

UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA
Facultad de Ciencias Químicas e Ingeniería
Maestría y Doctorado en Ciencias e Ingeniería



**Rastreo y Captura de Movimientos en Tiempo Real Basada
en Sensores de Inercia con Representación en un Entorno
de Realidad Aumentada**

TESIS
que para obtener el grado de
MAESTRO EN CIENCIAS

Sustenta:
HUMBERTO TRUJILLO PÉREZ

Bajo la dirección de:
DR. LEOCUNDO AGUILAR NORIEGA

Universidad Autónoma de Baja California

FACULTAD DE CIENCIAS QUÍMICAS E INGENIERÍA

FOLIO No. 254

Tijuana, B. C., a 7 de junio de 2018

C. Humberto Trujillo Pérez
Pasante de: Maestro en Ciencias
Presente

El tema de trabajo y/o tesis para su examen profesional, en la
Opción TESIS

Es propuesto, por el C. Dr. Leocundo Aguilar Noriega

Quienes serán los responsables de la calidad de trabajo que usted presente,
referido al tema Rastreo y Captura de Movimientos en Tiempo Real Basada en
Sensores de Inercia con Representación en un Entorno de Realidad Aumentada.

el cual deberá usted desarrollar, de acuerdo con el siguiente orden:

- I.- INTRODUCCIÓN
- II.- ANTECEDENTES Y MARCO TEORICO
- III.- ARMOCAP: RASTREO Y CAPTURA DE MOVIMIENTOS EN REALIDAD AUMENTADA
- IV.- CASO DE ESTUDIO: ASISTENTE DE ENTRENAMIENTO DEPORTIVO
- V.- EVALUACION Y RESULTADOS
- VI.- CONCLUSIONES

UNIVERSIDAD AUTONOMA
DE BAJA CALIFORNIA



FACULTAD DE CIENCIAS
QUÍMICAS E INGENIERÍA


Dr. José Luis González Vázquez
Sub-Director Secretario


Dr. Leocundo Aguilar Noriega
Director de Tesis


Dr. Luis Enrique Palafox Maestre
Director

Rastreo y Captura de Movimientos en Tiempo Real Basada en Sensores de Inercia con Representación en un Entorno de Realidad Aumentada

Humberto Trujillo Pérez

Resumen

Motion Capture, o captura de movimientos por su traducción al español, se refiere al proceso de rastrear, capturar y procesar movimientos reales para ser plasmados o representados en tiempo real sobre un modelo 3D conocido como *Avatar*. Este proceso es ampliamente utilizado no sólo en la industria de películas o videojuegos, sino también en otras áreas como robótica, tarapia física y ciencia del deporte.

Además de los procesos basados en técnicas ópticas, existe otro paradigma que utiliza un conjunto de dispositivos Micro Electromecánicos que, haciendo uso principalmente de acelerómetros y giroscopios, permiten realizar una medición de inercia corporal durante el movimiento. Este tipo de sistema se ha estado utilizando de manera cada vez más frecuente debido a que no depende de cámaras externas para su funcionamiento, por lo que el área permisible de movimiento se ve incrementada considerablemente.

Esta tesis de maestría presenta el diseño y construcción de un sistema de rastreo y captura de movimientos basado en sensores de inercia y su integración en un entorno de realidad aumentada. Dicho sistema está compuesto por una red de sensores de inercia inalámbricos vestibles con los que se obtienen datos de orientación en forma de quaternion, con los cuales es posible realizar un mapeo de movimientos realizados por el usuario a un avatar virtual en tiempo real.

En este trabajo también se abordó un caso de estudio, cuyo enfoque va dirigido hacía la aplicación de este sistema en el área de ciencia del deporte. Se trata de un asistente de entrenamiento deportivo en realidad aumentada, cuyo propósito es el de replicar los movimientos del usuario y mostrarlos en tiempo real. Esto con el objetivo de que el usuario pueda verse así mismo durante la activada deportiva y así desarrollar un criterio sobre su desempeño. De igual manera se pueden guardar sesiones de captura para un posible análisis posterior.

Índice

Resumen	ii
Índice	iii
Lista de Tablas	vi
Índice de Figuras	vii
1 Introducción	1
1.1 Antecedentes	2
1.1.1 Trabajo Relacionado	2
1.2 Planteamiento del Problema	3
1.3 Objetivos	3
1.4 Hipótesis	4
1.5 Estructura de Tesis	4
2 Marco Teórico	6
2.1 <i>Motion Capture</i>	6
2.1.1 Aplicaciones	7

2.1.2	Técnicas	8
2.2	Redes de Sensores Corporales	9
2.3	Sensores corporales	10
2.3.1	Sensores Fisiológicos	10
2.3.2	Sensores Biomecánicos	11
2.4	Unidad de Medición de Inercia	12
2.4.1	Características	12
2.4.2	Fusión de Datos	13
2.4.3	Quaternions	14
2.5	Realidad Aumentada	18
2.5.1	Vuforia	19
3	ARMoCap: Rastreo y Captura de Movimientos en Realidad Au-	
	mentada	21
3.1	Hardware	21
3.2	Colocación de Sensores	22
3.3	Mapeo de Segmentos	25
3.4	Cambio de Sistema de Coordenadas	26
3.5	Calibración	27
3.5.1	Compensación Inicial	27
3.5.2	Compensación por Importación de Modelo 3D	28
3.6	Resumen del proceso	29
3.7	Aplicación de Realidad Aumentada	31

3.8	Limitaciones del sistema	32
4	Caso de Estudio: Asistente de Entrenamiento Deportivo	34
4.1	Ciencia del Deporte	34
5	Evaluación y Resultados	35
5.1	Rendimiento	35
5.2	Deriva Acumulada	35
5.3	Escalabilidad del Sistema	36
6	Conclusiones	37
6.1	Aplicaciones	37
6.2	Trabajo futuro	37
A	Anexo: Hojas de Especificación Técnica	38
A.1	MPU6050	39
A.2	ESP8266	40
	Bibliografía y Referencias	41

Lista de Tablas

2.1	Multiplicación de componentes imaginarios de quaternion	15
2.2	Comparación de distintas plataformas para el desarrollo de aplicaciones de realidad aumentada	20

Índice de Figuras

2.1	Proceso de captura de movimiento	7
2.2	Uso de <i>Motion Capture</i> en la industria del cine.	7
2.3	Diseño teórico de una red de sensores corporales	10
2.4	Marco de Referencia Global	13
2.5	Visualización de propiedades de ij , jk , ki	15
2.6	Entorno real siendo aumentado con objetos virtuales.	18
3.1	Esquemático de nodo para captura de movimiento	22
3.2	Sensor dentro de su cubierta con cinta de velcro.	23
3.3	Configuración completa de colocación de sensores.	24
3.4	Modelo 3D de personaje utilizado como Avatar del usuario.	25
3.5	Orientación	26
3.6	Rotación de 180 grados en segmentos del avatar	29
3.7	Sistema jerárquico de Unity	31
3.8	Periféricos de realidad aumentada	31
3.9	Foto-captura del avatar siendo aumentado sobre el mundo real	32
3.10	Interfaz gráfica de usuario en realidad aumentada	32

Capítulo 1

Introducción

En la literatura se describe a *Motion Capture* como un proceso mediante el cual uno o varios dispositivos son utilizados para capturar patrones en el movimiento, los datos son transmitidos a una computadora donde un software de simulación asigna y muestra dicha captura representada por un actor virtual tridimensional conocido como *Avatar* [1].

El deseo de poder entender los movimientos del cuerpo humano y de otros seres vivos ha estado presente por siglos. Sin embargo, no fue sino hasta hace poco más de una década que gracias al desarrollo de nuevas tecnologías en el área de la computación se han podido desarrollar los métodos necesarios para observar y estudiar la cinemática del cuerpo humano. No obstante, dependiendo de la calidad y realismo que se desean obtener, el proceso se vuelve más complejo. En la mayoría de los casos, las lecturas obtenidas deben pasar por varias fases donde se les realizan ajustes como calibración y filtrado para obtener resultados óptimos.

Hoy en día, el proceso de *Motion Capture* toma un papel indispensable en la industria de las películas y videojuegos, así como en otras áreas como la interacción humano-computadora, medicina y ciencia del deporte. Esto hace de este campo de estudio un tema bastante interesante debido a su amplia variedad de aplicaciones.

1.1 Antecedentes

1.1.1 Trabajo Relacionado

En el mercado actual existe una amplia variedad en sistemas de Motion Capture cuya orientación específica se encuentra en el área del deporte, los cuales son excelentes añadiendo un valor agregado a atletas, equipos deportivos, hospitales e instalaciones privadas. La mayoría de estos sistemas están basados en técnicas ópticas. Sin embargo, desde que la tecnología de IMU ha estado disponible al público en general, el campo de Motion Capture basado en sensores de inercia ha ido creciendo.

Vicon es una compañía en el Reino Unido que se destaca por su sistema de captura basado en técnicas ópticas, pero recientemente adquirieron *IMeasureU*, una nueva empresa en Nueva Zelanda cuyo producto principal se trata de un solo sensor IMU para atletas. El enfoque principal de Vicon se encuentra en el desempeño deportivo e investigación biomecánica, aunque también han tenido éxito en área del entretenimiento. Líderes en biomecánica y expertos en la ciencia del movimiento utilizan Vicon como una forma de cuantificar su trabajo con atletas y pacientes [2].

Xsens es un pionero en Motion Capture utilizando dispositivos wearables y es uno de los líderes en análisis de movimientos basada en sensores de inercia. Esta compañía está situada en Holanda, pero también tiene una alta presencia en los Estados Unidos. Sus soluciones varían desde el deporte hasta la agricultura, ya que busca más que solo aportar al estudio e investigación. Están muy conectados al mercado y tienen muchas otras aplicaciones fuera del área de la ciencia y el deporte [3].

Entre las pocas compañías cuyos sistemas de Motion Capture están basados en sensores de inercia se encuentran *Xsens*, *Perception Neuron* [4] y *Notch* [5]. El enfoque que presentan difiere uno de otro, por ejemplo, Xsens utiliza Wifi como vía de comunicación entre sus sensores, mientras que Perception Neuron utiliza cables. Notch utiliza una combinación de estos dos métodos, sus sensores se comunican mediante Wifi para después enviar los datos a un teléfono inteligente mediante Bluetooth. Los precios también varían. El sistema de Notch necesita doce sensores para cubrir el cuerpo completo, lo que al momento cuesta alrededor de \$700 USD. El sistema de Perception Neuron también cubre el movimiento de los dedos, para lo que necesita dieciocho sensores para un total de \$1000 USD.

