



**Universidad**  
Zaragoza

## Trabajo de Fin de Grado

# **Simulador de prótesis visual en entornos 360° con gafas de realidad virtual**

Autora:

**María Santos Villafranca**

Directores:

Jesús Bermúdez Cameo  
José Jesús Guerrero Campo

Escuela de Ingeniería y Arquitectura  
2019



# Simulador de prótesis visual en entornos 360° con gafas de realidad virtual

## RESUMEN

El sentido del que más dependen las personas es la visión, la mayor parte de la información recibida es a través de éste sentido. Es muy importante en la vida cotidiana, se usa constantemente hasta en las tareas más sencillas, aquellas como reconocer objetos, personas, etc. Sin embargo, algunas patologías o enfermedades degenerativas pueden causar la ceguera total o parcial. Como alternativa para paliar los efectos de la ceguera, existen diferentes tipos de prótesis visuales que se pueden situar en la retina, la corteza visual o el nervio óptico en función del problema que cause la ceguera. Una de las posibilidades es que estas prótesis tengan una microcámara que capta la información visual que posteriormente es convertida a estimulaciones eléctricas lo que permite ver puntos de luz denominados fosfenos. Desgraciadamente, el campo de visión actual de estos dispositivos es en torno a los 20° por lo que se están investigando diferentes representaciones de mapas de fosfenos para mejorar la interacción de los pacientes con el entorno.

En este proyecto se ha desarrollado un simulador de visión fosfénica que permite, mediante las gafas de realidad virtual Oculus Rift DK2, mostrar un entorno virtual de 360° en tiempo real a partir de imágenes panorámicas. Este simulador permite introducir dos tipos de representaciones de mapas de fosfenos, el método Downsampling y el método SIE-OMS. El primer método reduce la resolución de color y espacial de la imagen panorámica y la transforma a un mapa de fosfenos mientras que el segundo realiza una extracción de objetos mediante un algoritmo de aprendizaje automático y lo combina con el layout de la imagen. El código se ha implementado en lenguaje C++, con la propia interfaz de programación de aplicaciones (API) de Oculus y las librerías OpenCV para el manejo de imágenes. Ha sido necesario realizar un cursillo previo para aprender a manejar las librerías.

Los resultados obtenidos han sido bastante satisfactorios cumpliendo todos los objetivos planteados. El objetivo principal era crear el simulador de prótesis visual que pueda ser empleado para el avance de la investigación de este campo.



# Índice

---

1. Introducción .....	1
1.1. Motivación .....	2
1.2. Estado del Arte .....	3
1.3. Objetivos y alcance .....	4
1.4. Entorno de trabajo .....	5
2. Modelo de cámara .....	9
2.1. Modelo de proyección perspectiva .....	9
2.2. Modelo de proyección esférica .....	12
3. Representación panorámica en gafas de realidad virtual .....	15
4. Mapa de Fosfenos .....	19
4.1. Tipos de Implantes .....	19
4.1.1. Implantes retinales .....	19
4.1.2. Nervio óptico .....	20
4.1.3. Cortex Visual .....	20
4.2. Representación de Fosfenos .....	21
5. Representaciones simuladas .....	25
5.1. Método de Downsampling .....	25
5.2. Método de SIE-OMS .....	28
6. Conclusiones .....	31
Anexo I . Fases del proyecto y duración .....	33
Bibliografía .....	37



# Índice de Figuras

---

Figura 1.1. Mapa de fosfenos de una habitación.....	2
Figura 1.2. Estructura de la retina [4] .....	3
Figura 1.3. Esquema general del código.....	5
Figura 1.4. Vista expandida de las gafas Oculus Rift DK2 [14].....	6
Figura 1.5. Distorsión de tipo barrilete .....	7
Figura 1.6. Corrección de la distorsión de tipo barrilete [11] .....	7
Figura 2.1. Modelo de proyección de la cámara pin-hole [19] .....	10
Figura 2.2. Transformación proyectiva de 3D a 2D [20] .....	10
Figura 2.3. Transformación matemática de 3D a 2D [20].....	11
Figura 2.4. Centro óptico de la imagen [20] .....	11
Figura 2.5. Transformación geométrica de la referencia cámara a la referencia absoluta [20] .....	11
Figura 2.6. Deformación en imágenes panorámicas.....	12
Figura 2.7. Variables de la proyección esférica .....	13
Figura 2.8. Esfera unidad [18].....	13
Figura 3.1. Ecuaciones de paso de cuaternios a matriz de rotación.....	16
Figura 3.2. Representación de la imagen como una matriz .....	16
Figura 3.3. Matriz de cambio de referencia del entorno simulado .....	17
Figura 3.4. Correspondencia entre imagen del visor e imagen panorámica.....	18
Figura 3.5. Comparativa de imagen de partida e imagen final.....	18
Figura 4.1. Comparación entre los dos modelos de prótesis [22] .....	20
Figura 4.2. Cálculo del radio de visión con un ángulo de 20º .....	21
Figura 4.3. Imagen a color representada en las Oculus .....	23
Figura 4.4. Aparición de colores por la distorsión.....	23
Figura 4.5. Representación de fosfenos en color verde .....	23
Figura 5.1. Ejemplo de habitación .....	26
Figura 5.2. Mapa de fosfenos para 10.000 fosfenos.....	26
Figura 5.3. Representación en las Oculus para 10.000 fosfenos .....	26

Figura 5.4. Mapa de fosfenos para 1.000 fosfenos.....	27
Figura 5.5. Representación del mapa de fosfenos para 1.000 fosfenos en Oculus .....	27
Figura 5.6. Vista con visión reducida a 20º para 679 fosfenos .....	27
Figura 5.7. Representación del método SIE-OMS.....	28
Figura 5.8. Como se forma la imagen de SIE-OMS .....	28
Figura 5.9. Visualización de la imagen de layouts .....	29
Figura 5.10. Mapa de fosfenos para 10.000 fosfenos para imagen de layouts .....	29
Figura 5.11. Mapa de fosfenos para 1.000 fosfenos para imagen de layouts .....	29
Figura 5.12. Representación del mapa de fosfenos para 10.000 fosfenos para imagen de layouts en Oculus .....	30
Figura 5.13. Representación del mapa de fosfenos para 1.000 fosfenos para imagen de layouts en Oculus .....	30
Figura 5.14. Mapa de fosfenos con 981 fosfenos y 50º de visión .....	30
Figura A.1. Cronograma del proyecto .....	33
Figura A.2. Proyección perspectiva aplicada a una imagen .....	34
Figura A.3. Ángulos de roll pitch yaw [32].....	35



# Capítulo 1

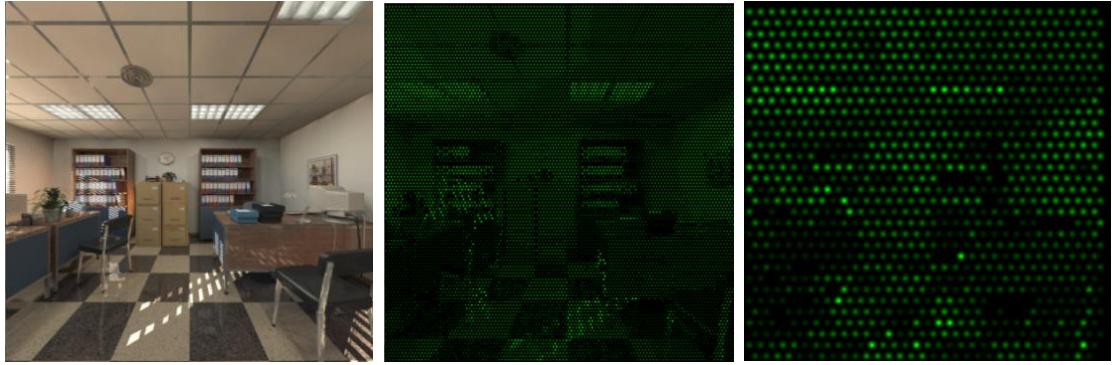
## Introducción

---

El sentido más desarrollado del ser humano es la visión, dado que la mayor parte de la información recibida es a través de éste sentido. Desgraciadamente, no todas las personas lo poseen, por eso la visión artificial para asistencia a personas con discapacidad visual es un campo de investigación tan importante actualmente. Según la Organización Mundial de la Salud (OMS) aproximadamente 1300 millones de personas viven con alguna deficiencia visual, de las cuales 36 millones son ciegas [1].

La visión artificial abarca muchos campos, entre ellos el reconocimiento de caras, seguimiento de personas, control autónomo de vehículos aéreos no tripulados (UAVs), etc. Sin embargo, el ámbito en el que se centra este proyecto es en el de las prótesis visuales.

Las prótesis visuales tratan de mejorar la visión de personas con enfermedades degenerativas como la retinitis pigmentosa. En este caso, los fotorreceptores mueren pero el circuito retinal restante no, el cual se estimula eléctricamente lo cual permite que la persona pueda percibir círculos de luz llamados fosfenos tal y como se muestra en la Figura 1.1:



(a) Foto original

(b) Mapa de fosfenos con 10.000 fosfenos

(c) Mapa de fosfenos con 1.000 fosfenos

Figura 1.1. Mapa de fosfenos de una habitación

En estas imágenes se puede observar la importancia del número de fosfenos para poder interpretar el entorno u objetos del mismo.

Las personas con estas prótesis disponen de pocos fosfenos y lo normal es que únicamente posean un campo de visión generalmente inferior a  $20^\circ$ , cualquier prueba experimental les resulta muy complicada. Por ello, tal y como explica el artículo [2], se han generado los simuladores de visión fosfénica (SPV) para poder realizar experimentos no invasivos en sujetos con visión normal e implementar técnicas que ayuden a interpretar mejor el entorno.

## 1.1. Motivación

La visión es muy importante en la vida cotidiana, se usa constantemente hasta en las tareas más sencillas, por ello, resulta muy frustrante carecer de este sentido sobre todo si alguna vez se ha disfrutado de él. Éste es el caso de personas ciegas que no lo son de nacimiento, sino que han perdido la vista debido a enfermedades degenerativas.

Por eso resulta tan importante este campo de investigación, para ayudar a estos pacientes a volver a obtener información visual del entorno o al menos de los elementos más importantes de éste, como por ejemplo reconocer en que estancia se encuentran y, en definitiva, ayudar a reparar esa carencia que ellos poseen.

