpadeos son detectados con el *algoritmo Continuos Wavelet Transform-Blink Detection* (CWT-BD), utilizando un umbral máximo de 500 ms entre los dos grandes picos que se producen en el vector de coeficientes de la señal EOGv como característica propia de los parpadeos.

En este punto es necesario introducir el concepto de dispersión (*D*): Se asume que la mirada permanece estable durante una fijación y que los puntos a donde se dirige la mirada en una escena visual se agrupan muy juntos en el tiempo. Por este motivo, las fijaciones pueden ser identificadas mediante un umbralizado de la dispersión de estos puntos de vista.

Inicialmente, un movimiento no sacádico es considerado como una fijación, después el algoritmo elimina los segmentos con una dispersión mayor a un umbral máximo o de una duración por debajo de 200 ms.

Finalmente, y para saber con certeza el tipo de movimiento realizado por los ojos, se tienen en cuenta unos determinados parámetros para obtener una puntuación. Por ejemplo, para saber si estamos ante una fijación, se tienen en cuenta:

- Media y varianza de la amplitud de la señal horizontal.
- Media y varianza de la amplitud de la señal vertical.
- Duración o ratio de las fijaciones.

Como resultado final de este método, encontramos una precisión de entre 69% y 93% en 6 de los 8 usuarios que prueban el sistema, mientras que en dos usuarios dan resultados menores que el 50% debido a la poca calidad de la señal EOG. Los principales culpables de esto son la piel seca y la mala localización de los electrodos.

Los autores, además, llegan a la conclusión de que teniendo en cuenta los movimientos oculares, son capaces de discernir entre 6 posibles tareas que está realizando el usuario. Estas tareas pueden ser copiar un texto, leer, tomar notas a mano, ver un video, navegar por la red y periodos de inactividad específica.

El EOG Clínico es un test electrofisiológico del potencial del epitelio pigmentario de la retina (RPE), el cual provoca que la parte frontal del ojo tenga carga positiva respecto a la parte trasera del mismo. M. Brown *et al.* en [5] tienen como objetivo la creación de un estándar para la medición de señales EOG gracias al potencial del RPE, el cual provoca que la parte frontal del ojo tenga carga positiva respecto a la parte trasera del mismo. Como resultado de esto, el potencial medido entre 2 electrodos situados a ambos lados del ojo cambia cuando el ojo gira (Figura 2).

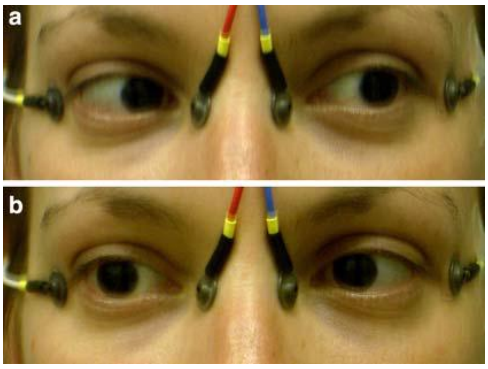


Figura 2. Representación de la posición de los electrodos en [5]. Además de estos 4, se utiliza un electrodo en la frente que hace las veces de “tierra”.

Este procedimiento tiene una serie de fases que se van sucediendo:

1. Se aplican gotas a los ojos para que la dilatación de las pupilas sea relativamente completa al inicio de las pruebas.
2. Se colocan cuatro pequeños electrodos a los lados de los ojos como se muestra en la Figura 2. Estos se conectan a canales separados de un amplificador diferencial. Además, se sitúa un quinto electrodo, que hace las veces de “tierra”, en la frente.
3. Se sitúan dos luces rojas con un ángulo respecto de la cabeza del usuario de +15 y -15 grados horizontales. Estas luces son en las que se tiene que fijar el usuario para medir la exactitud del sistema. La posición de la cabeza, por otra parte, se conoce gracias al uso de un soporte para el mentón, restringiendo el movimiento.

4. Fase de oscuridad: se introduce al usuario en un entorno con luz interior estable durante el máximo tiempo posible antes de las pruebas, para que después pase 15 minutos en total oscuridad a excepción de las luces de fijación, que van alternando mientras las señales EOG se graban.
5. Fase de luz: el usuario pasa 15 minutos con luz, continuando con la fijación del punto de vista en las luces rojas.
6. Con los datos obtenidos de las fases de oscuridad y de luz, se calcula la media de la amplitud de la señal EOG µV cada 10 segundos, para luego calcularse su media.
7. Se muestra por pantalla un grafico con estos cambios en la amplitud y se calcula el ratio de Arden: la amplitud máxima ocurrida en la fase de luz dividida por la amplitud mínima calculada en la fase de oscuridad.

Como resultado de todas las fases anteriores, se elabora un informe que contiene el ratio Arden, la amplitud en la fase oscura, el tiempo desde el comienzo de la fase de luz y el pico de la curva del ratio de Arden si lo hubiera y el tamaño de la pupila. Además, también se menciona cualquier incidencia ocurrida durante el test que pueda tener influencia en los resultados.

A.B. Isakali *et al.* [44] utilizan señales EOG para que pacientes que no son capaces de mover sus miembros o músculos faciales puedan interactuar con ordenadores.

En este sistema, se usan 5 electrodos, 2 para cada canal y otro para la toma de tierra, los cuales forman las señales EOG, que tras pasar por las etapas de filtrado y de amplificación son digitalizadas y transferidas al PC (Figura 3). Seguidamente son procesadas mediante un algoritmo de clasificación basado en el vecino más cercano, con un rendimiento de clasificación del 95%.

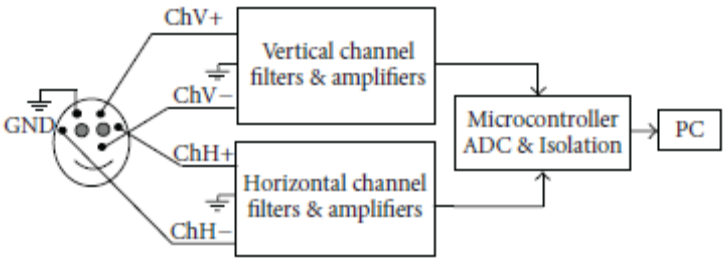


Figura 3. Colocación de los electrodos en [44].

Para tener una referencia, se compara el rendimiento de este sistema con otro llamado P300 BCI, basado en actividad cerebral para realizar la comunicación ordenador-persona. El deletreador P300 es un método no intrusivo, a diferencia de otros métodos BCI, que utiliza ondas cerebrales del usuario para que este escriba en la pantalla fijándose en la letra o numero que quiere escribir. Estos caracteres se presentan en la pantalla del ordenador en forma de una matriz de 6x6 celdas, y es la misma que la utilizada en este articulo (Figura 4).

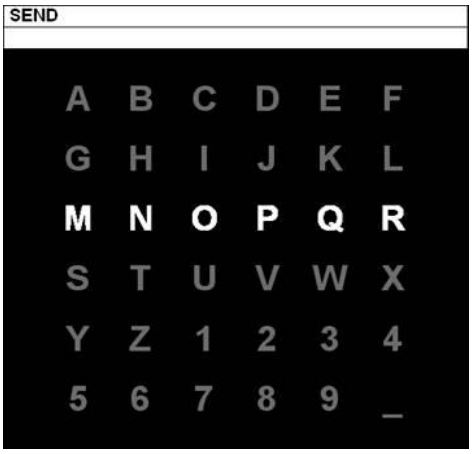


Figura 4. Interfaz utilizada en [44].

Comparando ambos métodos, el P300 tarda una media de 21 segundos para escribir una letra, y tiene una precisión de selección de carácter del 81% con una desviación típica del 14%. Por su parte, el método basado en EOG tarda una media de 24,7 segundos con una desviación típica de 3,2 segundos, y tiene una precisión del 100% en la selección del carácter. En cuanto a la comparación relativa a otros aspectos, como por ejemplo la monetaria, el P300 es un orden de magnitud más caro que el EOG.

Zhao Lv. *et al.* [31] utilizan 3 electrodos para captar las señales EOG (Figura 5): para el movimiento horizontal se coloca uno en la sien (H), para el movimiento vertical se coloca uno sobre la parte superior del ojo (V), sobre la línea del medio, y finalmente el electrodo referencia (R) se sitúa en el mastoideo.

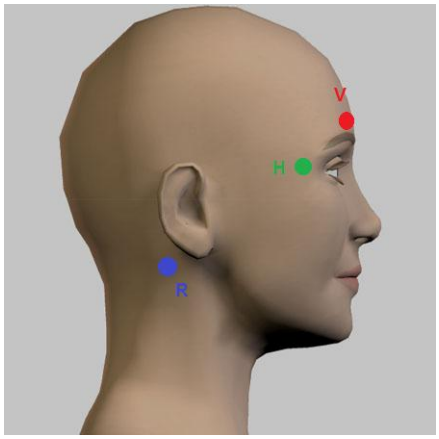


Figura 5. Posición de los electrodos en [31].

Después de amplificar la señal producida por los electrodos se realiza una etapa de preprocesado de la onda, donde se elimina el ruido mediante el filtro paso banda y se normaliza la señal mediante un umbralizado dinámico que evita las pequeñas fluctuaciones en la onda EOG.

El siguiente paso es la detección de parpadeos, tomando como parpadeo el conjunto de pulsos que se produce cuando entre un pulso positivo y un pulso negativo pasa un tiempo menor que 1,5 s.

Más tarde se procede a la detección de los movimientos oculares: se crean una serie de pulsos cuya polaridad está correctamente regulada, llamados pulsos referencia. Multiplicando estos pulsos referencia por la señal normalizada EOG se obtienen una serie de pulsos rectangulares, de forma que los pulsos positivos significan giros hacia arriba y negativos implican giros hacia abajo. La explicación dada corresponde a la señal vertical EOG, pero es extrapolable a la horizontal para detectar los giros a derecha e izquierda del ojo.

En cuanto al sistema de salida y validación, el usuario ve en pantalla en tiempo real los pulsos de salida que su mirada produce, y en caso de que considere que el sistema ha captado bien sus movimientos, parpadea tres veces para corroborar la corrección del sistema. De lo contrario, el usuario permanecerá 3 segundos con los ojos cerrados para reiniciar el sistema.

Una vez comprobada la secuencia de miradas del usuario, el sistema envía a un mini coche las órdenes correspondientes a las indicadas por el usuario: si éste mira dos veces hacia abajo, el coche va marcha atrás, si mira dos veces a la derecha, va hacia la derecha…

T. Zaveri *et al.* [56] obtienen señales EOG de 5 movimientos diferentes del ojo (arriba, abajo, derecha, izquierda y parpadeo) gracias a un modelo que los identifica. Estas señales se usan para controlar un juego para el ordenador (Quake II).

En cuanto al número y colocación de los electrodos, se colocan 5 (Figura 6): 2 para movimiento horizontal (B y E), 2 para movimiento vertical (C y D) y otro de referencia en la frente (A).

Figura 6. Posición de los electrodos en [56].

Para capturar el movimiento ocular, se usan las graficas de movimiento vertical y horizontal. En éstas se mide en cada ojo tanto el movimiento vertical como el horizontal. Nos encontramos con picos en las graficas que, dependiendo de cada grafica, tiene un significado u otro: Un pico en la grafica de movimiento horizontal significa movimiento lateral, mientras que un pico en la grafica de movimiento vertical significa un movimiento hacia arriba o hacia abajo. Cuanto mayor sean estos picos, mayor será la velocidad a la que se ha movido el ojo. Por otra parte, las características para determinar un parpadeo gráficamente son un pico positivo seguido por un leve pico negativo en la función de movimiento vertical.

Cabe destacar que es necesario realizar un proceso semiautomático de calibración, ya que las señales EOG varían dependiendo de factores tales como la colocación y conductividad de los electrodos o el hecho de que los patrones de amplitud varían entre usuarios distintos. Esta calibración consta de un patrón de movimiento ocular que se le pide al usuario que siga:

izquierda, derecha, arriba, abajo y parpadeo. Una vez hecho esto, se calcula la desviación estándar, que es usada como guía para elegir un umbral, que será como mínimo mayor que dicha desviación.

En cuanto a los resultados, se realizan pruebas a 3 sujetos, en las cuales se les pedía que realizaran 10 veces el siguiente patrón de movimientos: arriba, abajo, derecha, izquierda y parpadeo, con un porcentaje total de acierto del 95% y una utilización máxima del CPU del 5%. Hay que añadir que los usuarios calificaron como excitante la experiencia de jugar de esta forma a un juego y que no notaron ningún retardo en el sistema.

A. Bulling *et al.* [6] proponen un sistema basado en unas gafas que cuentan con 4 electrodos para capturar las señales EOG. Estas gafas son ligeras, tienen una gran autonomía y sensores aceleradores y lumínicos para compensar el ruido causado por actividades físicas y cambios en la iluminación (Figura 7).

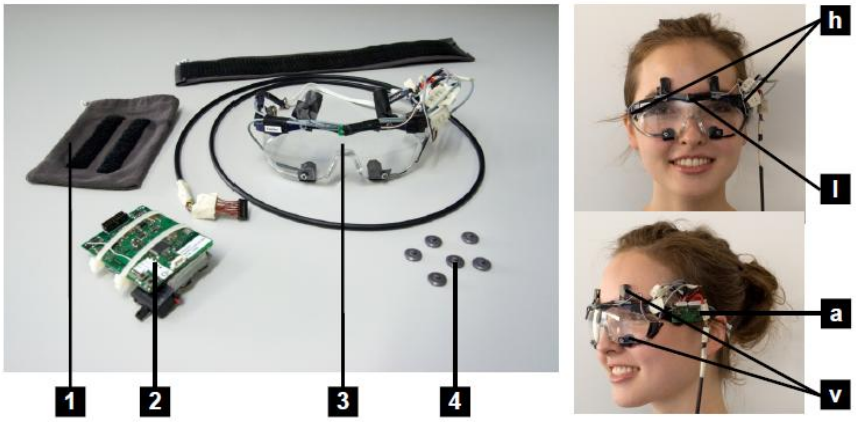


Figura 7. Componentes de [6]: brazalete con bolsa de tela 1), unidad de procesamiento 2), gafas 3), electrodos 4), electrodos para el movimiento horizontal h) y vertical v), sensor de luz l), acelerómetro a).

Por otra parte, para la detección de sacádicos, se utiliza el algoritmo CWT-SD. Después es necesario detectar los parpadeos debido a su similitud con movimientos verticales del ojo. Para esto se utiliza una plantilla de parpadeos creada manualmente usando segmentos cortados de igual tamaño de la señal EOG de 10 parpadeos de distintas personas, desplazadas verticalmente mediante su mediana y alineados en sus picos. Después de esto, para localizar los parpadeos, se calcula la distancia euclidea entre la plantilla y la señal vertical EOG mediante una aproximación de ventana deslizante. Si la distancia es menor que un determinado umbral, se califica a la porción de señal analizada como un parpadeo.

Para la eliminación de los parpadeos, es necesario analizar en paralelo los parpadeos y los sacádicos: los parpadeos que no tienen asociados un sacádico simultaneo son eliminados directamente. En el caso de que ambos coincidan en el tiempo, hay que diferenciar entre varias situaciones:

- Parpadeos presacádicos: causados por parpadeos cuyo último pico corresponde al inicio de un sacádico. Estos se eliminan, sustituyéndolos por el valor de la señal en el momento de inicio del parpadeo.
- Parpadeos intersacádicos: ocurren durante movimientos de ojo o periodos de fijación. Son eliminados, sustituyéndolos por una interpolación entre los valores que toma la señal a su inicio y a su fin.
- Parpadeos postsacádicos: su primer pico corresponde al fin de un sacádico. Se eliminan, sustituyéndolos por el valor de la señal al final del parpadeo.

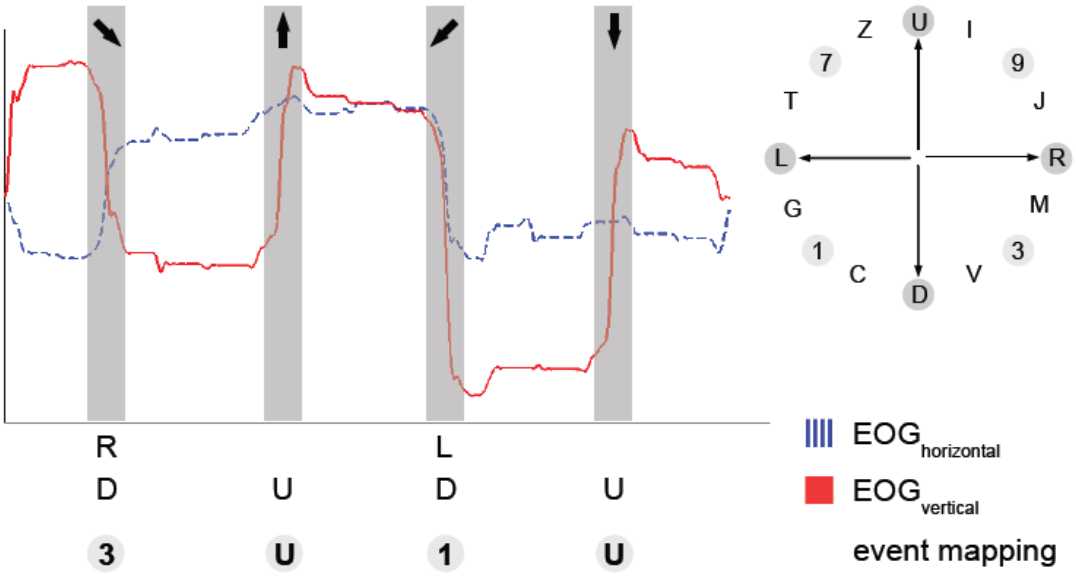


Figura 8. Cada carácter corresponde a uno de los 16 posibles destinos de la mirada detectados por el sistema.

Debido a que las gafas se pueden usar en movimiento, se ha optado por utilizar un algoritmo que utiliza un filtro adaptativo para explotar las características repetitivas producidas al andar y así eliminar el ruido que produce esta actividad. La otra alternativa es usar un filtro de mediana con un tamaño fijo de ventana, pero esto falla al eliminar el ruido, ya que las personas tienen distintas velocidades y longitud de paso.

Para probar el dispositivo, se han hecho dos tipos de pruebas:

1. **Movimientos oculares para HCI sin movimiento.**

   Los usuarios deben de seguir unos patrones de movimiento ocular en la pantalla del ordenador, estando sentados delante del mismo. Las señales analizadas se codifican mediante una serie de caracteres alfanuméricos (Figura 8). Inicialmente 14 usuarios comenzaron las pruebas, aunque 3 fueron descartados debido a la pobre calidad de señal resultante en la fase de calibración. La media de acierto es del 91% entre los 11 usuarios que completaron las pruebas, consistentes en un juego de 8 niveles, que cada usuario realizó tres veces, siendo la primera vez solo para familiarizarse. Cada uno de estos niveles lo forma un patrón de puntos rojos en los que el usuario debe de fijarse.

2. Movimientos oculares para HCI con movimiento.

Se usa un head-up display (HUD) a modo de pantalla, para que el usuario siga una recorrido con los ojos de forma similar al tipo de prueba anterior, pero mientras camina. Los usuarios son entrenados en el juego antes de las pruebas, pero lo hacen en el ordenador, no en el HUD. Una vez realizada la primera toma de contacto, los usuarios realizan 3 veces distintos recorridos en el HUD mientras caminan por un pasillo.

En estas pruebas lo que se pretende observar es la mejor combinación entre realizar la prueba en movimiento/sin movimiento y datos en crudo/filtro de mediana/filtro adaptativo. Como resultado, se afirma que se localizan 8 veces más sacádicos andando que sin movimiento, por lo que se afirma que no es posible interpretar la detección de movimiento ocular en el caso en que el usuario se encuentra en movimiento. Por otra parte, mientras el filtro de mediana de tamaño fijo falla al eliminar ruido, el filtro adaptativo tiene un buen rendimiento, particularmente en la señal EOG horizontal.

En conclusión, una vez vistos varios tipos de sistemas EOG, es posible afirmar que éstos tienen sus ventajas y desventajas respecto a otros métodos de seguimiento.

Respecto a las ventajas, cabe destacar:

- Equipamiento barato y fácilmente conseguible.
- Puede ser usado con gafas o lentillas.
- El equipamiento no obstruye el campo visual en ningún momento y es completamente insensible al movimiento de la cabeza del usuario, aunque una desviación significativa de la última posición calibrada requiera que el usuario repita la secuencia de calibración para un seguimiento preciso.

Respecto a los Inconvenientes, cabe destacar:

- Las señales medidas son variables respecto varias fuentes: cambios en la resistencia de la piel, deslizamientos o polarización de los electrodos, o incluso variaciones en el RPE debido al alojamiento de luz y el nivel de conciencia.
- La existencia de ruido proveniente de otros dispositivos eléctricos puede ser minimizada mediante una cuidadosa protección, pero los potenciales de acción de los músculos faciales puede enmascarar la señal.
- El inconveniente más obvio es la necesidad de colocar los electrodos en la piel del usuario.
- No realizan mediciones exactas de la posición exacta de mira, sino posiciones aproximadas a unas determinadas que el sistema espera obtener.

2.2.2. Video-Oculografía(VOG)

Esta técnica incorpora una cámara o dispositivo de adquisición de imágenes para tratar de determinar el movimiento de los ojos utilizando las imágenes obtenidas por dicho dispositivo.

Existen muchas variaciones entre sistemas basados en VOG:

- Número de cámaras: suelen usarse una o dos cámaras.
- Iluminación: Varía tanto el tipo de iluminación (se puede usa luz IR o luz normal), como el numero de luces para la correcta visualización de la característica que deseamos medir.
- Característica del ojo a controlar: Se usan la pupila, el iris, la esclerótica, reflejos especulares…
- Accesorios: Algunos de estos sistemas utilizan cascos, gafas o montajes que posibilitan fijar las cámaras a una distancia o posición determinada de los ojos.

En cuanto a la forma que tienen los sistemas VOG de obtener información del ojo para calcular el punto de vista del usuario, cabe destacar 3 métodos muy utilizados:

1. **Imágenes de Purkinje**

   Este método se basa en las reflexiones producidas por fuentes IR en distintas zonas del ojo. Según nuestro ojo esquemático, existen 4 imágenes de Purkinje:

   - Primera imagen de Purkinje: producida por la primera superficie de la córnea, es la más clara y brillante de las cuatro.
   - Segunda imagen de Purkinje: producida por la segunda superficie corneal, que suele estar casi siempre muy cerca o superpuesta a la primera, por no haber apenas distancia entre las dos superficies corneales.
   - Tercera imagen de Purkinje: producida por la primera capa del cristalino.
   - Cuarta imagen de Purkinje: producida por la capa interna del cristalino, la cual, al contrario que las otras tres, es una imagen invertida.

   El método de las imágenes de Purkinje se basa en la diferencia de posición entre la primera y la cuarta imagen de Purkinje (la generada en la superficie cóncava interna del cristalino), aunque también existen muchos otros que únicamente utilizan la primera imagen y otra característica ocular para el cálculo del punto de vista (Figura 9).

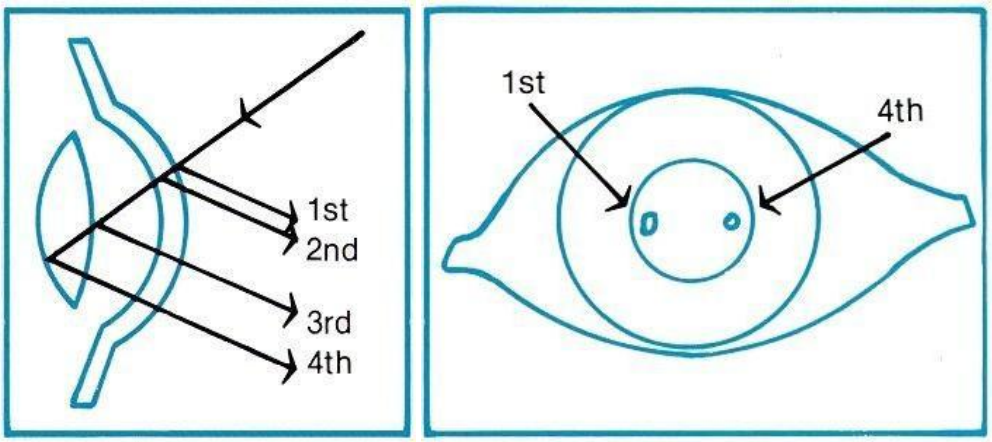


Figura 9. Imágenes de Purkinje

2. Pupila brillante y pupila oscura:

Uno de los factores importantes en VOG es la luz: la utilización de luz visible se complica debido a que la luz ambiente contiene componentes especulares de luz, lo que provocan reflexiones indeseadas. El uso de luz IR elimina esto y permite una iluminación uniforme de los ojos.

Dos métodos que utilizan luz IR y son muy utilizados son los de pupila brillante y pupila oscura:

- La técnica de pupila brillante: ilumina el ojo con una fuente de luz IR que se encuentra muy cerca del eje de la cámara. El resultado es una pupila claramente delimitada como una región brillante.
- La técnica de pupila oscura: ilumina el ojo con una fuente situada fuera del eje de la cámara, siendo así la pupila la parte más oscura del ojo.

La detección de la pupila tiene varias ventajas respecto a la esclerótica y el iris: el contorno de la pupila es pequeño, definido y difícil de ocluir por los parpados, aunque tiene un claro inconveniente: la luz IR no se puede utilizar en el exterior.

La técnica de la pupila brillante genera un mejor contraste iris/pupila debido a un seguimiento de ojos más correcto en relación a la pigmentación del iris y reduce significativamente las interferencias producidas por las pestañas y otras características ocultas

3. Vector PCCR

El vector entre el centro de la pupila y la primera imagen de Purkinje, también conocido como vector PCCR, se suele utilizar en lugar del centro de la pupila como única indicación para detectar la dirección de la mirada. Tiene varias ventajas sobre otros métodos, como que las características que necesita son fáciles de extraer y que el vector en cuestión es robusto ante movimientos de cabeza. Es un método ampliamente utilizado, siendo un ejemplo de esto las publicaciones [23], [15], [47] y [34] entre otros.

Entre los sistemas basados en VOG, es posible diferenciar entre dos tipos de aproximaciones: aproximaciones 2D y aproximaciones 3D.

2.2.2.1. Aproximaciones 2D

Estas aproximaciones transforman las coordenadas de las características o valores de intensidad del ojo en las imágenes 2D en las coordenadas del punto de vista en el plano del monitor 2D.

Algunas de las ventajas de esta aproximación son la facilidad de implementación y la ausencia de necesidad de calibrar la cámara o calcular la distancia al monitor.

En cuanto a los inconvenientes, suelen capturar la cara entera del sujeto, por lo que la región ocular no tiene una gran resolución. Existen excepciones, como en [34], que utiliza una imagen de gran resolución más propia de aproximaciones 3D.

Por lo general, estos métodos producen una precisión menor en el tracking ya que no usa información real 3D del ojo y necesitan de un proceso de calibración dependiente del usuario en cuestión en el cual éste debe fijar la mirada en unos puntos determinados en la pantalla durante el paso inicial, que en ocasiones puede ser muy largo [2].

En cuanto a las funciones de mapeado para transformar las coordenadas de las características extraídas del ojo en coordenadas de la pantalla, se usan aproximaciones polinómicas tanto de primer orden [11], [12] como de segundo [34], [33].

Un ejemplo de aproximación 2D es la introducida por Baluja *et al.* [2], los cuales proponen un eye tracker no intrusivo basado en la estimación del punto de vista mediante redes neuronales, que aporta una buena precisión en las mediciones mediante el uso no sólo de la posición de la pupila, sino también de los valores de intensidad de la imagen.

Uno de los beneficios de la implementación de la red neuronal en el eye tracking es la libertad de movimiento de la cabeza del usuario. Para tener en cuenta los desplazamientos relativos entre la cámara y el ojo, éste debe de localizarse en todo momento. En este sistema, el ojo derecho se localiza mediante la búsqueda del reflejo especular de una fuente de luz normal sobre el ojo. Se suele dar en forma de pequeño punto blanco rodeado por una región oscura (Figura 10). Una vez obtenida la posición del ojo, se limita la zona de búsqueda para el ojo en el siguiente instante de tiempo.

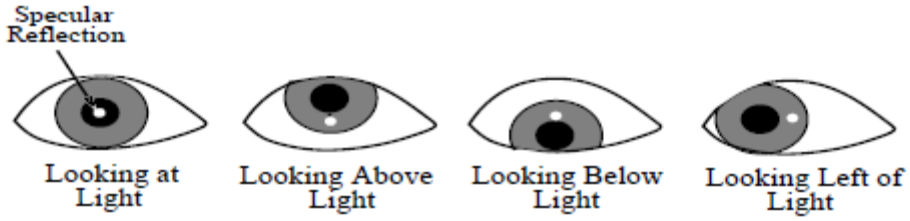


Figura 10. Posición relativa entre la reflexión especular y la pupila y su significado en [2].

Para determinar el punto de vista, a la red neuronal artificial (ANN) se le pasa como entrada los pixeles de la ventana extraída que forma el ojo. Esta ventana utiliza la información de la pupila y la cornea. Además de esta información, se utiliza la información de la posición de la pupila respecto a la cavidad ocular.

Para la búsqueda del punto de mira, se usa una representación gaussiana de los datos de salida de la red neuronal, en la que se encuentra un pico, que representa la localización del punto de vista en los ejes X e Y.

El entrenamiento de la red se hace pidiéndole al usuario que siga con la vista un camino trazado en la pantalla, siendo 2000 el número total de imágenes utilizadas para éste, mientras que para el test también se usan 2000. Se hace un entrenamiento mixto, pasándole a la ANN imágenes con distintas posiciones de la cabeza visualizando patrones de movimiento ocular tanto horizontales como verticales.

En cuanto a los resultados, se hacen diversas pruebas con el tamaño de las ventanas, y dividiendo o no la capa oculta en 2 (una para el cálculo de la coordenada *x* y otra para la

coordenada *y* del punto de vista), teniendo en cuenta el tamaño de los datos a procesar, la velocidad del sistema y el número de conexiones en la red neuronal. Con todo esto, el mejor resultado posible es de un error medio de 1.7 grados, lo que equivale a unos 1,27 *cm*, situando al usuario a unos 43 *cm*, con la siguiente configuración de entrenamiento:

- Imagen de entrada de 15x40 pixeles
- Capa oculta de la ANN de 8x2
- Tiempo aproximado de entrenamiento:30-40 minutos
- Maquina sobre la que se hace: Sun SPARC 10

Pese a la ventaja del rango de movimiento permitido al ojo, existe el inconveniente de que sea necesaria una etapa de entrenamiento propio de las redes neuronales.

Collet et al. [10] proponen un sistema en tiempo real que permite la detección y seguimiento de la cara, agujeros de la nariz y ojos con resultados de confianza que evita las restricciones propias de sistemas EOG.

Se basa en una cámara situada entre el teclado y el monitor del ordenador, orientada de forma que el techo esta de fondo por detrás de la cara del usuario, lo cual simplifica la separación de la cara del usuario y el resto. En cuanto a la luz, se sitúa una fuente de luz difusa a cada lado del monitor, orientadas de manera que se produzcan el menor número posible de reflejos en los ojos, para evitar fuentes de ruido en las mediciones.

Este sistema consta de las siguientes fases:

1. **Detección de la cara**

   Se calcula la diferencia entre frames consecutivos para localizar objetos en movimiento. Si se detecta un pequeño movimiento, significa que el usuario no se ha movido, por lo que se usa la caja de inclusión de la cara detectada en la imagen anterior. Para la primera localización, es necesario que la cámara capte un movimiento significativo de la cara.

2. **Detección de los agujeros de la nariz**

   Se buscan los pixeles oscuros candidatos a ser dichos agujeros y las pupilas, mediante umbralización. El siguiente paso es filtrar los candidatos mediante correlación de dos tipos de gradientes: el gradiente del candidato con el gradiente de la zona de este.

3. **Detección de los ojos**

   Se realiza la misma umbralización que la realizada en la etapa anterior, pero los candidatos se restringen a la zona superior izquierda y superior derecha de los agujeros de la nariz localizados. La misma correlación realizada en la fase anterior es realizada con los candidatos a ser pupilas.

4. **Orientación de los ojos**

   Gracias al vector formado por el centro de la pupila (pixel más oscuro) y el baricentro de la esclerótica, se calcula la orientación de la mirada.

5. **Orientación de la cara**

   Se evalúa la orientación de la cara mediante el vector formado por la posición de los puntos que representan los ojos y la nariz. Esto se realiza mediante el modelado 3D de la proyección del triangulo representado por los puntos previos [16].

