

TESIS DOCTORAL

2018

INTEGRACIÓN DE CÁMARAS DE RANGO Y REDES INALÁMBRICAS PARA LA LOCALIZACIÓN DE PERSONAS EN ENTORNOS ESTRUCTURADOS

JAIME DUQUE DOMINGO

PROGRAMA DE DOCTORADO EN INGENIERÍA
DE SISTEMAS Y CONTROL

CARLOS CERRADA SOMOLINOS
ENRIQUE VALERO RODRÍGUEZ

Daría todo lo que sé, por la mitad de lo que ignoro.

Descartes

Agradecimientos

Han pasado cuatro años desde que decidí iniciar esta importante etapa de mi vida. Tengo que agradecer a Carlos y a Enrique el enorme esfuerzo que han realizado para ayudarme a llegar a este punto. Cientos de revisiones de los artículos de revista o de congreso que se han traducido en importantes contribuciones científicas. Pese a la distancia entre todos, Carlos en Madrid, Enrique en Edimburgo, y yo en Bruselas, conseguimos trabajar de una manera organizada, manteniendo diversas reuniones por videoconferencia, y coincidiendo en varios congresos. Durante estos años han pasado muchas cosas, algunas malas y otras buenas. Pero si hay que destacar una cosa entre todas, que me llena de ilusión, es el nacimiento de mi hija Valeria. También tengo que agradecer a mis padres, Asun y Víctor, y a mi mujer, Henar, el apoyo que me han ofrecido continuamente.

Resumen

Esta tesis doctoral presenta un sistema de posicionamiento en interiores para estimar la ubicación de personas que se mueven dentro de entornos complejos. La técnica desarrollada combina los sistemas de posicionamiento WiFi y los mapas de profundidad, generados por sensores RGB-D, ofreciendo resultados prometedores en ambientes habitados complejos, que consisten en varias habitaciones conectadas, donde la gente se mueve libremente. Se trata de un sistema no intrusivo en el que no se necesita información personal sobre los usuarios, y aunque las cámaras RGB-D se instalan en el área de detección, las personas solo deben llevar sus teléfonos móviles. En la tesis se detallan los métodos desarrollados para combinar las tecnologías antes mencionadas y los experimentos realizados. La tesis presenta los resultados de posicionamiento dentro de una habitación, con una cámara RGB-D. A continuación, el sistema se generaliza a entornos complejos con múltiples habitaciones y cámaras. Los resultados obtenidos muestran una mejora significativa en términos de precisión y rendimiento con respecto a las anteriores soluciones basadas en WiFi, así como una extensión en el rango de operación.

La tesis también muestra varias técnicas que, aplicadas al IPS desarrollado, ofrecen diferentes soluciones al problema de predicción de movimientos en interiores, y se estudia cuál ofrece mejores resultados. La predicción de movimientos se lleva a cabo mediante una estrategia proactiva, obteniendo la próxima posición estimada de la persona. Esta estimación proporciona una información más rica de la localización y del contexto, que es útil en computación ubicua. Por ejemplo, el consumo de energía se puede optimizar si la iluminación o los dispositivos electrónicos se encienden/apagan por medio de la predicción de la trayectoria del usuario.

En las aplicaciones del método, se presenta una nueva ontología con el objetivo

de conectar un IPS a Europeana, la plataforma digital de la Unión Europea para el patrimonio cultural. El objetivo principal de este sistema es ofrecer información sobre los objetos del patrimonio cultural (CHO), a los usuarios que se mueven por museos, cuando se acercan a ciertas obras de arte. El potencial del método desarrollado se muestra utilizando datos obtenidos de los Museos Reales de Bellas Artes de Bélgica, una de las galerías de arte más importantes de Europa, con más de 6000 piezas maestras enumeradas en Europeana. También, se han realizado algunos experimentos en el Museo de Arte Antiguo, uno de los museos constitutivos de los Museos Reales que se dedica a los pintores europeos desde el siglo XV al XVIII.

Abstract

This doctoral thesis presents an *Indoor Positioning System* (IPS) to estimate the location of people navigating in complex indoor environments. The developed technique combines *WiFi Positioning Systems* and *depth maps*, delivering promising results in complex inhabited environments, consisting of various connected rooms, where people are freely moving. This is a non-intrusive system in which personal information about people is not needed and, although RGB-D cameras are installed in the sensing area, users are only required to carry their *smartphones*. The methods developed to combine the above-mentioned technologies and the experiments performed to test the system are detailed. The thesis presents the results for positioning inside one room with one RGB-D camera. Then, the system is generalized to complex environments with multiple rooms and RGB-D cameras. The obtained results show a significant improvement in terms of accuracy and performance with respect to previous WiFi-based solutions as well as an extension in the range of operation.

