

Trabajo Fin de Máster

Sistema de asistencia a la navegación humana
basado en la cámara de eventos

Event Camera System for the Assistance of Human
Navigation

Autor: Pedro Luis Compais Serrano

Director: Rubén Martínez Cantín
Ponente: José J. Guerrero Campo

Máster en Ingeniería Industrial
Escuela de Ingeniería y Arquitectura

Universidad Zaragoza
2019

Resumen

El objetivo principal de este trabajo es el desarrollo y evaluación de un sistema de asistencia a la navegación de personas basado en la innovación de la cámara de eventos. Esta novedosa tecnología tiene potencial suficiente para revolucionar el actual paradigma de los sensores de visión, gracias a su gran rapidez y sensibilidad al movimiento, abriendo el camino hacia las aplicaciones de detección de colisiones.

Con la realización del trabajo, se ha abordado el análisis de la información que proporciona la cámara de eventos, el diseño y simulación del sistema de asistencia a la navegación humana, la adaptación del procesamiento de imágenes a la cámara de eventos y su evaluación como sensor de visión de los sistemas de asistencia a la navegación humana.

Para conseguir los objetivos propuesto en este trabajo se ha realizado el estudio teórico de la cámara de eventos, el desarrollo del propio sistema de asistencia a la navegación y su validación ante escenarios y movimientos realistas mediante el simulador ESIM de la cámara de eventos y el motor gráfico Unreal Engine.

En este trabajo se ha conseguido desarrollar, simular y validar el sistema de asistencia a la navegación humana basado en la cámara de eventos, además de progresar en el conocimiento e implementación de la cámara de eventos como sensor de visión.

Agradecimientos

En primer lugar quería agradecer a la Cátedra Mobility City de la Fundación Ibercaja la ayuda al estudio concedida a este trabajo en el contexto de las tecnologías implicadas en el campo del Vehículo Conectado y Autónomo y la Movilidad Sostenible, así como su contribución al desarrollo de Aragón mediante el impulso de uno de sus sectores clave, como es la industria de la automoción, gracias a la adaptación de su nuevo paradigma de la movilidad sostenible.

También agradecer a Rubén Martínez y José J. Guerrero su dirección y orientación, así como la oportunidad de desarrollar mi Trabajo Fin de Máster dentro del grupo de Robótica, Percepción y Tiempo Real (RoPeRT) del Instituto Universitario de Investigación en Ingeniería de Aragón (I3A).

De la misma manera, dar las gracias a Melani Sánchez por su colaboración en la resolución de dudas técnicas respecto a la visión protésica, así como por la transferencia de sus conocimientos y experiencia en el desarrollo de algoritmos de representación de fosfenos.

Índice

1. Introducción	1
1.1 Objetivos	2
1.2 Alcance	3
1.3. Metodología.....	3
1.3.1 Herramientas	3
1.4 Estructura de la memoria	4
2. Estado del arte	5
2.1. La cámara de eventos y su simulación.....	5
2.1.1 Principio de funcionamiento de la cámara de eventos.....	5
2.1.2. Simulación de la cámara de eventos	6
2.1.3. Otras características del simulador ESIM	7
2.2. Las prótesis visuales.....	9
2.2.1. Representación de los fosfenos en la visión protésica.....	9
2.2.2. Simulación de la visión protésica	10
2.3. Procesamientos de imágenes	12
2.3.1. Procesamientos para la cámara de eventos.....	12
2.4. La biomecánica del cuerpo humano	12
3. El Sistema de Asistencia a la Navegación, SAN	13
3.1. El diseño del SAN	13
3.1.1. Sensibilidad al movimiento y las vibraciones	13
3.1.2. Versatilidad ante tareas de navegación	13
3.1.3. Adaptabilidad del procesamiento	14
3.2. La simulación del SAN	14
4. Generación de las trayectorias características.....	16
4.1. Sistemas de referencia del movimiento humano	16
4.2. Modelado de los movimientos humanos	17
4.3. Muestreo de los modelos	21
4.4. Transformaciones entre SR.....	21
4.5. Cuaternios y trayectoria definida	22
5. Evaluación y ajuste del simulador ESIM	23
5.1. Estudio de los Eventos, EE	23
5.1.1. Subestudio EE1	23
5.1.2. Subestudio EE2	26
5.2. Estudio de la Trayectoria, ET.....	27
5.2.1. Subestudios preliminares, ET1 y ET2.....	27
5.2.2. Subestudio en profundidad, ET3	28

6. La visión protésica	37
6.1. Programa de Visión Protésica, PVP	37
6.1.1. Lectura ordenada de las imágenes	37
6.1.2. Procesamiento de las imágenes	37
6.1.3. Simulación de la visión protésica	41
6.2. Estudio de la Visión Protésica, EVP	42
7. Conclusiones y trabajo futuro	50
7.1. Trabajo futuro	51
Bibliografía	52
Anexos	54
Anexo 1. Desarrollos matemáticos	55
Cálculo de la posición del SR trayectoria respecto al SR espacial.....	55
M1 (recta, bajo) con S_0 y S_4	55
M2 (recta, alto) con S_1 y S_4	55
M3 (curva, bajo) con S_2 y S_4'	56
M4 (curva, alto) con S_3 y S_4'	56
Desarrollo de las transformaciones entre SR	57
Demostración del producto de matrices de rotación en el mismo eje.....	58
Anexo 2. Representación en cuaternios	60

1. Introducción

La representación del movimiento en los medios visuales convencionales se consigue mediante la grabación o proyección de una serie de imágenes estáticas, de forma síncrona con un periodo constante. Esta representación del movimiento tiene dos problemas: se desconoce qué ocurre en los instantes entre imágenes y se graban elementos repetidos (Figura 1). Como alternativa a estos problemas, el aumento del número de imágenes utilizadas por segundo (fotogramas por segundo, fps) sigue sin ser eficaz en los ámbitos de alta velocidad, y tiene una baja eficiencia debido a que aún se generan más datos repetidos. La reciente innovación de la cámara de eventos consigue solucionar estos problemas, postulándose como un sensor de visión que complementa a las cámaras convencionales.

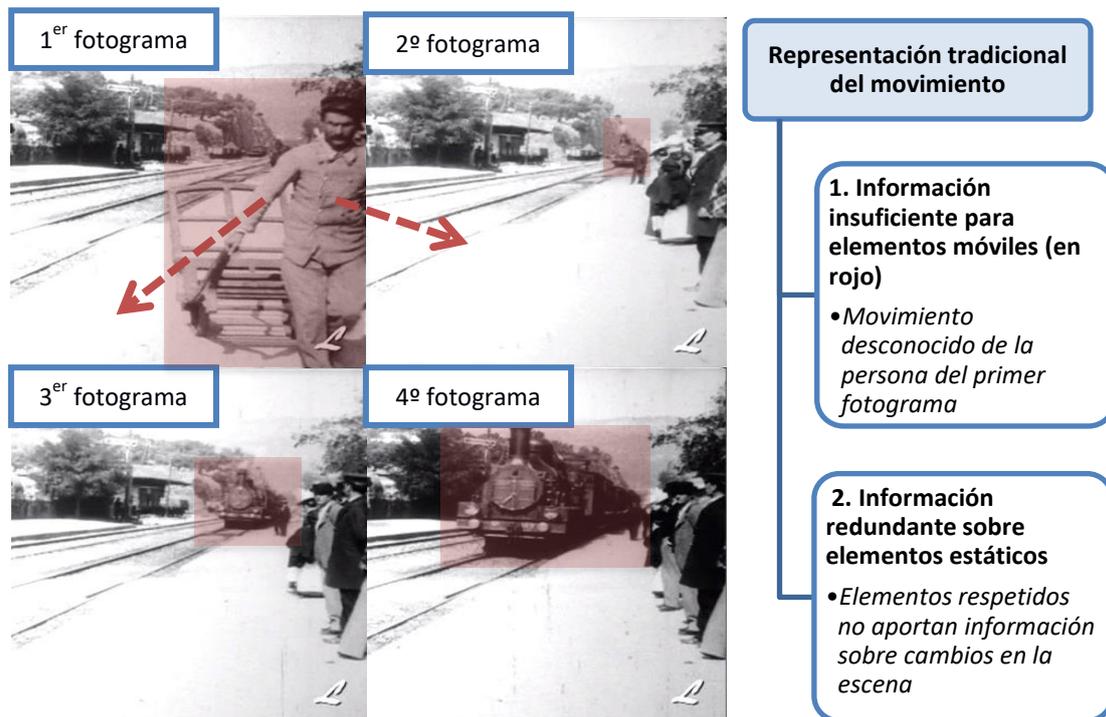


Figura 1. Selección de fotogramas de Llegada de un tren a La Ciotat (Louis Lumière, 1895) y problemas de la representación tradicional del movimiento. Vídeo disponible a través de https://www.moma.org/learn/moma_learning/louis-lumiere-arrivee-dun-train-a-la-ciotat-arrival-of-a-train-at-la-ciotat-1895/.

La nueva tecnología de la cámara de eventos tiene potencial suficiente para cambiar el actual paradigma de los sensores de visión. Estas cámaras almacenan información de los elementos móviles de la escena de manera asíncrona, cuando se producen los movimientos (Figura 2). Respecto a las cámaras tradicionales, la cámara de eventos es más rápida, puede utilizarse en entornos de menor luminosidad y la información que almacena es más eficiente, abriendo el camino hacia las tareas de detección de colisiones. Estas cámaras tienen un coste elevado debido a su novedad, por lo que en este proyecto se trabaja a través de la simulación. En particular, la aparición del *software* de ESIM (definido en la bibliografía como *un Simulador de libre distribución de cámara de eventos* (1)), ha permitido aproximar de forma precisa el comportamiento de estas cámaras. No obstante, existen distintos aspectos de su implementación que siguen necesitando trabajo para llegar a su consolidación definitiva.

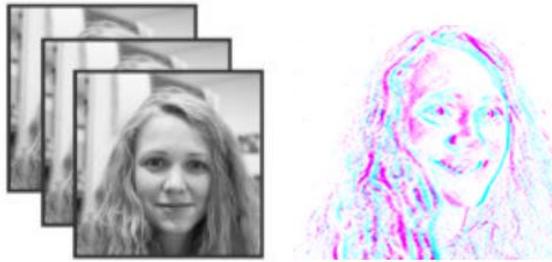


Figura 2. Información recogida por una cámara convencional (izquierda) y una de eventos (derecha), para una escena en la que una persona se mueve. Las imágenes han sido obtenidas de <https://inivation.com/dvs/>.

Una de las posibles aplicaciones de la cámara de eventos son los Sistemas de Asistencia a la Navegación (SAN) de personas o vehículos. Los SAN facilitan el desplazamiento del usuario al proporcionarle información adicional sobre el entorno, de manera que el usuario pueda elegir y realizar el movimiento más adecuado para sus necesidades. Los datos del entorno que rodea al usuario son captados por los sensores del SAN, que son procesados para extraer la información relevante al usuario. Por ejemplo, el usuario de un SAN podría ignorar la existencia de un obstáculo en su trayectoria que le podría hacer caer. En esta situación, el SAN tomaría datos del entorno del usuario, a través de los cuales detectaría el agujero y avisaría de ello al usuario. De esta manera, el usuario trazaría una nueva trayectoria con el fin de sortearlo.

En el caso de los SAN que utilizan sensores de visión, los datos del entorno se recopilan en forma de imágenes por una cámara, que luego son procesadas con algoritmos desarrollados en el área de visión por computador. Dentro de los SAN que emplean cámaras, se distingue un grupo que tiene por usuarios a las personas con discapacidad visual; los SAN para visión protésica. Generalmente dentro de esta tipología, se tiene como objetivo aumentar la capacidad de visión del usuario mediante una prótesis visual. Estas prótesis se componen de un implante de electrodos, una microcámara montada sobre una montura de gafas que lleva el usuario y una unidad de procesamiento de imagen. Tras procesar las imágenes tomadas por la cámara, la información extraída de ellas se transforma en señales eléctricas, que son enviadas de forma inalámbrica al implante de electrodos del paciente. Un SAN para visión protésica es un caso ejemplar de SAN, ya que puede generalizarse para SAN destinados a personas sin discapacidad visual, a excepción de las bloques del implante de electrodos y las señales eléctricas. De esta manera, un SAN para visión protésica basado en la cámara de eventos es de gran utilidad porque permite desarrollar simultáneamente su aplicación en personas con discapacidad visual o sin ella. Dado que la información que proporciona la cámara de eventos difiere de la de una cámara convencional, el procesamiento de los datos de las imágenes también es distinto, por lo que es necesario adaptar los algoritmos tradicionales de visión por computador a este nuevo tipo de cámara.

1.1 Objetivos

A raíz de las cuestiones que plantea la implementación de la cámara de eventos, este trabajo surge para evaluar su comportamiento dentro de los SAN para personas, paso previo a su implementación y aplicación real en estos sistemas. Para ello se ha diseñado, desarrollado, simulado y validado un SAN de personas para visión protésica basado en la cámara de eventos, que puede generalizarse para los usuarios sin discapacidad visual. Los objetivos del trabajo son:

- El análisis de la información proporcionada por las cámaras de eventos
- El desarrollo de un entorno de simulación que modele el comportamiento de la cámara de eventos y utilice escenarios realistas de interior y exterior y trayectorias de personas
- La adaptación de los algoritmos de percepción del entorno y movimiento propio
- La evaluación de la cámara de eventos como sensor de un SAN para personas

1.2 Alcance

Para conseguir los propósitos definidos, se ha realizado un estudio teórico de la cámara de eventos y la visión protésica, se han diseñado y desarrollado los distintos bloques que componen el SAN, y finalmente se ha simulado y validado el sistema con el *software* ESIM y el motor gráfico Unreal Engine.

1.3. Metodología

La metodología seguida en este proyecto se ha basado en la planificación de las actividades a realizar y el control del tiempo dedicado a cada una de ellas, así como en la presentación de los resultados y la consulta y debate de las dudas a través de reuniones con el director y el grupo de trabajo del I3A, o contactando con los desarrolladores de ESIM. De forma general, se han realizado reuniones semanales con el grupo del trabajo del I3A para tener un mayor control y efectuar de forma rápida reajustes de la planificación, debido al notable grado de incertidumbre que presentaba este proyecto, causado por el carácter innovador de la cámara de eventos.

1.3.1 Herramientas

Dentro de este trabajo, se han empleado diversas herramientas para la realización de cada tarea:

- La plataforma de ScienceDirect o el buscador Google Scholar para la búsqueda, lectura y estudio de artículos científicos, dentro del análisis del estado del arte
- El programa Calc de LibreOffice para la planificación y control de tareas de este proyecto, además del modelado y representación gráfica de los movimientos humanos, utilizados para la simulación e introducción de trayectorias de personas dentro del simulador de la cámara de eventos
- La terminal de Ubuntu (Linux) con el fin de ejecutar programas y simuladores, incluyendo ESIM, los escenarios de Unreal Engine y su complemento de UnrealCV, además de escribir comandos de almacenamiento y manejo de archivos y de comunicarse con las herramientas de ROS, englobado todas estas acciones en la simulación y experimentación del SAN
- El Entorno de Desarrollo Integrado (IDE) de PyCharm, los tutoriales y la documentación de Python 3 y sus módulos de *regular expression operations (re)*, *mathematical functions (math)* y *miscellaneous operating system interfaces (os)*, así como los paquetes de NumPy, OpenCV y SciPy, para el desarrollo de programas y funciones que den solución a distintas necesidades surgidas en este trabajo, como la lectura, ordenación y transformación de los

resultados y experimentos de ESIM, incluyendo el procesamiento de sus imágenes de eventos y la representación de la visión protésica.

1.4 Estructura de la memoria

Tras este primer capítulo de introducción al proyecto, el capítulo 2 se centra en el análisis del estado del arte de la cámara de eventos, las prótesis visuales, la visión por computador y la locomoción humana. En el capítulo 3 se realiza el diseño del SAN y del su entorno de simulación, cuyos componentes se desarrollarán en los siguientes capítulos. El capítulo 4 continuará con el modelado de las trayectorias características, que se utilizará después para evaluar y ajustar al simulador de cámara de eventos, en el capítulo 5. En el capítulo 6 se adaptarán distintos algoritmos de visión por computador para procesar la información simulada de la cámara de eventos y se realizará la simulación de la visión protésica. En el capítulo 7 se recogerán todas las conclusiones extraídas y las áreas de trabajo futuro dentro del ámbito de este proyecto. En la Figura 3 se establece una correspondencia entre los objetivos y el alcance del trabajo con los capítulos de la memoria.

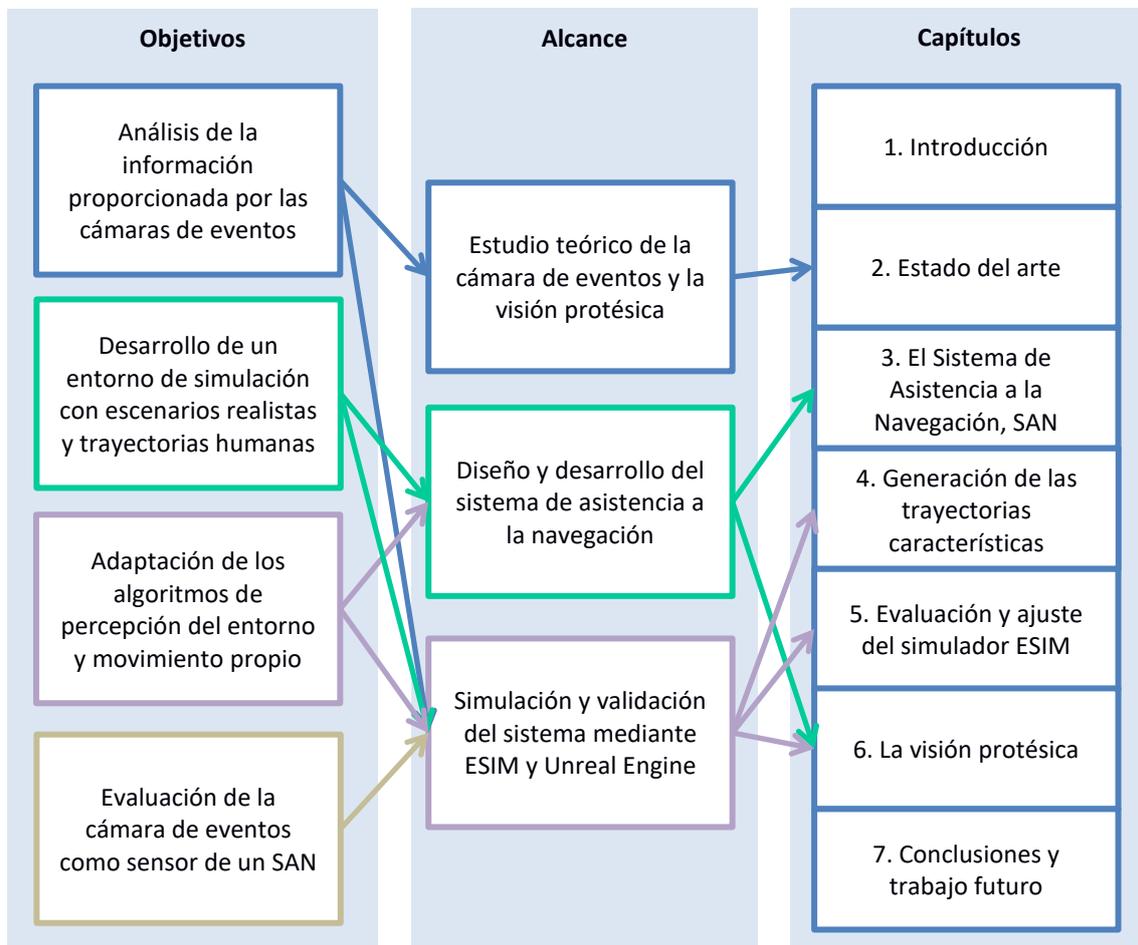


Figura 3. Correspondencia entre los objetivos, alcance y capítulos de la memoria.

2. Estado del arte

En primer lugar, se ha estudiado la cámara de eventos para su utilización como sensor de los SAN. Debido a que la aplicación del SAN se ha dirigido hacia las prótesis visuales, el área de éstas también ha sido abordada. La disciplina del procesamiento de imágenes ha sido considerada dada su relación con los sensores de visión. Finalmente, se ha abordado el estudio y caracterización de los movimientos y la locomoción de las personas

2.1. La cámara de eventos y su simulación

La cámara de eventos presenta una novedosa tecnología, que se explica a continuación a través de una comparativa con las cámaras convencionales. Como toda innovación, esta tecnología tiene un gran potencial pero un coste actual elevado, debido a que se encuentra en fase de desarrollo. Por ello, los simuladores de cámara de eventos se postulan como herramientas idóneas para el progreso y consolidación de esta tecnología, entre los que destaca el simulador ESIM.

2.1.1 Principio de funcionamiento de la cámara de eventos

Al realizar una fotografía con una cámara tradicional, se almacena información de la luz reflejada por la superficie de los objetos, generalmente su intensidad luminosa si las imágenes son en escala de grises o su combinación de las luces azul, roja y verde si son en color. La grabación de vídeo con estas cámaras se basa en la captura síncrona de imágenes con un periodo constante de tiempo. En contraposición a las cámaras convencionales, la cámara de eventos registra los cambios que se producen en una escena de forma asíncrona cuándo se producen. Los eventos son variaciones de luminosidad superiores a un umbral definido, y son positivos (o negativos) cuando se incrementa (o reduce) la luminosidad en una región de la escena (Figura 4, (1)).

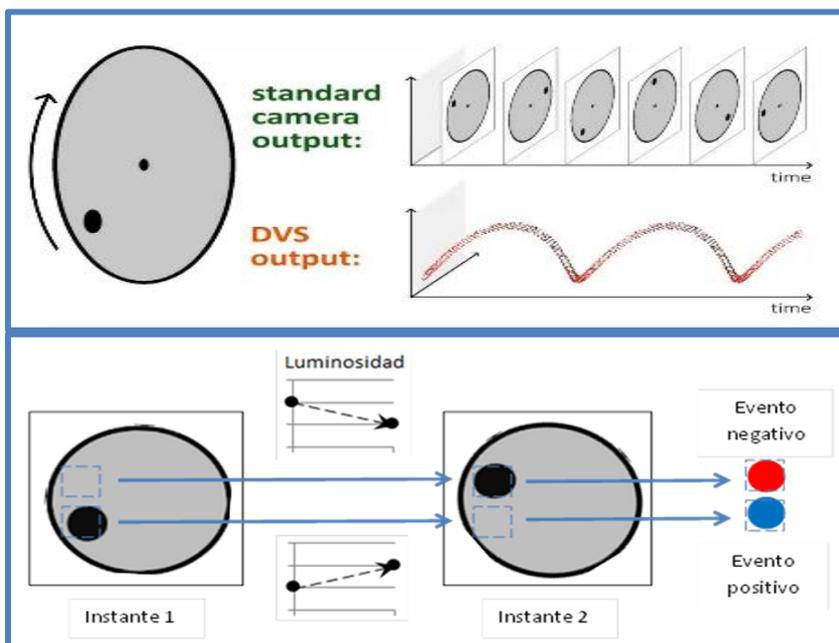


Figura 4. Superior: información obtenida por una cámara convencional y una de eventos, al observar un disco giratorio con un círculo negro (1). Inferior: generación de eventos respecto a las imágenes tomadas por una cámara convencional.

2.1.2. Simulación de la cámara de eventos

En la simulación de cámaras de eventos con ordenador, se parte de un escenario y cámara convencional virtuales. La simulación de la cámara convencional permite obtener imágenes con una tasa de fps aproximadamente infinitos, consiguiendo un flujo de información suficientemente continuo para replicar el comportamiento asíncrono y la baja latencia de la cámara de eventos. A partir de imágenes consecutivas, se estiman las regiones e instantes en los que se produce un evento, que puede no coincidir con el instante de almacenamiento de las imágenes. La simulación de eventos se puede calcular según dos variaciones: la de la luminosidad, al igual que el comportamiento real de la cámara de eventos, o los desplazamientos de píxel. Actualmente ya existen simuladores de la cámara de eventos, como por ejemplo los simuladores DAVIS e InteriorNet (2), (3), cuyo comportamiento se basa en el uso de una tasa de fps constante, definida por el usuario (Figura 5). Se destaca que en la simulación hay ocasiones en las que dos imágenes consecutivas no generan eventos, por lo que en esos casos se toman más imágenes de las necesarias. De esta manera, la simulación no reproduce en su totalidad la calidad de la información grabada por la cámara de eventos, dado que aunque no haya cambios en la escena, se seguirán tomados imágenes. La reducción de los fps para evitar obtener información redundante hace que la simulación de la cámara de eventos se aproxime menos a su comportamiento real (1).