1.2 Planteamiento del Problema

Los sistemas de *Motion Capture* comúnmente están basados ya sea en técnicas ópticas o en sensores de inercia. Por lo general, los sistemas ópticos son utilizados en grandes producciones de películas, videojuegos y en otras aplicaciones que requieran una precisión considerablemente alta. Sin embargo, estos sistemas de tecnología de punta conllevan varias desventajas. Por ejemplo, requieren del uso de cámaras especializadas, así como de trajes especiales que deben usar los usuarios. Además, los sistemas ópticos dependen en gran medida de condiciones ambientales específicas y son bastante costosos.

Por lo anterior mencionado, la integración de un sistema de captura de movimientos basado en técnicas ópticas en un entorno de realidad aumentada resultaría inconveniente cuando se busca un alto grado de inmersión y libertad de movimiento.

Debido a que los sistemas basados en sensores de inercia tienen un tiempo relativamente corto de existencia, estos aún no se han desarrollado en su totalidad. Estos sistemas se han vuelto un nuevo campo de estudio en el área de interacción humano-computadora dado su bajo costo y cercanía con el desarrollo de tecnologías emergentes, como lo es la realidad aumentada.

1.3 Objetivos

General:

Diseño y construcción de un sistema de rastreo y captura de movimientos realizados por personas durante actividades física mediante el uso de sensores de inercia para ser representados en un ambiente de realidad aumentada.

Específicos:

- Lograr la integración de una red de sensores corporales en base a unidades de medición de inercia.
- Modelo matemático para el mapeo de los movimientos de un usuario a un modelo 3D virtual.

- Representación de movimientos en tiempo real en un entorno de realidad aumentada.

1.4 Hipótesis

Mediante una serie de operaciones de multiplicación entre tipo de dato quaternion es posible el mapeo de las oerientaciones obtenidas por unidades de medición de inercia a un modelo virtual tridimensional para un completo rastreo y captura de movimientos.

1.5 Estructura de Tesis

Esta tesis empieza presentando una breve discusión acerca de los trabajos e investigaciones recientes relevantes en el área, los cuales ayudan a justificar la necesidad de esta investigación. También se plantean las problemáticas que se atacan y los objetivos que se tuvieron durante para este trabajo.

En el segundo capítulo se presenta el marco teórico relevante a esta investigación. Se abordan temas como el proceso de *Motion Capture*, redes de sensores corporales y tipos de sensores. Así mismo se habla sobre las unidades de medición de inercia, las cuales son la base fundamental en el desarrollo de nuestro sistema de captura de movimientos. Finalmente se habla sobre realidad aumentada, haciendo énfasis en las tendencias actuales en tecnologías y plataformas de desarrollo.

Dando paso al capítulo tercero, se describe a detalle el procedimiento llevado a cabo durante esta investigación. Empezando por la descripción del hardware utilizado y la comunicación entre dispositivos. Después se aborda la parte de procesamiento de los datos obtenidos por los sensores, los cuales se encuentran en forma de quaternion, para ser utilizados para representar los movimientos del usuario con un personaje virtual 3D.

En el capitulo cuarto se presenta un caso de estudio, el cual se utilizó como base para poder evaluar nuestro sistema de captura de movimientos. Se trata de un asistente de entrenamiento deportivo, cuyo objetivo es el de ayudar a deportistas a

llevar un control sobre su progreso durante su actividad deportiva.

En siguiente capítulo se presentan los resultados obtenidos con el sistema de captura de movimientos en un ambiente de realidad aumentada, se hace énfasis en el rendimiento del sistema, sus limitaciones y la experiencia de usuario que se lograron. Finalmente se presentan las conclusiones a las cuales se llegó en esta investigación y el posible trabajo futuro.

Capítulo 2

Marco Teórico

2.1 *Motion Capture*

La captura de movimientos, *Motion Capture*, es una forma de almacenar los movimientos de una persona de forma digital. Los datos almacenados son mapeados a un modelo tridimensional dentro de un software de gratificación 3D con el fin de que el personaje, mejor conocido como Avatar, se mueva de la misma forma que lo hace el usuario.

Es necesario tener presente la diferencia que existe entre captura y rastreo de movimientos, como lo muestra la figura 2.1. El rastreo de movimiento, conocido en el literatura como *Motion Tracking*, involucra las partes de sensar y procesar los datos de movimiento, mientras que la captura involucra también el almacenamiento de la información para su uso posterior. Por naturaleza, los datos de los sensores o cámaras pueden ser almacenados directamente, pero no son muy útiles en su estado crudo. Estos datos necesitan ser procesados para que la información del movimiento pueda ser de utilidad.

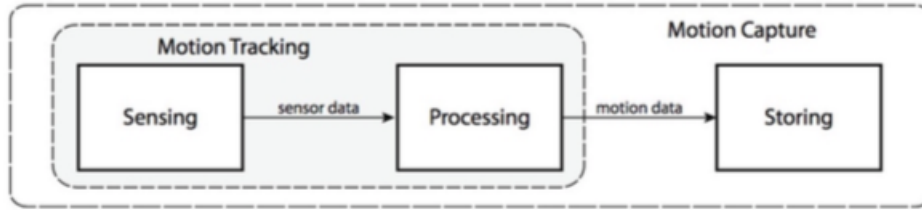


Figura 2.1: El proceso de *Motion Capture* consiste de tres pasos generales: sensado, procesamiento y almacenamiento [6].

2.1.1 Aplicaciones

Existe una amplia variedad de campos de estudio y entretenimiento en donde el proceso general de Motion Capture toma una parte bastante significativa. La industria de entretenimiento probablemente es el usuario más visible y reconocido por utilizar tecnología de captura de movimientos.

En el campo de los videojuegos se utiliza para animar los movimientos de los personajes, con esto se ahorra mucho tiempo valioso de producción. Hoy en día, casi todos los juegos recientes usan esta técnica para los movimientos de personajes y objetos dinámicos.

En la industria cinematográfica se utiliza esta tecnología para simular o aproximar la apariencia de actores con modelos de personajes digitales fotorealistas. Tenemos el ejemplo del personaje de Gollum de la película Señor De Los Anillos, donde se utilizó *Motion Capture* para darle vida al personaje, figura 2.2.



Figura 2.2: Uso de *Motion Capture* en la industria del cine.

En el campo de la medicina se puede utilizar esta tecnología para asistir con la obtención de diagnósticos y tratamientos de padecimientos físicos. Los datos de movimientos capturados también se pueden utilizar para mostrar progresos en los pacientes. Los datos de sesiones anteriores se pueden comparar en tiempo real para ver si la rehabilitación está teniendo los resultados deseados [7].

También en el área del deporte, el cual cada vez se vuelve más competitivo y siempre se busca mejorar el desempeño, Motion Capture se puede utilizar para analizar cada detalle de ciertos movimientos para lograr modificaciones de mejora de desempeño [8, 9].

2.1.2 Técnicas

Años de desarrollo tecnológico han dado como resultado varios tipos de sistemas, cuyo propósito es el mismo pero la técnica y la forma de implementación difiere uno de otro. El estudio presentado en [10] compara los diferentes técnicas de *Motion Capture*. Los siguientes tipos de sistema son las categorías donde comúnmente se agrupan los sistemas de captura de movimientos.

- **Magnéticos:** Estos utilizan sensores colocados sobre el cuerpo para medir los campos magnéticos de baja frecuencia generados por una fuente emisora. La desventaja principal en este aspecto es que a medida que incrementa la distancia entre los sensores y el emisor disminuye el alcance de los campos magnéticos, por lo que el área permisible de movimiento se ve reducida.
- **Ópticos:** Esta probablemente sea la técnica más común, la cual hace uso de marcadores pasivos colocados sobre el cuerpo del usuario en lugares predefinidos. Múltiples cámaras se encargan de seguir dichos marcadores que están alineados con ciertos puntos de referencia para recrear al usuario como un modelo 3D. El problema más grande con esta técnica es la obstrucción de visibilidad, conocido como oclusión.
- **Inercia:** La medición de inercia es en lo que se basa esta tesis. Con esta técnica se puede capturar los movimientos sin la necesidad de utilizar cámaras, por lo que no limita el área permisible de movimiento. Aquí el usuario viste sensores con que se obtienen datos de aceleración y orientación, los cuales son enviados a una computadora para ser visualizados en un programa de 3D. La desventaja

de utilizar estos sensores es que tienden a acumular errores debido al ruido en la lecturas.

2.2 Redes de Sensores Corporales

Una red de sensores corporales, BASN por sus siglas en inglés, es una red de sensores que es usada para determinar el estado actual de un individuo y los cambios que se presentan en éste. La red está compuesta de varios sensores que comúnmente son colocados sobre el cuerpo, aunque en casos especiales éstos también pueden estar implantados dentro del cuerpo. El tipo, la distribución y el número de sensores que se incluyen en una red varían dependiendo de los requerimientos específicos de la aplicación [11].

El enfoque médico es un prospecto donde una BASN puede resultar muy útil. Estos permite monitorio pacientes de forma remota [12, 13], detectar caídas de personas de la tercera edad [14], y mejores tratamientos en terapias físicas. También se han examinado aplicaciones en el área de entrenamiento deportivo profesional y entrenamiento físico en general. Lo cual es el tema de nuestro caso de estudio en esta trabajo de tesis. No obstante, los métodos que son abordados aquí se pueden aplicar a cualquier otra área donde una red de sensores corporales resulte de utilidad.

Cada sensor en la red, comúnmente denominado Nodo, sensa información correspondiente a un estado particular del usuario. Estos mismos datos son pasados a algún tipo de transmisor central también localizado sobre el cuerpo. El transmisor manda toda la información obtenida por los sensores a un dispositivo de procesamiento principal. El diseño de una arquitectura básica de una red de sensores corporales se puede apreciar en la figura 2.3, donde se muestran distintos componentes y cómo es transmitida la información de cada uno de ellos [15].

Dependiendo del diseño de la red y de los sensores, este transmisor central puede no estar presente, y en su lugar cada sensor transmite su propia información al receptor conectado al procesador principal, lo cual es caso de la configuración que se presenta en este trabajo.