Los pacientes en los que actualmente se centra esta investigación son aquellos con enfermedades como la retinitis pigmentosa, coroideremia o degeneración macular. En el caso de la retinitis pigmentosa, los fotorreceptores formados encargados de recoger la información luminosa mueren impidiendo que esta información pase al

nervio óptico y por ende al cerebro [3]. En la Figura 1.2. se puede observar la estructura de la retina:

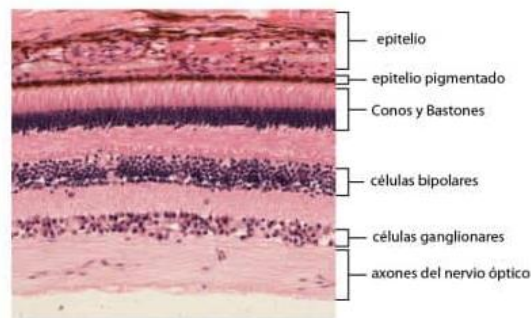


Figura 1.2. Estructura de la retina [4]

Las prótesis tratan, por medio de estimulaciones eléctricas de la corteza cerebral u otras partes de la vía visual, de generar puntos de luz llamados fosfenos [5]. Éstas prótesis, consisten en implantes que por medio de electrodos aplican un estimulación eléctrica en la retina, el nervio óptico o la corteza cerebral y generan un mapa de fosfenos similar a una imagen de puntos de baja resolución como se muestra en la Figura 1.1. Para lograr este efecto, pueden ser colocados en la retina, el nervio óptico o en la corteza cerebral [6]. Por desgracia, la resolución de la rejilla de fosfenos producidos se ve limitada por la biología, la tecnología y la seguridad del propio paciente [7]. Los dispositivos actuales proporcionan entre 90 y 1.000 fosfenos.

Dado que actualmente el campo de visión que poseen no es superior a 20°, resulta un reto interesante el crear un simulador de estas prótesis con el cual se pueda girar la cabeza en todas direcciones e implementar diferentes metodologías que ayuden a mejorar la calidad de imagen para experimentos futuros.

## 1.2. Estado del Arte

Actualmente el número de pacientes operados con prótesis es muy bajo, al ser un número reducido no resulta viable experimentar con ellos. Por ello los SPV son la técnica más prometedora para evaluar la forma en que se comunica la información visual a los pacientes.

Se han llevado a cabo varios experimentos hasta la fecha variando el campo de visión. Por ejemplo Cha en [8] encontró que con un campo de visión de 30° e introduciendo una serie de píxeles de  $25 \times 25$  distribuidos dentro del área visual foveal

podría proporcionar una movilidad útil en entornos que no requieren un alto grado de reconocimiento de patrones.

Por otro lado, Sommerhalder en [33] observó que los usuarios preferían un campo visual de  $33^{\circ} \times 23^{\circ}$  con una imagen pixelada de  $7 \times 10$ . Dagnelie en [9] exploró los requisitos mínimos de resolución visual de un conjunto de electrodos simulados para la movilidad en entornos reales y virtuales. Los sujetos requirieron  $6 \times 10$  puntos con un campo de visión de  $27^{\circ} \times 16.2^{\circ}$  para un desempeño adecuado. Zhao en [10] simuló un entorno visual más complejo con personas de visión normal como sujetos de prueba y les encomendó tareas de identificación y búsqueda de caminos con  $50^{\circ}$  de campo de visión.

### *1.3. Objetivos y alcance*

El objetivo principal de este trabajo es desarrollar un simulador del entorno con visión fosfénica, a partir de imágenes panorámicas, mediante las gafas Oculus Rift DK2 de realidad virtual. Se utilizará el modelo de cámara estenopeica (pin-hole) y se pretende analizar el impacto del campo de vista con representaciones de baja resolución para interpretar el entorno usando visión por computador.

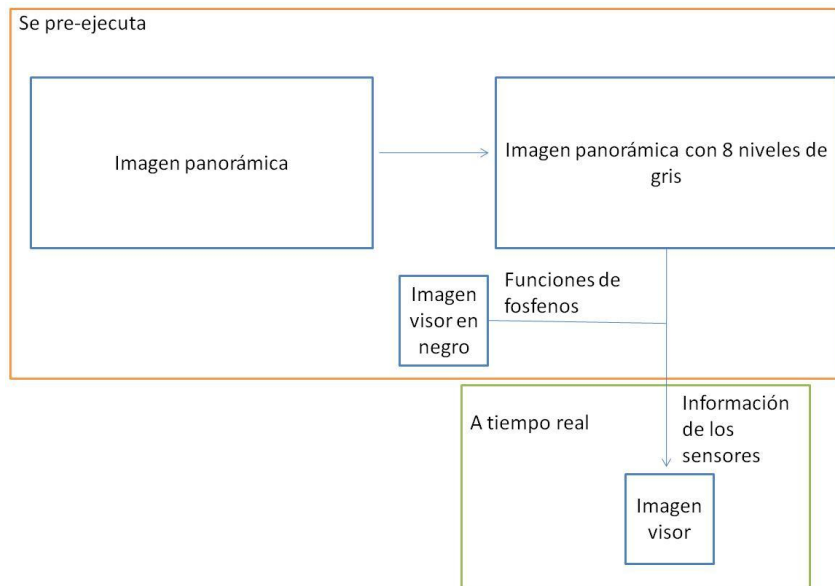
Los objetivos y el alcance estaban bien acotados desde un inicio puesto que el departamento de Informática e Ingeniería de Sistemas había planteado realizar el proyecto anteriormente.

Se escogen imágenes panorámicas para poder representar un entorno de  $360^{\circ}$  de tal forma que se pueda girar la cabeza en todas direcciones situándose desde el centro de la habitación. Este simulador no realiza traslaciones alrededor del entorno si no que únicamente contempla rotaciones.

Una visión general del método se muestra en la Figura 1.3. En primer lugar, se reduce la resolución de color a 8 niveles de gris a la imagen panorámica y se calcula el tamaño, posición y forma de los fosfenos en la imagen del visor. A partir de la imagen en escala de grises y de la información recogida de los sensores de las gafas Oculus se genera la matriz de rotación correspondiente que permite escoger el área de la imagen panorámica que se está observando en ese instante y generar una imagen de fosfenos a la que se le reduce el campo de visión a  $20^{\circ}$  y que finalmente se muestra por pantalla.

Todo esto sucede a tiempo real, de tal forma que es un simulador que permite una experiencia inmersiva sin distracciones del entorno real.

Al precalcular la posición de los fosfenos y su tamaño una única vez al iniciar el programa permite aligerar el funcionamiento de éste y que el funcionamiento a tiempo real sea fluido.



**Figura 1.3. Esquema general del código**

La razón por la cual se escoge realizar este proyecto es que a pesar de existir otros simuladores de prótesis, ninguno que usase gafas de realidad virtual permitía introducir un entorno virtual de 360° y que a la vez les permitiese sumergirse en la experiencia y realizar experimentos en diferentes entornos sin necesidad de desplazarse. En este trabajo no se van a realizar dichos experimentos si no que ha sido creado con la idea de poder ser usado en un futuro y poder comparar dos técnicas diferentes, el Downsampling sobre una imagen panorámica o bien realizar inicialmente una extracción de objetos mediante una red neuronal y combinarlo con el layout de la imagen y, finalmente, realizar el Downsampling.

#### **1.4. Entorno de trabajo**

Las gafas de realidad virtual Oculus Rift DK2 han mantenido un papel fundamental en este proyecto. Como bien explica Manuel Guerrero en [11], éstas gafas pertenecen a Oculus VR y se trata del modelo Rift Development Kit 2 [12]. Es la segunda versión de un casco de realidad virtual pensado para la venta a desarrolladores.

Fue lanzado al mercado en el 2014 poco después de ser retirada la primera versión. Se trata de un aparato maduro, con soporte técnico para desarrolladores y con un kit de desarrollo que facilita su programación. Cabe destacar que ya no se encuentra a la venta, el modelo más actual de ésta empresa es el Rift S [13]. Al tener disponible el sistema de software de Oculus se ha decidido trabajar con este modelo.

El kit incluye las propias gafas y una cámara externa para medir la traslación que en este proyecto no será necesaria dado que el simulador sólo contempla rotaciones con el usuario situado en el centro del entorno virtual.

Las gafas están formadas, por la pantalla del visor por la que se muestra el mundo virtual, dos lentes, una para cada ojo, para conseguir un buen enfoque, el hardware de procesamiento, los leds de iluminación, una unidad de medición inercial (inertial measurement unit, IMU) y la carcasa que sujeta todo el conjunto. Una vista generalizada de su constitución puede observarse en la Figura 1.4.



**Figura 1.4. Vista expandida de las gafas Oculus Rift DK2 [14]**

La pantalla proviene de Samsung, y de hecho emplea la misma pantalla que monta el modelo Galaxy Note 3 [15]. Es una pantalla de 5,7” con una resolución de 1920 píxeles horizontales por 1080 verticales [16]. Como la imagen mostrada por pantalla está duplicada ya que muestra la imagen de salida para cada ojo, la imagen de salida debe ser 1080x960 píxeles.

Como la pantalla se encuentra muy cerca del ojo, se necesita utilizar un sistema de lentes que permita un enfoque correcto a la vez que muestre un campo visual elevado. Las lentes usadas son las de tipo esférico para evitar errores ópticos y reducir el número de lentes necesarias. Estas lentes mas una distorsión de tipo barrilete (Figura

1.5b) introducida por el SDK permiten corregir las anomalías. El conjunto de las lentes y el globo ocular permite corregir la distorsión introducida (Figura 1.6.).



(a) Imagen sin distorsión

(b) imagen con distorsión enviada a las Oculus

Figura 1.5. Distorsión de tipo barrilete

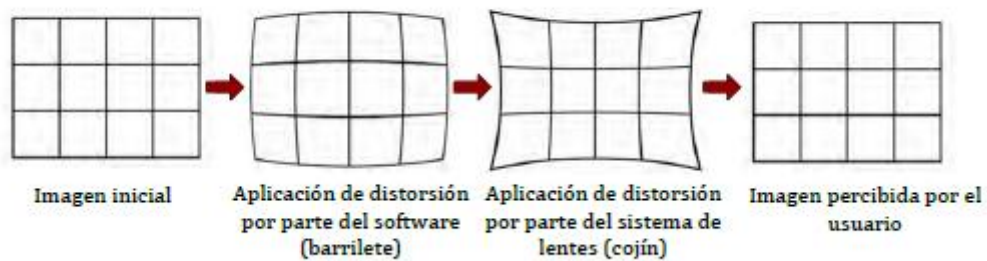


Figura 1.6. Corrección de la distorsión de tipo barrilete [11]

Además de las gafas, el programa usado para crear la aplicación ha sido el Visual Studio 2013 en lenguaje C++ y se han usado las librerías OpenCV para facilitar el manejo de imágenes. Desde su aparición en el año 2000 se han convertido en el estándar para las aplicaciones de visión por computador [17].





# Capítulo 2

## Modelo de cámara

---

En este proyecto se han escogido las imágenes panorámicas para representar un entorno virtual de 360° que el usuario pueda explorar completamente moviendo la cabeza. Una sola de estas imágenes contiene toda la información visual necesaria del entorno del usuario siempre y cuando no se traslade. Cabe destacar que no se visualizarán completamente a través de las gafas sino que según la rotación de la cabeza y del campo de visión del visor se determinará el área observada.

Esto es necesario porque existen diferencias entre la imagen panorámica y nuestra visión.

Se han usado dos tipos de proyecciones diferentes para lograr este efecto, la proyección esférica que modela la proyección en la imagen panorámica y la proyección perspectiva que modela la proyección en el visor de las gafas. Es necesario por lo tanto explicar a continuación los dos modelos de proyección usados en este proyecto. En el siguiente capítulo se explicará cómo se interrelacionan ambas representaciones y su introducción al código.

