Escuela Técnica Superior de Ingenieros
Industriales y de Telecomunicación

UNIVERSIDAD DE CANTABRIA

Trabajo Fin de Grado

Sistema de Posicionamiento Basado en Fusión Sensórica y Gestión de la Orientación

(Positioning System Based on Sensor Fusion and Orientation Management)

Para acceder al Título de

Graduado en Ingeniería de Tecnologías de Telecomunicación

Autor: Alba Ruiz Gómez

Octubre - 2018

E.T.S. DE INGENIEROS INDUSTRIALES Y DE TELECOMUNICACION

GRADUADO EN INGENIERÍA DE TECNOLOGÍAS DE TELECOMUNICACIÓN

CALIFICACIÓN DEL TRABAJO FIN DE GRADO

Realizado por: Alba Ruiz Gómez
Director del TFG: Eugenio Villar Bonet
Codirector del TFG: Héctor Posadas Cobo
Título: "Sistema de posicionamiento basado en fusión sensórica y gestión de la orientación"
Title: "Positioning system based on sensor fusion and orientation management"

Presentado a examen el día: 29/10/2018

para acceder al Título de

GRADUADO EN INGENIERÍA DE TECNOLOGÍAS DE TELECOMUNICACIÓN

Composición del Tribunal:
Presidente (Apellidos, Nombre): Villar Bonet, Eugenio
Secretario (Apellidos, Nombre): Ugarte Olano, Iñigo
Vocal (Apellidos, Nombre): Sánchez Espeso, Pablo Pedro

Este Tribunal ha resuelto otorgar la calificación de: .....

Fdo.: El Presidente

Fdo.: El Secretario

Fdo.: El Vocal

Fdo.: El Director del TFG
(sólo si es distinto del Secretario)

Vº Bº del Subdirector

Trabajo Fin de Grado Nº
(a asignar por Secretaría)

Agradecimientos

En primer lugar me gustaría agradecer la confianza depositada en mi a Eugenio Villar, mi tutor, por darme la oportunidad de formar parte de este proyecto. Por otra parte y muy especialmente a Hector Posada, por el seguimiento y el apoyo durante el mismo, por darme la fuerza para no abandonar a pesar de las dificultades encontradas por el camino.

A los integrantes del Grupo de Ingeniería Microelectrónica de la Universidad de Cantabria, por el buen ambiente de trabajo y por echar una mano siempre que fue necesario.

Por último a mi familia, a Mario y a mis amigos, los cuales forman parte de mi día a día y me han dado un apoyo inestimable en todo momento.

Palabras clave

IMU, Oculus rift, Acelerómetro, c++, seguimiento posicional, fusión sensórica, cuaterniones, gestión orientación, giroscopio, magnetómetro

Keywords

IMU, Oculus rift, Accelerometer, c++, positional tracking, sensor fusion, quaternions, orientation management, gyroscope, magnetometer

Índice general

Índice de figuras

Capítulo 1

Introducción

Durante los últimos años, una de las innovaciones por la que muchas compañías de diversos sectores están apostando es el cambio en la manera en la que percibimos nuestra realidad. Las principales compañías tecnológicas del mundo están trabajando con todo tipo de gafas y wearables[1] para sumergirnos en un entorno virtual. Todo ello es posible debido a que los sistemas embebidos continúan experimentando un crecimiento exponencial, la Ley de Moore [2] sigue vigente.

Actualmente muchas son las empresas que se introducen en el mundo de los *'Sistemas de Visión 3D'*. Además de interesarse por esta tecnología, son muchas las inversiones en el desarrollo de la misma por su posible proyección en el futuro. Las empresas que están empezando a trabajar en el mundo de los *'Sistemas de Visión 3D'* pertenecen a sectores tan variados como la medicina, robótica, industria, mundo del arte, etc. Las aplicaciones son muy diversas, van desde el uso de un sistema como gadget a la hora de realizar una operación dentro de un quirófano, hasta utilizarlo en entornos industriales para optimizar procesos.

Hace pocos años imaginarse que se iba a alcanzar esta tecnología se trataba de algo surrealista, a pesar de ello, ya eran muchas las películas o cómics que se atrevían a aventurar este hecho. En aquel entonces resultaba algo utópico, pero a día de hoy, el hecho de adentrase en un mundo de fantasía ya es una realidad tangible.

[1] Wearables: dispositivo electrónico que se coloca en alguna parte de nuestro cuerpo.

[2] La ley de Moore expresa que aproximadamente cada dos años se duplica el número de transistores en un microprocesador.

Debido al gran interés suscitado por estos sistemas en las empresas, la introducción de estas nuevas tecnologías en el día a día se haya producido a una velocidad vertiginosa para el usuario de a pie. Cuando apenas la sociedad se había familiarizado con el término realidad virtual, surgieron otros dos nuevos conceptos: la realidad aumentada y, más recientemente, la realidad mixta.

A modo de resumen, para clarificar brevemente cuales son las diferencias entre las mismas, se podría decir que la realidad virtual genera mundos totalmente inexistentes, la realidad aumentada combina elementos inexistentes con otros que sí están ahí, y la mixta es una mezcla entre ambas. En capítulos posteriores 2, se realizará un análisis más en profundidad de dichos conceptos.

1.1. Motivación

Las simulaciones tridimensionales actuales rara vez tienen en cuenta el entorno físico en el que se opera. Esto puede causar problemas a la hora de interactuar con el ambiente. Para estar al tanto del entorno físico del usuario, debe crearse un sistema que determine la posición de este en el espacio en cada instante de tiempo.

Este trabajo se engloba en un sistema y un método patentado por la UC que obtiene la localización espacial de objetos o individuos en un escenario bajo las condiciones del ambiente, ya sea en espacios interiores como exteriores.

Uno de los bloques de este proyecto es el sistema de posicionamiento. Se trata de un sistema de posicionamiento dual, esta dupla está compuesta por un lado por la fusión de la sensórica y por otro por los marcadores luminosos. Es un sistema dual puesto que el seguimiento posicional absoluto, del tipo que se necesita para una MR[3] adecuada, no se puede lograr usando una unidad de medida inercial (IMU) sin un marco de referencia 3D externo. Cuando se realiza un seguimiento de la posición en función de los datos del acelerómetro utilizando el cálculo de cuentas, la deriva se acumula de forma cuadrática,

[3]MR: *mixed reality*, realidad mixta

lo que significa que la velocidad de la deriva aumenta proporcionalmente al tiempo transcurrido.

El sistema de posicionamiento basado en marcadores luminosos ya se encuentra desarrollado [1]. Mi aportación al proyecto global es esencialmente, obtener la aceleración del dispositivo con la mayor precisión posible a partir de la fusión sensórica. De esta forma, se puede obtener la posición aproximada del sujeto dentro de la habitación al conocer la posición inicial y las aceleraciones posteriores. La unión de ambos métodos de calculo de la posición permitirá obtener unos resultados óptimos.

Mi segunda aportación al proyecto global, se trata de la gestión de la orientación. A la hora de gestionar la orientación que en el caso anterior, no sucede lo mismo; la deriva lineal (no cuadrática) en la dirección de las lecturas del giroscopio se puede corregir empujando la estimación hacia el marco de referencia fijo establecido por la dirección de la gravedad (."arriba") y la dirección al norte magnético medida por el magnetómetro. Por lo que los resultados en si mismos son válidos.

Para realizar ambas tareas, se ha empleado la sensórica de las *Oculus Rift DK1*. Las Oculus Rift cuentan con la *IMU, unidad de medición inercial*. Esta se trata de un sistema electrónico autónomo que mide e informa acerca de la velocidad, orientación y fuerzas gravitacionales de un aparato, usando una combinación de un acelerómetro y giroscopio.

1.2. Estructura del documento

A continuación, se va a detallar cual es la organización del documento.

En el primer capítulo se ha recogido una breve introducción al tema que se va a tratar así como la motivación del proyecto.

En el segundo capítulo se presenta un estudio del Estado del Arte. De esta forma nos permite situarnos en el contexto en el cual se engloba el proyecto. Se muestra un pequeño análisis y comparación de los diferentes tipos de tecnologías que sumergen al usuario en entornos virtuales, la realidad virtual *(VR, virtual reality)*, la realidad aumentada *(AR, augmented reality)*

y la realidad mixta *(MR, mixed reality)*. Además, se va a realizar un análisis de las tecnologías competidoras. En este caso, los dispositivos móviles que cuentan con tecnología de realidad mixta.

El tercer capítulo recoge la arquitectura del sistema. En primer lugar, se presenta el proyecto global en el cual se engloba este trabajo. Además, se expone el hardware que se va a utilizar a lo largo del proyecto y su sensórica. Por otra parte, se detallan los fundamentos teóricos necesarios para el desarrollo del proyecto.

Por otra parte, en el cuarto capítulo comienza el análisis de mi aportación al proyecto global. Este recoge como se extraen los datos de la sensórica. Para ello, se relata la calibración del hardware, eliminación de la aceleración causada por la gravedad y eliminación del ruido de la sensórica. En último lugar, se presentan los resultados obtenidos.

En el quinto capítulo se describe la siguiente aportación al proyecto, la gestión de los resultados en el entorno gráfico. Para ello se relata como ser realiza la adaptación de los drivers de las Oculus Rift para extraer los datos, así como la gestión de estos en el entorno gráfico. Además, se detallan los resultados obtenidos.

Por último, en el sexto capítulo, se encuentran las conclusiones y la líneas futuras.

Capítulo 2

Estado del arte

A la hora de entender el por qué del proyecto es importante conocer algunos conceptos. Estos se tratan de términos que están íntimamente relacionados y pueden llevar a equivoco si no se comprenden sus diferencias claramente. A continuación, se detalla las características mas relevantes de todos ellos. Además, este capítulo también recoge el entorno en el cual se desarrolla el proyecto. En este caso se va a hacer un análisis de los sistemas de posicionamiento de los terminales móviles, los cuales se tratan de uno de los competidores más importantes de nuestro método.

2.1. Diferencias entre la realidad virtual, aumentada y mixta

A pesar de que el término *realidad virtual (VR)* tiene más de 75 años, su significado se ha ido modificando a lo largo del tiempo en función de las tecnologías disponibles. El concepto de *"virtual reality"* está normalmente relacionado con *inmersión*, es decir, la sensación de estar dentro de un mundo virtual el cual está cambiando tal y como se mueve el usuario. Por ello, la *realidad virtual* también es conocida como *entorno multimedia inmersivo* o *realidad simulada por ordenador*. Esta tecnología permite simular una experiencia sensorial en un espacio real o imaginario. Se sustituye el entorno en el cual se encuentra el usuario, por otro generado de forma digital. Los usuarios se sumergen en este *"ambiente artificial"* con unas gafas o un casco que bloquean la visión y, en ocasiones, el oído, por lo que se encuentran su-

mergidos en un mundo puramente virtual. Existen dos métodos para crear la experiencia inmersiva, entorno virtual asistido por ordenador (CAVE) y gafas de realidad virtual.

Un CAVE es una habitación en la cual el entorno virtual se crea proyectando dicho entorno en las paredes, techo y suelo para mejorar la experiencia inmersiva. La desventaja de este método se trata de su coste. El coste limita la aplicabilidad de la tecnología, por lo que solo se puede encontrar en reducidos entornos específicos, como el desarrollo de productos o prototipado y planificación colaborativa en sectores como la construcción.

El segundo de los métodos, las gafas de realidad virtual, tiene un precio menor por lo que permite que muchas más personas puedan acceder a dicha tecnología. El desarrollo especialmente de las pantallas de cristal líquido (LCDs) han permitido una bajada de precio. El número de aplicaciones de este dispositivo han aumentado drásticamente. Aunque la primera, por la que surgió esta tecnología, y en la actualidad la más importante son los videojuegos. Para que ese entorno virtual cambie tal y como el usuario se mueva es necesario contar con unas gafas que contengan un giroscopio que sea capaz de detectar la orientación en cada instante de tiempo. De esta forma, la imagen que se proyecta al usuario será el punto al que está mirando.

La realidad virtual no se limita solamente a los sectores del ocio y del entretenimiento. Uno de los factores que hace muy especial esta tecnología es la capacidad de adaptación en diferentes disciplinas, como por ejemplo la medicina. Se trata de una tecnología muy prometedora en este sector puesto que está abriendo muchas posibilidades. Algunos ejemplos son la formación de estudiantes, operaciones de cirugía o tratamiento de enfermedades.

La formación ya sea dentro del sector de la medicina como en el resto campos se trata de una de las aplicaciones más prometedoras. En el campo de la educación, esta tecnología ofrece muchas opciones, por ejemplo con las gafas de realidad virtual el alumnado puede estar en un lugar virtual en función de lo que seleccione el profesorado para poder aprender de una forma más realista.

Algunas gafas en lugar de utilizar el display interno usan la pantalla del móvil. Para que sea posible, la pantalla del móvil se divide en dos partes, de esta forma cada parte de la pantalla la vé uno de los ojos. Así es como se crea el efecto 3D debido al fenómeno de la visión estereoscópica[1]. Por ello se evita

[1]La capacidad que tiene el hombre de integrar las dos imágenes que está viendo en una sola por medio del cerebro. Para hacerla posible, nuestro cerebro analiza los datos que

electrónica adicional, hecho que tiene como consecuencia que este hardware tenga un precio asumible.

Por otra parte, se encuentra la *Realidad Aumentada (AR)*. Esta se trata de una versión mejorada de la realidad creada por el uso de la tecnología para superponer información digital a una imagen. Por ello, la imagen original se ve enriquecida con información adicional. Proporciona imágenes en vivo del entorno real pero algunos elementos de este se "aumentan" con estímulos sensoriales generados por ordenador, como sonido, vídeos, gráficos y otros tipos de datos. Con la realidad aumentada, el usuario siempre verá el mundo físico que lo rodea, pero con una capa virtual añadida. La principal diferencia que existe respecto a la *VR, realidad virtual,* es que, en esta ocasión, no se obstruye nuestro sentido de la vista, sino que añadimos información, normalmente meta-información[2]. Las gafas utilizadas en el caso de la realidad aumentada son diferentes. Como su función es "añadir" información a la escena, cuentan con pequeños proyectores que muestran la imagen generada por ordenador en la superficie de las gafas. De esta forma el usuario ve ambas imagenes superpuestas, la imagen real y la "añadida".

En el caso que nos atañe, resulta de especial interés la *Realidad Mixta (MR)*. Esta integra objetos digitales en un entorno real. La realidad aumentada y mixta comparten muchas similitudes, pero la mixta integra todos los elementos con más precisión y ofrece mayor interacción con el usuario. En la MR lo que se hace ya no es superponer información sobre el mundo real, sino fusionar el mundo físico con el mundo digital. Esto quiere decir que, si tenemos un elemento, como puede ser una silla modelada en 3D, vamos a poder colocarla en el mundo físico y esa silla va a "ser consciente" del mundo que le rodea: va a entender dónde está el suelo y si pasa alguien por delante, va a tapar dicha silla. Esto no sucedía con la AR, por lo que ahora la sensación va a ser mucho más inmersiva: le va a afectar la iluminación del entorno y todo se va a ir adaptando de forma que se pueda llegar a tener un mundo indistinguible que mezcle lo físico y lo digital. Para obtener más ventajas de esta tecnología es importante que el usuario sea capaz de moverse con libertad en el espacio. Por lo que el correcto posicionamiento del usuario en el espacio de la MR es un componente esencial. La tecnología de posicionamiento debe ser lo más flexible posible. El libro que recoge el método de posicionamiento basado en marcadores luminosos [1] se refiere a este término como *Realidad Recreada*[3].

recibe de los dos ojos y genera una imagen única tridimensional.

[2]Los metadatos, literalmente «sobre datos», son datos que describen otros datos.

[3]RR: las gafas 3D equipadas con cámaras dan un control completo de la imagen sintética

2.2. Método de posicionamiento de los dispositivos móviles

Es importante realizar un estudio de las tecnologías actuales para conocer cual es el entorno en el cual se va a desarrollar el proyecto. Para ello se ha realizado un estudio de los métodos de posicionamiento utilizados por los dispositivos móviles en la actualidad.