ESIM lleva un paso más allá la simulación de eventos, y modifica la tasa de fps según el movimiento detectado en las imágenes, como se aprecia en la Figura 5. Si existe más movimiento, ESIM selecciona una tasa de fps superior y toma más imágenes. De igual manera, si el movimiento es menor, ESIM reduce los fps y almacena menos imágenes. Para conseguir este muestreo adaptativo, es necesario estimar el movimiento de instantes de tiempo futuros. Este movimiento depende del movimiento relativo entre la cámara y la escena, cuya representación se describe en el campo de la visión mediante el flujo óptico. El flujo óptico es entendido normalmente como la estimación de los desplazamientos vertical y horizontal de un píxel entre una imagen y la siguiente (1), (4). Para describir el movimiento de la cámara es necesario conocer su aceleración, y ESIM la calcula a partir de expresiones analíticas. Estas expresiones se obtienen gracias a una definición de la trayectoria de cámara que es continua en el tiempo, a través de la interpolación con *splines* de una serie de posiciones discretas. Los *splines* son funciones utilizadas en la parametrización del movimiento, de utilidad por su generación de funciones analíticas y simples en el tiempo (1), (5). El muestreo adaptativo de ESIM permite calcular de forma más precisa en qué momento es generado cada evento, a la vez que se reduce la información recogida cuando el movimiento se reduce. En contraposición a los simuladores DAVIS e InteriorNet, ESIM calcula y modifica automáticamente su tasa de muestreo según el movimiento en la escena, en lugar de necesitar su ajuste manual. Además, ESIM obtiene resultados igual de precisos que DAVIS e InteriorNet, pero con un 40% menos de número de muestras y tiempo de proceso (1).

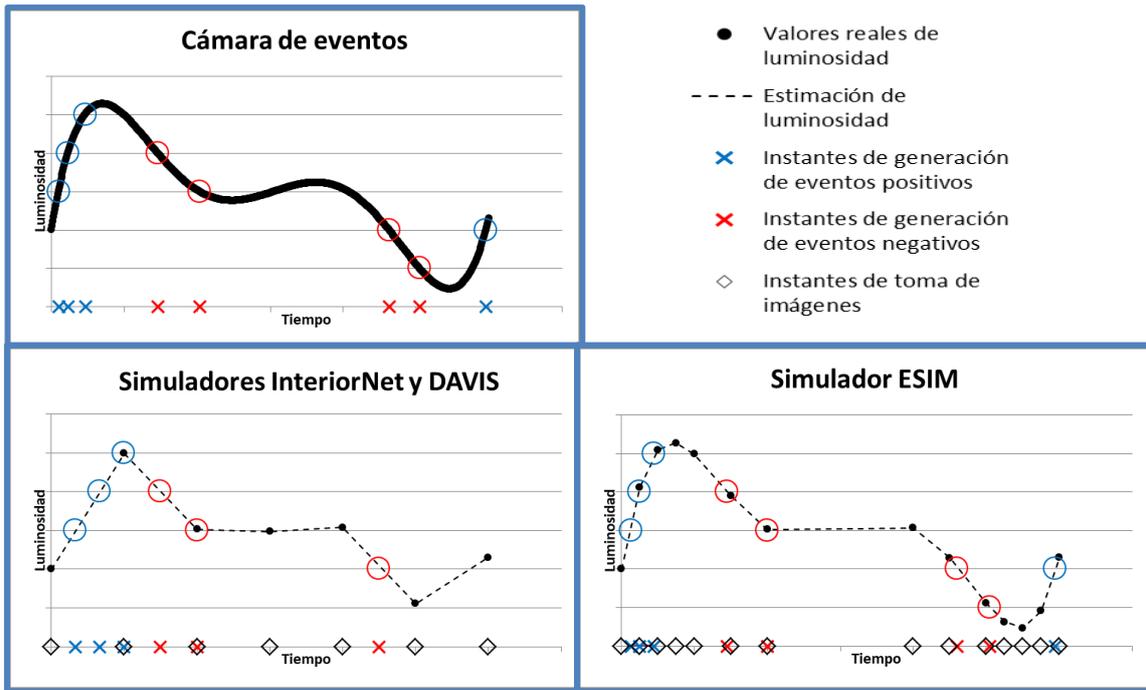


Figura 5. Comportamiento de la cámara de eventos y sus simuladores. Los simuladores toman valores reales de luminosidad en los instantes de muestreo (toma de imágenes). En la simulación, se observa que el instante en el que se produce cada evento no tiene por qué corresponder con el de la toma de imágenes, por lo que se simula correctamente el carácter asíncrono de la cámara de eventos.

2.1.3. Otras características del simulador ESIM

El simulador ESIM tiene disponibles distintos motores gráficos para representar escenas de dos y tres dimensiones, que en el último caso se basan en OpenGL y Unreal Engine 4 (UE4). El motor de OpenGL es el que tiene con diferencia la mayor velocidad de renderizado, pero su recreación de entornos no es fotorrealista, en contraposición al de UE4, que por otro lado está muy alejado de conseguir tiempo real en la simulación. En el caso de utilizar el simulador ESIM junto al motor gráfico de UE4, se requiere la extensión de UnrealCV, que permite la comunicación entre ambos y proporciona una librería de comandos específicos para la investigación de la visión por computador. En la Figura 6 se puede observar la diferencia en el nivel de realismo entre el motor de OpenGL y el de UE4 (6).



Figura 6. Primera y segunda imagen: ejemplo de un escenario tridimensional y una textura para el motor de OpenGL (6). Tercer y cuarta imagen: escenario virtual de un salón para el motor de UnrealCV (6).

Independientemente del motor gráfico escogido, el simulador ESIM calcula la polaridad, posición en la imagen y tiempo característico de cada evento generado, para una trayectoria de la cámara en un escenario de UE4. Esta trayectoria se basa en el ajuste automático de *splines* a un conjunto discreto de localizaciones de la cámara, que pueden ser obtenidas también automáticamente o definidas por el usuario. En este último caso, se necesita conocer la escala del escenario y la localización del origen de coordenadas, así como utilizar la nomenclatura preestablecida en ESIM para las localizaciones, descritas por su instante de tiempo en nanosegundos, posición tridimensional en dimensiones de la simulación y rotación expresada con cuaternios (Figura 7). Las localizaciones de la cámara, junto a otros datos, son publicadas en ROS y pueden almacenarse en ficheros de texto para su posterior uso. ESIM también permite relacionar la información obtenida por las cámaras convencionales y las de eventos, acumulando los eventos relacionados con cada imagen de la cámara convencional (Figura 8). De esta manera se obtiene una comparativa más intuitiva, aunque aproximada, de los dos tipos de información.

```

---
header:
  seq: 727
  stamp:
    secs: 0
    nsecs: 551000000
  frame_id: "map"
pose:
  position:
    x: 24.5131199989
    y: -19.7944304722
    z: 3.0229074819
  orientation:
    x: 0.00232959918355
    y: -0.000708785746816
    z: -0.706488753204
    w: 0.70772008039
---
header:
  seq: 728
  
```

Figura 7. Extracto del fichero de texto exportado desde ROS para la localización de la cámara.

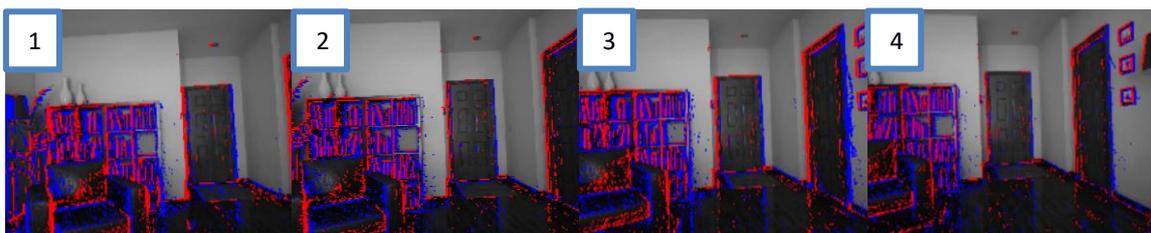


Figura 8. Secuencia de imágenes de eventos del simulador ESIM. Esta secuencia constituye una comparativa aproximada del comportamiento de una cámara convencional con el de una de eventos, donde los eventos se han acumulado sobre las fotografías tradicionales. Se ha usado el escenario de un salón, con la cámara moviéndose hacia su derecha.

2.2. Las prótesis visuales

Las prótesis visuales tienen como objetivo mejorar la capacidad de visión de las personas con ceguera o deficiencia visual, y así mejorar su calidad de vida. Las prótesis retinales se componen de un implante de electrodos, una microcámara montada sobre una montura de gafas y una unidad de procesamiento de imagen (Figura 9). El usuario se coloca las gafas en la cabeza, de manera que las fotografías capturadas por la microcámara se asemejen a lo que vería una persona sin discapacidad visual. Estas fotografías son enviadas a la unidad de procesamiento, que modifica las imágenes a partir de técnicas de visión por computador y transforma su información en señales eléctricas enviadas de forma inalámbrica al implante de electrodos del paciente. En función de las señales eléctricas, el implante genera estímulos eléctricos en una parte del sistema nervioso, concretamente en un punto del sistema visual, dando lugar a estímulos visuales, conocidos como fosfenos, similares a puntos luminosos (Figura 9, (7)).

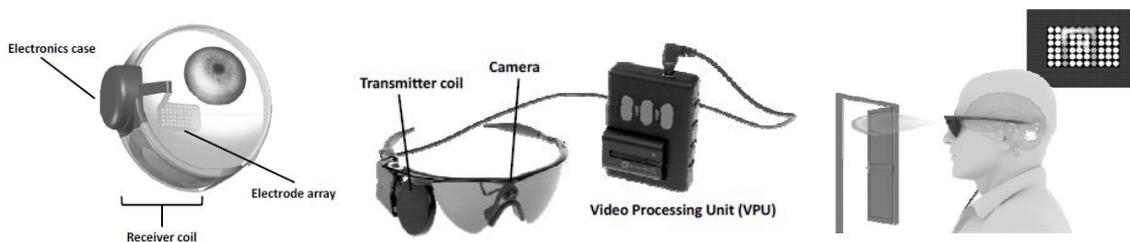


Figura 9. Izquierda: componentes internos de un sistema de prótesis visual Argus II. Centro: componentes externos de un sistema de prótesis visual Argus II. Derecha: imagen fosfénica que recibiría el paciente con la prótesis. En este caso, la cámara montada sobre las gafas fotografía el marco de la puerta. Esta imagen se procesa y la información se emite al implante de electrodos. De esta forma, la persona es capaz de ver un patrón de fosfenos en el que se distingue una forma semejante al marco de la puerta (7).

El desarrollo de un SAN para visión protésica permite su generalización para otras aplicaciones, debido a que los sensores de visión y las técnicas de visión por computador que se utilizan pueden ser los mismos. Independientemente de la aplicación, los SAN almacenan las fotografías de la cámara y aplican las técnicas de visión por computador; la obtención y el envío de las señales eléctricas, así como la generación de los estímulos eléctricos, son tareas posteriores y específicas de un SAN para visión protésica.

2.2.1. Representación de los fosfenos en la visión protésica

Las prótesis visuales se pueden dividir en dos grandes grupos según su representación de los fosfenos. Sin embargo, en la bibliografía no se ha encontrado una nomenclatura extendida para referenciarlos. Por ello, en este trabajo se propone utilizar los términos de *representación figurativa* (a la que se hace referencia con el término *scoreboard* en el trabajo de Macé et al. (8)) y *representación esquemática*. La representación figurativa se basa en la transcripción de las imágenes a fosfenos, con el objetivo de reproducir lo más fielmente posible la fotografía original y proporcionar a la persona con prótesis una visión similar a la de las personas sin discapacidad (Figura 10, (9)). Por otro lado, la representación esquemática precisa de un procesamiento de alto nivel que refine en mayor medida la información de las imágenes, cuya representación difiere significativamente de la escena real, lo que permite utilizar prótesis con una resolución de fosfenos inferior. Para explicarlo, se utiliza el siguiente ejemplo: una persona con prótesis visual se encuentra delante de una mesa con diferentes objetos, de entre los que quiere localizar uno de ellos para cogerlo con las manos. El objeto a

localizar se reconoce en las fotografías de la cámara gracias al procesamiento de alto nivel, y la posición de este objeto en el campo visual de la persona con la prótesis es luego representada mediante un único fosfeno. De esta manera no se representa la escena, ni el resto de objetos que aparecen en ella, ni siquiera la forma del objeto a localizar (Figura 11, (8)).

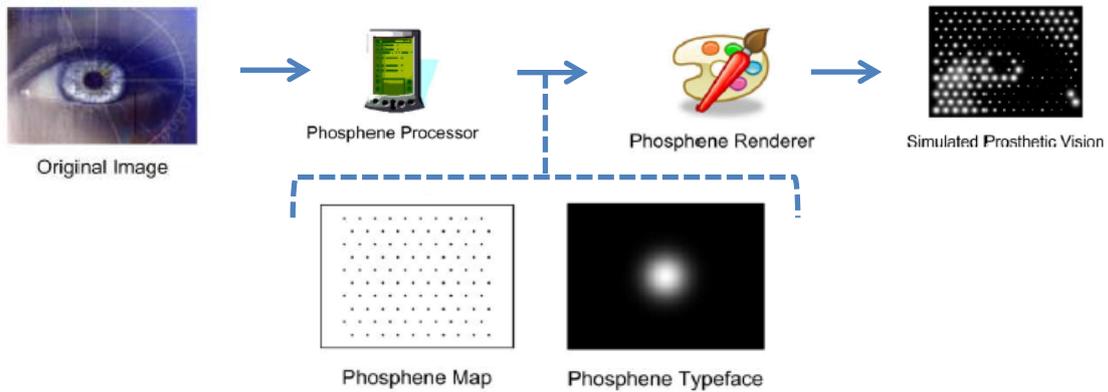


Figura 10. Representación figurativa de la visión protésica. Para obtener la visión protésica ante una imagen, primero se ajusta la resolución espacial y de luminosidad de la imagen original y después se asigna un fosfeno a cada uno de sus píxeles, según su intensidad (10), (9).

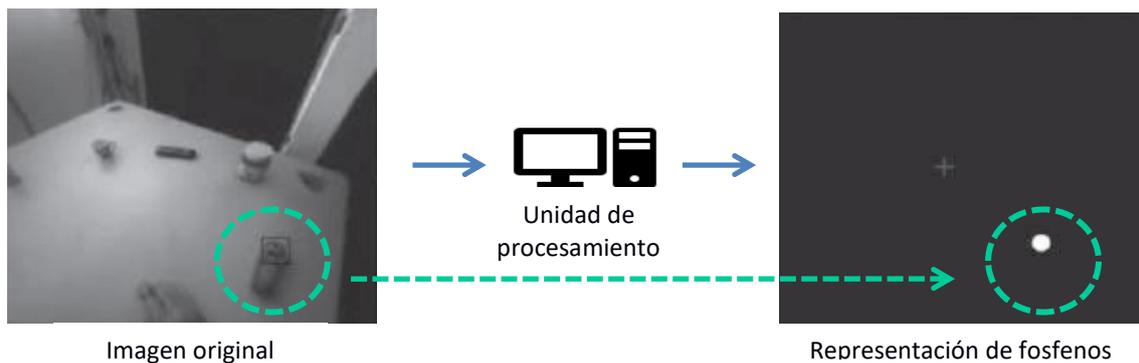


Figura 11. Representación esquemática de la visión protésica. En este ejemplo, en la imagen original aparecen distintos objetos, de los que sólo interesa uno de ellos. El objeto en cuestión es localizado, y su posición en el campo visual se señala con un fosfeno (8).

2.2.2. Simulación de la visión protésica

El actual proceso de desarrollo de las prótesis visuales hace que solamente un pequeño grupo de pacientes las esté utilizando, obteniéndose un reducido número de pruebas clínicas, lo que dificulta el estudio y avance de las prótesis. Para solventar esta cuestión se recurre a la simulación de la visión protésica, basada en replicar en imágenes la visión de una persona con prótesis y deficiencia visual, para analizarlas y evaluarlas en ensayos cuyos sujetos sean personas con visión normal (Figuras 12 y 13). De esta manera, se consigue incrementar el número de posibles sujetos de pruebas, y así poder realizar una investigación más precisa. Para estudiar lo que vería una persona con discapacidad visual y prótesis, se recurre al modelado visual de los fosfenos. A partir de la observación empírica, se han recogido distintos atributos de estos estímulos visuales, como por ejemplo: su forma, tamaño o distribución (9). Generalmente, los fosfenos se representan en escala de grises como pequeños círculos de distinta luminosidad, distribuidos en una red cuadrada o hexagonal. Dentro del área de cada fosfeno, la intensidad es máxima en el centro y se reduce al alejarse de él. Esta distribución de la luminosidad se describe mediante la siguiente función gaussiana $G(x, y)$ (10):

$$G(x, y) \propto \exp\left\{\frac{(x - \mu_x)^2 + (y - \mu_y)^2}{2\sigma^2}\right\}$$

En $G(x, y)$, μ_x y μ_y representan el centro del fosfeno, mientras que σ hace referencia a la desviación típica. Para obtener su representación en fosfenos, la imagen original se multiplica por la función $G(x, y)$, pero antes se cuantizan sus niveles de gris y reduce su resolución espacial para que coincida con la del implante de electrodos. Otro aspecto a modelar de los fosfenos es la incapacidad de estimular una porción de las células nerviosas conectadas con el implante, debido a su deterioro o necrosis. Este hecho se simula mediante el parámetro de *dropout*, que indica la proporción de fosfenos no generados, elegidos aleatoriamente entre todos los electrodos del implante (10).



Figura 12. Izquierda: una de las configuraciones de sistema para la simulación de la visión protésica, compuesto por una microcámara y dos monitores (9). Centro: fotografía de un salón obtenida por una cámara convencional. Derecha: simulación de la visión protésica para la imagen anterior (10).

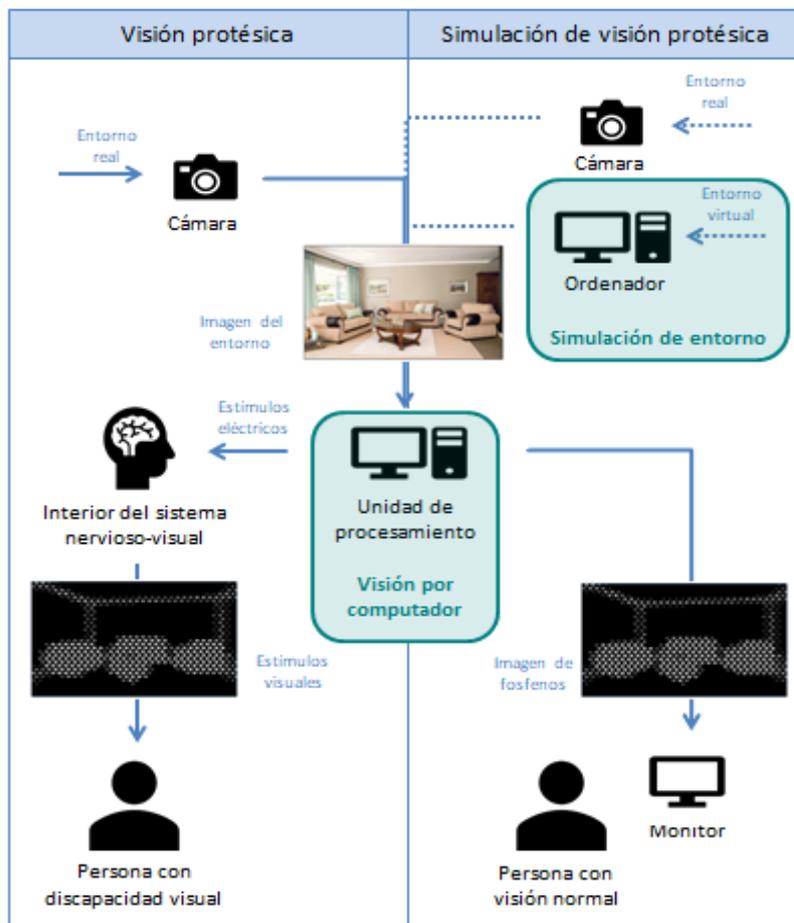


Figura 13. Comparativa entre la visión protésica y su simulación (10). En el gráfico, las imágenes del escenario y la visión por computador son comunes a la visión protésica y a su simulación. Destacar que en la simulación de prótesis visuales, el ordenador puede desempeñar también las funciones de unidad de procesamiento y monitor.

2.3. Procesamientos de imágenes

El procesamiento de imágenes se engloba dentro de la disciplina de la visión por computador, que se ha consolidado en los últimos años dando solución a los problemas de inspección, localización y navegación, entre otros. A continuación se describen dos procesamientos relacionados con el ámbito de este trabajo.

2.3.1. *Procesamientos para la cámara de eventos*

Ya sea en su simulación o implementación real, los datos proporcionados por una cámara de eventos se pueden procesar para extraer información más relevante. Actualmente se realizan procesamientos tanto de bajo como de alto nivel.

Los procesamientos de bajo nivel se han desarrollado para movimientos de alta velocidad en escenarios sencillos, como un dron dentro de una habitación vacía. En estos casos, la cámara de eventos está anclada al robot y observa una pared sin textura, sobre la que se encuentra o aparece una figura geométrica simple, pudiendo realizarse un seguimiento de la localización del robot o la detección de colisiones y cálculo de maniobras evasivas. Dichas tareas se basan en el conocimiento a priori de la figura que capta la cámara, así como de la transformada de Hough para realizar el reconocimiento de la figura (11), (12). Por otro lado, los procesamientos de alto nivel se han utilizado para la reconstrucción de imágenes a partir de eventos en aproximadamente tiempo real, o redes neuronales que conducen automáticamente vehículos por los carriles de calzadas (4), (13).

En el caso de los SAN para personas, el sistema debe facilitar la navegación humana, pero es la persona quien interpreta la información proporcionada, y decide su próximo desplazamiento. Al utilizar un SAN, la persona no es guiada automáticamente por un ordenador, por lo que no es necesario utilizar procesamientos de alto nivel. Por otro lado, los procesamientos de bajo nivel utilizados por Falanga et al. y Mueggler et al. (11), (12), se han desarrollado para escenarios sencillos cuyos modelos geométricos son conocidos. Sin embargo, en el caso de las implementaciones reales de los SAN, los escenarios son complejos y desconocidos generalmente. De esta manera, dentro del estudio realista de un SAN basado en la cámara de eventos, se precisa del desarrollo de nuevos procesamientos de bajo nivel.

2.4. La biomecánica del cuerpo humano

Los sistemas de asistencia a la navegación humana se pueden servir de distintos sensores para obtener información del entorno que rodea a la persona, y así poder guiarla a través de él. Generalmente estos sistemas son portátiles, de forma que los sensores recopilan distintos datos según en qué parte del cuerpo se coloquen, ya que éstas se mueven de manera diferente. Por ello, para desarrollar estos SAN, se requieren estudios de la locomoción humana, englobados en las áreas de neurología, fisiología y biomecánica. Los movimientos que realizamos las personas dependen de diversos factores, como el tipo y velocidad del desplazamiento efectuado, o las características físicas de la persona que realiza la locomoción. De forma general, se observa que las distintas partes del cuerpo se coordinan entre sí para reducir el movimiento de la cabeza y los ojos, o realizar desplazamientos con menor trabajo o mayor eficiencia. En el caso de un SAN para visión protésica, la microcámara es colocada sobre la cabeza del usuario, cuyos movimientos son modelados y aproximados en el capítulo 4 (14), (15), (16).