6. **Orientación de la mirada**

   Calcula la mirada del usuario relativa a la pantalla mediante la composición de los 2 vectores calculados en los 2 pasos anteriores.

Para comprobar la eficacia del sistema, 32 personas participaron en los tests, grabando en total 40 videos de 1 minuto cada uno. Se obtienen un 3.4% de errores para la localización de la caja de inclusión de la cara, un 11.5 % en la localización de la nariz y una correcta detección y localización de los agujeros de la nariz en un 79%. Sin embargo, no se incluyen resultados de evaluación del eye tracking, solo se especifica que se está trabajando en un sistema de evaluación.

En otro trabajo, Colombo et al. [11] [12] proponen un HCI no intrusivo gracias al uso de un sistema óptico para medir el punto de vista. Para esto se sitúa una cámara sobre la pantalla, posibilitando que el punto de vista del usuario se calcule gracias a la posición de una de sus pupilas en la imagen.

Las medidas tomadas de la imagen para el eye tracking son obtenidas mediante el emparejamiento de un modelo o plantilla del contorno del ojo y del iris, con las imágenes provenientes de la cámara, mediante una aproximación de plantillas deformables basadas en ajuste por mínimos cuadrados.

La plantilla elíptica, que tiene 8 parámetros (radio *r* y posición del iris $x_c = (x_c, y_c)$, centro de la elipse $x_e = (x_e, y_e)$, eje mayor 2b, eje menor 2a y orientación de la elipse Θ) (Figura 11).

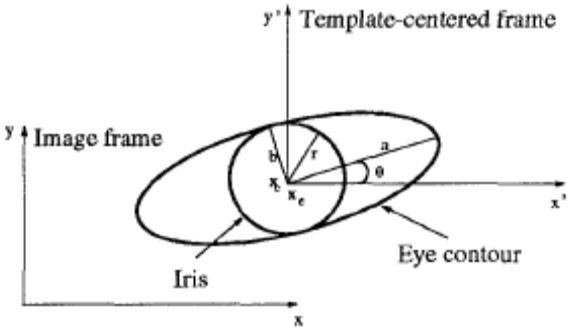


Figura 11.Plantilla de ajuste sobre el ojo en [11] [12].

Una vez situada la plantilla elíptica, se busca la posición del iris en el interior de ésta y se asume que el centro del iris es el centro de la pupila, algo necesario para estimar la dirección de la mirada.

El siguiente paso es la localización del punto de mira en la pantalla x_s, para lo cual se utiliza una correspondencia uno a uno con la localización de la pupila en la imagen x_p mediante un simple modelo de mapeo imagen-pantalla:

$$x_s = A \cdot x_p + t \qquad (1)$$

en donde *t* es una traslación y *A* es una transformación afín que puede contener una rotación rígida, escalado isotrópico y escalado anisotrópico. El modelo afín compuesto por *A* y *t* se estima mediante un procedimiento de calibración al inicio y después se va actualizando la calibración en tiempo de ejecución.

Para la calibración, se obtienen una serie de imágenes mientras el usuario realiza con la mirada unos caminos prefijados en la pantalla. El modelo afín se obtiene en ese momento utilizando los datos de los caminos en la pantalla y las imágenes en las que el usuario recorre dichos caminos. Será, pues, una solución por mínimos cuadrados para el sistema de ecuaciones sobredeterminado gracias a *N* ecuaciones del tipo (1), siendo *N* observaciones en imágenes de un determinado punto en la pantalla.

La actualización de la calibración en tiempo real tiene lugar asumiendo el movimiento de la cabeza de forma paralela a la pantalla, ya que de esta forma solo hay que tener en cuenta la actualización de la traslación *t*, algo fácilmente calculable restando las posiciones de dos posiciones consecutivas de las pupilas.

Este método tiene una gran limitación con la posición de la cabeza, ya que en las pruebas que hace, toma imágenes donde el movimiento de la misma es como máximo de 3 pixeles entre imágenes consecutivas. El error medio de la localización de la pupila es de unos 3 pixeles.

En cuanto a la precisión de la localización de la mirada, el error es de unos 3.25 *cm*, algo que se achaca a imprecisiones en el paso de la localización de la pupila y a leves movimientos de cabeza durante la calibración.

Otra de las propuestas es la realizada por Heinzmann *et al.* [20]. Estos plantean un sistema no intrusivo que requiere una cámara y es capaz de hacer frente al movimiento facial, incluyendo cambios en la profundidad. Esto es posible gracias a que no requiere imágenes muy cercanas del ojo por lo que permite un mayor movimiento de la cabeza siempre que los ojos no salgan del campo de vista de la cámara.

Para el cálculo del vector de dirección de la mirada, se necesitan 2 pasos:

1. Se calculan la localización del iris y los contornos interiores y exteriores de los ojos. Estas localizaciones son transformadas en dos vectores de orientación ocular que considera la posición de la cabeza, uno para cada ojo

2. La dirección de la mirada se calcula uniendo los resultados de cada ojo, de acuerdo a unos parámetros de confianza calculados para cada uno, de forma que el vector resultante parte del punto intermedio entre los 2 ojos y está formado por los vectores de orientación de cada ojo en mayor o en menor medida dependiendo del valor de confianza de cada ojo.

Finalmente, y tras obtener el vector de dirección de la mirada, es posible calcular el punto de mira realizando la intersección entre el vector director de la mirada y un modelo del mundo.

En este artículo, los autores no comentan resultado alguno, solo se limitan a mostrar imágenes (Figura 12) en las que se demuestra que son capaces de simbolizar la posición relativa entre los ojos, la posición de la pupila y de la cabeza, pero no existen datos sobre la precisión de los mismos.

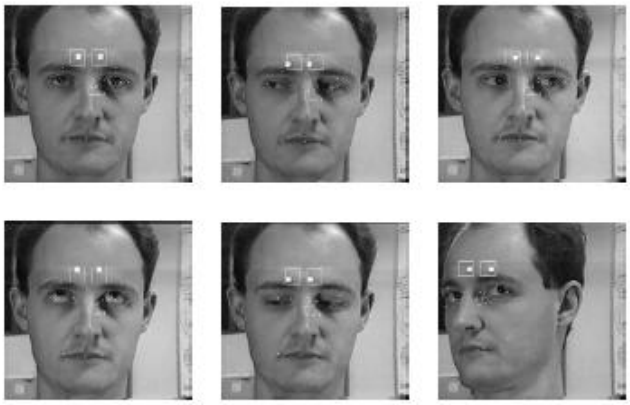


Figura 12.Ejemplo del rendimiento de [20].

Por su parte, Morimoto *et al.* [34] proponen un sistema de eye tracking de bajo coste que utiliza dos fuentes de luz IR y una cámara. Está basado en un detector robusto de pupila que utiliza iluminación IR para primero obtener la pupila mediante la técnica de pupila brillante y luego la técnica de pupila oscura para obtener la primera imagen de Purkinje (Figura 13).

Una vez obtenidas estas dos posiciones se utiliza un polinomio de segundo orden para modelar la transformación entre el vector PCCR al punto de vista sobre la pantalla, para lo cual hay que realizar una calibración.

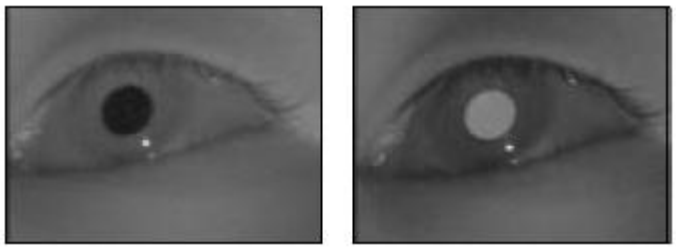


Figura 13.Ejemplo de las técnicas de pupila oscura y pupila brillante.

El proceso de calibración es simple y breve: consiste en que el usuario fije su mirada en una serie de puntos situados en la pantalla en forma de rejilla 3x3. En cada fijación, se guarda el vector entre el centro de la pupila y de la reflexión corneal, por lo que 9 puntos de la forma $E = (x_e, y_e)^t$ son almacenados y transformados a coordenadas de la pantalla $S = (x_s, y_s)^t$ dado por:

$$x_e = a_0 + a_1 x_s + a_2 y_s + a_3 y_s x_s + a_4 {x_s}^2 + a_5 {y_s}^2$$
$$y_e = a_6 + a_7 x_s + a_8 y_s + a_9 y_s x_s + a_{10} {x_s}^2 + a_{11} {y_s}^2 \quad (2)$$

, en donde a_i son los coeficientes del polinomio de segundo orden.

Dados 9 puntos a visualizar, se producen 18 ecuaciones y, por tanto un sistema sobredeterminado. Los coeficientes pueden ser obtenidos independientemente, así que tenemos dos sistemas lineales con 6 incógnitas y 9 ecuaciones cada uno, que se resuelven mediante el método de mínimos cuadrados.

El sistema produce un error de 1 grado, unos 0,872 *cm,* estando la cabeza del usuario a unos 50 *cm* de la pantalla. Los inconvenientes de este son la restricción de los movimientos de la cabeza, ya que el sistema utiliza una imagen de gran tamaño del ojo, y necesita que éste aparezca lo suficientemente bien colocado como para que se posible detectar la pupila y la primera imagen de Purkinje. Otro de los inconvenientes es que debido al uso de luz IR, no es posible probar el sistema en exteriores.

En interiores, el sistema ha sido probado con éxito en un gran número de personas y ha resultado robusto.

2.2.2.2. Aproximaciones 3D

Actualmente, muchos de los trackers utilizan una imagen del ojo con gran resolución gracias al zoom de las cámaras o a situar la cámara cerca del ojo. Esto produce una imagen con gran resolución, lo que posibilita el análisis 3D del ojo. Este hecho provoca que el movimiento de la cabeza esté muy limitado o que se busquen alternativas para posibilitarlo.

Una de estas alternativas es usar 2 cámaras, una para realizar el seguimiento de la cabeza y otra para el eye tracking.

Como ventajas respecto a las aproximaciones 2D, es posible afirmar que tiene una precisión mayor a la hora de calcular el punto de vista, que produce el vector 3D de la mirada respecto al marco de referencia de la cámara y que la calibración dependiente del usuario es más simple y rápida.

Como desventajas, encontramos la necesidad de realizar la calibración de la cámara y la mayor complejidad de implementación y de cálculos.

En las siguientes páginas se comentan sistemas de VOG que utilizan aproximaciones 3D observando las discrepancias entre las distintas configuraciones posibles: con una [41] [36] o dos cámaras [46] [41], y con distinto numero de luces: una [55], dos [54], tres [35] o cuatro [55].

Wang *et al.* [46] proponen un método de estimación del punto de vista robusto mediante imágenes de gran tamaño del iris, utilizando 2 cámaras, una para estimar la posición de la cabeza y otra para captar el iris de uno de los ojos.

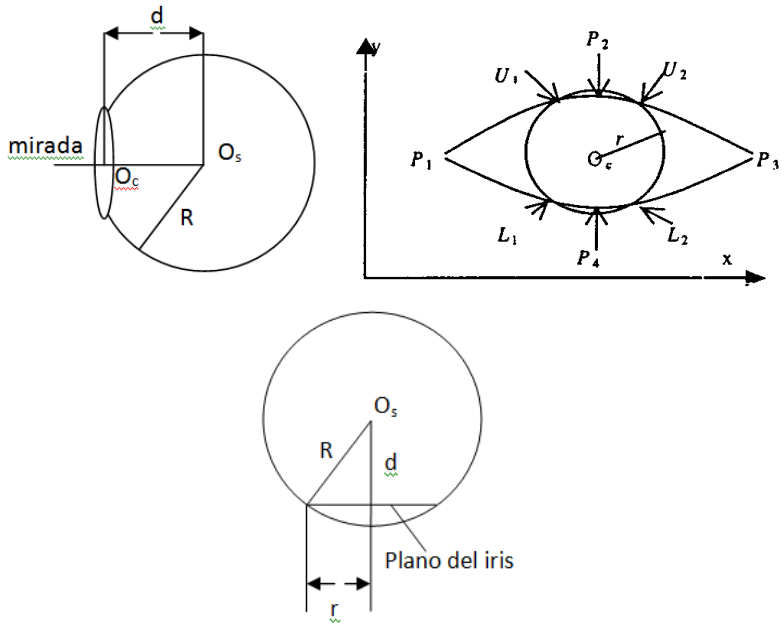


Figura 14. Modelo del ojo propuesto en [46]

Para localizar el contorno del iris (Figura 14), lo que nos interesa son las partes del contorno no ocluidas por los parpados: las curvas U_1L_1 y U_2L_2. Para obtener estas curvas, se realizan umbralizaciones, operaciones de aperturas y de extracción de bordes, para finalmente quedarse con los 2 bordes verticales más grandes y sobre estos se realiza el ajuste de una elipse.

Por otra parte, es posible conocer analíticamente el plano en el que se encuentra un círculo a partir de su proyección perspectiva conocido su radio. Esto se hace mediante el algoritmo posicionamiento de cámara monocular propuesto en [40]: en éste se afirma que dos soluciones posibles O_{s1} y O_{s2} de la posición 3D del plano del iris se pueden obtener a partir del circulo/elipse correspondiente. La elección de una de estas dos soluciones se hace mediante el algoritmo de un círculo, calculando las distancias entre O_{s1} y P_1, O_{s1} y P_3, O_{s2} y P_1 y O_{s2} y P_3, teniendo en cuenta 2 posibilidades:

- si $| O_{s1}P_1 - O_{s1}P_3| <= | O_{s2}P_1 - O_sP_3 |$ la solución es la formada por la normal n_1 y el centro del contorno del iris O_{c1}.
- de lo contrario, la solución es la formada por la normal n_2 y el centro del contorno del iris O_{c2}.

El problema de este método es que se obtiene el vector de dirección de la mirada, pero no el punto exacto de vista. Para solucionarlo, se fija la posición del usuario respecto al plano que observa, en este caso a 1,3 *m*, aportando un error medio de 1,26 *cm*.

Los problemas de este método son, por un lado la poca robustez en la detección y extracción de características faciales ante cambios en la posición del usuario, cambios de las condiciones de luz o incluso en la presencia/ausencia de gafas, barba y peinado, y por otro lado, el ajuste de la elipse al contorno del iris, ya que en imágenes en las que el número de pixeles que forman el contorno a ajustar puede ser pequeño.

En otro artículo, Shih *et al.* [41] proponen un método que utiliza múltiples cámaras y múltiples fuentes puntuales de luz para estimar el punto de vista del usuario sin necesidad de utilizar ningún parámetro dependiente de dicho usuario.

Se basa en el uso de la primera imagen de Purkinje para calcular la curvatura de la cornea, y utiliza el modelo del ojo que propone Le Grand [17], el cual aparece en la Figura 15.

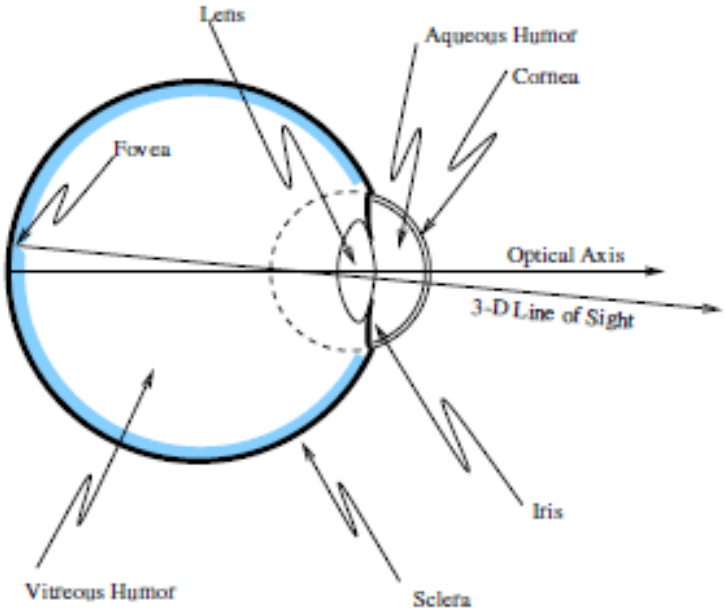


Figura 15. Modelo del ojo de Le Grand [17]

En cuanto al número de cámaras y luces a utilizar, se hace un estudio del mínimo número necesario dados los datos de los que se disponen para calcular el centro de la cornea, extrayéndose las siguientes conclusiones:

- Si se conoce el radio de la cornea *p*, se necesitan como mínimo una cámara y dos fuentes puntuales de luz con posiciones conocidas.
- Si no se conoce el radio de la cornea *p*, son necesarias como mínimo dos cámaras y dos fuentes puntuales de luz. Además, para obtener una única solución las fuentes de luz y uno de los centros ópticos de la cámara no deben ser colineales.

Tenemos los puntos p_i pertenecientes al borde del contorno de la imagen 2D de la pupila. La posición 2D del centro de la pupila p'_{ci} solo puede ser estimados usando los puntos p_i, pero es difícil debido a la no linealidad introducida por la proyección perspectiva. Sin embargo, y ya que el radio de la imagen virtual de la pupila es muy pequeño comparado con la distancia entre el ojo y la cámara, es posible utilizar una proyección afín para describir la relación entre la posición 3D de la pupila y su imagen 2D proyectada. La posición virtual de la pupila $p`_c$ se determina mediante (3) debido a la linealidad de la proyección afín.

$$p'_c = \frac{1}{N}\sum_{i=1}^{N} p_i \qquad (3)$$

Cuando p'_c se retro proyecta en el espacio 3D, el vector 3D resultante $\hat{p}'_c$ es paralelo a ${}^{C}T_A$ P'$_c$ donde ${}^{C}T_A$ es la matriz de transformación del sistema auxiliar de coordenadas 3D al sistema de coordenadas de la cámara. La posición en la imagen del centro de la pupila se usa luego para estimar la orientación del punto de vista.

Una vez calculado el centro de la pupila, y dado que se conocen tanto los parámetros de calibración de las cámaras, como el centro de la cornea respecto de cada una éstas, si el plano 3D formado por $\hat{p}'_{ci}$y k_{ci} no coincide, el eje óptico del ojo y el punto de vista puede ser calculado de forma única. Esto es debido a que el eje óptico del ojo es coplanario con la dirección del centro de la pupila $\hat{p}'_{ci}$ en la imagen a la cámara *i* y a que la dirección del centro de la cornea k_{ci} respecto a la cámara *i*.

El sistema finalmente propuesto, cuenta con 2 cámaras y 3 luces IR. Se usan 3 luces y no 2 como se había supuesto anteriormente para tener por seguro que el número total de primeras imágenes de Purkinje sea siempre suficiente para calcular el centro de la cornea, es decir, dos reflejos.

Es necesario realizar una calibración dependiente del usuario, consistente en calcular el ángulo entre el eje óptico y el eje visual, para lo que el usuario debe de mirar un punto en la pantalla durante unos instantes.

Para probar el sistema se sitúa al usuario a 45 *cm* de la pantalla, y después de realizar la calibración dependiente del usuario, se le pide que fije la vista en una serie de letras que aparecen en una cuadricula de 4x4 en la pantalla.

Los errores medios son de 0,49 *cm* en *x* y de 0,53 *cm* en *y*, ambas por debajo de 1 grado de precisión (0,63º en *x* y 0,686º en *y*), pero tienen una desviación típica grande: 0,135 *cm* en *x* y 0,181 *cm* en *y*. La desviación típica muestra que el algoritmo es muy dependiente del usuario, ya que se presume que tanta variación proviene de parámetros intra-oculares diferentes a los usados por el modelo propuesto.

Como limitaciones del sistema encontramos la necesidad de que el eje *x* de la cabeza deba ser paralelo al plano horizontal de la pantalla, provocando que el usuario no pueda girar su cabeza en el eje *y* mientras se utiliza el tracker, o la necesidad de que se realice la calibración dependiente del usuario para determinar el ángulo entre el eje óptico y el eje visual.

Otro de los sistemas es el propuesto por T. Ohno *et al.* [36], que plantean un sistema que solo necesita mirar a dos puntos en la pantalla para realizar la calibración dependiente del usuario.

Primero localiza la posición 3D de la pupila y de la primera imagen de Purkinje para después calcular el punto de mira gracias al modelo del ojo de la Figura 16.

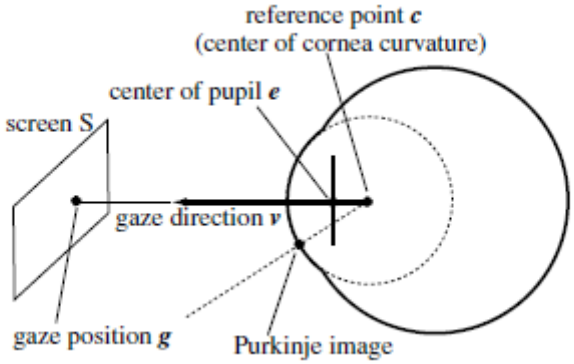


Figura 16. Modelo del ojo utilizado en [36]

Para detectar la posición de la pupila y de la primera imagen de Purkinje se toman con una cámara imágenes del ojo de gran tamaño, y posteriormente se generan imágenes de un cuarto de su tamaño, que se utilizarán en la segmentación de la misma mediante detección de regiones con valores de color similares, y en la detección de la pupila, donde para cada región segmentada es una candidata a ser la pupila. Después de esto, tiene lugar un doble ajuste de la elipse al contorno de la pupila detectada.

El siguiente paso es la detección de la imagen de Purkinje, buscando en las proximidades de la pupila un reflejo especular. En la mayoría de los trackers, se utilizan la posición calculada de la pupila y de la primera imagen de Purkinje, pero en [36] se opta por refinarlas mediante el modelo del ojo. La imagen de Purkinje necesita ser corregida debido a que la luz IR no está en el eje de la cámara, por lo que es necesario calcular la distancia entre imagen de Purkinje y el eje de la cámara. Por otro lado, los puntos del contorno deben ser corregidos por la refracción de la superficie de la cornea y la curvatura de la cornea.

Más tarde se realiza un ajuste de elipse de los contornos reales y se obtiene el centro de la elipse, lo que se traduce en la obtención del centro de la pupila *e*. En base a esto, el vector de mirada viene determinado por la diferencia entre las coordenadas del mundo correspondientes a *e* y al centro de curvatura *c.* Finalmente, la posición de la mirada se obtiene de la intersección de este vector con la pantalla *S*.

En este sistema, el movimiento permitido a la cabeza es escaso, ya que solo se utiliza una cámara, consiguiendo un área de movimiento posible de unos 4 cm^2 estando el usuario a 60 cm respecto a la pantalla.

La precisión varía entre 0,24 y 0,73 *cm* en la coordenada *x* y entre 0,31 y 0,94 *cm* en la coordenada *y*. Cabe destacar que en algunos casos, el error es mayor que 1 *cm*, y que en estos casos se recomienda repetir el proceso de calibración dependiente del usuario. En cuanto a los inconvenientes del sistema, cabe destacar la restricción de movimiento que sufre la cabeza (unos 4 cm^3 estando el usuario a 60 *cm* respecto a la pantalla) y la dificultad para localizar la primera imagen de Purkinje en personas que utilizan lentillas.

K.R. Park [37], por su parte, propone el uso de 4 luces IR en las esquinas de las pantallas, que producen 4 reflejos especulares en la pupila, lo que junto al centro de la pupila forman los

datos de entrada para obtener el punto de vista mediante la relación entre dicho centro y las 4 reflexiones.

Este sistema, posibilita el movimiento de cabeza, ya que utiliza unas gafas de visión a través (See-through), sobre las cuales va situada la cámara, que graba únicamente el ojo. Utilizan el Filtro Extendido de Kalman (EKF) para realizar el tracking continuo de movimiento 3D del ojo.

El EKF convierte las mediciones 2D de las posiciones del centro de la pupila y los cuatro reflejos especulares en estimaciones 3D de traslación y rotación del ojo usando un modelo de aceleración constante. Para esto se define un vector de estado a posteriori $\hat{x}(t)$ compuesto por la traslación, rotación, velocidades de traslación y rotación, y aceleración de traslación y rotación. Así, las ecuaciones del estado actual y la estimación son:

$$\hat{x}(t) = \hat{x}(t)^{-} + K(t)(y(t) - h(\hat{x}(t)^{-})) \quad (4)$$

$$P(t) = (\Phi(\Delta t)P(t-1)\Phi(\Delta t)^{T} + U)\, x\, \frac{\partial h}{\partial x(t)}|\hat{x}(t) - (I - K(t)) \quad (5)$$

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

Los datos resultantes de este sistema provienen de 23 puntos de vista de cada uno de los 120 usuarios que realizaron las pruebas. Dos pruebas distintas se hicieron:

1. Calculo del error en el punto de vista incluyendo solo movimiento de la cabeza
2. Calculo del error en el punto de vista incluyendo tanto movimiento de la cabeza como movimiento de los ojos.

Los resultados dieron lugar a un error de 0,48 *cm* en el caso 1 y de 0,51 *cm* en el caso 2, por lo que se afirma que el movimiento de la cabeza del usuario no afecta al método propuesto.

E.D. Guestrin *et al.* [19] plantean un método en el que varia el numero de cámaras y de fuentes de luz para reconstruir el eje óptico, partiendo del modelo matemático del ojo de la Figura 17.

En este trabajo, se muestran distintas configuraciones posibles en cuanto a número de cámaras y de luces:

1. Una cámara y una fuente de luz

Si los parámetros *R, K* y n_1 (radio de la cornea, distancia entre el centro de la pupila y el centro de curvatura de la cornea e índice de refracción del humor acuoso) son conocidos gracias a la calibración “pesada”, se debe incluir una restricción al modelo del ojo, para poder calcular el punto de vista con una cámara y una sola luz: la distancia entre el ojo y el punto nodal de la cámara es conocida, restringiendo el movimiento del ojo.

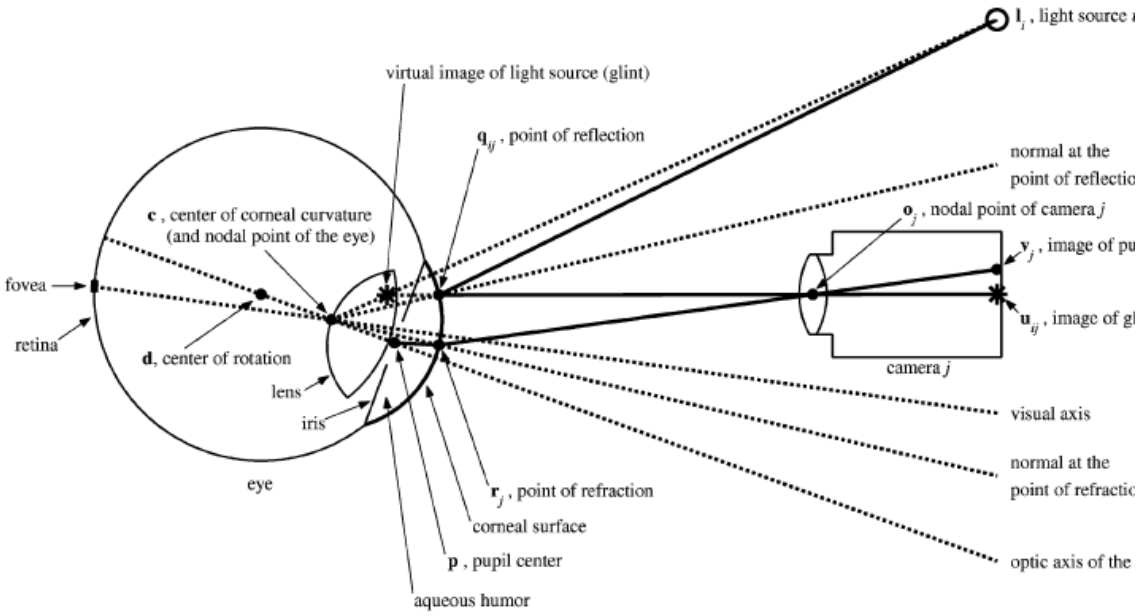


Figura 17. Modelo del ojo utilizado en [19]

2. Una cámara y dos fuentes de luz

Esta es la configuración más simple que permite reconstruir el eje óptico del ojo permitiendo el movimiento de cabeza. Para esto es necesario primero realizar un proceso de calibrado "pesado" dependiente del usuario para conocer los parámetros *R*, *K* y n_1.

3. Dos cámaras y 2 fuentes de luz

Es la configuración más simple para la reconstrucción del eje óptico sin la necesidad de conocer los parámetros propios de la calibración dependiente del usuario. Para esto es necesaria una calibración ligera.

Respecto a la calibración dependiente del usuario, encontramos dos alternativas:

1. Calibración pesada

El usuario fija su vista en 9 puntos distribuidos por la pantalla que aparecen secuencialmente. Usando las medias de la posición de las pupilas y los reflejos, los parámetros *R, K* y n_1 son optimizados para minimizar la suma de los cuadrados de los errores entre los puntos de la pantalla y la estimación de los puntos de vista.

2. Calibración ligera

Consiste en la fijación del punto de vista en un único punto en la pantalla para determinar la desviación angular entre el eje óptico y el eje visual del ojo ($\alpha_{ojo}, \beta_{ojo}$).

El sistema utilizado de entre los propuestos, cuenta con 2 luces IR, 1 cámara y el usuario puesto a una distancia del monitor de 65 *cm* aproximadamente. El movimiento de la cabeza permitido por este método es de 6 x 4 x 8 *cm*(ancho, alto, profundo), y la calibración "pesada" es necesaria. Este sistema produce un error en la localización del punto de vista menor que 1 *cm*.

En un trabajo posterior, los mismos autores [18] prueban un sistema con 2 cámaras y 4 luces IR. Se usan 4 luces para que al menos siempre se encuentren 2 reflejos especulares en el ojo. En éste sistema, se usa el contorno de la pupila y los dos reflejos especulares para calcular el eje óptico y el visual, y se utiliza la calibración ligera. Como resultado final, las pruebas realizadas mostraron un error menor que 7,27 *mm*.

Otro de los sistemas es el propuesto por T. Nagamatsu *et al.* [35], los cuales proponen un tracker que utiliza 2 cámaras y tres fuentes de luz para calcular el punto de vista. Esta solución necesita solamente de una calibración dependiente del usuario que consiste en fijar la vista en un único punto de la pantalla.

El centro de la pupila *B* viene determinado por la intersección entre 2 ecuaciones de rectas de 2 rayos provenientes de las 2 cámaras:

$$B = B_0'' + t\, t_0 \text{ de la cámara 0}$$
$$B = B_1'' + t\, t_1 \text{ de la cámara 1} \qquad (6)$$

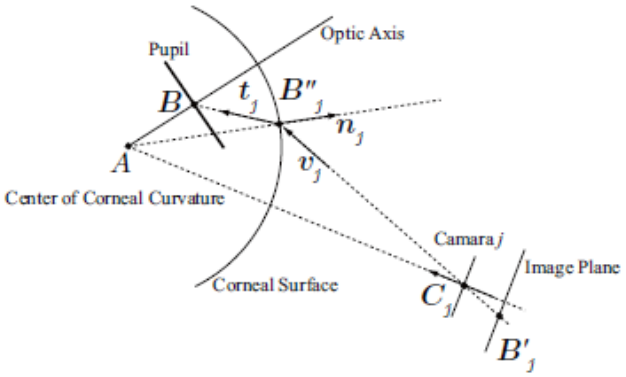


Figura 18.Esquema de refracción de la luz

Para calcular esto, es necesario calcular el punto B_j'', donde refracta un rayo (Figura 18) proveniente del centro de la pupila y que pasa a través del punto C_j e intersecta en el plano de la imagen de la cámara en el punto $B_{jz}' = (B_{jx}, B_{jy}, B_{jz})$, y el vector t_j refractado en el punto B_j'':

$$t_j = \left(-\rho n_j \cdot v_j - \sqrt{1 - \rho^2(1 - (n_j \cdot v_j)^2)}\right) n_j + \rho v_j \qquad (7)$$

en donde $v_j = (C_j - B_j')/|C_j - B_j'|$ es el vector incidente, $n_j = (B_j'' - A)/|B_j'' - A|$ es el vector normal en el punto de refracción y $\rho = n_1/n_2$, siendo n_1 el coeficiente de refracción del aire y n_2 el coeficiente de refracción efectivo del ojo. Una vez calculado el centro de la pupila, el eje óptico puede calcularse con la ecuación

$$x = A + t(B - A) \quad (8)$$

Para el cálculo del eje visual, se introduce el término posición primaria del ojo Figura 19), posición a partir de la cual cualquier posición ocular puede obtenerse mediante rotaciones.