Se puede decir que el auge de esta tecnología surge en 2016 con el lanzamiento internacionalmente de Pokémon GO. Hoy en día pocos son quienes desconocen el fenómeno que supuso la aplicación desarrollada por Niantic. El boom que alcanzó Pokémon Go en 2016 fue espectacular, rompió todas las previsiones y expectativas. El éxito del juego Pokémon Go y la presentación de las lentes Microsoft Hololens en el panorama tecnológico, trajo la realidad aumentada a primera línea. En la actualidad, esta nueva tecnología en los dispositivos móviles se trata de un hecho constatable, tanto Android como IOS ya han desarrollado sistemas que cuentan con la misma.

Figura 2.1: Icono ARKit y ARCore

En junio de 2017 Apple sorprendió con su propia librería para realidad aumentada: ARKit. La forma de realizar la fusión entre el entorno digital y la realidad, permite colocar objetos generados por el dispositivo Apple sobre la imagen de la cámara y sin que se pueda apreciar ninguna diferencia. Esto da una sensación de interacción con el mundo real llevando la utilidad de las aplicaciones móviles a un nivel más alto. Este fenómeno se basa en una tecnología denominada **odometría visual inercial (VIO)**.

La **odometría visual inercial (VIO)** se basa en el reconocimiento de las características más notables de las imágenes, fusiona la información de los

proporcionada al usuario generada como una combinación de realidad y RV.

sensores de movimiento con la cámara de los dispositivos iOS. El ARKit es capaz de interpretar las imágenes que capta la cámara y realiza una detección de características, esta consiste en reconocer ciertos puntos de interés en una imagen, llamados *keypoints*, (puntos clave o características, en castellano). Estos puntos pueden tratarse de: bordes, esquinas, manchas. Cuando ya se han detectado los puntos clave de la imagen captada por la cámara estéreo se realizan comparaciones para detectar diferencias en las posiciones a través de todas las imágenes del vídeo y lo mezcla con la información de los sensores. El problema que puede surgir en esta odometría es cuando circula por entornos muy homogéneos como pasillos en los que no tiene puntos singulares en los que fijarse. Con ayuda de estos puntos clave se es posible obtener un mapa 3D del entorno y calcular las distancias que hay entre los diferentes objetos desde la posición de la propia cámara y el dispositivo. También, utilizará sensores de luz para poder aplicar la cantidad apropiada a los objetos virtuales.

Por otra parte, se encuentra la tecnología desarrollada por el competidor más importante de Apple, Google. Esta tecnología desarrollada por Google se denomina ARCore es la nueva tecnología de realidad aumentada que competirá de tú a tú con el ARKit de Apple. Funciona de manera muy similar al ARKit, utilizando la cámara, los acelerómetros, el giroscopio y la información contextual del dispositivo móvil, ARCore realiza el mapeo del entorno a medida que se mueve el dispositivo. El software selecciona características visuales en el entorno, *keypoints*, rastrea y coordina esos datos con información de los sensores de inercia. Al igual que ARKit, la cámara también se utiliza para hacer referencia a la iluminación promedio en el área.

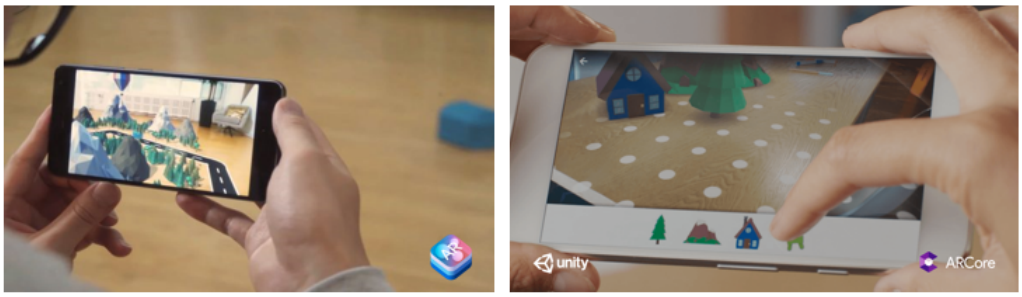

Figura 2.2: Ejemplo aplicación desarrollada mediante ARKit y ARCore

Capítulo 3

Arquitectura del sistema

En este capítulo se pretende dar una visión general de lo que el proyecto se refiere; además de describir el hardware utilizado así como los fundamentos matemáticos en los cuales se engloba.

3.1. Sistema global

El presente proyecto se engloba en un método patentado por la UC que obtiene la localización espacial de objetos o individuos en un escenario bajo las condiciones del ambiente. Este sistema se trata de un sistema de posicionamiento dual, es decir, para hallar la posición del usuario son dos los métodos que obtienen dicha posición de manera independiente y los resultados se van autoajustando para tener un sistema más robusto.

Tras haber realizado un análisis de la tecnología en la que se basan los sistemas de realidad aumentada actuales, se detectó una posible mejora muy significativa. Estos sistemas de realidad aumentada se basan en la detección de puntos característicos de las imágenes obtenidas a través de la cámara, *keypoints*. Esto produce que el nivel de cómputo necesario para procesar toda la información obtenida por las cámaras, sea muy alto. Como consecuencia, produce un aumento del consumo energético necesario para que el sistema esté en marcha. Una de las mejoras que se introducen con el sistema anteriormente planteado, es eliminar este paso, ya no se van a tener que localizar puntos característicos de la imagen, sino que estos puntos ya están determinados en la habitación y se van a tratar de leds. El sistema deberá localizar

dichos leds en las imágenes capturadas por las cámaras y a partir de estos datos y de los valores obtenidos por la sensórica, se determinará la posición del usuario dentro de la habitación.

En conclusión, el procedimiento cambia pero el objetivo es el mismo. Se pretende crear una simulación gracias a un casco de realidad virtual que determine la posición del usuario en su entorno. Una de las posibles aplicaciones serían crear hologramas virtuales para hacer conferencias a distancia, o para desarrollar videojuegos que tengan en cuenta el entorno real del usuario.

Para lograr este objetivo, se decidió desarrollar un programa en C ++. Dentro del sistema global se cuenta con una cámara, LEDs como marcadores específicos en las paredes de la habitación y el casco de realidad Virtual Oculus Rift. Dicho proyecto esta dividido en los siguientes cuatro bloques:

- **LED_Detection**: bloque del sistema encargado de la detección de LEDs en la imagen la cual es devuelta por la cámara.
- **Space**: es una de las entradas del sistema complementaria, se trata de la creación de un modelo 3D del espacio en el que el usuario interactua.
- **Pos_Calculation**: bloque que recoge el cálculo de la posición del usuario a través de los datos recabados por la sensórica del casco de realidad virtual junto con el sistema de detección de Leds.
- **Image_Syn**: es el último bloque del sistema, se trata de la creación de imágenes visibles a través del casco de realidad virtual.

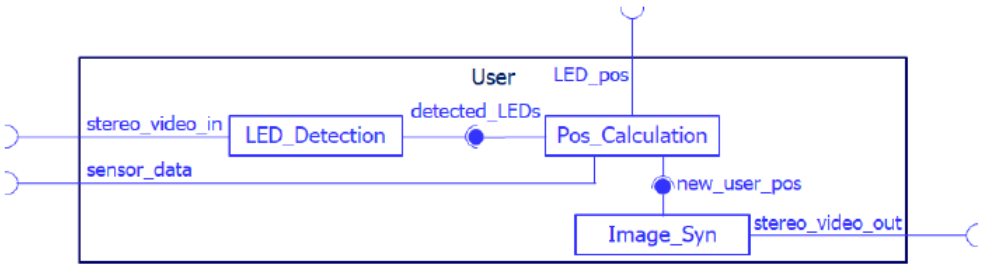

Figura 3.1: Esquema del proyecto global

Esta figura muestra los diferentes bloques, las entradas del sistema así como la salida esperada de este. Este programa utilizará datos de los sensores de Oculus Rift y el sistema de detección LED para determinar la posición.

Tal y como se dijo con anterioridad, se desean monitorizar los movimientos del usuario en una sala. En nuestro caso, la IMU, *Unidad de Medición Inercial*, se encargará de proveer al sistema de una referencia de orientación. El bloque *Pos_Calculation* se trata de un sistema de posicionamiento dual basado en la fusión de la sensórica y marcadores luminosos. El posicionamiento mediante la IMU va a trabajar simultáneamente con otro sistema de posicionamiento, es decir, se puede decir que uno y otro se complementan entre sí.

El primer sistema de posicionamiento ya desarrollado que compone el proyecto global, se trata de un sistema de posicionamiento basado en la detección de marcadores mediante una cámara estéreo [1]. Como se puede observar en la *Figura 3.2*. El sistema esta formado por los siguientes componentes y funciona tal y como se muestra en el esquema:

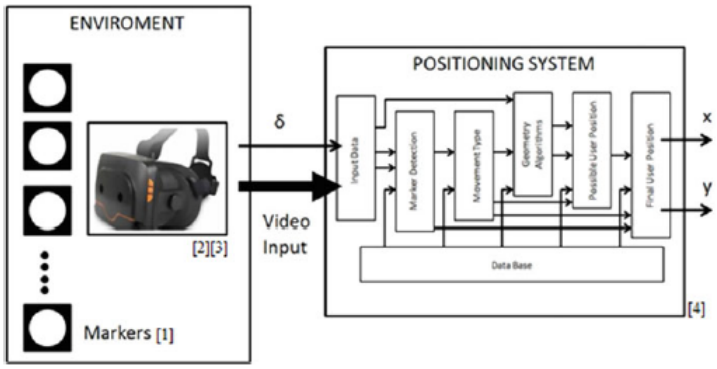

Figura 3.2: Esquema del sistema de posicionamiento basado en marcadores luminosos

1. Fuentes de luz que se utilizan como marcadores para calcular la posición relativa

2. Una cámara estéreo que muestre estos marcadores en la imagen de la escena

3. Un dispositivo que mida ángulos (como un giroscopio) para obtener el ángulo de rotación en cada instante

4. Un procesador de señal digital, el cual utiliza las coordinadas almacenadas y los parámetros de salida obtenidos a partir de la cámara estéreo y el angulo de medida para determinar cual es la posición 3D en el ambiente.

Por otra parte, el segundo sistema de posicionamiento, se trata del basado en la IMU. Esta es una de las tareas desarrolladas en este proyecto, por lo que será detallado en los próximos capítulos.

3.2. Oculus Rift, casco de Realidad Virtual

Tras tener claro como esta organizado el proyecto, es importante conocer el hardware con el cual se va a trabajar. Para desarrollar el proyecto global se contó con el casco de VR *Oculus Rift*. Se utilizó este hardware debido a su bajo coste. En comparación con otros dispositivos con características similares se trataba del mejor en relación calidad-precio. A continuación, se va a realizar un análisis en términos de sus características técnicas.

Las *Oculus Rift* son un casco de realidad virtual que fue desarrollado y fabricado por Oculus VR y llego al mercado el 28 de marzo de 2016. Se trata de un sistema *'Head Mounted Display Tethered'*, ya que las gafas están atadas a un hardware de procesamiento externo, el cual puede tratarse de un ordenador o una consola.

El equipo es un sistema de realidad virtual donde el casco es una parte del sistema que se ajusta a la cabeza de los jugadores y funciona como un monitor y control. Los usuarios, al mover la cabeza, controlan al personaje que aparece en pantalla y pareciera como si estuvieran dentro del mundo virtual del juego. Oculus Rift se conecta a un ordenador, en donde se genera la mayoría del procesamiento de las imágenes que aparecen frente a ellos.

El efecto 3D se genera por medio de una pequeña pantalla OLED Pentile de 7 pulgadas, y un sistema de lentes que crea una ilusión casi perfecta. Además, para que la experiencia sea mucho más inmersiva cuentan con unos auriculares integrados que proporcionan un efecto de audio 3D.

Dentro del visor o las gafas hay dos pantallas por donde se proyecta la imagen que el usuario percibe. Un par de lentes se encuentran en la parte superior de las pantallas con el objetivo de enfocar y ajustar la imagen para cada ojo y crear una imagen en 3D estereoscópica. Esta visión 3D estereoscópica tiene una excelente profundidad, escala y paralaje. A diferencia de los televisores 3D o películas 3D, la visión se consigue mediante la pre-

sentación de imágenes únicas y paralelas para cada ojo. Esta es la forma en que los ojos perciben las imágenes en el mundo real, creando una experiencia única y mucho más natural y cómoda.

Además, estas gafas cuentan con sensores que monitorean los movimientos de la cabeza del usuario para tener mayor precisión proyectando la imagen. El sistema de seguimiento posicional, llamado *Constelación*, se realiza mediante un sensor infrarrojo fijo USB que capta la luz que emiten los LED IR integrados en la pantalla montada en la cabeza. El sensor normalmente se sienta en el escritorio del usuario. Esto crea espacio en 3D, lo que permite al usuario usar el Rift mientras está sentado, de pie o caminando. Además, para monitorizar los movimientos de cabeza del usuario cuenta con una *Unidad de Medición Inercial, IMU*. Esta es la parte del hardware más relacionada con el proyecto.

Figura 3.3: Oculus Rift

Tal y como se ha dicho anteriormente, el funcionamiento de las Oculus se trata de la visión 3D estereoscópica. Este se trata de cómo vemos la tercera dimensión gracias a los dos ojos. Cuando miramos con los dos ojos, en cada ojo recibimos una imagen diferente, es decir, toma información del medio distinta. Sin embargo, cuando miramos, vemos una sola imagen de las cosas y no dos distintas, lo que nos dificultaría mucho desarrollar nuestra vida con normalidad. La capacidad que tiene el hombre de integrar las dos imágenes que está viendo en una sola por medio del cerebro es la que nos permite disfrutar de la llamada visión estereoscópica. Para hacerla posible, nuestro cerebro analiza los datos que recibe de los dos ojos y genera una imagen única tridimensional.

3.3. Unidad de medición inercial, IMU

Tras conocer un poco más las Oculus Rift resulta interesante profundizar en la parte de la sensórica en la cual se va a centra el estudio, esta es la *Unidad de Medición Inercial*.

Una *Unidad de Medidas Inerciales (IMU o UMI en castellano)* es en general un sistema cerrado que se usa para detectar la orientación, localización y movimiento, usando una combinación de acelerómetros, giroscopios y magnetómetro.

Estos dispositivos fueron inicialmente desarrollados para su uso en aeronavegación, y concretamente para vehículos no tripulados. En estos vehículos era necesario el uso de la IMU puesto que al controlarse de forma autónoma era indispensable conocer la trayectoria que seguía el vehículo.

El uso de estos dispositivos por primera vez en la historia fue para fines militares. Fueron desarrollados por científicos alemanes durante la segunda guerra mundial para incorporarlo a sus misiles V-2. Estos se trataban de los primeros misiles de largo alcance que lograron una gran precisión.

Después de la segunda guerra mundial el ritmo de desarrollo creció exponencialmente. Los sensores que formaban parte de estos dispositivos sufrieron grandes mejoras y la filosofía de montaje también cambió.

En la actualidad, las IMU se encuentran integradas en placas con tamaños y pesos muy reducidos (40x30mm y 6gr).

Este proyecto se basa en la extracción de datos de la IMU que se encuentra en las gafas de realidad virtual. Tal y como se comentó, normalmente este dispositivo usa una combinación de acelerómetros y sensores de velocidad angular (giroscopios) para conocer cómo se está moviendo y en qué posición se encuentra.

Al tener que obtener la posición en 3 dimensiones es importante que los sensores sean de tres ejes, estos sean totalmente ortogonales y que estén perfectamente alineados, Figura 3.4

Típicamente, una IMU detecta la aceleración y los cambios de orientación instantáneamente (i.e. ángulos de roll, pitch y yaw), además los integra para averiguar el cambio total sobre la posición inicial, Figura 3.5.