3. El Sistema de Asistencia a la Navegación, SAN

A partir de la aplicación considerada en este trabajo para el SAN, basado en la cámara de eventos y para la visión protésica y su generalización para usuarios sin discapacidad visual, se han extraído unos requisitos de diseños particulares. La validación de este SAN se ha realizado mediante su simulación, debido al elevado coste actual de la cámara de eventos, causado por su actual fase de desarrollo. Los distintos simuladores que se han empleado en dicha validación se han descrito al final de este capítulo. En este capítulo se enuncian distintas técnicas de procesamiento y algoritmos de visión por computador, que se han explicado más detalladamente en el capítulo 6.

3.1. El diseño del SAN

Se han identificado varios requerimientos de diseño específicos del SAN considerado en este trabajo:

- Sensibilidad al movimiento y las vibraciones
- Versatilidad ante tareas de navegación
- Adaptabilidad del procesamiento

3.1.1. Sensibilidad al movimiento y las vibraciones

Generalmente, las cámaras convencionales requieren dispositivos de estabilización o suspensión para evitar el desenfoque de movimiento o conseguir una experiencia más agradable a la hora de reproducir vídeos. A diferencia de las cámaras convencionales, las cámaras de eventos no tienen desenfoque de movimiento, por lo que en principio no requieren dispositivos de estabilización o suspensión que lo eviten. Como se ha comentado anteriormente, las cámaras de eventos necesitan movimiento para almacenar información. En particular, requieren de la existencia de movimiento relativo entre la cámara y los objetos de la escena: captarán datos si la cámara se mueve y los objetos no, o viceversa, o los dos se mueven con distinta velocidad. De esta manera, los dispositivos de suspensión reducirían el movimiento de la cámara de alta frecuencia y se captaría menos información de objetos inmóviles, por lo que se ha descartado su uso.

Respecto a la colocación de la cámara de eventos en el usuario, ésta se debe situar sobre una zona con movimiento y no completamente estática. Como se ha mencionado en el capítulo 2, las distintas partes del cuerpo humano se coordinan entre sí para estabilizar el movimiento de la cabeza ante la traslación. De esta manera, la cabeza se ve sometida a movimientos del orden de centímetros, como se detallará en el capítulo 4. En consecuencia, se ha comprobado que la región de la cabeza también cumple el requisito de sensibilidad al movimiento y las vibraciones, por lo que la cámara de eventos también se puede colocar sobre ella.

3.1.2. Versatilidad ante tareas de navegación

Los SAN para personas facilitan información adicional y de utilidad que permite escoger mejor el próximo desplazamiento a realizar. Existen distintas tareas de navegación que se relacionan con los desplazamientos humanos, como la detección de colisiones, definición de rutas o reconocimiento y localización de objetos, y según la tarea a realizar, los datos que necesita la persona son diferentes. Para conseguir que el SAN proporcione datos provechosos

a su usuario independientemente de la tarea que esté realizando, se ha decidido incrementar la calidad de la información común a todas las tareas, a través de un procesamiento de bajo nivel. El procesamiento de bajo nivel incluye una acumulación de imágenes de eventos que permite obtener imágenes con una mayor densidad de información y diversas transformaciones morfológicas para filtrar el ruido o resaltar los resultados, como por ejemplo la combinación de erosión y dilatación.

3.1.3. Adaptabilidad del procesamiento

Se ha determinado diseñar los algoritmos de visión por computador de forma modular, seleccionando varias técnicas de procesamiento de entre las distintas posibilidades, para aplicarlas en su combinación; así se consigue una modificación sencilla y rápida de la configuración de la visión por computador en los ensayos y pruebas. Las técnicas de procesamiento entre las que se puede elegir son:

- Acumulación de imágenes de eventos al principio o final del procesamiento, acumulando todas las imágenes o subconjuntos de ellas
- Apertura, cierre, apertura y cierre o cierre y apertura morfológicos, eligiendo el tamaño de máscara
- Dilatación, seleccionado la dimensión de la máscara
- Filtro percentil, eligiendo el valor del percentil y el tamaño de la máscara

3.2. La simulación del SAN

Para simular el SAN basado en la cámara de eventos para la visión protésica se han necesitado cuatro simuladores, que representan los distintos componentes o información requerida por el SAN. De esta manera, se puede sustituir cualquiera de los simuladores por la información correspondiente obtenida mediante su la implementación real. Los componentes y datos simulados se recogen en la Figura 14, y son:

- Los escenarios, proporcionados como *game binaries* de uso libre por el equipo de UnrealCV (17)
- El movimiento humano, emulado gracias al generador de trayectorias que se ha desarrollado en este trabajo
- La cámara de eventos, que se ha simulado con ESIM (1) y al que se comunica la información sobre el escenario y el movimiento humano considerados
- La prótesis visual, cuya representación de fosfenos utiliza los algoritmos de la unidad de procesamiento de Sánchez-García et al. (10), según se explica en el apartado 2.2.2.

Los movimientos y trayectorias a introducir en ESIM (*trayectorias definidas*) deben tener el formato adecuado y configurarse automáticamente al cambiar su dirección, localización inicial o escenario. Esta última característica es importante ya que la escala y sistema de coordenadas del escenario son desconocidos a priori, por lo que puede ser necesario introducir factores de escala o transformaciones del sistema de coordenadas. Para cumplir estos requisitos se define el bloque del *generador de trayectorias*, que se ha analizado en profundidad en el capítulo 4.

Por último, a diferencia de las simulaciones de Scaramuzza et al. (1), en este trabajo se han considerado las diferencias entre la trayectoria definida y la ajustada, debido a que la utilización de *splines* hace que la trayectoria definida se suavice para la obtención de la

trayectoria ajustada. Esta diferencia se debe al carácter oscilatorio de los movimientos humanos, que para ser aproximado de forma precisa necesita el ajuste del número de segmentos de los *splines* y de su grado. En el capítulo 5 se ha realizado la comparativa entre las trayectorias definidas y ajustadas.

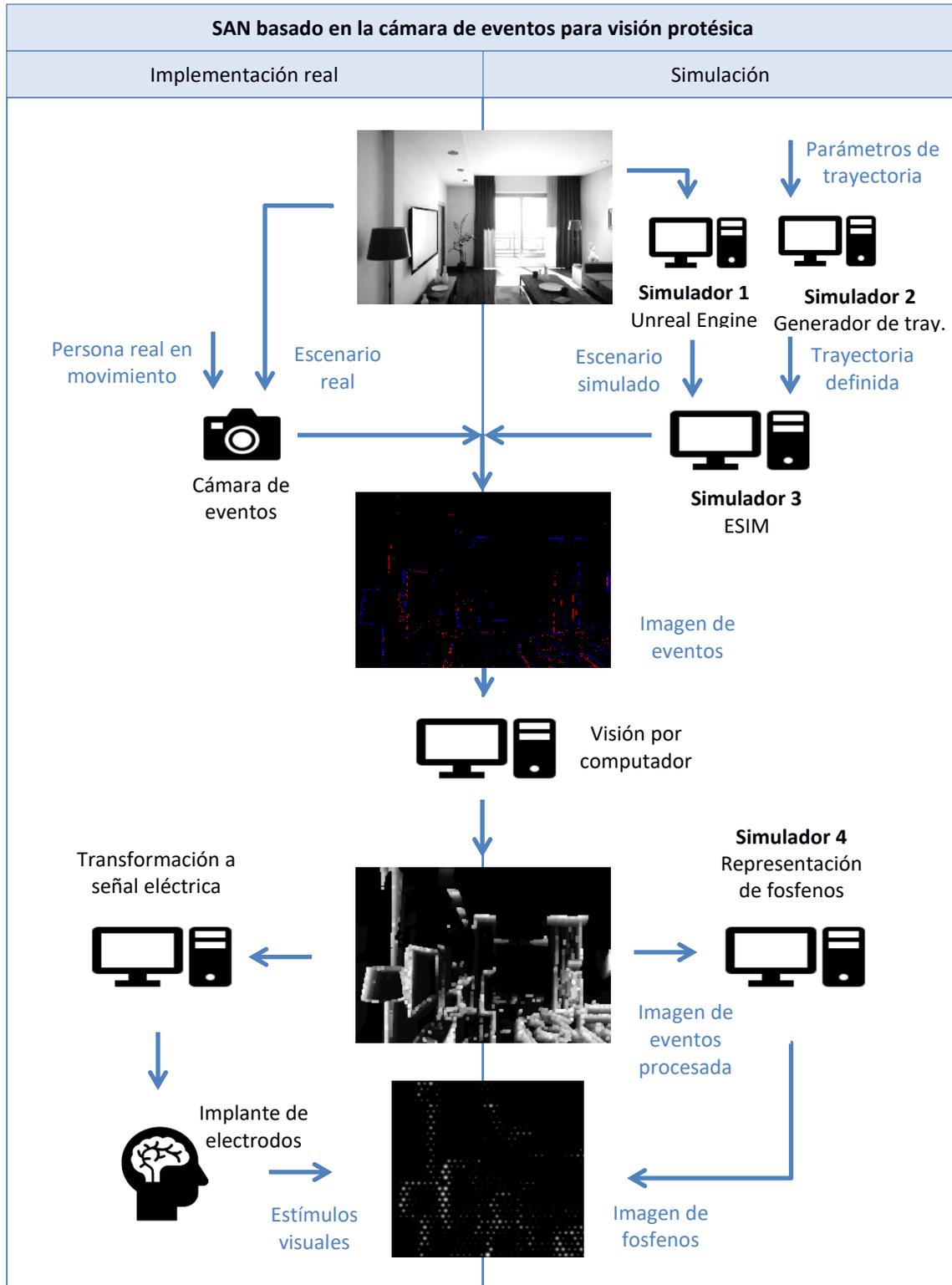


Figura 14. Comparativa entre la implementación real y la simulación del SAN, para la visión protésica y basado en la cámara de eventos. Los elementos presentados en mitad de la figura son comunes a la implementación real y su simulación.

4. Generación de las trayectorias características

El primer paso a dar dentro del entorno de simulación del SAN es la generación de trayectorias, ya que es necesario definir el movimiento de la cámara de eventos para obtener información de ella, y así evaluar el simulador ESIM y los algoritmos de visión por computador. Para realizar las distintas tareas del generador de trayectorias, incluyendo la representación gráfica de los modelos, se han desarrollado varias hojas de cálculo con el programa Calc de LibreOffice. Los distintos bloques del generador de trayectorias se presentan en la Figura 15.

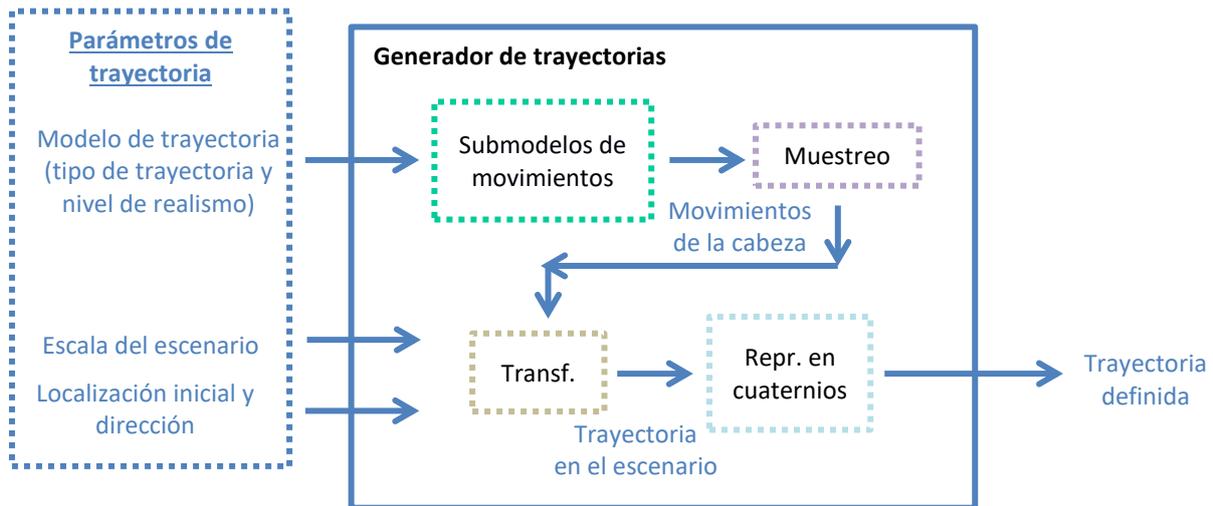


Figura 15. Diagrama de bloques del generador de trayectorias.

4.1. Sistemas de referencia del movimiento humano

Para simular el movimiento de la cámara del SAN de este trabajo se requiere conocer el movimiento de la cabeza: como el objetivo es obtener los movimientos de la cabeza dentro de un escenario, se necesitan los Sistemas de Referencia (SR) de la cabeza y del escenario. Dado que estos SR difieren de los empleados por Imai et al. (14) y Hirasaki et al. (15), se deben hallar las relaciones y transformaciones (T) entre todos los SR. Cada SR tiene un índice asociado, que se ha usado en la nomenclatura de sus transformaciones y componentes: un superíndice a la izquierda representa el SR respecto al que se localiza el otro SR, indicado como un subíndice a la derecha.

En el artículo de Imai et al. (14), la localización tridimensional de cada SR se describe mediante las componentes de traslación y rotación, estas últimas expresadas en los ángulos de Euler (*yaw* (φ), *pitch* (ϑ) y *roll* (ψ)) en secuencia 321 (ZYX). Imai et al. (14) definen el SR espacial (índice S) como un sistema fijo en el que el eje Z_s es paralelo a la vertical espacial y el eje X_s coincide con la dirección inicial encarada por la persona. Por otro lado, el SR cabeza (índice H), también empleado por Hirasaki et al. (15), es móvil y solidario a la cabeza; su eje X_H es paralelo al eje *nasooccipital* y el eje Y_H es paralelo al eje *interaural*. Al caminar, se considera que el cuerpo sigue una trayectoria en el plano X_S-Y_S , definida por un punto en este plano. Sobre este punto se sitúa el SR trayectoria (índice T), que es móvil y cuyo eje X_T (definido como *heading*) corresponde con la velocidad del cuerpo (\hat{v}_T) en cada punto de la trayectoria, y su eje Z_T es paralelo a Z_S . El ángulo de *heading* (${}^S\Phi_T$) es el comprendido entre los ejes X_T y X_S . La Figura 16 recoge la relación entre estos sistemas de referencia de Imai et al. (14). La utilización del simulador ESIM precisa de la utilización del SR escenario (índice S_c) para considerar una

localización inicial concreta de la persona dentro del escenario, que se corresponde con la localización del SR espacial respecto al SR escenario, y que se puede definir libremente. El SR escenario también contempla la escala del escenario, que en este trabajo se ha obtenido de manera aproximada al equiparar las dimensiones de los objetos simulados en el escenario con sus equivalentes reales.

Debido a que en el trabajo de Imai et al. (14) no se disponían de datos de la traslación del SR cabeza, se ha supuesto que es la misma que la del SR trayectoria, salvo en la coordenada Z , donde el primero de estos SR se sitúa a una distancia igual a la altura de la persona. De esta manera, se ha aproximado la velocidad en X e Y de la cabeza a la del cuerpo, y la posición de la cámara sobre el suelo a la de la altura de la persona. Se destaca también que la velocidad no ha sido estudiada en profundidad en el artículo de Imai et al. (14), dónde se ha reducido a una constante.

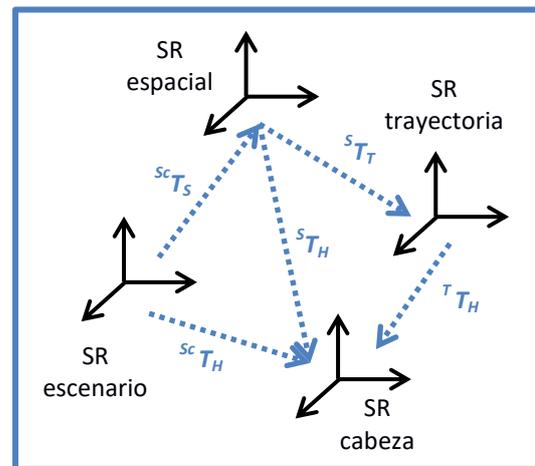
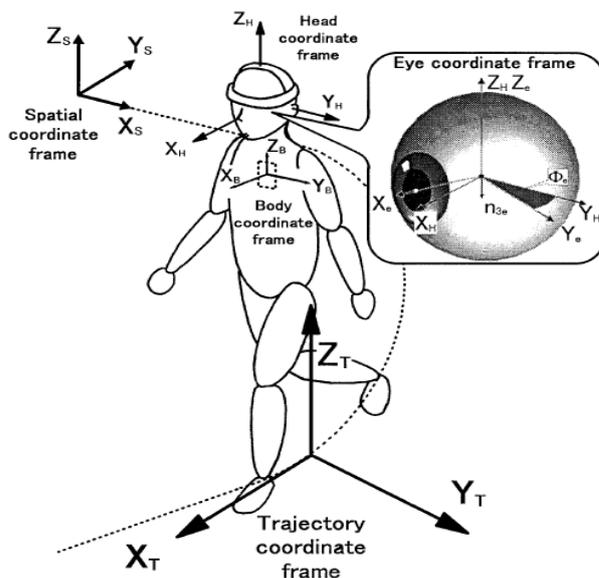


Figura 16. Izquierda: sistemas de referencia empleados por (14). Derecha: relación y transformaciones entre los sistemas de referencia definidos para este trabajo.

4.2. Modelado de los movimientos humanos

El modelado de los movimientos de la cabeza se ha obtenido a partir de los estudios de la bibliografía, cuyas mediciones han sido además contrastadas entre ellos y validadas (14), (15), (18). Los resultados de Imai et al. (14) han sido obtenidos con un sistema de análisis visual del movimiento y 5 sujetos de 29 años de edad media. Los datos se han procesado luego para obtener los modelos de sólido rígido de las distintas partes del cuerpo. Los resultados se presentan en unidades de longitud normalizadas (% *stride*) respecto a la unidad de longitud característica, que puede ser la distancia recorrida entre los instantes en que se apoya un mismo pie (zancada) o la caminata para realizar una curva, en el caso de las trayectorias rectas o giros, respectivamente (14).

En este trabajo se han definido 4 modelos para representar el comportamiento humano en su locomoción (Tabla 1), según la trayectoria (recta o giro de 90° de radio) y el nivel de realismo (bajo o alto). En cada uno de los modelos, se han realizado distintas aproximaciones (submodelos, S) a las variables medidas por Imai et al. (14) y Hirasaki et al. (15) (Tablas 1 y 2 y Figuras 17 y 18, figuras en las que se representan los submodelos que no se aproximan a constantes). Dentro de los submodelos, el S_0 es la aproximación por defecto, en la que se asigna el valor nulo a la variable en cuestión.

El Modelo 1 (M1) corresponde a la trayectoria recta de bajo realismo, en la que ${}^S\Phi_T$, ${}^S\varphi_H$, ${}^T\vartheta_H$ y ${}^T\psi_H$ son nulos (S0). La componente en el eje X de Sv_T se ha aproximado a la velocidad media en la trayectoria (S4), y Sz_H se corresponde con la altura media (S5). La trayectoria recta de alto realismo se describe mediante el Modelo 2 (M2), que respecto al M1 incluye la representación de ${}^S\Phi_T$ (S1), las oscilaciones de Sz_H (S6) y las rotaciones de ${}^S\varphi_H$, ${}^T\vartheta_H$ y ${}^T\psi_H$ (S7, S9 y S10, respectivamente). Por otro lado, el Modelo 3 (M3) se encarga de representar la trayectoria curva de bajo realismo. Las rotaciones de la cabeza ${}^T\varphi_H$, ${}^T\vartheta_H$ y ${}^T\psi_H$ son nulas (S0) y Sz_H se define con la altura media (S5), como en el M1. En este caso, se ha supuesto que el módulo de Sv_T es la velocidad media (S4'), y ${}^S\Phi_T$ se ha aproximado suponiendo que la trayectoria realizada es un arco de circunferencia (S2). El Modelo 4 (M4) corresponde a la trayectoria curva de alto realismo, en la que se han incorporado ${}^T\varphi_H$ y ${}^T\psi_H$ respecto al M3 (S8 y S11). ${}^T\vartheta_H$ se ha supuesto nulo (S0) porque su forma es compleja y se necesita una función polinómica de grado superior a 6 para representarlo, lo que se ha considerado excesivo. Para el tipo de trayectoria considerado en el M4 no se disponían datos de las oscilaciones de Sz_H , por lo que se ha vuelto a aproximar con la altura media (S5). El módulo de Sv_T también se ha aproximado a la velocidad media (S4') y ${}^S\Phi_T$ se ha modelado a partir de los datos de la bibliografía (S3), que no se corresponden exactamente con el arco de una circunferencia, utilizado en el M3.

Modelo	Trayectoria	Realismo	Submodelo					
			${}^S\Phi_T$	Sv_T	Sz_H	${}^S\varphi_H$ o ${}^T\varphi_H$	${}^T\vartheta_H$	${}^T\psi_H$
M1	Recta	Bajo	S0	S4	S5	S0	S0	S0
M2		Alto	S1	S4	S6	S7	S9	S10
M3	Curva	Bajo	S2	S4'	S5	S0	S0	S0
M4		Alto	S3	S4'	S5	S8	S0	S11

Tabla 1. Modelos y submodelos utilizados para la aproximación a los movimientos humanos.

Submodelo	Variable	Persona media (edad, altura, velocidad y zancada)	Aproximación
S1	${}^S\Phi_T$	29 años, 1,68 m, 1,6 m/s, 150 cm	Senoide (A, f, α): 6°, 1 Hz, 90°
S4	Sv_T		Constante (v_x): 1,6 m/s
S5	Sz_H		Constante (z): 1,68 m
S6	Sz_H	31 años, 1,68 m, 1,6 m/s, 150 cm	Senoide (A, f, α): 25 mm, 2 Hz, -120°
S7	${}^S\varphi_H$	29 años, 1,68 m, 1,6 m/s, 150 cm	Senoide (A, f, α): 0, 75°, 1 Hz, 160°
S9	${}^T\vartheta_H$		Senoide (A, f, α): 3°, 2 Hz, 180°
S10	${}^T\psi_H$		Senoide (A, f, α): 2°, 1 Hz, 180°
S2	${}^S\Phi_T$		29 años, 1,68 m, 1,5 m/s \approx 1,6 m/s, 150 cm
S3	${}^S\Phi_T$	Polinomio (grado): 4	
S4'	Sv_T	Constante ($ v $): 1,6 m/s	
S8	${}^T\varphi_H$	Polinomio (grado): 4	
S11	${}^T\psi_H$	Polinomio (grado): 4	

Tabla 2. Aproximación usada en cada submodelo. Se detallan también las características de la persona media de cada aproximación.