The thesis also shows how several techniques, applied for the developed IPS, offer different solutions to the indoor prediction problem, and it discusses about which of them gives better results. The prediction of movements is carried out by means of a proactive strategy, delivering the next estimated position of the person. This estimation provides a richer location and context information, which is useful for ubiquitous computing purposes. For example, energy consumption can be optimized if lighting or electronic devices are turned on/off by means of the user trajectory prediction.

In the applications of the method, it is presented a novel ontology aiming to connect an IPS to Europeana, the European Union digital platform for cultural heritage. The main purpose of this system is to deliver information about Cultural Heritage

Objects (CHO), to users navigating in museums, when they approach to certain pieces of art. The potential of the developed method is shown using data obtained from the Royal Museums of Fine Arts of Belgium, one of the most important European art galleries, with more than six thousand master pieces listed in Europeana. Some experiments have been also carried out in the Old Masters Museum, one of the constituent museums of the Royal Museums that is dedicated to European painters from the 15th to the 18th centuries.

Índice general

1. Introducción	1
1.1. Objetivos	2
1.2. Metodología	3
1.3. Marco de realización	4
1.4. Estructura de la tesis	4
2. Estado del arte	7
2.1. Sistemas de posicionamiento en interiores	7
2.1.1. Posicionamiento mediante satélites	10
2.1.2. Posicionamiento mediante torres de telefonía móvil (GSM)	11
2.1.3. Sistemas de posicionamiento WiFi	11
2.1.4. Sistemas de posicionamiento RFID	13
2.1.5. Posicionamiento mediante visión artificial	13
2.1.6. Sistemas de posicionamiento combinado	16
2.2. Predicción de movimientos	18
2.3. Sistemas semánticos de posicionamiento	19
2.3.1. Ingeniería del conocimiento	21
3. Tecnologías	23
3.1. Sensores RGB-D	23
3.1.1. Mapas de profundidad	26
3.1.2. Kinect para Windows	27
3.2. Teléfonos móviles basados en Android	32
3.3. Puntos de acceso WiFi	33

4. Análisis de métodos utilizados	35
4.1. Sistema de posicionamiento inalámbrico (WPS) asistido por RGB-D	35
4.1.1. Descripción del funcionamiento del sistema	36
4.2. Posicionamiento WPS mediante RGB-D en entornos complejos	44
4.2.1. <i>Fase de Aprendizaje</i>	45
4.2.2. <i>Fase de Ejecución</i>	49
4.2.3. <i>Transformación de coordenadas bajo un único UCS</i>	56
4.3. Predicción de movimientos	57
4.3.1. Red neuronal para problema de posicionamiento	61
5. Experimentación	63
5.1. Análisis de datos de posicionamiento	63
5.1.1. Datos RSSI de un punto de acceso próximo	66
5.1.2. Datos RSSI de un punto de acceso lejano	68
5.1.3. Análisis de datos Kinect	70
5.2. Experimentación de posicionamiento con WPS	73
5.2.1. Determinación de la habitación donde se encuentra la persona	74
5.2.2. Determinación de la celda donde se encuentra una persona	77
5.3. Experimentos de posicionamiento WPS con RGB-D en una sala	82
5.4. Experimentación de posicionamiento WPS con RGB-D en entornos complejos	85
5.4.1. Despliegue simulado de sensores y trayectorias	89
5.4.2. Ubicación de los sensores en el UCS	91
5.4.3. Cálculo de la distancia euclídea sincronizada	95
5.4.4. Resultados	96
5.5. Experimentos en predicción de movimientos	98
6. Integración del IPS con la plataforma Europea	103
6.1. Europea	104
6.1.1. Consultas contra Europea	105
6.2. Nueva ontología para integrar el IPS con Europea	106
6.2.1. Ontología propuesta	107
6.2.2. Integración de la ontología con Europea	110

6.2.3. Integración de IPSs con la ontología	111
6.3. Desarrollo de una aplicación móvil para integrar un IPS con la plata- forma Europea	114
6.3.1. Obtener el camino óptimo entre dos salas	115
6.4. Experimentación	116
7. Conclusiones	119
7.1. Aplicaciones del método propuesto	121
7.2. Publicaciones	121
7.3. Trabajos futuros	123
Glosario	125
Bibliografía	128