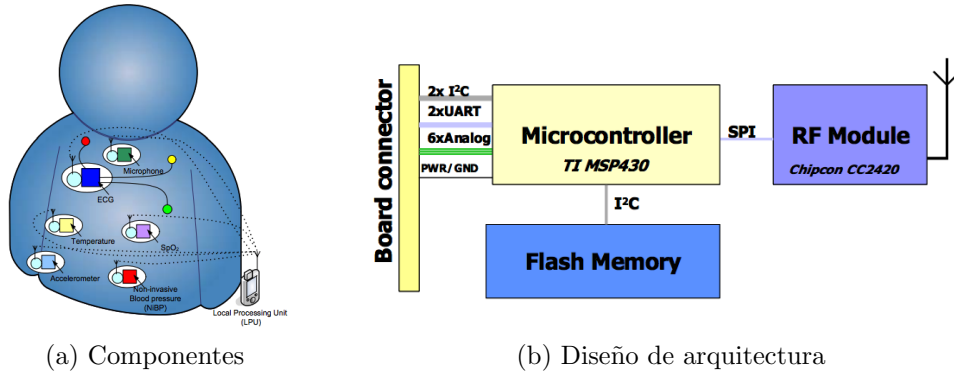


Figura 2.3: Diseño teórico de una red de sensores corporales [16]

2.3 Sensores corporales

Los avances tecnológicos en los últimos años han tenido un gran impacto en la mejora de las BASN. Los sensores se han vuelto mucho más pequeños con el desarrollo de tecnologías como los Sistemas Micro-Electromecánicos, conocidos como MEMS por sus siglas en inglés, logrando que los sensores sean menos estorbosos cuando son vestidos. Tecnologías de transmisión inalámbrica han permitido que muchos de estos sistemas se puedan comunicar de una forma más sencilla y puedan operar con baterías, permitiendo así un área de operación más extensa. El mejoramiento de velocidades de procesamiento de datos ha permitido explorar el campo de aplicaciones de tiempo real.

2.3.1 Sensores Fisiológicos

Existen un buen número de sensores que pudieran ser incorporados en una BASN. Podemos dividir este número en dos categorías principales. La primera categoría son los sensores fisiológicos, los cuales detectan el estado actual y los cambios de propiedades fisiológicas. Los siguientes ejemplos se pueden considerar como parte de esta categoría.

- **Sensor de ritmo cardíaco:** Estos detectan el ritmo cardíaco de una persona en pulsos por minuto. Esto se logra a través de pulseras, clips en los dedos o con electrodos. El ritmo cardíaco es usado para detectar factores como nivel esfuerzo en alguna actividad física [17, 18], estrés y frustración [19].

- **Electromiografía (EMG):** Cuando los músculos se contraen se produce una diferencia de potencial eléctrico que se puede medir sobre la superficie de la piel colocando electrodos cerca del músculo que se desea medir. Estos sensores pueden ser usados para medir contracciones que se hacen de forma voluntaria o de forma involuntaria para medir niveles de estrés [20].
- **Interfaz Cerebro-Computadora (BCI):** De forma similar a los músculos, el cerebro presenta señales eléctricas detectables cuando éste se encuentra trabajando. Los sistemas que detectan estas señales usando electrodos u otras técnicas se pueden utilizar para obtener un amplio rango de señales como medio de entrada para controlar otras cosas o simplemente para detectar cierto tipo de señales.
- **Sensor de respiración:** Este tipo de sensores por lo regular usan bandas alrededor del pecho. Esta banda se expande con el pecho para medir el ritmo de la respiración [20].

Este tipo de sensores miden propiedades fisiológicas presentes en el usuario, de las cuales éste puede o no tener control. Estos sensores se pueden utilizar en un amplio rango de aplicaciones, principalmente en el campo médico donde las señales biológicas se pueden monitorear y analizar [12].

2.3.2 Sensores Biomecánicos

La segunda categoría de sensores corporales son biomecánicos, los cuales son el tipo de sensor que se utilizan en este trabajo. Estos son usados para medir la posición física y los movimientos de un individuo. Los siguientes sensores hacen posible el funcionamiento de nuestro sistema de captura de movimientos.

- **Acelerómetro:** Estos miden cualquier aceleración que ocurra sobre los ejes del dispositivo. Debido a esto, comúnmente se utilizan acelerómetros de 3 ejes para poder medir aceleraciones en cualquier dirección. Los acelerómetros proveen información sobre su ángulo de inclinación con respecto a dirección hacia abajo mediante la aceleración de la gravedad. Sin embargo, no les es posible distinguir entre esta fuerza gravitacional y otras aceleraciones [21]. Aunque realizando una doble integración teóricamente se podría obtener información

de posicionamiento, el ruido en los datos hace que esto sea en extremo difícil e impreciso, incluso aplicando distintos tipos de filtrado [22].

- **Giroscopio:** Mientras que los acelerómetros detectan aceleraciones en forma lineal, los giroscopios miden la velocidad angular. Sin embargo, estos carecen de la capacidad de determinar su orientación absoluta y hasta los giroscopios de mejor calidad padecen problemas de *drift* [22]. *Drift*, también conocido como deriva, sucede cuando se presentan pequeños cambios de velocidad angular aún cuando el dispositivo se encuentra completamente en reposo. Con el tiempo, esta deriva se acumula y puede resultar en un problema mayor. Por esto, los giroscopios se adecuan más para detectar rotaciones rápidas que para rotaciones prolongadas.
- **Magnetómetro:** Como su nombre lo indica, estos son sensibles a los campos magnéticos y comúnmente son usados para conocer la dirección del campo magnético local Terrestre usándola como referencia de la misma forma que un acelerómetro usa la dirección de fuerza de gravedad. El campo magnético de la Tierra es débil y fácilmente puede ser superado por algún objeto metálico o dispositivos electrónicos cercanos, por lo que esto se debe tener en consideración al usar magnetómetros.

Cuando se utilizan de manera conjunta acelerómetros, giroscopios y magnetómetros nos referimos a ellos como una unidad de medición de inercia, IMU por sus siglas en inglés. Esto es debido a la forma en la que operan, haciendo uso de las leyes físicas de la masa e inercia para determinar los movimientos y orientaciones. Combinando los datos obtenidos de cada uno de ellos se obtiene una mejor medición que con cualquier sensor individual, a este proceso se le conoce como fusión de datos.

2.4 Unidad de Medición de Inercia

2.4.1 Características

Para detectar la orientación de una parte del cuerpo en específico se utilizó de forma conjunta el microcontrolador ESP8266 módulo ESP-01 y un MPU-6050 de la compañía InvenSense [23]. El MPU-6050 se compone de un acelerómetro y un giroscopio, ambos de 3 ejes. Cada dispositivo, al cual llamaremos sensor, es alimentado

por una batería de litio de 3.7V, pero debido a que el ESP8266 solo trabaja con un voltaje de 3.3V fue necesario incluir un regulador de voltaje externo. El microcontrolador ESP8266 cuenta con un módulo de Wi-Fi 2.4GHz con el cual se transmiten los datos de orientación a un teléfono inteligente bajo el protocolo TCP/IP.

Cada sensor presenta como salida su orientación actual en forma de un quaternion, los cuales se ven a mayor detalle en las secciones posteriores. Al iniciar, el sensor genera su propio marco de referencia global tal que el eje z positivo está orientado verticalmente hacia arriba y los ejes x y y son determinados por la orientación del dispositivo, como se muestra en la figura 2.4. Posteriormente, el MPU de cada sensor realiza lecturas del acelerómetro y giroscopio para calcular una estimación de orientación actual, la cual es relativa a sus ejes iniciales.

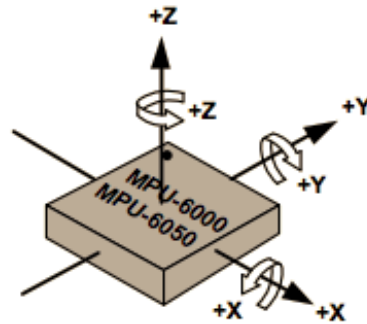


Figura 2.4: Marco de referencia global generado por el sensor de inercia.

Al proceso descrito anteriormente se le conoce como fusión de datos y existe una gran cantidad de información en la literatura acerca de éste [24][25][26]. Los detalles del algoritmo de fusión de datos utilizado por InvenSense no se encuentran en la documentación proporcionada del dispositivo. Sin embargo, el concepto básico de fusión de datos es cubierto en la siguiente sección.

2.4.2 Fusión de Datos

El giroscopio del IMU puede medir cualquier rotación que sea efectuada sobre el sensor, pero no le es posible determinar una orientación absoluta por sí mismo. Con el acelerómetro es posible obtener la dirección en la que se encuentra aplicada la fuerza de gravedad, ésta es usada como una referencia invariante que ayuda a deter-

minar dónde queda la dirección hacia abajo. El giroscopio es usado durante lapsos de tiempo cortos para determinar cambios en la orientación y la referencia constante del acelerómetro es usada en periodos de tiempo prologados para compensar por el desvío erróneo del giroscopio. Filtros como Kalman u otros algoritmos de filtrado similares determinan la forma en que cada sensor contribuye al cálculo general de orientación [27][28].

El acelerómetro solamente provee una referencia para la dirección vertical utilizando la gravedad. Debido a esto, el giroscopio puede desviarse libremente sobre el eje vertical, esto resulta en que los ejes x y y globales del sensor roten lentamente con respecto al marco de referencia global. A esto se le conoce como *sensor drift*. Por esta razón los magnetómetros son utilizados frecuentemente para anclar esta rotación utilizando la dirección del campo magnético de la Tierra, de la misma forma en el acelerómetro utiliza la gravedad. El MPU6050 no cuenta con magnetómetro embebido por lo que es susceptible al *sensor drift*.

2.4.3 Quaternions

Los quaternion son la base fundamental de este trabajo. Cada sensor presenta como salida su orientación en forma de quaternion, los cuales utilizamos para mapear los movimientos del usuario al modelo 3D, el avatar. En esta sección cubriremos la teoría detrás de los quaternion, la misma que nos ayudará a comprender a detalle las matemáticas detrás del proceso de rastreo y captura de movimientos.