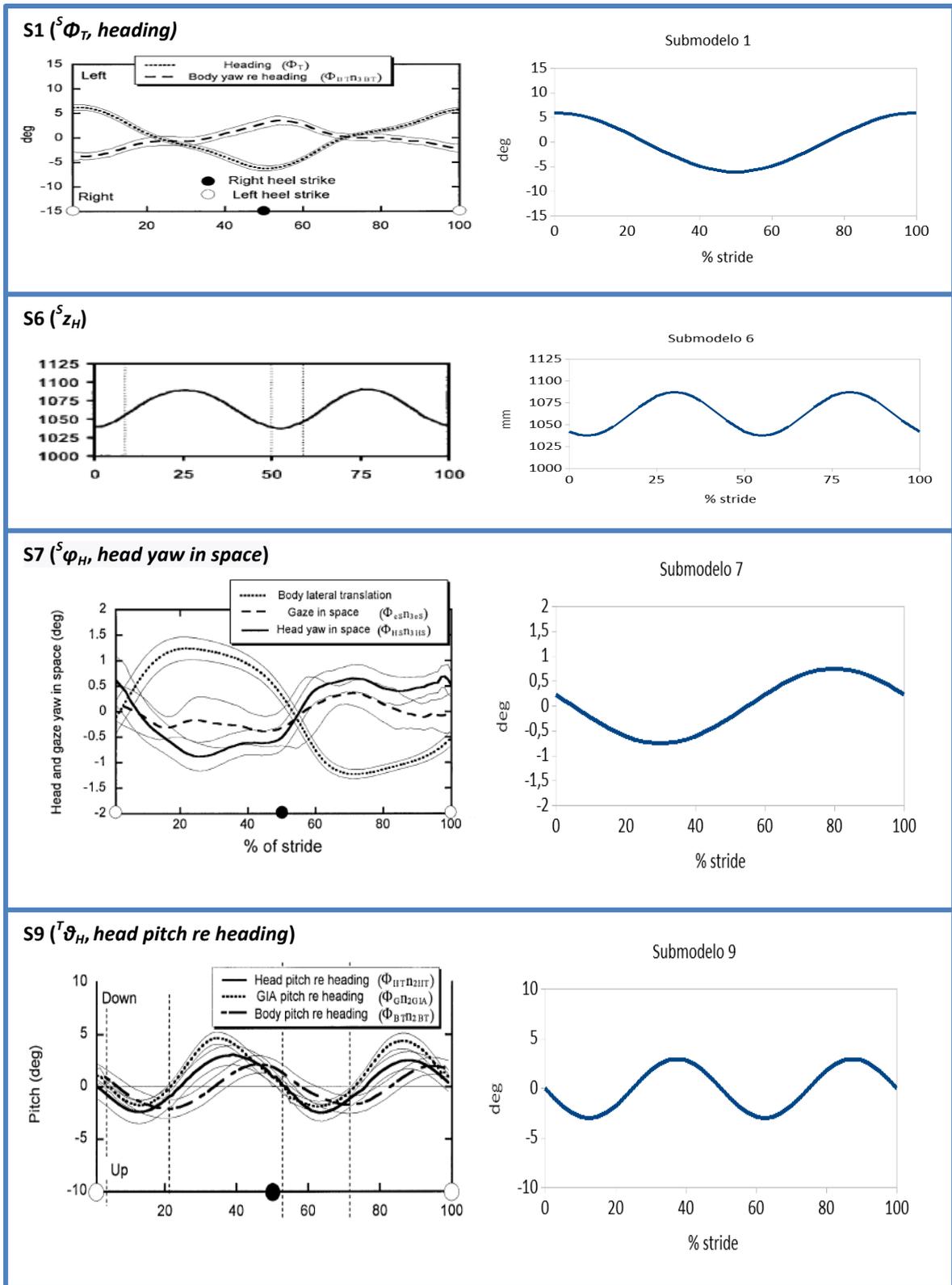


Figura 17. Comparativa entre los datos de Imai et al. (14) y Hirasaki et al. (15) con las aproximaciones realizadas para S1, S6, S7 y S9. $^S z_H$ no se corresponde con la posición absoluta de la cabeza en Z, pero sí con sus variaciones.

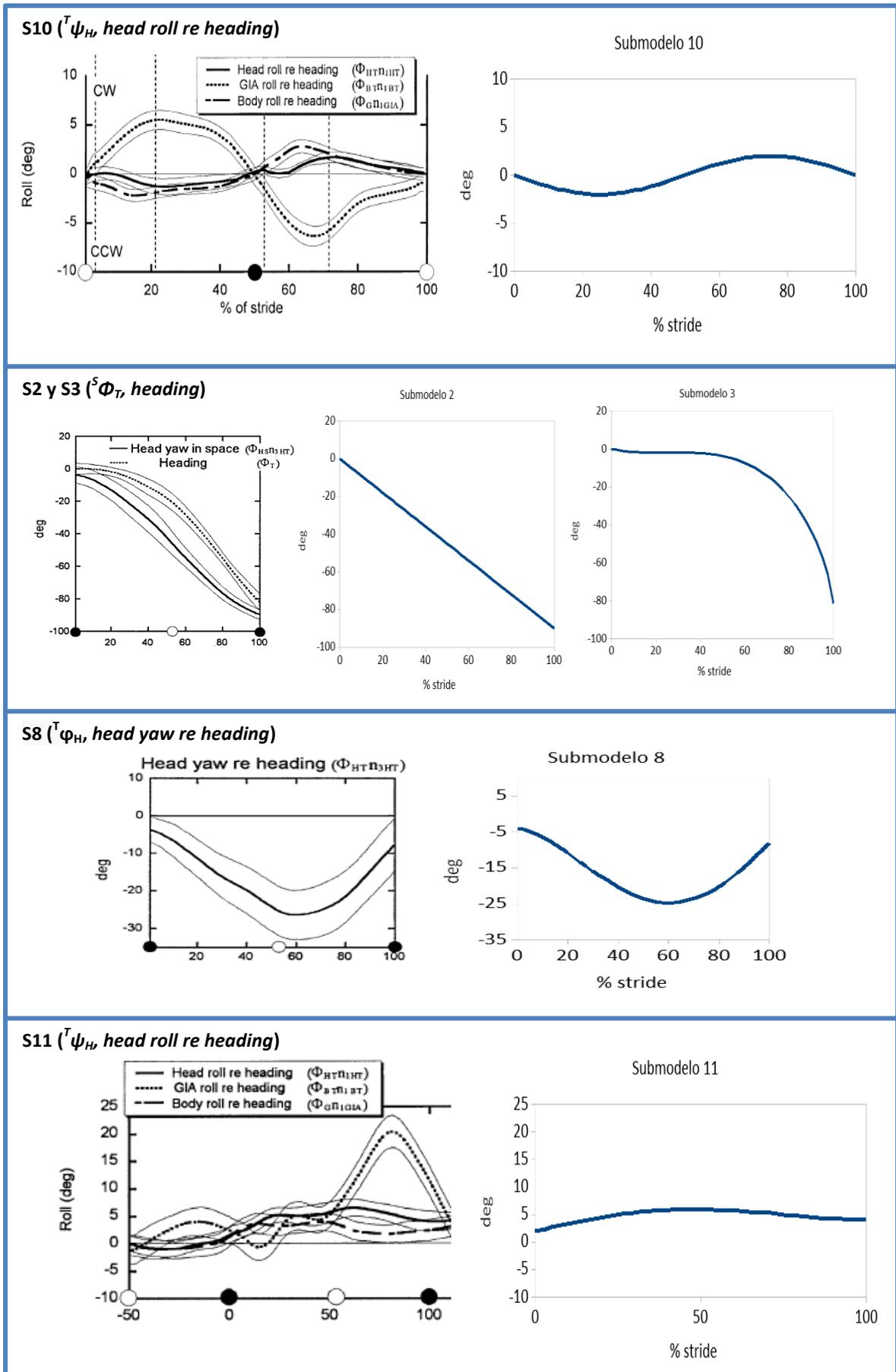


Figura 18. Comparativa entre los datos de Imai et al. (14) con las aproximaciones realizadas para S10, S2, S3, S8 y S11.

4.3. Muestreo de los modelos

La utilización del simulador ESIM precisa de la definición de movimientos discretos, por lo que se debe realizar un muestreo de los modelos obtenidos anteriormente. La frecuencia más elevada de estos movimientos es 2 Hz, característica de las senoides de ${}^S z_H$ y ${}^T \vartheta_H$ de la trayectoria recta, por lo que la tasa de muestreo debe ser suficientemente alta para representar la forma de estas señales. Por ello se ha elegido realizar 41 muestras tanto para la trayectoria recta como la curva (Figura 19), lo que supone aproximadamente unas tasas de muestreo de 40 y 75 Hz respectivamente, debido a que la longitud de la trayectoria curva es menor que la de la recta.

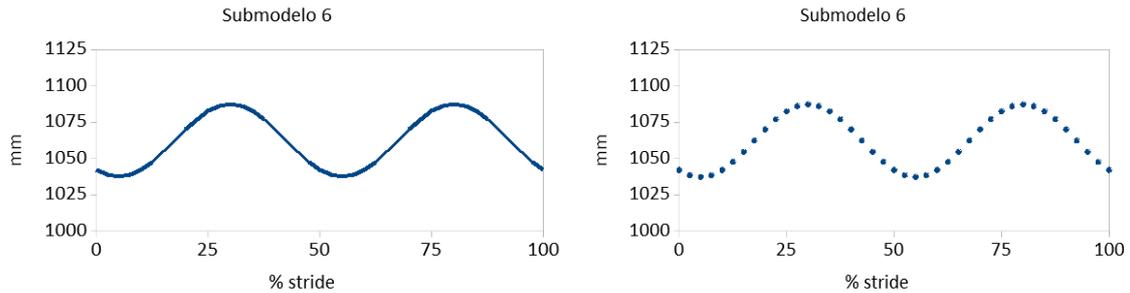


Figura 19. Comparativa de ejemplo entre el modelado y el muestreo de S6.

4.4. Transformaciones entre SR

Los movimientos de la cabeza respecto al SR escenario se han obtenido a través de un proceso en el que se han calculado los elementos de ${}^S x_T$ y ${}^S y_T$ en función de ${}^S v_T$ y ${}^S \Phi_T$, ${}^S T_H$ a partir de las transformaciones del SR trayectoria y ${}^{Sc} T_H$ mediante las transformaciones del SR espacial. La obtención de los elementos de las transformaciones se ha resumido en la Tabla 3.

T	SR base	SR terminal	Traslación			Rotación			v	s	Leyenda	
			x	y	z	φ	ϑ	ψ			Color	Obtención
${}^{Sc} T_S$	Escenario	Espacial	✓	✓	✓	✓	0	0	0	✓	Azul	Def. SR
${}^S T_T$	Espacial	Trayectoria	✓	✓	0	✓	0	0	0	1	Rojo	Biblio.
${}^T T_H$	Trayectoria	Cabeza	0	0	✓	✓	✓	✓	0	1	Verde	Cálc. X e Y
${}^S T_H$	Espacial	Cabeza	✓	✓	✓	✓	✓	✓	0	1	Amarillo	Transf. SR
Objetivo												
${}^{Sc} T_H$	Escenario	Cabeza	✓	✓	✓	✓	✓	✓	0	✓		

Tabla 3. Transformaciones y sus elementos entre los distintos sistemas de referencia. Un 0 ó 1 indica el valor numérico del elemento, mientras que ✓ indica un valor distinto de 0 y 1. El método de obtención de los componentes se indica a través del color de sus celdas: azul si se ha obtenido con la definición de los SR, rojo para los submodelos extraídos de la bibliografía, verde en el cálculo de la posición en X e Y según la velocidad y heading, y amarillo para los calculados a través de las transformaciones entre los SR.

El SR trayectoria se mueve con el cuerpo de la persona, variando su posición respecto al SR Espacial en los ejes X e Y. Sin embargo, estas posiciones no han sido medidas explícitamente por Imai et al. (14), por lo que en este trabajo se han calculado a partir de los submodelos de heading y velocidad extraídos de Imai et al. (14), utilizando su relación geométrica e integrando la velocidad (Figura 20). El muestreo de los modelos de movimiento

hace que las integrales se tengan que aproximar en la mayoría de casos, para lo que se ha elegido usar el método trapezoidal.

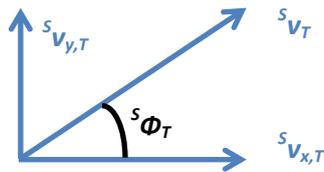


Figura 20. Relación geométrica entre la velocidad y heading del SR trayectoria respecto al SR espacial.

Para alternar entre varios SR, se emplean sus transformaciones, que en este caso son transformaciones proyectivas, definidas como aplicaciones biyectivas de P^3 en P^3 , compuestas por la matriz de rotación (R), la matriz de traslación (t), la matriz de perspectiva (v) y el escalado (s). En los SR elegidos, v será nula y s valdrá 1, salvo para la transformación entre el SR escenario y el SR espacio. La utilización de los ángulos de Euler tiene dos casos degenerados cuando ϑ se acerca a 90° o -90° (19), pero en el estudio de los movimientos humanos durante la acción de caminar, estas situaciones no se alcanzan, por lo que no se han considerado.

A partir del desarrollo y cálculo matemático de las transformaciones para los SR, se destacan dos aspectos. El primero de ellos es que a parte de su relación con ${}^S v_T$ y ${}^S \Phi_T$, ${}^{Sc} x_H$ y ${}^{Sc} y_H$ también dependen de la localización inicial y dirección del movimiento en el escenario. Por otro lado, la utilización de los ángulos de Euler en secuencia 321, la rotación del SR cabeza respecto al SR escenario es equivalente a la realización de tres giros. El primero de ellos es la suma de las componentes φ del SR cabeza respecto al SR espacial y del SR espacial respecto al SR escenario. Los otros dos giros se corresponden con los valores de las componentes ϑ y ψ para el SR cabeza respecto al SR trayectoria.

A través de la evaluación de ESIM y las diferencias entre trayectorias definidas y ajustadas, de las que se ocupa el capítulo 5, se ha observado que el simulador ESIM utiliza un SR diferente según la componente de la localización considerada. En particular, las componentes de traslación son respecto al SR escenario, pero las componentes de rotación se expresan respecto a otro SR, el SR rotación, obtenido al girar el SR escenario -90° en el eje Z. Para tener en consideración este nuevo SR, gracias a la representación de la rotación mediante ángulos de Euler en secuencia 321, se han sumado 90° adicionales en la componente de φ calculada para el SR cabeza respecto al SR escenario.

4.5. Cuaternios y trayectoria definida

La trayectoria definida se introduce en ESIM utilizando la representación en cuaternios para los ángulos de Euler en secuencia 321 (19), (20), y con un fichero .csv creado con el programa Calc de LibreOffice, que almacena los elementos de ${}^{Sc} T_H$, cuyas componentes de traslación se introducen en unidades de longitud del escenario.

5. Evaluación y ajuste del simulador ESIM

Con la obtención de los movimientos de la cabeza, que se corresponden con las trayectorias de la cámara de eventos, ya se pueden realizar simulaciones con ESIM. El funcionamiento de este simulador ha sido comprobado a través del Estudio de los Eventos (EE) y el Estudio de las Trayectorias (ET). Estos estudios se han elaborado con hojas de cálculo del programa Calc de Libreoffice, y la automatización de la recogida de datos y su manejo se ha realizado mediante dos programas de Python 3 desarrollados en este trabajo, el Programa de Trayectorias (PT) y el Programa de Partición (PP).

5.1. Estudio de los Eventos, EE

El EE se compone de dos subestudios. En el primero, EE1, no se ha utilizado la estrategia de partición para corregir la trayectoria ajustada de los experimentos, mientras que en el EE2 sí, lo que modifica el número de eventos generados. Esta estrategia se describe más en detalle en el apartado 5.2.

El simulador ESIM permite calcular la generación de eventos de forma precisa, obteniendo el instante de generación, la polaridad y la localización del evento. No obstante, la representación de estas características es compleja debido al elevado número de eventos generados. Por ello, ESIM proporciona además una representación mucho más intuitiva y sencilla de los eventos, aunque aproximada, a través de las imágenes de eventos, en las que se acumulan los eventos relacionadas con las imágenes que obtiene una cámara convencional. Para estimar con precisión la generación de los eventos, el simulador almacena más imágenes de la cámara convencional cuando hay más movimiento. De esta manera, un mayor número de imágenes de eventos se relaciona con una mayor generación de eventos debido al aumento del movimiento relativo.

5.1.1. Subestudio EE1

El EE1 se ha basado en la medición del número de las imágenes de eventos generadas en 16 experimentos (Tabla 4), entre los que se han comparado sus resultados (Tabla 5). Debido a que las trayectorias de la recta y curva se realizan en un tiempo distinto, se calcula el número de imágenes de eventos por segundo (fps) para relacionar sus datos. En estos experimentos se han comprobado los efectos de una variación en la localización (visión de distintos objetos del escenario), realismo (uso de modelos de alto o bajo realismo), trayectoria (recta o curva) y escenario (salón o ciudad). En cada una de las comparativas de la variación de la localización y realismo, se han estudiado dos experimentos, uno de los cuales tenía la localización 1 (o realismo alto) y el otro la 2 (o bajo). En el caso de las comparativas de la variación de la trayectoria o escenario, al modificar estos aspectos también se altera la localización, por lo que en estas comparativas se utilizan cuatro experimentos cotejados en parejas: los que tienen trayectoria recta (o escenario del salón) con los de la curva (o ciudad), independientemente de su localización y promediando los fps de cada una de las parejas. En la Figura 21 se representa la variación de los fps para cada una de las comparativas consideradas. Los resultados se expresan como variación porcentual de los fps respecto al primer o dos primeros experimentos de la comparativa, según se hayan utilizado dos o cuatro experimentos en ella.

Estudio	Experimento	Escenario	Modelo (trayectoria, realismo)	Tiempo (s)	Localización	fps
EE1	1	Salón	M1	2,81	1	298
	2		(recta, bajo)		2	301
	3		M2		1	310
	4		(recta, alto)		2	312
	5		M3	0,52	1	1842
	6		(curva, bajo)		2	1831
	7		M4		1	1594
	8		(curva, alto)		2	1585
	9	Ciudad	M1	2,81	1	498
	10		(recta, bajo)		2	484
	11		M2		1	505
	12		(recta, alto)		2	485
	13		M3	0,52	1	1875
	14		(curva, bajo)		2	1808
	15		M4		1	1602
	16		(curva, alto)		2	1562

Tabla 4. Experimentos, sus características y fps para el EE1.

Estudio	EE1							
Variación	Localización							
Comparativa	1	2	3	4	5	6	7	8
Experimentos	1, 2	3, 4	5, 6	7, 8	9, 10	11, 12	13, 14	15, 16
Variación	Realismo							
Comparativa	1	2	3	4	5	6	7	8
Experimentos	1, 3	2, 4	5, 7	6, 8	9, 11	10, 12	13, 15	14, 16
Variación	Trayectoria							
Comparativa	1		2		3		4	
Experimentos	1, 2, 5, 6		3, 4, 7, 8		9, 10, 13, 14		11, 12, 15, 16	
Variación	Escenario							
Comparativa	1		2		3		4	
Experimentos	1, 2, 9, 10		3, 4, 11, 12		5, 6, 13, 14		7, 8, 15, 16	

Tabla 5. Relación de experimentos para las comparativas del EE1.

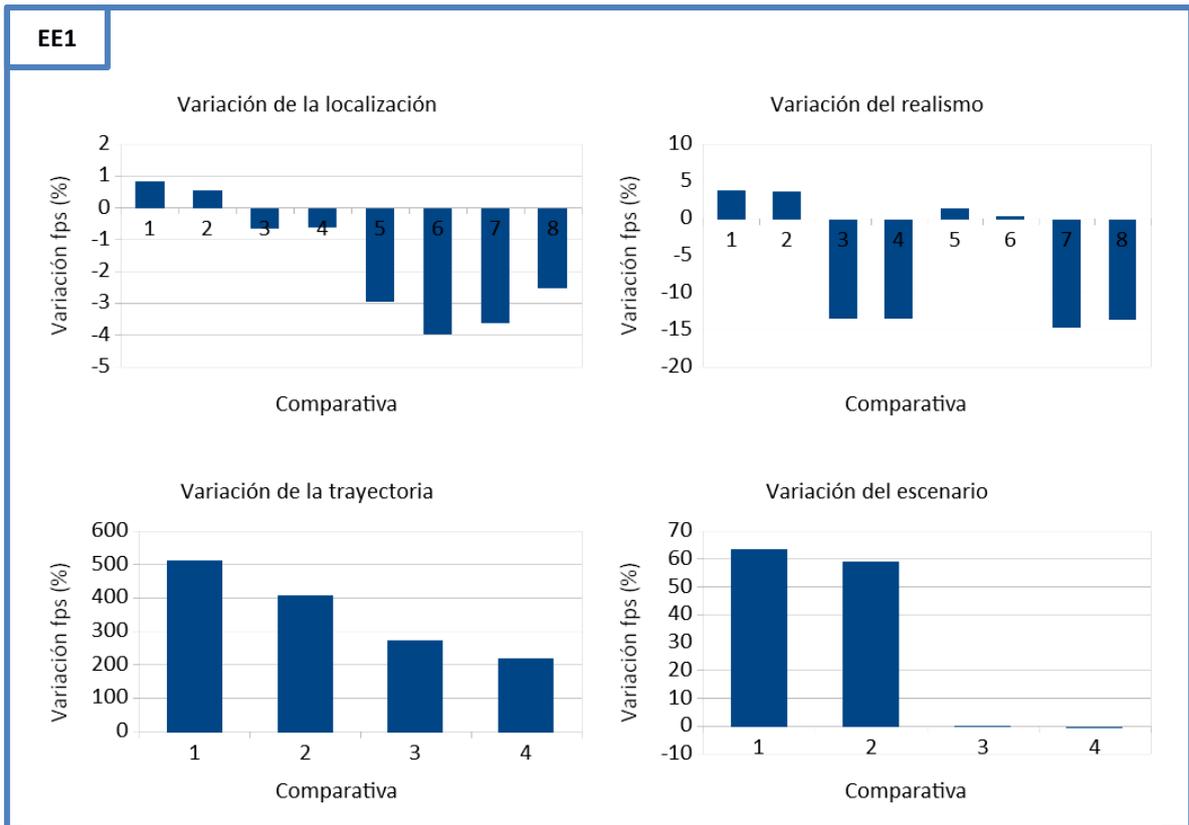


Figura 21. Gráficas de las comparativas entre los 16 experimentos del EE.

La modificación de la localización no afecta significativamente a la generación de eventos, con variaciones de los fps inferiores al 5%. El aumento del realismo en las trayectorias rectas produce un cambio similar, pero se observa sin embargo una reducción del 13% en las trayectorias curvas; este hecho se debe a dos razones. La primera de ellas es que al realizar giros de 90° la cabeza sólo recorre 85°, hecho considerado en la trayectoria con realismo alto pero no en la de bajo, por lo que se generaría un menor número de eventos en la trayectoria de alto realismo. La otra causa de este descenso es que la trayectoria definida difiere de la trayectoria ajustada para estos casos, lo que se ha estudiado en el apartado 5.1.2. y 5.2.

Se aprecia que ante una trayectoria curva en vez de recta, se generan de 2 a 5 veces más eventos; en concreto, de 4 a 5 veces más para los escenarios del salón y de 2 a 3 para los de la ciudad. Con este estudio se denota que la rotación en el eje Z, movimiento característico de la trayectoria curva, se consigue una mayor generación de eventos, incrementando en gran medida el potencial de la cámara de eventos para las tareas de inspección, reconocimiento y localización. En último lugar, se ha observado que la generación de eventos es un 60% mayor en trayectorias rectas ante el escenario de ciudad en lugar del de salón, mientras que en las curvas no se aprecia ese aumento. Este hecho puede deberse a que la generación de eventos ya es todo lo elevada posible en las trayectorias curvas, con un número de eventos 2 o 5 veces mayor.

A la vista de los resultados, se han extraído varias conclusiones. La primera de ellas es que la consideración de distintas localizaciones dentro de un entorno no altera significativamente el número de fps. Por otro lado, el aumento del realismo en las trayectorias curvas produce en torno a un 15% menos de fps, debido a que la rotación de la cabeza se reduce un 6%. Donde sí se experimenta una gran mejoría es con el uso de las trayectorias

curvas frente a las rectas, de 2 a 5 veces más eficaces. Sin embargo, este aumento en el número de fps puede hacer que éstos se saturen en las trayectorias curvas, ya que la utilización de un escenario más rico en objetos y texturas, como es la ciudad frente al salón, no supone un incremento de fps en esas trayectorias.

5.1.2. Subestudio EE2

El uso de la partición consigue replicar con mayor precisión las trayectorias humanas en los modelos de realismo alto, lo que repercute en un mayor movimiento de la cámara y un incremento en el número de eventos generados. Se han realizado 4 experimentos de alto realismo y con partición (Tabla 6) para compararlos respecto a los experimentos con el mismo nivel de realismo, tipo de trayectoria, escenario y localización del EE1, en la denominada *variación de la partición*. Además, los experimentos de alto realismo y con partición del EE2 se han cotejado con los experimentos de bajo realismo del EE1 con los que comparten tipo de trayectoria, escenario y localización, a lo que se referirá como *variación del realismo con partición* (Tabla 7 y Figura 22).

Al introducirse la partición, se ha obtenido un aumento de los fps del 10 al 20%, salvo en el caso de la trayectoria recta en ciudad, donde se han reducido un 16%. Este hecho se debe a que las trayectorias rectas del EE1 recorren una distancia tres veces superior a las del EE2, que en el escenario de la ciudad ha supuesto el acercamiento a un árbol con hojas, lo que ha disparado la generación de eventos en los últimos tramos de la trayectoria. El aumento del nivel de realismo junto al uso de la partición ha limitado el descenso de los fps en las trayectorias curvas, que ha pasado del 15% al 6%, además de elevar el incremento de los fps para la trayectoria recta del salón, de un 5% a un 25%. Por otro lado, en la trayectoria recta y escenario de la ciudad, el aumento del nivel de realismo y uso de la partición ha supuesto un descenso del 15% de los fps, mientras que si no se usa partición se tiene un incremento del 1%. Es en este último caso donde se llegaba a la posición de un árbol con hojas debido a la diferencia de longitudes entre las trayectorias de los experimentos, por lo que en este caso los resultados no son concluyentes.