Para ello, cuenta con un acelerómetro. Un acelerómetro se trata de un

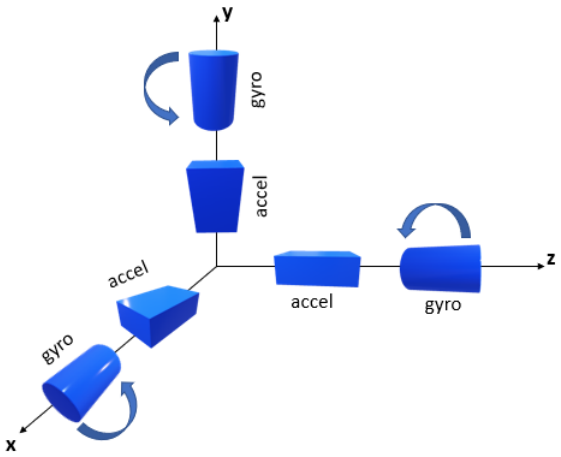

Figura 3.4: Disposición de los sensores de la IMU

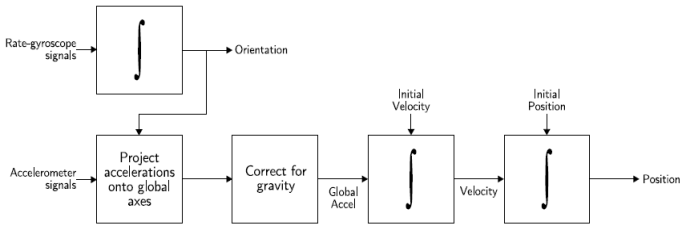

Figura 3.5: Algoritmo de navegación inercial [16]

dispositivo que se encarga de medir la aceleración a la que se encuentra sometido. Por lo que tiene unas unidades de m/s^2. Esta aceleración no se trata de la aceleración lineal directamente, sino que es la suma de muchos componentes: la aceleración lineal, que es la que resulta al experimentar el objeto un desplazamiento, la gravedad, la aceleración angular cuando el dispositivo se desplaza en una trayectoria curva, etc. Para realizar el cálculo de la posición el único componente de esta aceleración que resulta interesante es la aceleración lineal, por lo que el resto deberán de ser eliminados.

Una IMU cuenta con al menos dos sensores. Típicamente a parte del acelerómetro se encuentra el giroscopio. Un giroscopio se trata de un dispositivo que mide la velocidad de giro del eje en el que se encuentra, el resultado de estas medidas tiene unas unidades de $^o/s$ o rad/s. Por lo que si se quiere conocer el ángulo de giro total es necesario que se realice una integración de todas las medidas.

Otro de los sensores con los que puede contar una IMU es el magnetóme-

tro. Un magnetómetro se trata de un dispositivo que mide el campo magnético. Este sensor se trata del menos problemáticos si se encuentra en un entorno adecuado, es decir, si no tiene cerca fuentes de campo magnético diferentes al terrestre. Esto se debe a que sus medidas no son relativas sino absolutas con el tiempo.

La combinación de estos sensores permite obtener una única medida con mejores prestaciones que cada uno de ellos por separado.

Las IMUs suelen tener un error acumulado o deriva. Esto se debe a que una IMU esta sumando continuamente los cambios detectados en la posición, por lo que cualquier error en esta medida es acumulado. Esto da lugar a la "deriva", o un error creciente entre la posición hallada por la IMU y la posición real de ésta. Por ello, estas son normalmente uno de los componente de un sistema de navegación. Otros sistemas tales como los GPS (usados para corregir el término de deriva en la posición), un sistema barométrico (para la corrección de la altitud), o un compás magnético (para la corrección de la orientación) compensan las limitaciones de una IMU. Todos los sistemas que tienen la función de determinar la posición tienen sus propios defectos, por ello es importante que trabaje simultáneamente varios sistemas, de esta forma estos defectos son compensados entre ellos.

Oculus Rift cuenta con tres versiones, dos de ellas diseñadas para desarrolladores; y la última y más reciente, la versión para el consumidor. Las dos primeras, las versiones de desarrollo, son conocidas por las siglas DK o "Development Kit" (Kit de desarrollo) y el número de versión correspondiente.

Dentro del departamento se contaba tanto con las *Oculus Rift DK1*, versión de desarrollo; como con las *Oculus Rift CV1*. A continuación, se va a profundizar en el estudio de la sensórica de cada una de ellas.

3.3.1. Análisis en profundidad de la IMU

Los dos tipos de se tratan de módulos con tecnología *MEMS, Sistemas Microelectromecánicos (Microelectromechanical Systems, MEMS)*. Estos se refieren a la tecnología electromecánica, micrométrica y sus productos, y a escalas relativamente más pequeñas. A continuación se va a realizar un análisis de cada uno de ellos en profundidad, prestando especial interés al

modelo *Oculus Rift DK1*.

3.3.1.1. DK1: Invensense MPU 6000

El *MPU-6000* es el módulo MEMS que forma parte de las *Oculus Rift DK1*. Se trata de un dispositivo de seguimiento de movimiento con 6 Axis, *InvenSense*. Combina un giroscopio de 3 ejes y un acelerómetro de 3 ejes junto con un procesador de movimiento digital (DMP) a bordo, todo en un solo dispositivo.

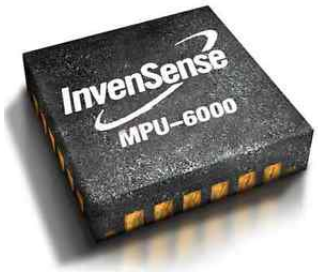

Figura 3.6: Invensense MPU6000

El dispositivo de seguimiento de movimiento incorpora algoritmos MotionFusion y calibración que también accederán a sensores externos a través del bus maestro auxiliar I2C. Esto permite a los dispositivos recopilar un conjunto completo de datos del sensor sin la intervención del procesador del sistema.

Este dispositivo es ampliamente utilizado en teléfonos inteligentes, tabletas y tecnología de sensor portátil. Esto se debe a todas las funcionalidades que posee y las cuales resultan muy útiles en estas aplicaciones, como las que se describen a continuación.

- Filtros digitales programables por el usuario para giroscopio, acelerómetro y sensor de temperatura
- Sensor de temperatura de salida digital
- 9-Axis MotionFusion por el procesador de movimiento digital en el chip (DMP)
- Bus auxiliar I2C y SPI maestro para leer datos de sensores externos

- Sensibilidad mínima del eje cruzado entre el acelerómetro y los ejes del giroscopio
- Estructura MEMS sellada herméticamente y unida al nivel de la oblea
- 10,000 g tolerantes a golpes
- Interrupciones programables por el usuario
- Detección y señalización de orientación
- Interfaz serial SPI de 1MHz para comunicarse con todos los registros
- Interfaz serial SPI de 20MHz para leer el sensor e interrumpir registros

El precio de este dispositivo ronda los 10€.

3.3.1.2. CV1: Bosch BMI055

El *BMI055* es un sensor inercial de 6 ejes que forma parte de las *Oculus Rift CV1*. Una de sus características principales, es que permite medir muy poco ruido de velocidades y aceleraciones angulares en 3 ejes perpendiculares y así detectar inclinación, movimiento, choque y vibración en teléfonos celulares, dispositivos de bolsillo, periféricos de computadora, interfaces hombre-máquina, controladores remotos y juegos.

Figura 3.7: Bosch BMI055

Además, el *BMI055* integra una multitud de características que facilitan su uso, especialmente en el área de aplicaciones de detección de movimiento,

como la medición de la orientación del dispositivo, juegos, HMI y control del navegador de menú. A continuación se van a detallar algunas de sus posibles aplicaciones.

- Monitoreo de la actividad, conteo por pasos
- Navegación
- Medición de vibraciones, también para amortiguación activa
- Seguimiento de trayectorias en seis dimensiones
- Detección plana, detección de tap, menú de desplazamiento
- Compensación de inclinación para compás electrónico
- Administración avanzada de energía para dispositivos móviles
- Detección de golpes y caída libre
- Estabilización de imagen

Este dispositivo tiene un precio en el mercado de aproximadamente de 6€.

3.3.2. Hardware utilizado

El primer punto a tratar después de haber analizado en profundidad todas las herramientas disponibles para el trabajo, era decidir cual iba a ser el hardware que se iba a utilizar para el desarrollo de la tarea.

Tras realizar este análisis, se decidió trabajar con las *Oculus Rift DK1* y por tanto con la IMU *Invensense MPU 6000*. Este hecho fue incentivado por numerosos motivos. El primero de ellos, ya que las *Oculus Rift CV1* se tratan de una versión de usuario, por lo cual no tiene soporte en Linux. Esto provoca que a la hora de desarrollar cualquier proyecto supondría un problema. Las otras gafas de realidad virtual que se encontraban disponibles en el laboratorio, la cual se trataba de la primera versión desarrollada por Oculus, eran las *Oculus Rift DK1*. Esta versión se trata de una versión para

desarrolladores. Esto tiene unas consecuencias muy relevantes, la primera de ellas y en contraposición con el otro modelo, es que este si que tiene soporte Linux. Como esta versión estaba pensada para desarrolladores resulta mucho mas sencillo realizar cualquier proyecto. Por otro lado, al tratarse de una versión de desarrolladores, ya existía un gran colectivo el cual había trabajado con este hardware. Este hecho produce que la cantidad de documentación que se encontraba disponible es mucho mayor. Por otra parte, para desarrollar el proyecto se partió de unos drivers específicos de las *DK1* por lo que suponía una ventaja evidente.

3.4. Fundamentos matemáticos en los que se basa proyecto

Para realizar el desarrollo de la tarea, muchos desarrollos se han apoyado en los fundamentos matemáticos que se describen a continuación.

3.4.1. Cuaterniones

En este apartado se va a abordar uno de los temas más importantes y más significativos del proyecto, la representación de la orientación mediante los cuaterniones. Una rotación puede ser representada matemáticamente desde diferentes perspectivas. En general, existen cuatro principales representaciones:

- Par Eje-Ángulo
- Ángulos de Euler
- Matrices de rotación
- Cuaternios.

De estas cuatro representaciones, la última de ellas, los cuaterniones, se trata del método más utilizado en la actualidad para la representación de

las rotaciones en 3D. Son útiles en aplicaciones de gráficos por computadora, robótica, navegación y mecánica orbital de satélites. Esto se debe a las ventajas que presenta este método frente a los otros tres anteriormente mencionados. Comparados con los ángulos de Euler, son más simples de componer y evitan el problema del bloqueo de cardán (gimbal lock[10]). Comparados con las matrices de rotación, son más eficientes y más estables numéricamente.

Como curiosidad, un conocido incidente de bloqueo de cardán (gimbal lock) sucedió en la misión *Apollo 11 Luna*, Figura 3.8 Se produjo la pérdida de un grado de libertad en una de tres dimensiones. En lugar de tratar de conducir los ejes más rápido de lo que podrían ir, el sistema simplemente se dio por vencido y congeló la plataforma. A partir de este punto, la nave espacial tendría que alejarse manualmente de la posición de bloqueo, y la plataforma tendría que realinearse.

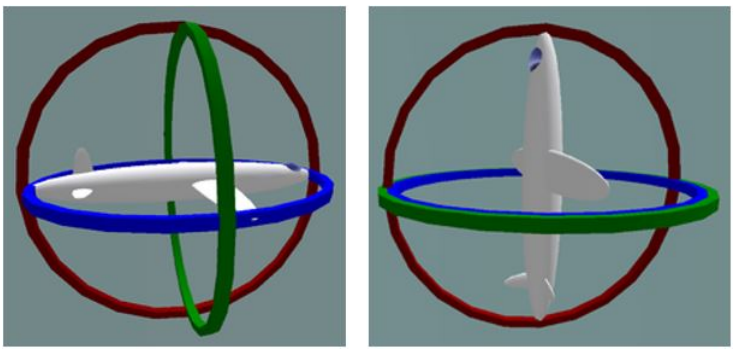

Figura 3.8: Bloqueo de cardán. A la izquierda situación normal: los tres ejes son independientes. A la derecha bloqueo de cardan: dos de los tres ejes se encuentran en el mismo plano, por lo que se pierde un grado de libertad

Los cuaternios fueron inventados por *Sir William Rowan Hamilton* en 1843. Hamilton estaba empeñado en generalizar los números complejos a tres dimensiones. Una de las motivaciones de Hamilton para buscar números complejos de dimensión n= 3, fue precisamente encontrar una descripción de rotación en el espacio correspondiente a los números complejos, donde una multiplicación corresponde a la rotación y aun escalamiento en el plano.

Los cuaterniones (también llamados cuaternios) son una extensión de los números reales. Son similares a la de los números complejos añadiendo las unidades imaginarias i, j y k a los números reales tal que:

$$i^2 = j^2 = k^2 = ijk = -1, \tag{3.1}$$

Los elementos *1, i, j y k* son los componentes de la base de los cuaternios considerado como un $\mathbb{R}$-espacio vectorial de dimensión 4. Una real y 3 imaginarias.

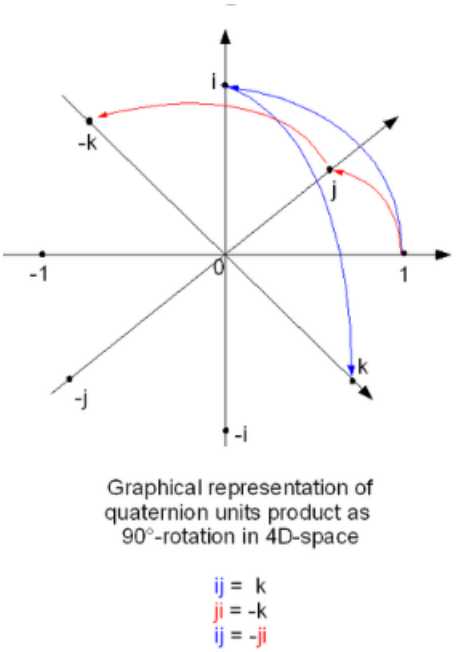

Figura 3.9: Representación gráfica de un cuaternion

Cada una de estas unidades imaginarias representará un giro de 180º, siendo su cuadrado un giro de 360º, de forma que tanto +1, como -1, representarán el mismo giro, aunque por caminos diferentes. El cuaternión i representa una rotación de 180º alrededor del eje X, el j sobre el eje Y, y el k sobre el Z. De esta manera, $i * i = -1$, representa una rotación de 360º alrededor del eje X.

Los cuaterniones pueden representar rotaciones 3D, escalados y reflexiones, sin embargo, no pueden representar traslaciones. Un cuaternio es usualmente representado como [s,v] con s $\in \mathbb{R}$ y v $\in \mathbb{R}^3$. Aquí *s* se llama la parte escalar, y v= (x,y,z) la parte vectorial.

Si se habla en términos de orientaciones, un cuaternión representa una rotación de un ángulo *tetha* alrededor del eje representado por un vector v.

Los cuaternios, gracias a las propiedades algebraicas que poseen, son útiles para representar rotaciones de un objeto respecto a otro.

Primeramente definimos un cuaternio que representa un giro de valor θ sobre un eje n como:

$$Quat = Rot(\theta, n) = (cos(\theta/2), nsin(\theta/2)) \tag{3.2}$$

Consideremos dicho eje como un vector unitario n =(n1,n2,n3) $\in \mathbb{R}^3$, $\|n\| = 1$. Una rotación por un ángulo θ, teniendo como eje de rotación el vector n, queda representada mediante la matriz:

$$R(\theta, n) = \begin{bmatrix} c + n_1^2(1-c) & n_1n_2(1-c) - sn_3 & n_1n_3(1-c) + sn_2 \\ n_2n_1(1-c) + sn_3 & c + n_2^2(1-c) & n_2n_3(1-c) - sn_1 \\ n_3n_1(1-c) - sn_2 & n_3n_2(1-c) + sn_1 & c + n_3^2(1-c) \end{bmatrix} \tag{3.3}$$

donde $c = cos\theta$ y $s = sin\theta$.