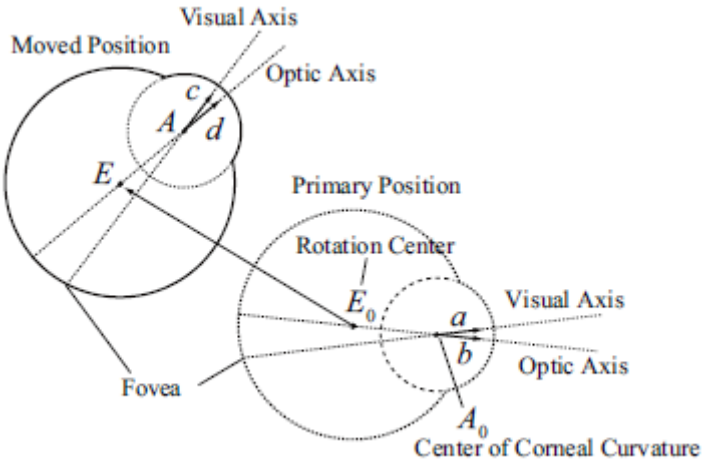


Figura 19. Posición primaria del ojo

La relación entre el eje visual de la posición primaria *a* y el eje visual de cualquier otra posición vienen determinada por la matriz de rotación *R* del vector dirección de *a*.

Para realizar la calibración, un usuario mira un punto *D* en la pantalla, por lo que el eje visual de cualquier posición $c = (D - A)/|D - A|$, mientras el eje óptico de cualquier posición $d = (B - A)/|B - A|$.

El eje visual en la posición primaria *a* viene determinado por la ecuación $x=A_0+t\,a$, y ya que *a* se determina de forma relativa a la cabeza, los cálculos siguientes fijan su sistema de coordenadas de referencia a *a.*

El vector del eje óptico en la posición primaria $b=(b_x,b_y,b_z)$ se puede obtener de:

$$P_b = (0; b_x, b_y, b_z) = QP_dQ^{-1} \quad (9)$$

en donde $Q = \left(\cos(-\psi/2); l_x \sin(-\psi/2), l_y \sin(-\psi/2), l_z \sin(-\psi/2)\right)$ es la rotación sobre el eje *l*(vector director unitario del eje de rotacion de *a* y *b*) mediante el ángulo –ψ:

$$l = \frac{a \times c}{|a \times c|} \quad (10)$$

$$\psi = \arccos\left(\frac{\mathrm{a} \cdot \mathrm{c}}{|\mathrm{a}||\mathrm{c}|}\right) \quad (11)$$

Por último, $P_d = (0; d_x, d_y, d_z)$ es la representación mediante un cuaternión del vector que forma el eje óptico $d = (d_x, d_y, d_z)$ en cualquier posición del ojo.

Una vez calculados *a*, *b*, *c* y *d,* se calcula la intersección entre la pantalla de esquinas D_{TL}, D_{TR}, D_{BL} y D_{BR} con normal n_d, y el eje visual de cualquier posición, para así obtener el punto de vista en la pantalla.

Las pruebas para comprobar la eficacia de este sistema se realizaron sobre 5 adultos sin gafas ni lentes de contacto, que debían fijar la mirada en 25 puntos que aparecen en la pantalla. En cuanto a los errores que produce este sistema, encontramos una media de desviación entre los puntos a observar y el eje visual de 0,95 *cm*, y entre los puntos a observar y el eje óptico de unos 2,1 *cm*.

Por otra parte, el movimiento permitido al usuario da un margen de 3 *cm* laterales, 2.5 *cm* verticales y 5 *cm* de profundidad estando a unos 57.5 *cm* de la pantalla.

D.H. Yoo *et al.* [55] utilizan una cámara y 4 luces IR situadas en las esquinas de la pantalla, para producir las correspondientes 4 primeras imágenes de Purkinje. Estos destellos están en una posición determinada respecto al centro de la pupila. Este método realiza un mapeado de esta relación sobre la pantalla para localizar el punto de vista (Figura 20).

Además de estas 4 luces, utiliza otra situada en el centro de la cámara para conseguir imágenes de la pupila mediante la técnica de pupila brillante.

Para detectar los 4 reflejos de Purkinje se realizan operaciones de umbralizado y erosión/dilatación estando las 4 luces IR del monitor encendidas, quedándose con las cuatro zonas o puntos más blancos.

La detección de la pupila se realiza mediante la diferencia entre imágenes de pupila brillante y pupila oscura en instantes de tiempo muy cercanos, de forma que la pupila tiene valores altos como resultado y por lo tanto es fácil extraerla mediante segmentación.

Es necesario calcular los puntos de fuga del los segmentos $\overline{AD}$ Y $\overline{BC}$ y el punto E, que resulta de la intersección de las diagonales del polígono ABCD (Figura 20):

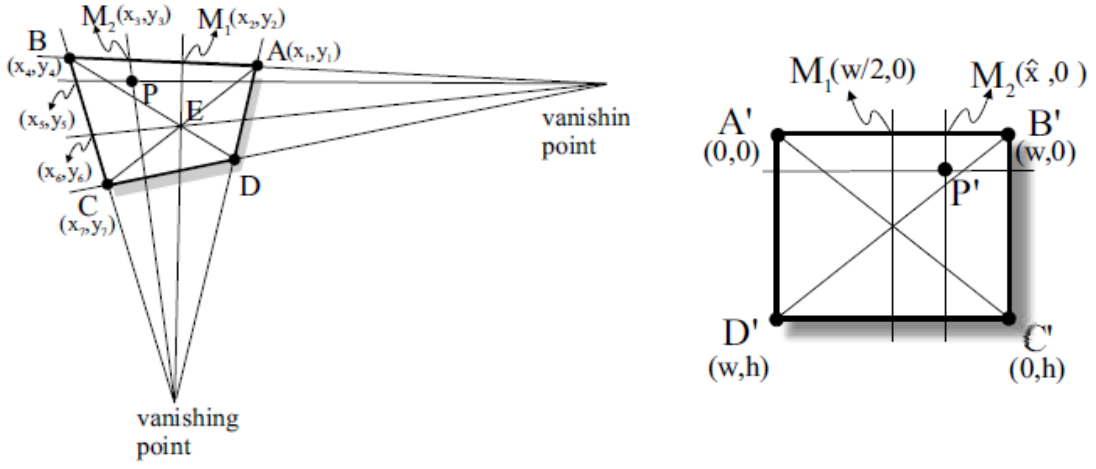


Figura 20. Esquema de puntos de fuga

La prolongación de la línea que une el punto de fuga y el punto E da lugar a $M_1 = (x_2, y_2)$ mientras que la prolongación de la línea que une el punto de fuga y P, da lugar a $M_2 = (x_3, y_3)$.

Gracias a estos dos puntos y a los puntos $A = (x_1, y_1)$ y $B = (x_4, y_4)$ se puede calcular el ratio de cruce invariante sobre la coordenada x en el espacio de la pupila

$$CR_x = \frac{(x_1y_2 - x_2y_1)(x_3y_4 - x_4y_3)}{(x_1y_3 - x_3y_1)(x_2y_4 - x_4y_2)} \quad (12)$$

Por otro lado, es posible calcular mediante los puntos $M_1 = (x_6, y_6)$, $M_2 = (x_5, y_5)$, $B = (x_4, y_4)$ y $C = (x_7, y_7)$ el ratio de cruce sobre la coordenada y en el espacio de la pupila

$$CR_y = \frac{(x_4y_5 - x_5y_4)(x_6y_7 - x_7y_6)}{(x_4y_6 - x_6y_4)(x_5y_7 - x_7y_5)} \quad (13)$$

Finalmente, para una pantalla de *w* pixeles de ancho y h pixeles de alto, la coordenada *x* y la coordenada *y* del punto de vista se calculan mediante

$$CR_{\hat{x}} = \frac{w \cdot CR_x}{1 + CR_x} \qquad CR_{\hat{y}} = \frac{h \cdot CR_y}{1 + CR_y} \quad (14)$$

Para probar el esquema propuesto, el usuario se sienta a unos 30-40 *cm* de la pantalla y mira a una de las zonas de la pantalla (queda dividida en 6x4 zonas), realizándose 200 mediciones del punto de vista del usuario, dando lugar a un error medio de 1,2 *cm* en *x* y de 0,92 *cm* en *y*.

En cuanto a los inconvenientes, es apreciable que la precisión del cálculo del centro de la pupila no es totalmente satisfactoria, ya que en ocasiones la segmentación mediante un umbral causa malos resultados porque dicho umbral no es correcto. Otra fuente de error es la asunción por la cual se utiliza una superficie plana para modelar la pupila, ya que la pupila tiene curvatura. Se afirma además que los resultados pueden variar dependiendo del movimiento de la cabeza del usuario, pero no se dan indicaciones numéricas de la magnitud de este cambio.

En una versión posterior [54], los autores utilizan 2 cámaras en lugar de una, de forma que una cámara se encarga de detectar la cara del usuario permitiendo el movimiento de la cabeza, y la otra se encarga de obtener las características de las pupilas y los reflejos especulares mencionados.

En este caso, se utiliza un método distinto para la localización de la pupila, que consta de tres fases:

1. **Proyecciones iterativas de la imagen**

   Mediante proyecciones iterativas de la imagen, es posible encontrar el borde de la pupila gracias a que, en estas proyecciones, la región de la pupila tiene los valores más bajos (Figura 21-Figura **24**)

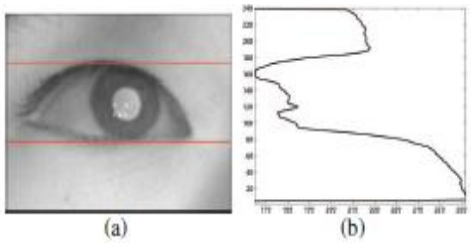


Figura 21. Paso 1 de la proyección recursiva de la imagen para el cálculo de la posición vertical del iris

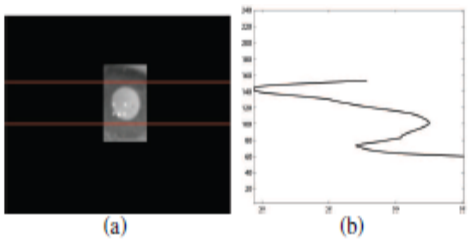


Figura 22. Paso 3 de la proyección recursiva de la imagen para el cálculo de la posición vertical del iris

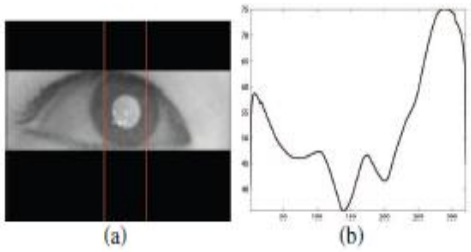


Figura 23. Paso 2 de la proyección recursiva de la imagen para el cálculo de la posición horizontal del iris

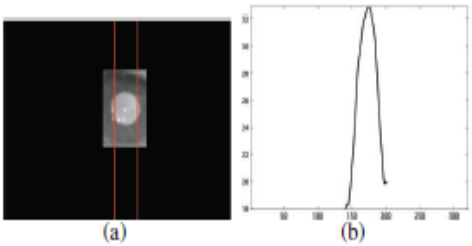


Figura 24. Paso 4 de la proyección recursiva de la imagen para el cálculo de la posición horizontal del iris

2. Etapa de contornos activos (snakes)

El borde localizado en la etapa anterior se usa como la posición inicial de la serpiente. El algoritmo de serpiente es un algoritmo iterativo basado en contorno activo que puede localizar el borde de mayor energía que se ajusta a un borde inicial, en este caso el borde localizado en la etapa de proyección iterativa.

3. Ajuste de elipse

El contorno resultante de la etapa anterior se ajusta a la forma de una elipse.

Para probar este sistema, un usuario sentado a una distancia de unos 40-50 *cm* tiene que mirar una serie de puntos que hay en la pantalla en forma de matriz de 5x5. Esta prueba se realizo unas 300 veces con movimiento de cabeza libre dentro de la zona mencionada, produciendo un error medio de 4,7 *mm* en *x* y 4,4 *mm* en *y*.

Z. Zhu *et al.* [58] proponen dos métodos: el primero de ellos permite obtener el punto de vista sin la necesidad de conocer ningún parámetro ocular dependiente del usuario (aunque más adelante veremos que no es cierto), mientras que el segundo permite el movimiento libre de la cabeza. Estos autores utilizan el modelo de ojo presentado en la Figura 25.

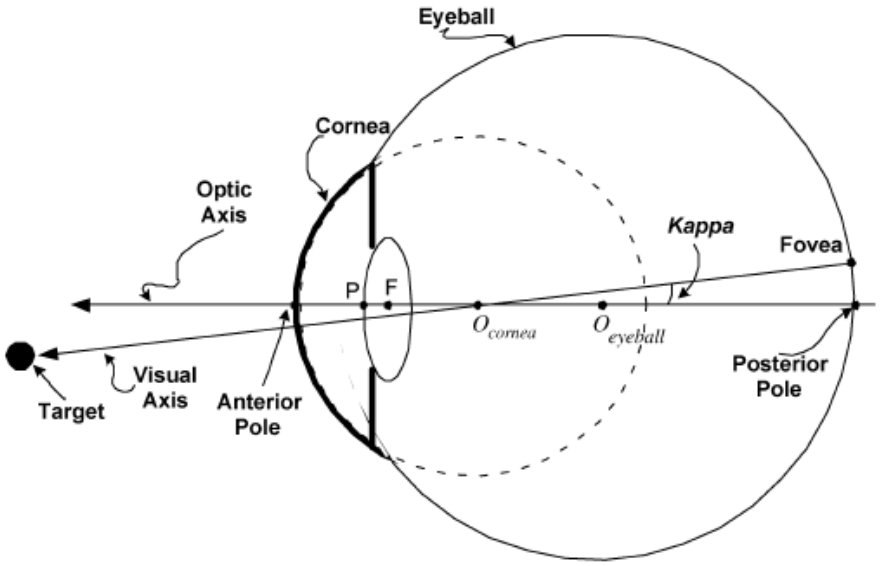


Figura 25. Sistema ocular utilizado por [58]

La primera técnica de Zhu *et al.* realiza una estimación directa del punto de vista 3D. En este método se afirma que si es posible conocer las coordenadas 3D tanto del centro de la pupila como del centro de la cornea, el eje visual puede ser calculado.

El primer paso necesario para el tracking es conocer las coordenadas 3D de los 2 reflejos sobre la superficie de la cornea ($L_1{}'$proveniente de la luz L_1 y $L_2{}'$proveniente de la luz L_2), algo que se consigue mediante el uso de dos cámaras.

Una vez obtenidos las posiciones de las dos cámaras y las posiciones de los dos reflejos producidos por éstas, $L_1{}'$ y $L_2{}'$, es posible conocer la localización del centro de curvatura de la cornea O_{cornea} intersectando las líneas $\overline{L_1 L_1'}$ y $\overline{L_2 L_2'}$.

El siguiente paso es calcular la imagen virtual del centro de la pupila P' gracias a la ley de refracción de la luz sobre superficies esféricas y al uso de 2 luces IR, tras lo que es posible calcular el eje óptico V_p y el visual V_v:

$$V_p = O_{cornea} + k(P' - O_{cornea}) \quad (15)$$

$$\overrightarrow{V_v} = M\overrightarrow{V_p} \quad (16)$$

en donde M es la matriz de rotación construida gracias a los ángulos de desviación entre $\overrightarrow{V_v}$ y $\overrightarrow{V_p}$, resultante tras un proceso de calibración en el que se le pide al usuario que observe 9 puntos S_i en la pantalla, almacenando los vectores $\overrightarrow{V_v}$ y $\overrightarrow{V_p}$ para cada punto.

Para comprobar la precisión de esta técnica, es necesario realizar tanto la calibración del sistema como la calibración dependiente del usuario (matriz *M*) para conocer el ángulo de desviación entre el eje óptico y el eje visual, proceso que suele tardar unos 5 s. Como resultado de las pruebas, se obtiene un error medio de 8,98 *mm* en el eje horizontal y 11,42 *mm* en el eje vertical estando el usuario a 35 *cm* de la cámara.

La segunda técnica de Zhu *et al.* realiza una estimación de la mirada basada en mapeo. Ésta consiste en el uso del vector PCCR para crear el vector $v = (v_x, v_y)$. Este vector, junto al punto donde se mira $S_s = (x_{gaze}, y_{gaze})$ con coordenadas de la pantalla, da lugar a la función $S_s = f(v)$:

$$\begin{cases} x_{gaze} = a_0 + a_1 * v_x + a_2 * v_y + a_3 * v_x * v_y \\ y_{gaze} = b_0 + b_1 * v_x + b_2 * v_y + a_3 * {v_y}^2 \end{cases} \quad (17)$$

Los coeficientes $a_0, a_1, a_2, a_3, b_0, b_1, b_2, b_3$ se estiman de un conjunto de parejas de vectores v y puntos de mirada en la pantalla recogidos en el procedimiento de calibración.

Esta técnica permite el movimiento de la cabeza, teniendo en cuenta la gran diferencia entre los vectores v calculados para ambos ojos. Esto tiene dos factores causales:

- Los ojos se encuentran en diferentes posiciones respecto de la cámara
- Los ojos giran de distinta forma para llegar al punto *S* debido a que están en posiciones distintas.

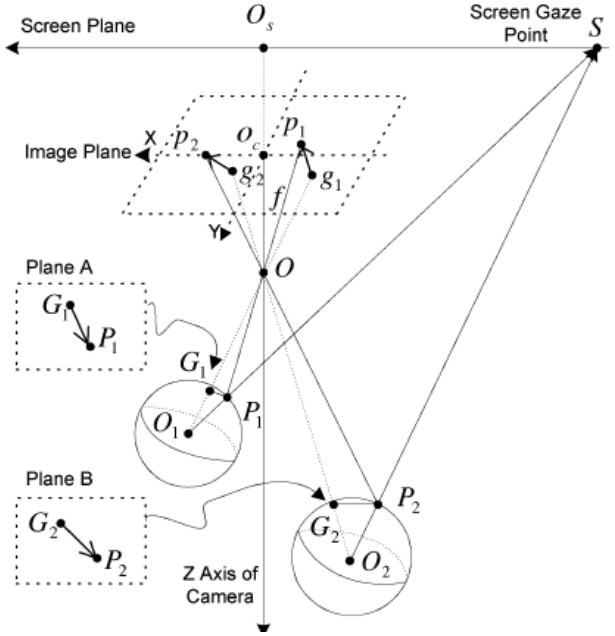


Figura 26. Representación del sistema ojos-pantalla-cámara en [58]

Los movimientos de cabeza provocan estas variaciones en v, por lo que son corregidos mediante el modelo propuesto en la Figura 26.

Para esta técnica de estimación, 7 usuarios probaron el sistema con un rango de movimiento de 200 *mm* en los ejes *x* e *y*, y de 300 *mm* en el eje *z*. Al igual que en la primera técnica, cuanto más separado está el usuario de la cámara, mas grande es el error producido: 2,72 *mm* en el eje horizontal y 3,19 *mm* en el eje vertical a una distancia de la cámara de 30 *cm*, y 16,6 *mm* en el eje horizontal y 22,5 *mm* en el eje vertical a 55 *cm* de la cámara. Como resultado de las pruebas, se obtiene un error medio de 9,18 *mm* en el eje horizontal y 10,83 *mm* en el eje vertical estando el usuario a 45 *cm* de la cámara.

Otro de los sistemas es el planteado por C. Yang *et al.* [52], el cual utiliza un sistema de pre-procesado para eliminar las influencias causadas por gafas u otros accesorios, para detectar con precisión las pupilas y los reflejos en la cornea. Este sistema utiliza la diferencia de grises

entre la cara, las pupilas y los puntos reflectantes en la córnea para detectar los ojos. En cuanto a hardware, utiliza una cámara y 4 luces IR que producen cuatro reflejos en la cornea.

El primer paso es eliminar las influencias causadas por el fondo de la imagen. Para esto se utilizan fuentes de luz IR para producir reflejos en la cornea (Figura 27), situando al usuario a una distancia de la cámara de entre 60 y 70 *cm*. De esta forma, es razonable pensar que la proyección integral en las direcciones verticales y horizontales son suficientes para acotar la región de la cara (x_{max}, y_{max}, y_{min}, y_{min}).

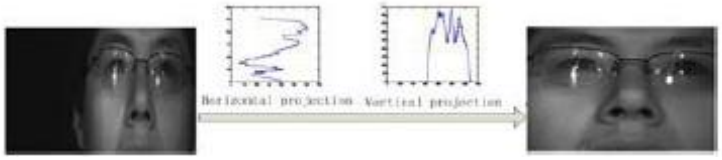


Figura 27. Extracción de la zona de la cara a analizar.

Tras extraer la zona de la cara gracias a los reflejos de las luces IR sobre la cornea, se dividen en cuatro las zonas de esta dependiendo de los reflejos que se observan en estas:

Una vez extraída la región de la cara, se extraen los reflejos mediante un umbralizado, posibilitando cuatro tipos de regiones, de las cuales se eliminarán las 2 primeras mediante un umbralizado por tamaño: reflexiones de luz IR sobre la montura de las gafas, reflexiones de luz IR sobre los cristales de las gafas, reflexiones sobre la cara o la montura de las gafas causadas por fuentes de luz mixtas y reflexiones de luz IR sobre la pupila.

El resultado es la obtención de las regiones correspondientes a reflejos P' candidatos a ser los producidos en la cornea.

El siguiente paso es calcular la zona de la pupila. Para esto, se utiliza la técnica de pupila oscura y se segmenta la región facial, utilizando la media de la intensidad de los pixeles alrededor de cada punto brillante. Seguidamente se eliminan tanto las zonas correspondientes al pelo como las zonas correspondientes a los agujeros de la nariz de acuerdo a su posición espacial respecto a otras zonas encontradas.

En el peor de los casos, todavía quedarían zonas propias de la montura de las gafas cuyo tamaño y posición fueran similares al de las regiones propias de las pupilas. Estas zonas se eliminan gracias a la posición de los agujeros de la nariz.

Para determinar el punto de vista, una vez conocida la posición tanto de las pupilas como de los cuatro reflejos en la superficie de la cornea, se utiliza el algoritmo de ratio de cruce invariante presentado en [55].

Para probar el sistema propuesto, diferentes sujetos, algunos de ellos con gafas, miraron una serie de puntos de coordenadas conocidas. Este método no propone el cálculo del punto de vista, aunque aporta resultados de las pruebas de gaze tracking con el algoritmo de ratio de cruce invariante, previo cálculo de las coordenadas mediante el método propuesto, aportando un error medio de 2,7 *mm* en el eje *x* y de 4,4 *mm* en el eje *y*, situando al usuario a 60 *cm* de la cámara. En cuanto al rango de movimiento permitido a la cabeza, este sistema permite un área de movimiento de 14x20x10 *cm*.

Los mismos autores, Yang *et al*., proponen en un artículo [51] un sistema basado en el uso de una cámara y 4 luces IR, que tras calcular la posición de la pupila y los 4 reflejos sobre la cornea producidos por las luces, utiliza el algoritmo de ratio de cruce invariante para obtener el punto de vista del usuario.

La detección de los ojos se realiza mediante una diferencia de imágenes consecutivas, donde se producen parpadeos, lo que implica que no puede haber movimiento de cabeza por parte del usuario. Una vez detectadas las zonas de los ojos, se procede a calcular el área correspondiente a la pupila de uno de ellos, para lo que se utilizan las proyecciones integrales horizontal y vertical, utilizadas en [52].

Para calcular la posición de los reflejos corneales, se utiliza una técnica de umbralizado adaptativo de la imagen de grises de la zona de la pupila. El umbral comienza tomando el valor de gris más alto, decreciendo iterativamente hasta que se detecten los 4 reflejos, momento en el cual se calculan sus centroides. El siguiente paso es rellenar los reflejos con los valores de gris que se encuentran en sus alrededores. Este paso se realiza para después poder detectar el borde de la pupila mediante el operador Canny. Después se utilizan las coordenadas del borde para ajustarles una elipse cuyo centro es el centro de la pupila. Seguidamente, se calcula el punto de vista $P'(\hat{x},\hat{y})$ mediante el algoritmo del ratio de cruce invariante.

Con el fin de calcular el ángulo formado por el eje óptico y el eje visual del ojo, se realiza un proceso de calibración dependiente del usuario por el cual el usuario tiene que mirar a 5 puntos en la pantalla. De las 5 mediciones realizadas, se obtiene el error medio tanto en *x* como en *y*. Esto se utiliza posteriormente para corregir el punto de vista P', formando $P''(x,y)$, gracias a las coordenadas *x* e *y* calculadas en (18).

$$x = \hat{x} + \sum_{i=1}^{5}\left(\frac{1/dx_i}{\sum_{i=5}^{5} 1/dx_i}\cdot Ex_i\right) \qquad y = \hat{y} + \sum_{i=1}^{5}\left(\frac{1/dy_i}{\sum_{i=5}^{5} 1/dy_i}\cdot Ey_i\right) \qquad (18)$$

en donde $Ex_i = x_i - \hat{x_i}$, $dx_i = |\hat{x} - \hat{x_i}|$ y $dy_i = |\hat{y} - \hat{y_i}|$.

Para probar el sistema, 20 usuarios miraron 16 puntos durante 1,5 s cada uno. Como resultado, se produce un error medio de 3,42 *mm* en el eje *x* y de 3,14 *mm* en el eje *y*, situando al usuario a una distancia de 60 *cm* de la cámara.

En un trabajo posterior [42], los mismos autores plantean un sistema con una sola cámara y 2 luces, que utiliza el triangulo formado por las reflexiones de las dos luces sobre las corneas y el centro de la pupila para calcular el punto de vista mediante triangulaciones.

Una vez calculado el triangulo formado por los reflejos corneales $\mathrm{V_{L1}}$y $\mathrm{V_{L2}}$ y el centro de la pupila P, hay que ajustar sus componentes z, ya que sus componentes z son muy parecidas, se toma una de ellas como componente z de las 3, en este caso $z_{\mathrm{V_{L1}}}$. Así, se aproxima el plano formado por los puntos P,$\mathrm{V_{L1}}$ y $\mathrm{V_{L2}}$ al formado por $\mathrm{P''}$, $\mathrm{V_{L1}}$ y $\mathrm{V''_{L2}}$. Así mismo, de la misma forma que el centro de curvatura de la cornea C, venia dado por la intersección entre las líneas $\overline{\mathrm{L_1V_{L1}}}$ y $\overline{\mathrm{L_2V_{L2}}}$, ahora C′ viene determinado por la intersección entre las líneas $\overline{\mathrm{L_1V_{L1}}}$y $\overline{\mathrm{L_2V''_{L2}}}$ (Figura 28).

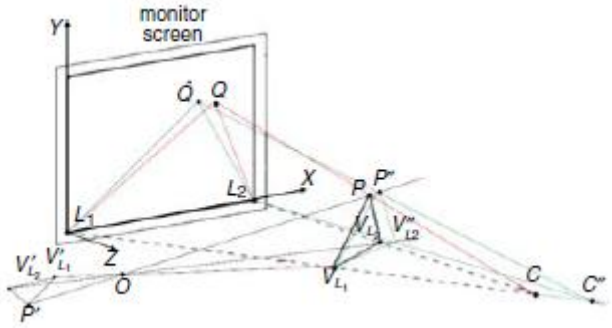


Figura 28. Esquema del sistema utilizado en [42]

El punto de vista $\hat{Q} = (\hat{x}_Q, \hat{y}_Q)$ es la intersección entre el plano del monitor y la línea $\overline{C''P''}$, cuyas coordenadas se calculan como:

$$\begin{cases} \hat{x}_Q = \dfrac{x'_{V_{L1}} - x'_p}{x'_{V_{L1}} - x'_{V_{L2}}} (x_{L_2} - x_{L_1}) \\ \hat{y}_Q = \dfrac{y'_{V_{L1}} - y'_p}{x'_{V_{L1}} - x'_{V_{L2}}} (x_{L_2} - x_{L_1}) \end{cases} \quad (19)$$

en donde x_{L_1} es 0 y x_{L_2} es la anchura del monitor en pixeles.

Los puntos $P'(x'_p, y'_p)$, $V'_{L1}(x'_{V_{L1}}, y'_{V_{L1}})$ y $V'_{L2}(x'_{V_{L2}}, y'_{V_{L2}})$ se calculan mediante el método utilizado en [52].

Para probar este método, el usuario debe realizar primero una calibración en la que debe de mirar 5 puntos en la pantalla, para después fijarse en 16 puntos, sobre los que se calcula la eficacia del sistema. Éste produce un error máximo de 1,14 *cm* y un error medio de 0,82 *cm* a una distancia de 65 *cm* de la cámara, permitiendo un rango de movimiento de la cabeza del usuario de 20x17x10 *cm*.

3. CALIBRACIÓN DE LA CÁMARA

Primeramente, y antes de explicar el método utilizado para la calibración de la cámara, es necesaria la explicación matemática y geométrica del modelo de cámara.

3.1. Modelo de la cámara

Matemáticamente, la formación de una imagen se puede definir como la proyección de una escena 3D sobre un plano.

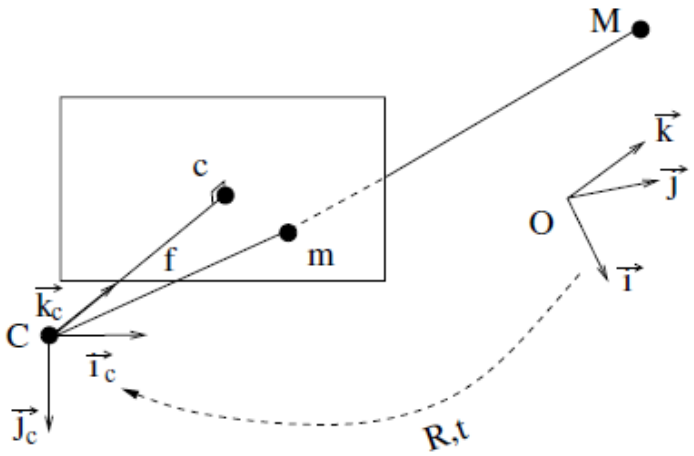


Figura 29. Modelo de escena 3D

En la Figura 29 se observa el modelo de escena 3D propio del modelo de proyección perspectiva, donde *O* representa el sistema de coordenadas del mundo, mientras que *C* representa el sistema de coordenadas de la cámara. Por otra parte, *M* es un punto 3D y m es la proyección de dicho punto en el plano antes mencionado.

La transformación entre las coordenadas 3D del punto *M=[X,Y,Z]* expresadas en el sistema de coordenadas euclídeo y las coordenadas del punto 2D correspondiente en el espacio de la imagen $\tilde{m}$*=[u,v]*, viene dada por (20)

$$s\tilde{m} = P\tilde{M} \qquad (20)$$

en donde *s* es un factor de escala, $\tilde{m}$ y $\tilde{M}$ son las coordenadas homogéneas de los puntos *m* y *M* respectivamente y *P* es una matriz de proyección, la cual se descompone de la siguiente forma:

$$P = K[R|t] \qquad (21)$$

en donde *K* es la matriz de parámetros intrínsecos y $[R|t]$ es la matriz 3x4 de parámetros externos que corresponde con la transformación Euclidea desde el sistema de coordenadas del mundo al sistema de coordenadas de la cámara: R representa la matriz de rotación de 3x3 mientras que *t* representa la traslación.

Dada *P*, tenemos el vector columna p_i =[P_{i1}, P_{i2}, P_{i3}]^T y existen dos restricciones básicas que más adelante utilizaremos:

$$p_1^T K^{-T} K^{-1} p_2 = 0 \qquad (22)$$

$$p_1^T K^{-T} K^{-1} p_1 = p_2^T K^{-T} K^{-1} p_2 \qquad (23)$$

Por otra parte, la matriz *K* de parámetros intrínsecos se descompone de la siguiente forma:

$$K = \begin{bmatrix} \alpha_u & \gamma & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \qquad (24)$$

en donde:

- α_u y α_v son el factor de escala en las coordinadas u y v respectivamente. Estos son proporcionales a la distancia focal f de la cámara: $\alpha_u = k_u$ f y $\alpha_v = k_v$ f, en donde k_u y k_v son el numero de píxeles por unidad de distancia en las direcciones *u* y *v* respectivamente. Se suele asumir que estos dos factores son iguales, por lo que la razón de aspecto de los pixeles sería cuadrada.

- c=[u_0 , v_0]^T representa las coordenadas de la imagen de la intersección entre el eje óptico y el plano de la imagen, más conocido como punto principal. Se suele asumir que este punto principal está en el centro de la imagen.