Estudio	Exp.	Esc.	Modelo (trayectoria, realismo)	Tiempo (s)	Localización	Partición	fps
EE2	1	Salón	M2 (recta, alto)	0,94	1	✓	371
	2		M4 (curva, alto)		2		1731
	3	Ciudad	M2 (recta, alto)	0,52	1		423
	4		M4 (curva, alto)				1763

Tabla 6. Experimentos, sus características y fps para el EE2.

Estudio	EE1 y EE2			
Variación	Partición			
Comparativa	1	2	3	4
Experimentos	3 (EE1), 1 (EE2)	8 (EE1), 2 (EE2)	11 (EE1), 3 (EE2)	15 (EE1), 4 (EE2)
Variación	Realismo con partición			
Comparativa	1	2	3	4
Experimentos	1 (EE1), 1 (EE2)	6 (EE1), 2 (EE2)	9 (EE1), 3 (EE2)	13 (EE1), 4 (EE2)

Tabla 7. Relación de experimentos para las comparativas del EE1 con EE2.

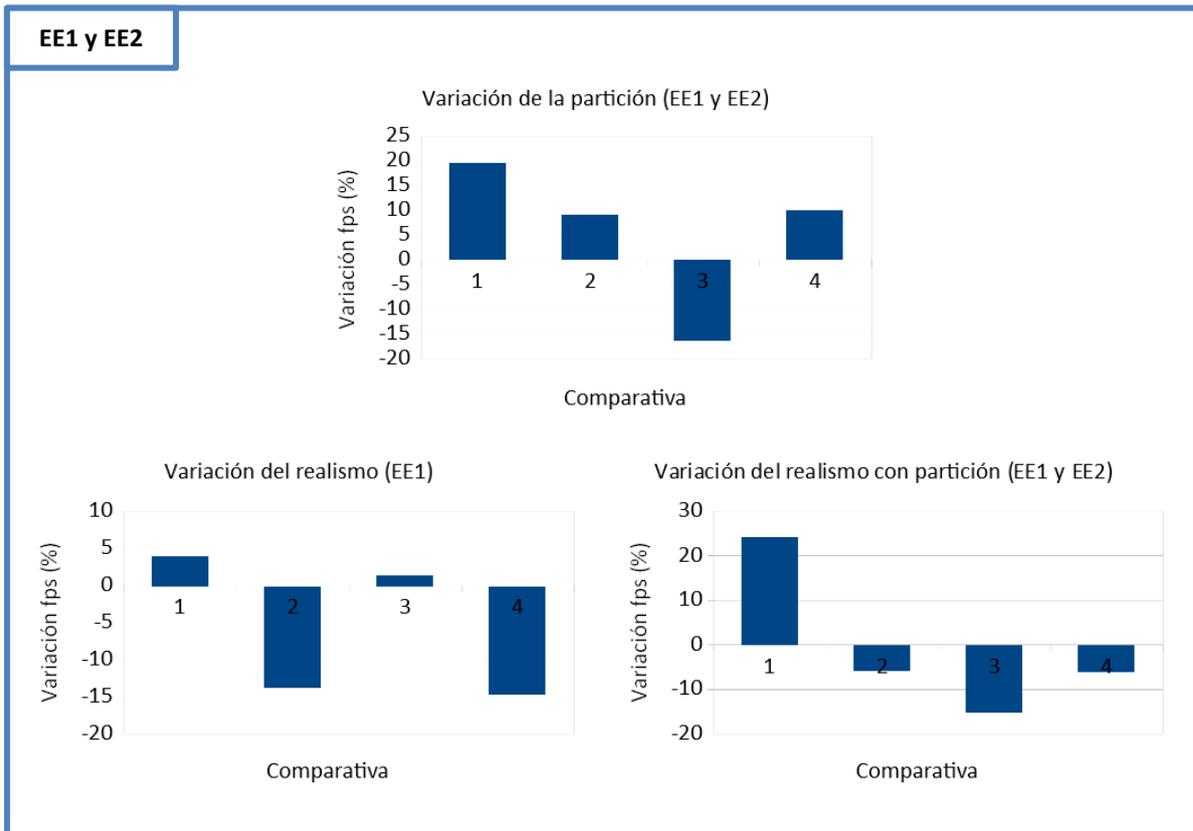


Figura 22. Gráficas de las comparativas entre los experimentos del EE1 y EE2 correspondientes. Se muestra la variación del realismo del EE1 para compararla con la variación del realismo con partición del EE1 y EE2.

5.2. Estudio de la Trayectoria, ET

El ET se ha dividido en tres subestudios con distinto nivel de detalle: se han realizado dos preliminares (ET1 y ET2) y uno con mayor precisión (ET3).

5.2.1. Subestudios preliminares, ET1 y ET2

El primero de ellos es ET1, y es un estudio preliminar que parte de los 16 experimentos de EE1 (Tabla 4). En cada uno de esos experimentos se ha medido la localización de la cámara para un instante de tiempo, que se ha comparado respecto a la localización definida en aproximadamente el mismo instante ($\pm 1\%$, Figura 23). Se ha observado que las variaciones en y y z (menores al 5%) son tolerables, lo que no ocurre en las componentes de x

(superiores al 20%), *yaw*, *pitch* y *roll* (del 0,5 a 2°, que suponen del 15 al 65% de la amplitud de oscilación de sus submodelos).

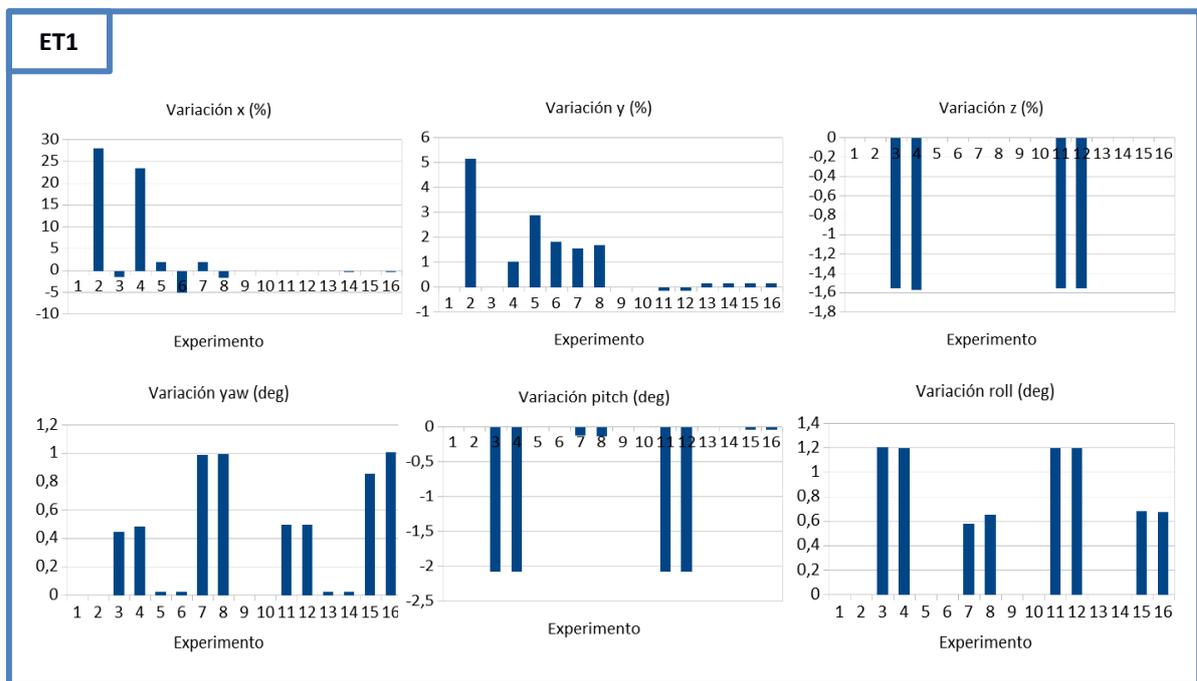


Figura 23. Variación de los componentes de la localización para los distintos experimentos del ET1.

Para profundizar en los resultados del ET1 se ha realizado el subestudio ET2. En este caso se han comparado las 41 localizaciones que describían la trayectoria definida del experimento 4 del ET1, con las correspondientes de la trayectoria ajustada (Figura 24). Se ha apreciado que en el experimento considerado al menos, el simulador ESIM no ajusta con precisión los movimientos con oscilaciones, de manera que los movimientos humanos no se simulan correctamente. Por lo tanto, se precisa de un análisis más extenso para comprobar su comportamiento en el resto de situaciones, así como de la corrección de las diferencias entre las trayectorias definidas y ajustadas, mediante la modificación de distintos parámetros del simulador o el desarrollo de soluciones alternativas.

5.2.2. Subestudio en profundidad, ET3

La necesidad de profundizar en la comparativa entre trayectorias hace que se requiera automatizar el manejo y el procesamiento de un volumen de datos mucho mayor que el de los experimentos ET1 y ET2, para lo que se ha desarrollado en Python 3 un Programa de Trayectorias (PT). En el proceso de desarrollo del PT, se ha realizado el tutorial de Python 3 de Pycharm y se ha consultado la documentación oficial del lenguaje de programación (21).

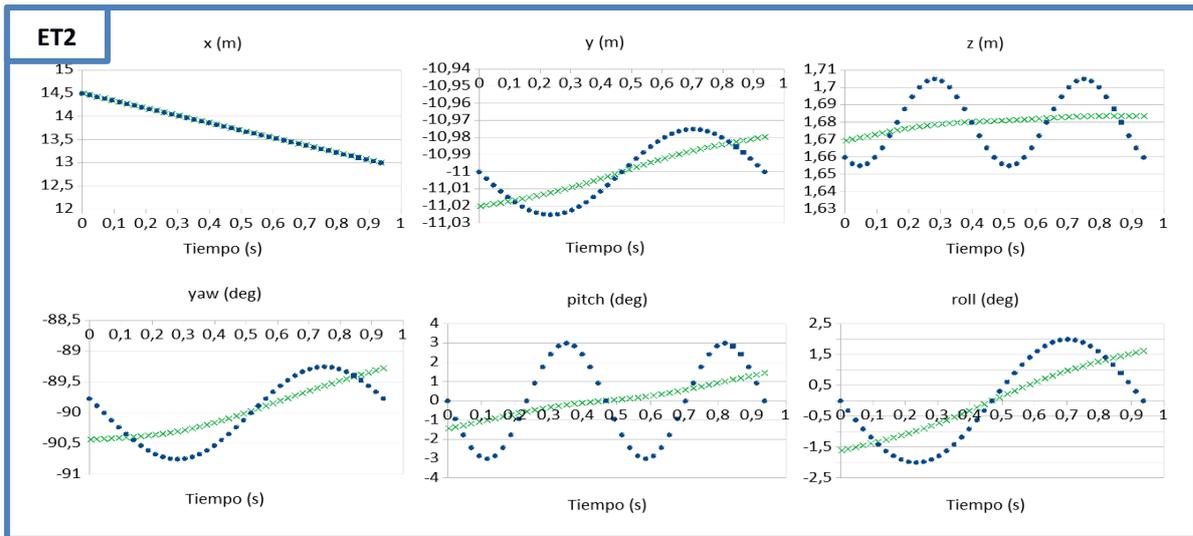


Figura 24. Comparativa entre las componentes de la localización de la trayectoria definida (círculos azules) y ajustada (cruces verdes) del ET2.

Programa de Trayectorias, PT

El simulador ESIM permite exportar y guardar como fichero de texto los datos de sus trayectorias ajustadas, a través de la herramienta ROS, del que se extraen distintos valores numéricos a través del PT desarrollado. El PT utiliza los módulos de *re* y *math*. Las funciones del PT son:

- Detectar la información relevante
- Comprimir el volumen de datos
- Adecuar la representación de la localización
- Generar un fichero de fácil manejo

El PT es capaz de detectar las distintas componentes de la localización y almacenarlas temporalmente para su procesamiento, gracias al uso del módulo *re* de expresiones regulares. Los valores numéricos que definen la localización de la cámara se encuentran en las líneas del fichero de texto que tienen una de las siguientes cadenas de caracteres: *secs*, *x*, *y*, *z* o *w*. Cuando una de estas líneas es detectada, las cadenas de caracteres de espacios en blanco (*"\s+"*) se sustituyen por las vacías (*""*) mediante *re.sub*. Después, la línea resultante se divide según la ocurrencia del carácter *:* con *re.split*, obteniéndose una lista de dos cadenas de caracteres: la primera de ellas es la designación de las componentes de localización y la segunda sus valores numéricos. De esta manera, al seleccionar el segundo elemento de la lista se obtiene la cadena de caracteres del valor de la variable considerada, cuyo valor numérico se obtiene con *float* y se almacena en la Lista de las Localizaciones (LL). Dado que siempre se guardan en orden nueve valores por localización, se puede conocer la variable a que corresponde cada valor según su posición en la LL.

ESIM describe las trayectorias ajustadas con un número de localizaciones unas 1000 veces mayor al de las trayectorias definidas, debido a su muestreo adaptativo del movimiento. Sin embargo, para comparar ambas trayectorias sólo se necesita que la trayectoria ajustada tenga un número de localizaciones similar a la trayectoria definida. Por ello, los datos de las localizaciones de la trayectoria ajustada se han comprimido para facilitar su manejo. La compresión de los datos se ha conseguido al dividir la trayectoria ajustada en un número de intervalos temporales igual al del número de localizaciones de la trayectoria definida. La LL es

recorrida con un cursor que suma los dos primeros componentes (*secs* y *nsecs*) de cada subconjunto de nueve para obtener el tiempo correspondiente a esa localización. Si ese tiempo está comprendido en el intervalo temporal actual, los otros componentes de su subconjunto se acumulan y se actualiza el número de subconjuntos que se han acumulado. Si el tiempo de la localización excede al actual intervalo temporal, las componentes acumuladas se dividen por el número de subconjuntos acumulados (se promedian), y su tiempo se aproxima a la mitad del intervalo temporal. Después, las componentes promediadas se escriben en un nuevo fichero, se calcula el nuevo intervalo temporal y las variables acumuladas se inicializan con los valores de la nueva componente.

Antes de acumular las componentes de cada localización, las de traslación se obtienen en dimensiones reales y las de rotación se representan con ángulos de Euler en secuencia 321 (19), (20). El fichero de texto resultante contiene una localización por línea, en la que las componentes están separadas por un espacio, lo que permite su selección y copiado a una hoja de cálculo para su comparativa posterior.

Estrategia de partición de los experimentos

Ante los problemas de ESIM para ajustar movimientos oscilatorios, en este proyecto se ha propuesto una alternativa adicional a la variación de sus parámetros, la *partición*. Esta alternativa consiste en dividir un experimento en múltiples subexperimentos, de forma que en cada uno de ellos, las componentes de localización sean funciones rectas en el tiempo. De esta forma, las trayectorias definidas que se introducen en ESIM son rectas en lugar de senoides o curvas polinómicas de grado dos o superior. Al realizar la partición, el muestreo adaptativo de ESIM pierde eficiencia en los instantes de inicialización de cada subexperimento, debido a que se desconocen los movimientos previos de la cámara. Cabe destacar que al describir las trayectorias con rectas, el número de localizaciones utilizadas para definir cada movimiento es de dos, en contraposición a las 40 localizaciones empleadas anteriormente, por lo que la comparativa entre trayectorias definidas y ajustadas tiene también un menor volumen de datos. El número de subexperimentos en el que hay que dividir cada experimento viene dado por la frecuencia máxima de sus movimientos, que en los casos considerados en este trabajo es de aproximadamente 2 Hz. Para reducir los subexperimentos a realizar, se ha reducido la duración del experimento en el caso de la trayectoria recta a 1 segundo, ya que durante ese periodo se puede observar como mínimo un ciclo de cada movimiento oscilatorio. Se ha elegido realizar 8 subexperimentos por experimento, de manera que las senoidales de mayor frecuencia se aproximen a señales triangulares si su desfase es nulo.

Programa de Partición, PP

Al realizar la partición de experimentos, el simulador ESIM genera un fichero de texto y conjunto de imágenes de la trayectoria de la cámara para cada uno de los 8 subexperimentos. Dentro de un subexperimento, cada una de sus imágenes está ordenada según el número que aparece en su nombre, por ejemplo *frame1654.png*, y también se conoce qué subexperimento es anterior o posterior a otro. Sin embargo, al realizarse varios subexperimentos, los nombres de las imágenes se repiten, lo que impide la organización del conjunto total de imágenes de los 8 subexperimentos. Como ejemplo, la última imagen del primer subexperimento puede llamarse *frame0391.png*, mientras que la primera imagen del siguiente subexperimento tiene el nombre *frame0000.png*, en vez de *frame0392.png*. Para solucionar este problema, se ha vuelto a desarrollar un programa específico para este trabajo, el Programa de Partición (PP),

que emplea los módulos de *os* y *math* y los paquetes de OpenCV y SciPy. Este programa tiene las siguientes funciones:

- Leer ordenadamente las imágenes
- Asignar nombres unívocos
- Almacenar los resultados en un único directorio

El PP recorre cada uno de los ficheros de los subexperimentos, de los que extrae su lista de archivos mediante *os.listdir*. Dentro de un subexperimento, el PP utiliza el Cursor de Subexperimento (CP) para leer de forma ordenada todas las imágenes, a las que asigna un nombre según el Cursor Global (CG) y las almacena en un nuevo directorio mediante *scipy.misc.imsave*. Para alternar entre el valor numérico y la cadena de caracteres de los CP y CG es necesario descomponer los números en millares, centenas, decenas y unidades, por lo que se han realizado operaciones de división y se han calculado los restos obtenidos con los divisores de 1000, 100 y 10, que además han sido truncados usando *math.trunc*.

Resultados del ET3

Se han realizado 11 experimentos (Tabla 8 y Figuras 25 a 28) para comprobar la correlación entre las trayectorias definidas y las ajustadas con distintos modelos de movimiento, considerando la modificación de los parámetros del simulador. Se ha estudiado también el comportamiento del simulador ante la partición y el uso de trayectorias de menor duración, lo que implica la realización de trayectorias con formas más rectas o cuadráticas en vez de oscilatorias. Independientemente del escenario y la localización elegidos, las funciones que describen los movimientos tienen la misma forma, aunque sus valores absolutos pueden variar debido a la escala del escenario o el uso de una localización inicial diferente. Por lo tanto, no se han considerado las modificaciones del escenario y la localización en los experimentos; todos los experimentos de esta sección se han realizado en el escenario de la ciudad y en la localización 1.

Estudio	Esc.	Exp.	Modelo (trayectoria, realismo)	Tiempo (s)	Loc.	Grado splines	Nº splines	Partición
ET3	Ciudad	1	M1 (recta, bajo)	2,81	1	5	200	-
		2	M2 (recta, alto)				800	
		3					200	
		4		8				
		5		200				
		6	0,94	200			✓	
		7	M3 (curva, bajo)	0,52		5	200	-
		8	M4 (curva, alto)				800	
		9					3	
		10	200			✓		
		11	5					

Tabla 8. Experimentos para ET3.

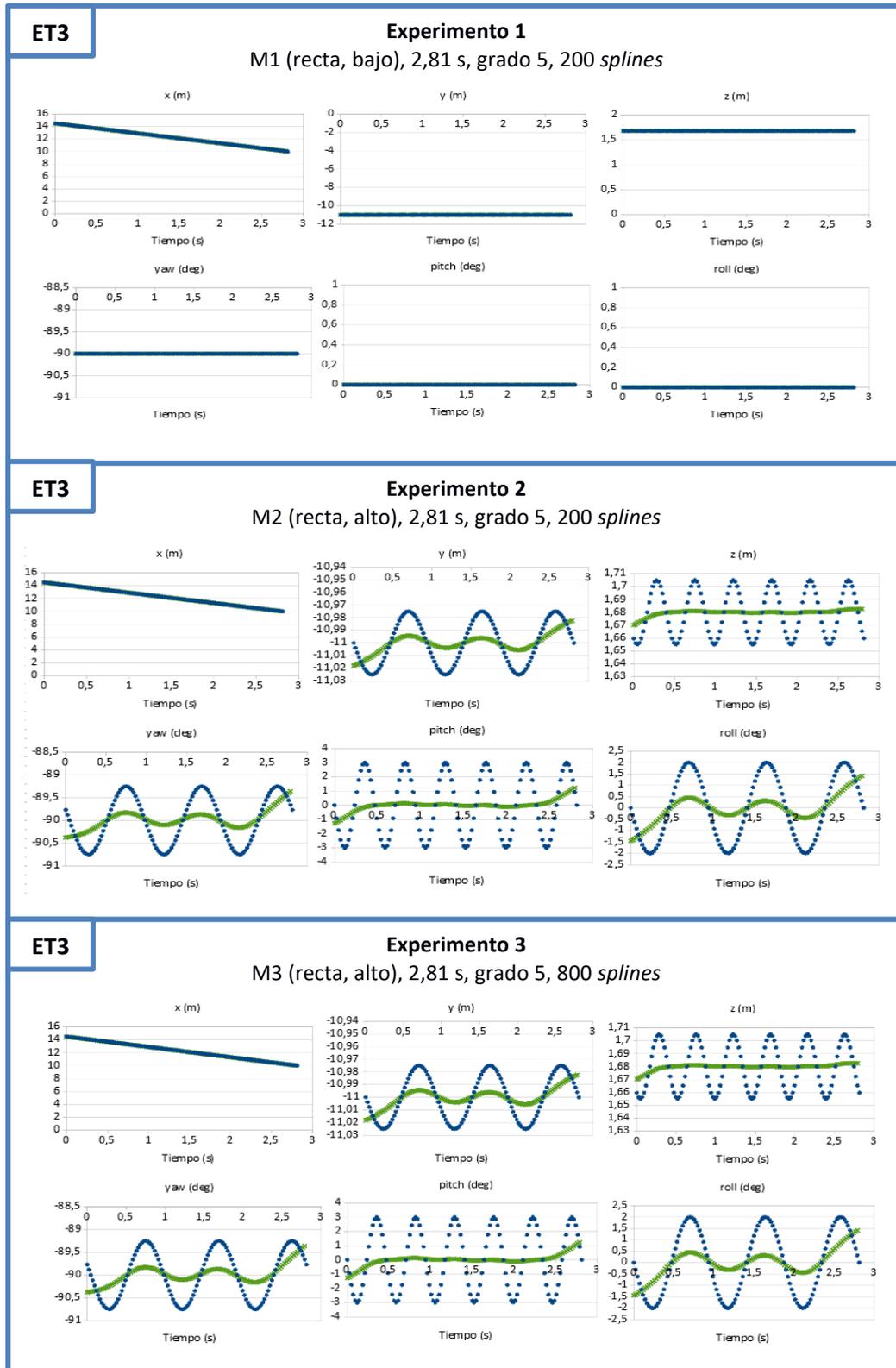


Figura 25. Comparativa entre la trayectoria definida (círculos azules) y la ajustada (cruces verdes) de los experimentos 1, 2 y 3 del ET3.