Otra representación de la matriz R es la siguiente. Al sustituir q = (q0,q1,q2,q3) por

$$\begin{aligned} q0 &= n1sen(\theta/2), \\ q1 &= n2sen(\theta/2), \\ q2 &= n3sen(\theta/2), \\ q3 &= cos(\theta/2), \end{aligned} \tag{3.4}$$

y teniendo en cuenta que:

$$n_1^2 + n_2^2 + n_3^2 = 1 \tag{3.5}$$

se obtiene la siguiente matriz de rotación:

$$R(q) = \begin{bmatrix} q_3^2 + q_0^2 - q_1^2 - q_2^2 & 2q_0q_1 - 2q_3q_2 & 2q_0q_2 + 2q_3q_1 \\ 2q_0q_1 + 2q_3q_2 & q_3^2 - q_0^2 + q_1^2 - q_2^2 & 2q_1q_2 - 2q_3q_0 \\ 2q_0q_2 - 2q_3q_1 & 2q_1q_2 + 2q_3q_0 & q_3^2 - q_0^2 - q_1^2 + q_2^2 \end{bmatrix} \tag{3.6}$$

Ambas matrices nos serán de extrema utilidad para los cálculos posteriores a la hora de gestionar la orientación.

Capítulo 4

Fusión sensórica

En este capítulo se recoge la primera de las aportaciones al proyecto global. Esta se trata de la gestión de la posición del usuario a partir de los datos obtenidos por la sensórica de las gafas. Tal y como se ha comentado con anterioridad se trata de un sistema de posicionamiento dual puesto que el posicionamiento absoluto a través de la IMU es imposible. Dentro de este proyecto se recoge la parte de posicionamiento realizado con ayuda de la IMU de las Oculus Rift. Para poder recabar los datos de las Oculus Rift se ha trabajado con unos drivers en C++. De esta forma se podrá trabajar con los datos obtenidos por la *MPU-6000*.

Los datos recabados con ayuda de la IMU, necesitan ser tratados. Todo este proceso está influido por problemas típicos de los sensores, como pueden ser mala calibración, ruido excesivo, etc., y por errores inherentes al sistema, como la no linealidad de las orientaciones, la deriva en el tiempo debido a medidas incrementales, etc. Estos datos, las aceleraciones y los ángulos de giro, no determinan la aceleración lineal del sujeto por lo que para ser capaces de conocer cuanto se ha desplazado el sujeto dentro de una habitación hay que enfrentarse a una serie de aspectos.

- Calibración de los sensores
- Ruido de la sensórica
- Eliminación aceleración causada por la gravedad

A continuación se describirá con detalle cada uno de estos aspectos y las soluciones aportadas a los mismos.

4.1. Calibración de la MPU-6000

En primer lugar, para poder obtener los datos de la sensórica de las Oculus Rift con una cierta coherencia es necesario realizar la calibración de las mismas. Este procedimiento es necesario realizarlo siempre y cuando se quieran recabar datos con cierta fiabilidad. Se trata de un procedimiento imprescindible ya que la sensórica propia es muy sensible a cambios tanto de temperatura como de presión. Por ello dependiendo del entorno en el cual se este trabajando es indispensable que se realice este tramite para poder obtener los resultados sin que estos se vean afectados por las variables externas.

Para realizar este proceso de calibración, dentro de los drivers de las propias gafas con los cuales se esta trabajando, existe una función la cual se encarga de actualizar los valores del fichero de calibración '*updatecalibration*'. A la hora de calibrarlas, es necesario que el sistema almacene los valores obtenidos por la sensórica en todas las posiciones del sistema. Debido a esto, es necesario que las Oculus Rift se roten y pasen por todas las posiciones para que la fiabilidad del proceso de calibración se incremente.

Para poder realizar el proceso de calibración contamos con dos variables de suma importancia. En primer lugar, '$recordRawMeasurements(false)$', almacenar las nuevas mediciones. Es decir, en el caso de que esta variable esté en '*true*' los valores del fichero de calibración se actualizan. Por otra parte, se encuentra '$rawMeasurementsDirty(false)$' esta variable tal y como su nombre indica, determina si las mediciones están '*sucias*, es decir, si necesitan ser las gafas calibradas para obtener unos valores con mayor fiabilidad. Por lo tanto, si esta variable se encuentra a '*true*' los drivers tomaran los datos del fichero de calibración, en caso contrario, las gafas no serán calibradas.

Tal y como se ha descrito, en el caso de que '$recordRawMeasurements$' tenga valor '*true*' el fichero de calibración se actualiza. Este se trata de un archivo con una extensión *.calib*, el cual almacena los datos recabados en la IMU tal y como se muestra a continuación.

El fichero de calibración está esencialmente dividido en dos partes. La primera de ellas se trata de la calibración del acelerómetro. En esta parte se encuentra la matriz de calibración del mismo, así como otros parámetros que nos serán de utilidad. Por otra parte, la segunda parte del fichero se encarga de calibrar el magnetómetro. De igual modo, se encuentra la matriz de calibración del magnetómetro y otros parámetros de interés. Gracias a

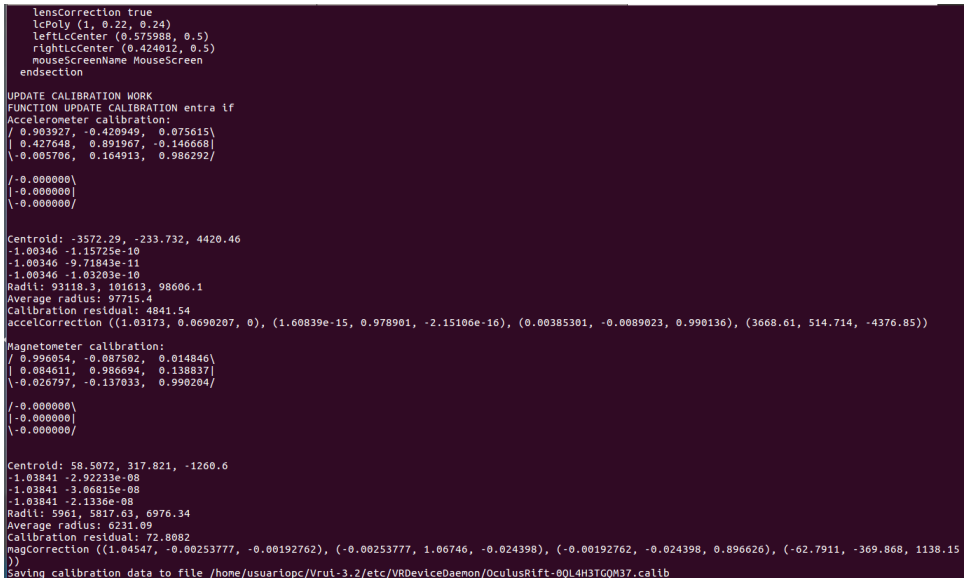

Figura 4.1: Esquema proyecto.

estos datos recabados los resultados obtenidos por la sensórica tendrán mayor veracidad.

4.2. Eliminación aceleración causada por la gravedad

La gravedad es un fenómeno natural por el cual los objetos con masa son atraídos entre sí. Básicamente, es la fuerza que actúa impidiendo que los cuerpos con masa floten y se mantengan unidos a La Tierra. Por lo que todo cuerpo con masa presenta este fenómeno inherente.

Se espera que si las gafas están posadas en cualquier posición estática, es decir, no se las aplica ninguna fuerza, el resultado de la aceleración aportado por el acelerómetro sea cero en todos sus ejes. Este concepto tan simple no es así en la realidad. Tal y como se ha descrito anteriormente, la gravedad afecta a todos los cuerpos con masa, pues este caso no iba a ser menos.

La aceleración de la gravedad afecta en el *eje vertical* de la tierra. El

principal problema surge ya que los ejes de referencia del acelerómetro y los ejes de la tierra no son los mismo. El acelerómetro tiene tres ejes, es decir, un punto de referencia marcado. Por ello, en función de como estén colocadas las gafas estos ejes también cambiaran la posición. Debido a esto resulta imposible eliminar el componente de la aceleración directamente. Para ser capaces de eliminar este fenómeno inherente de los cuerpos, es necesario que los ejes de las gafas correspondan a la perfección con los ejes de la tierra. En este punto es cuando entra en juego el segundo elemento de la IMU, el giroscopio.

El giroscopio se trata del segundo elemento que constituye la IMU, es un dispositivo mecánico que se encarga de medir la orientación. La orientación de un objeto nos indica su posición en un espacio 3D. La manera de representar esta orientación es mediante los cálculos matemáticos que nos permitan pasar del sistema de coordenadas del objeto respecto al sistema de coordenadas de referencia.

Los términos de orientación y de rotación a pesar de que matemáticamente se representan de la misma forma no son exactamente lo mismo. Se llamará orientación a la posición relativa del sistema de coordenadas del dispositivo respecto al sistema de coordenadas de referencia. Por otra parte, la rotación se trata del cambio de orientación de un instante a otro. Es decir, un objeto tiene una cierta orientación en un instante, y al aplicarle una rotación dicha orientación cambia.

A partir de los resultados de la orientación se obtienen los cuaterniones. Al tener un sistema de coordenadas móvil respecto a uno fijo, en tres dimensiones, es posible dar la orientación del sistema móvil en un momento dado con ayuda de estos cuaterniones. Se trata del método más eficiente usados para describir la orientación de un objeto en tres dimensiones. Los cuaterniones se explicarán con más detalle en el capítulo anterior 3.

4.2.1. Aplicación de los cuaterniones a la fusión de la sensórica

Para realizar la fusión de los datos del acelerómetro y giroscopio se realiza el siguiente procedimiento.

El vector *gAccel* se trata de una transformación de la variable *currentLinearAcceleration*. A esta variable se aplica la orientación que en este caso esta

expresada en forma de cuaternión. De esta forma se consigue que independientemente de la posición del objeto la variable de la gravedad se encuentre en el 'eje y' y sea posible eliminarse.

Paso 1: en primer lugar, se calcula la velocidad angular a través de los datos que obtenidos con el giroscopio.

```
currentAngularVelocity=Scalar(sensorData.samples.gyro)*Scalar
    (0.0001);
```

Paso 2: a continuación, se realiza el calculo de la orientación realizando una rotación de la velocidad angular, de esta forma obtiene el cuaternión orientación:

$$OrientacionActual = OrientacionPrevia + 0{,}001 * w$$

```
currentOrientation*=Rotation::rotateScaledAxis(
    currentAngularVelocity*Scalar(0.001));
```

Paso 3: tras haber obtenido el valor de la orientación representado por un cuaternión, se fusionan los datos del giroscopio con los del acelerómetro a través de la matriz de rotación. Esta matriz se obtiene a partir del cuaternión tal y como se muestra a continuación en el código. Coincide con la matriz de rotación anteriormente descrita en la teoría fundamental de los cuaterniones.

```
Vector gAccel=currentOrientation.transform(
    currentLinearAcceleration);
```

La rotación en 3D se representa mediante el operador cuaterniónico *(teoría Cuaterniones: 3)*. Definimos el operador cuaterniónico de rotación, ρ_q, asociado con el cuaternio q y el cual se aplica a un vector $v \in \mathbb{R}^3$ correspondiente al cuaternio puro v, por la ecuación:

$$w = \rho_q(v) = qvq^{-1} \tag{4.1}$$

Para obtener el vector de la aceleración fusionado con los valores del giróscopo, se va a aplicar el cuaternión que representa la orientación del sistema. La función $.transform$ es la siguiente:

```
Vector transform(const Vector\& v) const // Transforms a vector
{
Scalar wxvx=q[1]*v[2]-q[2]*v[1]+v[0]*q[3];
Scalar wxvy=q[2]*v[0]-q[0]*v[2]+v[1]*q[3];
Scalar wxvz=q[0]*v[1]-q[1]*v[0]+v[2]*q[3];
Vector result;
result[0]=v[0]+Scalar(2)*(q[1]*wxvz-q[2]*wxvy);
result[1]=v[1]+Scalar(2)*(q[2]*wxvx-q[0]*wxvz);
result[2]=v[2]+Scalar(2)*(q[0]*wxvy-q[1]*wxvx);
return result;
}
```

Si se desarrollan estas expresiones, se puede observar como el código se corresponde con el procedimiento matemático.

$a_{x,result} = a_x + 2 * q[1] * (q[0] * a_y - q[1] * a_x + q[3] * a_z) - 2 * q[2] * (q[2] * a_x - q[0] * a_z + q[3] * a_y);$

$a_{y,result} = a_y + 2 * q[2] * (q[1] * a_z - q[2] * a_y + q[3] * a_x) - 2 * q[0] * (q[0] * a_y - q[1] * a_x + q[3] * a_z);$

$a_{z,result} = a_z + 2 * q[0] * (q[2] * a_x - q[0] * a_z + q[3] * a_y) - 2 * q[1] * (q[1] * a_z - q[2] * a_y + q[3] * a_x);$

Se multiplican todos los términos:

$a_{x,result} = a_x + 2 * q[1] * q[0] * a_y - 2 * q[1] * q[1] * a_x + 2 * q[1] * q[3] * a_z - 2 * q[2] * q[2] * a_x + 2 * q[2] * q[0] * a_z - 2 * q[2] * q[3] * a_y;$

$a_{y,result} = a_y + 2 * q[2] * q[1] * a_z - 2 * q[2] * q[2] * a_y + 2 * q[2] * q[3] * a_x - 2 * q[0] * q[0] * a_y + 2 * q[0] * q[1] * a_x - 2 * q[0] * q[3] * a_z;$

$a_{z,result} = a_z + 2 * q[0] * q[2] * a_x - 2 * q[0] * q[0] * a_z + 2 * q[0] * q[3] * a_y - 2 * q[1] * q[1] * a_z + 2 * q[1] * q[2] * a_y - 2 * q[1] * q[3] * a_x;$

Se saca factor común a las aceleraciones:

$a_{x,result} = a_x * (1 - 2 * q[1] * q[1] - 2 * q[2] * q[2]) + a_y * (2 * q[1] * q[0] - 2 * q[2] * q[3]) + a_z * (2 * q[1] * q[3] + 2 * q[2] * q[0]);$

$a_{y,result} = a_x * (2 * q[2] * q[3] + 2 * q[0] * q[1]) + a_y * (1 - 2 * q[2] * q[2] - 2 * q[0] * q[0]) + a_z * (2 * q[2] * q[1] - 2 * q[0] * q[3]);$

$a_{z,result} = a_x * (2 * q[0] * q[2] - 2 * q[1] * q[3]) + a_y * (2 * q[0] * q[3] + 2 *$

$q[1] * q[2]) + a_z * (1 - 2 * q[0] * q[0] - 2 * q[1] * q[1])$;

Se construye la interpretación matricial de las expresiones anteriores:

$$\begin{bmatrix} a_{x,result} \\ a_{y,result} \\ a_{z,result} \end{bmatrix} = \begin{bmatrix} 1 - 2q_1^2 - 2q_2^2 & 2q_1q_0 - 2q_2q_3 & 2q_1q_3 + 2q_2q_0 \\ 2q_2q_3 + 2q_0q_1 & 1 - 2q_2^2 - 2q_0^2 & 2q_2q_1 - 2q_0q_3 \\ 2q_0q_2 - 2q_1q_3 & 2q_0q_3 + 2q_1q_2 & 1 - 2q_0^2 - 2q_1^2 \end{bmatrix} * \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} \quad (4.2)$$

Si se observa la matriz de rotación resultante y se compara con la la anteriormente descrita en el desarrollo matemático, se observa que la única diferencia reside en la diagonal. Sabiendo que:

$$q_0^2 + q_1^2 + q_2^2 + q_3^2 = 1 \quad (4.3)$$

Si se igualan los términos de la diagonal de ambas matrices:

$q_0^2 + q_1^2 - q_2^2 - q_3^2 = 1 - 2q_1^2 - 2q_2^2 \rightarrow q_0^2 + q_1^2 + q_2^2 + q_3^2 = 1$

Y así se cumple la norma para todos los elementos de la diagonal.

Paso 4: se calcula la nueva posición y la velocidad para la próxima iteración. En este punto, para calcular la velocidad se elimina la componente de la aceleración en el eje Y.