### *2.1. Modelo de proyección perspectiva*

Se usa como modelo de cámara la estenopeica o más comúnmente conocida como pinhole que corresponde con el modo de funcionamiento de las primeras cámaras desarrolladas. Estas consistían en una caja con un pequeño agujero de tal forma que de

entre todos los rayos luminosos reflejados por un punto del objeto, solo uno alcanza la película fotográfica [19].

Por lo tanto este modelo supone que por cada punto solo un rayo luminoso es emitido, el cual es el que une el punto focal y el punto del objeto. El sensor de la cámara se encontrará a una distancia llamada distancia focal ( $f$ ) y el punto de corte del rayo en el plano de la imagen dará lugar a la proyección del punto del objeto en la imagen tal y como se ve en la Figura 2.1:

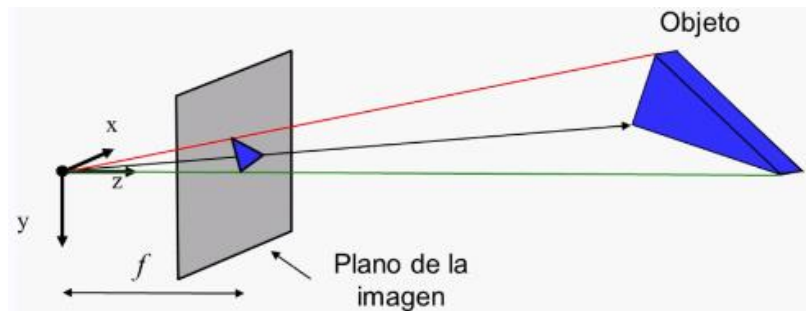


Figura 2.1. Modelo de proyección de la cámara pin-hole [19]

Y mediante triángulos semejantes (Figura 2.2.) se pueden obtener las relaciones matemáticas siguientes:

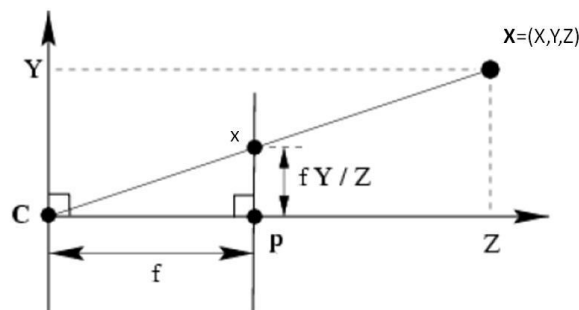


Figura 2.2. Transformación proyectiva de 3D a 2D [20]

$$x = \frac{fX}{Z} \quad (2.1)$$

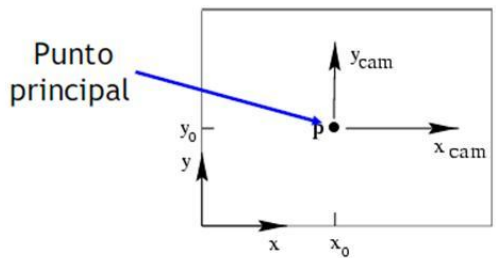
$$y = \frac{fY}{Z} \quad (2.2)$$

En forma matricial quedaría como en la Figura 2.3:

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \sim \begin{pmatrix} fX \\ fY \\ Z \end{pmatrix} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

Figura 2.3. Transformación matemática de 3D a 2D [20]

Como además se desea que el origen de coordenadas de la imagen esté situado en el centro de ésta, la expresión final sería la que se muestra en la Figura 2.4:

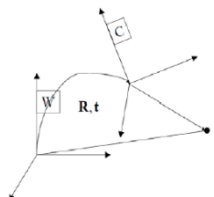


$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \sim \begin{pmatrix} fX \\ fY \\ Z \end{pmatrix} = \underbrace{\begin{bmatrix} f & 0 & x_0 \\ 0 & f & y_0 \\ 0 & 0 & 1 \end{bmatrix}}_K \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

Figura 2.4. Centro óptico de la imagen [20]

A esta matriz K se le llamará matriz de calibración.

Una vez modelada la proyección 3D a 2D surge otro problema que ha de ser resuelto, la concordancia entre sistemas de referencias. Generalmente los puntos 3D observados en el espacio se expresan en un sistema de referencia absoluto distinto del que utiliza la cámara (Figura 2.5a). Por eso se deben realizar transformaciones geométricas para expresar la transformación relativa del sistema de referencia de la cámara con respecto al sistema de referencia absoluto, por lo que la expresión final será la de la Figura 2.5b:



$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \sim \begin{bmatrix} \alpha_x & s & x_0 \\ 0 & \alpha_y & y_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{R}_{CW} & \mathbf{t}_{CW} \\ 0 & 1 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

(a) Representación geométrica

(b) Representación matemática

Figura 2.5. Transformación geométrica de la referencia cámara a la referencia absoluta [20]

## 2.2. Modelo de proyección esférica

Todo el código creado para este simulador de prótesis visuales se basa en la simulación del entorno 3D a través de la introducción de la imagen panorámica del entorno deseado. Una imagen panorámica que se obtiene gracias a la proyección esférica, por ello es necesario explicarla ya que tiene un papel muy importante dentro del código.

Este tipo de proyección, por lo tanto, permite proyectar puntos 3D en una imagen 2D. Como consecuencia de esto, líneas que deberían ser rectas en una habitación se deforman hasta aparecer curvas o (Figura 2.6a), en el caso de los mapamundi (Figura 2.6b) el polo sur se deforma hasta ocupar toda la parte inferior de la imagen.



(a) Panorámica de una habitación

(b) Mapamundi en representación esférica [18]

Figura 2.6. Deformación en imágenes panorámicas

Antes de comenzar, es necesario definir ciertas variables, como muy bien explica [18]:

- Aquellas variables definidas en mayúsculas se refieren al entorno 3D, por lo tanto  $P(X,Y,Z)$  será un punto cualquiera del espacio definido mediante sus coordenadas espaciales
- La resolución de la imagen se define como  $W \times H$  píxeles siendo  $W$  el número de píxeles que tiene por anchura y  $H$  el número que tiene por altura. En este tipo de imágenes se representa en horizontal los  $360^\circ$  y en vertical  $180^\circ$  por lo que se cumple siempre que  $W=2H$ .

- Una vez proyectado el punto 3D, se puede describir con tres magnitudes, la distancia en píxeles sobre la imagen ( $x_s, y_s$ ), la colatitud ( $\theta$ ), el cual tiene un rango entre  $-90^\circ$  y  $+90^\circ$ , y el azimut ( $\Phi$ ), con un rango entre  $-180^\circ$  y  $+180^\circ$ .
- Las coordenadas del centro de la imagen ( $x_o, y_o$ ) se corresponden con ( $W/2, H/2$ ).

De una forma más representativa se puede observar a continuación estas variables en la Figura 2.7:

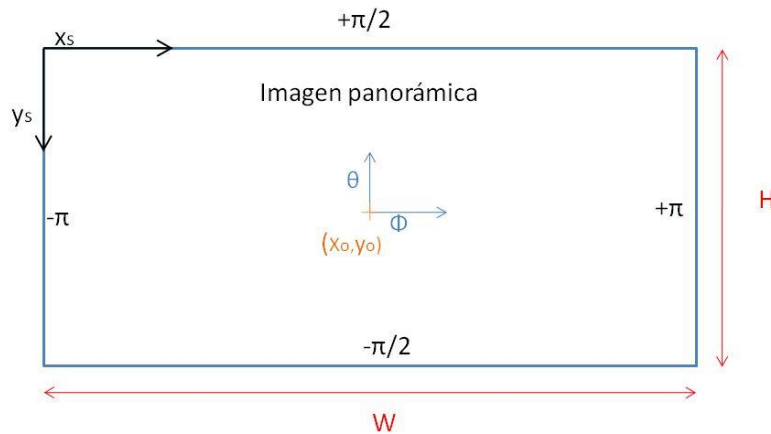


Figura 2.7. Variables de la proyección esférica

Partiendo de un punto en el espacio  $P(X, Y, Z)$ , se proyecta sobre la esfera unidad  $p(x, y, z)$  (Figura 2.8.)

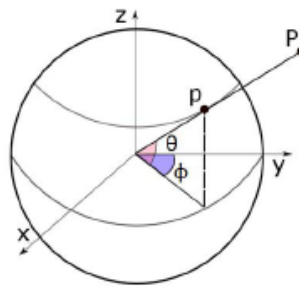


Figura 2.8. Esfera unidad [18]

De tal forma que para pasar a coordenadas esféricas:

$$X = \sqrt{X^2 + Y^2 + Z^2} * \cos \theta * \cos \Phi \quad (2.3)$$

$$Y = \sqrt{X^2 + Y^2 + Z^2} * \cos \theta * \sin \Phi \quad (2.4)$$

$$Z = \sqrt{X^2 + Y^2 + Z^2} * \sin \theta \quad (2.5)$$

Por lo tanto se despejan los ángulos del azimut y de la colatitud:

$$\Phi = \text{atan2}(Y, X) \quad (2.6)$$

$$\theta = \sin^{-1}\left(\frac{Z}{\sqrt{X^2+Y^2+Z^2}}\right) \quad (2.7)$$

Y las coordenadas de la imagen se obtienen a partir de la Figura 2.8:

$$x_s = x_o + \frac{\Phi}{\pi} * \frac{W}{2} \quad (2.8)$$

$$y_s = y_o - \frac{\theta}{\pi} * H \quad (2.9)$$

Este cambio de coordenadas es muy importante para conseguir relacionar la imagen panorámica introducida y la imagen visualizada finalmente, aunque esto se explicará en el Capítulo 3 en más detalle.

## Capítulo 3

# Representación panorámica en gafas de realidad virtual

---

Si no existe traslación toda la información visual del entorno queda codificada en una imagen esférica. Esto significa que si una persona rota la cabeza sin desplazarse y se conoce esta rotación, se puede simular virtualmente la imagen perspectiva que recibe a partir de un panorama esférico. En el caso de las gafas de realidad virtual Oculus esta rotación se puede estimar utilizando un sistema de medición inercial (IMU) embarcado en el visor. En este capítulo se explica la relación geométrica entre un sistema de proyección esférico como el introducido en 2.2 situado en la referencia absoluta y un sistema de proyección perspectivo (2.1) en el sistema de referencia del visor de realidad virtual.

Antes de explicar cómo se han combinado estos dos modelos de proyección, es necesario definir qué es una imagen digital. La imagen es una matriz de tamaño  $m \times n$ , donde  $m$  es el número de píxeles que posee de alto y  $n$  el que posee de ancho. Cada elemento de esta matriz almacena tres valores correspondientes a los colores primarios de la luz (rojo, verde y azul).