- γ indica la oblicuidad de *u* y *v*, y es distinto de cero solo si las direcciones *u* y *v* no son perpendiculares.

Se dice que la cámara esta calibrada cuando sus parámetros internos son conocidos, aunque el modelo anterior puede no ser suficiente para representar con fidelidad el modelo de la cámara con la que tratamos. Esto se debe a que no se tiene en cuenta la posible distorsión debida a la lente de la cámara. Esta distorsión puede ser modelada como una deformación 2D de la imagen:

Siendo $\breve{u} = [\breve{u}, \breve{v}]^T$ las coordenadas de la imagen de un pixel distorsionado, y $\breve{x} = [\breve{x}, \breve{y}]^T$ las correspondientes coordenadas normalizadas tales que $\breve{u} = u_0 + \alpha_u \breve{x}$ y $\breve{v} = v_0 + \alpha_u \breve{y}$, en donde u_0, v_0, α_u y α_v son los parámetros intrínsecos obtenidos de la matriz *K* anteriormente citada.

La distorsión sufrida por la imagen se descompone de la siguiente forma:

$$\begin{aligned} \breve{x} &= x + dx_{radial} + dx_{radial} \\ \breve{y} &= y + dy_{radial} + dy_{radial} \end{aligned} \qquad (25)$$

en donde la distorsión radial se aproxima como

$$\begin{aligned} dx_{radial} &= (1 + k_1 r^2 + k_2 r^4 + \cdots) x \\ dy_{radial} &= (1 + k_1 r^2 + k_2 r^4 + \cdots) y \end{aligned} \qquad (26)$$

,siendo $r = ||x|| = \sqrt{(x^2 + y^2)}$, y k_1 y k_2 los coeficientes de distorsión radial.

Por su parte, la distorsión tangencial (27) tiene mucha menos influencia e incluso suele ser ignorada. Esta distorsión se calcula como:

$$dx_{tangencial} = \begin{bmatrix} 2 \times dist1 \times x \times y + dist2(r^2 + 2x^2) \\ dist1(r^2 + 2y^2) + 2 \times dist2 \times x \times y \end{bmatrix} \qquad (27)$$

siendo *dist1* y *dist2* los coeficientes de distorsión tangencial.

3.2. Tipo de calibraciones

Hay varios tipos de calibraciones de cámara:

- **Calibración fotogramétrica**

  Se utiliza un objeto para calibrar la cámara. Dicho objeto tiene una geometría 3D conocida con gran precisión, que suelen ser dos o tres planos ortogonales entre ellos. Esta calibración puede hacerse de forma muy eficiente.

- **Auto-calibración**

  Se basa en mover la cámara en una escena estática, sin la necesidad de utilizar ningún objeto predeterminado. Si las imágenes captadas son de la misma cámara, una serie de correspondencias 2D entre las mismas son suficientes para obtener los parámetros intrínsecos y extrínsecos de la cámara.

  Este método es muy flexible, pero no está demasiado consolidado, y no es siempre posible obtener resultados correctos con total confianza.

- **Otros**

  Existen otros como los puntos de fuga para las direcciones ortogonales o la calibración desde la rotación pura, aunque no llegan al nivel de uso de los primeros dos, sobre todo, al nivel de la calibración fotogramétrica.

El método que se propone es una mezcla entre la calibración fotogramétrica, ya que utiliza un objeto con un patrón impreso sobre él, y la auto calibración, ya que se usan medidas 2D en lugar de medidas 3D propias de varios planos ortogonales entre sí.

La elección de este método se debe a que es muy flexible y tiene mayor robustez que la auto-calibración propiamente dicha.

3.3. Método de calibración propuesto

Dentro del tipo de calibración propuesto, uno muy utilizado y con muy buenos resultados, el método propuesto por Zhang [57]. En resumen, los pasos que debe de seguir la calibración esta calibración son:

1. Impresión del patrón gráfico para la calibración: en nuestro caso será un tablero de ajedrez.
2. Tomar una serie de imágenes del patrón desde distintas posiciones y orientaciones.
3. Detección de los puntos característicos (esquinas en el tablero de ajedrez) en las imágenes capturadas.
4. Obtención de los parámetros intrínsecos y extrínsecos de la cámara mediante una solución de forma cerrada.
5. Estimar los parámetros de distorsión radial y tangencial
6. Refinar todos los parámetros por minimización.

En las siguientes líneas se describe el procedimiento que sigue este método para calcular los parámetros intrínsecos y extrínsecos de la cámara, así como los parámetros de distorsión de la misma.

$$B = K^{-T}K^{-1} \qquad (28)$$

Dada la matriz *K* de parámetros intrínsecos, es posible calcular la matriz *B*(28), de la cual se extraen los parámetros extrínsecos finalmente. Por otra parte, siendo $b=[B_{11},B_{12},B_{22},B_{13},B_{23},B_{33}]^T$ el vector que representa B(ya que es simétrica), y siendo *P* la matriz de proyección antes mencionada, y el vector columna $p_i =[P_{i1}, P_{i2}, P_{i3}]^T$, se forma la ecuación

$$p_i{}^T\, B\, p_i = v_{ij}^T b \qquad (29)$$

en donde

$$v_{ij}^T = \left[p_{\ i1}p_{j1}, p_{i1}p_{j2} + p_{\ i2}p_{j1}, p_{\ i2}p_{j2}, p_{\ i3}p_{j1} + p_{\ i1}p_{j3}, p_{\ i3}p_{j2} + p_{\ i2}p_{j3}, p_{\ i3}p_{j3}\right]^T \qquad (30)$$

El siguiente paso es utilizar las restricciones (22) y (23) para reescribirlas y formar un sistema de ecuaciones:

$$\begin{bmatrix} v_{12}^T \\ (v_{11} - v_{22})^T \end{bmatrix} b = 0 \qquad (31)$$

Si para la calibración utilizamos *n* imágenes, apilando estas *n* imágenes en *n* fórmulas como la anterior tendríamos:

$$V\, b = 0 \qquad (32)$$

en donde *V* es una matriz de 2*n* x 6, y b es el vector que aparece en (29) y que representa a B:

$$B = \begin{bmatrix} B_{11} & B_{12} & B_{13} \\ B_{12} & B_{22} & B_{23} \\ B_{13} & B_{23} & B_{33} \end{bmatrix} \qquad (33)$$

$$b = [B_{11}, B_{12}, B_{22}, B_{13}, B_{23}, B_{33}]^T \qquad (34)$$

Tendríamos así un sistema de *n* ecuaciones, que tras resolver, nos daría la matriz *B*. Siendo λ un factor de escala arbitrario y descomponiendo la matriz B cómo $B = \lambda\, A^{-T}A$, es posible calcular fácilmente los parámetros intrínsecos:

$$v_0 = (B_{12}B_{13} - B_{11}B_{23})/(B_{11}B_{22} - B_{12}^2) \quad (35)$$

$$\lambda = B_{33} - [B_{13}^2 + v_0(B_{12}B_{13} - B_{11}B_{23})]/B_{11} \quad (36)$$

$$\alpha_u = \sqrt{\lambda/B_{11}} \quad (37)$$

$$\alpha_v = \sqrt{\lambda B_{11}/(B_{11}B_{22} - B_{12}^2)} \quad (38)$$

$$\gamma = -B_{12}{\alpha_u}^2\alpha_v/\lambda \quad (39)$$

$$u_0 = \frac{\gamma v_0}{\alpha_u} - B_{13}{\alpha_u}^2/\lambda \quad (40)$$

Para obtener una única solución *b* correcta, necesitamos 4 o más imágenes al hacer la calibración, aunque en la mayoría de los trabajos vistos en la bibliografía se realiza la calibración con 5 imágenes, por lo que tomaremos este número como estándar.

Por otra parte y en lo que respecta a los parámetros extrínsecos, una vez calculados *b y* los parámetros intrínsecos, podemos obtener *A*, de donde calculamos:

$$\lambda = 1/||A^{-1}p_1|| \quad (41)$$

$$r_1 = \lambda A^{-1}p_1 \quad (42)$$

$$r_2 = \lambda A^{-1}p_2 \quad (43)$$

$$r_3 = r_1\, x\, r_2 \quad (44)$$

$$t = \lambda\, A\, p_3 \quad (45)$$

siendo r_1, r_2, r_3 y t rotación en *x*, rotación en *y*, rotación en *z* y traslación, respectivamente.

Una vez obtenidos los parámetros extrínsecos e intrínsecos de la cámara, se procede al cálculo de la estimación de los parámetros de distorsión.

Al estimar los parámetros intrínsecos, es posible obtener las coordenadas ideales *(u,v)* de un punto en pixeles, mientras que $(\breve{u}, \breve{v})$ son las coordenadas reales del mismo punto en pixeles y $(\breve{x}, \breve{y})$ son las coordenadas ideales normalizadas sin distorsión.

Así, calculando las siguientes ecuaciones para cada punto o esquina del tablero de ajedrez, se obtienen los coeficientes:

$$\begin{bmatrix}(u-u_0)(x^2+y^2) & (u-u_0)(x^2+y^2)^2\\ (v-v_0)(x^2+y^2) & (v-v_0)(x^2+y^2)^2\end{bmatrix}\begin{bmatrix}k_1\\k_2\end{bmatrix}=\begin{bmatrix}\breve{u}-u\\ \breve{v}-v\end{bmatrix} \quad (46)$$

, con lo que tendríamos 2*mn* ecuaciones, siendo *m* el numero de imágenes y *n* el número de puntos de cada imagen. Simplificando, el sistema de ecuaciones anterior se puede escribir comoe $DK = d$, en donde K=$[k_1,k_2]^T$. En base a esto la solución lineal por mínimos cuadrados se reescribe como

$$K = (D^T D)^{-1} D^T d \quad (47)$$

Finalmente, y tras obtener los parámetros de distorsión, se refinan los parámetros obtenidos mediante minimización, gracias a la siguiente función:

$$\sum_{i=1}^{n}\sum_{j=1}^{m} ||m_{ij} - \breve{m}(K, k1, k2, R_i, t_1, M_j)||^2 \quad (48)$$

, siendo $\breve{m}(K, k1, k2, R_i, t_1, M_j)$ la proyección del punto M_j en la imagen *i*, con los parámetros intrínsecos y extrínsecos, y los coeficientes de distorsión radial calculados anteriormente: *K* es la matriz de parámetros intrínsecos, *k1* y *k2* los coeficientes de distorsión radial, *R* es el vector de rotación y t es la traslación.

Esta minimización no es un problema lineal y es resuelto con el Algoritmo de Levenberg-Marquardt [49].

4. LOCALIZACIÓN DE LA ESFERA

En este capítulo se realiza una comparativa entre tres métodos bases para localizar dos parámetros propios de la esfera: centro y radio. Estos tres métodos son la transformada de Hough, Camshift y el método de detección de momentos. De estos tres, Hough y momentos necesitan una etapa de preprocesado, mientras que en Camshift no es necesaria.

Una vez comparados estos tres métodos, el que mejores resultados proporciona, sirve como base para una primera localización que se refina en un proceso posterior mediante una minimización del error.

Tras la minimización del error se procede a solucionar dos errores de naturaleza geométrica que provocan la desviación del centro de la esfera calculado.

4.1. Etapa de preprocesado

La etapa de preprocesado tiene como fin la obtención de una imagen transformada I' a partir de la original I (Figura 30), de forma que sea más fácil extraer información relevante.

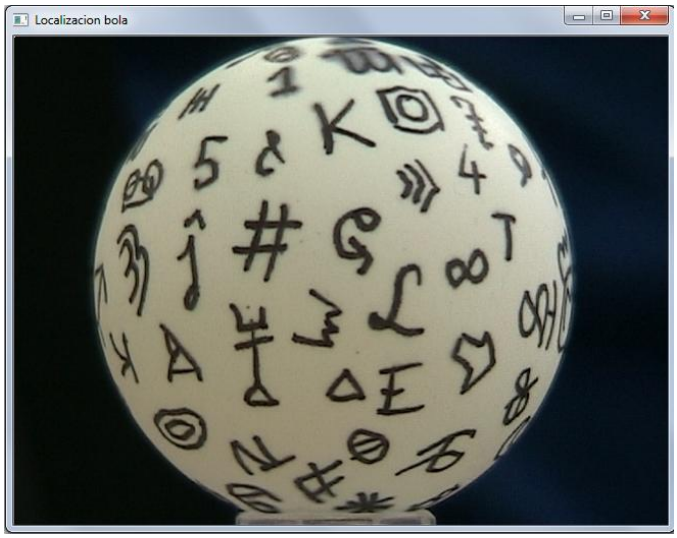


Figura 30. Imagen original I

Las etapas que se explican a continuación tienen su base en la información de color que nos proporciona I. Para explotar esta información de una forma más eficaz, utilizamos el modelo de color HSV. Este modelo de color separa la imagen en tres canales: Hue, Saturation and Value, o lo que es lo mismo matiz, saturación y valor.

Durante la etapa de preprocesado, en lugar de utilizar una única imagen de la esfera, se analizan 20 frames para obtener la imagen I', de forma que la robustez de esta etapa se ve incrementada debido a la menor influencia que tienen de esta forma posibles cambios en la iluminación o la presencia de objetos blancos cercanos al contorno de la esfera. Cualquier otro objeto de color blanco que se encuentre en la escena se elimina mediante un proceso de detección de contornos, como veremos más adelante.

Cada uno de los 20 frames que intervienen en esta etapa sufren, pues, los siguientes procesos:

1. **Transformación del espacio de color de la imagen**

   La imagen original I pasa del espacio de color RGB al HSV para aprovechar mejor la información de color.

2. **Extracción del canal Value**

   Mientras que Hue indica la gama de color y Saturation indica la pureza del color, Value indica un tono más claro o un tono más oscuro. Debido a esto, el uso de la componente Value es la más indicada para localizar una esfera de color blanco y para servir como base para los siguientes pasos (Figura 30).

3. **Operaciones lógicas con imágenes**

   Se realizan una serie de combinaciones de imágenes para diferenciar el contorno de la esfera del resto de la imagen (Figura 33). Esta combinación tiene una explicación completamente empírica, y para realizarla se ha buscado únicamente maximizar todo lo posible la diferencia entre la esfera y el fondo (Figura 34). Como imágenes de entrada de estas operaciones, tenemos el canal Value de la imagen original V y la inversa de esta iV (Figura 32):

   $$Imagen\ resultante = (V - iV) + V + V \qquad (49)$$

4. **Umbralizado**

   El siguiente proceso a realizar es la umbralización de la imagen resultante en el paso anterior. De esta forma, cualquier pixel con un color muy cercano al blanco, pasa a formar parte de la esfera (Figura 35).

   Este proceso no evita los agujeros provocados por las marcas arbitrarias realizadas en la superficie de la esfera, por lo que en los siguientes pasos será necesario eliminarlos en la medida que sea posible.

5. **Operación de cierre de la imagen**

   Las operaciones morfológicas de dilatación (Figura 36) y erosión (Figura 37) permiten ampliar y reducir respectivamente las regiones blancas de una imagen en blanco y negro. Mediante la combinación de estos operadores, es posible realizar el cierre o apertura de una imagen: mientras que la aplicación de la erosión y posteriormente la dilatación a una misma imagen permite la apertura y posibilita la desaparición de puntos sueltos o estructuras finas, la aplicación de la dilatación y posteriormente la erosión permite el cierre, posibilitando esta ultima que se rellenen los huecos negros en el interior o el contorno de una figura blanca.

   Debido a los pequeños agujeros que aparecen en el contorno y en el interior de la imagen resultante del proceso de umbralizado, se aplica una operación de cierre de 5

iteraciones a ésta. Se realizan 5 iteraciones, porque es el número de iteraciones máximo que permite el cierre de agujeros sin que el contorno de la esfera se vea demasiado afectado por la dilatación, evitando que la esfera localizada tenga unas un aspecto en el que se pierden un poco las curvas del contorno, dejando en su lugar un contorno más recto.

6. Localización del contorno de la esfera

El siguiente paso es buscar el contorno de la figura de mayor tamaño en la escena, correspondiente al contorno de la esfera.

7. Inserción en el buffer de esferas

Con el fin de eliminar posible ruido durante este proceso, sobre todo en el contorno de la esfera, se crea una imagen auxiliar que hace las veces de buffer de esferas detectadas. Cada vez que se detecta un contorno en el paso anterior, el valor los píxeles del buffer correspondientes a los pixeles de este contorno se ve incrementado. De esta forma, al final del preprocesado, el buffer tendrá en colores más blancos los píxeles que más veces hayan aparecido en los contornos de las diferentes esferas localizadas.

Finalmente, el último proceso necesario es la umbralización del buffer de esferas, de forma que únicamente los valores con mayor número de apariciones son considerados como contorno de la esfera de la imagen resultante de la etapa de preprocesado I' (Figura 38).

Figura 31. Componente Valor o imagen *V*

Figura 32. Imagen $\boldsymbol{iV}$

Figura 33. $\boldsymbol{V - iV}$

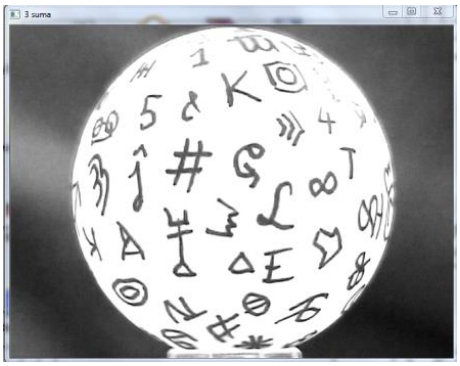


Figura 34. $(\boldsymbol{V - iV}) + \boldsymbol{V} + \boldsymbol{V}$

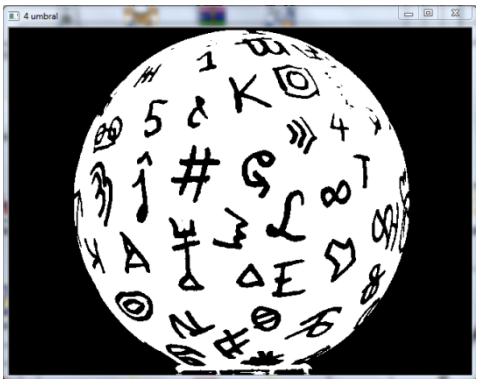


Figura 35. Umbralizado

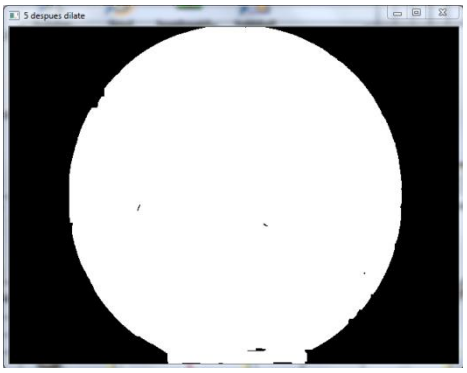


Figura 36. Dilatado

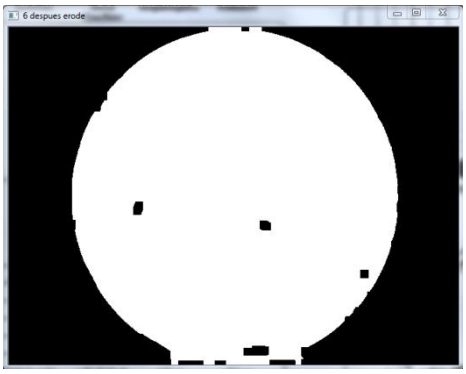


Figura 37. Erosión

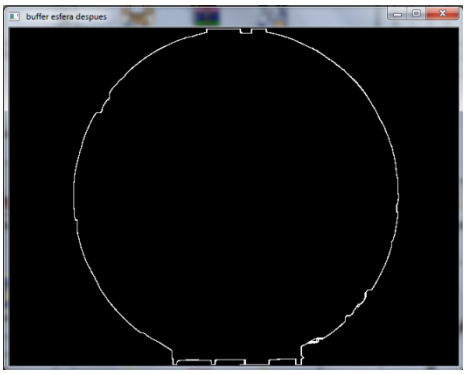


Figura 38. Contorno resultante de l´

4.2. Métodos de localización

Los métodos analizados para obtener el mejor rendimiento a la hora de localizar la esfera son tres: círculos de Hough, el método de los momentos y Camshift.

4.2.1. Círculos de Hough

La transformada de Hough [14] es un método de ajuste de contornos a figuras parametrizables, como pueden ser líneas o círculos. Esta técnica es muy conocida por ser robusta frente al ruido y a la existencia de huecos en la frontera del objeto, cualidades que se ajustan a nuestras necesidades. Sin embargo, como veremos en pruebas posteriores, la robustez que se presume a este método no es comparable a la alcanzada por el método de momentos.

Además de la etapa de preprocesado comentada en la sección 4.1, este método necesita en última instancia de un proceso de suavizado de la imagen para su correcto funcionamiento, algo indispensable para que adquiera una robustez que, de por sí mismo, no tiene.

Este método consiste básicamente en recorrer todos los puntos del contorno de la figura a caracterizar, y para cada uno de ellos plantea los infinitos círculos que pasan por ese punto, con distintos radios y distintos centros, acumulando un voto en un espacio de parámetros tridimensional (coordenada x_n del centro, coordenada y_n del centro y radio r_n) para aquellas curvas que cumplan con la ecuación del círculo

$$(x - x_0)^2 + (y - y_0)^2 = r^2 \qquad (50)$$

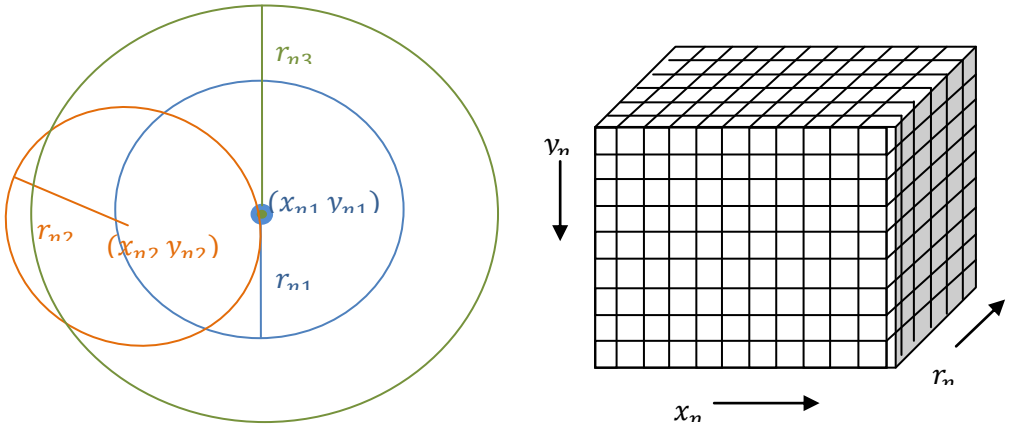


Figura 39. Representación de posibles círculos que pasan por el punto (x_{n1}, y_{n1}) (izda.) y sistema de votos (dcha.)

Para cada punto de frontera de un segmento que cumple con la ecuación del circulo, se añade un voto a la matriz tridimensional de votos, de forma que una vez barridos todos los puntos de la imagen candidatos a ser centro de un círculo, obtendremos en la matriz de votos el círculo más probable en el cuadrante que tenga más votos (Figura 39).

4.2.2. El método de los momentos

El cálculo de momentos [50] en imágenes ha sido muy usado en aplicaciones de análisis de imágenes y tiene como principal ventaja las medidas de aspecto invariantes. Las dos que se van a utilizar en este método son el área del objeto localizado, y el centro del mismo. Gracias al área, es posible calcular el radio de la circunferencia de la esfera.

Después del preprocesado, como se ha visto, obtenemos el contorno que representa la esfera. Este método se encarga de calcular los momentos espaciales de la figura interior a dicho contorno.

El primer paso es el cálculo del área de la esfera, mediante el cálculo del momento de orden 0

$$a = \sum_{i=1}^{n} m_i \qquad (51)$$

lo que representa la suma de los momentos individuales.

Después de esto es posible calcular el centro de masas de la esfera utilizando el área y los momentos en *x* (M_x) y en *y* (M_y):

$$M_x = \sum_{i=1}^{n} m_i \, y_i \qquad (52)$$

$$M_y = \sum_{i=1}^{n} m_i \, x_i \qquad (53)$$

en donde m_i es un momento individual y x_i y y_i son las coordenadas *x* e *y* de dicho momento individual. De esta forma es posible calcular las coordenadas *x* e *y* del centro de masas de la esfera:

$$C_x = \frac{M_y}{a} \qquad C_y = \frac{M_x}{a} \qquad (54)$$

Una vez realizado esto, solo queda calcular el radio y dibujar el círculo correspondiente a estos datos:

$$radio = \sqrt{\frac{a}{\pi}} \qquad (55)$$

El problema de este método es la necesidad de una correcta umbralización de la esfera (Figura 40), de forma que las figuras dibujadas en su superficie no resulten un impedimento para la correcta extracción de los bordes de la esfera. Debido a que el método de momentos utiliza la información del área de la figura a medir, las variaciones en el contorno (zonas resaltadas en la Figura 40 mediante elipses rojas) producen variaciones en el área de la esfera, y por tanto, errores en el centro de la esfera estimado.

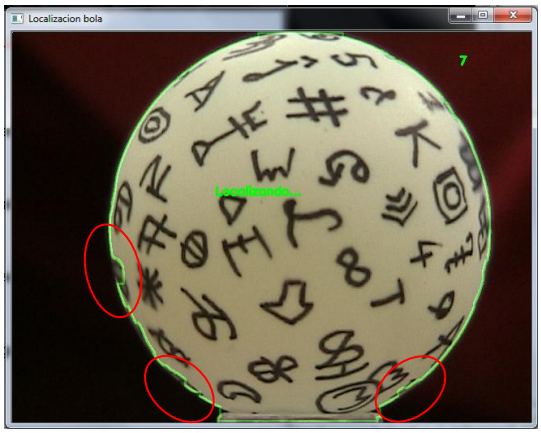


Figura 40. Umbralización de esfera para el método de momentos

4.2.3. Camshift

El algoritmo Continuosly Adaptive Mean Shift Algorithm o CamShift [1] es una adaptación del algoritmo Mean Shift que, dada la densidad de probabilidad de una imagen, encuentra la media de la distribución mediante la iteración en la dirección de incremento máximo de la densidad de probabilidad.

La principal diferencia entre Mean Shift y CamShift es que mientras éste último utiliza una distribución de probabilidad continuamente adaptativa, Mean Shift se basa en distribuciones

de probabilidad estáticas, que no se actualizan a menos que el objetivo sufra cambios significativos en la forma, el tamaño o el color.

Tal y como se ha mencionado, Camshift no necesita de la etapa de preprocesado que aparece en el apartado 4.145. El único proceso previo que necesita es la transformación del espacio de color RGB al espacio de color HSV, y el posterior aislamiento de la componente Hue (Figura 41), que servirá como base para los cálculos que se verán a continuación.

Camshift se basa en el análisis del histograma de una zona de la pantalla que el usuario le pasa como entrada. En el caso en el que nos encontramos, se selecciona una porción de la esfera a localizar para que analice su histograma. Este histograma cuantifica en bins o contenedores, que reducen la complejidad espacial y computacional permitiendo que los colores similares se almacenen juntos.

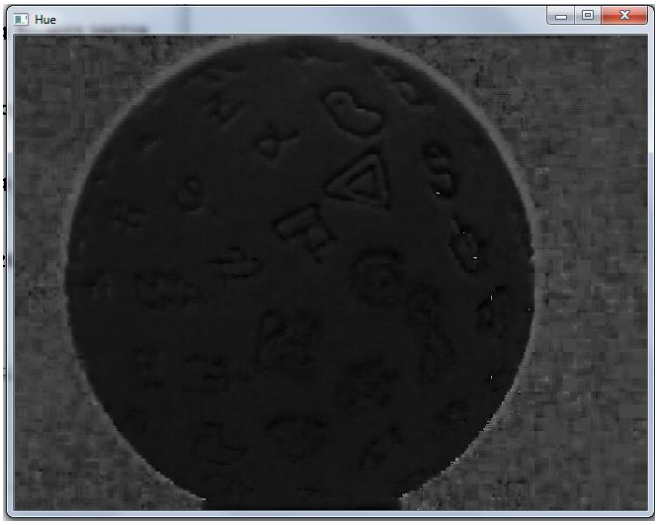


Figura 41. Componente H de la esfera

Una vez calculado el histograma de la esfera a localizar en el primer frame, el sistema se encarga de calcular la retroproyección del histograma objetivo en las nuevas imágenes. Dado el uso de histogramas con *m* bins, se definen *n* localizaciones de píxeles en una imagen x_i, un histograma $\hat{q}$ y la función $c: \Re^2 \rightarrow \{1 \dots m\}$ que asocia la localización del pixel x_i con el índice $c\,(x_i^*)$ del recipiente del histograma, de forma que el histograma calculado $\hat{q}_u$ tiene la siguiente forma:

$$\hat{q}_u = \sum_{i=1}^{n} \delta\,[c\,(x_i^*) - u] \qquad (56)$$

Este proceso se encarga de generar una imagen en escala de grises donde cada pixel tiene como intensidad la probabilidad de que dicho pixel en la imagen pertenezca al objeto que se busca, en nuestro caso, la esfera. Así, la región con píxeles mas blancos es la que más posibilidades tiene de ser la esfera, y por tanto, es la región sobre la que se realiza el siguiente paso.

Una vez calculado el histograma y cuantificados los valores de cada bin $\hat{p}_u$, el histograma se escala entre sus intensidades máximas y mínimas (57).

$$\left\{\hat{p}_u = min\left(\frac{255}{\max(\hat{q})}\hat{q}_u, 255\right)\right\}_{u=1\ldots m} \qquad (57)$$

El siguiente paso es calcular el centro de masas mediante el método de momentos (sección 4.2.2). Se realizan mediciones del centro de masas para la ventana de inclusión de la región calculada en la iteración anterior, hasta que o bien no existe cambio significativo entre ambas posiciones o bien se realizan 20 iteraciones.

Una de las virtudes del algoritmo CamShift es el poder que tiene para obtener la posición del objeto a localizar en distintas orientaciones mediante momentos de segundo orden, aunque no es un hecho significativo para el objetivo de este TFM.

4.3. Refinamiento de parámetros

Tal y como se verá en la sección 0, el método de momentos es el más preciso y rápido, pero su resultado no nos proporciona una localización exacta de la esfera. Para realizar una correcta localización, es necesario considerar la existencia de dos inconvenientes: la distorsión de la perspectiva y la distorsión sufrida por el centro de la esfera, problemas que hacen necesaria una fase de refinamiento o corrección de los parámetros calculados en la localización de la esfera. Estos inconvenientes tienen una naturaleza geométrica y tienen que ver con la posición relativa entre la esfera y la cámara.

Según la RAE, la palabra "perspectiva" hace referencia a un conjunto de objetos que desde un punto determinado se presentan a la vista del espectador de una forma determinada. Esta definición remarca la importancia de la posición y dirección del punto de vista, es nuestro caso, posición y dirección a donde apunta la cámara que usamos. Mientras que la posición de la cámara origina la distorsión de la perspectiva, la dirección de observación de la cámara origina la distorsión sufrida por el centro de la esfera.

En esta sección se describen estos dos problemas y las soluciones que se desarrollan para enmendarlos.

4.3.1. Distorsión de la perspectiva

La distorsión de la perspectiva es la transformación que sufre un objeto y su entorno circundante debido a la proximidad del mismo respecto a la cámara. De esta forma, cuanto más cerca de la cámara se encuentra el objeto en cuestión, en nuestro caso la esfera, más distorsionado aparece en la imagen.

El efecto que tiene este fenómeno en nuestro caso, es el estiramiento de la esfera, cambiando su contorno, dándole una apariencia elíptica en lugar de la apariencia circular que debería tener. Por este motivo, es necesario refinar los parámetros obtenidos en la localización, algo que realizamos mediante el ajuste de una elipse al contorno obtenido en la

etapa de preprocesado tomando los datos de localización obtenidos mediante el método de Momentos como referencia.

Este ajuste se basa en la búsqueda de la elipse (coordenada *x* del centro *Cx*, coordenada *y* del centro *Cy*, radio horizontal *Rh* y radio vertical *Rv*) que minimice la operación lógica XOr entre el contorno extraído de la etapa de preprocesado y el contorno de dicha elipse (figuras Figura 42-Figura 44).