$PosActual = PosAnterior + DifHoraria * VelObservada$

```
currentPosition+=currentLinearVelocity*Scalar(0.001);
currentLinearVelocity+=(gAccel-globalAccelAverage)*Scalar(0.001)
    ;
```

A medida que la diferencia de tiempo se reduce a cera, la fórmula proporciona la distancia exacta, todo esto asumiendo que la velocidad sea 100 %. Pero como la diferencia de tiempo no es cero (0.001seg) y los datos de la velocidad son imperfectas, se causa un *error de deriva*.

Paso 5: ya se ha resuelto el cálculo de la nueva posición. A continuación, se va a describir la obtención del cuaternión orientación el cual nos va a servir para gestionar la orientación del usuario. El cuaternión orientación se ha obtenido en el **Paso 2**, a partir de la velocidad angular obtenida con los datos del giroscopio. Para corregir la deriva lineal (no cuadrática) de la

orientación se van a utilizar el resto de sensores de la IMU, acelerómetro y magnetómetro. Las lecturas del giroscopio se van a corregir empujando la estimación hacia el marco de referencia fijo establecido por la dirección de la gravedad (arriba) y con la ayuda del norte magnético medida con el magnetómetro. De esta forma se van a obtener los resultados de la orientación del usuario sin ningún tipo de deriva.

```
  gMag.orthogonalize(gAccel); //Modifica gMag
  Rotation globalFrame=Rotation::fromBaseVectors(gMag,gAccel);

  globalFrame.doInvert();
  Vector globalRotation=globalFrame.getScaledAxis();
  currentOrientation.leftMultiply(Rotation::rotateScaledAxis(
    globalRotation*driftCorrectionWeight));
}
currentOrientation.renormalize();
```

4.3. Ruido de la sensórica

Por otra parte encontramos el ruido de la sensórica. Este se trata de unos de los quebraderos de cabeza más importantes dentro del proyecto, si no el mayor. La propia sensórica introduce un ruido al sistema aleatorio que hace que las medidas resulten erróneas.

La IMU utilizada en el proyecto se trata de un dispositivo de bajo coste. Este hecho supone un problema en si mismo. Por ejemplo, una simple vibración en la superficie donde se encuentra el dispositivo implica una aceleración distinta de cero. Debido a este factor, los valores obtenidos presentan valores distintos de cero cuando el cuerpo no esta experimentando ningún tipo de aceleración. Resulta que se obtiene un resultado erróneo el cual es necesario filtrar de algún modo. De esta forma los resultados obtenidos serán unicamente los producidos por el movimiento del usuario.

4.3.1. Primera aproximación, media

La primera aproximación realizada para intentar eliminar estas medidas espúreas fue la media de los valores. La frecuencia con la cual el dispositivo toma los datos de la sensórica, es decir, la frecuencia de muestreo, es muy alta, concretamente de 1000Hz. Esto significa que tenemos un gran número de datos. Para que los resultados sean más estables, una posible solución se trata de trabajar con la media de dichas muestras. En este caso, cada diez datos de entrada se obtenía un dato de salida con el cual se trabajará para el cálculo posterior de la posición.

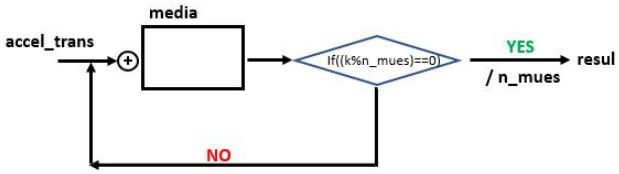

Figura 4.2: Algoritmo 1: Esquema de eliminación de ruido, primera aproximación.

Este método resulta un tanto tosco puesto que, en primer lugar, los cambios en la aceleración de entrada tardan diez veces más en verse reflejados en la salida del sistema. Es decir, introducimos un retardo en el sistema extra. Por otra parte, existe por llamarlo de alguna manera, una perdida de información. Puesto que estas diez muestras se comprimen en una sola. Por todas estas razones, resulta una primera aproximación bastante poco efectiva.

4.3.2. Segunda aproximación, almacenamiento buffer

La segunda aproximación y la definitiva, resuelve estos contratiempos. Esta, se puede decir que esta minimamente basada en la anterior pero consta de ciertas mejoras. El hecho de que se realice la media entre las muestras sigue siendo una característica de este método pero con algún cambio de por medio.

La frecuencia de muestreo en este caso no se ve mermada, es decir, el mismo número de datos con los cuales se cuenta a la entrada, se obtienen a la salida. Para ello se cuenta con un buffer de tamaño de *n_mues*. Este buffer almacena el valor de la última aceleración registrada, así como las anteriores *n_mues - 1*.

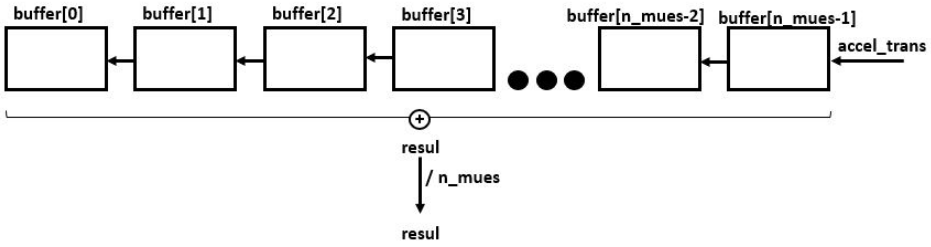

Figura 4.3: Algoritmo 2: Esquema del almacenamiento en el buffer de los datos en el sistema.

De esta forma, en el momento que una aceleración nueva se vea registrada en el sistema se calcula la media de la misma junto con las anteriores. Esto implica que los dos problemas principales del método anterior quedan solventados. El primero de ellos, la frecuencia con la cual se toman los datos es exactamente igual a la que existe a la salida del sistema. Por otra parte, no existe esa cierta perdida de información. Esto se debe a que el 100 % del valor aparece reflejado a la salida del sistema. En función del tamaño del buffer el valor de entrada aparece en un porcentaje en cada salida y en tantas salidas como *n_mues*. Por ejemplo, en el caso de nuestro algoritmo se implemento un buffer con *n_mues = 10*; por lo que, la muestra aparecerá en la salida con una representación de un *10 %* en cuanto se interne en el sistema hasta que alcance la última posición del *buffer* y que por tanto complete su *100 %* de representación.

Por otro lado, se tomaran ciertas asunciones, para conseguir eliminar el ruido residual que aparece en el resultado tras haber sido tratados dichos valores. Este ruido puede ser resultado de muchas variantes, pero se puede considerar que se trata de un resultado de la baja calidad de la sensórica. Estas asunciones siempre se han tomado teniendo en cuenta la aplicación para la cual está pensado el sistema. Por ello, dependiendo de la aplicación para la cual se destinen estos datos dicho error será o no más significativo. En el caso que nos ocupa, el cálculo de la posición, dicho ruido es muy significativo. Esto se debe a que la posición se obtiene a partir de una doble integral respecto al tiempo. Esto es lo que refleja el código anteriormente descrito en el **Paso 4**.

Sabiendo que la velocidad tal y como se muestra a continuación:

$$v(t) = v_0 + \int_0^t a \cdot dt \tag{4.4}$$

Se obtiene la posición a partir del resultado de la velocidad:

$$x(t) = x_0 + \int_0^t v \cdot dt \tag{4.5}$$

A partir de esto se alcanza la siguiente conclusión, la aceleración se trata de una variable instantánea, es decir, cada vez que el dispositivo toma una nueva muestra el valor de la aceleración instantánea toma este nuevo valor. Por otro lada, tanto la velocidad como la posición son variables acumulativas. Esto implica que cualquier error de la aceleración se incrementa a ritmos agigantados con el tiempo. Por ello, hay que intentar que los valores de partida sean lo más certeros posibles e ir auto-corrigiendo estos resultados con el otro sistema de posicionamiento para que dicha acumulación de error sea insignificante.

Como se va a analizar el resultado unicamente de este sistema de posicionamiento se va a intentar tomar unos supuestos para evitar que dicho error afecte a la salida. A continuación se van a relatar dichas asunciones.

Cuando el dispositivo esta totalmente inmóvil aparece una aceleración en el mismo. Este valor se trata de ruido implícito de la sensórica. La primera asunción que se va a tomar se trata de marcar un límite mínimo de aceleración, es decir, por debajo de este se considerará que la aceleración es nula. Se considera que cuando una persona se mueve es altamente improbable que en esos movimientos alcance una aceleración tan menor al límite marcado. Debido a este hecho, se van a tomar esas aceleraciones como ruido, por lo que serán cero. Para llevar esto a cabo se marcó un límite de $0{,}05 m/s^2$. De esta forma, cuando las gafas estén inmóviles la aceleración resultante será cero, y tanto la velocidad como la posición no varían.

La segunda de las asunciones, se trata de un límite máximo de aceleración. Una persona no puede moverse con aceleraciones ínfimas, pero tampoco con grandes dentro de esta aplicación. En este caso se marcó un límite de $100 m/s^2$.

La tercera de ellas también está relacionada con la aceleración. Tras haber aplicado todos estos límites, se considera que es altamente improbable que una persona se mueva solamente a lo largo de un eje sin experimentar ningún tipo de aceleración en cualquier otro. Por lo que, se va a considerar que si dos de las tres componentes de la aceleración tienen valor cero, no existe aceleración y por tanto todos sus valores pasarán a ser nulos.

Por otra parte, también se tomaron unas ciertas asunciones en el campo

de la velocidad. Puesto que los resultados se van a observar en el campo de la posición y los errores en la velocidad también implican errores en la posición. En este caso, se tomó un nivel mínimo de velocidad razonable $0{,}0005m/s = 0{,}05cm/s$, en el caso que la velocidad se encontrase por debajo de este nivel se considera nula.

Por último, la quinta asunción también tiene que ver con la velocidad. Teóricamente al finalizar un movimiento, es decir, al volver el objeto a la situación de reposo, la velocidad también debería volverse nula. Pero en el caso que nos atañe, por ciertas pérdidas de aceleraciones esto no sucede. Por ello, se va a considerar que en el momento que se obtenga tres veces consecutivas unas aceleraciones en todas sus direcciones nulas, la velocidad también pasará a ser cero. De esta forma la posición será fija.

4.4. Resultados

En este apartado se van a mostrar los resultados obtenidos tanto de la aceleración y la posición como de los cuaterniones orientación. Además, se va a realizar un análisis de estos y explicar algunas conclusiones obtenidas a partir de este análisis.

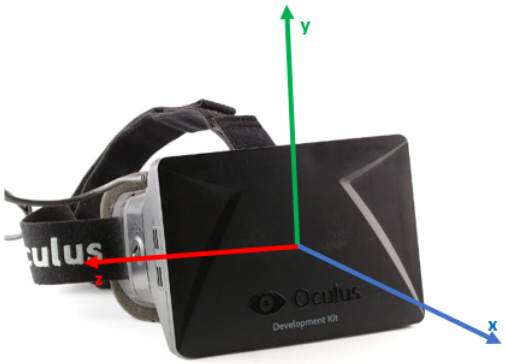

Figura 4.4: Ejes Oculus Rift DK1

Los ejes de las gafas, o lo que sería lo mismo de la IMU, ya que es está la que recopila los datos, corresponden con los de la imagen posterior. Por

lo que los resultados obtenidos en cada caso se han ido tomando siguiendo tales ejes para realizar los movimientos.

Todos estos experimentos se han realizado a partir de los datos tomados por la IMU. A la hora de imprimir los resultados, para poder ver cual era la influencia del movimiento respecto de la posición inicial, resultaba más sencillos imprimir valores tanto de aceleración como de posición con una velocidad menor que la de muestreo, ya que esta era muy alta, $1000Hz$.

En primer lugar, se va a realizar una descripción de los resultados de la gestión de la posición del usuario. Cabe destacar que la realización de experimentos repetitivos resulta una tarea muy complicada. El resultado que se puede obtener en cada experimento puede variar, ya que no solo depende del desplazamiento, sino también depende del tiempo en el que comience el movimiento y del movimiento en sí. Esto se debe a que el error de partida aumenta con el tiempo y que siempre no se va a mover el dispositivo con a misma aceleración. Por lo que se van a analizar los resultados desde un punto de vista orientativo.

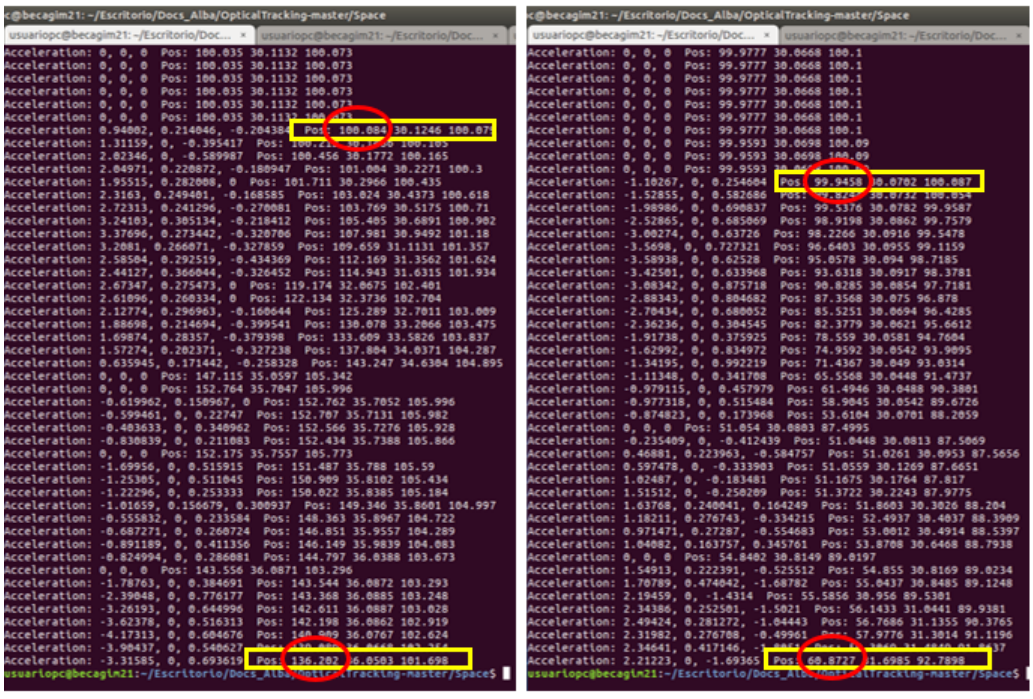

Figura 4.5: Resultados desplazamiento eje X

En la figura 4.5 se describe un movimiento realizado en el eje X. En todos tomamos como punto de partida el punto (100, 30, 100). En este

caso que nos atañe, la imagen de la izquierda representa un movimiento positivo en el eje de las x. El resultado muestra que se ha realizado un movimiento de $136{,}202 - 100{,}084 = 36{,}118cm$. Por el contrario, la imagen situada a la derecha, representa un movimiento negativo en este eje de $99{,}9458 - 60{,}8727 = 39{,}0731m$.