Los quaternions son un sistema numérico originalmente propuesto por William Hamilton en 1843 [29]. Los quaternions son una extensión de los números complejos que nos ayudan a rotar puntos en tres dimensiones. Dicha extensión se realiza añadiendo otros dos componentes imaginarios al sistema numérico complejos, para un total de tres componentes imaginarios, que representan cada uno de los ejes en un sistema de coordenadas, y un número real escalar. La forma general para expresar un quaternion es:

$$q = (w, x\hat{i}, y\hat{j}, z\hat{k}) \quad (2.1)$$

En la expresión anterior, $w, x, y, z \in \mathbf{R}$, mientras que $\hat{i}$, $\hat{j}$ y $\hat{k}$ son componentes imaginarios que representan cada uno de los ejes de un sistema de coordenadas.

Aunque es más común ver representado un quaternion como un par ordenado.

$$q = (w, \vec{v}) \quad (2.2)$$

De acuerdo a la famosa expresión de Hamilton se tiene que:

$$i^2 = j^2 = k^2 = ijk = -1 \quad (2.3)$$

Lo primero que hay que tener en cuenta al trabajar con quaternions es que no son conmutativos. La tabla 2.1 muestra los resultados de la multiplicación entre sus componentes imaginarios.

Tabla 2.1: Multiplicación de componentes imaginarios de quaternion

	i	j	k
i	-1	k	-j
j	-k	-1	i
k	j	i	-1

En la figura 2.5 se puede visualizar la relación entre los vectores unitarios de un sistema cartesiano representados por i , j y k .

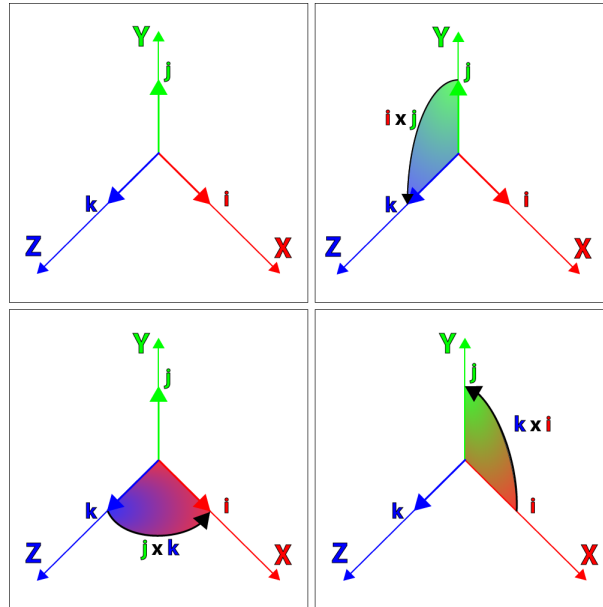


Figura 2.5: Visualización de propiedades de ij , jk , ki

La multiplicación entre quaternion es la operación clave que nos permite rotar un punto en tres dimensiones. Se puede multiplicar dos quaternion utilizando operaciones comunes sobre vectores.

$$q_1 q_0 = (w_1 w_2 - \vec{v}_1 \cdot \vec{v}_0, w_1 \vec{v}_0 + w_0 \vec{v}_1 + \vec{v}_1 \times \vec{v}_0) \quad (2.4)$$

La siguiente operación 2.5 muestra el resultado de la multiplicación de dos quaternion a y b .

$$\begin{aligned} q &= (a_w, a_x, a_y, a_z) * (b_w, b_x, b_y, b_z) \\ &= (a_w b_w - a_x b_x - a_y b_y - a_z b_z, \\ &\quad (a_w b_x + a_x b_w + a_y b_z - a_z b_y) \hat{i}, \\ &\quad (a_w b_y + a_y b_w - a_x b_z + a_z b_x) \hat{j}, \\ &\quad (a_w b_z + a_z b_w + a_x b_y - a_y b_x) \hat{k}) \end{aligned} \quad (2.5)$$

Rotación

Al igual que con los números complejos, los quaternion unitarios, quaternions cuya magnitud es uno, nos ayudan a representar una rotación. Para rotar un objeto en tres dimensiones con un ángulo θ sobre un eje unitario $\vec{n}$ se tiene:

$$q = \left(\cos \frac{\theta}{2}, \vec{n} \sin \frac{\theta}{2} \right) \quad (2.6)$$

Nuestro modelo debe ser capaz de hacer y deshacer rotaciones. Se puede obtener el conjugado de un quaternion q negando la parte que contiene el vector. Se denota el conjugado de un quaternion con el superíndice $*$, .

$$q^* = (w, -\vec{v}) \quad (2.7)$$

Esto nos permite también definir la inversa, q^{-1} , de un quaternion q , tal que $qq^{-1} = 1$.

$$q^{-1} = \frac{q^*}{qq^*} \quad (2.8)$$

Para rotar a un quaternion q_0 en su estado actual por una cantidad especificada

por q_1 , se multiplica por q_1 por la izquierda, recordando que la propiedad conmutativa no está presente, como se muestra en la ecuación 2.9.

$$q_2 = q_1 q_0 \quad (2.9)$$

Este quaternion q_2 es equivalente a rotar primero por q_0 y luego por q_1 . Una serie de rotaciones puede simplemente ser representada por una serie de multiplicaciones entre quaternions. Se debe tener en consideración que la propiedad conmutativa no aplica en la multiplicación de quaternions, por lo que el orden en que se realizan estas operaciones es muy importante. En este caso el orden de los factores no altera el producto.

Un vector estándar en tres dimensiones, $\vec{r}$, puede ser representado en forma de quaternion haciendo que su componente real sea igual a cero.

$$\vec{r} = (x, y, z) \equiv q_r = (0, x\hat{i}, y\hat{j}, z\hat{k}) \quad (2.10)$$

Se puede rotar este vector por cualquier quaternion de rotación q multiplicando el vector por ambos lados.

$$\begin{aligned} \vec{r}' &= q\vec{r}q^{-1} \\ &= q\vec{r}\left(\frac{q^*}{\|q\|}\right) \\ &= q\vec{r}q^* \end{aligned} \quad (2.11)$$

El primer paso en la ecuación anterior parte de la ecuación 2.8 y del hecho de que qq^* es igual a la magnitud de q . El segundo paso es el resultado de rotaciones con quaternion de magnitud de uno.

Interpolación

Una de las razones más importantes para usar quaternions en gráficos de computadora es que son muy buenos representando rotaciones en el espacio. Los quaternions superan los problemas presentes en otros métodos de rotación como el bloqueo del cardán, *Gimbal Lock*.

Los quaternions tienen muchos beneficios sobre otras representaciones de rota-

ciones [30]. No se requiere del cálculo de muchas funciones trigonométricas, por lo que son eficientes en computo. Uno de sus mayores beneficios es que no son afectados por los problemas del Bloqueo del cardán, conocido en la literatura como *Gimbal Lock*, el cual está presente en métodos como el de Ángulos de Euler [31]. A pesar de sus beneficios, el concepto de quaternion es difícil de comprender y tampoco son fáciles de visualizar.

2.5 Realidad Aumentada

La realidad aumentada se define como una extensión de la realidad física añadiendo capas de información generada por computadora al ambiente real. En otras palabras, *AR* tiene la capacidad de mejorar cualquier objeto real con información adicional valiosa. Esta información puede ser cualquier tipo de contenido virtual u objeto, incluyendo texto, vídeo, gráficas, sonidos, datos de GPS, incluso hasta aromas [32]. Esta tecnología ha sido usada por más de una década en varios campos, como en la medicina, ingeniería, y entretenimiento para mejorar la percepción que tiene el usuario del mundo real [33].

Típicamente, los sistemas de realidad aumentada utilizan un sistema de rastreo basado en técnicas de *computer vision*. Esta técnica es precisa, pero solo funciona correctamente sobre distancias cortas [34] y por lo regular requiere de distintos marcadores colocados alrededor del mundo real. No obstante, también se han propuesto alternativas para remediar este problema, como el uso de GPS [35], rastreo de estructuras 3D en lugar de imágenes 2D [36, 37], y el uso de sensores de inercia.



Figura 2.6: Entorno real siendo aumentado con objetos virtuales [38].

En este trabajo de tesis se propone un sistema de realidad aumentada en donde el usuario está completamente inmerso en el ambiente de realidad aumentada con el uso de un sistema de captura de movimientos en tiempo real. Mediante esta implementación se incrementa en gran medida las posibilidades de interacción entre el usuario y el mundo virtual, con lo que se logra obtener una mayor libertad de movimiento.

En la actualidad existen varias bibliotecas de desarrollo de realidad aumentada. Puesto que se está usando Unity como entorno principal de desarrollo, se exploraron distintas bibliotecas de realidad aumentada compatibles con Unity., siendo *Vuforia* como la herramienta de desarrollo elegida.

2.5.1 Vuforia

Vuforia es una de las plataformas más populares para el desarrollo de aplicaciones de realidad aumentada. Aquí se implementan funcionalidades como reconocimiento de distintos tipos de objetos visuales, texto y ambiente. A su vez, la integración con Unity es bastante simple.

Todas las funcionalidades de esta plataforma están disponibles de forma gratuita pero incluyen una marca de agua. Las únicas limitaciones son la cantidad de marcadores disponibles y algunos servicios de reconocimiento con la nube de Vuforia. La funcionalidad principal en Vuforia de la cual se hace uso en este trabajo es la de detección de superficies planas, la cual permite colocar contenido virtual sobre el piso o alguna superficie plana en el mundo real.

En se realizó una investigación más exhaustiva sobre las ventajas y desventajas de distintas plataformas de desarrollo para realidad aumentada. Como se observa en la tabla 2.2, Vuforia resulta como la elección más viable [39].

Tabla 2.2: Comparación de distintas plataformas para el desarrollo de aplicaciones de realidad aumentada

	Vuforia	ARToolKit	Kudan	MaxST	DeepAR
Distancia máxima	1.2 / 3.7	3 / 3	0.8 / 3	0.5 / 0.9	0.7 / 5
Estabilidad	10	8	10	7	8
Ángulo mínimo de reconocimiento	30	10	30	50	35
Reconocimiento 2D	Sí	Sí	Sí	Sí	Sí
Reconocimiento 3D	Sí	–	Sí	Sí	–
Geo-Localización	–	–	–	–	–
Reconocimiento en la nube	Sí	–	–	–	–
Total (rating)	7.7	2.8	6.9	5.2	4.7

Capítulo 3

ARMoCap: Rastreo y Captura de Movimientos en Realidad Aumentada

En este capítulo se describe a detalle el proceso y el desarrollo que se abordó en esta tesis. Empezando por la descripción del hardware utilizado, la configuración de los sensores de inercia que permiten el rastreo y captura de movimientos y el modelo matemático implementado para llevar a cabo los objetivos.