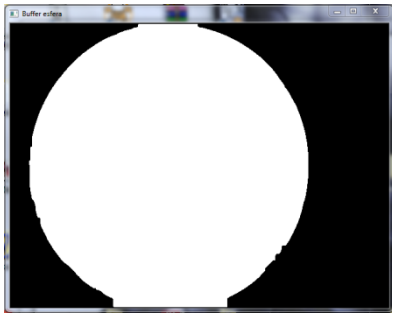


Figura 42. Buffer de esferas

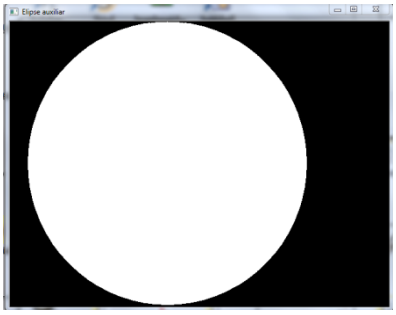


Figura 43. Elipse auxiliar

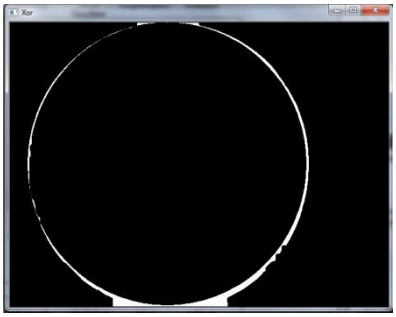


Figura 44. XOr resultante

Para acotar la búsqueda se utilizan los parámetros obtenidos en la localización, aunque hay que diferenciar dos casos:

1. **Localización primera**

   Primera localización a realizar, que tras realizar la etapa de preprocesado y la de localización del centro mediante Momentos, ajusta la elipse al contorno del buffer de esferas basándose en otorgar un cierto margen de error a los datos obtenidos mediante Momentos. Este margen posibilita un número total de $7 \times 7 \times 11 \times 11 = 5929$ elipses a analizar mediante la operación lógica XOr a partir de las coordenadas obtenidas mediante Momentos. El número de iteraciones corresponde con el margen de búsqueda que se le da al ajuste de la elipse: 7 x 7 se refiere a un cuadrado en el que el pixel obtenido en la localización por el método de momentos se sitúa en el centro, permitiendo refinar este centro teniendo en cuenta el resto de píxeles del cuadrado. En cuanto a los radios de la elipse, se utilizan 11 iteraciones en *x* y 11 iteraciones en *y*, de forma que la búsqueda se realice con una holgura de +5 y -5 pixeles respecto del radio calculado por el método de momentos.

2. **Localizaciones posteriores**

   Tras realizar la etapa de preprocesado, se otorga un margen de error menor al anterior, tomando como base los datos de la elipse calculada en el paso 1. El margen de error es menor en este caso debido a que se presupone que los desplazamientos en la esfera son muy pequeños y ya que la posición de ésta se ha calculado en iteraciones anteriores, es conveniente no perder el tiempo al intentar localizar la esfera en una gran porción de la imagen. De esta forma, el margen en estos casos posibilita un total de $7 \times 7 \times 5 \times 5 = 1225$ elipses a analizar mediante la operación lógica XOr a partir de la elipse calculada en la localización anterior.

Esta operación es sensible al ruido producido por los símbolos dibujados en la superficie de la esfera, ya que para una XOr perfecta, el contorno de la esfera debería de estar segmentado perfectamente, algo complicado debido a la presencia de estos símbolos en las zonas próximas al contorno de la esfera.

El resultado final de la localización, visto por pantalla, da lugar a una imagen como la observada en la Figura 45.

Figura 45. Resultado de la localización

4.3.2. Distorsión sufrida por el centro de la esfera

Una vez tenemos localizada la esfera gráficamente, es necesario realizar una transformación del centro de la esfera calculado. Esto se debe a la posición que ocupa ésta respecto del centro de coordenadas de la cámara, que se encuentra en la coordenada (0,0,0).

Si el centro de la esfera *(CX, CY, CZ)* no se encuentra centrado respecto a la cámara, el plano de la imagen no capta el centro correcto de la esfera (*Cx', Cy', Cz'*), si no un centro desplazado (*Cx, Cy*, *f*, siendo *f* la distancia focal extraída de la matriz intrínseca de la cámara).

Para corregir este error, utilizamos 2 puntos que pasan por los laterales de la esfera localizada gráficamente, es decir, forman parte de los extremos del diámetro de la esfera: (*LXd, LYd, LZd*) y *(LXi, LYi, LZi*).

Estos dos puntos tienen sus proyecciones sobre el plano de la imagen en (*Lxi, Lyi, f*) y (*Lxi, Lyi, f*) respectivamente. Pese a lo que pueda parecer, la colocación de estos puntos no es la correcta, ya que no se encuentran a la misma distancia *d* de la cámara, por lo que es necesario transformarlos para que se cumpla la igualdad de distancia entre ambos puntos y la cámara. Gracias a estos dos puntos, es posible reconstruir el centro correcto.

El siguiente paso es, pues, el cálculo del punto (*Lxi, Lyi, Lzi*) igualando el módulo de los vectores $\overrightarrow{V_1}$ y $\overrightarrow{V_2}$, formados por $\overrightarrow{(Lxı, Lyı, f)(0,0,0)}$ y $\overrightarrow{(Lxd, Lyd, f)(0,0,0)}$.

Una vez disponemos de los puntos (*Lxi, Lyi, Lzi*) y *(Lxd, Lyd, f)*, se calcula *(Cx', Cy', Cz')* como el punto central entre ambos puntos, con lo que el centro de la esfera quedaría calculada correctamente (Figura 46).

Cabe destacar que, al igual que el centro de la esfera calculado en el plano de la imagen necesita de una corrección, los puntos invariantes que se calculan en la sección 5.2 también necesitan de una corrección.

Esta corrección se conoce como retroproyección, y se basa en que cualquier punto de coordenadas (Px,Py,f) situado en el plano de la imagen, tiene su retroproyección en el punto de coordenadas (Px',Py',Pz').

Este punto corregido se consigue mediante la intersección entre el circulo con centro en (Cx',Cy',Cz') y el vector formado por los puntos (Px,Py,f) y el centro de coordenadas o posición de la cámara (0,0,0).

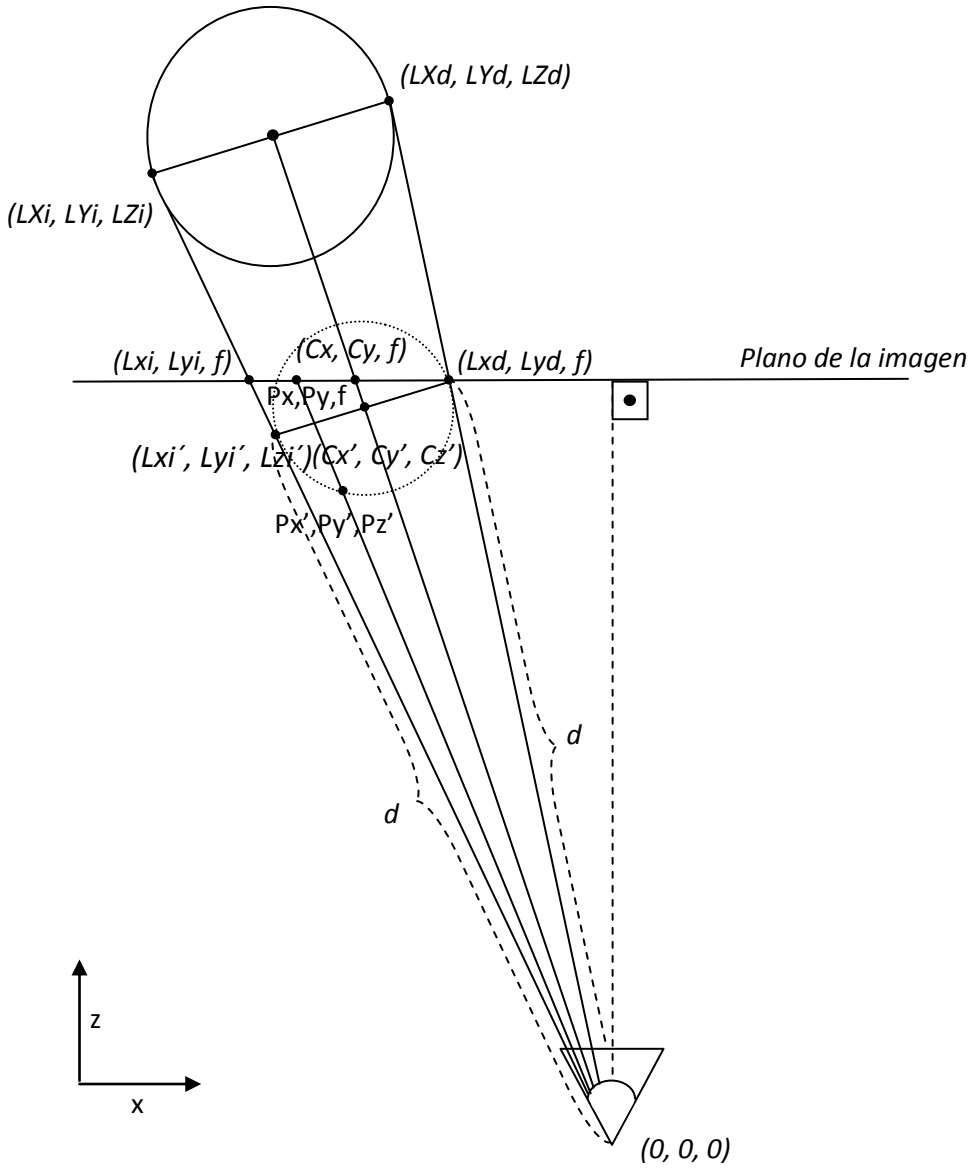


Figura 46. Distorsión sufrida por el centro de la esfera

De esta forma, se consiguen las coordenadas 3D de cada uno de los puntos que forman parte de las correspondencias calculadas entre los puntos invariantes de dos imágenes, algo necesario para el cálculo de la rotación efectuada por la esfera entre estas dos imágenes. Una representación gráfica de este problema se da en la Figura 46.

4.4. Evaluación de los métodos de localización

En esta sección, se presentan tanto la forma de calcular los datos de referencia de la esfera que consideraremos como válidos para los análisis de rendimiento, como las medidas que determinan el rendimiento de cada uno de los tres métodos de localización

4.4.1. Obtención de medidas reales

El proceso utilizado para obtener las mediciones referencia de localización 2D de la esfera (parámetros *Cx*, *Cy* y *radio*, correspondientes a la coordenada del centro en *x*, la coordenada del centro en *y* y al radio de la esfera respectivamente), es un procedimiento totalmente manual realizado sobre un programa de diseño asistido por ordenador. El procedimiento es el siguiente:

1. **Colocación de la cámara**

   Se fija tanto la posición de la cámara como la posición de la esfera respecto a la misma, de forma que estas posiciones sean fijas.

2. **Obtención de la imagen**

   Se realiza una fotografía con la cámara (Figura 47).

3. **Dibujado de las mediciones**

   Primero se dibuja un cuadrado verde que haga las veces de caja de inclusión (Figura 48), seguidamente se unen las esquinas de la caja de inclusión para obtener el centro del cuadrado mediante la intersección de estas dos líneas (Figura 49), y finalmente se dibuja un rectángulo desde el pixel de la esquina superior izquierda de la imagen hasta el punto central del cuadrado , de forma que la posición del centro de la esfera viene determinada por la longitud de los lados de este rectángulo (Figura 50).

4. **Obtención de las medidas**

   Mientras el radio se calcula dividiendo el lado de la caja de inclusión por dos, *Cx* y *Cy* se extraen de la longitud de los lados del rectángulo que une la esquina superior izquierda de la imagen con el punto central de la caja de inclusión.

Figura 47. Imagen de entrada

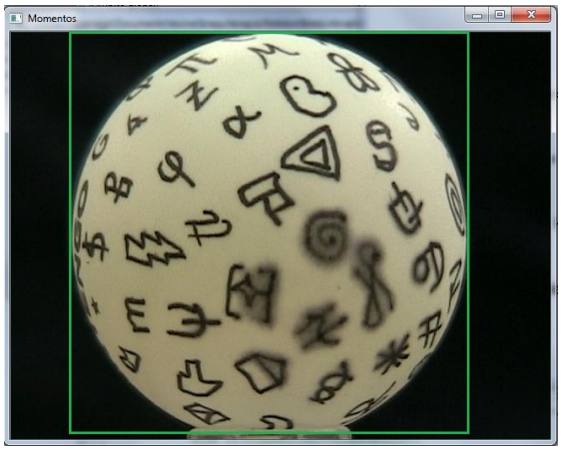


Figura 48. Caja de inclusión de la esfera

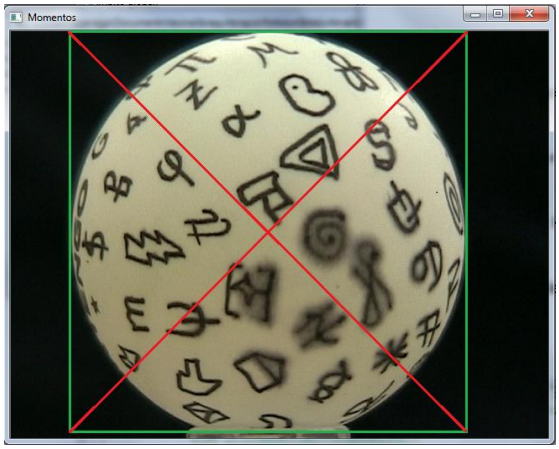


Figura 49. Calculo del punto central

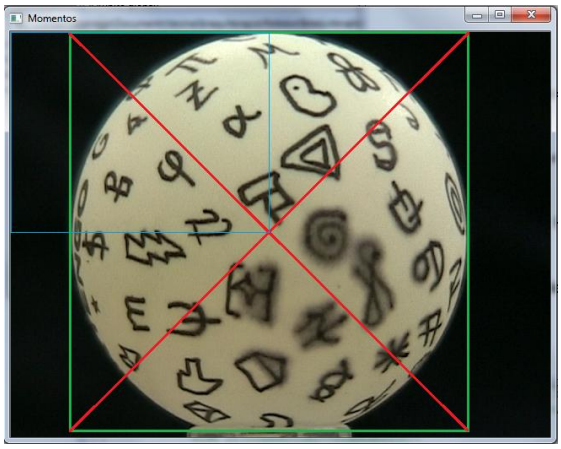


Figura 50. Posición del punto central

4.4.2. Error relativo

Para conocer la exactitud de la localización de la esfera, se ha calculado una medida de error ER sobre *Cx*, *Cy* y *radio*. Dicha medida viene en función del radio referencia de la esfera, sin unidades, representando el valor el porcentaje de distancia entre valores respecto al radio de la esfera.

Así, obtenemos las siguientes formulas:

$$ER\ Cx = \frac{\sum_{durante\ n\ frames} abs(Cx' - Cx)}{n * radio} \qquad (58)$$

siendo Cx' la coordenada *x* del centro obtenida por el programa, *Cx* la medida real de la coordenada *x* del centro, *radio* el radio real de la esfera y *n* el número de frames durante los cuales se realiza la medición.

$$ER\ Cy = \frac{\sum_{durante\ n\ frames} abs(Cy' - Cy)}{n * radio} \qquad (59)$$

en donde Cy' es la coordenada *y* del centro obtenida por el programa, *Cy* es la medida real de la coordenada *y* del centro, *radio* es el radio real de la esfera y *n* es el número de frames durante los cuales se realiza la medición.

$$ER\ radio = \frac{\sum_{durante\ n\ frames} abs(radio' - radio)}{n * radio} \qquad (60)$$

siendo radio' el radio de la esfera medido por el programa, *radio* es el radio real de la esfera y *n* el número de frames durante los cuales se realiza la medición.

Cabe destacar que en *ER Cx* y *ER Cy* se dividen por el radio para tener un error que sea relativo al tamaño de la esfera independientemente de su posición. Si no fuera así, un error de la misma magnitud no tendría el mismo valor si el centro de la circunferencia se encuentra en distintas posiciones (Figura 51 y Figura 52).

Las mediciones de *ER*, en distintas posiciones, y con un error de 2 píxeles de diferencia entre Cx y Cx', el error seria:

Figura 51: $ER\ Cx = \frac{abs(152-150)}{25}$=0.08

Figura 52: $ER\ Cx = \frac{abs(502-500)}{25}$=0.08

Por tanto, el mismo error en zonas distintas de la imagen, daría el mismo resultado, y tendrá el valor de 0.08 veces el radio.

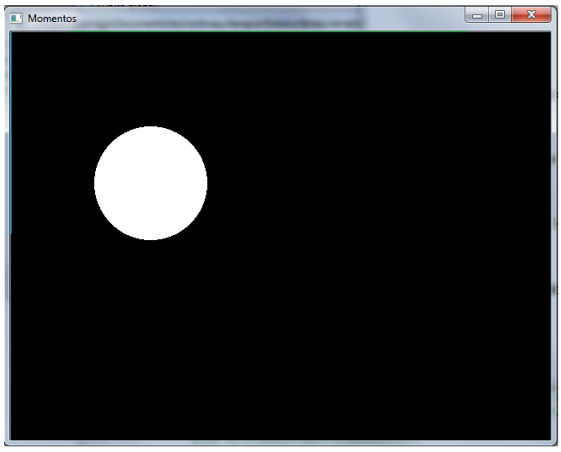


Figura 51. *Cx*=150, *Cy*=100, *radio*=25

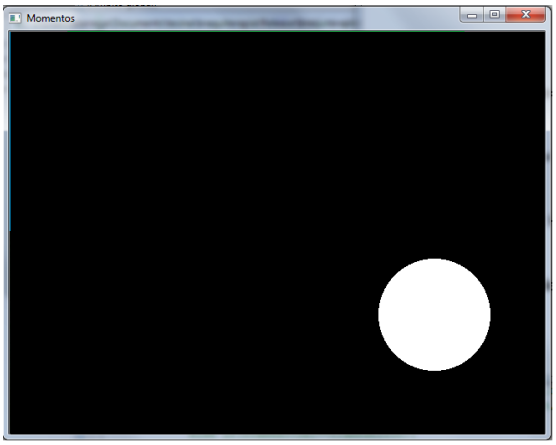


Figura 52. *Cx*=500, *Cy*=350, *radio*=25

4.4.3. FPS

Para calcular los FPS, se divide 100 entre el tiempo que tardan en mostrarse 100 frames:

$$FPS = \frac{100}{t\ en\ mostrar\ 100\ frames} \qquad (61)$$

4.4.4. Rendimiento

En esta sección se va a analizar tanto la velocidad como el error que se produce en cada uno de los tres métodos. Para esto, se calculan ambas medidas en 5 series de 100 análisis para cada uno de los métodos, para finalmente mostrar las medias de estas mediciones.

En cuanto al equipo utilizado, las pruebas se han realizado en un PC DELL con procesador Intel® Core™ i7 860 a 2.80 GHz, con 4 Gb de memoria RAM, tarjeta gráfica NVIDIA GeForce 310 de 512 Mb y SO Windows 7.

Cada una de las series de mediciones ha sido realizada con la esfera situada en diferentes posiciones respecto a la posición de la cámara, las cuales se detallan en la Figura 53.

	Cx	***Cy***	***radio***
Caso 1	307	238	236.5
Caso 2	264	238	235
Caso 3	374	237	235
Caso 4	268	238	238.5
Caso 5	296	237	238

Figura 53. Casos de prueba

Una vez presentado el marco en el que se desarrollan las pruebas y tras realizar las mediciones oportunas, obtenemos las graficas de velocidad (Figura 54) y las gráficas de error relativo (Figura 55-Figura 57).

MÉTODO	**Caso 1**	**Caso 2**	**Caso 3**	**Caso 4**	**Caso 5**	**Media**
MOMENTOS	22,22	25,00	27,04	25,00	25,13	24,87
HOUGH	17,24	18,86	17,90	17,12	19,03	18,03
CAMSHIFT	9,00	10,14	11,38	9,87	10,95	10,26

Figura 54. Mediciones FPS

Como se aprecia en la Figura 54, el método más rápido es el de momentos con casi 25 FPS de media, siguiéndolo con 18 FPS el método de la transformada de Hough, y finalmente Camshift, con casi 8 FPS.

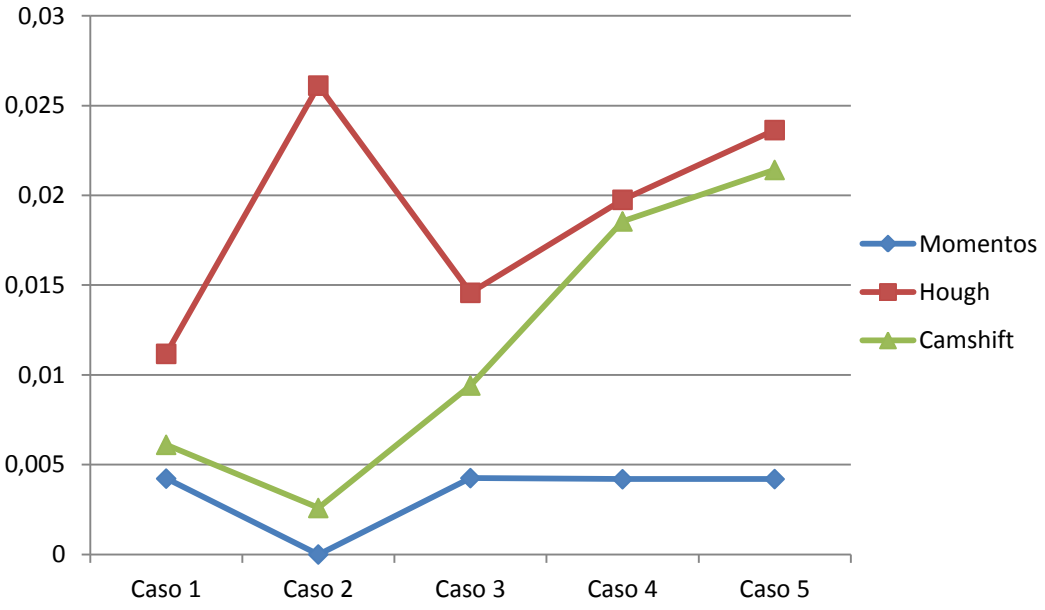


Figura 55. Error de *Cx*

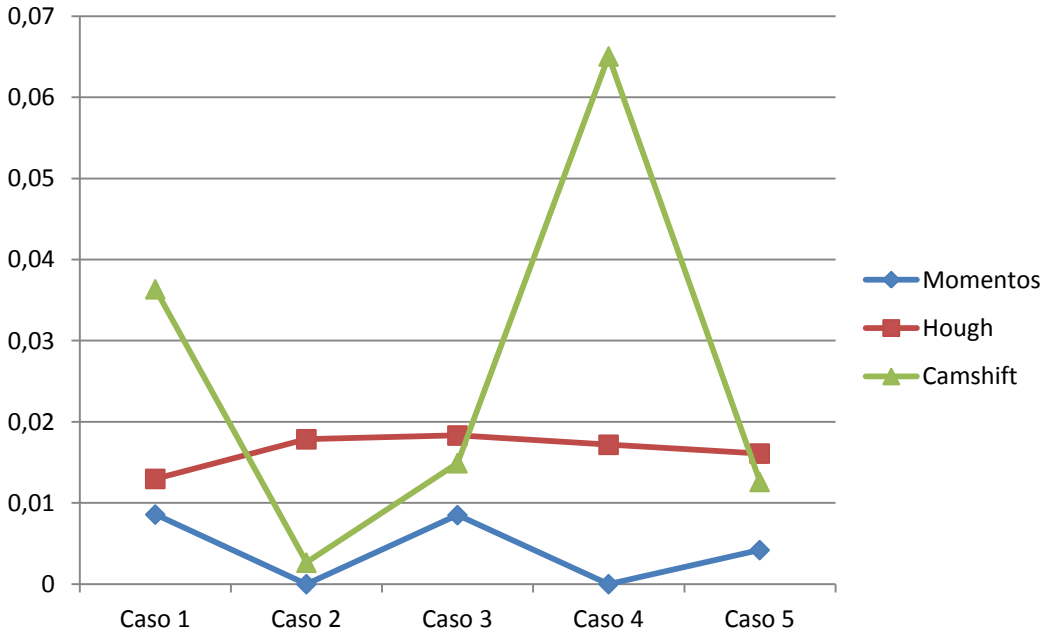


Figura 56. Error de *Cy*

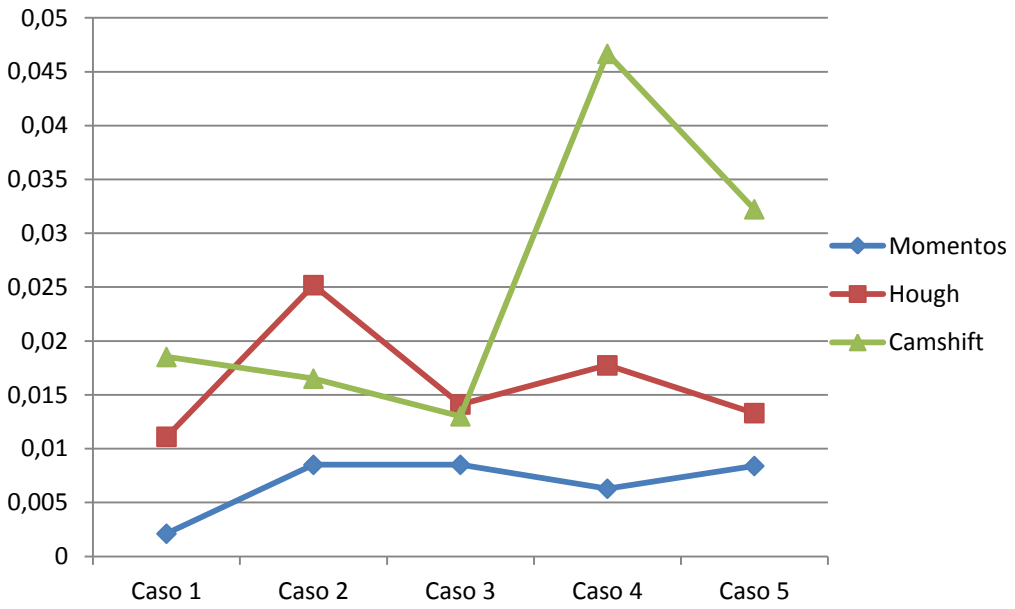


Figura 57. Error de *radio*

El método que más precisión tiene en todos y cada uno de los casos medidos es el de momentos, siendo además el más estable y robusto, tal y como se puede ver en la Figura 58,

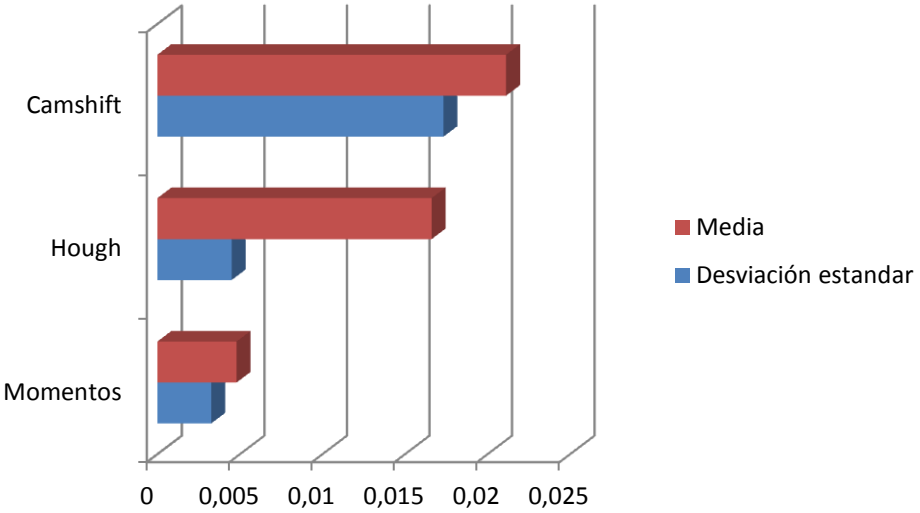


Figura 58. Medias y desviaciones típicas del error en la localización.

Los otros dos métodos sufren una gran variación en cuanto a la precisión que aportan. Observando el funcionamiento visual de Hough, se puede decir que es el método más variable de todos, ya que en frames sucesivos el círculo localizado puede variar bastante, tanto en posición como en escala. En la figuras Figura 59-Figura 61 se puede apreciar este comportamiento.

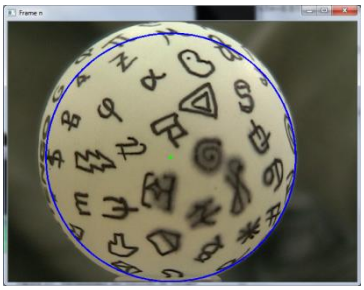


Figura 59. Frame n

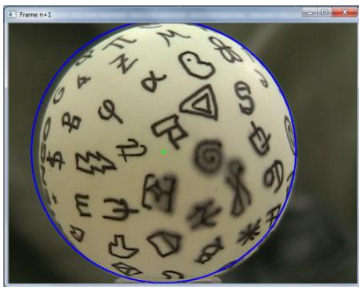


Figura 60. Frame n+1

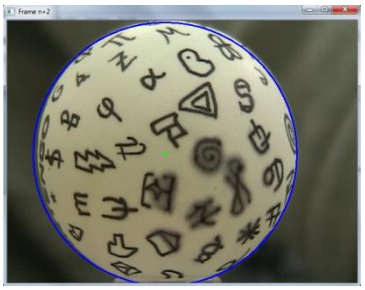


Figura 61. Frame n+2

Por otra parte, el comportamiento poco estable del método Camshift proviene del área de selección de la esfera que el usuario pasa al algoritmo para calcular el histograma. Las variaciones tanto de posición como de tamaño de dicha área posibilitan distintos histogramas y, por tanto, distintas retroproyecciones que producen distintas localizaciones y tamaños de la esfera objetivo:

SELECCIÓN 1

Figura 62. Área de selección 1

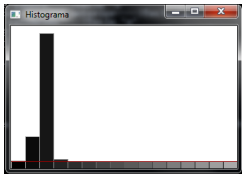


Figura 63. Histograma 1

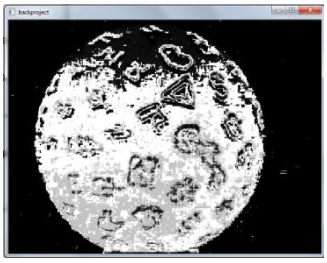


Figura 64. Retroproyección 1

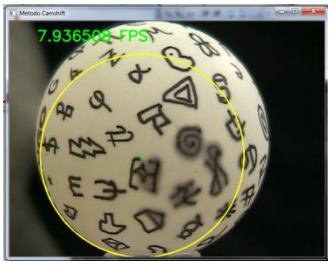


Figura 65. Circulo detectado 1

SELECCIÓN 2

Figura 66. Área de selección 2

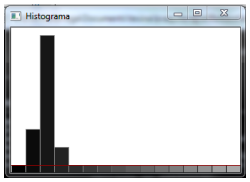


Figura 67. Histograma 2

Figura 68. Retroproyección 2

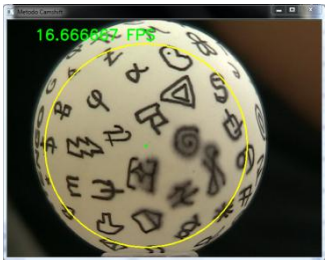


Figura 69. Círculo detectado 2

SELECCIÓN 3

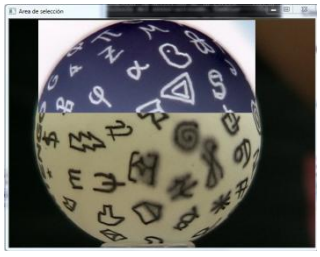


Figura 70. Área de selección 3

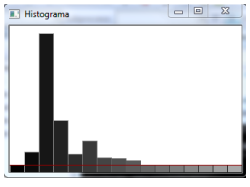


Figura 71. Histograma 3

Figura 72. Retroproyección 3

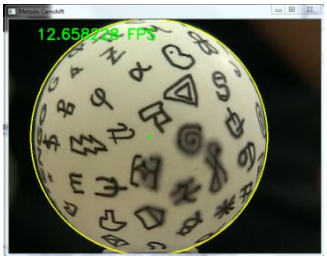


Figura 73. Círculo detectado 3

4.4.5. Elección del método de localización

Las pruebas realizadas muestran que tanto Hough como Camshift aportan no solo una tasa mayor de error que el método de Momentos (figuras Figura 55-Figura 57) para los tres parámetros medidos, sino también una mayor desviación típica (Figura 58). El método de momentos nunca alcanza la cota de error del 1% del radio de la esfera, llegando incluso en los mejores casos a alcanzar errores del 0%, mientras que el error mínimo que produce Hough sobrepasa el 1% del radio y Camshift únicamente rebaja el 1% del error en el caso 2 de la Figura 56, llegando incluso en ocasiones a errores del 6,5% del radio en el caso 4 de la Figura 56.