Por otro lado, se debe explicar cómo se genera la matriz de rotación que se va a usar a continuación. La IMU de las gafas recoge la información del movimiento de cabeza realizado por el usuario y la almacenan en cuaternios. A través de las ecuaciones de la Figura 3.1. se convierten los cuaternios en una matriz de rotación. Cabe destacar, tal y como se explica en el Anexo I, que esta matriz de rotación no es la que realmente se usa ya que uno de los sensores recoge la información de giro al revés y por eso es

necesario obtener sus ángulos de roll-pitch-yaw, cambiar uno de ellos de signo y componer la nueva matriz de rotación.

```

Mat quaternionToMatrix(glm::quat q)
{
    Mat R = Mat_<double>(3, 3);

    R.at<double>(0, 0) = 1 - 2 * (q.y*q.y + q.z*q.z);
    R.at<double>(1, 0) = 2 * (q.x*q.y + q.z*q.w);
    R.at<double>(2, 0) = 2 * (q.x*q.z - q.y*q.w);

    R.at<double>(0, 1) = 2 * (q.x*q.y - q.z*q.w);
    R.at<double>(1, 1) = 1 - 2 * (q.x*q.x + q.z*q.z);
    R.at<double>(2, 1) = 2 * (q.y*q.z + q.x*q.w);

    R.at<double>(0, 2) = 2 * (q.x*q.z + q.y*q.w);
    R.at<double>(1, 2) = 2 * (q.y*q.z - q.x*q.w);
    R.at<double>(2, 2) = 1 - 2 * (q.x*q.x + q.y*q.y);

    return(R);
}

```

Figura 3.1. Ecuaciones de paso de cuaternios a matriz de rotación

A continuación se procede a explicar cómo se han implementado la generación de la imagen perspectiva del visor a partir de la imagen panorámica y la rotación que nos proporciona la IMU. Inicialmente se calcula para cada punto de la imagen del visor su rayo correspondiente, esto es, su vector dirección. Para que se entienda de una forma más fácil se va a escoger un punto genérico de la imagen del visor de coordenadas (i,j) como muestra la Figura 3.2:

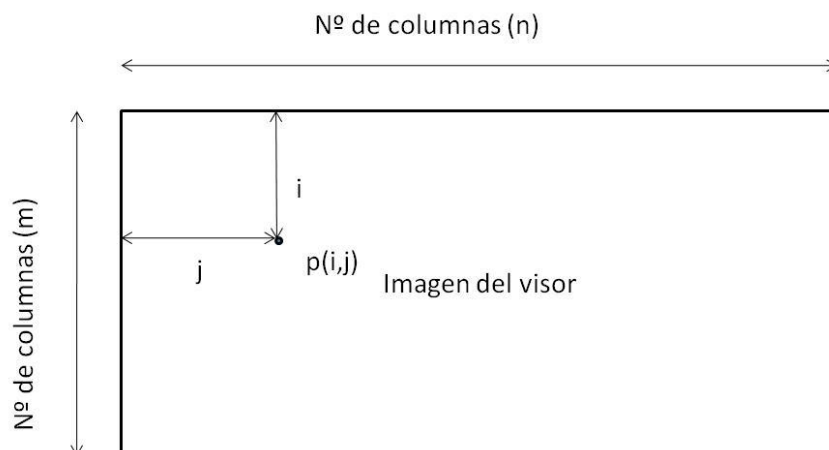


Figura 3.2. Representación de la imagen como una matriz

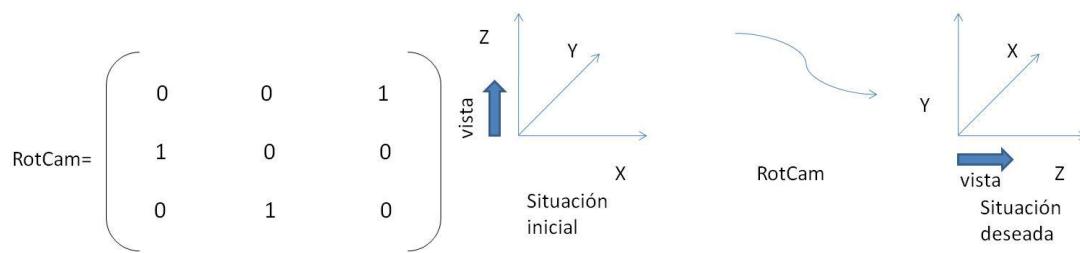
El punto genérico (i,j) se premultiplica por las inversas de la matriz K y la matriz de rotación desarrolladas en el apartado 2.1. Figura 2.5. De ésta forma se realiza el proceso inverso al explicado en el apartado 2.1. ya que se desea obtener el rayo



correspondiente a un punto 2D de la imagen con respecto al sistema de referencia absoluto. El rayo obtenido se llamará  $v$  como muestra la ecuación 3.1:

$$v = \text{Rot}^{-1} * K^{-1} * \begin{pmatrix} j \\ i \\ 1 \end{pmatrix} \quad (3.1)$$

El sistema de referencia absoluto no es conocido, y si se trabajase con el rayo  $v$ , al iniciar el programa el visor mostraría el techo de la habitación cuando el usuario mirase hacia el frente. Por eso ha sido necesario cambiar el sistema de referencia absoluto por otro que cumpla con los requisitos necesarios mediante la premultiplicación del rayo  $v$  por la matriz  $\text{RotCam}$ . Ésta matriz se representa en la Figura 3.3:



(a) Representación matemática de  $\text{RotCam}$

(b) Representación geométrica de  $\text{RotCam}$

Figura 3.3. Matriz de cambio de referencia del entorno simulado

Por lo que finalmente se obtiene el rayo  $v_s$  con respecto al nuevo sistema de referencia definido. La expresión resultante se observa en la ecuación 3.2:

$$v_s = \text{RotCam} * \text{Rot}^{-1} * K^{-1} * \begin{pmatrix} j \\ i \\ 1 \end{pmatrix} \quad (3.2)$$

Este rayo corresponde al nuevo punto 3D de la proyección esférica, por lo que el procedimiento a seguir una vez calculado  $v_s$  es el mismo que el explicado en el apartado 2.2. en el que primero han de calcularse los ángulos de azimut y colatitud de las ecuaciones 2.6 y 2.7 teniendo en cuenta que las componentes de  $v_s$  son nombradas como en la igualdad 3.3:

$$V_s = \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \quad (3.3)$$

Y las nuevas coordenadas del punto en la imagen panorámica correspondiente al punto genérico según las ecuaciones 2.8 y 2.9. Se realiza la correspondencia geométrica según la Figura 3.4. en la que  $w$  correspondería con  $x_s$  y  $k$  con  $y_s$ , coordenadas de la imagen panorámica introducida, siendo  $(i,j)$  las coordenadas de la imagen del visor como se observa en la Figura 3.2.

```

if ((k >= 0) && (k < src.rows) && (w >= 0) && (w < src.cols) && (i >= 0) && (i < dst.rows) && (j < dst.cols) && (j >= 0)) {
    dst.at<Vec3b>(i, j)[0] = src.at<Vec3b>(k, w)[0];
    dst.at<Vec3b>(i, j)[1] = src.at<Vec3b>(k, w)[1];
    dst.at<Vec3b>(i, j)[2] = src.at<Vec3b>(k, w)[2];
}

```

**Figura 3.4. Correspondencia entre imagen del visor e imagen panorámica**

La imagen destino, o imagen del visor, es  $dst$  y la imagen panorámica, o imagen fuente, es  $src$ . Se realizan tres correspondencias ya que como se ha comentado anteriormente, un mismo punto de la imagen alberga tres valores correspondientes a los colores primarios.

El resultado final se puede observar en la Figura 3.5b, en la imagen del visor final ya no aparecen líneas curvas cuando debieran ser rectas.



**(a) Imagen panorámica**

**(b) Resultado del uso de ambas proyecciones**

**Figura 3.5. Comparativa de imagen de partida e imagen final**

Como este proyecto tiene por finalidad mostrar por pantalla una imagen de fosfenos en vez de una imagen a color, este proceso cambia ligeramente, esto se explicará en el Capítulo 4.

# Capítulo 4

## Mapa de Fosfenos

---

Las prótesis visuales son la tecnología más prometedora para mejorar la visión en pacientes con enfermedades degenerativas de la visión ya que actualmente no existe tratamiento médico.

Los implantes retinales actuales constan de una micro cámara que captura la información visual y un procesador la convierte en microestímulos en el implante. La imagen resultante de este proceso corresponde a un mapa de fosfenos de muy baja resolución espacial y con un grado dinámico bajo [21].

### *4.1. Tipos de Implantes*

Como muy bien explica Alberto Badías en [22], existen diferentes tipos de prótesis según el problema que cause ceguera.

#### *4.1.1. Implantes retinales*

Estimulan las células de la retina. El conjunto de electrodos pueden ser epiretinales, subretinianos o supracoroideos.

**Epiretinales:** un ejemplo de este tipo es el conocido Argus II perteneciente a la empresa estadounidense Second Sight Medical Products, Inc.. Es uno de los dos modelos de prótesis aprobados por la Unión Europea para el tratamiento de retinitis pigmentosa [23]. Se trata de un implante epirretinal de 60 electrodos con un ángulo de visión máximo de 20° en diagonal como se observa en la Figura 4.1.a. Convierte información visual procedente de una cámara en patrones de estimulación que se

transmiten al implante a tiempo real. La separación entre los centros de los electrodos es de  $525\mu\text{m}$ , que corresponde a unos  $1,88^\circ$  de campo visual, suponiendo que  $1^\circ$  equivale a  $280\mu\text{m}$  de longitud en la retina [24].

**Subretinales:** un ejemplo es el Alpha-IMS de Retina Implant AG. Se trata de un implante subretinal que no posee una cámara como en el anterior caso si no que dispone de 1500 fotodiodos que capturan la luz proyectada sobre la retina y la transforman en señales eléctricas. El implante tiene un tamaño de  $3 \times 3\text{ mm}$  y los electrodos tienen un diámetro de  $50\mu\text{m}$ , con una distancia entre centros de electrodos de  $70\mu\text{m}$  [25].