3.1 Hardware

Cada nodo en la red de sensores está compuesto por un microcontrolador ESP8266, el cual cuenta con un módulo WiFi que utilizamos para la comunicación con el dispositivo móvil, donde es ejecutada la aplicación de realidad aumentada. La unidad de medición de inercia es cubierta mediante el MPU6050, el cual cuenta con un módulo de DMP, *Digital Motion Processor*, que utiliza para calcular orientación en base a la combinación del acelerómetro y giroscopio.

La comunicación entre el ESP8266 y el MPU6050 es llevado a cabo mediante el protocolo I2C, el cual es manejado por dos señales. SCL es la señal de reloj que es usada para sincronizar todas las transferencias de datos, y SDA es la línea por donde se transmiten los datos.

Dado que tanto el microcontrolador como la unidad de medición e inercia deben ser alimentados con un voltaje no superior a los 3.3V para su correcto funcionamiento, se incluyó un regulador de voltaje que otorga el nivel de voltaje necesario. En la figura 3.1 se muestra el esquemático de un nodo sensor.

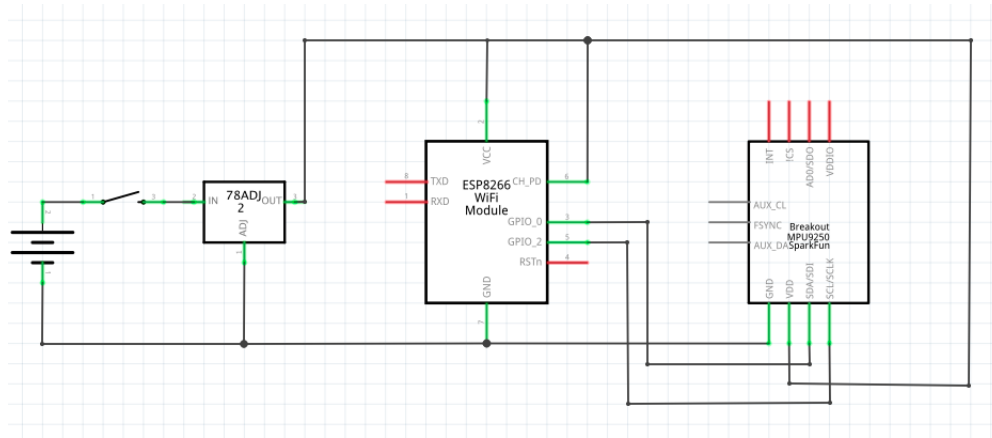


Figura 3.1: Esquemático de nodo para captura de movimiento

3.2 Colocación de Sensores

El propósito del sistema es poder capturar la postura y movimientos de una persona. Para lograr esto se adquirieron cubiertas de plástico para sostener los sensores. También se añadieron cintas de velcro para poder colocarlos al cuerpo de forma sencilla, en la figura 3.2 se muestra uno de los sensores ya con cubierta y cinta. Al asegurar un sensor a un segmento del cuerpo, también denominado *hueso*, los cambios en orientación de dicha parte son equivalentes a los cambios en la orientación del sensor colocado sobre éste. Este sistema de captura es modular, por lo que el usuario puede decidir cuántos sensores incluir en su sesión de captura.

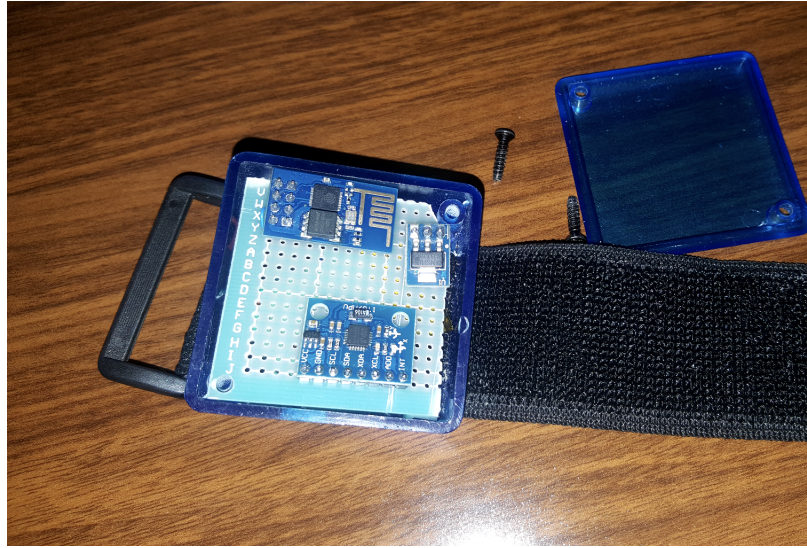


Figura 3.2: Sensor dentro de su cubierta con cinta de velcro.

Un sensor es asignado a cada parte del cuerpo que se quiera capturar y cada uno de ellos fue programado para operar sobre su propia conexión al dispositivo móvil que corre la aplicación de realidad aumentada mediante sockets. En la figura 3.3 se observa una configuración completa del sistema colocando diez sensores de inercia sobre el cuerpo. A su vez, cada sensor cuenta con su propio identificador, con el cual nos es posible asignarlos con mayor facilidad. Por ejemplo, el sensor "1" es el antebrazo derecho, el "2" es el hombro derecho, etc. De esta forma el sistema puede atribuir la orientación recibida por ese sensor a la parte del cuerpo correspondiente dentro del programa.

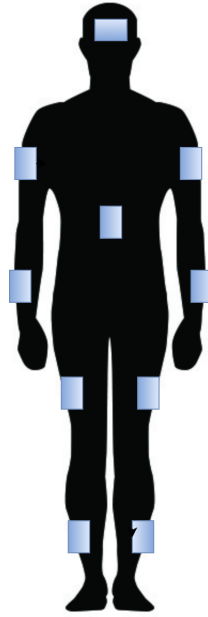


Figura 3.3: Configuración completa de colocación de sensores.

Se adquirió un personaje 3D, figura 3.4, de la plataforma en línea de Unity [40], el cual representa al Avatar del usuario dentro del ambiente de realidad aumentada. El modelo consiste de todos los segmentos de cuerpo necesarios para realizar el mapeo con los sensores de forma correcta. Dicho modelo fue importado a Unity [41] para poder desarrollar el código que permite animar al personaje en base de los datos de orientación provenientes de los sensores. Partes del cuerpo a las que no se les realiza mapeo son configurados con cuerpos rígidos, por lo que se mueven y rotan en conjunto con su segmento padre.



Figura 3.4: Modelo 3D de personaje utilizado como Avatar del usuario.

3.3 Mapeo de Segmentos

Al iniciar el sistema, cada sensor genera su propio marco global de referencia. Todas las orientaciones obtenidas como quaternion representan rotaciones con respecto al marco de referencia inicial. En la figura 3.5 se muestra el marco de referencia inicial generado por el sensor, su orientación al ser colocado en el hombro del usuario, y la orientación inicial del hombro del personaje en el sistema de coordenadas de Unity. El quaternion obtenido por el sensor en segundo panel ha tenido una rotación de 90 grados con respecto al eje y . El marco global del sensor difiere al de Unity. Unity tiene en su sistema de coordenadas al eje z apuntando hacia dentro de la pantalla, mientras que en el sensor este eje está apuntado hacia arriba. Además, el sistema de coordenadas de Unity sigue una regla de mano izquierda para sus rotaciones, mientras que el sensor utiliza una de mano derecha. Si esta rotación tal cual es aplicada al brazo del personaje no solo causaría que el brazo del modelo rote fuera de su posición inicial, sino que también rotaría sobre los ejes equivocados.



Figura 3.5: Orientación

Para poder realizar el mapeo de los movimientos del usuario al personaje se ocupa tener en consideración los siguientes puntos. Primero es necesario convertir el quaternion del marco global del sensor al sistema de coordenadas de Unity para hacer que la dirección de las rotaciones sea la correcta. Después se necesita encontrar una forma de compensar el valor del quaternion para que la orientación de las partes del personaje sean iguales a las del usuario al inicio del programa.

3.4 Cambio de Sistema de Coordenadas

Los dos sistemas de coordenadas se muestran en la figura 3.5. en las cuales se observa que el eje x de ambos sistemas se encuentran en paralelo. También se puede observar que bajo esa condición el eje z de Unity va dirigido hacia dentro de la pantalla, correspondiente al eje y del sensor. De igual forma, el eje vertical y de Unity corresponde al eje z del sensor. Esto nos da el mapeo básico que se necesita, solamente intercambiando los ejes y y z .

Sin embargo, debido a que Unity maneja un sistema de rotación de mano izquierda, todas las rotaciones del sensor, al ser mapeadas, tendrían una dirección incorrecta dentro del programa de Unity. Los sistemas de mano derecha rotan sobre un eje en la dirección contraria a las manecillas del reloj, mientras que los de mano izquierda rotan en el sentido opuesto. Para solucionar este problema y asegurarnos de que las rotaciones se efectúen en la dirección correcta, partiendo de la ecuación 2.6, se cambió el valor de θ a $-\theta$. Por lo tanto, el mapeo final del quaternion de un sensor

q_0 a un quaternion dentro Unity q'_0 está dado por la siguiente ecuación 3.1.

$$\begin{aligned} q_0 &= (q_w, q_x, q_y, q_z) \\ q'_0 &= (q_w, -q_x, -q_z, -q_y) \end{aligned} \tag{3.1}$$

Con esto queda definido el cambio de sistema de coordenadas de un sensor de inercia al sistema de coordenadas de Unity. De aquí en adelante, nos referiremos a un quaternion crudo de un sensor como q y a un quaternion mapeado a Unity como q' .