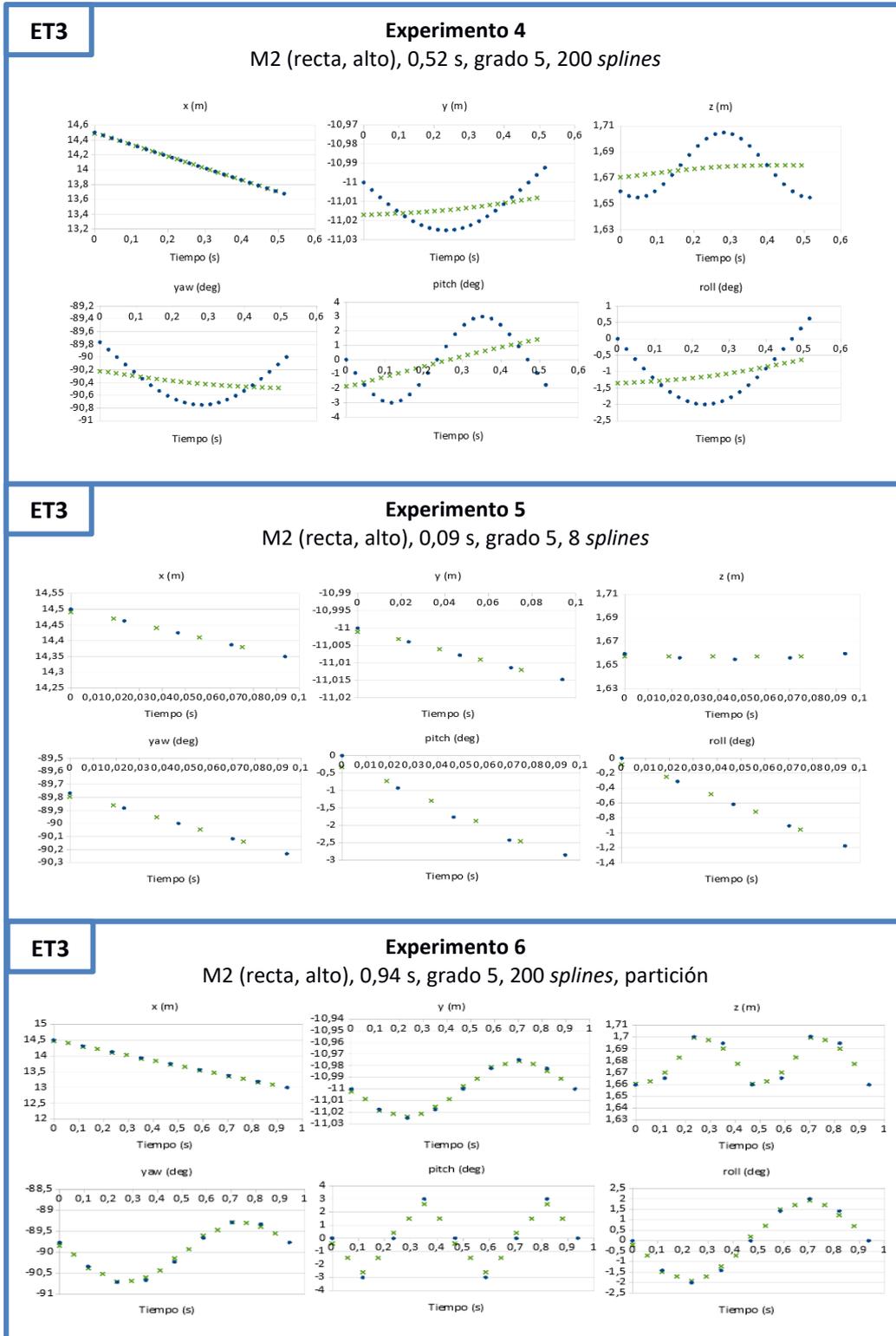


Figura 26. Comparativa entre la trayectoria definida (círculos azules) y la ajustada (cruces verdes) de los experimentos 4, 5 y 6 del ET3.

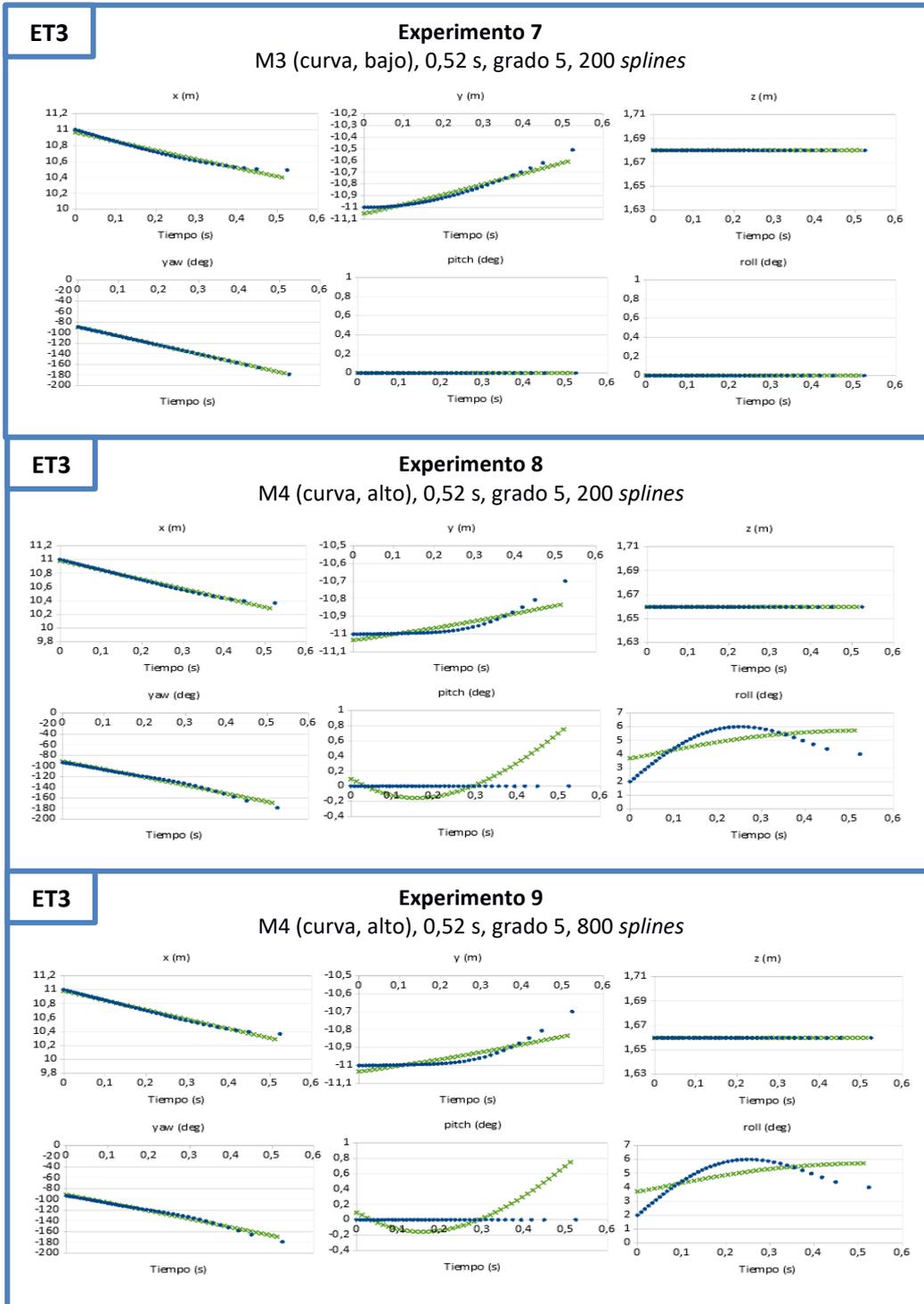


Figura 27. Comparativa entre la trayectoria definida (círculos azules) y la ajustada (cruces verdes) de los experimentos 7, 8 y 9 del ET3.

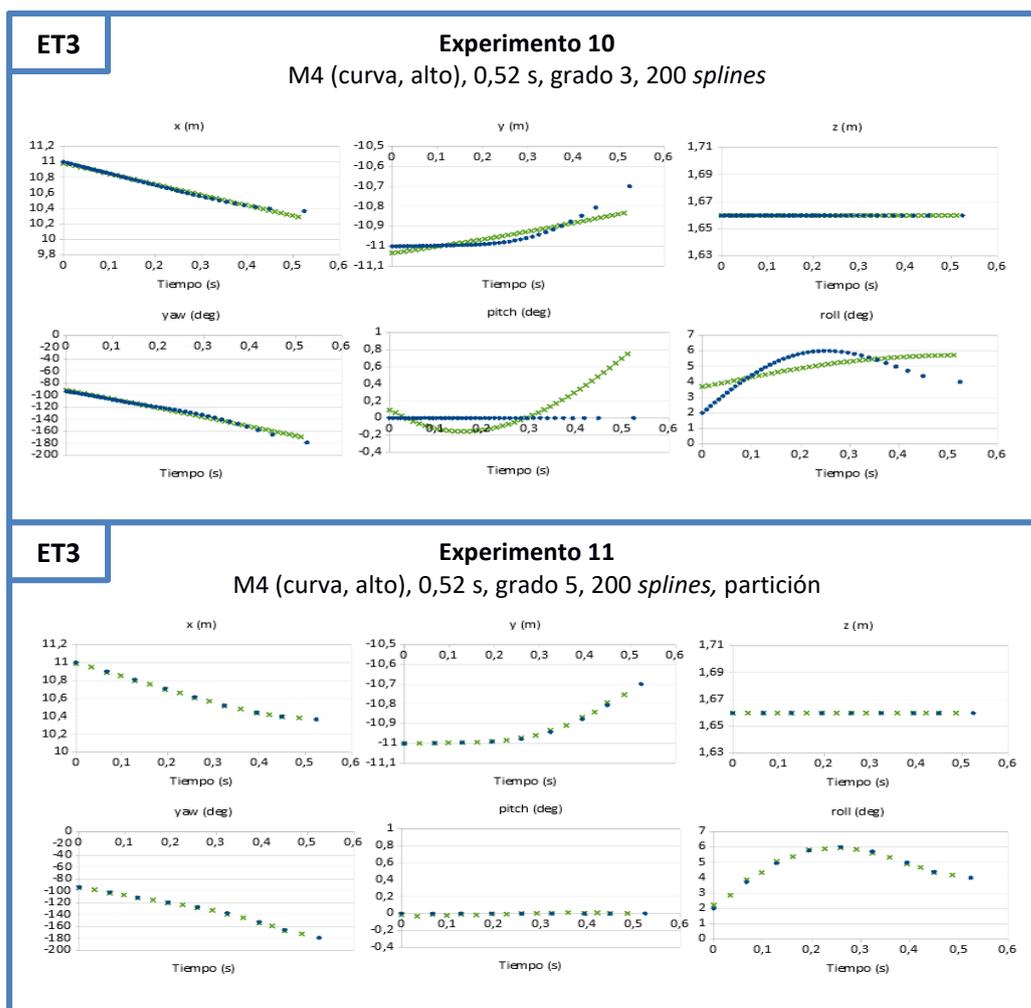


Figura 28. Comparativa entre la trayectoria definida (círculos azules) y la ajustada (cruces verdes) de los experimentos 10 y 11 del ET3.

En el experimento 1 (Figura 25) se comprueba que el simulador ESIM se comporta de la forma esperada ante un movimiento en línea recta en X e Y, unos valores constantes en Z y *yaw*, y unos *pitch* y *roll* nulos (Modelo 1). Los experimentos 2 y 3 (Figura 25) ponen de manifiesto que el simulador suaviza los movimientos oscilatorios del modelo de trayectoria recta y alto realismo (Modelo 2). Independientemente del número de *splines* utilizados, no se consigue una mejora del ajuste. Sin embargo, los tramos creciente y decreciente observados al comenzar y finalizar los movimientos oscilatorios hacen suponer un comportamiento anómalo del ajuste. De hecho, en el experimento 4 (Figura 26) se ha simulado únicamente una parte de la trayectoria del experimento 2 (Figura 25), obteniéndose movimientos que se podrían aproximar mejor con ecuaciones cuadráticas o rectas. No obstante, el simulador no es capaz de ajustar estas formas con precisión, a pesar de que 200 *splines* son suficientes para ello. Este último razonamiento se comprueba en el experimento 5 (Figura 26), en el que se ha vuelto a representar una parte aún más pequeña de la trayectoria de la cabeza, en la que sus movimientos se pueden aproximar por rectas. Manteniendo aproximadamente el número de *splines* por segundo (en este caso se han usado 8 *splines* para una trayectoria de 0,09 s, en vez de 200 para una de 2,81 s), se ha comprobado que el simulador sí que podría ajustar con precisión las trayectorias anteriores con el número de *splines* definidos, por lo que existe algún problema con la programación del ajuste de trayectorias del simulador ESIM. Con el experimento 6 (Figura 26) se concluye el conjunto de experimentos de trayectorias rectas y se

valida la estrategia de partición como una solución efectiva a las diferencias existentes entre las trayectorias definidas y ajustadas.

Respecto a los experimentos con trayectorias curvas, en los experimentos 7, 8 y 9 (Figura 27) se aprecian comportamientos similares a los descritos previamente, independientemente del nivel de realismo escogido (Modelo 3 y Modelo 4). En estos experimentos, el aumento del número de *splines* no permite mejorar significativamente la precisión del ajuste. Además, se observa algún otro rasgo anómalo del ajuste del simulador, ya que en los experimentos 8 y 9 (Figura 27) se define un *pitch* nulo, pero su ajuste obtiene valores de hasta $0,8^\circ$. En el experimento 10 (Figura 28) se reduce el grado de los *splines* para que estos tengan menos incógnitas asociadas y se puedan comportar correctamente, pero el resultado es de nuevo negativo. Finalmente, el experimento 11 (Figura 28) vuelve a confirmar la efectividad de la partición.

El suavizado de las oscilaciones producido por la utilización de *splines* se ha comentado con Scaramuzza et al. (1), que han propuesto aumentar el número de *splines* por encima de 200, para reducir las diferencias entre las trayectorias definidas y ajustadas. En función de los resultados observados, se concluye que estas diferencias no se deben al uso de *splines*, sino a la propia programación del ajuste. En este proyecto se ha desarrollado la estrategia de partición para solventar dicho problema y conseguir un comportamiento más aproximado al esperado.

6. La visión protésica

En este capítulo se realiza la simulación de la visión protésica a partir del procesamiento de las imágenes de eventos, adaptando las técnicas convencionales de visión por computador a las características especiales de las cámaras de eventos. Estas tareas se han conseguido mediante el desarrollo de un nuevo programa en Python 3, el Programa de Visión Protésica (PVP), que realiza el procesamiento de las imágenes de eventos y su representación en fosfenos. El bloque de procesamiento se ha diseñado para elegir entre distintos algoritmos de visión por computador, de los que se ha hallado su combinación óptima. Se ha realizado el Estudio de la Visión Protésica (EVP) con 4 experimentos, en los que las imágenes de eventos obtenidas con el simulador ESIM se han procesado y representado en fosfenos para analizar los resultados.

6.1. Programa de Visión Protésica, PVP

El desarrollo de las tareas propias de un SAN requiere del empleo de distintas técnicas de visión por computador. Adicionalmente, en el caso de que el SAN esté basado en la nueva cámara de eventos, es necesario adaptar las técnicas convencionales de visión por computador al particular comportamiento de los eventos. Para ello, el PVP diseñado incluye una umbralización, un aumento de la cantidad de información por imagen y la introducción de información relativa al movimiento, así como el filtrado y resaltado de los eventos según su vecindad. Adicionalmente, el PVP también simula la visión protésica, lee las imágenes de eventos y almacena las imágenes de fosfenos en un nuevo directorio. El código de visión protésica utilizado es del trabajo de Sánchez-García et al. (10). El PVP emplea el módulo de *math* y los paquetes de NumPy, OpenCV y Scipy (para sus utilidades de *scipy.misc*), y sus funciones son:

- Leer ordenadamente las imágenes
- Realizar una umbralización
- Aumentar la cantidad de información por imagen
- Incluir información relativa al movimiento
- Filtrar y resaltar los datos
- Simular la visión protésica
- Almacenar las imágenes de fosfenos

6.1.1. Lectura ordenada de las imágenes

Al igual que el PP, el PVP recorre un fichero de imágenes de manera ordenada utilizando un único cursor, cuyo valor numérico se descompone en millares, centenas, decenas y unidades. Esta tarea se ha realizado con parte del código del PP explicado en el capítulo 5.

6.1.2. Procesamiento de las imágenes

Umbralización

Los píxeles de las imágenes de eventos pueden ser de color azul, rojo o negro. En este trabajo se han considerado tanto los eventos positivos como negativos, a los que se les ha asignado la intensidad de luminosidad máxima mediante una umbralización (Figura 29). Para ello, se ha obtenido la imagen de eventos en escala de grises mediante *cv2.cvtColor*, en la que

los eventos tienen dos niveles de gris según su polaridad, pero no son blancos. La imagen de eventos en escala de grises se compara píxel a píxel con un umbral distinto al valor de intensidad nula, obteniéndose el valor de *True* si en el píxel hay un evento, y generándose una máscara con estos valores booleanos. Después se utiliza un *numpy.place* con la máscara obtenida para asignar la máxima intensidad (el nivel de blanco) a los píxeles con eventos.

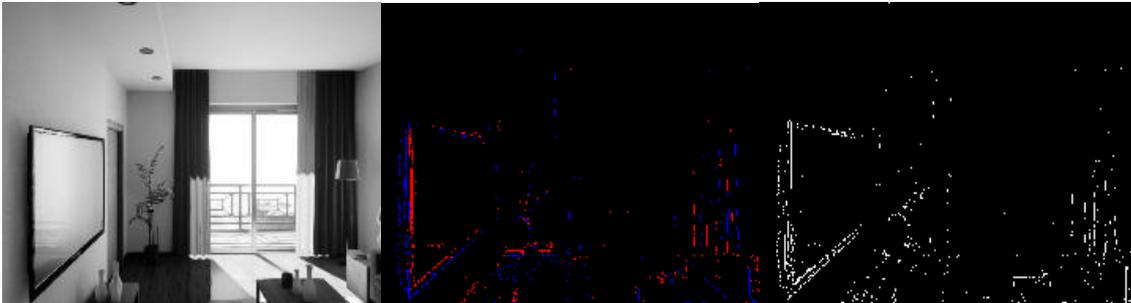


Figura 29. Imagen del escenario del salón (izquierda), su imagen de eventos (centro) y su imagen umbralizada (derecha) obtenida para una trayectoria recta.

Acumulación de eventos

Las imágenes de eventos tienen una elevada tasa de fps (de 300 a casi 2000). Dado que cada una de las imágenes de eventos comprende un intervalo de tiempo muy pequeño, la información almacenada es escasa para su interpretación individual, de manera que el reconocimiento de objetos es complejo. Para solucionar este problema, se ha realizado una acumulación de los eventos mediante la suma de las imágenes de eventos. Esta suma se puede realizar previamente al resto de procesamiento del PVP o después de la representación de fosfenos, y se basa en la adición de subconjuntos de imágenes (cuyo tamaño se puede elegir) o en la acumulación de todas ellas. La acumulación de todas las imágenes está inspirada por las técnicas de patrones de movimiento (*motion templates*) del paquete de OpenCV, basada a su vez en las técnicas de Davis et al. (22), (23).

En el trabajo de Davis et al. (22), (23), la representación del movimiento en imágenes se basa en la acumulación de siluetas relacionadas con distintos instantes de tiempo. Estas siluetas que configuran la Imagen del Historial Temporal del Movimiento (tMHI en inglés). Con esta representación, se puede calcular la próxima localización del objeto en movimiento y realizar tareas de segmentación, reconocimiento y medida de los movimientos. En particular, la dirección en que se produce el movimiento se obtiene a través del gradiente de la orientación del tMHI (Figuras 30 y 31 (22), (23)).

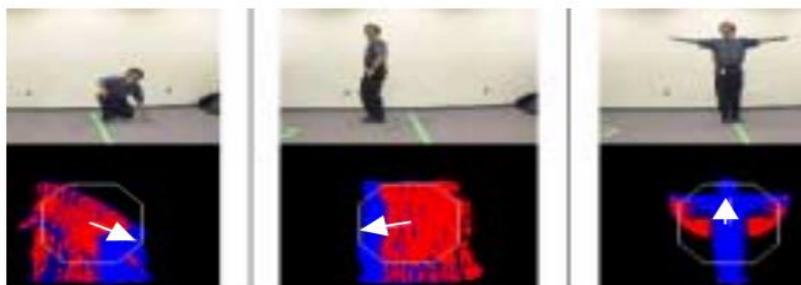


Figura 30. Imágenes de secuencias de distintos movimientos (agacharse, caminar y levantar los brazos) y sus tMHI y direcciones del movimiento (23).

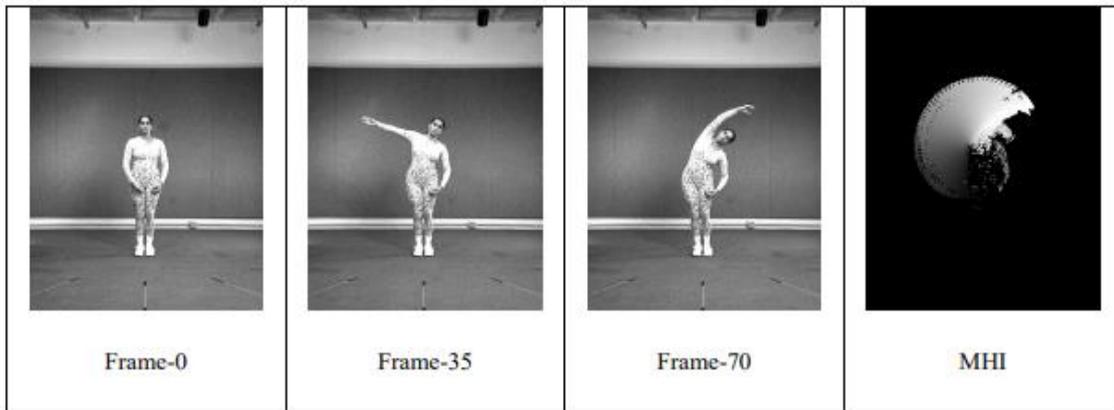


Figura 31. Fotogramas y su tMHI (o MHI) para un movimiento de estiramiento de brazo de tiempo real (23).

En el caso de la suma de subconjuntos, el total de las imágenes de eventos se divide en distintos grupos, cuyas imágenes se suman entre sí. De esta manera, el número de imágenes final será inferior al de imágenes de eventos, por lo que habrá que modificar su denominación en consecuencia, para lo que se han utilizado dos cursores de forma similar al PP. Para realizar la suma de subconjuntos se ha empleado `cv2.add`. Si por el contrario se acumulan todas las imágenes, la imagen total será oscurecida antes de que se añada una nueva imagen, generando estelas (Figura 32). La localización actual de los objetos será la de mayor luminosidad, y sus localizaciones previas serán más oscuras cuanto más antiguas sean. La suma con estela se ha obtenido mediante `cv2.addWeighted` para tener en consideración la reducción de luminosidad aplicada a la imagen acumulada anterior.

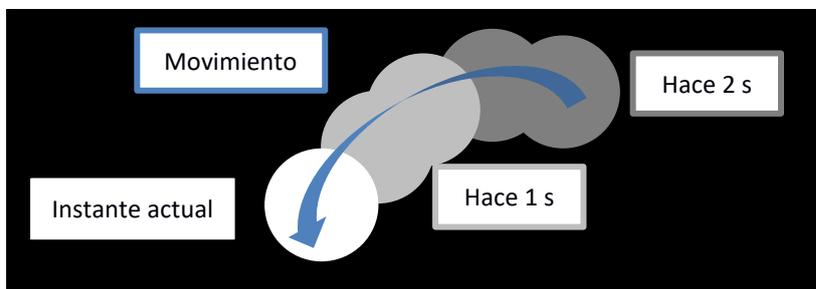


Figura 32. Generación de la estela en un círculo móvil. En azul se indica su movimiento.

Tras probar las opciones mencionadas, se han extraído las siguientes conclusiones. En primer lugar, la utilización de algoritmos de filtrado según la vecindad de los píxeles ha hecho que se descarte la acumulación de imágenes al final, ya que las imágenes de eventos no tienen suficiente densidad de datos. Realizar el filtrado de estas imágenes antes que su suma provoca la obtención de imágenes vacías, ya que la gran mayoría de eventos se han eliminado. En su lugar, una suma de imágenes al principio proporciona una mayor cantidad de eventos en las figuras de objetos, por lo que un filtrado posterior por vecindad sí que diferencia la información relevante del ruido.

Por otro lado, la suma de subconjuntos de imágenes y la utilización de estelas proporcionan una cantidad de información similar. Sin embargo, el uso de subconjuntos reduce en varias veces el número de imágenes procesadas resultantes, en contraposición al de estelas, que mantiene los fps obtenidos. Además, las estelas indican la dirección de los movimientos de las figuras en la imagen, incorporando información sobre las imágenes anteriores. Atendiendo a estas razones, se ha elegido la acumulación con estelas con una

reducción del 5% (factor del 0,95) de la luminosidad de la imagen anterior (Figura 33), lo que proporciona imágenes acumuladas en escala de grises.