Dado que el error máximo del método de momentos es inferior al 1% del radio, y que el radio de esta es de 20 mm, obtenemos un error máximo de 0,2 *mm* en la medición del radio y del centro de la esfera.

5. MOVIMIENTO DE LA ESFERA

En esta sección se presenta un método para detectar el movimiento de la esfera, de forma que cuando lo detecta, se procede a determinar qué giro ha sufrido.

Para calcular dicho giro, es preciso calcular los puntos invariantes en las imágenes en que el giro empieza y acaba respectivamente, para posteriormente calcular una serie de correspondencias entre ellas, que sirven como base para calcular la rotación mediante una descomposición en valores singulares.

En cuanto a los métodos de extracción de puntos invariantes, se utilizan los descriptores SIFT y SURF, que veremos con más detalle a continuación.

5.1. Detección de movimiento.

El sistema propuesto trabaja de forma que no es necesario calcular siempre los puntos invariantes entre 2 imágenes consecutivas para calcular la rotación sufrida por la esfera, sino que tiene un contador que se activa cuando se detecta movimiento en la esfera. Este contador permite que el usuario siga realizando el movimiento en la esfera que desee, hasta que, tras detectar el fin de la rotación, tras unos instantes en reposo, se procede a calcular los puntos invariantes y la rotación que realiza la esfera.

Para calcular el movimiento de la esfera, se utiliza la estimación del flujo óptico, que se define como el patrón de movimiento aparente de un patrón de intensidad causado por el movimiento relativo entre el observador y la escena.

El algoritmo utilizado para estimar el flujo óptico se basa en el método de Lucas-Kanade [29], que asume que el flujo es esencialmente constante en el área colindante al pixel sobre el que se está calculando. Este método resuelve las ecuaciones básicas del flujo óptico para todos los píxeles en dicha área mediante mínimos cuadrados:

$$\begin{aligned} I_x(q_1)V_x + I_y(q_1)V_y &= -I_t(q_1) \\ I_x(q_2)V_x + I_y(q_2)V_y &= -I_t(q_2) \\ &\vdots \\ I_x(q_n)V_x + I_y(q_n)V_y &= -I_t(q_n) \end{aligned} \qquad (62)$$

dónde q_i son los píxeles del área sobre la que se trabaja, $I_x(q_i)$ es la derivada parcial de la imagen I respecto a la posición *x*,*y*, evaluada en el pixel q_i en el tiempo actual , y (V_x, V_y) es el vector de velocidades de flujo.

Además se incluye una matriz de pesos *W* para que los píxeles más cercanos al central tengan un mayor peso. Los pesos w_i se asignan mediante una función gaussiana de la distancia entre q_i y el pixel central *p*. En el caso de este TFM, el pixel *p* que se le pasa a este método como entrada es el píxel más cercano al centro de la esfera que forma parte de uno de los contornos dibujados en la esfera. Estos contornos se detectan mediante umbralizaciones.

Así, se forma el siguiente sistema de ecuaciones

$$v = (A^T W A)^{-1} A^T W b \qquad (63)$$

que también se puede leer como

$$\begin{bmatrix} V_x \\ V_y \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{n} w_i I_x(q_i)^2 & \sum_{i=1}^{n} w_i I_x(q_i) I_y(q_i) \\ \sum_{i=1}^{n} w_i I_x(q_i) I_y(q_i) & \sum_{i=1}^{n} w_i I_y(q_i)^2 \end{bmatrix}^{-1} \begin{bmatrix} -\sum_{i=1}^{n} w_i I_x(q_i) I_t(q_i) \\ -\sum_{i=1}^{n} w_i I_y(q_i) I_t(q_i) \end{bmatrix} \qquad (64)$$

Finalmente, tras extraer el vector de velocidades, si sobrepasa un cierto umbral, se da por hecho que la esfera ha girado, momento en el cual se procede a calcular la nueva posición de la esfera mediante la localización y, posteriormente, se calcula el giro.

Partiendo del método de Lucas-Kanade, Jean-Yves Bouguet [4] desarrolla una extensión piramidal, en la cual se calcula el flujo óptico sobre una representación piramidal de la imagen sobre la que trabaja. Esta pirámide (Figura 74) se corresponde a una serie de imágenes resultado de una reducción de escala de la imagen inicial en factores múltiplos de dos.

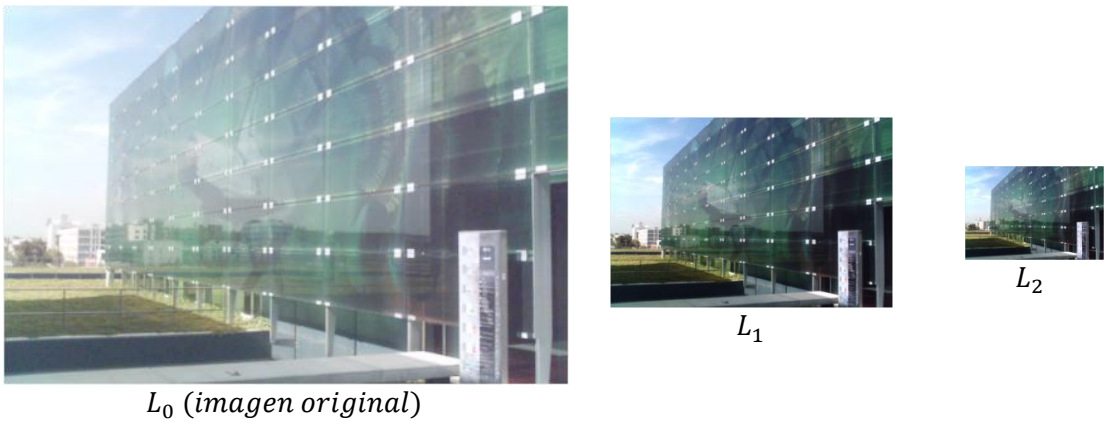


Figura 74. Representación piramidal de Lucas-Kanade

El uso de los diferentes tamaños tiene como explicación, por una parte, la necesidad del uso de imágenes pequeñas para la detección de movimientos bruscos (debido a que la ventana de la imagen que se analiza es mayor sin incrementar mucho el tiempo necesario para su análisis) y, por otra parte, la necesidad del uso de imágenes de gran tamaño para realizar los cálculos con mayor precisión.

El trabajo piramidal se basa en calcular el vector de flujo óptico v entre dos imágenes para la región de interés dada en un nivel, de forma que seguidamente se calcula para un nivel superior teniendo en cuenta regiones que no aparecen en nivel inferior. Así, el área de búsqueda en el nivel final de la pirámide constituye la imagen original completa.

Una vez obtenido el vector de flujo óptico, si el movimiento realizado es mayor a un determinado umbral, se activa el contador que permite que continúe el movimiento hasta que deje de producirse, momento en el cual pasan a calcularse las correspondencias de puntos invariantes.

5.2. Extracción de puntos invariantes

Las técnicas de extracción de puntos invariantes permiten extraer puntos de información relevante de una imagen mediante descriptores locales.

Un descriptor local busca extraer información de un área localizada de la imagen, como puede ser la textura alrededor de un punto de interés, y define un método de codificación que sea invariante a cambios de iluminación y transformaciones afines.

Mediante este tipo de métodos es posible detectar marcas en una superficie invariantes ante cambios de posición y orientación de la ésta, dada la posición y orientación de una cámara. Una vez calculados estos, los mismos algoritmos se encargan de buscar correspondencias entre imágenes, para lo que es necesario fijar un cierto umbral, que estas correspondencias no deben de sobrepasar para considerarse como un acierto en el matching.

Dentro de este tipo de técnicas se encuentran los descriptores SIFT y SURF, que son los que se van a usar en este método debido a que se adecuan perfectamente a lo que se busca (detectores de puntos invariantes) y a que son métodos ampliamente utilizados en infinidad de aplicaciones que buscan correspondencias entre imágenes.

5.2.1. Scale Invariant Feature Transforms (SIFT)

El algoritmo SIFT propuesto por Lowe [28] es uno de los más utilizados hasta la fecha en la extracción de puntos invariantes para imágenes en entornos reales. Estos puntos son invariantes ante cambios de rotación y de escala, además de que son parcialmente inmunes ante cambios de iluminación. El método SIFT realiza los siguientes pasos:

1. **Detección de máximos en el espacio de escala.**

   El primer paso es obtener un espacio piramidal de Diferencias de Gaussianas (DoG) mediante un banco de filtros, donde se evalúan aquellas localizaciones en escala y posición susceptibles de ser un punto de interés. Estos puntos coinciden con los máximos y mínimos de ese espacio.

2. **Localización de puntos de interés.**

   En cada punto candidato de ser un punto de interés se realiza un ajuste de modelo que permite encontrar con precisión la escala y la posición del punto.

3. **Obtención de la orientación local.**

   Una o varias orientaciones principales son asignadas a cada punto clave obtenido en el proceso anterior. Dicha orientación se obtiene a través de un histograma de direcciones de gradientes en el espacio de la imagen.

Una vez asignadas, todas las operaciones posteriores se referencian a la escala y orientación calculadas. La invariancia a la rotación y escala se obtiene en este punto.

4. **Descriptor.**

   Los gradientes locales de la imagen, referentes a la escala y la rotación calculada, se utilizan para construir un descriptor que posee inmunidad ante distorsiones y cambios de iluminación.

El resultado final es un descriptor que generalmente se compone de 128 componentes que se compara en sucesivas imágenes mediante distancia euclidea.

La Figura 75 es un ejemplo de los descriptores localizados en una imagen (hasta 3062 descriptores).

Figura 75. Ejemplo de imagen con descriptores SIFT calculados.

El método SIFT tiene un coste computacional bastante alto y el descriptor que propone tiene un tamaño bastante grande. También tiene problemas con algunos descriptores que se encuentran en el borde de los objetos.

Aun así este método está ampliamente aceptado y ha sido fuente de muchas mejoras propuestas por otros investigadores, que han conseguido mejorar algunas de las deficiencias que éste posee. Un ejemplo de modificación de SIFT es el método SURF, que se describe a continuación.

5.2.2 Speed Up Robust Features (SURF)

La versión estándar del método SURF [3] es una modificación de SIFT que según su autor es mucho más rápido y es robusto ante diferentes transformaciones que el método SIFT.

Tanto SIFT como SURF obtienen puntos de interés de la imagen invariantes a escala y orientación, así como a cambios en iluminación. Ambos obtienen un vector descriptor por cada punto de interés, aunque calculados de distinta manera.

El algoritmo SURF utiliza una aproximación básica de la matriz Hessiana para reducir el tiempo de computación. La matriz Hessiana se utiliza debido a su buena relación entre la precisión y el coste temporal. Además, el determinante de la Hessiana se utiliza para la localización de los puntos y para la determinación de la escala.

Otra de las diferencias tiene lugar en los datos extraídos: mientras que SIFT guarda la posición, la escala y la orientación (puesto que es posible que en una misma posición (x, y) encontremos varios puntos de interés con distinta escala *s* y/u orientación σ), en SURF, en cambio, en una misma posición solamente aparece un único punto de interés, por lo que no guarda la escala y la orientación, aunque sí que registra la matriz de segundo momento y el signo de la laplaciana.

Esto se traduce en que, por defecto, el tamaño del descriptor SIFT es el doble de grande que el del descriptor SURF (128 componentes frente a 64), aunque es posible fijar el tamaño del descriptor SURF a 128 al igual que el SIFT, para darle mayor robustez, aunque de esta forma se pierda velocidad.

5.2.3. Comparación entre SIFT y SURF en bibliografía

Gracias trabajos de evaluación de ambos métodos realizados en [39], [30] o [45] es posible conocer resultados empíricos de la comparación entre SIFT y SURF.

En cuanto a las conclusiones extraíbles de la extracción de puntos, se afirma que SIFT obtiene 2.68 veces más puntos que SURF, mientras que SIFT es 3.39 veces más lento que SURF.

Gracias a estos datos es posible concluir que si el objetivo es realizar la detección de puntos lo más rápidamente posible, el algoritmo a utilizar debería ser SURF, mientras que si el objetivo es tener en cuenta un mayor número de datos para tener más precisión, se optaría por el uso de SIFT.

En cuanto a la búsqueda de correspondencias entre puntos localizados, se realizan diversos experimentos, variando escalas, rotaciones, calidad de imagen, condiciones de iluminación y transformaciones afines.

En general, se puede afirmar que SIFT funciona mejor que SURF en casos donde varían factores tales como la escala, la rotación y condiciones de emborronamiento, pero que SURF da mejores resultados en variaciones de iluminación y en tiempo, mientras que aportan resultados iguales ante transformaciones afines.

Por otra parte, pese a que SIFT es mucho más lento para obtener las correspondencias, encuentra más correspondencias que SURF, algo que puede ser debido a que SURF no permite que haya varios puntos invariantes en una misma posición con distinta escala y/u orientación, mientras que en SIFT sí que es posible, lo que hace pensar que la distancia entre el número de características detectada disminuiría si SIFT no “duplicase” puntos.

5.2.4. Pruebas comparativas entre SIFT y SURF

Con el objetivo de tener una idea propia sobre el funcionamiento de ambos métodos dentro del tipo de imágenes que se usan en este TFM, se han realizado una serie de pruebas sobre las implementaciones de estos dos algoritmos de las que se dispone en la versión 2.3 de OpenCV. Las pruebas consisten en la toma de 6 instantáneas de la esfera en posiciones continuas de rotación, y en el cálculo de los siguientes valores:

- Numero de descriptores localizados por cada método.
- Tiempo empleado en la localización de los descriptores.
- Tiempo necesario en el cálculo de las correspondencias.
- Fallos en el cálculo de correspondencias.

En cuanto al primer parámetro a medir, el descriptor SIFT produce una media de 966,6 descriptores (figuras Figura 76-Figura 81) mientras que SURF, con un tamaño de descriptor de 64, produce una media de 1049,3 descriptores (figuras Figura 82-Figura 87). La diferencia entre los puntos localizados es que, si bien SIFT localiza los puntos mayoritariamente en la silueta de los signos dibujados en la esfera, SURF los localiza no tanto sobre la silueta sino más bien en las inmediaciones de las mismas.

Figura 76. 945 descriptores SIFT en la imagen referencia

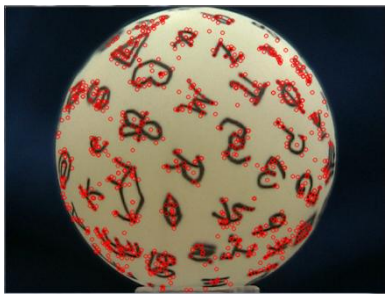


Figura 77. 1002 descriptores SIFT en la imagen 1

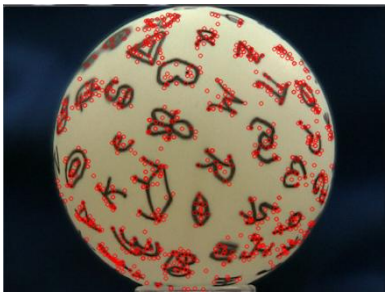


Figura 78. 946 descriptores SIFT en la imagen 2

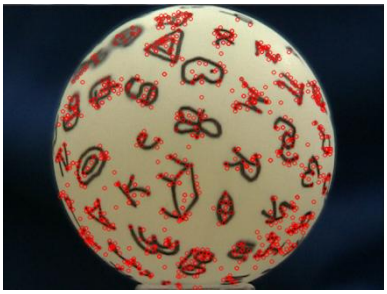


Figura 79. 958 descriptores SIFT en la imagen 3

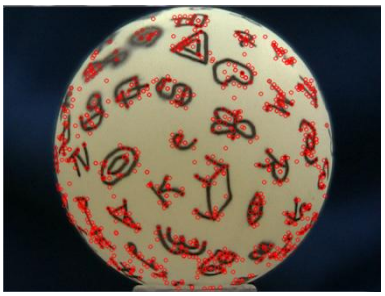


Figura 80. 961 descriptores SIFT en la imagen 4

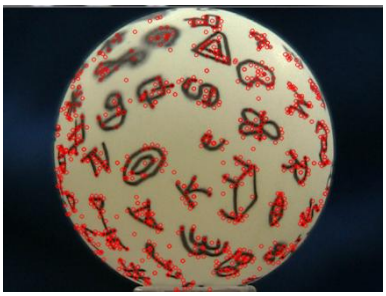


Figura 81. 988 descriptores SIFT en la imagen 5

Por otra parte, en lo referente a la velocidad a la que ambos algoritmos calculan estos descriptores, encontramos mucho más rápido a SURF que a SIFT, tardando 5 veces más (0,785 *s* frente a 4,162 *s* para calcular descriptores en 6 imágenes), con unas medias de 0,130 *s* para SURF y de 0,693 *s* para SIFT por cada detección de puntos en una imagen.

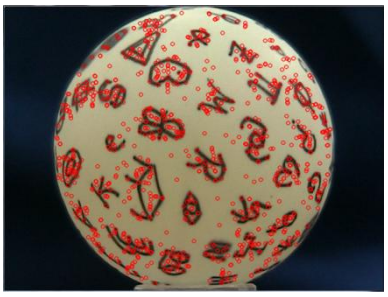


Figura 82. 1005 descriptores SURF en la imagen referencia

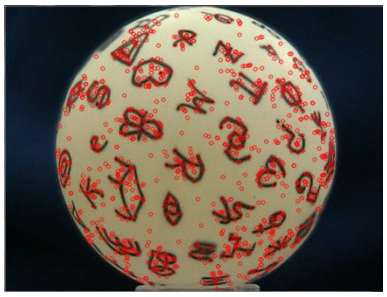


Figura 83. 1005 descriptores SURF en la imagen 1

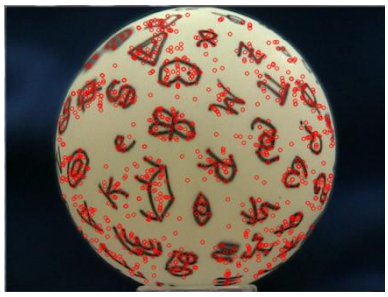


Figura 84. 997 descriptores SURF en la imagen 2

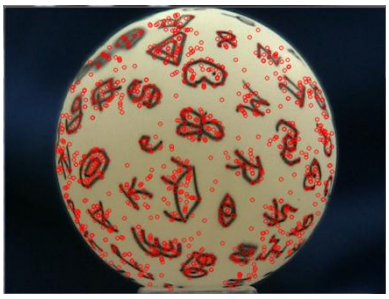


Figura 85. 1067 descriptores SURF en la imagen 3

Figura 86. 1126 descriptores SURF en la imagen 4

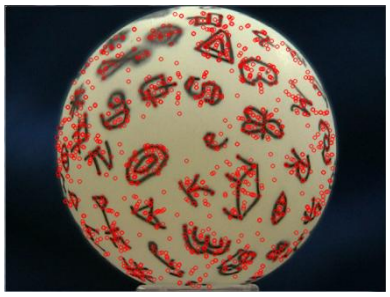


Figura 87. 1086 descriptores SURF en la imagen 5

En cuanto a la duración del establecimiento de correspondencias, cabe destacar que se utiliza un algoritmo de fuerza bruta basado en la distancia euclidea entre los puntos detectados entre dos imágenes.

Usando este algoritmo, encontramos que la elaboración de correspondencias es más rápida en SURF (0,237 *s)* que en SIFT (0,340 *s*) de media entre el conjunto de puntos invariantes propios de dos imágenes.

Figura 88. Correspondencias SIFT entre referencia e imagen 1. aciertos: 143, fallos: 11

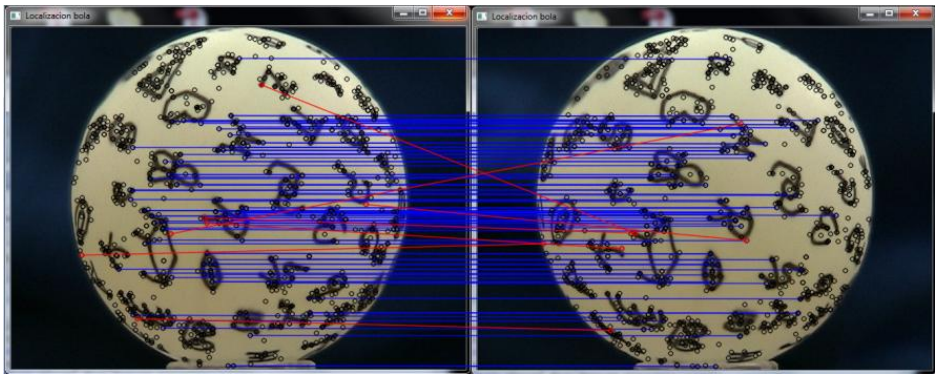


Figura 89. Correspondencias SIFT entre imagen 1 e imagen 2. aciertos: 95, fallos: 7

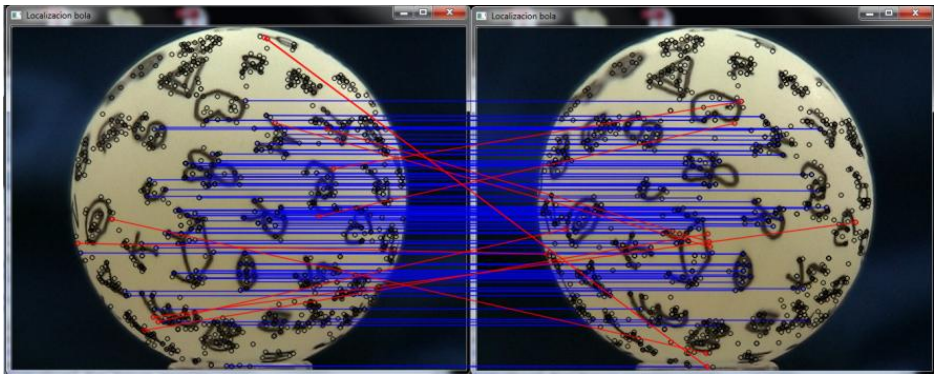


Figura 90. Correspondencias SIFT entre imagen 2 e imagen 3. aciertos: 105, fallos: 11

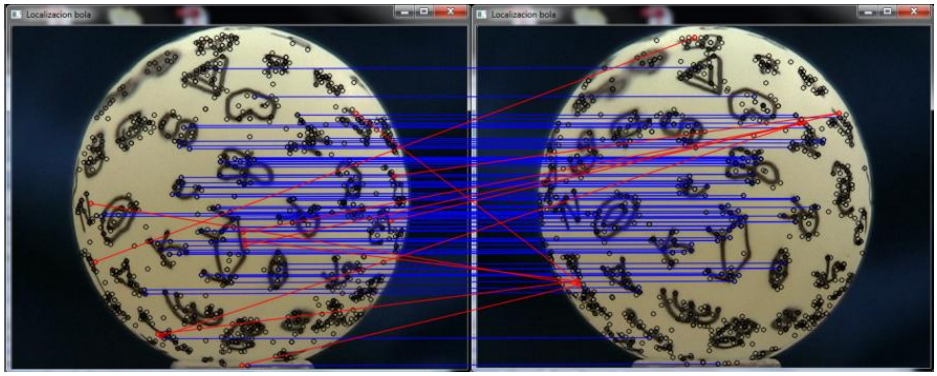


Figura 91. Correspondencias SIFT entre imagen 3 e imagen 4. aciertos: 83, fallos: 11

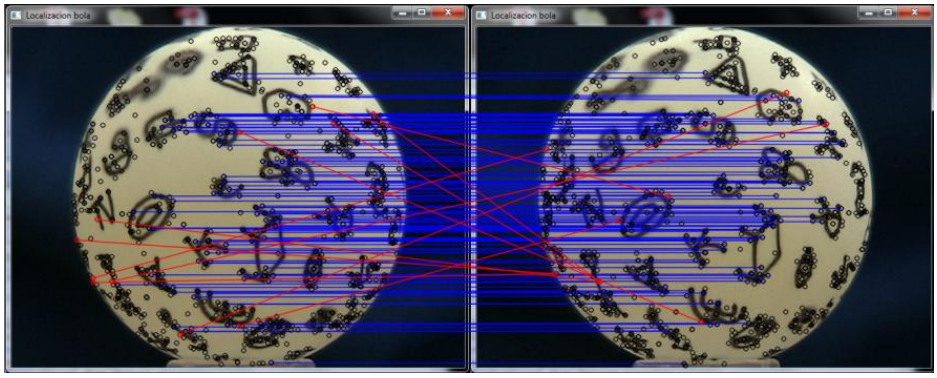


Figura 92. Correspondencias SIFT entre imagen 4 e imagen 5. aciertos: 118, falllos: 11

Para realizar las pruebas de tasa de acierto y de fallo se han escogido dos umbrales que intenten igualar el número de fallos, para así tener una mejor idea del ratio aciertos/fallos. Estos umbrales son 90 para SIFT y 0.13 para SURF.

Para tomar la decisión de determinar que correspondencia es un acierto y cual es un fallo, se calcula la pendiente entre todas las correspondencias, agrupando las que tengan una pendiente similar, de forma que toda aquella que sea lo suficiente distinta como para no pasar un umbral de distancia de 0.01 será considerada como falsa.

Tal y como se puede ver en las figuras Figura 88Figura 92 para SIFT, y en las figuras Figura 93-Figura 97 para SURF, este último detecta aciertos en mayor proporción: 20,69 aciertos por cada fallo frente a los 10,66 aciertos por cada fallo de SIFT.

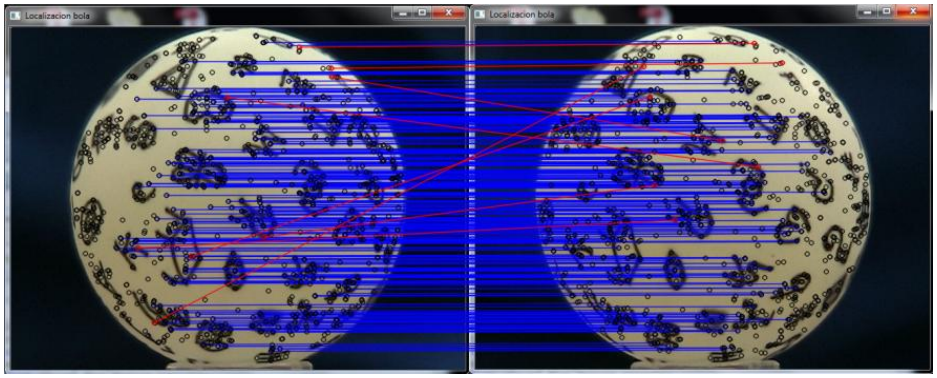


Figura 93. Correspondencias SURF entre referencia e imagen 1. aciertos: 280, fallos: 8

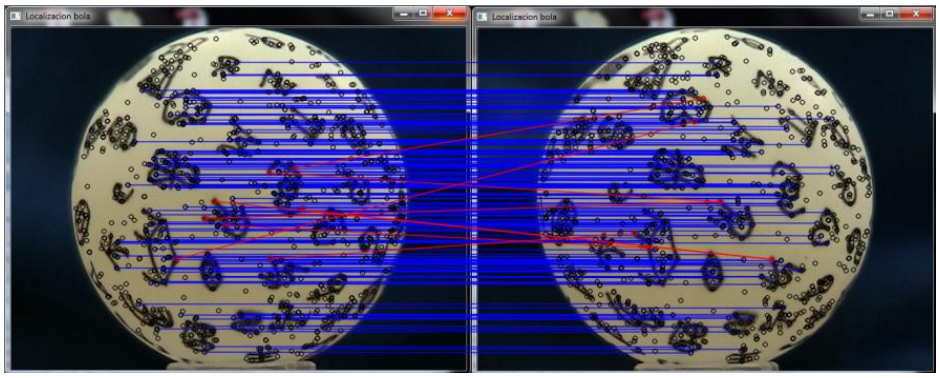


Figura 94. Correspondencias SURF entre imagen 1 e imagen 2. aciertos: 186, fallos: 8

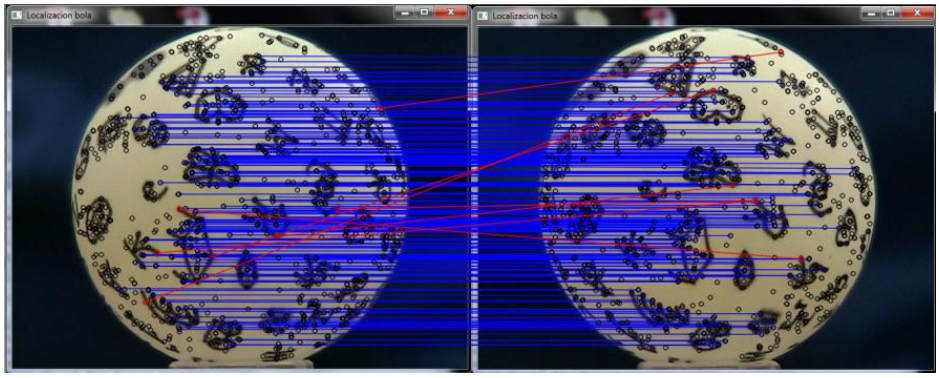


Figura 95. Correspondencias SURF entre imagen 2 e imagen 3. aciertos: 218, fallos: 7

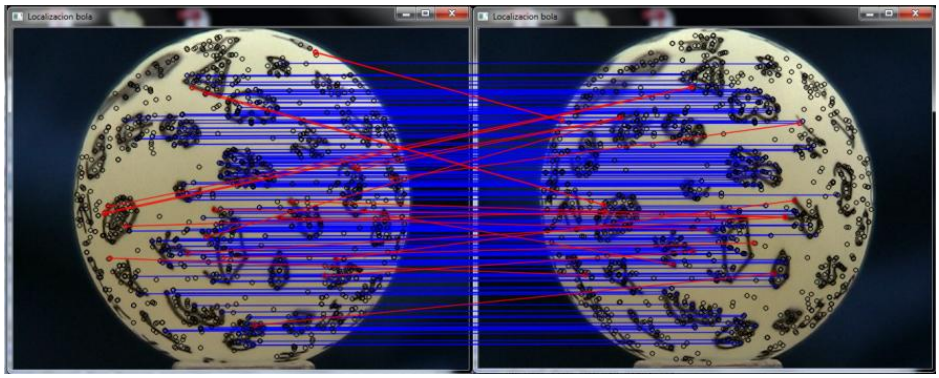


Figura 96. Correspondencias SURF entre imagen 3 e imagen 4. aciertos: 198, fallos: 22

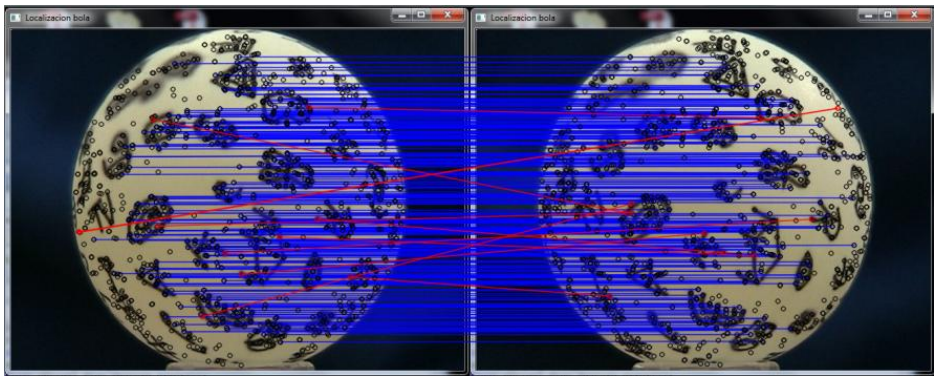


Figura 97. Correspondencias SURF entre imagen 4 e imagen 5. aciertos: 277, fallos: 11

Como resultado de las pruebas realizadas, SURF resulta claro vencedor, ya que sus números medios ganan en todas las facetas a los números medios resultantes de SIFT (Figura 98).