3.5 Calibración

3.5.1 Compensación Inicial

Una vez resuelta la diferencia entre sistemas de coordenadas se necesita encontrar una forma en que las orientaciones de los segmentos corporales del avatar vayan de acuerdo a las del usuario. Para lograr esto, cuando inicia el programa hacemos que el usuario tome una postura inicial igual a de la figura 3.4. En este punto en el tiempo se sabe que el usuario y el avatar están exactamente en la misma posición. Por la ecuación 2.9, se deduce que una serie de rotaciones en forma de quaternion se pueden representar por una secuencia de multiplicaciones. Si definimos q'_0 como el quaternion de un sensor en esta pose inicial y q'_t como cualquier otra orientación válida en otro punto en el tiempo, entonces se tiene otra rotación q'_1 que hace que q'_0 rote hacia q'_t . Escrito de manera formal se tiene la ecuación 3.2.

$$q'_t = q'_1 q'_0 \tag{3.2}$$

Esta rotación q'_1 es la que queremos utilizar para la orientación de los segmentos del avatar. Las rotaciones por las que pasó el sensor para llegar a la posición inicial no son importantes. Como el usuario y el avatar se encuentran en la misma posición o pose al inicio del programa, solo se necesita rotar el segmento proporcionalmente a los movimientos del usuario después de haber iniciado el programa. Al inicio del programa se guarda en memoria el primer quaternion de cada sensor, q'_0 . Entonces,

en cualquier tiempo posterior t , se ocupa que el segmento del avatar rote por:

$$q_1' = q_t' q_0'^* \quad (3.3)$$

La ecuación anterior sale en base a las propiedades descritas en la sección [REF]. Al hacer esto estamos reajustando la orientación inicial de cada sensor para que sea relativa a la orientación de cuando inició el programa, en lugar de que sea a la orientación de cuando se encendió el sensor. Básicamente lo que realmente se está haciendo es que estamos ignorando las rotaciones que ocurrieron antes de que iniciara el programa, representadas por q_0' , de la rotación total obtenida más adelante, q_t' .

3.5.2 Compensación por Importación de Modelo 3D

Con lo anterior solucionado, ahora se necesita realizar un último ajuste para cada segmento del avatar. Al momento de importar el modelo 3D a Unity, los segmentos tienen un desplazamiento en su rotación. A pesar de que el avatar se encuentra en la pose correcta, a causa de como fue importado el modelo 3D en Unity, algunos de sus segmentos no se encuentran en una orientación de cero rotación. Esto ocasiona que al momento de realizar la asignación de rotación, presentada en la ecuación 3.3, el resultado sea incorrecto. Dicha diferencia en rotación es de 180 grados sobre el eje z , solo presente en algunos segmentos del avatar, como se puede apreciar en la figura 3.6. Queremos que esta rotación esté incluida en el cálculo para que estos segmentos inicien con la orientación apropiada. Esta rotación la llamaremos q_I y la podemos describir de la siguiente manera:

$$\begin{aligned} q_I &= (\cos \frac{\pi}{2}, 0, 0, \sin \frac{\pi}{2}) \\ q_I &= (0, 0, 0, 1) \end{aligned} \quad (3.4)$$

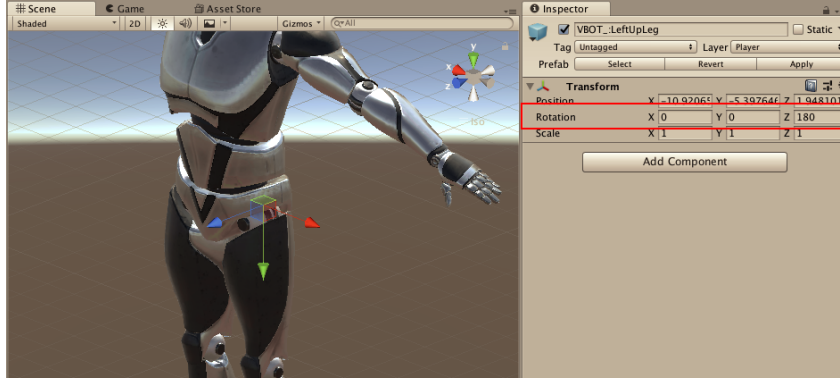


Figura 3.6: Rotación de 180 grados sobre eje z en segmentos del avatar a causa de su importación en Unity

Con esto ya tenemos el quaternion final que representa la rotación total que queremos que tengan los segmentos del avatar, la cual está dada por:

$$q_1' = q_t' q_0'^* q_I \quad (3.5)$$

Este método de obtener la orientación correspondiente tiene sus beneficios pero también tiene algunas debilidades. No se necesita ser meticuloso con la colocación de los sensores sobre el cuerpo del usuario para tener una orientación exacta con respecto al segmento del cuerpo. Estas posibles diferencias en orientación, cuando se colocan los sensores, son eliminados del cálculo ya que todas las rotaciones son relativas a la posición que toma el usuario al inicio del programa. El factor principal que determina la precisión en la captura de movimiento es qué tanto puede el usuario igualar la pose del avatar al inicio del programa. Es importante mencionar que, una vez iniciado el programa, evitar que los sensores se muevan o se resbalen de su lugar ya que esto causaría una discrepancia en la orientación de los segmentos del avatar con respecto a los del usuario.

3.6 Resumen del proceso

En las secciones anteriores de este capítulo se explicó a detalle las matemáticas detrás del proceso de mapeo, rastreo y captura. Sin embargo, para tener un mejor entendimiento de como funciona este sistema y sobre la experiencia del usuario, es importante tener un conocimiento general sobre las etapas generales por las que pasa

el sistema. Una vez que el usuario se ha colocado los sensores sobre su cuerpo, como se muestra en la figura 3.3, está listo para dar inicio del programa. Cuando el sistema arranca se le pide al usuario que se coloque en la posición inicial, como se ve en la figura 3.4. Tras finalizar una cuenta regresiva de cinco segundos, el quaternion actual de cada sensor es registrado. Estos quaternions son utilizados para la compensación inicial, q_0' , de cada sensor. Con esto nos aseguramos de eliminar todas las rotaciones anteriores para que el avatar pueda igualar de forma apropiada los movimientos del usuario. En caso de ser necesario, se debe realizar una segunda compensación por aquellos segmentos del avatar cuya orientación es distinta a cero por la forma en que fue importado a Unity.

Una vez registradas la compensación inicial y de importación, el programa empieza a mapear los movimientos del usuario a los del avatar. En cada actualización se lee un nuevo quaternion de cada sensor. Este valor es usado como q_t , visto en la ecuación 3.5, para el nuevo valor para q_1 , el cual se vuelve q_1' tras el cambio de sistema de coordenadas. Es importante hacer énfasis en que tras cada actualización estamos rotando cada segmento del avatar desde su posición inicial, cero rotación, en lugar de utilizar el valor de su última rotación conocida.

Esta secuencia permite que cada uno de los segmentos del avatar igualen la orientación de los del usuario en tiempo real. Sin embargo, este proceso por sí mismo solo iguala la orientación de los huesos o segmentos, las cuales se mantienen como instancias independientes rotando sobre sus puntos de origen. Esto provoca que los segmentos del modelo 3D se desconecten, por ejemplo, si el usuario sube su brazo éste subiría sin el antebrazo. Para esto utilizamos el sistema jerárquico de Unity que nos ayuda a formar cadenas cinemáticas con cuerpos rígidos, figura 3.7, lo cual soluciona el problema del posicionamiento de los segmentos. Esto se hace con el fin de asegurarnos de que el esqueleto no se separe y cada segmento se mantenga en su lugar.

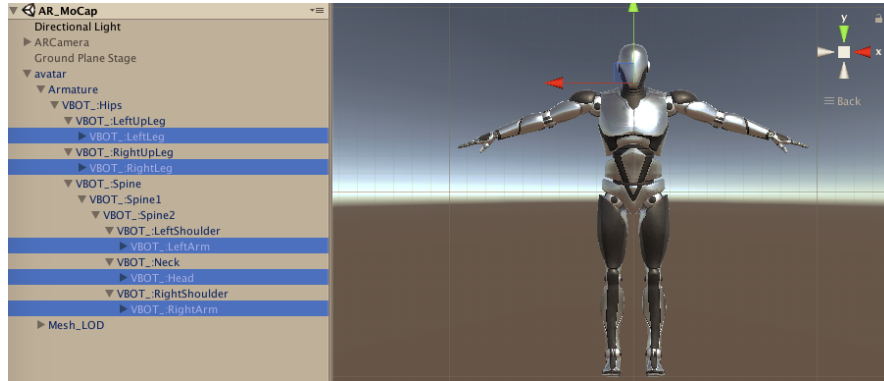


Figura 3.7: El sistema de jerarquía de objetos de Unity se encarga de mantener una cadena cinemática de cuerpo rígido

3.7 Aplicación de Realidad Aumentada

Una vez obtenido correctamente la configuración de sensores es momento de pasar al desarrollo de la aplicación de realidad aumentada utilizando de las Mediante esta aplicación el usuario puede observar un avatar de sí mismo replicando sus movimientos en tiempo real. El periférico para visualización que se utilizó fue teléfono Samsung Galaxy S6 Edge, el cual tiene un procesador Quad-core 2.1 GHz Cortex-A57 y un procesador gráfico Mali-T760 MP8.



Figura 3.8: Periféricos de realidad aumentada

En la figura 3.9 se muestra el personaje o avatar que representa al usuario siendo aumentado sobre el mundo real, posicionado en espera de iniciar sesión de captura.

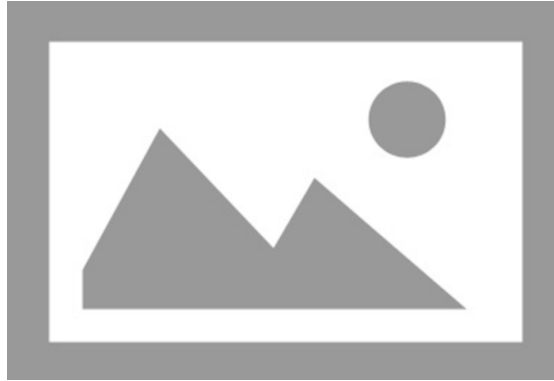


Figura 3.9: Foto-captura del avatar siendo aumentado sobre el mundo real

Por medio de una interfaz gráfica el usuario puede elegir entrar en modo espejo o iniciar una sesión de captura. Una vez iniciada la sesión, el usuario puede detenerla mediante la misma interfaz gráfica. En la figura 3.10 se muestra la interfaz que utiliza el usuario para seleccionar dichas funcionalidades.



Figura 3.10: Interfaz gráfica de usuario en realidad aumentada

A pesar de que no se realizaron pruebas cuantitativas para conocer con exactitud la precisión de rastreo

3.8 Limitaciones del sistema

Se tuvieron algunos inconvenientes al incorporar el dispositivo móvil con la aplicación de realidad aumentada. El tamaño original del modelo 3D del avatar tuvo que ser escalado varias veces para prevenir que se mirara muy grande o muy pequeño.