Figura 33. Imagen de eventos umbralizada (izquierda) y su acumulada con estela y un factor del 0,95 (derecha).

Transformaciones morfológicas

Las imágenes de eventos son procesadas después con operaciones morfológicas que las modifican en función de los píxeles vecinos de cada píxel. De esta manera se reduce el ruido de las figuras trazadas con eventos, se rellenan sus huecos internos y/o se aumenta su grosor, con las transformaciones de apertura, cierre y dilatación, respectivamente. En esta aplicación, se ha podido elegir entre las transformaciones de apertura, cierre, apertura y cierre o cierre y apertura, realizadas con una máscara de dimensiones ajustables. Además, existe la posibilidad de aplicar una dilatación final con una máscara diferente.

En las imágenes de eventos obtenidas por el simulador ESIM, se aprecian dos tipos de eventos: los que pueden ser fácilmente asociados a formas de objetos o estructuras y los que no. Estos últimos pueden haber sido generados por cambios locales en la textura de las superficies u otras causas, pero su representación en las imágenes dificulta la interpretación global de la escena. De esta manera, se estima necesario un filtrado que elimine estos eventos, aprovechando el carácter disperso de su distribución; así se consigue que las demás operaciones de procesamiento se apliquen solamente al conjunto de información más relevante, y se evita la amplificación de este ruido. Por lo tanto, la primera transformación morfológica que se ha elegido usar es la de apertura, a través de `cv2.morphologyEx` y el argumento de `cv2.MORPH_OPEN`. Una vez realizado el filtrado de los eventos, se aplica la operación de cierre para reducir los huecos internos en las representaciones de objetos macizos, utilizando de nuevo `cv2.morphologyEx`, pero ahora con el argumento `cv2.MORPH_CLOSE`. Finalmente se ha aplicado una dilatación para aumentar el grosor de las figuras trazadas en la imagen, con `cv2.morphologyEx` y su argumento `cv2.MORPH_DILATE`. En relación al tipo de máscara, se ha usado una máscara centrada de dimensiones mínimas (3 por 3 píxeles), ya que una máscara mayor suavizaría la imagen, que perdería nitidez (Figura 34).

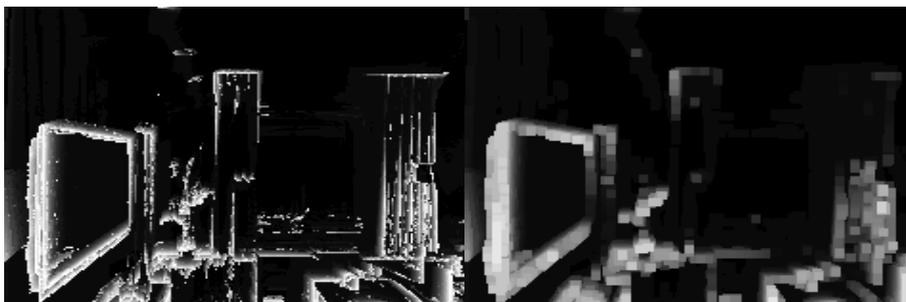


Figura 34. Imagen de eventos umbralizada y acumulada (izquierda) y su transformada morfológica con apertura y cierre (derecha).

Filtro de percentil

En este trabajo se ha introducido un filtro de percentil previo a la simulación de la prótesis visual, inspirado en el código de Sánchez-García et al. (10), y cuyo objetivo es aumentar la luminosidad de los fosfenos y obtener una representación de ellos más comprensible. El filtro de percentil recorre la imagen calculando un percentil concreto de los píxeles comprendidos por una máscara, para asignar su valor a todos ellos. De esta manera, se consigue aumentar o disminuir (en función del percentil elegido) la luminosidad total de los píxeles de la máscara. Un caso particular del filtro de percentil es el del percentil 50, que se corresponde con el filtro de mediana.

Se ha utilizado una máscara de las mismas dimensiones que los fosfenos del código de Sánchez-García et al. (10): 5 por 5 píxeles en una imagen de 180 por 180 píxeles. Tras experimentar con el filtro de percentil, se ha conseguido aumentar la luminosidad de los fosfenos, pero también se dificulta el reconocimiento de objetos. Debido a esta relación de compromiso, se escoge utilizar el percentil 70, pero los resultados obtenidos respecto a la imagen de fosfenos sin el filtro de percentil son bastante similares, por lo que finalmente se ha descartado su uso (Figura 35).

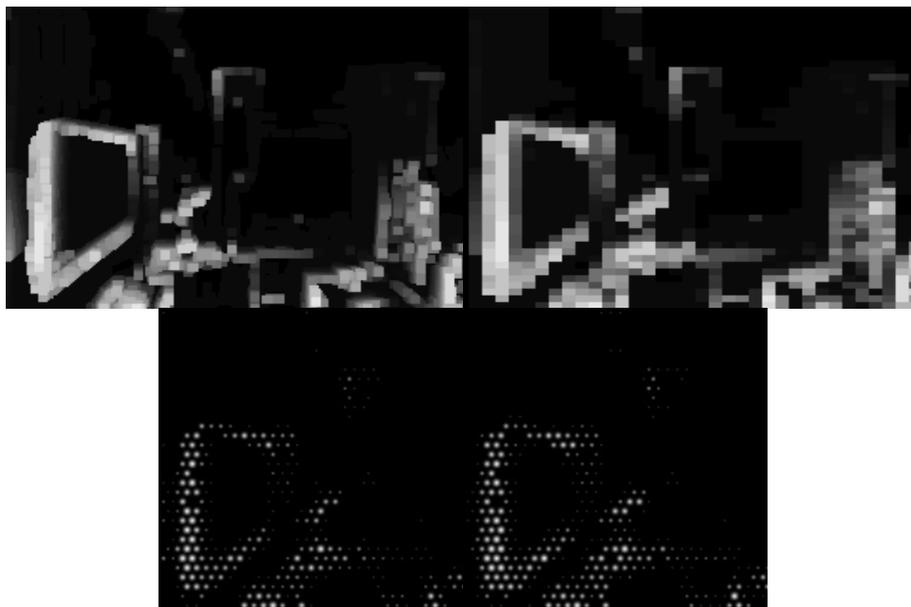


Figura 35. Superior: imagen de eventos umbralizada, acumulada y transformada morfológicamente (izquierda) y su procesamiento con un filtro de percentil 70 (derecha). Inferior: imagen de fosfenos de la imagen procesada sin filtro de percentil y con él (izquierda y derecha respectivamente).

6.1.3. Simulación de la visión protésica

En este trabajo se ha utilizado el algoritmo de Sánchez-García et al. (10) para representar los fosfenos. Este algoritmo incluye varias características de las prótesis visuales, como son su limitada resolución, distribución, *dropout* y perfil de luminosidad gaussiano de los fosfenos. Los parámetros utilizados en cada una de estas propiedades son los mismo que en Sánchez-García et al. (10), a excepción de que la luminosidad de los fosfenos ha sido calculada con media en vez de mediana (Tabla 9). Esta elección se basa en que a pesar de una menor nitidez de las imágenes, la media proporciona una representación más luminosa de la estela.

Por lo tanto, se obtiene una mayor información acerca de los movimientos, que en definitiva es uno de los aspectos a priorizar en un SAN basado en la cámara de eventos (Figura 36). Para utilizar el algoritmo de simulación de las prótesis visuales, se debe adaptar el formato de las imágenes procesadas, que han sido recortadas mediante la selección de un rango concreto de su matriz y luego aumentadas mediante *cv2.resize*.

Resolución prótesis	Distribución	Dropout	Cálculo luminosidad
32 x 32 fosfenos	Hexagonal	10 %	Media

Tabla 9. Valores de los parámetros empleados en la simulación de prótesis visuales en este trabajo.



Figura 36. Imagen del salón (izquierda), su imagen de eventos procesada (centro) y su representación para la visión protésica (derecha).

6.2. Estudio de la Visión Protésica, EVP

Con la selección previa de los algoritmos de visión por computador y sus parámetros, se ha realizado el EVP planteando 4 experimentos para analizar los resultados tras el procesamiento de las imágenes de eventos (Tabla 10). En cada uno de estos experimentos, se han comparado las imágenes de eventos de ESIM con las obtenidas con el PVP, incluyendo la imagen procesada antes de su representación en fosfenos (Figuras 38 a 41). Para ello, se han escogido 4 imágenes por experimento, que corresponden aproximadamente con el 25, 50, 75 y 100 % de la trayectoria, suponiendo una generación de imágenes uniforme en el tiempo. A modo de referencia, también se han incluido las imágenes que obtiene una cámara convencional al realizar el mismo experimento. Estas imágenes no son correspondientes porque ESIM interpola los eventos entre los instantes de las imágenes de la cámara convencional. Además, se ha usado un modelo de movimiento de bajo realismo para la cámara convencional, lo que ha permitido obtener sus imágenes de forma más rápida al no utilizarse la estrategia de partición. De esta manera, las imágenes de la cámara convencional mostradas se han obtenido con una trayectoria diferente pero aproximada a la de las imágenes de eventos, por lo que existen diferencias entre ellas respecto a la localización y orientación de la cámara.

Todas las imágenes consideradas se muestran a un tamaño pequeño para poder agrupar los resultados de cada experimento en una misma figura. En el estudio individual de cada imagen, o la implementación real del SAN, estas imágenes se amplían para apreciar todos los detalles de eventos y fosfenos iluminados. Las imágenes de fosfenos de la simulación de visión protésica no representan toda la escena grabada por la cámara de eventos porque han sido recortadas, de acuerdo a la resolución de la prótesis visual. En los siguientes experimentos se utilizan distintos objetos como referencia, definidos en la Figura 37.

Estudio	Experimento	Escenario	Modelo (trayectoria, realismo)	Tiempo (s)	Localización
EVP	1	Salón	M2 (recta, alto)	0,94	1
	2		M4 (curva, alto)	0,52	2
	3	Ciudad	M2 (recta, alto)	0,94	1
	4		M4 (curva, alto)	0,52	

Tabla 10. Experimentos para la validación de las técnicas de visión por computados.

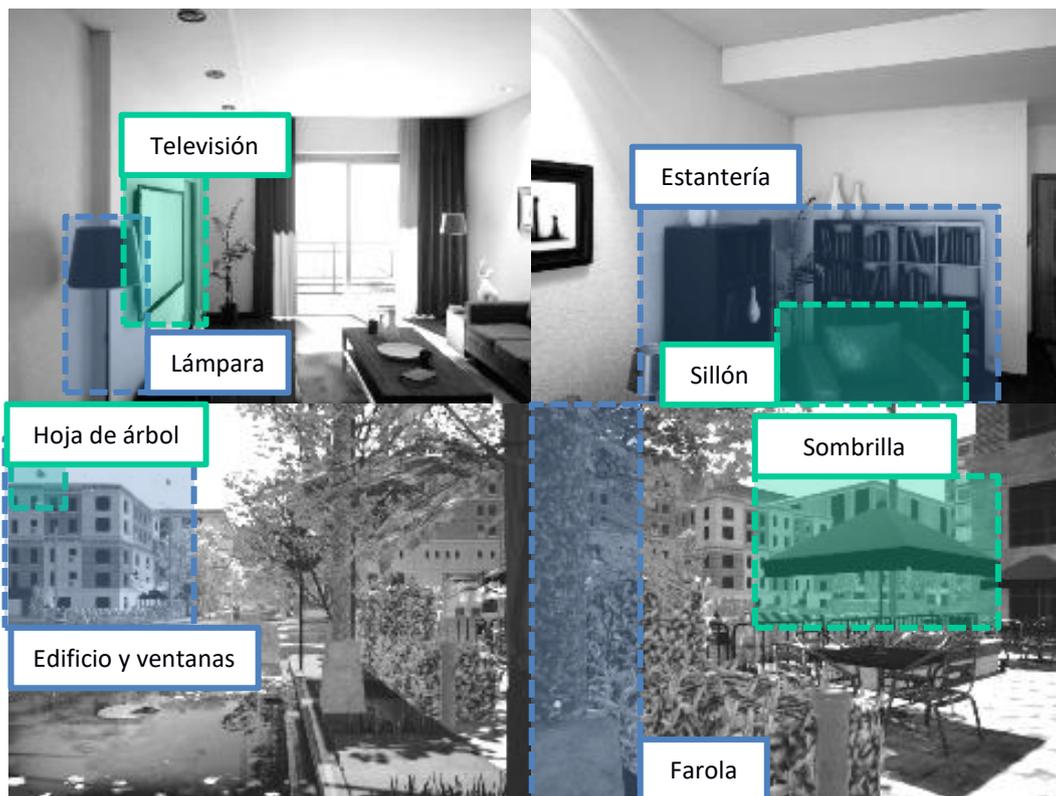


Figura 37. Objetos de referencia en los escenarios utilizados.

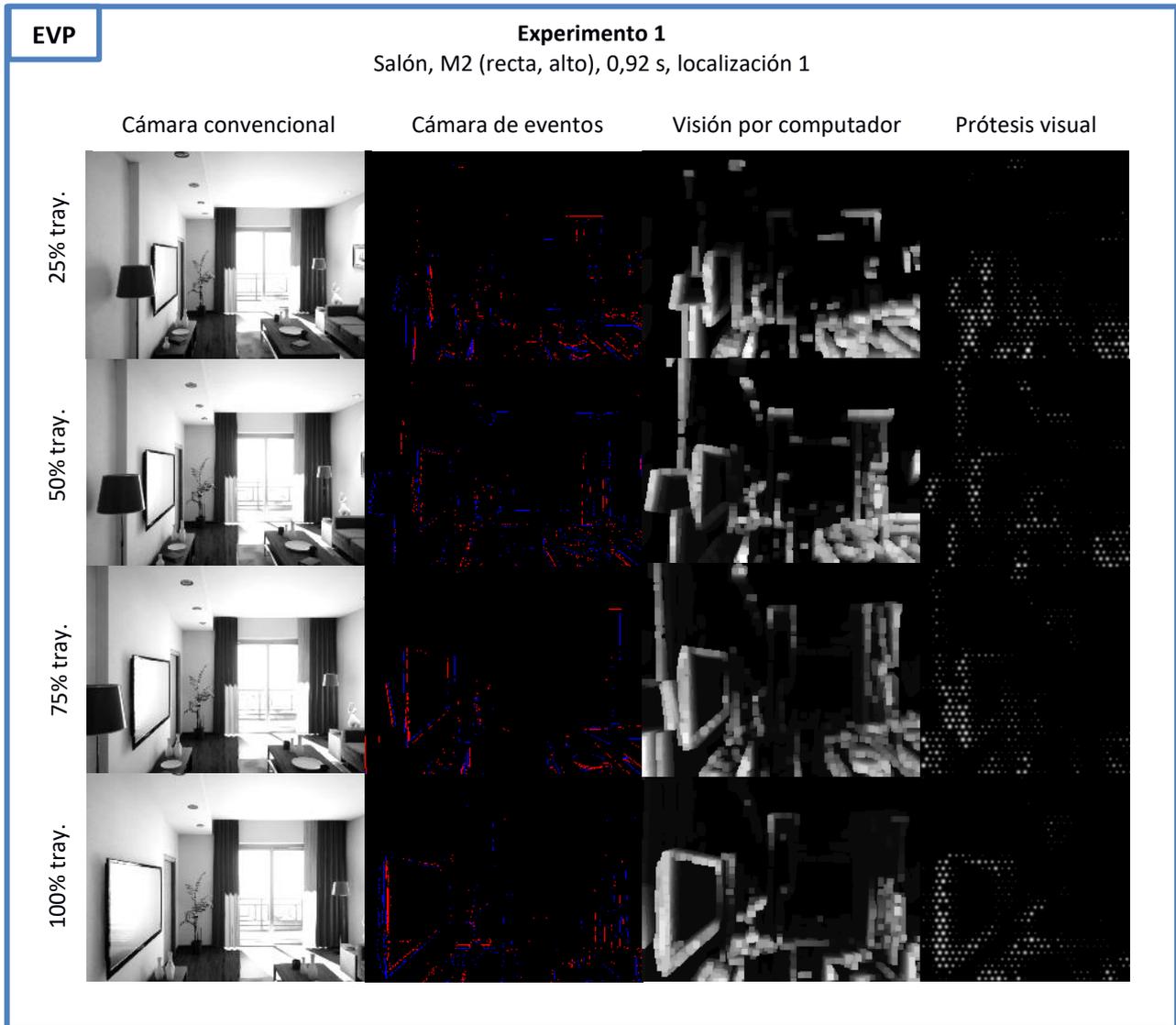


Figura 38. Resultados del experimento 1 del EVP.

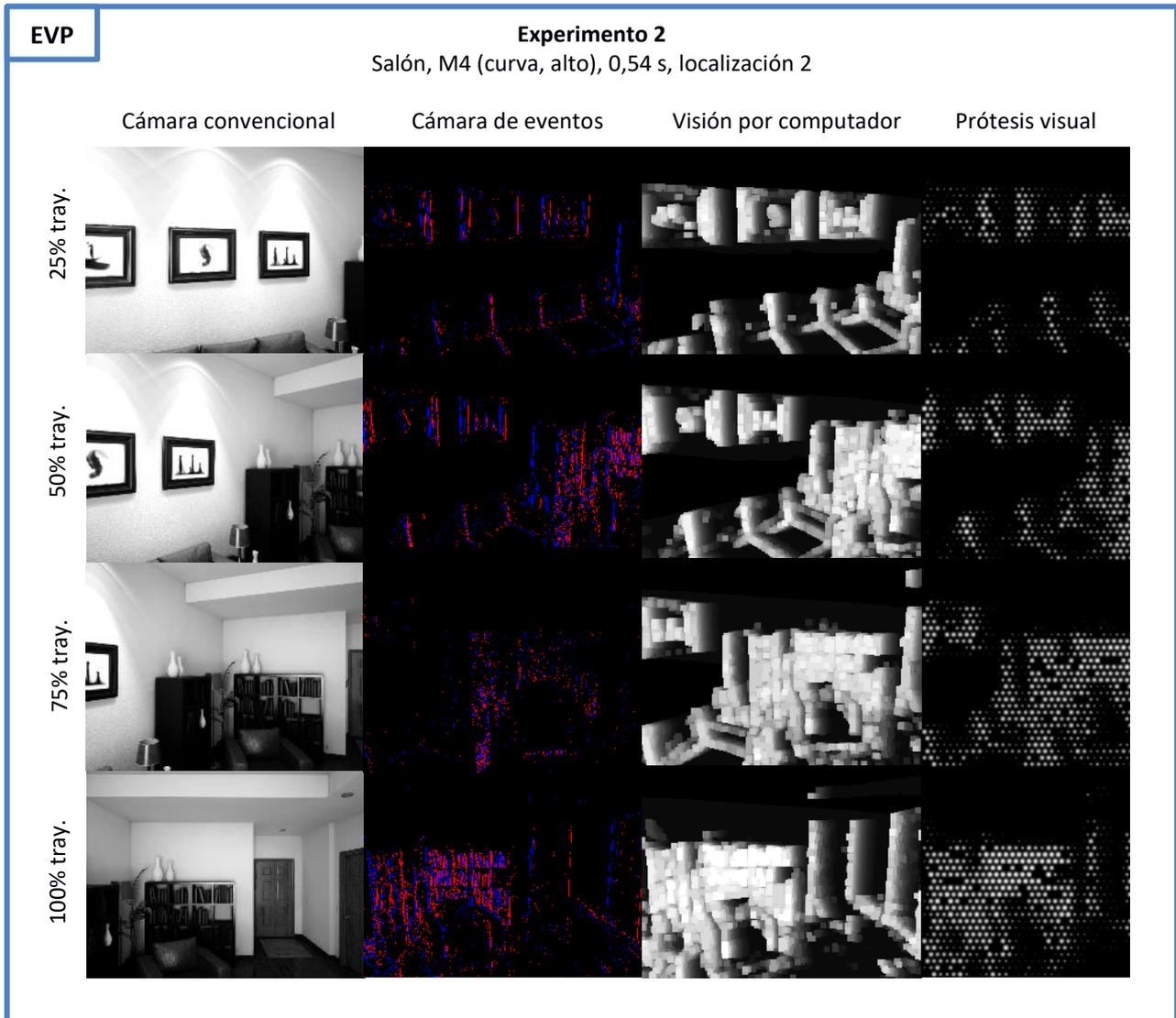


Figura 39. Resultados del experimento 2 del EVP.

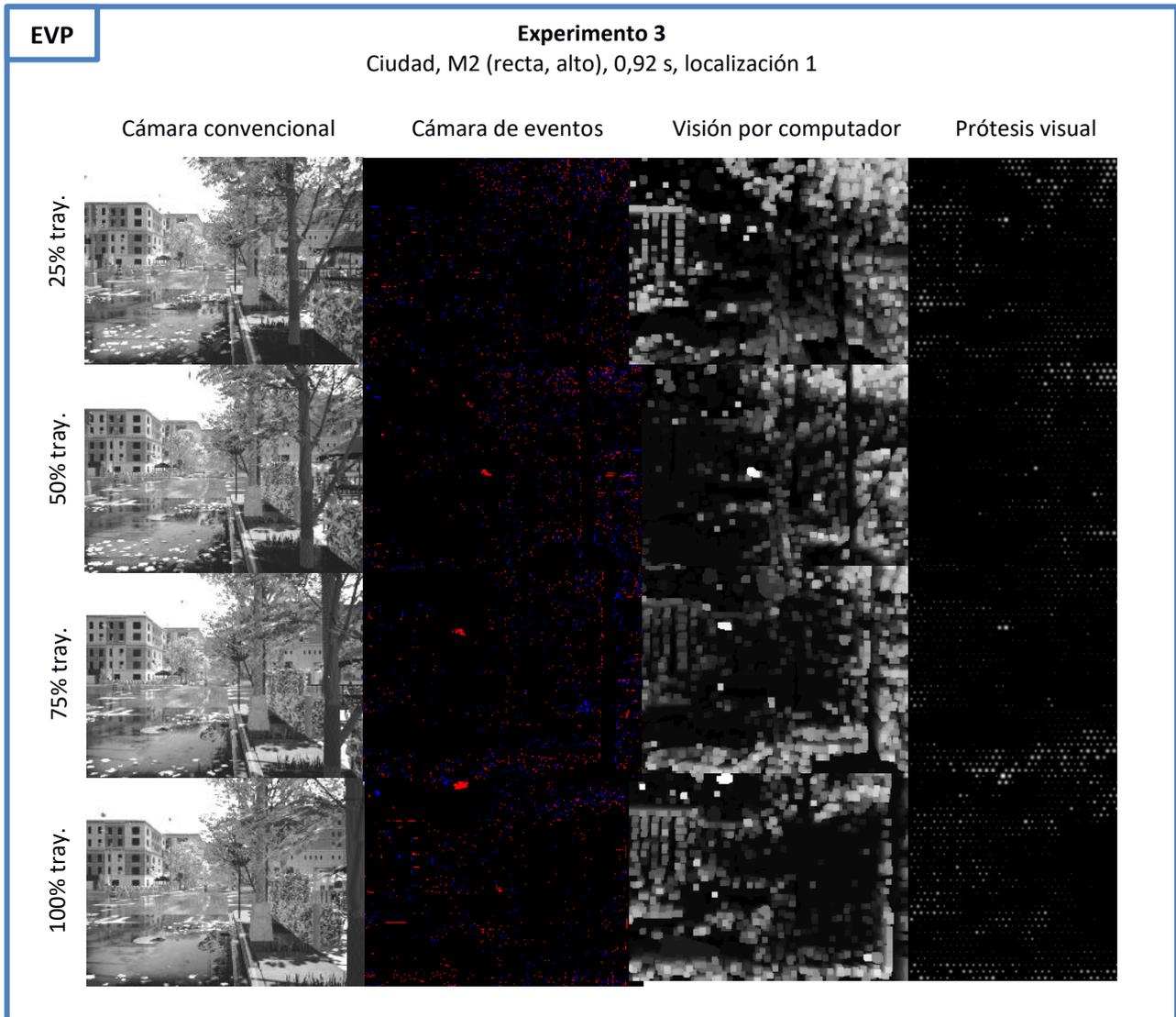


Figura 40. Resultados del experimento 3 del EVP.