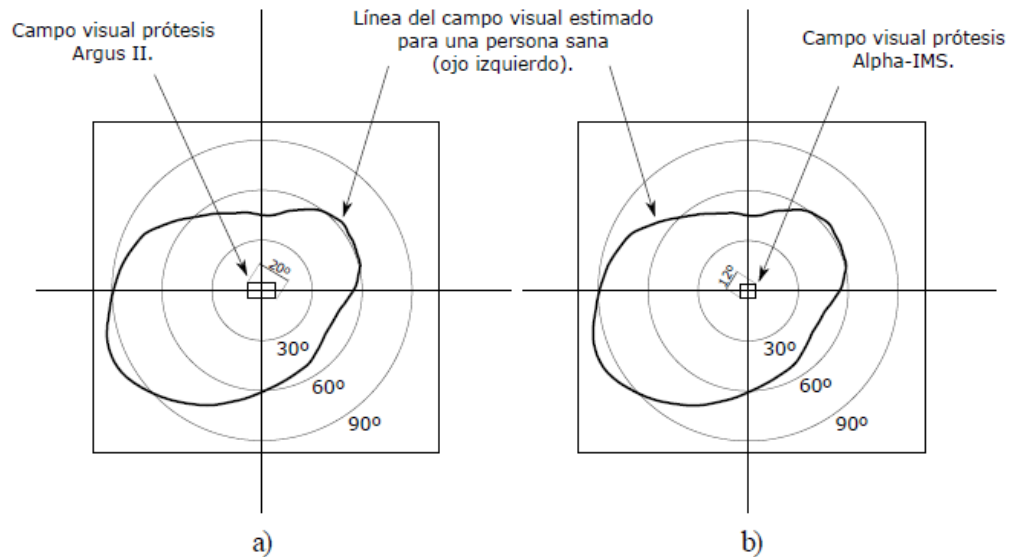


Figura 4.1. Comparación entre los dos modelos de prótesis [22]

#### 4.1.2. Nervio óptico

La primera estimulación eléctrica del nervio óptico fue conseguida por [26]. Se demostró que los fosfenos podían reproducirse con estimulaciones seguras, y predecir la ubicación y el tamaño de las percepciones en función de los parámetros de estímulo [27]. Con entrenamiento, los receptores podrían reconocer y orientar formas complejas [28] y realizar la localización y discriminación de objetos [29].

#### 4.1.3. Cortex Visual

Las primeras estimulaciones eléctricas de fosfenos fueron en ésta área. En 1968, [5] se utilizó una serie de receptores de radio conectados a electrodos en contacto con el polo occipital derecho de un paciente de 52 años.

Al estimular varios electrodos simultáneamente, el paciente podía ver patrones simples. Y además, durante los movimientos voluntarios de los ojos, los fosfenos se movían de la misma manera que el ojo.

Generalmente los fosfenos desaparecían inmediatamente pero en algún caso permanecían hasta 2 minutos después de un estímulo fuerte.

## 4.2. Representación de Fosfenos

Se ha escogido un campo de visión circular de 20° para presentar un simulador más realista como se observa en la Figura 4.2:

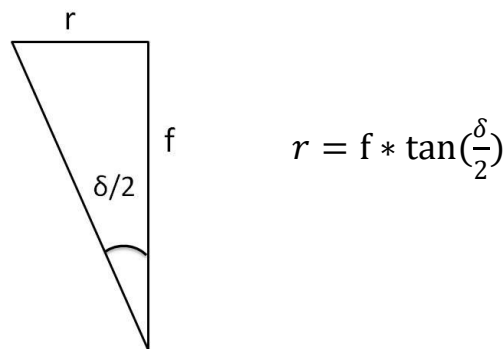


Figura 4.2. Cálculo del radio de visión con un ángulo de 20°

A continuación se explica la estrategia seguida para la representación del mapa de fosfenos. Contrariamente a lo que se puede llegar a pensar, se parte de la imagen final que será mostrada por el visor, no de la panorámica introducida. A partir del tamaño de la imagen y del número de fosfenos que queremos representar se calcula el tamaño de éstos a través de la función *computeDeltaFromNumberOfPhosphenes*.

Una vez que se conoce el tamaño de los fosfenos, se calcula su posición a través de la función *genPhosphenesPos* a partir del tamaño de la imagen, el radio de visualización, el tamaño de los fosfenos, la matriz de calibración y el punto central de la imagen. Esta función nos devuelve en una matriz los rayos asociados a cada fosfeno.

Ahora que ya se sabe la posición y el tamaño de los fosfenos es hora de generarlos, esto se consigue con la función *genPhoshenSprite*. Para representarlos se ha optado por fosfenos circulares cuya iluminación siga una distribución gaussiana de manera que el centro obtiene más intensidad que la parte exterior de acuerdo con estudios de otros SPV [30].

Estas funciones se ejecutan una única vez al inicio del programa ya que el tamaño y posición de los fosfenos así como su forma icónica no varía a lo largo de la ejecución del programa.

La implementación del código en tiempo real cambia ligeramente con respecto al Capítulo 3. Al haber precalculado los rayos correspondientes a cada fosfeno con la función *genPhospenesPos* ya no es necesario recorrer la imagen del visor pixel a pixel. En su lugar se recorre la matriz de rayos con un tamaño mucho menor ya que corresponde con el número de fosfenos (d). El vector  $v_s$  ya no se obtiene según la ecuación 3.2. sino que al ser d un rayo ya no se proyecta en una imagen perspectiva mediante la matriz K y sigue la ecuación 4.1:

$$v_s = \text{RotCam} * \text{Rot} * d \quad (4.1)$$

Se continúa el procedimiento del Capítulo 3 desde el cálculo de las coordenadas esféricas del rayo en la referencia absoluta hasta la asignación del valor de color de los píxeles. El trabajar con fosfenos ya no se asigna directamente el valor de color de la imagen panorámica como en la Figura 3.4. La imagen panorámica ha sido inicialmente convertida a escala de gris con 8 niveles de grises, que es un rango dinámico aceptable para una prótesis actual, por lo que ya no almacena 3 valores para cada posición de la matriz si no que solo almacena un único valor, el de gris. El valor del píxel en  $(x_s, y_s)$  se almacena en una variable denominada *PhospeneFlag*. Ésta variable es introducida en la función *visualize* de tal manera que el fosfeno anteriormente creado se multiplica por un factor de corrección dependiendo del valor de *PhospeneFlag* para ese fosfeno. Con esto se consiguen diferentes intensidades para cada fosfeno dependiendo de la luminosidad del punto correspondiente en la imagen panorámica.

Aberración del color blanco en representación monocromática con lente esférica:  
En este trabajo las imágenes mostradas por pantalla son mapas de fosfenos, no tienen color pero al introducir la distorsión de tipo barrilete explicada en el apartado 1.4. y colocarnos las gafas aparecen trazas de colores en fosfenos sobre todo de la periferia que no deberían estar (Figura 4.4). Por eso se ha optado por representar los fosfenos en color verde (Figura 4.5.) como en las televisiones más antiguas al ser el color primario de más resolución.

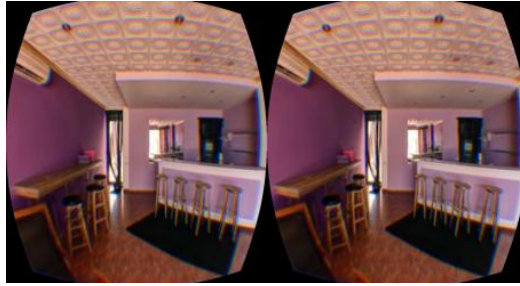
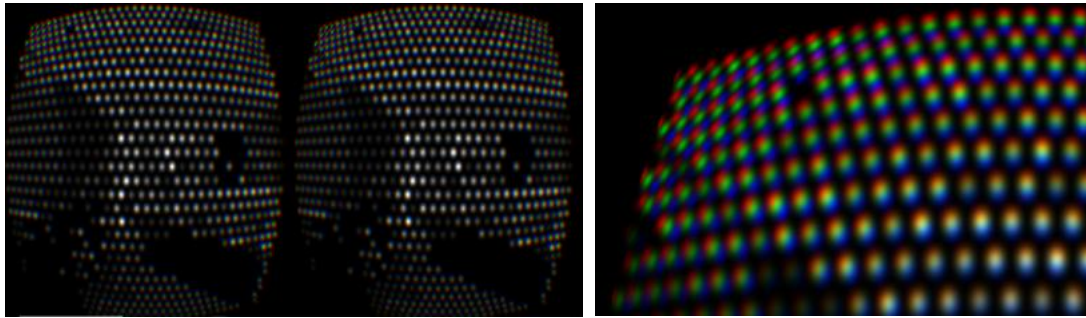


Figura 4.3. Imagen a color representada en las Oculus



(a) Representación con 1.000 fosfenos

(b) Ampliación de la zona donde los colores con más claridad

Figura 4.4. Aparición de colores por la distorsión

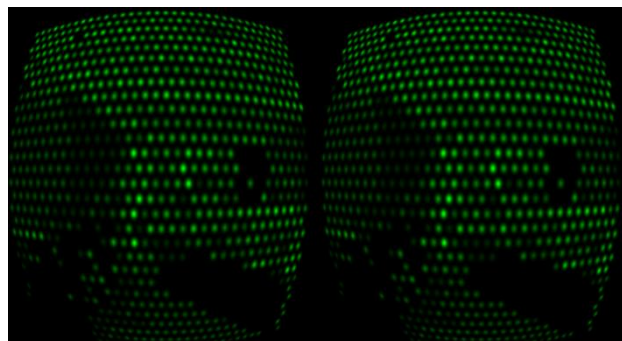


Figura 4.5. Representación de fosfenos en color verde





# Capítulo 5

## Representaciones simuladas

---

Uno de los grandes problemas de las prótesis visuales es la baja resolución y bajo rango dinámico de la imagen percibida que hace que una adaptación directa de la imagen adquirida por la cámara no sea siempre la mejor solución para cualquier tarea. El uso de un simulador de prótesis visual 360° nos permite simular la sensación percibida por un usuario de prótesis visual y experimentar diversas representaciones del entorno así como la influencia de estas representaciones y del campo de visión en el reconocimiento de objetos y lugares cuando se explora el entorno mediante rotaciones de cabeza. Las dos representaciones utilizadas para éste simulador son el método downsampling y el método SIE-OMS. Estas representaciones han sido escogidas para que sean de utilidad en los experimentos futuros.

### *5.1. Método de Downsampling*

Éste método consiste en reducir la resolución de color de la imagen a 8 niveles de gris y reducir su resolución espacial, esto último se consigue por si solo al observar una parte de la imagen por el visor.