Índice de figuras

2.1. Tecnologías utilizadas en posicionamiento en interiores	9
3.1. Partes de Kinect 360	24
3.2. Esquema de funcionamiento del emisor/cámara IR de Kinect 360 . . .	24
3.3. Triangulación para obtener mapa de profundidad con Kinect 360 . . .	25
3.4. Sensores RGB-D	26
3.5. Imagen RGB y mapa de profundidad. Por Dominic Alves (2010) . . .	27
3.6. Sensores Microsoft Kinect.	29
3.7. Nube de puntos, correspondiente a un entorno habitado, generada a partir de los datos obtenidos por un dispositivo Kinect 2	31
3.8. Articulaciones devueltas para cada esqueleto por el sensor Kinect v2 .	32
3.9. Router Linksys WIFI 54 Mbps WRT54GL	33
4.1. Componentes del sistema	36
4.2. Diagrama del sistema durante los estados de Aprendizaje/Ejecución donde los valores 1 y 0 representan estado de ejecución satisfactorio/insatisfactorio respectivamente. “Guardar datos” es el tercer estado durante la fase de Aprendizaje mientras que “Proceso de posicionamiento” es el tercer paso durante la fase de Ejecución	37
4.3. Obteniendo un vector RSSI para una celda (ejemplo con 4 APs) . . .	38
4.4. Esquema de dos personas en la habitación	42
4.5. Ejemplo de trayectorias de $n = 2$ usuarios en $m = 4$ <i>time stamps</i> , incluyendo las distancias euclídeas entre la combinación de $P_i(t)$ y $P'_j(t)$	44
4.6. Escenario del sistema	45

4.7. Diagrama de actividades durante la Fase de Aprendizaje	46
4.8. Diagrama de actividades durante la Fase de Ejecución	49
4.9. Esquema de las llamadas a <i>web services</i>	50
4.10. Celdas con trayectorias WPS y RGB-D durante la Fase de Ejecución	52
4.11. Ejemplo de matriz S para un distinto número de usuarios WPS y RGB-D.	55
4.12. Los ejes X_{RGB-D} , Y_{RGB-D} , y Z_{RGB-D} respecto a la dirección del sensor RGB-D	57
4.13. Red neuronal multicapa.	59
4.14. Red neuronal para la predicción de posicionamiento	62
5.1. Esquema de aplicación desarrollada	64
5.2. Estimación de la densidad de probabilidad RSSI (dBm) sobre NUBE en una localización concreta	66
5.3. Estimación simulada de la densidad de probabilidad RSSI en una lo- calización concreta para un <i>router</i> próximo	67
5.4. Estimación de la densidad RSSI (dBm) sobre MOVISTAR_FA71 en una localización concreta	68
5.5. Estimación simulada de la densidad de probabilidad RSSI en una lo- calización concreta para un <i>router</i> lejano	69
5.6. Diagrama de cajas de la distribución de los datos	70
5.7. Sistema de coordenadas, tomando el sensor Kinect como origen . . .	71
5.8. Estimación de la densidad de probabilidad de los 3 ejes Kinect	72
5.9. Estimación simulada de la densidad de probabilidad de los 3 ejes Ki- nect en una posición concreta	73
5.10. Esquema de entorno para detección de habitación	74
5.11. Porcentaje de éxito en la detección de la habitación donde se encuentra el usuario	76
5.12. División del suelo de la habitación en celdas virtuales	77
5.13. Esquema de aplicación desarrollada para determinación de la celda donde se encontraba una persona	78
5.14. Esquema de la habitación donde se han realizado los experimentos . .	79
5.15. Diferencia entre posicionamiento WPS y RGB-D	81

5.16. Sensor Kinect montado en una plataforma de madera	82
5.17. Trayectorias en la habitación donde se han realizado los experimentos	83
5.18. Algunas de las trayectorias seguidas por los usuarios	84
5.19. Plano de la oficina donde se han realizado los experimentos	86
5.20. Diagrama de secuencia del sistema	87
5.21. Captura de la aplicación desarrollada para dispositivos Android . . .	88
5.22. Algunas de las trayectorias seguidas por los usuarios	88
5.23. Trayectorias generadas durante la simulación de los experimentos . .	90
5.24. Ubicación de los sensores y rango de trabajo dentro de una habitación	92
5.25. Transformación mediante matriz de rotación y traslación de las coor- denadas al UCS	93
5.26. Resultados de la predicción de trayectorias de los esqueletos	100
6.1. Ontología implementada	107
6.2. Esquema de aplicación desarrollada	115
6.3. Obteniendo el camino óptimo entre dos salas en los Museos Reales . .	116
6.4. Experimentos desarrollados usando <i>Google Location Services API</i> . . .	117