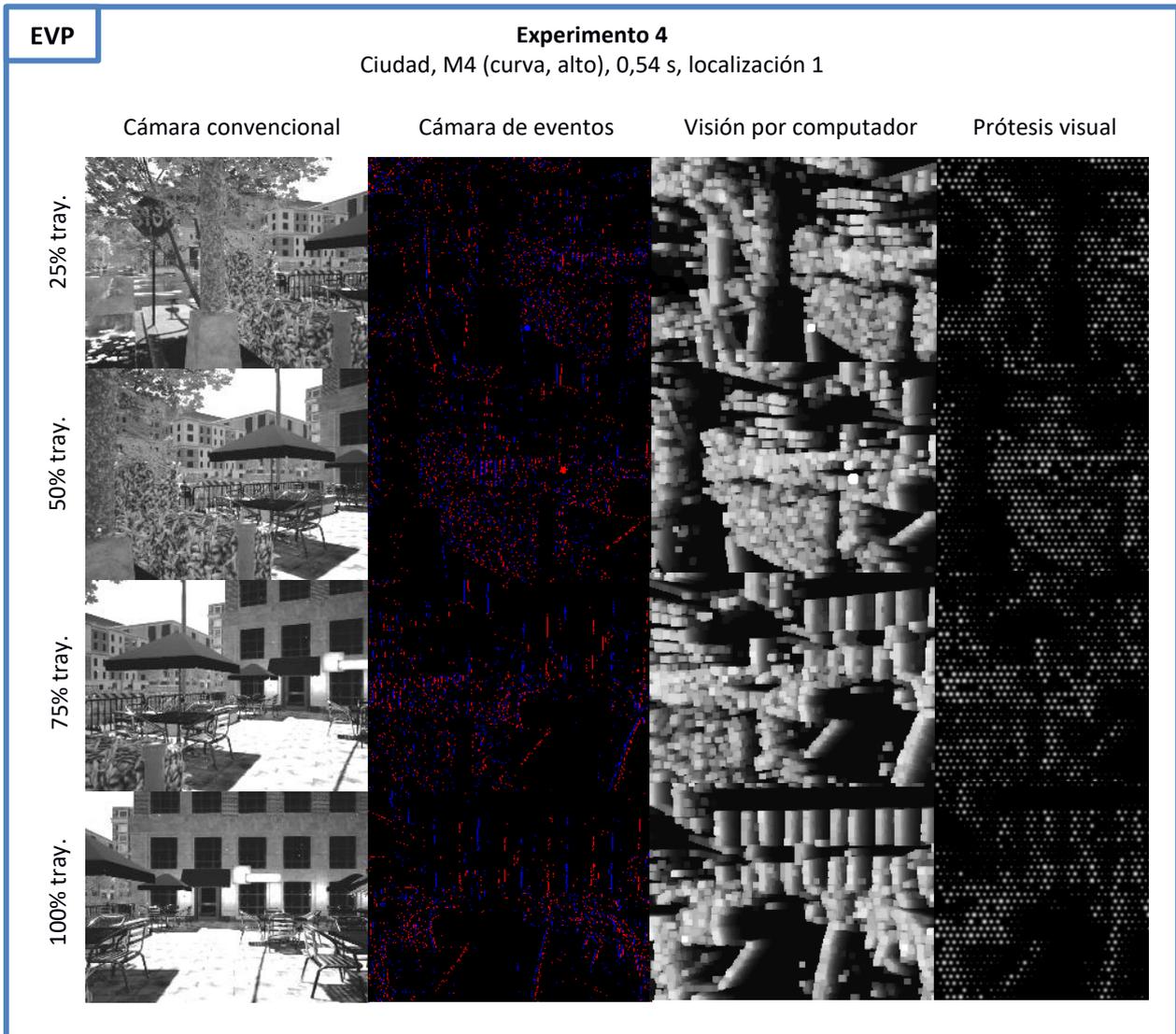


Figura 41. Resultados del experimento 4 del EVP.

Se observan dos tipologías distintas del reconocimiento de objetos. En la primera de ellas, los objetos se detectan gracias a la especial generación de eventos en su contorno, como ocurre con la televisión o la lámpara del experimento 1 (Figura 38). Este reconocimiento es posible debido a que los objetos y texturas circundantes, y la propia textura del objeto, son de dimensiones muy inferiores a las que definen la forma del objeto. De esta manera, el contorno del objeto se destaca sobre una vecindad que no genera eventos. Por otro lado, en las regiones con un gran número de eventos, las formas y figuras se aprecian a través de las regiones negras sin eventos. Este hecho se observa en el sillón de delante de la estantería en el experimento 2 (Figura 39). Este objeto es reconocido por su superficie negra sin eventos en vez de por los eventos acumulados en su contorno, ya que estos últimos se confunden con los eventos excitados por los libros.

Los resultados de los experimentos 3 y 4 (Figuras 40 y 41) ponen de relieve el mayor número de objetos y aumento de las texturas del escenario de la ciudad respecto al del salón. Se vuelven a diferenciar las dos tipologías de reconocimiento de objetos. Se tienen tanto objetos destacados por sus contornos, como edificios y ventanas, como aquellos reconocibles por superficies negras, por ejemplo el tronco del árbol, la farola o la sombrilla de la terraza del bar. El aumento en la cantidad de objetos y texturas hace que las imágenes obtenidas mediante la visión por computador sean más complejas que las del salón, en especial las imágenes de fosfenos, cuya comprensión se dificulta.

Los resultados obtenidos en las imágenes de fosfenos, tanto en el salón como en la ciudad, son equiparables a los obtenidos por las técnicas actuales. Estas técnicas se basan en el uso cámaras convencionales, escenarios de interior más sencillos y técnicas de procesamiento más complejas, como las redes neuronales. De esta manera, la visión protésica desarrollada en este trabajo tiene una eficiencia mayor que estas técnicas actuales (Figura 42).

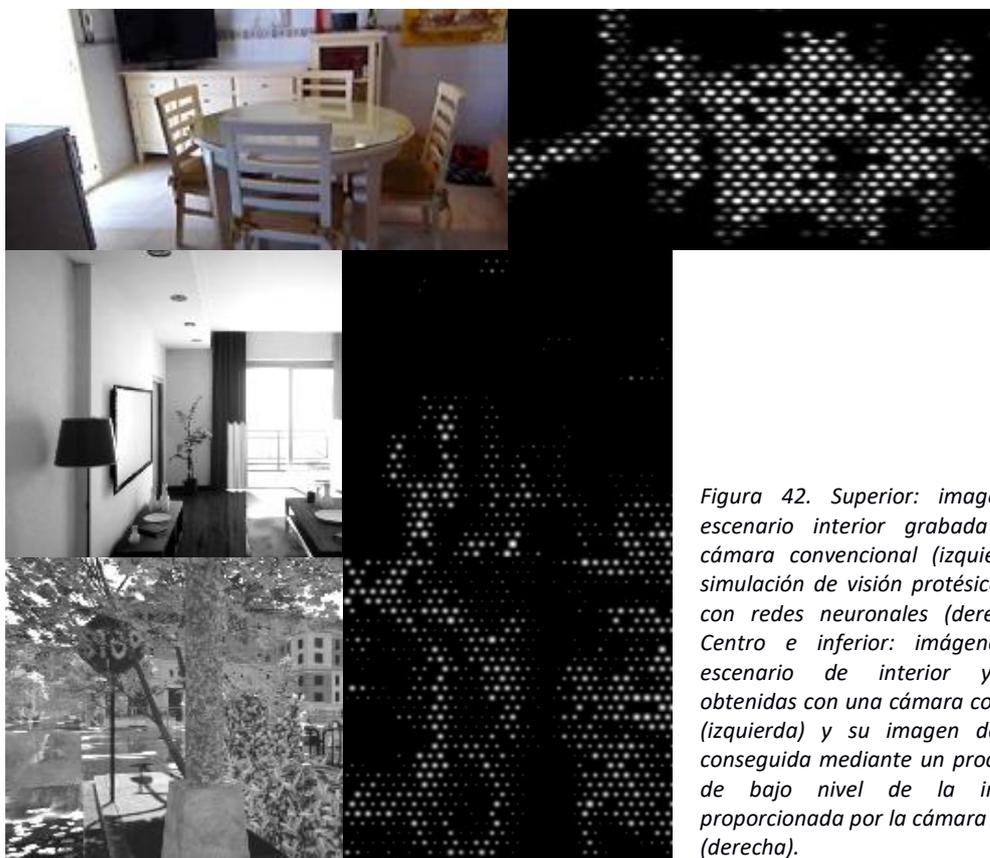


Figura 42. Superior: imagen de un escenario interior grabada con una cámara convencional (izquierda) y su simulación de visión protésica obtenida con redes neuronales (derecha) (10). Centro e inferior: imágenes de un escenario de interior y exterior obtenidas con una cámara convencional (izquierda) y su imagen de fosfenos conseguida mediante un procesamiento de bajo nivel de la información proporcionada por la cámara de eventos (derecha).

Cabe destacar que los escenarios de ciudades son de los más complejos para las tareas de navegación, y que dentro de ellos, el escenario usado en este trabajo es uno de los más desfavorables. En este escenario se representa una caída continua de hojas de los árboles, una situación que además no se da generalmente cuando las personas nos desplazamos por la ciudad. La simulación de este fenómeno dificulta la interpretación de las imágenes, que se mejora con la utilización de un escenario de ciudad no tan desfavorable, que es a la vez más cercano a las condiciones normales de navegación.

La utilización de este escenario desfavorable ha puesto en relieve el potencial de la cámara de eventos en la detección de objetos móviles. En los escenarios del salón y la ciudad, la gran mayoría de objetos son estáticos y el movimiento relativo entre la cámara y los objetos viene definido sólo por el movimiento de la cámara. En el caso de las hojas que caen de los árboles, en el escenario de la ciudad, el movimiento relativo entre las hojas y la cámara es mayor, ya que los dos se mueven de forma distinta, lo que permite un reconocimiento excepcional de las hojas dentro de la escena (Figura 43).

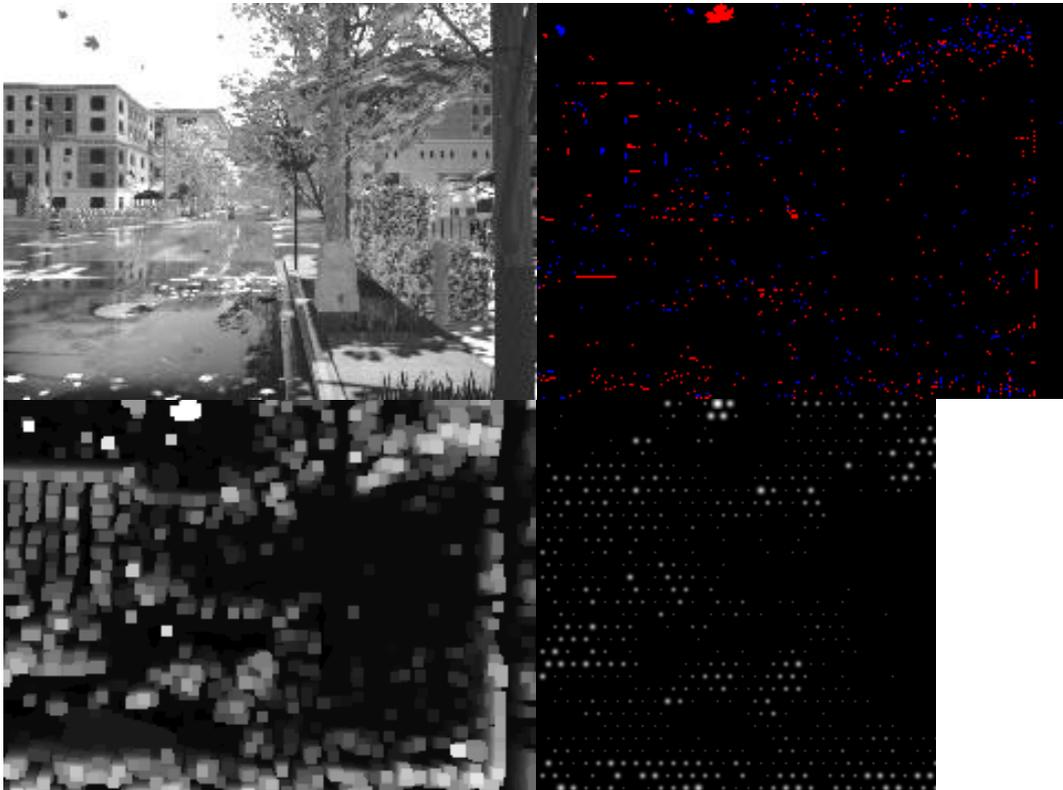


Figura 43. Superior: Imagen de cámara convencional de la ciudad (izquierda) y su imagen de eventos (derecha) Ambas imágenes no concuerdan exactamente porque se han utilizado modelos de movimiento de distinto realismo y por la interpolación de eventos de ESIM. Inferior: procesamiento de la imagen de eventos (izquierda) y su imagen de fosfenos (derecha).

A través de los experimentos realizados, se ha confirmado la eficacia de los algoritmos de visión por computador adaptados y desarrollados, en la mejora la información proporcionada por la cámara de eventos. La simulación de prótesis visuales ha alcanzado resultados equiparables a los obtenidos con cámaras convencionales y redes neuronales, consiguiéndose una mayor eficiencia. Se ha observado que la cámara de eventos mejora su comportamiento en el reconocimiento de objetos si estos son móviles, como las hojas que caen en la ciudad.

7. Conclusiones y trabajo futuro

El desarrollo de un SAN para personas, basado en la cámara de eventos, es una tarea compleja que aborda distintas áreas de trabajo. El diseño de estos sistemas depende de los sensores utilizados, de su ubicación en el usuario, y de los movimientos relativos entre éste y los objetos del escenario. En este TFM se han considerado todos estos aspectos, y en su función se han extraído distintos requerimientos de diseño, escogiendo las mejores soluciones para ellos. Los SAN para visión protésica son utilizados por personas con discapacidad visual, pero pueden generalizarse para personas sin esta discapacidad. Esta generalización permite desarrollar simultáneamente ambas aplicaciones, razón por la que se ha elegido desarrollar un SAN para visión protésica. Debido a la novedad de la cámara de eventos y su elevado coste actual, en este trabajo se ha realizado una simulación completa y modular del SAN. Esta simulación se ha dividido en la simulación de la cámara de eventos, el movimiento humano, el escenario y la visión protésica. La división de las simulaciones permite la sustitución de cualquiera de ellas por sus elementos correspondientes en la implementación real.

Los SAN para personas están condicionados por los distintos movimientos humanos. Cuando las personas nos desplazamos, andamos en línea recta, escogiendo el camino más corto hacia el lugar al que queremos llegar. No obstante, este movimiento solamente es recto en apariencia. En el estado del arte se ha observado que cuando andamos en línea recta, las partes de nuestro cuerpo se mueven de izquierda a derecha, de arriba abajo y trazando círculos, y de forma similar ocurre cuando realizamos giros en esquinas. Dentro de este proyecto se han aproximado y modelado todos estos movimientos reales del cuerpo humano, obteniendo una simulación fidedigna del desplazamiento de las personas.

La simulación de la cámara de eventos ha sido realizada mediante el simulador ESIM, en el que se calculan los eventos generados por distintas trayectorias de la cámara. Este simulador ha sido estudiado para analizar la generación de eventos y las discrepancias entre la trayectoria definida para la cámara y la que se simulaba finalmente. En su estudio, se han considerado distintos escenarios y localizaciones en ellos, así como varios movimientos de la cámara, según su trayectoria (recta o curva) o su nivel de realismo respecto a los movimientos humanos. Para realizar este estudio, se ha realizado un programa en Python 3 (el Programa de Trayectorias), que automatiza la toma de datos de los experimentos. Gracias a estos estudios, se han analizado las discrepancias en las trayectorias, que impiden una simulación de alto realismo en los movimientos humanos, no considerada en la bibliografía consultada. Para solucionar este problema, se ha desarrollado una estrategia de experimentación, la partición, que permite conseguir una simulación más realista. Para recopilar la información de los experimentos con partición, se ha desarrollado un nuevo programa en Python 3, el Programa de Partición, que ha permitido agrupar los resultados de las distintas partes de un experimento.

En este trabajo se ha refinado y mejorado la información de la cámara de eventos mediante el desarrollo de un procesamiento de bajo nivel, incluido dentro del Programa de Visión Protésica, que también representa la visión con fosfenos. Este programa se ha utilizado para obtener las imágenes del SAN basado en la cámara de eventos, con el que se ha conseguido una eficacia similar a las técnicas actuales, pero con un aprovechamiento de los recursos mucho más eficiente gracias a la cámara de eventos.

En resumen, este proyecto llega a su fin con sus objetivos cumplidos. Se ha analizado la información y evaluado el comportamiento de la cámara de eventos dentro de un SAN para personas, se ha desarrollado un entorno de simulación realista y se han adaptado los

algoritmos de visión por computador a la información de la cámara de eventos. Además de estos objetivos, con la simulación del SAN se han extraído conclusiones de gran valor, se ha avanzado en el desarrollo e implementación de la cámara de eventos y se han obtenido resultados que superan ya en varios aspectos a los conseguidos mediante técnicas actuales consolidadas.

7.1. Trabajo futuro

La realización de este proyecto abre nuevas posibilidades de aplicación a la cámara de eventos y motiva el desarrollo de trabajos que utilicen el conocimiento adquirido. Dentro de las aplicaciones de la cámara de eventos, como la detección de colisiones o el seguimiento de objetos, se hace necesaria la creación de un tipo de escenarios concretos, no encontrados en el estado del arte. Estos escenarios deben ser realistas e incorporar elementos móviles de los que se conozca su trayectoria, para lo que se puede usar Unreal Engine. El escenario de ciudad utilizado en este proyecto contiene hojas de árboles que caen al suelo, pero no se disponen de los datos de sus movimientos, y su determinación mediante observación es compleja.

La existencia de dos tipologías de reconocimiento dentro de una misma imagen dificulta la interpretación conjunta de la imagen. Para mejorar su reconocimiento, se propone detectar las regiones con una elevada densidad de eventos y representar sólo sus contornos. Por otro lado, el reconocimiento también se puede mejorar mediante un mayor filtrado de los eventos. De todos los eventos generados, los más relevantes son aquellos que son más cercanos o que se relacionan con objetos a una mayor velocidad. Para realizar su filtrado, se considera el uso de técnicas de detección y seguimiento de objetos o de sensores adicionales.

Bibliografía

1. *ESIM: an Open Event Camera Simulator*. **Henri Rebecq, Daniel Gehrig, Davide Scaramuzza**. Zürich, Switzerland : Proceedings of The 2nd Conference on Robot Learning, 2018. PMLR 87:969-982.
2. **Wenbin Li, Sajad Saeedi, John McCormac, Ronald Clark, Dimos Tzoumanikas, Qing Ye, Yuzhong Huang, Rui Tang, Stefan Leutenegger**. *InteriorNet: Mega-scale Multi-sensor Photo-realistic Indoor Scenes Dataset*. 2018. arXiv:1809.00716v1 [cs.CV].
3. *The Event-Camera Dataset and Simulator: Event-based Data for Pose Estimation, Visual Odometry, and SLAM*. **Elias Mueggler, Henri Rebecq, Guillermo Gallego, Tobi Delbruck and Davide Scaramuzza**. 142–149, s.l. : The International Journal of Robotics Research, 2016, Vol. 36(2).
4. *Simultaneous Optical Flow and Intensity Estimation from an Event Camera*. **Patrick Bardow, Andrew J. Davison, Stefan Leutenegger**. s.l. : IEEE Conference on Computer Vision and Pattern Recognition, 2016.
5. *Unified Temporal and Spatial Calibration for Multi-Sensor Systems*. **Paul Furgale, Joern Rehder, and Roland Siegwart**. Tokyo, Japan : IEEE, 2013. 10.1109/IROS.2013.6696514.
6. **Henri Rebecq, Daniel Gehrig and Davide Scaramuzza**. uzh-rpg / rpg_esim. [En línea] https://github.com/uzh-rpg/rpg_esim.
7. *THE ARGUS® II RETINAL PROSTHESIS SYSTEM: AN OVERVIEW*. **David D. Zhou, Ph.D., Jessy D. Dorn, Ph.D., Robert J. Greenberg, M.D., Ph.D., Argus II Study Group, Second Sight Medical Products, Inc. Sylmar, CA**. San Jose, CA, USA : IEEE International Conference on Multimedia and Expo Workshops (ICMEW), 2013. 978-1-4799-1604-7.
8. *Simulated prosthetic vision: The benefits from computer-based object recognition and localization*. **Macé, Marc and Denis, Grégoire and Jouffrais, Christophe**. 7, s.l. : Artificial Organs, 2015, Vol. 39, págs. E102-E113. 0160-564X.
9. *Simulating prosthetic vision: I. Visual models of phosphenes*. **Spencer C. Chen, Gregg J. Suaning, John W. Morley, Nigel H. Lovell**. s.l. : Elsevier Ltd., 2009, Vision Research, Vol. 49, págs. 1493-1506.
10. *Melani Sanchez-Garcia, Ruben Martinez-Cantin, Jose J. Guerrero. Structural and object detection for phosphene images*. 2018. arXiv:1809.09607 [cs.CV].
11. *How Fast is Too Fast? The Role of Perception Latency in High-Speed Sense and Avoid*. **Davide Falanga, Suseong Kim, Davide Scaramuzza**. 2, s.l. : IEEE ROBOTICS AND AUTOMATION LETTERS, 2019, Vol. 4. DOI:10.1109/LRA.2019.2898117.
12. *Event-based, 6-DOF Pose Tracking for High-Speed Maneuvers*. **Elias Mueggler, Basil Huber and Davide Scaramuzza**. Chicago : IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2014. DOI: 10.1109/IROS.2014.6942940.
13. *Towards a framework for end-to-end control of a simulated vehicle with spiking neural networks*. **Jacques Kaiser, J. Camilo Vasquez Tieck, Christian Hubschneider, Peter Wolf, Michael Weber, Michael Hoff, Alexander Friedrich, Konrad Wojtasik, Arne Roennau, Ralf Kohlhaas, Rüdiger Dillmann, J. Marius Zöllner**. San Francisco, CA, USA : 2016 IEEE International Conference on Simulation, Modeling, and Programming for Autonomous Robots (SIMPAN), 2016. DOI: 10.1109/SIMPAN.2016.7862386.
14. *Interaction of the body, head, and eyes during walking and turning*. **Takao Imai, Steven T. Moore, Theodore Raphan, Bernard Cohen**. 1, s.l. : Experimental Brain Research, 2001, Vol. 136. <https://doi.org/10.1007/s002210000533>.
15. *Effects of walking velocity on vertical head and body movements during locomotion*. **Eishi Hirasaki, Steven T. Moore, Theodore Raphan, Bernard Cohen**. 117, s.l. : Experimental Brain Research, 1999, Vol. 127. <https://doi.org/10.1007/s002210050781>.

16. *The up and down bobbing of human walking: a compromise between muscle work and efficiency.* **Firas Massaad, Thierry M. Lejeune and Christine Detrembleur.** 582, s.l. : The Journal of Physiology, 2007. doi:10.1113/jphysiol.2007.127969.
17. **UnrealCV.** Getting started. [En línea]
http://docs.unrealcv.org/en/master/tutorials/getting_started.html.
18. *How do humans turn? Head and body movements for the steering of locomotion.* **Halim Hicheur, Alain Berthoz.** Tsukuba, Japan : 5th IEEE-RAS International Conference on Humanoid Robots, 2005. DOI: 10.1109/ICHR.2005.1573578.
19. *A tutorial on SE(3) transformation parameterizations and on-manifold optimization.* **Blanco, Jose-Luis.** 2013.
20. **Co., McDonnell-Douglas Technical Services.** *Euler Angles, Quaternios and Transformation Matrices.* Houston, TX, United States : NASA, 1977. NASA-CR-151435, DN-1.4-8-020.
21. **Python Software Foundation.** Python 3.7.4rc1 documentation. [En línea]
<https://docs.python.org/3/>.
22. *The Representation and Recognition of Action Using Temporal Templates.* **Davis, J.W. and Bobick, A.F.** s.l. : IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 1997. CVPR97.
23. *Motion Segmentation and Pose Recognition with Motion History Gradients.* **Davis, J.W. and Bradski, G.R.** 3, s.l. : Machine Vision and Applications, 2002, Vol. 13: 174.
<https://doi.org/10.1007/s001380100064>.