Sumado a esto, el posicionamiento del avatar como objeto aumentado en el mundo real también presentó varios problemas de inestabilidad. Como se explicó en la sección 3.7, se optó por utilizar Vuforia como base de nuestra aplicación de realidad aumentada, debido a su alto rango de compatibilidad tanto con dispositivos Android como IOS. Sin embargo, su algoritmo de detección de planos depende en gran medida del hardware del dispositivo.

A pesar de que el proceso de interpolación lineal esférica de quaternion entre valores actuales y posteriores ayudó a obtener mejores animaciones, reduciendo el temblor o inestabilidad en el mapeo de movimientos, éste no pudo eliminarlo por completo.

También se presentaron inconvenientes con la frecuencia de actualización de la información proveniente de los sensores. Por momentos breves y esporádicos el avatar parecía quedarse inmóvil cuando el sistema esperaba una lectura de los sensores. Para mitigar este problema, la parte del código que maneja las lecturas de los sensores fue separada en varios hilos de procesador, uno por sensor.

Aún con los detalles mencionados anteriormente, el desempeño del sistema es aceptable y adecuado para su combinación con ambientes de realidad aumentada. Aunque aún existen áreas que todavía se pueden mejorar, el sistema actual es una base sólida y funcional sobre la cual se puede expandir en el futuro.

Capítulo 4

Caso de Estudio: Asistente de Entrenamiento Deportivo

4.1 Ciencia del Deporte

Los especialistas en el deporte frecuentemente utilizan algún tipo de sistema de captura de movimientos para observar la función del cuerpo desde múltiples perspectivas y extraer los ángulos y posición de las coyunturas en la extremidades del cuerpo. Los especialistas médicos buscan conocer como es que un atleta se lesiona, indagando en las complicaciones antes, durante y después de alguna lesión. Los que se dedican al campo de estudio del desempeño deportivo necesitan estos datos para estudiar como se mueve un atleta, con el objetivo de conocer aquello que lo hace sobresalir y tener éxito.

El uso más común de *Motion Capture* en esta área del deporte es el análisis del modo de caminar, conocido como *Gait Analysis* en inglés. Es verdad que otro tipo de movimientos son importantes en el deporte, pero casi todas las actividades físicas que son realizadas sobre un piso incluyen algún tipo de movimiento que involucra correr, trotar o caminar.

Para los atletas o deportistas en general es muy valioso tener la oportunidad de observarse a sí mismos en alguna pantalla desde diferentes perspectivas y velocidades. La mayoría de los atletas profesionales de alto rendimiento en deportes olímpicos experimentan alguna forma de *Motion Capture*.

Capítulo 5

Evaluación y Resultados

5.1 Rendimiento

A pesar de los inconvenientes discutidos en la sección 3.8, el sistema puede igualar los movimientos del usuario con un alto nivel de precisión. La latencia en el rastreo solo se vuelve perceptible cuando el usuario realiza movimientos bruscos o rápidos. Esto se debe en parte al proceso de suavizado de rotaciones mediante interpolación que se implementó para reducir la inestabilidad. Esta es una compensación aceptable ya que solo es un pequeño retardo que realmente no es suficiente para ser un distraer o dañar la experiencia del usuario de ninguna forma. Como trabajo futuro se podría realizar algún tipo de prueba que ayude a determinar un balance entre suavizar o permitir cierta inestabilidad.

5.2 Deriva Acumulada

El mayor problema que presenta nuestro sistema actual es el *sensor drift* o deriva acumulada. Esto se pudo mitigar gracias al proceso de calibración inicial. No obstante, los sensores siguen acumulando este error mientras transcurre el tiempo. A pesar de que este no es problema que impida el funcionamiento del sistema, éste se vuelve notable para el usuario después de algunos minutos.

5.3 Escalabilidad del Sistema

Se decidió utilizar un total de diez sensores para nuestro sistema, permitiendonos cubrir las principales partes del cuerpo. Sin embargo, el diseño de este sistema es altamente escalable y puede ser ajustado para adoptar un mayor número de sensores. Por ejemplo, si se añaden más sensores en los pies y manos se obtendría una visualización mejor detallada. De igual forma, si la aplicación solo requiere de la captura de los brazos y no el cuerpo completo, se puede ajustar dado que la calibración seguiría funcionando de la misma forma.

Capítulo 6

Conclusiones

6.1 Aplicaciones

Dado el crecimiento acelerado del mercado de la tecnología *wearable* y el auge de las tecnologías de realidad virtual y realidad aumentada, este trabajo pudiera tener un uso práctico y una amplia variedad de contextos y aplicaciones.

Este tipo de sistema de captura de movimiento puede resultar bastante útil en aplicaciones como fisioterapia. Se podría otorgar retroalimentación precisa a pacientes sobre qué tan bien realizan sus estiramientos y ejercicios sin la necesidad de una supervisión constante de un terapeuta.

6.2 Trabajo futuro

Como trabajo futuro se plantea la integración de un módulo de reconocimiento de gestos. Esto con el fin de poder clasificar distintos tipos de movimientos y así realizar un mejor análisis sobre ellos. Como se vio en el capítulo 4, esto podría beneficiar en gran medida a los deportistas en el progreso de su entrenamiento, dado que podrían recibir retroalimentación en tiempo real sobre su desempeño y así lograr una mejora en poco tiempo.

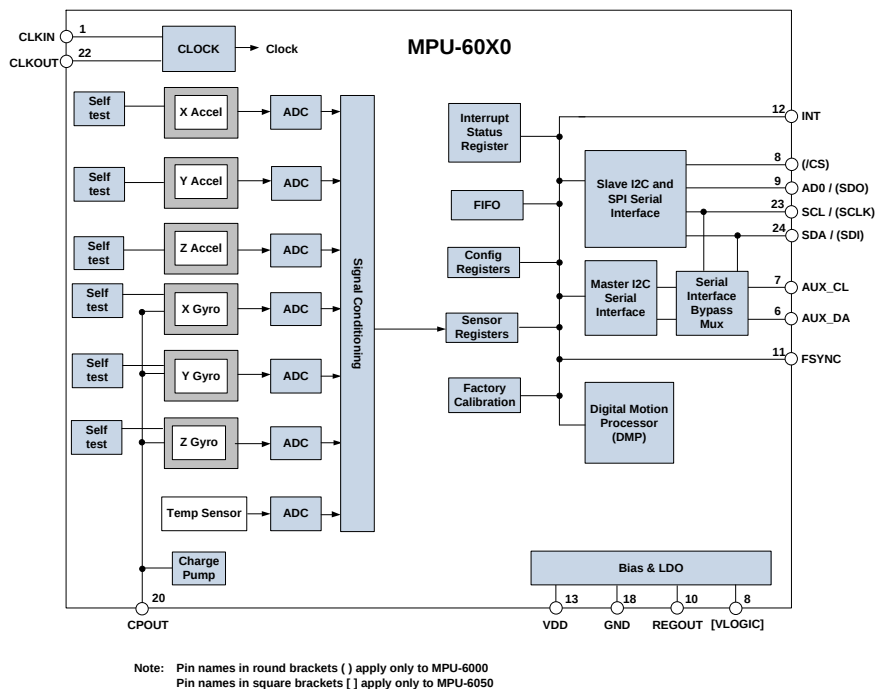
Anexos A

Anexo: Hojas de Especificación Técnica

A.1 MPU6050

	MPU-6000/MPU-6050 Product Specification	Document Number: PS-MPU-6000A-00 Revision: 3.4 Release Date: 08/19/2013
---	--	---

7.5 Block Diagram



7.6 Overview

The MPU-60X0 is comprised of the following key blocks and functions:

- Three-axis MEMS rate gyroscope sensor with 16-bit ADCs and signal conditioning
- Three-axis MEMS accelerometer sensor with 16-bit ADCs and signal conditioning
- Digital Motion Processor (DMP) engine
- Primary I²C and SPI (MPU-6000 only) serial communications interfaces
- Auxiliary I²C serial interface for 3rd party magnetometer & other sensors
- Clocking
- Sensor Data Registers
- FIFO
- Interrupts
- Digital-Output Temperature Sensor
- Gyroscope & Accelerometer Self-test
- Bias and LDO
- Charge Pump

A.2 ESP8266



1. General Overview

1.1. Introduction

Espressif Systems' Smart Connectivity Platform (ESCP) is a set of high performance, high integration wireless SOCs, designed for space and power constrained mobile platform designers. It provides unsurpassed ability to embed WiFi capabilities within other systems, or to function as a standalone application, with the lowest cost, and minimal space requirement.

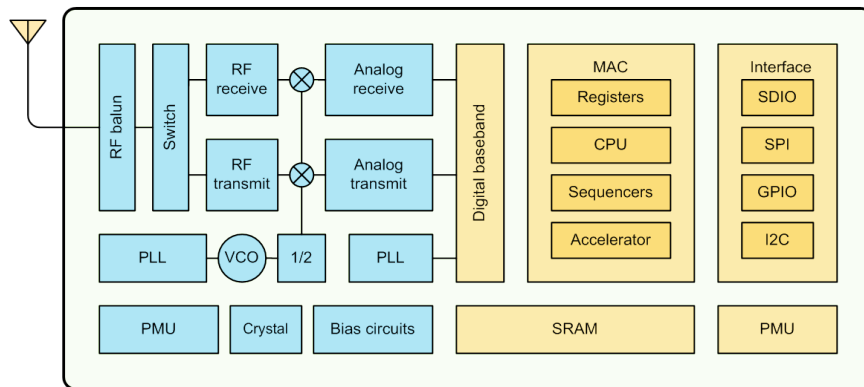


Figure 1 ESP8266EX Block Diagram

ESP8266EX offers a complete and self-contained WiFi networking solution; it can be used to host the application or to offload WiFi networking functions from another application processor.

When ESP8266EX hosts the application, it boots up directly from an external flash. It has integrated cache to improve the performance of the system in such applications.

Alternately, serving as a WiFi adapter, wireless internet access can be added to any micro controller-based design with simple connectivity (SPI/SDIO or I2C/UART interface).

ESP8266EX is among the most integrated WiFi chip in the industry; it integrates the antenna switches, RF balun, power amplifier, low noise receive amplifier, filters, power management modules, it requires minimal external circuitry, and the entire solution, including front-end module, is designed to occupy minimal PCB area.