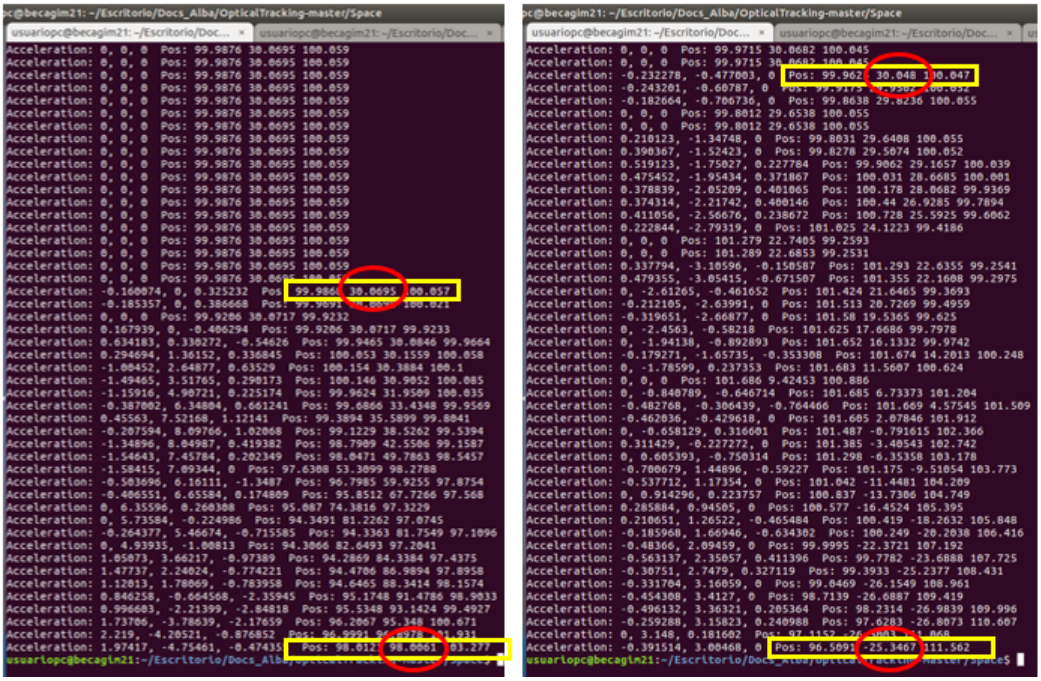

Figura 4.6: Resultados desplazamiento eje Y

Por otra parte, la figura 4.6 describe un movimiento realizado en este caso sobre el eje Y. La imagen de la izquierda representa un movimiento positivo en el eje de las Y de $98{,}0061 - 30 = 68cm$, y la imagen situada a la derecha, representa un movimiento negativo sobre dicho eje de $30 - (-25{,}3467) = 55{,}3467cm$.

Por otra parte, la figura 4.7 describe un movimiento realizado en el eje Z. La imagen de la izquierda representa un movimiento positivo en el eje de las Z de $123{,}234 - 100{,}091 = 23{,}143cm$. La imagen situada a la derecha, representa un movimiento negativo en este eje de $100{,}098 - 72{,}5404 = 27{,}5576cm$.

Todos los supuestos de los que se habló anteriormente tienen consecuencias que caben destacar. Lo importante es encontrar un equilibrio a la hora de fusionar ambos sistemas de posicionamiento. Es decir, en este caso las asunciones tomadas pueden considerarse algo restrictivas ya que para evitar

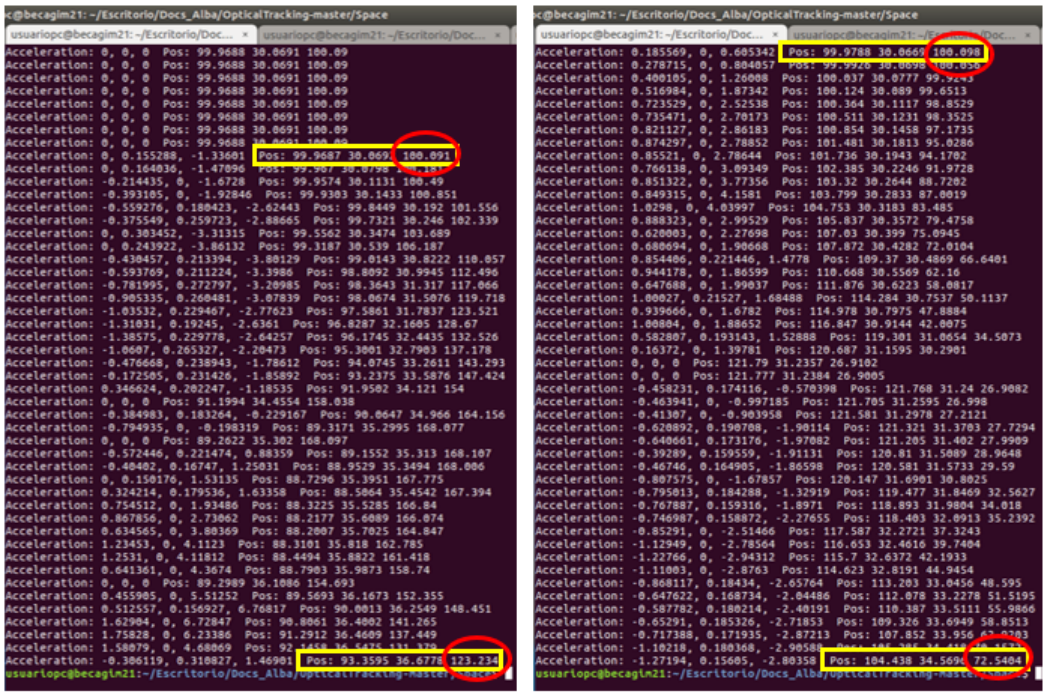

Figura 4.7: Resultados desplazamiento eje Z

un error alto durante un periodo de tiempo relativamente largo, se eliminan ciertas componentes tanto de la aceleración como de la velocidad. Dichos recortes están pensados para evitar el ruido de nuestro sistema, pero en algunas ocasiones también puede recortar datos válidos resultado del movimiento. Por ello se habla de equilibrio, al tratarse de un sistema de posicionamiento dual, posición obtenida a partir de la IMU y posición obtenida a partir de la detección de marcadores luminoso, la posición resultante se obtendrá como combinación de ambas o que un sistema corrija al otro. En el caso de que dichas correcciones se produzcan cada poco tiempo, los límites marcados dejarán de ser tan restrictivos y la posible información perdida desaparecerá. De esta forma, el sistema será mucho más robusto y estable.

En el caso que nos atañe, los movimientos con aceleraciones muy tenues son considerados ruido. Por ello, el sistema tiene mejores resultados con movimientos con mayor aceleración.

Por otra parte, hay otro aspecto que debe de ser analizado. Cuando se inicial el movimiento del cuerpo experimenta una aceleración y al frenar experimenta de nuevo otra aceleración con la misma magnitud pero con signo contrario. Por ello, se concluye que la suma de todas las aceleraciones desde

que se comienza el movimiento hasta que el dispositivo está inmóvil debe valer 0.

$$\sum a_t = 0 \tag{4.6}$$

Durante el tiempo que el cuerpo esté en movimiento la velocidad tiene valor no nulo. En teoría al finalizar el movimiento la velocidad debería pasar a tener un valor nulo de nuevo. El hecho de que exista ruido en el sistema implica que la suma de aceleraciones no es exactamente cero, esto produce que al final el movimiento la velocidad tampoco sea nula por lo que existe un error. Este hecho se soluciona tal y como se ha relatado con anterioridad.

Dejando de lado todo esto, tal y como se ve las aceleraciones en los tres ejes dan como resultados valores de posiciones adecuados y que corresponden tanto con la dirección como con el sentido del movimiento. Por lo que a la hora de ir comparando con el otro sistema de posicionamiento se trata de un sistema fiable.

Para mejorar los resultados obtenidos de la fusión sensórica se intentó aplicar un Filtro Kalman. El filtro de Kalman es un algoritmo de fusión sensórica que posibilita la estimación del estado no medible de un sistema dinámico a partir de mediciones ruidosas. Este filtro requiere importantes supuestos de modelado y ajuste para alcanzar sus beneficios teóricos. Se trata de un estimador óptimo para sistemas lineales con mediciones lineales y ruido gaussiano, pero su rendimiento fuera de ese rango no está garantizado. Además, se trata de un método apropiado para sistemas con una tasa de muestreo más baja, pues nuestro sistema tiene una tasa de muestreo muy alta, 1000Hz; y con un alto grado de predictibilidad debido a un modelo de movimiento más fuerte. Los resultados tras intentar su implementación no fueron positivos, por lo que se decidió eliminar y seguir trabajando con el algoritmo de fusión ya existente. Esto puede ser porque los datos de la sensórica puede que no sean directamente los obtenidos por los sensores o puede ser por las razones anteriormente relatadas.

A continuación, se va a hacer un pequeño análisis de los resultados de la gestión de la orientación a partir de los cuaterniones orientación. No se puede hacer un análisis más profundo ya que los cuaterniones se tratan de números de tres dimensiones muy difíciles de interpretar directamente. Por lo que el impacto de estos resultados se podrá observar con mucha más claridad en el próximo capitulo representando la orientación en un entorno gráfico.

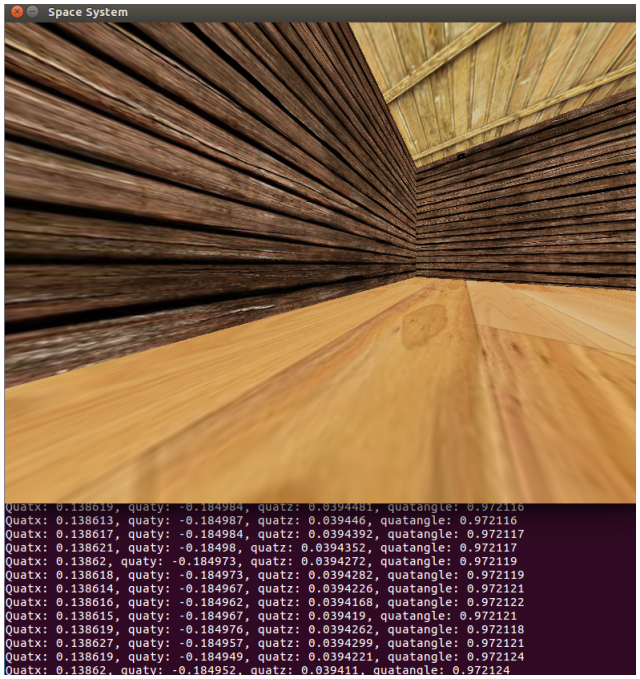

Figura 4.8: Ejemplo 1 cuaternión orientación

La figura 4.8 representa el cuaternión orientación, se puede analizar matemáticamente tanto el ángulo de giro como los ejes de giro. El resultado de ambos cálculos se muestran a continuación, estos se basan en la teoría de los cuaterniones del capítulo anterior 3.

$quatangle = cos(\theta/2) = 0{,}972124 \rightarrow \theta/2 = arccos(0{,}972124) = 0{,}23667 \rightarrow \theta = 0{,}47334 \simeq \pi/6 = 30^{o}$

$ejex = n1 * sin(\theta/2) = 0{,}13862 \rightarrow n1 = 0{,}13862/sin(\pi/12) = 0{,}535586$

$ejey = n2 * sin(\theta/2) = -0{,}184952 \rightarrow n2 = -0{,}184952/sin(\pi/12) = -0{,}714599$

$ejez = n3 * sin(\theta/2) = 0{,}039411 \rightarrow n3 = 0{,}039411/sin(\pi/12) = 0{,}152274$

Se realiza un análisis de otro caso, es decir se colocan las gafas en otra posición tal y como se observa en la figura 4.9. Los resultados tanto del ángulo de giro como de los ejes de giro se muestran a continuación.

$quatangle = cos(\theta/2) = 0{,}705609 \rightarrow \theta/2 = arccos(0{,}705609) = 0{,}787514 \rightarrow \theta = 1{,}575028 \simeq \pi/2 = 90^{o}$

Figura 4.9: Ejemplo 2 cuaternión orientación

$$ejex = n1 * sin(\theta/2) = 0{,}11583 \rightarrow n1 = 0{,}11583/sin(\pi/4) = 0{,}163808$$

$$ejey = n2 * sin(\theta/2) = -0{,}161758 \rightarrow n2 = -0{,}161758/sin(\pi/4) = -0{,}228760$$

$$ejez = n3 * sin(\theta/2) = -0{,}680098 \rightarrow n3 = -0{,}680098/sin(\pi/4) = -0{,}961803$$

A la hora de analizar los valores obtenidos en cada caso, tal y como se dijo anteriormente, resulta muy complicado debido a la difícil interpretación de los cuaterniones. Se han mostrado algunos ejemplos y se va a ver realmente su utilidad y funcionamiento en el próximo capítulo 5.

Capítulo 5

Gestión de los resultados en entorno gráfico

A la hora de visualizar los resultados obtenidos a través de la sensórica de las Oculus, se decidió utilizar un bloque ya implementado dentro del proyecto denominado *Space*. Dicho proyecto C++ se trata de una simulación digital del entorno en el cual se pretende desarrollar el proyecto. El entorno se trata de una habitación que cuenta con marcadores luminosos. Este tenía dos funcionalidades marcadas, en primer lugar, se encargaba de identificar la posición de los leds en la habitación y por lo tanto devolvía las coordenadas de su posición. En segundo lugar, existía la posibilidad de desplazarse dentro de la habitación con la ayuda de los periféricos del ordenador. En este caso, los periféricos utilizados se trataban el teclado y el mouse. Con la ayuda del teclado, era posible desplazarse dentro de la habitación. Por otro lado, con ayuda del mouse se gestiona la orientación, es decir, se marcaba la dirección hacía donde se miraba. Para poder gestionar los resultados obtenidos con ayuda de los drivers de las Oculus Rift hay que realizar varias modificaciones tanto en el código de los drivers como en al código del propio proyecto *Space*.

5.1. Adaptación de los drivers de las Oculus Rift

Se va trabajar con un entorno gráfico diferente. Por ello se eliminará el propio de los drivers y con el cual se han ido observando los primeros

resultados. Este tenía dibujado los ejes de la tierra y los ejes de las gafas. A partir de ellos se podía observar cual era la posición de las gafas en cada instante de tiempo, cuando se experimentaba un movimiento los ejes de las gafas se desplazaban de los ejes de la tierra. Por otra parte también se podía observar si se había experimentado algún giro, es decir, algún cambio en la orientación de las gafas. Cuando se había producido este hecho, los ejes se rotaban y la posición de los mismos variaba al igual que los de las gafas.

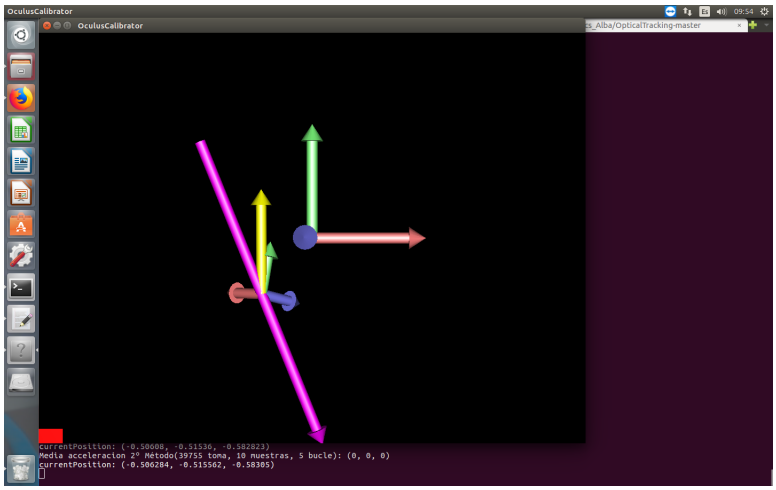

Figura 5.1: Resultados gráficos tras haber experimentado una movimiento en el entorno gráfico de las Oculus Rift

Por otra parte, a la hora de extraer los resultados obtenidos se van a implementar cuatro funciones las cuales serán declaradas en una nueva librería *OculusCalibrator.h*. Dichas funciones extraen los datos tanto la aceleración como la velocidad, posición y matriz de rotación.