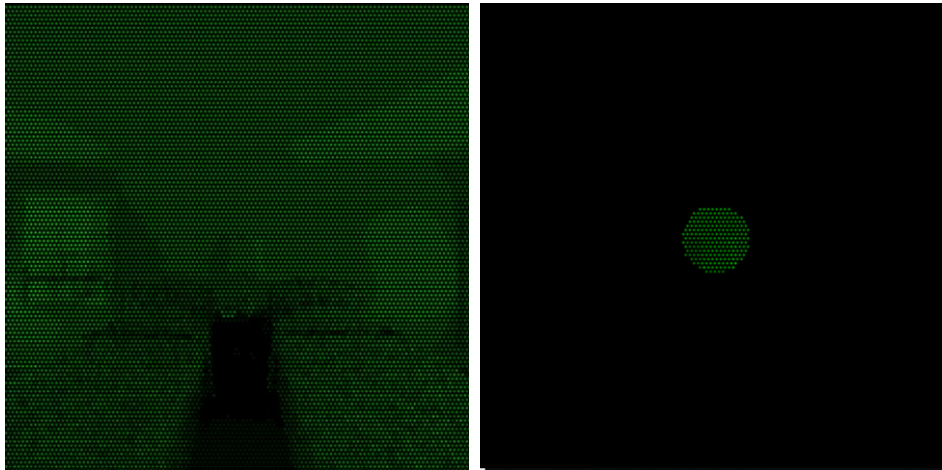
Para la imagen de la Figura 5.1a se obtiene el trozo de imagen que se visualizaría si se mirase hacia el frente (Figura 5.1b) y su representación en las Oculus (Figura 5.1c):



(a) Imagen original de la habitación (b) Lo que se vería si se mirase al frente (c) Representación en las Oculus

Figura 5.1. Ejemplo de habitación

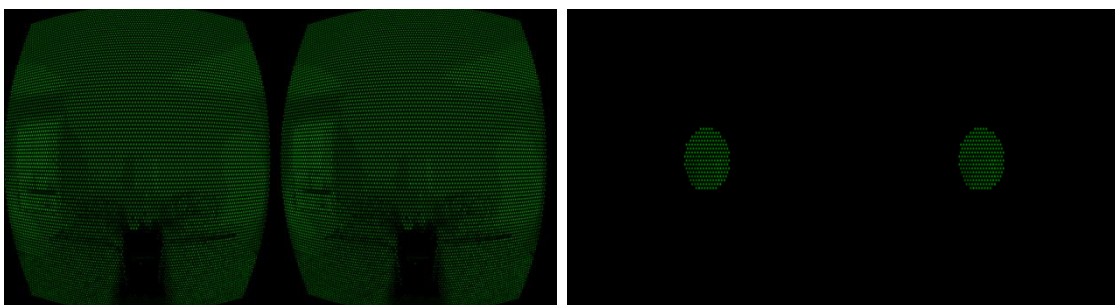
A continuación se muestran los mapas de fosfenos de la habitación anterior para 10.000 fosfenos (Figura 5.2a), y con la misma densidad de fosfenos el resultado de reducir el campo de visión a 20° (Figura 5.2.b) y su representación en las Oculus (Figura 5.3a y Figura 5.3b):



(a) Vista general

(b) Vista con visión reducida a 20°

Figura 5.2. Mapa de fosfenos para 10.000 fosfenos

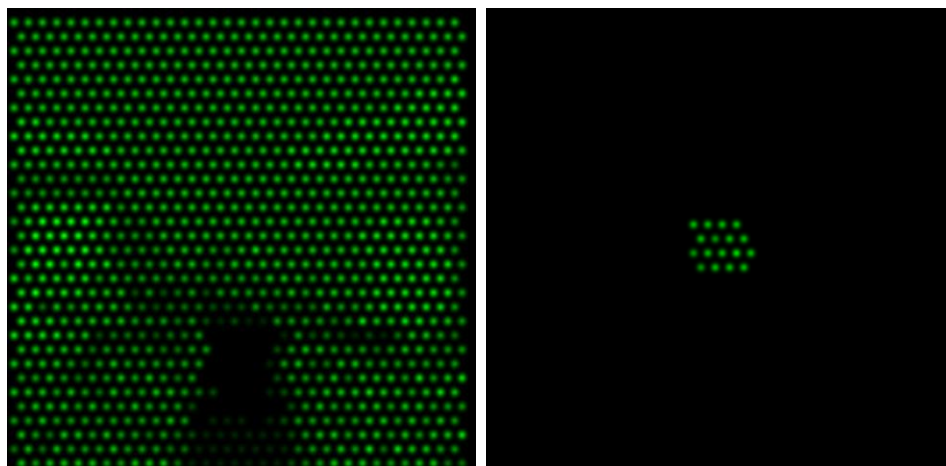


(a) Vista general

(b) Vista con visión reducida a 20°

Figura 5.3. Representación en las Oculus para 10.000 fosfenos

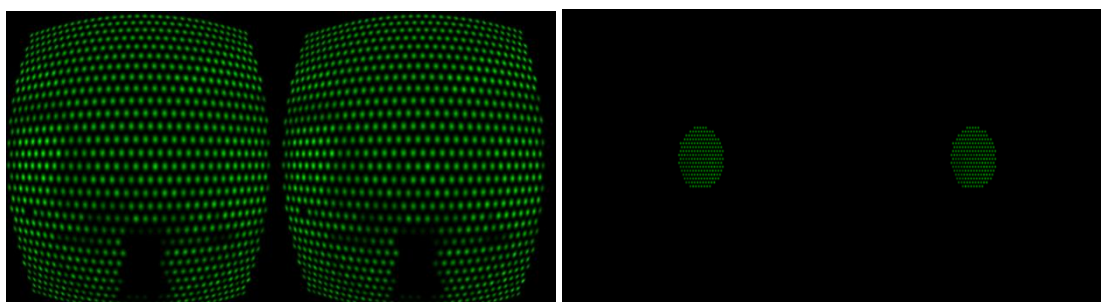
Se puede observar en las Figuras 5.4 y 5.5. cómo cambia drásticamente la capacidad de interpretar el entorno al reducir el número de fosfenos a 1.000.



(a) Vista general

(b) Vista con visión reducida a 20°

Figura 5.4. Mapa de fosfenos para 1.000 fosfenos

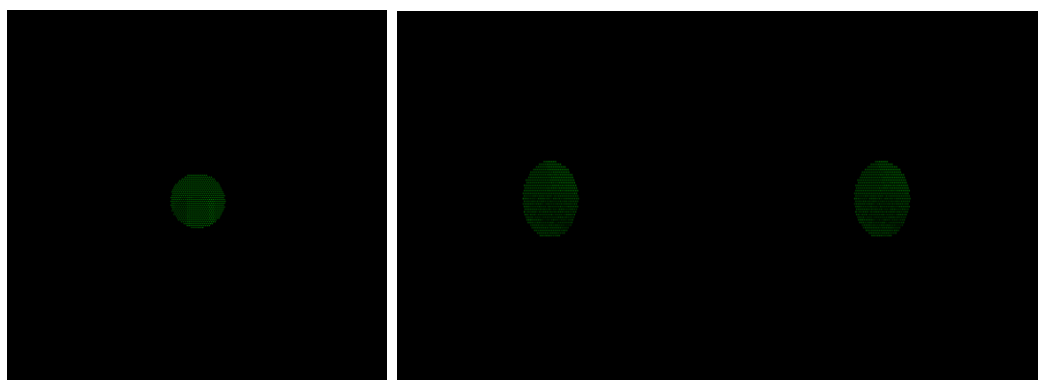


(a) Vista general

(b) Vista con visión reducida a 20°

Figura 5.5. Representación del mapa de fosfenos para 1.000 fosfenos en Oculus

Y por último, en la Figura 5.6. se muestra la situación más cercana a la realidad con un campo de visión de 20° y 679 fosfenos en el interior ya que lo general es que en ese campo reducido los pacientes observen entre 90 y 1.000 fosfenos:



(a) Vista al corregir la distorsión

(b) Representación de las Oculus

Figura 5.6. Vista con visión reducida a 20° para 679 fosfenos

## 5.2. Método de SIE-OMS

En este método la imagen panorámica original es procesada por dos redes neuronales que han sido entrenadas mediante aprendizaje profundo. La primera red extrae información del trazado de planta de una habitación (o layout). La segunda red extrae las mascararas de los objetos más importantes de la escena. Ambas imágenes se combinan en una representación simplificada de una habitación (como la que se observa en la Figura 5.1) que precisa de muy pocos niveles de gris (ver Figura 5.7).

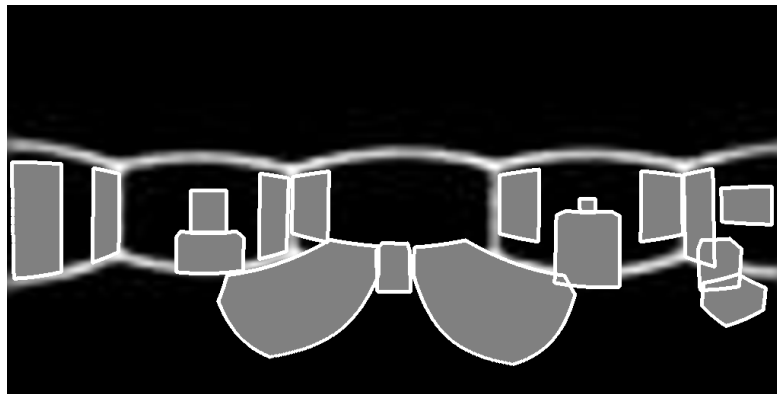
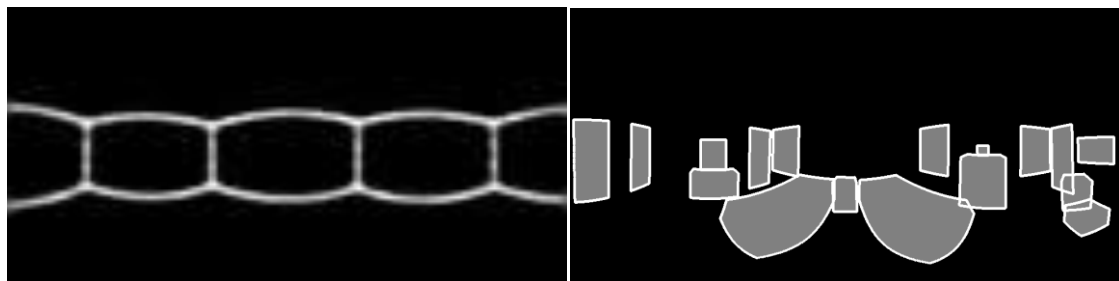


Figura 5.7. Representación del método SIE-OMS

La imagen de la Figura 5.7. es resultado de la unión del layout con el extractor de objetos como se observa en la Figura 5.8:



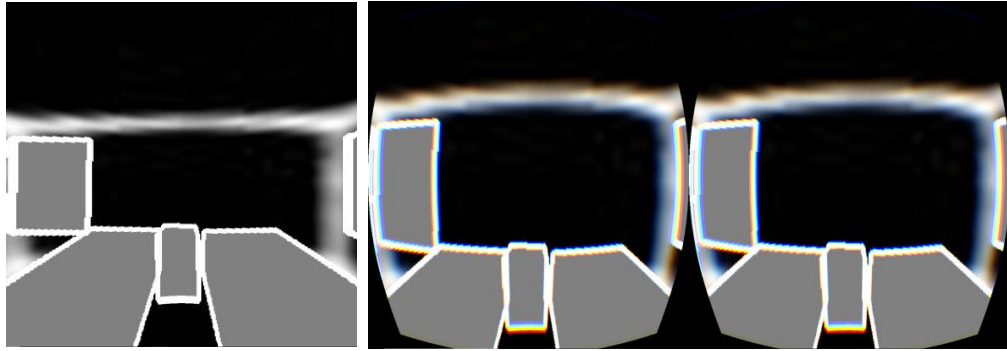
(a) layouts de la habitación

(b) Resultado de la extracción de objetos

Figura 5.8. Como se forma la imagen de SIE-OMS

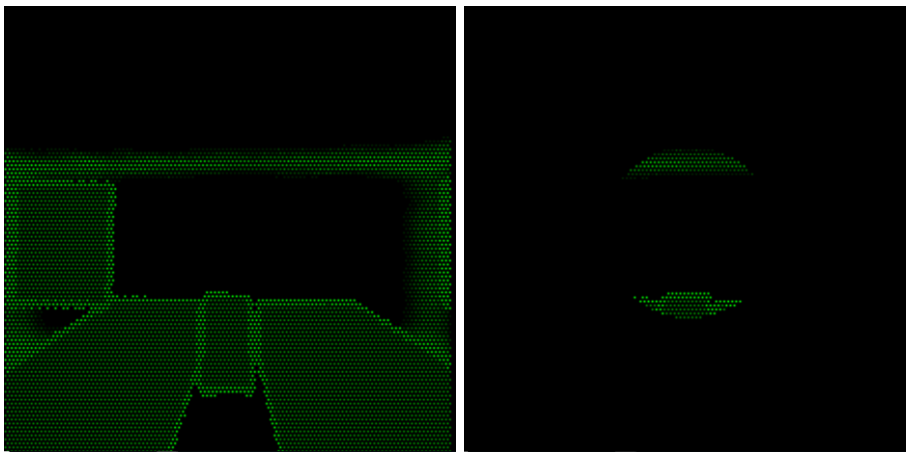
Como en el método anterior, esta imagen debe reducirse a 8 niveles de gris