	SIFT	**SURF**
Descriptores localizados	966,6	1049,3
Tiempo para localizar descriptores	0,693 *s*	0,130 *s*
Acierto/Fallo en el cálculo de correspondencias	10,66/1	20,69/1
Tiempo para localizar correspondencias	0,340 *s*	0,237 *s*

Figura 98. Medias de la comparación entre SIFT y SURF

5.3. Cálculo de la rotación

Una vez calculadas las correspondencias entre dos conjuntos de puntos P_1 y P_2 pertenecientes a las imágenes I_1 y I_2 respectivamente, las cuales forman las matrices M_1 y M_2

$$M_1 = \begin{bmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ x_3 & y_3 & z_3 \\ \vdots & \vdots & \vdots \\ x_n & y_n & z_n \end{bmatrix} \quad M_2 = \begin{bmatrix} x'_1 & y'_1 & z'_1 \\ x'_2 & y'_2 & z'_2 \\ x'_3 & y'_3 & z'_3 \\ \vdots & \vdots & \vdots \\ x'_n & y'_n & z'_n \end{bmatrix} \quad (65)$$

es posible definir la rotación entre I_1 y I_2 mediante la matriz de rotación *R*:

$$M_2 = R * M1 \quad (66)$$

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \quad (67)$$

por lo que nos encontramos ante un sistema de *n* ecuaciones con 9 incógnitas (elementos de la matriz *R*), siendo *n* el número de correspondencias detectadas. Como se ha visto en el apartado 5.2.4, el número de correspondencias detectadas es mucho mayor que 9, el número de incógnitas presentes, por lo que nos hallamos ante un sistema sobredeterminado.

Para resolver este problema se ha optado por la solución aportada por Kabsch [27] [26], que se basa en el cálculo de la matriz de covarianza de las matrices M_1 y M_2 (68), para luego realizar la descomposición en valores propios sobre dicha matriz de covarianza, y posteriormente calcular R:

$$A = M_1^T M_2 \quad (68)$$

La descomposición en valores singulares (Singular Value Descomposition, SVD) es una técnica muy utilizada en el campo de la visión por ordenador para descomponer matrices. Una matriz *A* se descompone como

$$A = UDV^T \quad (69)$$

en donde *U* y *V* son matrices ortogonales, mientras que D es una matriz diagonal cuyos valores $(d_1, \dots, d_n)$ son los autovalores propios de *A.* Los vectores propios, autovectores o autovalores de un operador lineal son los vectores no nulos que, cuando son transformados por el orperador, dan lugar a un múltiplo escalar de sí mismos, con lo que no cambian su dirección. Este escalar λ recibe el nombre de valor propio, autovalor o valor característico.

De esta forma, dada una matriz *A* de orden *n*, el escalar λ recibe el nombre de autovalor asociado a *A* si existe algún autovector no nulo *x* de orden $n \times 1$ tal que $A * x = \lambda * \mathrm{x}$.

Dicho de otro modo, siendo *A* una matriz de dimensiones $\mathrm{m}\ x\ \mathrm{n}$:

- *U* es una matriz de dimensiones $m\ x\ n$ con columnas ortogonales. Sus columnas son los valores singulares izquierdos de *A*.

- *D* es una matriz diagonal de dimensiones $n\ x\ n$ con todos sus valores positivos. Tiene todos sus elementos iguales a 0, excepto los que forman parte de su diagonal

principal, que son valores reales, están ordenados de mayor a menor y reciben el nombre de valores singulares de *A*.

- V^T es una matriz ortogonal de dimensiones $n\ x\ n$. Los valores de su diagonal se llaman valores singulares derechos de A.

En definitiva, la descomposición en valores propios consiste en encontrar los valores propios y vectores propios de AA^T y A^TA. Los vectores propios de A^TA los forman las columnas de V, mientras que los vectores propios de AA^T los forman las columnas de U.

Una vez realizada la descomposición, la matriz de rotación *R* puede calcularse como

$$R = U * V^T \quad (70)$$

Por último, para extraer de la matriz de rotación la magnitud de la rotación en cada uno de los ejes (r_x, r_y, r_z), se debe de descomponer la matriz R:

$$\sin(\theta)\begin{bmatrix} 0 & -r_z & r_y \\ r_z & 0 & -r_x \\ -r_y & & 0 \end{bmatrix} = \frac{R - R^T}{2} \quad (71)$$

en donde $r = \{r_x, r_y, r_z\}$, θ es la longitud del vector r y es necesario normalizar r, de forma que $r = r/\theta$.

Para que poder realizar estos cálculos, son necesarios al menos 4 correspondencias entre M_1 y M_2, aunque es recomendable utilizar más datos, ya que aumenta la robustez de los cálculos.

5.4. Evaluación del método para calcular la rotación

Con el fin de dar una cota de error al método propuesto, se ha realizado una prueba consistente en realizar sobre la esfera una rotación de 360 grados respecto al eje *y*.

Para realizar esta prueba es necesario fijar tanto la cámara como la posición de la esfera (en la medida de lo posible). Para conseguir únicamente realizar el giro sobre la esfera en el eje *y*, se ha utilizado una rueda (Figura 99) y se ha fijado su eje a la superficie sobre la que se sitúa la cámara, de forma que el único movimiento posible para la esfera es la rotación sobre el eje *y*, aunque debido a una pequeña inestabilidad en el cilindro de la parte superior de la rueda, se producen pequeños desplazamientos del eje de giro, algo solucionado con las localizaciones posteriores a la primera calculada (sección 4.3.1).

Figura 99. Mecanismo para la rotación de la esfera

Por otra parte, la videocámara utilizada para la captura del flujo de video es una Canon MV600 , que graba en formato miniDV.

Figura 100. Cámara Canon MV600

Para comprobar cuando se ha llegado al giro de 360 grados, se utilizan la imagen inicial (Figura 101) y la imagen en el instante actual (Figura 102), de forma que al hacer una superposición de ambas (Figura 103), si son iguales, se considera que la esfera ha realizado un giro de 360 grados (Figura 104).

De esta forma, cualquier desviación de los cálculos de rotación respecto de las medidas de rotación de 0 grados sobre el eje x, 360 grados sobre el eje *y* y 0 grados en el eje z, será considerado como un error.

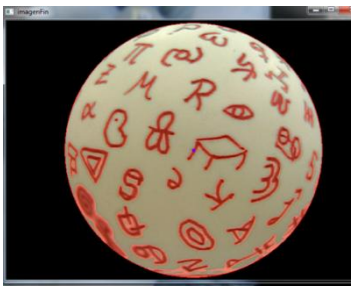


Figura 101. Imagen inicial

Figura 102. Imagen instante n

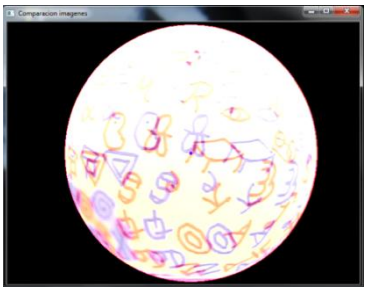


Figura 103. Comparación

Figura 104. Comparación al realizar una rotación de 360 º

En cuanto a la magnitud de la rotación máxima captada por ambos métodos, el método SURF puede detectar una rotación máxima de un ángulo de 37 grados aproximadamente (figuras Figura 112Figura 120), mientras que SIFT detecta una rotación máxima de un ángulo de 27 grados (figuras Figura 105Figura 111). En estas figuras es fácil observar que las correspondencias detectadas por SURF son mayores que en SIFT tal y como se había comprobado en la sección 5.2.4.

CORRESPONDENCIAS SIFT

Figura 105. 75 correspondencias SIFT

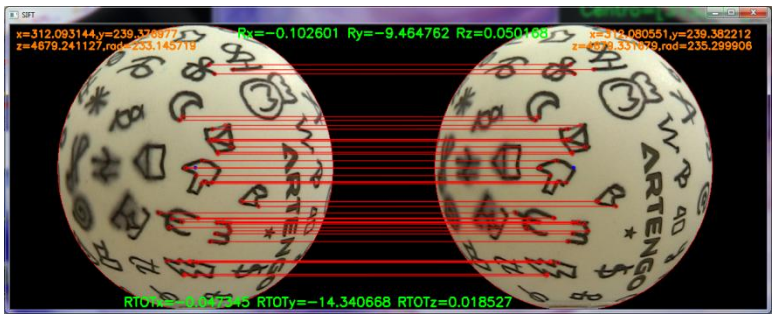


Figura 106. 43 correspondencias SIFT

Figura 107. 27 correspondencias SIFT

Figura 108. 14 correspondencias SIFT

Figura 109. 14 correspondencias SIFT

Figura 110. 5 correspondencias SIFT

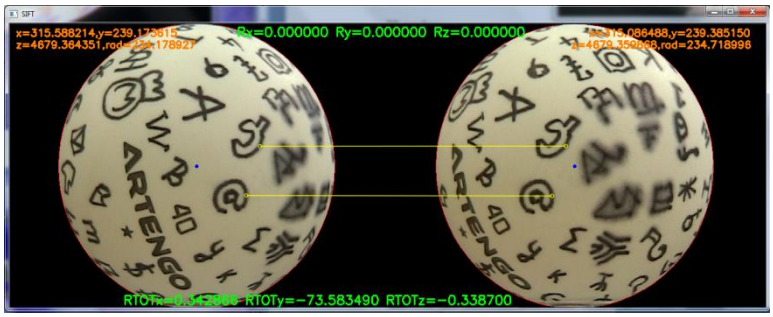


Figura 111. 2 correspondencias SIFT

CORRESPONDENCIAS SURF

Figura 112. 91 correspondencias SURF

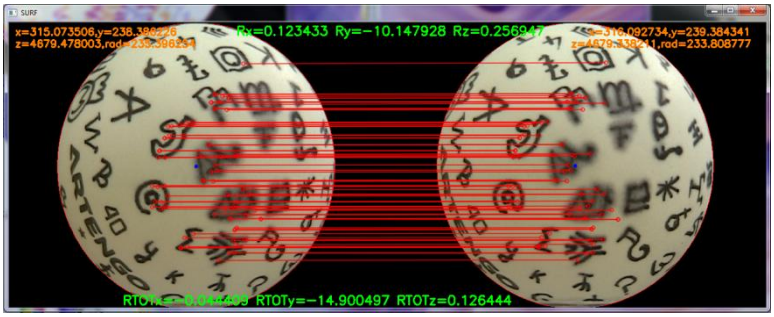


Figura 113. 74 correspondencias SURF

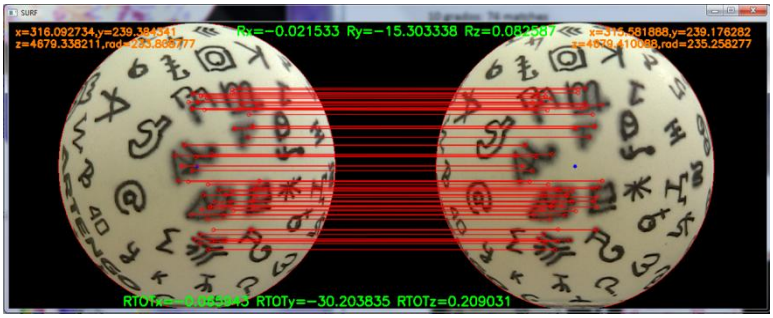


Figura 114. 68 correspondencias SURF

Figura 115. 36 correspondencias SURF

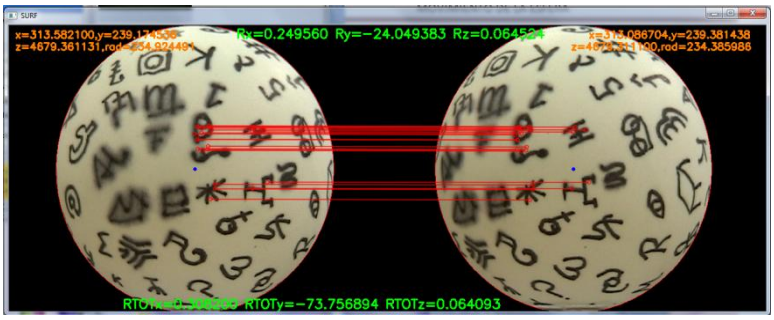


Figura 116. 22 correspondencias SURF

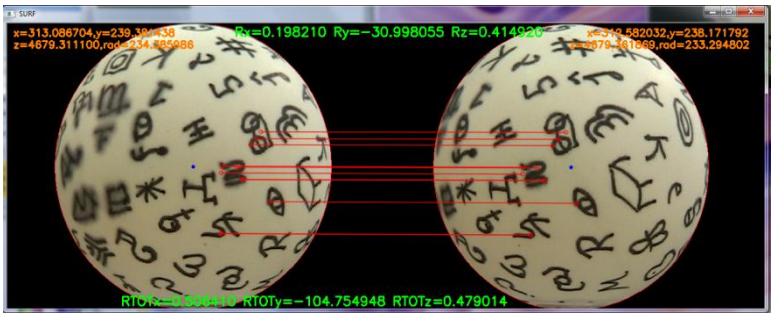


Figura 117. 11 correspondencias SURF

Figura 118. 10 correspondencias SURF

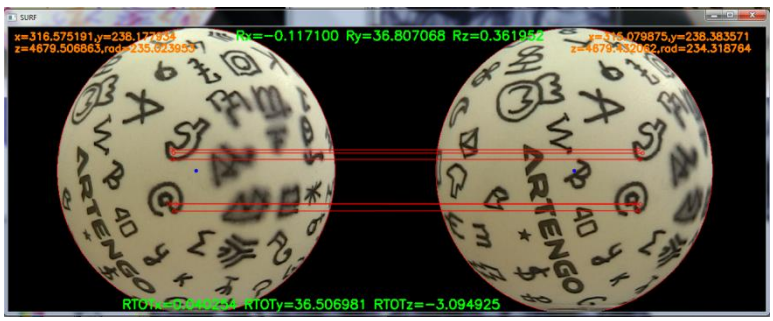


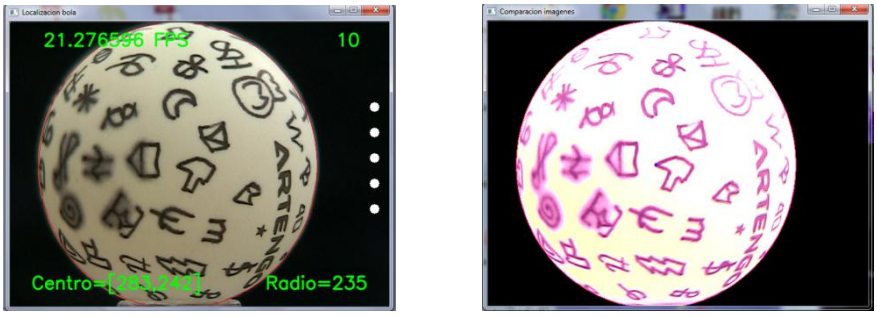
Figura 119. 6 correspondencias SURF

Figura 120. 2 correspondencias SURF

Para comparar la cota de error que alcanza el modelo escogido para calcular la rotación, se han realizado 5 análisis (Figura 121), en los que, de uno a otro, varía la posición en la que se encuentra la esfera (coordenadas C*x* y C*y* del centro de la esfera). Para cada uno de estos escenarios, se ha realizado un giro completo midiendo la rotación tanto con SIFT como con SURF, de forma que sea posible una comparación entre los resultados aportados por los dos tipos de descriptores.

	Cx	Cy
CASO 1	283	242
CASO 2	360	242
CASO 3	312	242
CASO 4	293	243
CASO 5	352	242

Figura 121. Casos de análisis de rotación



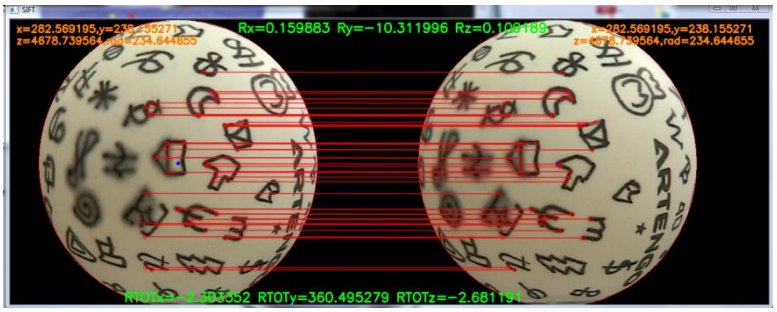


Figura 122. Resultado del giro en el caso 1 para SIFT: X=-2.3, Y=360.49,Z=-2.68

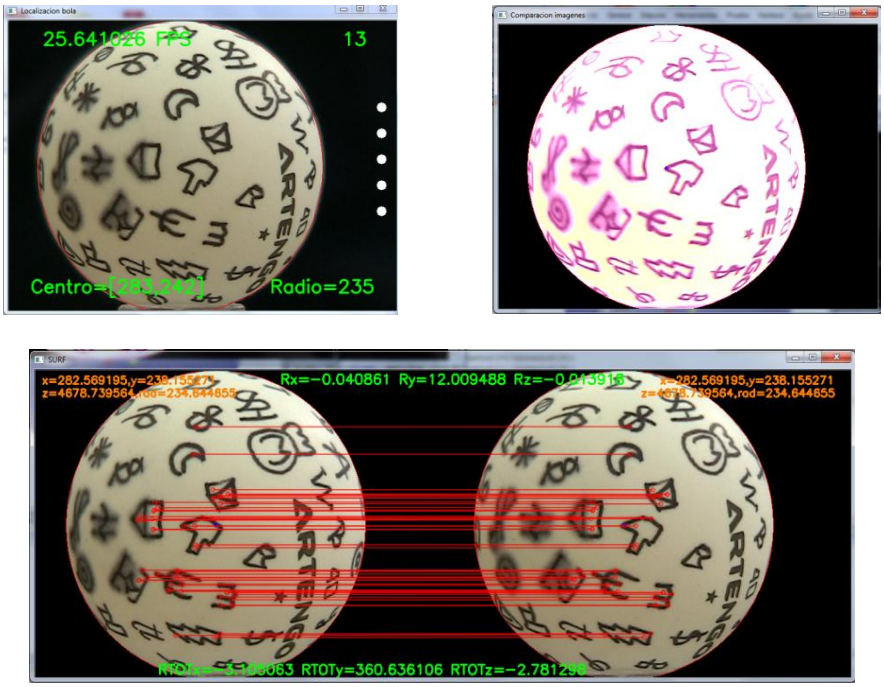


Figura 123. Resultado del giro en el caso 1 para SURF: X=-3.1,Y=360.63, Z=-2.78

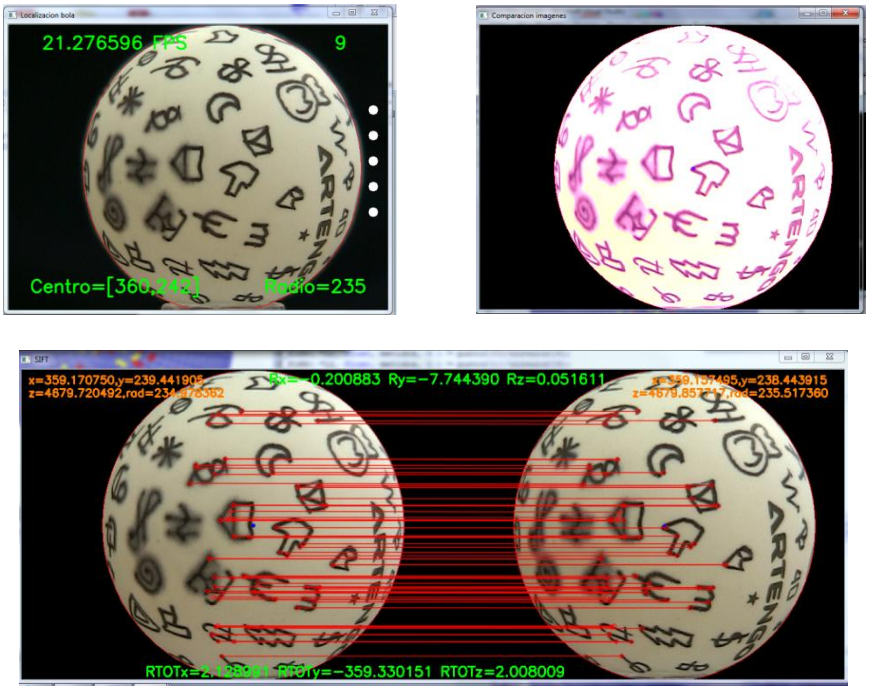


Figura 124. Resultado del giro en el caso 2 para SIFT: X=2.12,Y=-359.33,Z=2.0

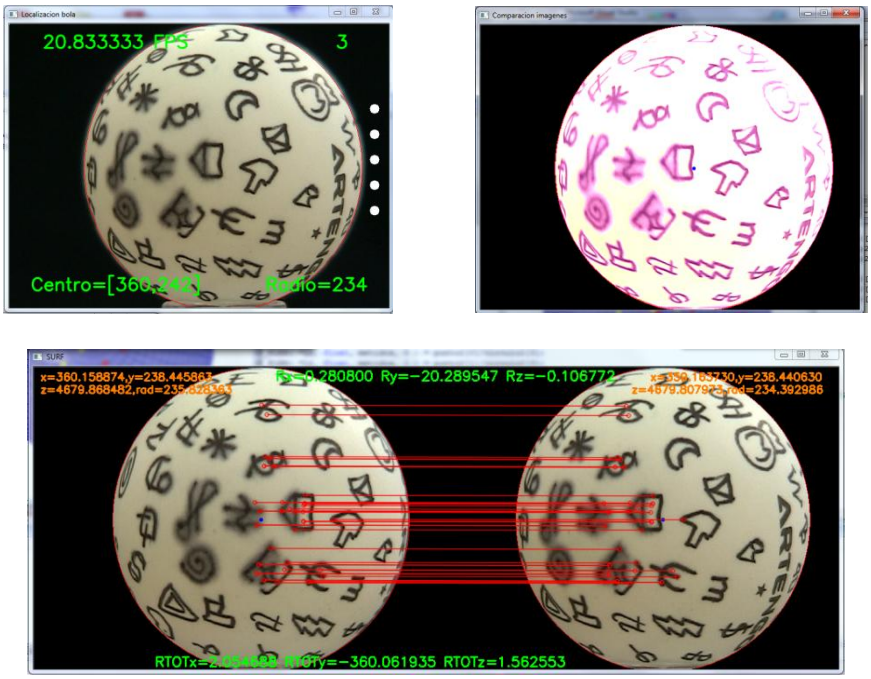


Figura 125. Resultado del giro en el caso 2 para SURF:X=2.0,Y=-360,06,Z=1.56

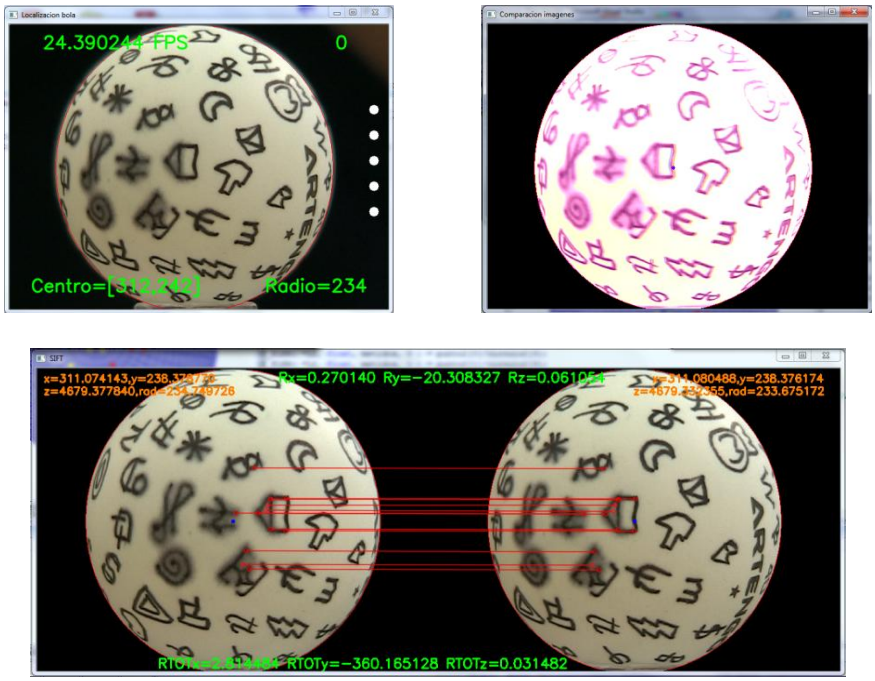


Figura 126. Resultado del giro en el caso 3 para SIFT: X=2.81,Y=-360.16,Z=0.03

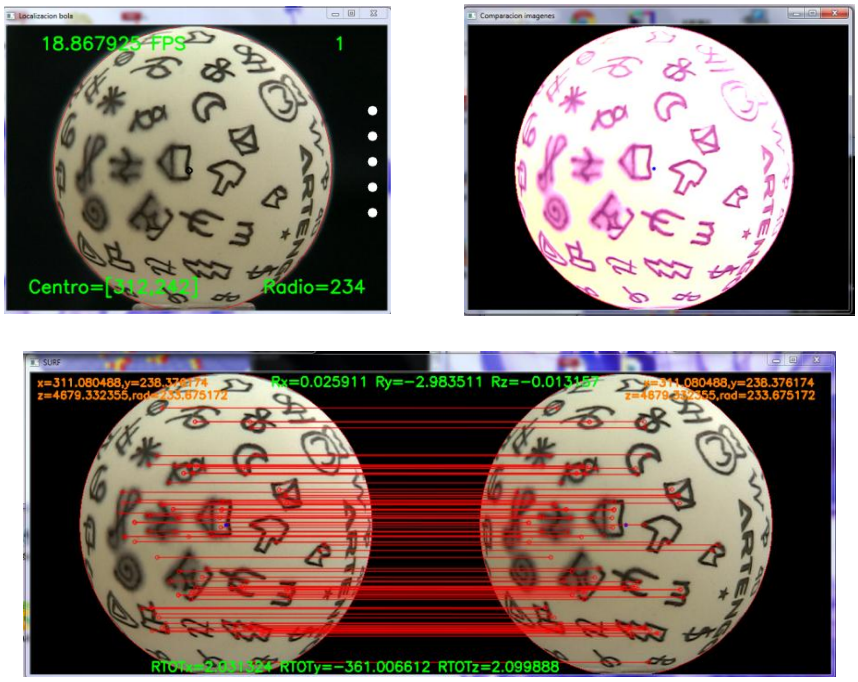


Figura 127. Resultado del giro en el caso 3 para SURF:X=2.03,Y=-361,Z=2.09

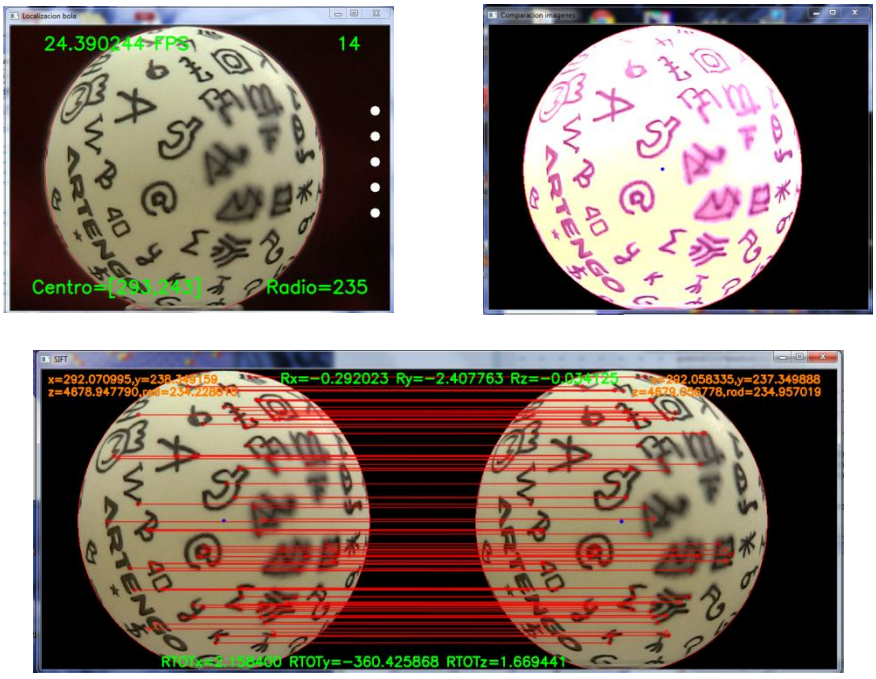


Figura 128. Resultado del giro en el caso 4 para SIFT: X=2.15,Y=-360.42,Z=1.66

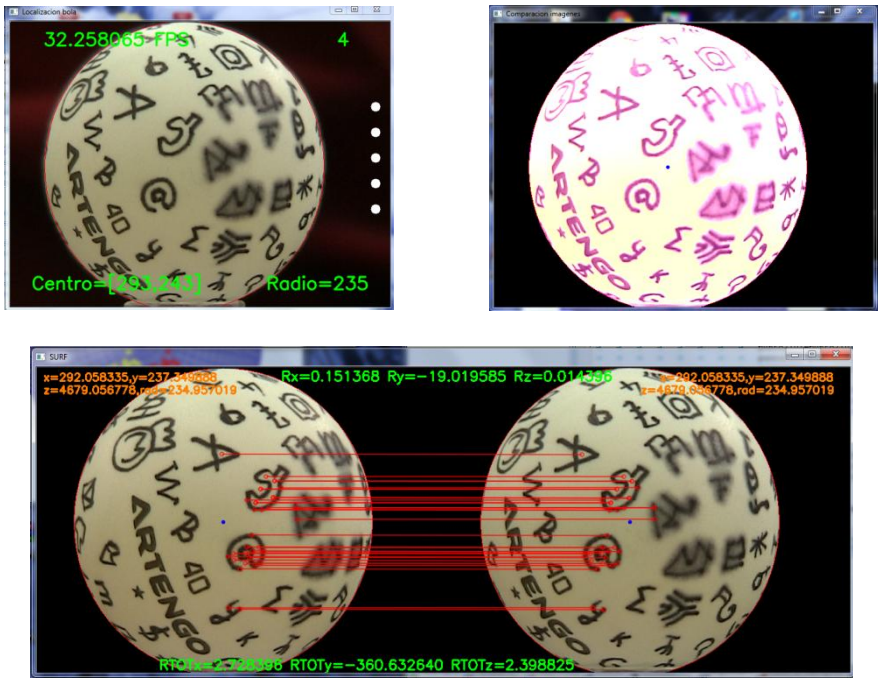


Figura 129. Resultado del giro en el caso 4 para SURF: X=2.72,Y=-360.63,Z=2.39

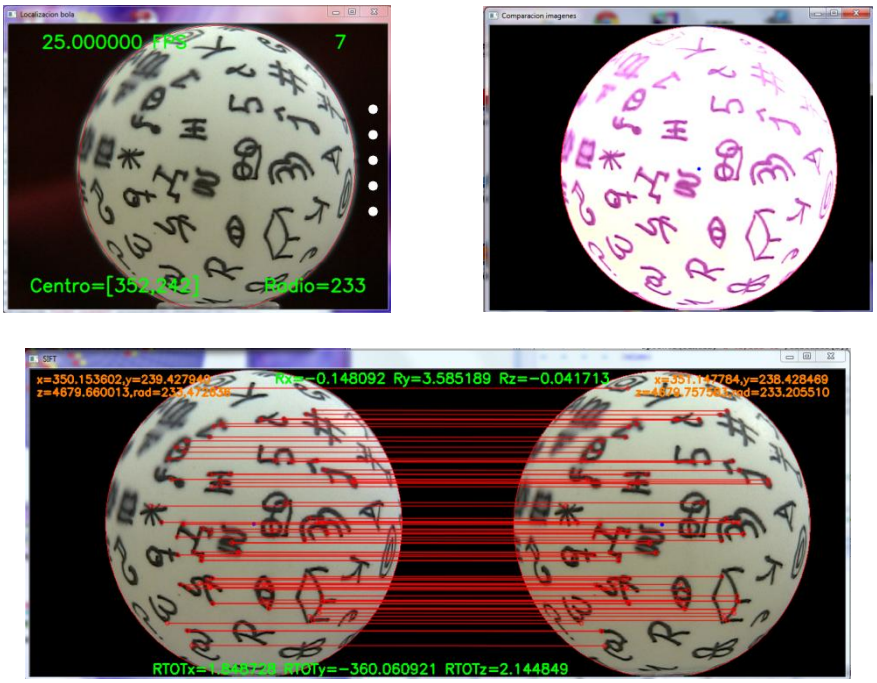


Figura 130. Resultado del giro en el caso 5 para SIFT: X=1.84,Y=-360.06,Z=2.14

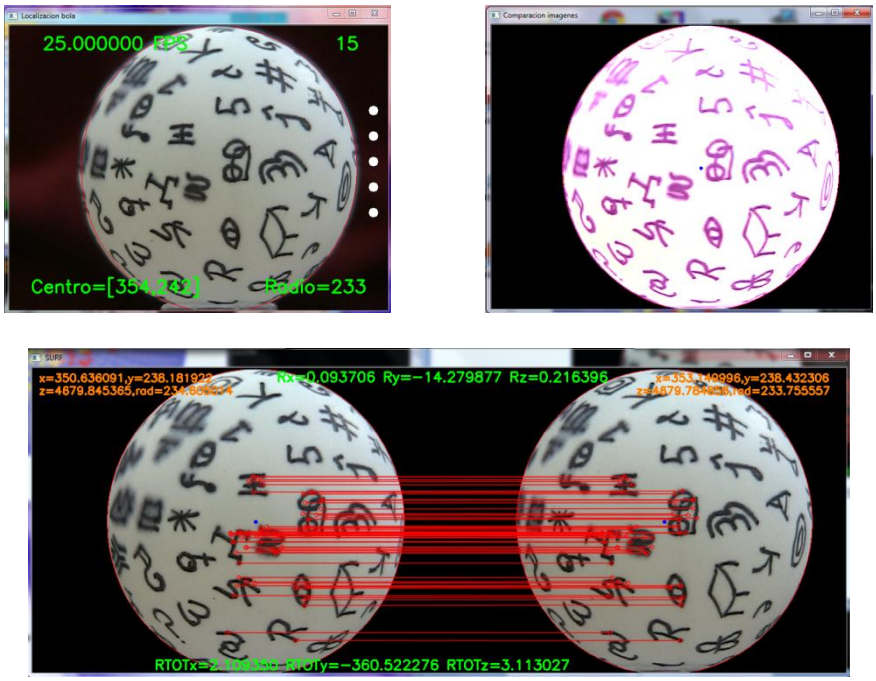


Figura 131. Resultado del giro en el caso 5 para SURF:X=2.10,Y=-360.52,Z=3.11

En la Figura 132, se puede apreciar la comparación entre los errores producidos por cada método, dados en grados. Para hacernos una idea del error total producido por cada método en cada caso, vamos a sumar los ángulos de los errores en los 3 ejes, y los vamos a pasar a *mm*.