ESP8266EX also integrates an enhanced version of Tensilica's L106 Diamond series 32-bit processor, with on-chip SRAM, besides the WiFi functionalities. ESP8266EX is often integrated with external sensors and other application specific devices through its GPIOs; sample codes for such applications are provided in the software development kit (SDK).

Bibliografía y Referencias

- [1] “Real-Time Human Motion Capture Driven by a Wireless Sensor Network”. In: *International Journal of Computer Games Technology* 2015 (2015), pp. 1–14. URL: <https://www.hindawi.com/journals/ijcgt/2015/695874/>.
- [2] *Biomechanics and Sport*. 2018. URL: <https://www.vicon.com/motion-capture/biomechanics-and-sport> (visited on 04/24/2018).
- [3] *Xsens*. 2018. URL: <http://www.xsens.com/products/xsens-mvn/> (visited on 04/24/2018).
- [4] *PERCEPTION NEURON*. 2018. URL: <https://neuronmocap.com/> (visited on 04/24/2018).
- [5] *Wear Notch*. 2018. URL: <https://wearnotch.com/> (visited on 04/24/2018).
- [6] Marc Leman and Rolf Godøy Inge. “Why study musical gestures?” In: *Musical Gestures: Sound, Movement, and Meaning*. 2009, pp. 3–11.
- [7] Eduardo Filgueiras Damasceno, Alexandre Cardoso, and Edgard Afonso Lamounier. “An middleware for motion capture devices applied to virtual rehab”. In: *Proceedings - IEEE Virtual Reality*. 2013, pp. 171–172. ISBN: 9781467347952. DOI: 10.1109/VR.2013.6549417.
- [8] Maxine Kwan et al. “Investigation of high-speed badminton racket kinematics by motion capture”. In: *Sports Engineering* 13.2 (2011), pp. 57–63. ISSN: 13697072. DOI: 10.1007/s12283-010-0053-0.
- [9] Sian Barris and Chris Button. “A Review of Vision-Based Motion Analysis in Sport”. In: *Sports Medicine* 38.12 (2008), pp. 1025–1043. ISSN: 0112-1642. DOI: 10.2165/00007256-200838120-00006. URL: <http://link.springer.com/10.2165/00007256-200838120-00006>.

- [10] Matthew Field et al. “Motion capture in robotics review”. In: *2009 IEEE International Conference on Control and Automation, ICCA 2009*. 2009, pp. 1697–1702. ISBN: 9781424447060. DOI: 10.1109/ICCA.2009.5410185.
- [11] Sana Ullah et al. “A comprehensive survey of wireless body area networks on PHY, MAC, and network layers solutions”. In: *Journal of Medical Systems* 36.3 (2012), pp. 1065–1094. ISSN: 01485598. DOI: 10.1007/s10916-010-9571-3. arXiv: 1208.2351.
- [12] Victor Shnayder et al. “Sensor networks for Medical Care”. In: *Proceedings of Third International Conference on Embedded Networked Sensor Systems* June (2005), p. 314. ISSN: 1022-0038. DOI: 10.1145/1098918.1098979. URL: <http://portal.acm.org/citation.cfm?doid=1098918.1098979>.
- [13] Anthony D. Wood et al. “Context-aware wireless sensor networks for assisted living and residential monitoring”. In: *IEEE Network* 22.4 (2008), pp. 26–33. ISSN: 08908044. DOI: 10.1109/MNET.2008.4579768.
- [14] Chin Feng Lai et al. “Adaptive body posture analysis for elderly-falling detection with multisensors”. In: *IEEE Intelligent Systems*. Vol. 25. 2. 2010, pp. 20–30. ISBN: 1541-1672. DOI: 10.1109/MIS.2010.39.
- [15] Min Chen et al. *Body area networks: A survey*. 2011. DOI: 10.1007/s11036-010-0260-8.
- [16] Benny P L Lo, Surapa Thiemjarus, and Rachel King. “Body Sensor Network – a Wireless Sensor Platform for Pervasive Healthcare Monitoring”. In: *Architectural Design* 13.2-3 (2005), pp. 77–80. URL: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.94.9003%7B%5C%7Drep=rep1%7B%5C%7Dtype=pdf>.
- [17] Soh Masuko and Junichi Hoshino. “A fitness game reflecting heart rate”. In: *Proceedings of the 2006 ACM SIGCHI international conference on Advances in computer entertainment technology - ACE '06*. 2006, p. 53. ISBN: 1595933808. DOI: 10.1145/1178823.1178886. URL: <http://portal.acm.org/citation.cfm?doid=1178823.1178886>.
- [18] Grahan Baradoy, Larry Katz, and James Parker. “A Physiological Feedback Controlled Exercise Video Game”. In: *The Journal of the Canadian Game Studies Association* 6.9 (2012), pp. 113–120.

- [19] Kiel Gilleade, Alan Dix, and Jen Allanson. “Affective Videogames and Modes of Affective Gaming: Assist Me, Challenge Me, Emote Me”. In: *Proceeding of DIGRA 2005* (2005), pp. 1–7.
- [20] Lennart Erik Nacke et al. “Biofeedback game design”. In: *Proceedings of the 2011 annual conference on Human factors in computing systems - CHI '11*. 2011, p. 103. ISBN: 9781450302289. DOI: 10.1145/1978942.1978958. URL: <http://dl.acm.org/citation.cfm?doid=1978942.1978958>.
- [21] Anthony Whitehead et al. “Sensor networks as video game input devices”. In: *Proceedings of the 2007 conference on Future Play - Future Play '07* (2007), p. 38. DOI: 10.1145/1328202.1328210. URL: <http://portal.acm.org/citation.cfm?doid=1328202.1328210>.
- [22] Greg Welch and Eric Foxlin. “Motion tracking: No silver bullet, but a respectable arsenal”. In: *IEEE Computer Graphics and Applications* 22.6 (2002), pp. 24–38. ISSN: 02721716. DOI: 10.1109/MCG.2002.1046626.
- [23] MPU-6050 Six-Axis (Gyro + Accelerometer) MEMS MotionTrackingTM Devices. 2018. URL: <https://www.invensense.com/products/motion-tracking/6-axis/mpu-6050/> (visited on 04/24/2018).
- [24] Z Q Zhang et al. “Adaptive Information Fusion for Human Upper Limb Movement Estimation”. In: *Ieee Transactions on Systems Man and Cybernetics Part a-Systems and Humans* 42.5 (2012), pp. 1100–1108.
- [25] H. J. Luinge and Peter H. Veltink. “Measuring orientation of human body segments using miniature gyroscopes and accelerometers”. In: *Medical and Biological Engineering and Computing* 43.2 (2005), pp. 273–282.
- [26] “Design, Implementation, and Experimental Results of a Quaternion-Based Kalman Filter for Human Body Motion Tracking”. In: *Robotics, IEEE Transactions on* 22.6 (2006), pp. 1216–1227. URL: 10.1109/TR0.2006.886270.
- [27] Walter T. Higgins. “A Comparison of Complementary and Kalman Filtering”. In: *IEEE Transactions on Aerospace and Electronic Systems* AES-11.3 (1975), pp. 321–325.
- [28] Greg Welch and Gary Bishop. “An Introduction to the Kalman Filter”. In: *In Practice* 7.1 (2006), pp. 1–16. URL: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.79.6578%7B%5C%7Drep=rep1%7B%5C%7Dtype=pdf>.

- [29] William Rowan Hamilton. “Elements of Quaternions”. In: *Longsmans, Green & Co* (1866), p. 352.
- [30] R Mukundan. “Quaternions : From Classical Mechanics to Computer Graphics , and Beyond”. In: *Proceedings of the 7th Asian Technology Conference in Mathematics* (2002), pp. 97–106.
- [31] *Gimbal Lock*. 2018. URL: https://en.wikipedia.org/wiki/Gimbal_lock (visited on 04/24/2018).
- [32] *3 Reasons Augmented Reality is the Future of Supply Chain*. 2016. URL: <http://www.appcessories.co.uk/augmented-reality-is-the-future-of-supply-chain/> (visited on 04/24/2018).
- [33] Mark Billinghurst, Adrian Clark, and Gun Lee. “A Survey of Augmented Reality”. In: *Foundations and Trends® in Human-Computer Interaction* 8.2-3 (2015), pp. 73–272. ISSN: 1551-3955. DOI: 10.1561/11000000049. arXiv: arXiv:1011.1669v3. URL: <http://www.nowpublishers.com/article/Details/HCI-049>.
- [34] M Billinghurst. “The future of augmented reality in our everyday life”. In: *Proceedings of the International Display Workshops* 3 (2011), pp. 1975–1978. ISSN: 18832490. URL: <http://www.scopus.com/inward/record.url?eid=2-s2.0-84870665793%7B%5C%7DpartnerID=40%7B%5C%7Dmd5=>.
- [35] Gerhard Reitmayr and Tom W. Drummond. “Going out: Robust model-based tracking for outdoor augmented reality”. In: *Proceedings - ISMAR 2006: Fifth IEEE and ACM International Symposium on Mixed and Augmented Reality*. 2007, pp. 109–118. ISBN: 1424406501. DOI: 10.1109/ISMAR.2006.297801.
- [36] Georg Klein and David Murray. “Parallel tracking and mapping for small AR workspaces”. In: *2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality, ISMAR*. 2007. ISBN: 9781424417506. DOI: 10.1109/ISMAR.2007.4538852. arXiv: arXiv:1407.5736v1.
- [37] Vincent Lepetit and Pascal Fua. “Monocular Model-Based 3D Tracking of Rigid Objects: A Survey”. In: *Foundations and Trends® in Computer Graphics and Vision* 1.1 (2005), pp. 1–89. ISSN: 1572-2740. DOI: 10.1561/06000000001. URL: <http://www.nowpublishers.com/article/Details/CGV-001>.
- [38] *Microsoft Hololens*. 2016. URL: <https://www.microsoft.com/en-ca/hololens> (visited on 04/24/2018).

- [39] *AR SDK Feature Comparison*. 2018. URL: <https://thinkmobiles.com/blog/best-ar-sdk-review/> (visited on 04/24/2018).
- [40] Unity Technologies. *Unity Asset Store*. 2018. URL: <https://assetstore.unity.com/> (visited on 04/24/2017).
- [41] Unity Technologies. *Unity Asset Store*. 2018. URL: <https://unity3d.com/es> (visited on 04/24/2017).