Índice de tablas

4.1. Ejemplo de Tabla RSSI	39
4.2. Ejemplo de coordenadas de esqueleto (en metros)	39
4.3. Ejemplo de coordenadas del cuello de un usuario obtenidas por un sensor RGB-D denominado E para diferentes <i>time stamps</i> (en metros)	46
4.4. Ejemplo de tabla RSSI por <i>Time Stamp</i>	47
4.5. Ejemplo de valores para el <i>fingerprint map</i> reducido en la celda (3, 4)	48
5.1. Mediciones de registros RSSI	65
5.2. Interpretación de los valores RSSI	65
5.3. Interpretación de los valores RSSI	70
5.4. Ejemplo de mediciones de registros RSSI	79
5.5. Ejemplo de mediciones de registros Kinect (en metros)	79
5.6. Centroides RSSI para una celda	80
5.7. Posiciones mediante <i>fingerprinting</i>	81
5.8. Resultados de experimento WPS con RGB-D	84
5.9. Composición de las distintas trayectorias	90
5.10. Tabla de parámetros de emplazamiento de los sensores Kinect (en metros)	94
5.11. Porcentaje de éxito en función del número de usuarios y <i>time stamps</i>	97
5.12. Error de distancia medio (metros) para diferente número de usuarios y <i>time stamps</i>	97
5.13. Éxito al emparejar correctamente usuarios WPS con RGB-D para $m = 10$ <i>time stamps</i>	98
5.14. Éxitos en la predicción de movimientos en interiores.	101

6.1. Resultados para la consulta de Las Meninas 105

Capítulo 1

Introducción

Los sistemas de posicionamiento en interiores (IPS, del inglés *Indoor Positioning Systems*), son técnicas utilizadas para obtener la posición de personas u objetos dentro de edificios [1]. Entre estas técnicas, los sistemas de posicionamiento WiFi (WPS del inglés *WiFi Positioning Systems*), son aquellas técnicas que utilizan dispositivos portátiles, como teléfonos móviles, para localizar personas u objetos por medio de las medidas de nivel de señal recibidas de diferentes puntos de acceso (APs, del inglés *Access Points*), por ejemplo *routers* WiFi. Estas técnicas, no son particularmente propicias para suministrar datos de posicionamiento muy precisos ya que el error cometido en el posicionamiento es alto [2].

Los IPS, como se menciona en [1], pueden clasificarse como activos o pasivos, dependiendo de la interacción de los usuarios. En sistemas activos, actualmente más desarrollados, las personas tienen dispositivos electrónicos que envían información a un sistema de posicionamiento para estimar su ubicación en el entorno.

En el campo de la detección de personas y objetos, se han utilizado cada vez más tecnologías basadas en visión artificial. De hecho, el reconocimiento de objetos puede ser considerado esencial en el ámbito de la visión artificial, y un importante número de autores han desarrollado métodos y aplicaciones para la detección y el posicionamiento de personas. Dentro de los métodos más recientes y, por tanto, menos abundantes están aquellos que implican el uso de tecnologías modernas como los sensores RGB-D, que proporcionan información 3D en forma de mapas de profundidad de escenas (*depth maps*). Los sensores RGB-D se han desarrollado rápidamente

en los últimos años, ofreciendo resultados prometedores para la identificación de las personas [3], el posicionamiento [4–6] y el reconocimiento de los gestos [7].

En esta tesis se presenta un nuevo enfoque de IPS basado en la combinación de estas dos tecnologías: WPS y *depth maps*, de manera activa. Por la combinación activa se indica que el método desarrollado combina y explota coordinadamente ambos tipos de información sensorial: la intensidad de señales inalámbricas medidas y los mapas de profundidad.

Este enfoque es particularmente ventajoso cuando varios usuarios están simultáneamente en una habitación. En este caso, el sistema es capaz de detectar a cada usuario con la ayuda de las coordenadas de las personas ubicadas en un *depth map*. El WPS aproxima la posición de los usuarios, pero cuando están realmente cerca, el método propuesto es capaz de ofrecer una ubicación más precisa. Esto se consigue con la ayuda de las trayectorias de los usuarios, considerando simultáneamente ambas trayectorias: las WPS y las de personas en los mapas de profundidad.

Los usuarios solo requieren el uso de teléfonos móviles inteligentes, o *smartphones*, además de la instalación de sensores RGB-D en el área de detección. Además, el sistema no es intrusivo, no siendo necesario conocer la identidad de las personas. La tesis expone el método desarrollado para reunir y explotar ambos tipos de información sensorial en un escenario complejo.