En la Figura 5.9. se muestra como se vería la imagen anterior al mirar hacia el frente, en la Figura 5.10a su mapa de fosfenos con 10.000 fosfenos y con reducción del campo de vista a 50° (Figura 5.10b) ya que con 20° los fosfenos aparecían apagados al mostrar únicamente la pared, lo mismo ocurre en la Figura 5.11 para el mapa de fosfenos de 1.000 fosfenos:



(a) Como se vería al mirar hacia el frente      (b) Representación en las Oculus

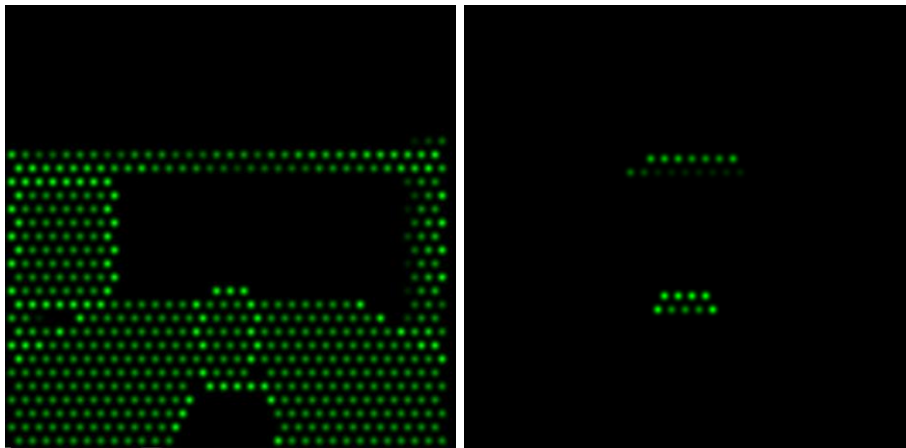
Figura 5.9. Visualización de la imagen de layouts



(a) Vista general

(b) Vista con visión reducida a 50°

Figura 5.10. Mapa de fosfenos para 10.000 fosfenos para imagen de layouts



(a) Vista general

(b) Vista con visión reducida a 50°

Figura 5.11. Mapa de fosfenos para 1.000 fosfenos para imagen de layouts

En la Figuras 5.12 y 5.13 se muestra el resultado en las Oculus:

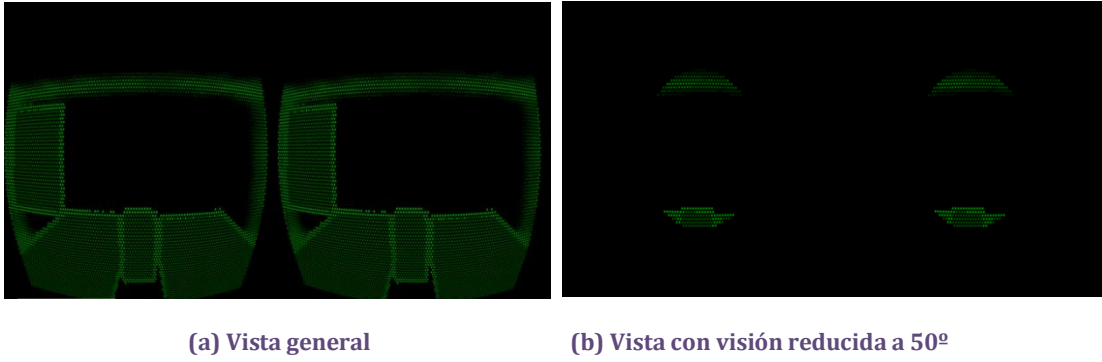


Figura 5.12. Representación del mapa de fosfenos para 10.000 fosfenos para imagen de layouts en Oculus

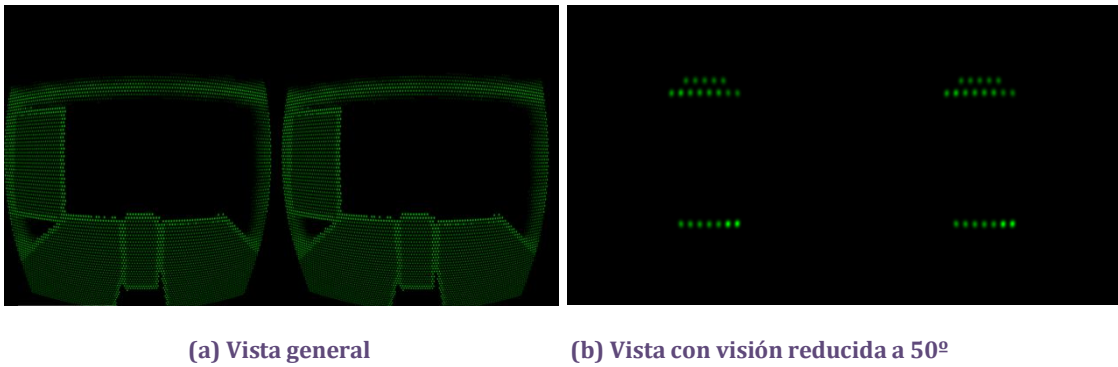


Figura 5.13. Representación del mapa de fosfenos para 1.000 fosfenos para imagen de layouts en Oculus

Y por último, en la Figura 5.14. la representación de 981 fosfenos en un campo de visión de 50°:

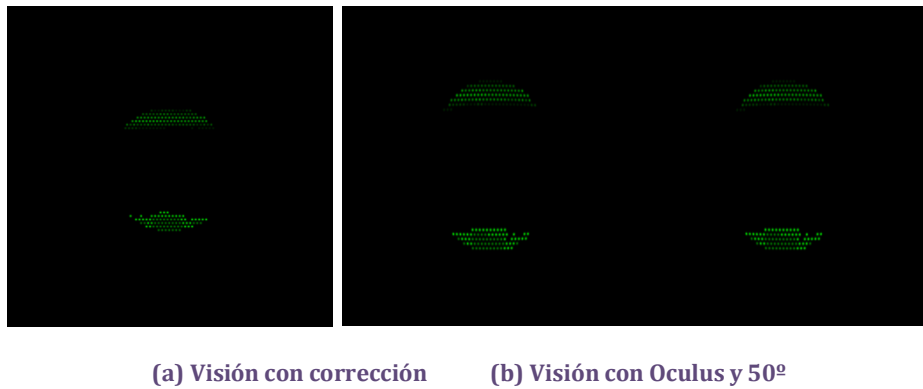


Figura 5.14. Mapa de fosfenos con 981 fosfenos y 50° de visión

# Capítulo 6

## Conclusiones

---

Para concluir el trabajo se van a resumir los diferentes aspectos desarrollados en la memoria. Inicialmente se introduce el estado del arte actual de las prótesis, la motivación que ha dado lugar a este proyecto y los objetivos planteados.

Es necesario introducir las herramientas usadas y familiarizarnos con ellas, una vez concluida la tarea, se introducen los dos tipos de proyecciones usados de manera que se obtiene una mejor comprensión cuando se explica su uso interrelacionado para poder crear el entorno 3D a partir de una imagen panorámica. El siguiente paso ha sido la introducción de la representación mediante mapas de fosfenos y las dos representaciones que se usarán en un futuro para experimentar con personas.

Este trabajo tiene como finalidad el crear un SPV en tiempo real que permita la evaluación de los dos métodos de representación de fosfenos para futuras pruebas con usuarios para poder comparar ambos y evaluar de manera efectiva cual ofrece un mejor entendimiento del entorno. A diferencia de otros simuladores, éste es capaz de representar diferentes entornos virtuales sin necesidad de desplazarse y realizar una exploración mediante rotaciones la cual es una parte fundamental en el proceso de aprendizaje.

Para terminar, añadir que todos los objetivos fijados inicialmente han sido cumplidos y el terminar este trabajo ha sido satisfactorio a pesar de alguna dificultad en el proceso explicada más detalladamente en el Anexo I. La expectativa final es que este

trabajo sirva como apoyo para continuar la investigación y ayude a mejorar la calidad de vida de los pacientes con estas enfermedades visuales.



## Anexo I . Fases del proyecto y duración

En este Anexo se especificará cómo se ha desarrollado el proyecto paso por paso, el tiempo que se ha empleado en cada parte y los problemas presentados.

FASES	TIEMPO	OBSERVACIONES
Curso de OpenCV en C++	Diciembre - Enero	
Lecturas del tema presentado	Enero - Febrero	
Implementación de la proyección perspectiva	1ª mitad Febrero - 2ª mitad Febrero	Complicaciones típicas del aprendizaje de C++
Implementación de la proyección esférica	2ª mitad Febrero - Marzo	Complicaciones típicas del aprendizaje de C++
Transformación de la imagen resultado a una de fosfenos	Abril - Mayo	Adaptación de las funciones de fosfenos para el código
Prueba de diferentes representaciones	Mayo - Junio	

Figura A.1. Cronograma del proyecto

Como se puede observar en la Figura A.1. la primera fase fue aprender a utilizar las bibliotecas OpenCV mediante un cursillo online [31], tras lo cual se procedió a conocer mejor el campo de investigación, los diferentes tipos de prótesis los experimentos con SPV realizados y las diferentes representaciones de mapas de fosfenos en las que se está trabajando.

A continuación se comenzó con la creación del programa implementando la proyección perspectiva hasta conseguir rotar una imagen sobre sí misma en los tres giros posibles (Figura A.2.).



(a) Imagen original

(b) Imagen resultado de aplicar un giro de 80° en el eje perpendicular a la imagen

Figura A.2. Proyección perspectiva aplicada a una imagen

Las complicaciones encontradas fueron a causa del lenguaje C++ ya que al nunca haberlo usado a veces resultaba un poco dificultoso encontrar errores de compilación o maneras de escribir ciertas instrucciones.