Posición:

```
void getOculusPos(float *posx, float *posy, float *posz){
  Tracker::Point pnt = my_OculusCalibrator.getPos();
  *posx = pnt[0];
  *posy = pnt[1];
  *posz = pnt[2];
}
```

Velocidad:

```
void getOculusSpeed(float *speedx, float *speedy, float *speedz)
    {
```

```
  Tracker::Vector spe = my_OculusCalibrator.getSpeed();
  *speedx = spe[0];
  *speedy = spe[1];
  *speedz = spe[2];
}
```

Acceleración:

```
void getOculusAccel(float *accx, float *accy, float *accz){
  Tracker::Vector vtc = my_OculusCalibrator.getAccel();
  *accx = vtc[0];
  *accy = vtc[1];
  *accz = vtc[2];
}
```

La última de las funciones da como resultado la matriz de rotación. A continuación se va a mostrar partes del código significativas para la obtención de la matriz de rotación.

```
// Quaternions values
Tracker::Rotation rot = my_OculusCalibrator.getOrient();
const Tracker::Scalar *quat_const = rot.getQuaternion();

// n and theta calculation
x = quat[0]; y = quat[1]; z = quat[2]; qangle = quat[3];

angle_rad = 2*(acos(qangle)); //Rotation angle (radians)
angle = angle_rad*180.0/M_PI; //Rotation angle (degrees)

// Rotation axis calculation
x = x/sinf(angle_rad/2);
y = y/sinf(angle_rad/2);
z = z/sinf(angle_rad/2);

// Calculate rotation matrix
m[0] = (one_minus_c * xx) + c;
m[1] = (one_minus_c * xy) + zs;
m[2] = (one_minus_c * zx) - ys;
m[3] = 0.0;

m[4] = (one_minus_c * xy) - zs;
m[5] = (one_minus_c * yy) + c;
m[6] = (one_minus_c * yz) + xs;
m[7] = 0.0;
```

```
m[8] = (one_minus_c * zx) + ys;
m[9] = (one_minus_c * yz) - xs;
m[10] = (one_minus_c * zz) + c;
m[11] = 0.0;

m[12] = 0.0;
m[13] = 0.0;
m[14] = 0.0;
m[15] = 1.0;
```

A partir de esta matriz de rotación extraídas de los cuaterniones es posible representar la orientación de un objeto en tres dimensiones. Estas matrices se tratan de matrices absolutas, es decir, al vector de partida se le aplican dichas rotaciones y se obtiene el valor de la orientación del objeto en cada instante de tiempo.

$$\begin{bmatrix} Vec\_rot[0] \\ Vec\_rot[1] \\ Vec\_rot[2] \\ Vec\_rot[3] \end{bmatrix} = \begin{bmatrix} m\_rot[0] & m\_rot[4] & m\_rot[8] & m\_rot[12] \\ m\_rot[1] & m\_rot[5] & m\_rot[9] & m\_rot[13] \\ m\_rot[2] & m\_rot[6] & m\_rot[10] & m\_rot[14] \\ m\_rot[3] & m\_rot[7] & m\_rot[11] & m\_rot[15] \end{bmatrix} * \begin{bmatrix} Vec\_orig[0] \\ Vec\_orig[1] \\ Vec\_orig[2] \\ Vec\_orig[3] \end{bmatrix} \tag{5.1}$$

Esto es importante puesto que al no tratarse de matrices relativas siempre son aplicadas al mismo vector y de esta forma el resultado se obtiene satisfactoriamente.

5.2. Gestión de la posición y orientación en el entorno gráfico

Al tener ya los resultados aportados por la sensórica de las gafas disponibles, era posible trabajar dentro del nuevo entorno gráfico con ellos. El manejo tanto de la posición como de la orientación dentro de esta simulación de habitación ha de ser controlada por la sensórica de las Oculus Rift, la segunda funcionalidad del proyecto, *Space*. Tal y como se ha comentado con

anterioridad, el posicionamiento absoluto con la IMU es imposible, pero de momento se utilizarán estos resultados de manera intuitiba.

Para realizar esta tarea se utilizan estas funciones $camera.deplacerOculus()$; y $camera.lookAt(view)$;. La primera de ellas, $camera.deplacerOculus()$; es la encargada de realizar los cálculos de las variables necesarios para que la segunda de las funciones utilice. Esta se trata de la modificación principal, puesto que es la encargada de calcular las posiciones y orientación. Las variables de las que se hablaba, las cuales son necesarias para representar tanto la orientación como la posición son las siguientes: $m\_position$, $m\_orientation$, $m\_pointCible$ y $m\_axeVertical$.

En cuanto a la primera variable, $m\_position$, se trata directamente de la posición del sistema. Por lo cual, se pasará a esta variable los datos recabados y procesados por las Oculus Rift.

Las tres variables restantes son necesarias para determinar la orientación del sistema.

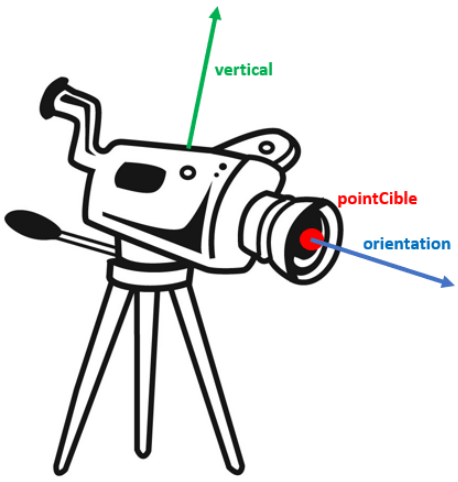

Figura 5.2: Vectores camara.

Tal y como se muestra en la ilustración, dos de las variables se tratan de vectores para determinar la 'posición' de la cámara. Y el restante se trata del punto en el cual se encuentra el objetivo. En cuanto a los vectores, el primero de ellos determina la dirección hacia la cual mira la cámara, la orientación. Por otra parte, el otro se encarga de determinar el giro de la cámara, es decir, determina su vertical.

A la hora de calcular $m\_orientation$, se trata del vector dirección de las

Oculus Rift. Este vector es el resultado del vector original que determina la dirección, multiplicado por la matriz de rotación anteriormente descrita. De esta forma, se obtiene la dirección actual de la cámara tras haber experimentado cualquier tipo de movimiento rotatorio.

Por otra parte, se encuentra $m\_axeVertical$, el procedimiento es similar al anterior. Se multiplica al vector que representa la vertical original por la matriz de orientación. Así obtenemos el vector vertical resultante.

Por último, $m\_pointCible$. Esta variable se trata del punto donde se encuentra el objetivo de la cámara en la habitación. Es el resultado de la suma de la posición y la orientación.

La segunda de las funciones que se utilizan es $camera.lookAt(view)$;. Esta función toma los parámetros anteriormente calculados para determinar un $'modelview'$ o $'modelodevista'$. Para ello, utiliza la función $lookAt$ de la librería $OpenGL$.

Si se define un espacio de coordenadas usando 3 ejes perpendiculares (o no lineales) puedes crear una matriz con esos 3 ejes más un vector de traducción y se puede transformar cualquier vector en ese espacio de coordenadas multiplicándolo por esta matriz. Esto es exactamente lo que hace la matriz LookAt. Teniendo 3 ejes perpendiculares y un vector de posición se puede definir el espacio de la cámara, por ello, se puede crear nuestra propia matriz $LookAt$:

$$LookAt = \begin{bmatrix} Rx & Ry & Rz & 0 \\ Ux & Uy & Uz & 0 \\ Dx & Dy & Dz & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 0 & 0 & -Px \\ 0 & 1 & 0 & -Py \\ 0 & 0 & 1 & -Pz \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.2}$$

Dónde R es el vector derecha, U es el vector ascendente, D es el vector de dirección y P es el vector de posición de la cámara. Hay que tener en cuenta que el vector de posición está invertido ya que finalmente se quiere traducir el mundo en la dirección opuesta a la que nos movemos. Usando esta matriz $LookAt$ como la matriz de vista transforma, tal y como se esperaba, las coordenadas del mundo al espacio de vista que acabamos de definir.

Afortunadamente, GLM hace todo este trabajo. Solo hay que especificar la posición de cámara, la posición del objetivo y un vector que representa el vector ascendente en el espacio mundial (el vector ascendente que usamos para calcular el vector derecho). GLM luego crea la matriz $LookAt$ que

podemos usar como matriz de vista.

Por otra parte, encontramos la función $glm :: perspective$. A la hora de realizar una proyección el usuario debe especificar la ventana 3D de visión. Para ello se emplea esta función, la cual cuenta con los siguientes parámetros:

$$glm :: perspective(floatfovy, floataspect, floatzNear, floatzFar);$$

fovy: FOV (field of view). Se trata del campo de visión vertical, la cantidad de "zoom". Piensa en el lente de la cámara. Usualmente está entre 90° (extra ancho) y 30° (zoom aumentado)

aspect: Proporción. Depende del tamaño de tu ventana $4/3 = 800/600 = 1280/960$

zNear: Plano de corte cercano. Tan grande como sea posible o tendrás problemas de precisión.

zFar: Plano de corte lejano. Tan pequeño como se pueda.

Por último, con ayuda de la matriz obtenida con la ayuda de $glm :: LookAt$ y la perspectiva anteriormente calculada ya es posible construir la 'habitación', mediante la línea de código $room.printcube(projection, view)$;.

5.3. Resultados

A continuación, se van a analizar los resultados alcanzados tras haber integrado los valores obtenidos a partir de la sensórica de las Oculus Rift, con el entorno gráfico generado que simula el espacio en el cual se va a implantar el sistema. Se van a mostrar los resultados que se obtienen en el entorno gráfico tras haber realizado diferentes desplazamientos o giros.

El entorno gráfico en el cual se va a trabajar, tal y como se ha comentado con anterioridad, se trata de un proyecto ya implementado denominado *Space*. Este proyecto representa una habitación generada digitalmente. Esta se trata de una simulación del caso práctico en el cual se pretende establecer el proyecto. Además, dentro de la habitación se pueden encontrar leds simulando el caso practico real.

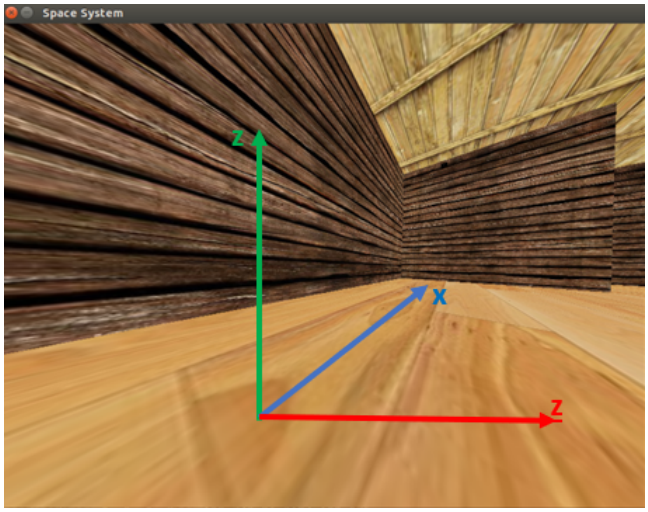

Figura 5.3: Captura entorno gráfico en el cual se va a hacer el análisis de los resultados obtenidos

En primer lugar, se van a analizar los resultados obtenidos tras realizar desplazamientos dentro de la habitación. Los movimientos en el entorno gráfico corresponden con los resultados obtenidos en el capítulo anterior. Se van a analizar los resultados de los movimientos en los tres ejes, tanto en sentido positivo como en sentido negativo. En cuanto a los movimientos realizados en el eje X los resultados son los que se muestran a continuación.

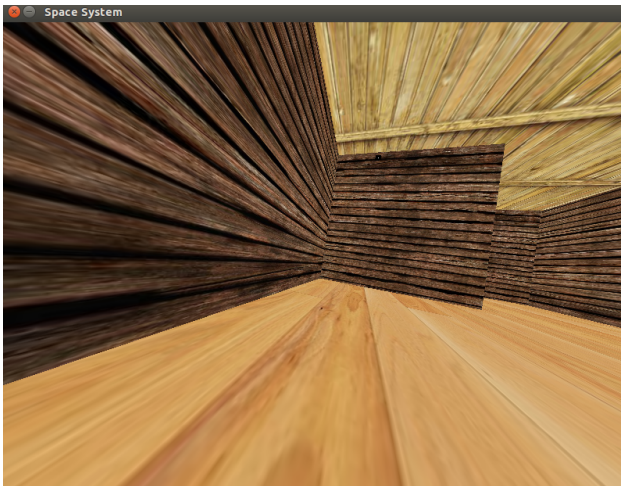

Figura 5.4: Representación movimiento en el sentido positivo del eje X

En la imagen 5.4 se representa un movimiento realizado en el sentido

positivo del eje X. Tal y como era de esperar se obtiene un desplazamiento en dicha dirección dentro de la habitación.

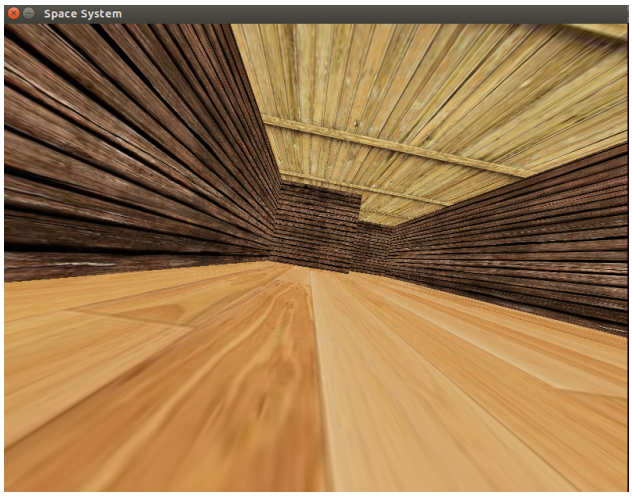

Figura 5.5: Representación movimiento en el sentido negativo del eje X

En cuanto a la imagen 5.5, al igual que en el caso anterior, se representa un desplazamiento en la dirección del eje X pero en este caso en el sentido negativo. Se observa un retroceso en dicho eje.

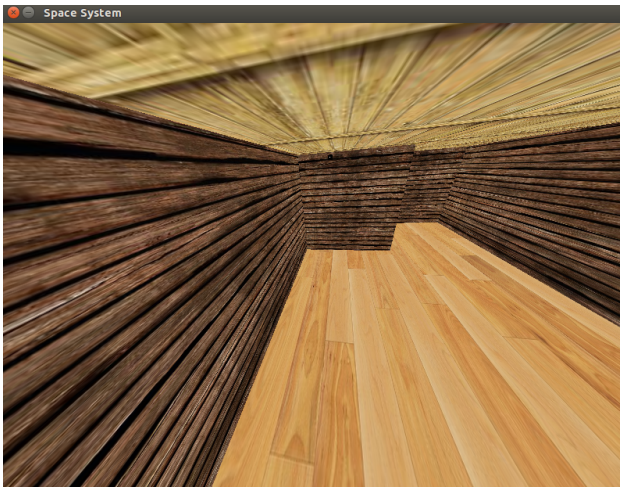

Figura 5.6: Representación movimiento en el sentido positivo del eje Y

En la imagen 5.6 en cambio, se analiza el movimiento en el eje Y en el sentido positivo. En este caso, representa que el usuario sube el eje Y, por lo que se obtiene el resultado observado en la imagen.

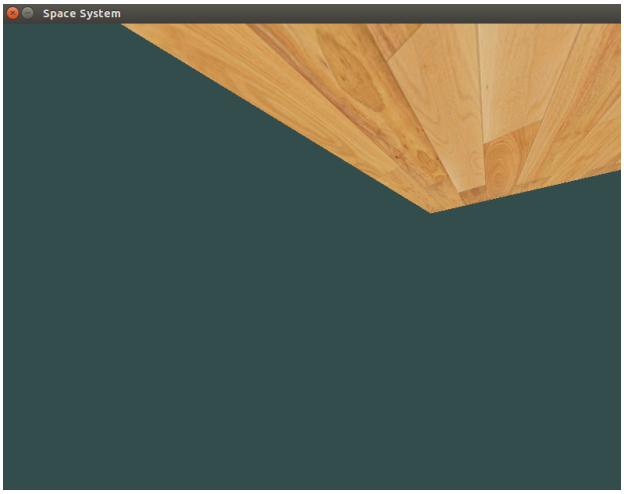

Figura 5.7: Representación movimiento en el sentido negativo del eje Y

La figura 5.7 representa exactamente el movimiento en la misma dirección pero con un sentido contrario. Ya que el punto de partida se encuentra cerca del suelo, al realizar este movimiento, el "usuario" se sale de la habitación por el suelo.