	SIFT	**SURF**
CASO 1	X=2,3033 | Y=0,4952 | Z=2,6811	X=3,1056 | Y=0,6361 | Z=2,6811
CASO 2	X=2,1289 | Y=0,6361 | Z=2,7812	X=2,0546 | Y=0,0619 | Z=1,5625
CASO 3	X=2,8144 | Y=0,1651 | Z=0,0314	X=2,0313 | Y=1,0066 | Z=2,0998
CASO 4	X=2,1584 | Y=0,0609 | Z=1,6694	X=2,7283 | Y=0,6326 | Z=2,3988
CASO 5	X=1,8487 | Y=0,6326 | Z=2,1448	X=2,1093 | Y=0,5222 | Z=3,1130

Figura 132. Comparación de errores, en grados, entre SIFT y SURF para cada uno de los casos

Para esto, calculamos la longitud del arco equivalente a 1 grado para la esfera que estamos utilizando, la cual tiene 20 *mm* de radio. Como resultado, se afirma, que un error de un grado en la esfera, produce un error de un arco de longitud 0,34 *mm*, lo que produciría una tabla de errores como la que se puede observar en la Figura 133.

	SIFT	**SURF**
CASO 1	5,4796 ° = 1,86 *mm*	6,422 ° = 2,18 *mm*
CASO 2	5,5462 ° = 1,88 *mm*	3,679 ° = 1,25 *mm*
CASO 3	3,0109 ° = 1,02 *mm*	5,137 ° = 1,74 *mm*
CASO 4	3,8887 ° = 1,32 *mm*	5,759 ° = 1,95 *mm*
CASO 5	4,6261 ° = 1,57 *mm*	5,744 ° = 1,95 *mm*

Figura 133. Comparación de errores, en *mm*, entre SIFT y SURF para cada uno de los casos

Estas mediciones producen una media de error de 1,4919 *mm* y 1,7831 *mm* para SIFT y SURF respectivamente, además de una desviación típica de 0,3664 *mm* en el caso de SIFT y 0,35161 *mm* en el caso de SURF, tal y como se puede observar en la Figura 134.

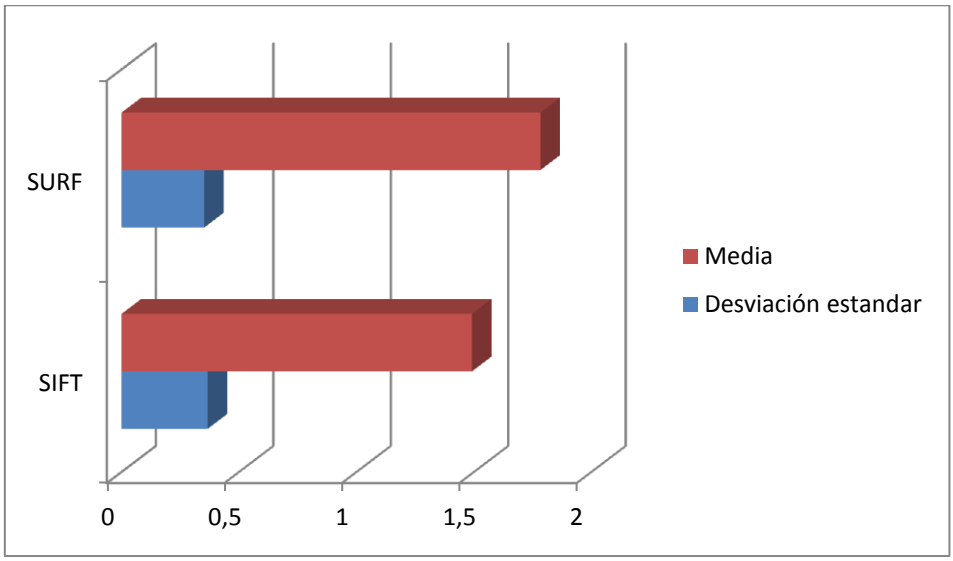


Figura 134. Errores y desviaciones estandar SIFT y SURF

En cuanto al tiempo utilizado por ambos métodos, el método ganador es SURF, ya que como se puede ver en la Figura 98. Medias de la comparación entre SIFT y SURF de la sección 5.2.4, los tiempos de cálculo empleados por éste son menores que los empleados por SIFT, a lo que hay que añadir que SURF es capaz de detectar rotaciones más grandes, tal y como se ha visto en esta sección. Por tanto, SURF necesita de menos iteraciones para calcular la rotación, y además el cálculo de estas rotaciones es más rápido, aunque el error que produce es ligeramente mayor.

En definitiva, y dado que este TFM tiene un carácter médico, en el que la precisión es algo de vital importancia, el método de puntos invariantes ganador es SIFT, ya que es más preciso que SIFT aunque también es más lento.

6. CONCLUSIONES

En este TFM se valida un método de eye tracking mediante el seguimiento de una esfera con marcas aleatorias en su superficie como aproximación al seguimiento del movimiento que realiza un ojo durante una operación de braquiterapia.

El seguimiento se produce en tiempo real, y solo existe un pequeño retraso en el momento en el que se calcula la rotación, La precisión que se consigue es de un error medio de 1,4919 *mm* y 1,7831 *mm* para SIFT y SURF respectivamente, un resultado que al compararlo con los métodos vistos en la bibliografía, sale claramente ganador, aunque como se afirma en la sección 2.2, el fin que buscan los autores de estos trabajos no sean los mismos exactamente.

La conclusión, pues, es que estamos ante un método que puede servir como base para prestar una ayuda eficiente al cirujano en operaciones quirúrgicas de braquiterapia.

Una de las bases para conseguirlo es la utilización de los algoritmos SIFT y SURF para extraer puntos invariantes de la superficie de la esfera, por lo que es necesario probar que estas técnicas también sirven para hacer lo mismo sobre imágenes reales de operaciones quirúrgicas oculares. Para comprobarlo, se han extraído una serie de fotogramas de un video en el cual se realiza una operación de cirugía extraescleral, la cual se utiliza en casos de desprendimiento de retina y cuenta con muchas similitudes respecto a la braquiterapia, y se ha procedido a realizar la extracción de puntos invariantes entre estos.

Para filtrar falsos positivos, es necesario realizar un filtrado de colores, para eliminar la sangre, los brillos especulares, y los hilos negros que aparecen en los fotogramas. Tras realizar este filtrado, se encuentran las correspondencias, tal y como se puede ver a continuación.

La comparación entre las imágenes de las figuras Figura 135Figura 136 produce las correspondencias SIFT que se aprecian en la Figura 137 y las correspondencias SURF que se observan en la Figura 138.

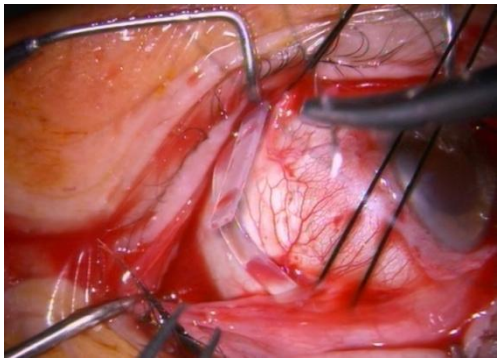


Figura 135. Imagen real 1

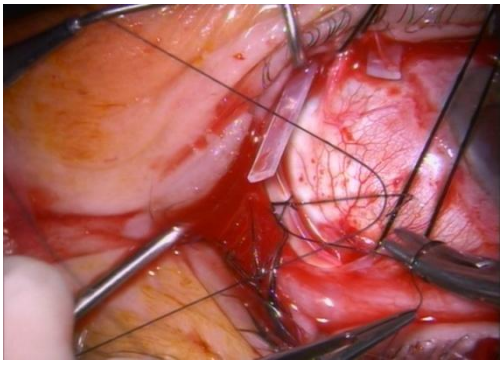


Figura 136. Imagen real 2

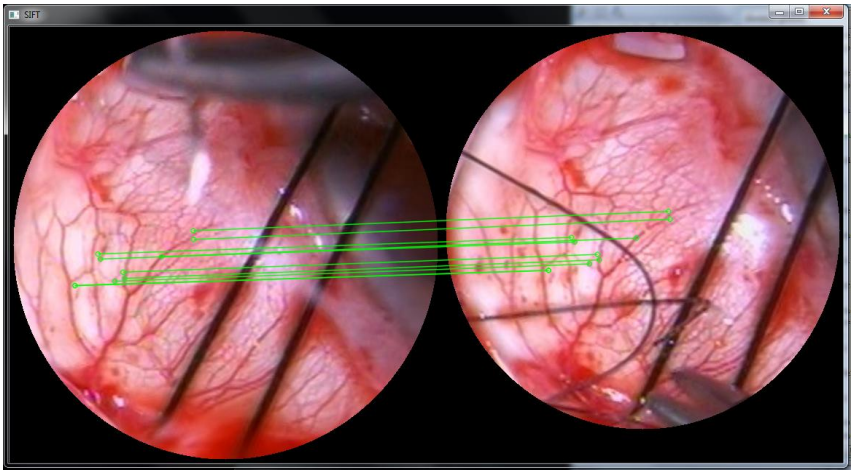


Figura 137. Correspondencias SIFT entre imagen real 1 y 2

En otro ejemplo de extracción de puntos invariantes, la comparación entre las imágenes de las figuras Figura 139Figura 140 produce los puntos SIFT en la Figura 141, y los puntos SURF en la Figura 142.

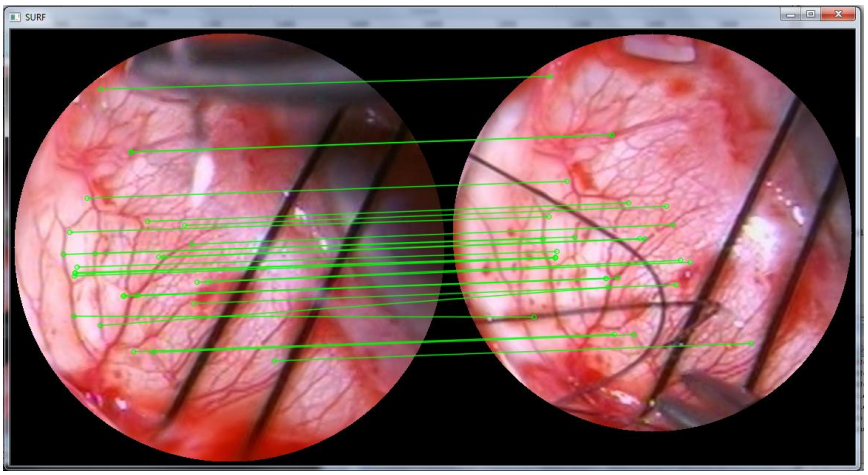


Figura 138. Correspondencias SURF entre imagen real 1 y 2

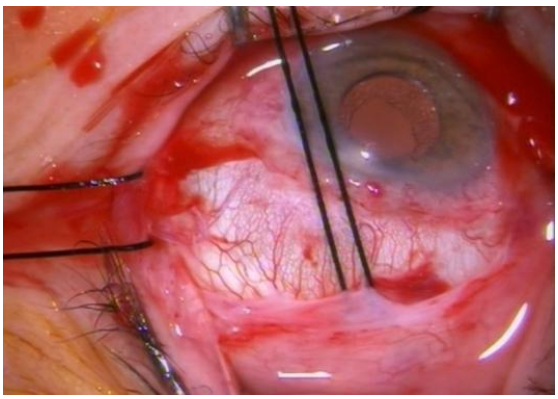


Figura 139. Imagen real 3

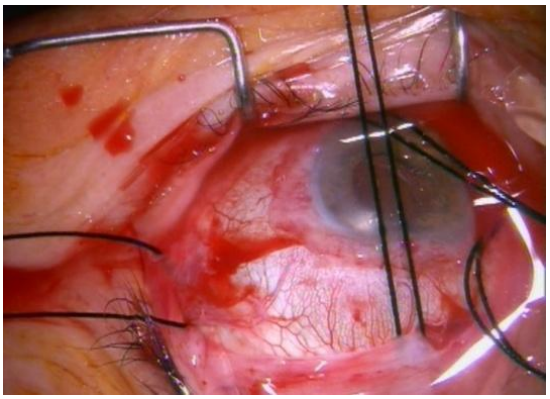


Figura 140. Imagen real 4

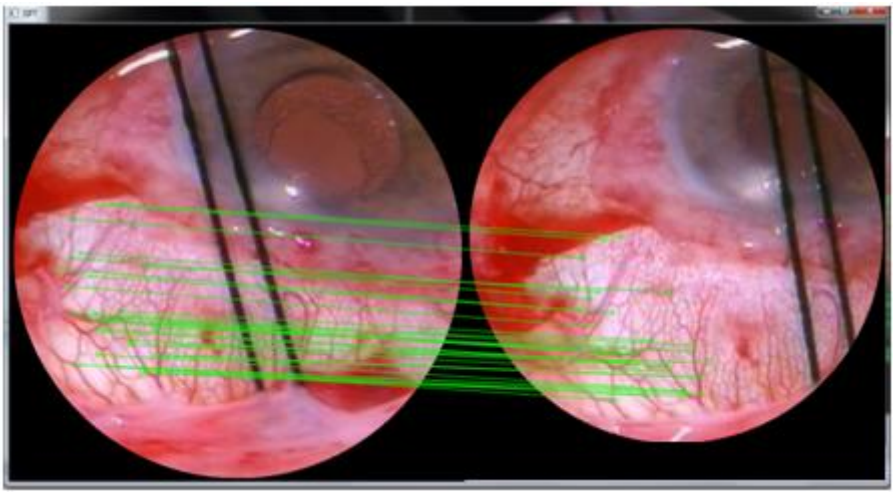


Figura 141. Correspondencias SIFT entre imagen real 3 y 4

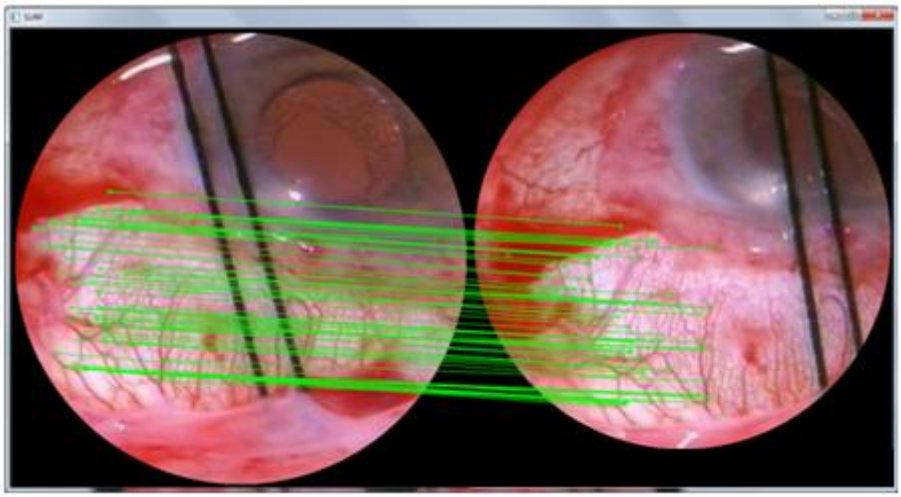


Figura 142. Correspondencias SURF entre imagen real 3 y 4

Por tanto, una vez comprobado que es posible extraer correspondencias de puntos entre imágenes reales casos de operaciones oculares de braquiterapia, no es descabellado pensar que es posible adaptar el método presentado a imágenes reales y conseguir un resultado preciso y robusto.

7. CONTRIBUCIONES DEL TFM Y TRABAJOS FUTUROS

El eye tracking, a lo largo de la historia, ha tenido unos objetivos distintos a los que se persiguen en este TFM. Por este motivo es necesario desmarcarse de éstos, ya que ni persiguen el mismo objetivo ni utilizan técnicas similares para lograrlo.

Mediante un enfoque totalmente distinto al empleado en cualquier trabajo de la bibliografía, se ha conseguido validar un algoritmo de tracking óptico capaz de monitorizar el movimiento ocular sin la necesidad de tener siempre presente información que se antoja imprescindible para los trabajos de este tipo: la presencia de la pupila y el iris en la imagen. Este seguimiento se produce, además, con un error mínimo y en tiempo real, confirmando las hipótesis planteadas en el apartado 1.3.

De esta forma es posible obtener en todo momento la información necesaria para calcular la rotación que sufre el ojo mediante los vasos capilares propios de la pared ocular, a la vista gracias a la apertura de la conjuntiva.

Esta manera de proceder resulta perfecta para operaciones oculares tales como la braquiterapia, tratamientos de desprendimiento de retina o cualquier otra operación en la cual se de la circunstancia por la cual no esté presente información relativa a la posición de la pupila/iris, y no se pueden utilizar recursos tales como las imágenes de Purkinje, el vector PCCR o ajuste de plantillas. En general, cualquier tipo de cirugía ocular que necesite localizar un punto determinado en zonas oculares sin tener la referencia de la posición de la pupila puede ser una potencial benefactora de este método.

Pese a la novedad y precisión mostrada por este método, cabe destacar algunas de las debilidades que deberían ser objeto de superación en trabajos futuros:

- El problema más grande es la correcta localización del centro de la esfera, ya que este tiene mucho peso en la retroproyección de los puntos localizados sobre la superficie de la esfera, y por ende, en el cálculo de la rotación, ya que depende completamente de la precisión con la que se calcule este centro. Como se explica en la sección 4.2.2, al localizar la esfera mediante el método de momentos, el umbralizado que se realiza para detectar el contorno de la misma debe de ser muy preciso, cosa que no ocurre cuando las marcas dibujadas en la superficie aparecen en las zonas más exteriores de la esfera, momentos en los cuales el contorno calculado no se ajusta al real. Además, si ya es complicado realizar la umbralización en unas condiciones ideales, en imágenes reales de operaciones quirúrgicas, en las cuales no es posible ver todo el contorno del ojo debido a que éste se encuentra en las cuencas oculares, la localización del centro del globo ocular es un reto que implica la investigación de otros enfoques.

- Aunque la precisión alcanzada es buena, otro problema que presenta este método es la imposibilidad de calcular con precisión las coordenadas 3D de los puntos localizados mediante el uso de una sola cámara. Esta precisión mejoraría ostensiblemente con la utilización de dos cámaras en lugar de una, con lo que no sería necesario calcular el centro de la esfera: utilizando dos cámaras, sería posible calcular los puntos 3D exactamente, sin necesidad de realizar retroproyección ninguna. De esta forma, el problema del cálculo de la rotación entre dos imágenes se vería reducido al cálculo de correspondencias sin falsos positivos, y al cálculo de la rotación a partir de la matriz de covarianzas de los conjuntos de puntos de las correspondencias formadas por estas dos imágenes.

•

BIBLIOGRAFÍA

[1] J. Allen, R.Y.D. Xu, and J.S. Jin, "Object tracking using camshift algortihm and multiple quantized feature spaces," School of Information Technologies. University of Sydney, Sydney, Australia, 2006.

[2] S. Baluja and D. Pomerlau, "Non-Intrusive Gaze Tracking Using Artificial Neural Networks," School of Computer Science. Carnegie Mellon University, Pittsburgh, USA, 1994.

[3] H. Bay, T. Tuytelaars, and L.V. Gool, "SURF: Speeded up robust features," in *9th European Conference on Computer Vision*, Graz, Austria, 2006, pp. 404-417.

[4] J.Y Bouguet, "Pyramidal Implementation of the Lucas Kanade Feature Tracker. Description of the algoritm.," Intel Corporation. Microprocessor Research Labs, 2001.

[5] M. Brown, M., Zrenner, E. Marmor, M. Brigell, and M. Bach, "ISCEV Standard for Clinical Electro-oculography (EOG)," *Documenta Ophthalmologica*, vol. 113, no. 3, pp. 205-212, 2006.

[6] A. Bulling, D. Roggen, and G. Tröster, "It´s in Your Eyes - Towards Context-Awareness and Mobile HCI Using Wearable EOG Googles," in *Proc. of the 10th International Conference on Ubiquitous Computing*, Seoul, South Korea, 2008, pp. 84-93.

[7] A. Bulling, J.A. Ward, H. Gellersen, and G. Trösten, "Eye Movement Analysis for Activity Recognition Using Electrooculography," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 4, pp. 741-753, 2011.

[8] G.T. Buswell, *How adults read*. Chicago, USA: University of Chicago Press, 1937.

[9] G.T Buswell, *How people Look at Pictures*. Chicago, USA: University of Chicago Press, 1935.

[10] C. Collet, A. Finkel, and R. Gherbi, "CapRe: a gaze tracking system in man-machine interaction," in *Proceedings of the IEEE International Conference on Intelligent Engineering Systems*, Budapest, Hungary, 1997, pp. 577-581.

[11] C. Colombo, S. Andronico, and P. Dario, "Prototype of a vision-based faze driven man-machine interface," in *RSJ International Conference on Intelligent Robots and Systems*, Pittsburgh, USA, 1995, pp. 188-192.

[12] C. Colombo and A. Del Bimbo, "Interacting through eyes," *Robotics and Autonomous Systems*, vol. 19, no. 3-4, pp. 359-368, 1997.

[13] R. Dodge and T.S. Cline, "The angle velocity of eye movements," *Psychological Review*, vol. 8, no. 2, pp. 145-157, 1901.

[14] R. Duda and Z. Hart, "Use of the hough Transformation to detect lines and curves in pictures," *Communications of the ACM*, vol. 15, pp. 11-15, January 1972.

[15] L. A. Frey, K.P. White, and T.E. Hutchinson, "Eye-Gaze Word Processing," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 20, no. 4, pp. 944-950, July 1990.

[16] D.M. Gavrila and L.S Davis, "3-D model-based tracking of humans in action: a multiview approach," in *Conference on Computer Vision and Pattern Recognition*, San Francisco, USA, 1996, p. 73.

[17] Y.L. Grand, *Light, Color and Vision*. London, England: Chapman and Hall, 1957.

[18] E.D. Guestrin and M. Eizeman, "Remote point-of-gaze estimation with single point personal calibration based on the pupil boundary and corneal reflections," in *24th Canadian Conference on Electrical and Computer Engineering (CCECE)*, Toronto, Canada, 2011, pp. 971-976.

[19] E.D Guestrin and M. Eizenman, "General theory of remote gaze estimation using the pupil center and corneal reflections," *IEE transactions on biomedical engineering*, vol. 53, no. 6, pp. 1124-1133, June 2006.

[20] J. Heinzmann and A. Zelinsky, "3-D Facial Pose and Gaze Point Estimation using a Robust Real-Time Tracking Paradigm," in *Third IEEE International Conference on Automatic Face and Gesture Recognition*, Nara, Japan, 1998, pp. 142 - 147.

[21] (2012) http://en.wikipedia.org/wiki/Eye_tracking.

[22] E. Huey, *The Psychology and Pedagogy of Reading*.: Macmillan, 1908.

[23] T.E. Hutchinson, K.P. White JR., W.N. Martin, K.C Reichert, and L.A. Frey, "Human-Computer Interaction Using Eye-Gaze Input," *IEE transactions on systems, man and cybernetics*, vol. 19, no. 6, pp. 1527-1534, November 1989.

[24] E. Javal, "Essai sur la Physiologie de la Lecture," *Annales d'Oculistique*, vol. 80, pp. 135-147, 1879.

[25] C.H. Judd, C.N. McAllister, and W.M. Steel, "General introduction to a series of studies of eye movements by means of kinetoscopic photographs.," *Psychological Review, Monograph Supplements*, vol. 7, no. 1, pp. 1-16, 1905.

[26] W. Kabsch, "A discussion of the solution for the best rotation to relate two set of vectors," *Crystal Physics, Diffraction, Theoretical and General Crystallography*, vol. 34, no. 5, pp. 827-828, September 1978.

[27] W. Kabsch, "A solution of the best rotation to relate two set of vectors," *Acta Crystallographica*, vol. 32, no. 5, pp. 922-923, 1976.

[28] D.G. Lowe, "Object recognition from local scale-invariant features," *Proceedings of the Seventh IEEE International Conference on Computer Vision*, vol. 2, pp. 1150-1157, August 1999.

[29] B.D. Lucas and L. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proceedings of the 7th international joint conference on Artificial intelligence*, vol. 2, San Francisco, USA, 1981, pp. 674-679.

[30] J. Luo and O. Gwun, "A Comparison of SIFT, PCA-SIFT and SURF," *International Journal of Image Processing*, vol. 3, no. 4, pp. 143-152, 2009.

[31] Z. Lv, X. Wu, M. Li, and C. Zang, "Implementation of the EOG-based Human Computer Interface System," in *Proceedings of the 2nd International Conference on Bioinformatics and Biomedical Engineering*, Shangai, China, 2008, pp. 2188-2191.

[32] J. Merchant, R. Morrissette, and J.L. Porterfield, "Remote Measurement of Eye Direction Allowing Subject Motion Over One Cubic Foot of Space," *IEE transactions on biomedical engineering*, vol. 21, no. 4, pp. 309-317, July 1974.

[33] M. R.M. Mimica and C.H. Morimoto, "A computer Vision Framework for Eye Gaze Tracking," in *XVI Brazilian Symposium on Computer Graphics and Image Processing*, São Carlos, Brazil, 2003, pp. 406-412.

[34] C.H. Morimoto, D. Koons, A. Amir, M. Flickner, and S. Zhai, "Keeping an Eye for HCI," in *XII Brazilian Symposium on Computer Graphics and Image Processing*, Campinas, Brazil, 1999, pp. 171-176.

[35] T. Nagamatsu, J. Kamahara, and N. Tanaka, "3D Gaze Tracking with Easy Calibration Using Stereo Cameras for Robot and Human Communication," in *Te 17th International Symposium on Robot and Human Interactive Communication*, Munich, Germany, 2008, pp. 59-64.

[36] T. Ohno, N. Mukawa, and A. Yoshikawa, "FreeGaze: A gaze Tracking system for Everyday Gaze Interaction," in *Proceedings of the symposium on ETRA 2002: eye tracking research & applications symposium*, New Orleans, USA, 2002, pp. 125-132.

[37] K.R. Park, "Robust Gaze Estimation for Human Computer Interaction," in *Proceedings of the 9th Pacific Rim international conference on Artificial intelligence*, Guilin, China, 2006, pp. 1222-1226.

[38] K. Rayner, "Eye movements in reading and information processing.," *Psychological Bulletin*, vol. 85, no. 3, pp. 618-660, May 1978.

[39] A.M. Romero and M. Cazorla, "Comparativa de detectores de caracteristicas visuales y su aplicacion al SLAM," in *Workshop of Physical Agents*, Caceres, Spain, 2009.

[40] H.S. Sawhney, J. Oliensis, and A.R. Hanson, "Description and reconstruction from image trajectories of rotational motion," in *Third International Conference on Computer Vision*, Osaka, Japan, 1990, pp. 494-498.

[41] S. Shih and J. Liu, "A Novel Approach to 3-D Gaze Tracking Using Stereo Cameras," *IEEE transactions on systems, man and cybernetics.*, vol. 34, no. 1, pp. 234-245, February 2004.

[42] J. Sun, C. Yang, J. Liu, and X. Yang, "Gaze tracking based on similarity between spatial triangles and two-stage calibration," *Electronics Letters*, vol. 47, no. 4, pp. 254-255, February 2011.

[43] S.E. Taylor, *The dynamic activity of reading: A model of the process.*: Educational Developmental Laboratories, 1971.

[44] A.B. Usakli, S. Gurkan, F. Aloise, G. Vechiato, and F. Babiloni, "On the Use of Electrooculogram for Efficient Human Computer Interfaces," *Computational Intelligence and Neuroscience*, 2009.

[45] C. Valgren and A. Lilienthal, "SIFT, SURF and Seasons: Long-term Outdoor Localization Using Local Features," in *3rd European Conference on Mobile Robots*, vol. 58, Freiburg, Germany, Ferb 2007, pp. 149-156.

[46] J. Wang and E. Sung, "Study on Eye Gaze Estimation," *IEEE transactions on systems, man and cybernetics*, vol. 32, no. 3, p. 2002, June 2002.

[47] K.P. White JR, T.E. Hutchinson, and J.M. Carley, "Spatially Dynamic Calibration of an Eye-Tracking system," *IEEE transactions on systems, man and cybernetics*, vol. 23, no. 4, pp. 1162-1168, July 1993.

[48] Wikipedia. (2012) [Online]. http://es.wikipedia.org/wiki/Tiempo_real

[49] Wikipedia. (2012) [Online]. http://en.wikipedia.org/wiki/Levenberg%E2%80%93Marquardt_algorithm

[50] Wikipedia. (2012) [Online]. http://en.wikipedia.org/wiki/Image_moment

[51] X. Yang et al., "A gaze tracking scheme for eye-based intelligent control," in *Proceedings of the 8th World Congress on Intelligent Control and Automation*, Jinan, China, 2010, pp. 50-55.

[52] C. Yang et al., "A gray difference-based pre-processing for gaze tracking," in *IEEE 10th International Conference on Signal Processing*, Beijing, China, 2010, pp. 1293-1296.

[53] A.L. Yarbus, *Eye movements and vision*, Plenum Press, Ed., 1967.

[54] D.H. Yoo, M.J. Chung, D.B. Ju, and I.H. Choi, "Non-intrusive Eye Gaze Estimation using a Projective Invariant under Head Movement," in *Proceedings of the 2006 IEEE International Conference on Robotics and Automation*, Orlando, USA, 2006, pp. 3443-3448.

[55] D.H. Yoo, J.H. Kim, B.R. Lee, and M.J. Chung, "Non-contact eye Gaze Tracking System by Mapping of Corneal Reflections," in *Proceedings of the Fifth IEEE International Conference on Automatic Face and Gesture Recognition*, Washington, USA, 2002, p. 101.

[56] T. Zaveri, J. Winters, M. Wankhede, and I. Park, "A fast and accurate method for discriminating five choices with EOG," Department of Biomedical Engineering, Case Western Reserve University Clevelando FES Center of Excellence , Cleveland, USA,.

[57] Z. Zhang, "A flexible new technique for camera calibration," *IEEE transactions on pattern analysis and machine intelligence*, vol. 22, no. 11, pp. 1330-1334, November 2000.

[58] Z. Zhu and Q. Ji, "Novel Eye Gaze Tracking Techniques under natural head movement," in *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition* , vol. 1, San Diego, USA, 2005, pp. 918-923.