Conseguido este paso, se coordinó con la proyección esférica hasta conseguir el entorno virtual 360°. Al comprobar con las gafas se observó que si se miraba hacia el frente, los giros se realizaban de manera correcta mientras que si se giraba 180° todos los giros exceptuando el azimut se realizaban al revés y si solo se giraba +-90° los giros se intercambiaban entre sí. La información de los sensores era devuelta en forma de cuaternios y éstos, a través de una función los transformaba a una matriz de rotación.

Tras comprobar que dicha función era correcta, se modificó la ecuación 3.2. de la siguiente manera:

$$v_s = \text{RotCam} * \text{Rot} * K^{-1} * \begin{pmatrix} j \\ i \\ 1 \end{pmatrix} \quad (\text{A.1})$$

Al no utilizar traslaciones, la matriz inversa de la rotación equivale a la transpuesta. En la ecuación A.1. los giros deberían haber sido todos inversos al giro real pero no fue el caso, los giros de azimut y de elevación del mentón se realizaban al revés mientras que el de rotación seguía el movimiento real de la cabeza. Dado que la causa del problema era que uno de los sensores recogía un giro al revés, al

realizar la inversa de la matriz de rotación provocaba estos resultados tan inesperados. La solución fue calcular los ángulos de roll-pitch-yaw y cambiar de signo el ángulo de roll, tras lo cual se introducían en una nueva matriz de rotación. Recordemos que la situación final del sistema de referencia absoluto es la de la Figura 3.3b y que como se observa en la Figura A.3. el giro en torno al eje z es el de roll.

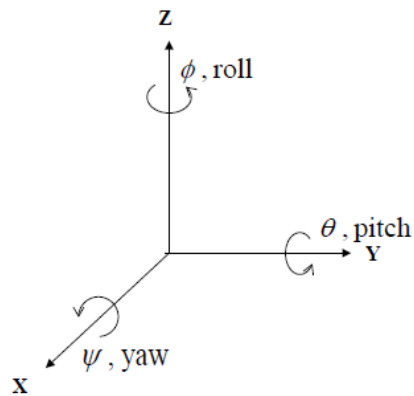


Figura A.3. Ángulos de roll pitch yaw [32]

Tras arreglar este problema y reducir el número de FPS para que fuera más rápido a tiempo real, se introdujeron las funciones de fosfenos y se implementaron como se ha explicado en el Capítulo 4, al ejecutar las funciones una única vez y tener todos los rayos correspondientes a los fosfenos en una única matriz el funcionamiento a tiempo real es más rápido que sin usar fosfenos.

Por último se probaron las dos representaciones de mapas de fosfenos.



## Bibliografía

---

- [1] R. R. A. Bourne *et al.*, “Magnitude, temporal trends, and projections of the global prevalence of blindness and distance and near vision impairment: a systematic review and meta-analysis,” *Lancet Glob. Heal.*, vol. 5, no. 9, pp. e888–e897, Sep. 2017.
- [2] M. Sanchez-Garcia, R. Martinez-Cantin, and J. J. Guerrero, “Indoor Scenes Understanding for Visual Prosthesis with Fully Convolutional Networks,” 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications, 2019
- [3] “Home - Fighting Blindness Canada (FBC).” [Online]. Available: <https://www.fightingblindness.ca/>. [Accessed: 15-May-2019].
- [4] “Conos, bastones y ipRGCs: los fotorreceptores del ojo humano - Curiosoando.” [Online]. Available: <https://curiosoando.com/conos-bastones-y-iprgcs-los-fotorreceptores-del-ojo-humano>. [Accessed: 15-May-2019].
- [5] G. S. Brindley and W. S. Lewin, “The sensations produced by electrical stimulation of the visual cortex,” *J. Physiol.*, vol. 196, no. 2, pp. 479–493, May 1968.
- [6] G. Dagnelie, *Visual prosthetics : physiology, bioengineering and rehabilitation*. Springer, 2011.
- [7] W. Lui, D. Browne, ... L. K.-... C. of the, and undefined 2012, “Transformative reality: improving bionic vision with robotic sensing,” *ieeexplore.ieee.org*.
- [8] K. Cha, K. W. Horch, and R. A. Normann, “Mobility performance with a pixelized vision system,” *Vision Res.*, vol. 32, no. 7, 1992.

- [9] G. Dagnelie, P. Keane, V. Narla, L. Yang, J. Weiland, and M. Humayun, "Real and virtual mobility performance in simulated prosthetic vision," *J. Neural Eng.*, vol. 4, no. 1, Mar. 2007.
- [10] Y. Zhao, X. Geng, Q. Li, G. Jiang, Y. Gu, and X. Lv, "Recognition of a Virtual Scene via Simulated Prosthetic Vision," *Front. Bioeng. Biotechnol.*, vol. 5, Oct. 2017.
- [11] M. Guerrero-Viu, "Detección de personas para simulación de prótesis visual." TFG EINA, Universidad de Zaragoza 2016.
- [12] "Oculus Rift: visor de realidad virtual para ordenadores optimizados para la realidad virtual | Oculus." [Online]. Available: <https://www.oculus.com/rift/#oui-csl-rift-games=robo-recall>. [Accessed: 26-May-2019].
- [13] "Oculus Rift S | Oculus." [Online]. Available: <https://www.oculus.com/rift-s/>. [Accessed: 26-May-2019].
- [14] "Open Source Release of Rift DK2 | Oculus." [Online]. Available: <https://developer.oculus.com/blog/open-source-release-of-rift-dk2/>. [Accessed: 25-May-2019].
- [15] "Oculus Rift Development Kit 2 Teardown - iFixit." [Online]. Available: <https://es.ifixit.com/Desmontaje/Oculus+Rift+Development+Kit+2+Teardown/27613>. [Accessed: 26-May-2019].
- [16] "Samsung Galaxy Note 3 specs | Android Central." [Online]. Available: <https://www.androidcentral.com/samsung-galaxy-note-3-specs>. [Accessed: 26-May-2019].
- [17] "Qué son las OpenCV | Tema 4. La biblioteca OpenCV | Material del curso ISA.1x | edX." [Online]. Available: [https://courses.edx.org/courses/course-v1:UC3Mx+ISA.1x+3T2017/courseware/1b3mlg10f4mmo91brl192j1g65p/1b3mlg10982pit1jtv1olrbpct/1?activate\\_block\\_id=block-v1%3AUC3Mx%2BISA.1x%2B3T2017%2Btype%40vertical%2Bblock%401b3mlg101hrqtfoa2u1te71j7u8](https://courses.edx.org/courses/course-v1:UC3Mx+ISA.1x+3T2017/courseware/1b3mlg10f4mmo91brl192j1g65p/1b3mlg10982pit1jtv1olrbpct/1?activate_block_id=block-v1%3AUC3Mx%2BISA.1x%2B3T2017%2Btype%40vertical%2Bblock%401b3mlg101hrqtfoa2u1te71j7u8). [Accessed: 25-May-2019].

- [18] C. Fernandez-Labrador, J.M. Facil, A., Perez-Yus, C. Demonceaux, J. Civera, & J.J. Guerrero, (2019). Corners for Layout: End-to-End Layout Recovery from 360 Images. arXiv preprint arXiv:1903.08094.
- [19] “Modelado de una óptica | Tema 2. Óptica | Material del curso ISA.1x | edX.” [Online]. Available: [https://courses.edx.org/courses/course-v1:UC3Mx+ISA.1x+3T2017/courseware/1b3mlg10f4mmo91brl192j1g65p/1b3mlg101os2f21koe12vnl7qu/2?activate\\_block\\_id=block-v1%3AUC3Mx%2BISA.1x%2B3T2017%2Btype%40vertical%2Bblock%401b3mlg101d25c3412mekuh9o6g](https://courses.edx.org/courses/course-v1:UC3Mx+ISA.1x+3T2017/courseware/1b3mlg10f4mmo91brl192j1g65p/1b3mlg101os2f21koe12vnl7qu/2?activate_block_id=block-v1%3AUC3Mx%2BISA.1x%2B3T2017%2Btype%40vertical%2Bblock%401b3mlg101d25c3412mekuh9o6g). [Accessed: 19-May-2019].
- [20] Apuntes de la asignatura: Visión y Roboótica, Area ISA, Universidad de Zaragoza, 2019
- [21] M. Sanchez-Garcia, R. Martinez-Cantin, J. J. Guerrero, “Semantic and structural image segmentation for prosthetic vision,” Artículo en revisión 2019.
- [22] A. Badías Herbera, “Simulación de prótesis visual con sensor RGB-D.” TFM, EINA Universidad de Zaragoza 2016.
- [23] J. Weiland and M. Humayun, “Retinal prosthesis,” in *Neural Engineering: Second Edition*, Springer US, 2013, pp. 635–655.
- [24] J. Hirsch and C. A. Curcio, “The spatial resolution capacity of human foveal retina,” *Vision Res.*, vol. 29, no. 9, pp. 1095–1101, 1989.
- [25] E. Zrenner *et al.*, “Subretinal microelectrode arrays allow blind retinitis pigmentosa patients to recognize letters and combine them to words,” in *Proceedings of the 2009 2nd International Conference on Biomedical Engineering and Informatics, BMEI 2009*, 2009.
- [26] C. Veraart *et al.*, “Visual sensations produced by optic nerve stimulation using an implanted self-sizing spiral cuff electrode,” *Brain Res.*, vol. 813, no. 1, pp. 181–186, Nov. 1998.
- [27] J. Delbeke, M. Oozeer, and C. Veraart, “Position, size and luminosity of phosphenes generated by direct optic nerve stimulation,” *Vision Res.*, vol. 43, no. 9, pp. 1091–1102, 2003.

- [28] M. E. Brelén, F. Duret, B. Gérard, J. Delbeke, and C. Veraart, "Creating a meaningful visual perception in blind volunteers by optic nerve stimulation," in *Journal of Neural Engineering*, 2005, vol. 2, no. 1.
- [29] F. Duret, M. E. Brelén, V. Lambert, B. Gérard, J. Delbeke, and C. Veraart, "Object localization, discrimination, and grasping with the optic nerve visual prosthesis.," *Restor. Neurol. Neurosci.*, vol. 24, no. 1, pp. 31–40, Jan. 2006.
- [30] S. C. Chen, G. J. Suaning, J. W. Morley, and N. H. Lovell, "Simulating prosthetic vision: I. Visual models of phosphenes," *Vision Research*, vol. 49, no. 12, pp. 1493–1506, Jun-2009.
- [31] "Introducción a la visión por computador: desarrollo de aplicaciones con OpenCV | edX." [Online]. Available: <https://www.edx.org/es/course/introduccion-la-vision-por-computador-uc3mx-isa-1x-0>. [Accessed: 28-May-2019].
- [32] "Tema localización espacial." Apuntes de la asignatura: Automatización Flexible y Robótica, Area ISA, Universidad de Zaragoza, 2019
- [33] J.R. Sommerhalder, A. Perez Fornos, K. Chanderli, L. Colin, X. Schaer, F. Mauler, A.B. Safran, M. Pelizzone; Minimum Requirements for Mobility in Unpredictable Environments . *Invest. Ophthalmol. Vis. Sci.* 2006;47(13):3204.