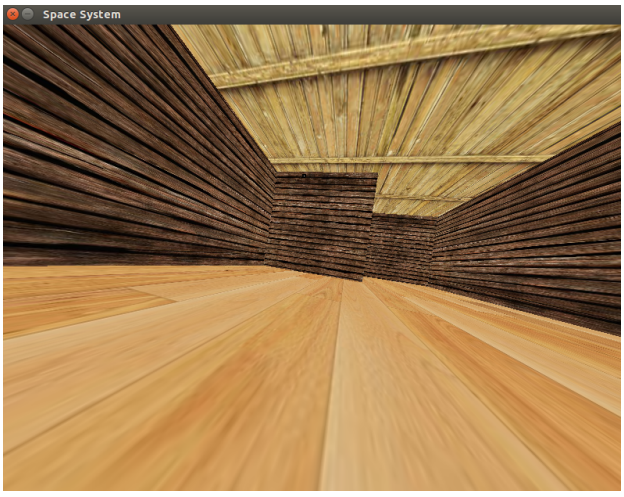

Figura 5.8: Representación movimiento en el sentido positivo del eje Z

Se analizan los resultados del último eje, el eje Z. La imagen 5.8 representa un movimiento en el sentido positivo del eje Z, tal y como se ve en la imagen nos desplazamos hacia la derecha en el espacio.

Por último, se analiza el movimiento en el sentido negativo del eje Z.

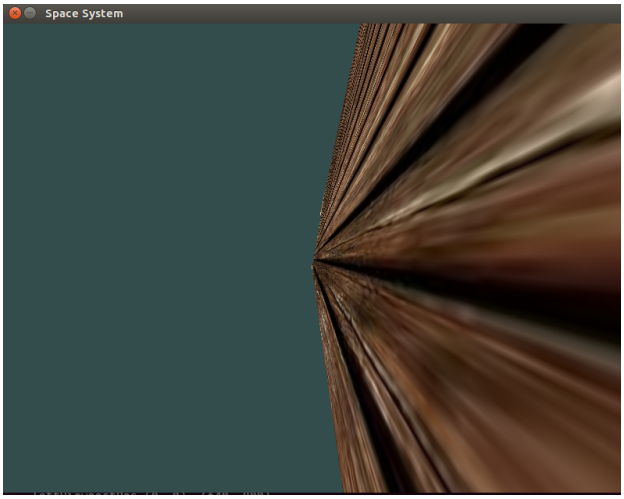

Figura 5.9: Representación movimiento en el sentido negativo del eje Z

Al igual que en el caso del movimiento en el sentido negativo del eje Y, al estar próximo al límite de la habitación y haber realizado un desplazamiento grande, el "usuario" se sale de la habitación por la pared. El movimiento se realiza a la izquierda de la habitación y se observan los resultados en la figura 5.9.

Además, se van a representar los resultados tras realizar algún cambio en la orientación del sistema. Al representar la orientación gráficamente resulta más sencillo observar si funciona o no correctamente, ya que el análisis matemático de los cuaterniones se trata de un hecho muy complejo.

A la hora de representar los resultados obtenidos a partir de los cuaterniones orientación vamos a contar con la nomenclatura utilizada en la regla de la mano derecha o del sacacorchos.

La regla de la mano derecha o del sacacorchos es un método para determinar sentidos vectoriales, y tiene como base los planos cartesianos. Se emplea prácticamente de dos maneras: para sentidos y movimientos vectoriales lineales, y para movimientos y direcciones rotacionales. Cuando se hace girar un sacacorchos o un tornillo "hacia la derecha" (en el sentido de la agujas de un reloj) el sacacorchos o el tornillo "avanza", y viceversa, cuando se hace girar un sacacorchos o un tornillo "hacia la izquierda" (contrario a las agujas del reloj), el

sacacorchos o el tornillo "retroceden".

En primer lugar, se van a representar los giros en el eje X. El eje X de las gafas se trata del que está representado en la *Figura: 4.4*. Para poder observar la correcta funcionalidad se van a realizar dos 'experimentos', es decir, se va a girar el dispositivo en ambos sentidos de la dirección del eje de rotación X. En la imagen que se muestra a continuación, 5.10, se realiza un giro del dispositivo en sentido contrario a las aguas del reloj y el resultado coincide con lo esperado.

Figura 5.10: Representación giro de retroceso sobre el eje X

En la imagen 5.11 por el contrario, se realiza un giro del dispositivo en sentido de las aguas del reloj y al igual que anteriormente, el resultado coincide con lo esperado.

Figura 5.11: Representación giro de avance sobre el eje X

A continuación, se va a realizar el estudio sobre el eje de rotación Y. Por

ello, se va a girar el dispositivo en ambos sentidos de la dirección del eje de rotación Y.

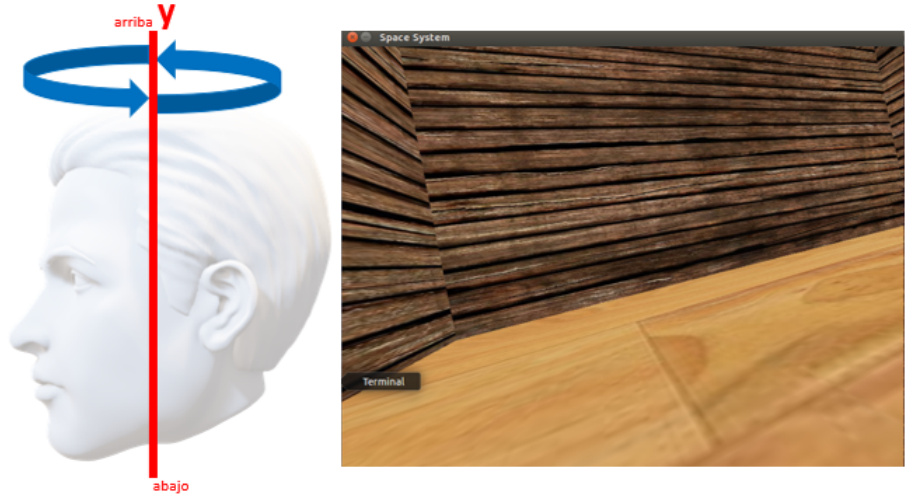

Figura 5.12: Representación giro de retroceso sobre el eje Y

En este caso, se va a realizar en primer lugar un giro con la dirección del eje Y con sentido contrario a las agujas del reloj. Tal y como se puede observar, 5.12, el giro de la cabeza sobre ese eje en ese sentido coincide con el resultado obtenido, es decir con la parte de la habitación que se observa.

Figura 5.13: Representación giro de avance sobre el eje Y

En la imagen que se muestra seguidamente, corresponde a un giro en la dirección del eje Y, pero en este caso con sentido de las agujas del reloj.

Los resultados concuerdan con la parte del entorno gráfico que se esperaba observar, 5.13.

Por último, se va a realizar el giro del dispositivo en ambos sentidos de la dirección del eje de rotación Z. En primer lugar, se realiza el giro en el sentido contrario de las agujas del reloj, el resultado se muestra en la *Figura 5.14*.

Figura 5.14: Representación giro de retroceso sobre el eje Y

En el caso de la *Figura 5.15*, el giro se realiza en el sentido de las agujas del reloj. Por lo tanto, se obtiene el siguiente resultado.

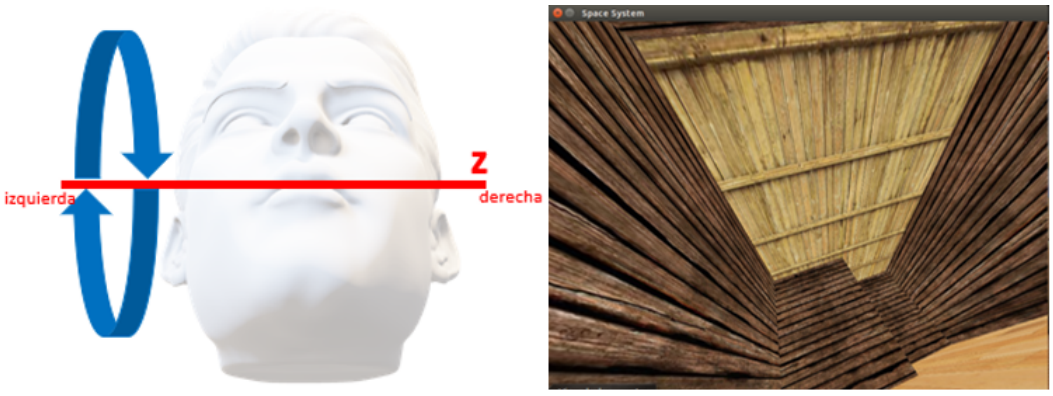

Figura 5.15: Representación giro de avance sobre el eje Y

En cuanto a los resultados obtenidos a partir del cuaternión orientación, se puede concluir que los resultados son perfectos, la orientación de las gafas concuerda con la parte de la habitación que se muestra. Por ello, tal y como

se había descrito en el capítulo anterior, las lecturas obtenidas por el giroscopio han sido perfectamente corregidas con ayuda del resto de los sensores. Teniendo esto como consecuencia la eliminación total de la deriva y por lo tanto obteniendo los resultados con una alta exactitud.

Capítulo 6

Conclusiones y líneas futuras

6.1. Conclusiones

En relación a los objetivos planteados en la introducción, puede concluirse lo siguiente:

- A pesar de que en principio la gestión de la posición de un usuario a partir de un acelerómetro parece trivial, ha supuesto un reto muy importante. Matemáticamente el hecho de conseguir la posición a partir de unas aceleraciones dadas y sabiendo los instantes de tiempo en los que se producen, se trata de un procedimiento automático. El problema surge cuando aparecen los errores de medida, o ruido. Al tratarse tanto la velocidad como la posición de dos variables acumulativas, el hecho de que aparezca ruido en cualquier punto del sistema implica un error acumulado. Este error o ruido puede surgir por diversos motivos, pero la mayoría ellos están fundados en que la sensórica con la que cuentan estos dispositivos se trata de hardware de bajo precio. Estos sensores son suficientes para gestionar la orientación, funcionalidad que las Oculus Rift incorporan en su versión comercial dentro de los videojuegos. Pero cuando se analiza si esto es suficiente para gestionar la posición surgen dudas.
- El hecho de que este sistema no sea suficiente para gestionar la posición del usuario por si solo con precisión abre una serie de posibilidades. La primera y quizás la más automática es hablar de que cuanto menor sea el error introducido por los sensóres, y en consecuencia el ruido,

el sistema pasa a ser mucho más estable. Por ello, puede ser posible gestionar la posición con exactitud. En el proyecto en el que nos encontramos, esta alternativa no es posible, por ello surge una segunda alternativa y en la cual se basa nuestro proyecto. Contar con un sistema de posicionamiento dual, es decir, que la posición se obtiene ya sea por la combinación como por la corrección de dos resultados de posición obtenidos por diferentes medios. En este caso, los sistemas que se usan en este proyecto son el desarrollado en este trabajo, posición obtenida mediante la IMU y mediante marcadores luminosos. Al haber otra fuente de datos, los resultados obtenidos por un sistema u otro se pueden ir corrigiendo y dando como resultado un sistema mucho más fiable y robusto. Tiene que existir un compromiso entre la frecuencia a la cual se van corrigiendo los datos y los límites marcados como ruido dentro del sistema de posicionamiento basado en fusión sensórica. Es decir, si los 'errores' se van corrigiendo cada poco tiempo, quizá ese límite deba de ser mucho menor o quizás inexistente ya que el error se va eliminando paulatinamente.

- En cuanto al aspecto de gestión de la orientación del usuario en el entorno, los resultados son muy favorables. A pesar de haber tenido que trabajar con cuaterniones, los cuales se tratan de números muy complejos y que resulta muy complicado analizarlos. Tal y como se había observado en los estudios realizados, la gestión de la orientación con alta fiabilidad es posible debido a que los errores no son acumulativos y se pueden ir corrigiendo con el resto de sensores que forman la IMU. Al obtener la matriz de rotación a partir de los cuaterniones fue mucho más sencillo obtener los resultados en el entorno gráfico y por lo tanto observar el correcto funcionamiento del sistema.

6.2. Líneas futuras

En cuanto a las líneas futuras que se pueden seguir para continuar con el proyecto hay varios aspectos a destacar.

En primer lugar, se debería de hacer una optimización del código. Durante todo el proyecto, el objetivo se trataba de obtener los resultados con

la mayor fiabilidad posible por lo que la parte de optimización del código siempre ha quedado en un segundo plano. Uno de los puntos importantes que suponía este método de mejora sobre los dispositivos de posicionamiento móvil era que el nivel de cómputo disminuía. Por ello, es necesario optimizar el código para que de esta forma el cómputo se reduzca y este hecho resulte más constatable.

Por una parte, una de las opciones más interesantes es realizar la fusión de ambos bloques de posicionamiento. El sistema de posicionamiento se trata de un sistema de posicionamiento dual, son dos los sistemas que trabajan simultáneamente para obtener la posición del usuario dentro de una habitación. Tal y como se ha dicho con anterioridad, el posicionamiento basado en IMU es imposible, por lo que resulta interesante realizar la unión de ambos supondrá que el error posible de un cálculo resulte compensado por los resultados del otro.

Además, podría resultar interesante el estudio de un nuevo filtro de fusión de datos. Se intentó la implementación de un filtro Kalman pero este directamente no obtenía mejores resultados. Puede que exista una posibilidad que mejore los datos obtenidos.

Por último, una posible línea futura sería una mejora del procedimiento de calibración de los sensores. Actualmente se realiza recabando una serie de datos al comienzo del estudio y se construye una matriz de calibración de 3x3 tanto para el acelerómetro como giroscopio. Puede que resulte interesante almacenar un mayor volumen de datos para que el resultado tras haber realizado la calibración del dispositivo sea más fiables.

Bibliografía

[1] Martínez Mediavilla, Patricia y Villar Bonet, Eugenio: «*Model-implementation fidelity in cyber physical system design*», Capítulo 8, Universidad de Cantabria, 2016

[2] Diferencias entre la Realidad Virtual, Realidad Aumentada y Realidad Mixta,
https://computerhoy.com/noticias/hardware/realidad-virtual-aumentada-mixta-conoces-diferencias-45378

[3] Peñaloza González, Andrés: « *Implementación de odometría visual utilizando una cámara estereoscópica*», Universidad de Chile, 2015

[4] ARKit tecnología desarrollada por Apple de Realidad Aumentada,
https://www.applesfera.com/apple-1/todo-que-necesitas-saber-arkit-2-espectacular-realidad-aumentada-apple

[5] ARCore tecnología desarrollada por Google de Realidad Aumentada,
https://whatis.techtarget.com/definition/ARCore

[6] Rajesh Desai, Parth, Nikhil Desai, Pooja y Mehta, Khushbu: « *A Review Paper on Oculus Rift-A Virtual Reality Headset*», International Journal of Engineering Trends and Technology (IJETT) – Volume 13 Number 4 – Jul 2014

[7] Todo sobre las gafas Oculus Rift de Realidad Virtual
https://www.gafasoculus.com/rift/

[8] IMU Oculus Rift DK1: MPU-6000
https://es.farnell.com/invensense/mpu-6000/ic-gyro-accel-9-axis-fusion-24qfn/dp/1862383?st=mpu%206000

[9] IMU Oculus Rift CV1: BMI055
`https://www.bosch-sensortec.com/bst/products/all_products/bmi055`

[10] Bloqueo de cardan (Gimbal lock)
`https://en.wikipedia.org/wiki/Gimbal_lock`

[11] Rotaciones y cuaternios
`http://tesis.uson.mx/digital/tesis/docs/21070/Capitulo4.pdf`

[12] Reséndiz Martínez, Rafael: «*La descripción hipercompleja como una alternativa en la solución de ecuaciones diferenciales*», Universidad Autónoma Metropolitana, 2009

[13] Cuaterniones
`https://es.wikipedia.org/wiki/Cuaterni%C3%B3n`

[14] Cuaterniones y rotaciones en el espacio
`https://es.wikipedia.org/wiki/Cuaterniones_y_rotaci%C3%B3n_en_el_espacio`

[15] Fusión sensórica blog Oculus Rift
`https://developer.oculus.com/blog/sensor-fusion-keeping-it-simple/`

[16] Woodman, Oliver J.: «*An introduction to inertial navigation*», University of Cambridge, 2007