Los resultados obtenidos muestran una mejora significativa en términos de posicionamiento con respecto a sistemas comunes basados en WiFi. Las cámaras RGB-D permiten mejorar la precisión del IPS tradicional de una manera fácil y económica.

Adicionalmente, este trabajo también presenta una nueva ontología para conectar el sistema de posicionamiento desarrollado con Europeana, la plataforma digital de la Unión Europea para el patrimonio cultural.

1.1. Objetivos

El objetivo principal es la mejora del posicionamiento en interiores, a través de un método económico, que no vulnere la privacidad y sencillo de instalar. Este objetivo puede dividirse en los siguientes objetivos específicos:

1. Identificar las líneas de investigación emergentes en el campo de posicionamiento

to de personas en interiores.

2. Lograr un conocimiento de diferentes sensores RGB-D y estudiar y evaluar el sistema de posicionamiento WPS.
3. Desarrollar un sistema de posicionamiento que ofrezca resultados, mejorando el posicionamiento tradicional WPS, gracias al uso de la visión artificial.
4. Conseguir un sistema económico, que no vulnere la privacidad y sencillo de instalar: Los medios utilizados consisten en varias cámaras RGB-D Kinect v2, diversos teléfonos móviles y *routers*. Las cámaras RGB-D se instalan con facilidad y el sistema no guarda información sobre los usuarios más allá de la necesaria para obtener su posicionamiento.
5. Experimentación del sistema, simulando el completo despliegue de cámaras RGB-D en entornos complejos.
6. Desarrollar un sistema ontológico para conectar el IPS con la base de conocimiento Europea y realizar experimentos en un museo.

1.2. Metodología

Para la consecución de los objetivos indicados, se ha establecido la siguiente metodología de trabajo:

1. Realización de una documentación bibliográfica, estudiando los distintos ámbitos de actuación, e identificando los logros alcanzados por diferentes autores en los últimos años. Para ello, se han recopilado artículos y otros documentos especializados en las áreas en las que se ha desarrollado la tesis.
2. Estudio de la tecnología a emplear y de los entornos de trabajo. Se ha estudiado inicialmente la generación de mapas de profundidad y obtención de los esqueletos de las personas mediante el sensor Kinect, para después continuar con el estudio de los sistemas de posicionamiento de personas en interiores, y su integración con Kinect.

3. Asistencia a eventos relacionados con la temática: seminarios, workshops, congresos nacionales e internacionales y realización de publicaciones parciales de los distintos trabajos, tanto en estos eventos como en revistas especializadas.
4. Desarrollo del contenido de tesis. Se han realizado reuniones periódicas con los directores de tesis, donde se ha comprobado la evolución del trabajo. El desarrollo ha avanzado, desde el caso más simple donde se realizaba el posicionamiento de unas pocas personas en una habitación, hasta el caso más complejo donde se posicionaban a varias personas en un entorno complejo formado por varias habitaciones. Se han realizado distintas tomas de datos para cada experimento realizado, y en algunos casos donde se requería un número grande de sensores, se ha simulado el completo despliegue de cámaras RGB-D.

1.3. Marco de realización

Los diferentes trabajos, vinculados al desarrollo de esta tesis doctoral, han sido llevados a cabo en el Departamento de Ingeniería del Software y Sistemas Informáticos de la Escuela Técnica Superior de Ingeniería Informática de la UNED.

Este trabajo ha sido desarrollado gracias a la ayuda del proyecto de investigación DPI2013-44776-R y DPI2016-77677-P del MICINN. También pertenece a las actividades llevadas a cabo en el marco de la red de investigación CAM RoboCity2030 S2013/MIT-2748, de la Comunidad de Madrid.

1.4. Estructura de la tesis

La tesis se ha estructurado en siete capítulos, siendo el primero de ellos esta introducción, más un apartado final en el que se muestran los trabajos referenciados.

En el Capítulo 2 se hace referencia a los trabajos realizados en los ámbitos de actuación de la tesis. Se presentan trabajos relacionados con los sistemas de posicionamiento en interiores actualmente existentes, prestando especial atención a los sistemas basados en WPS, sensores RGB-D y utilizando ambas tecnologías de manera conjunta.

En el Capítulo 3 se muestran las distintas tecnologías utilizadas durante el desarrollo de la tesis.

El Capítulo 4 presenta el sistema de posicionamiento desarrollado. Dentro de este capítulo se muestra la evolución del sistema hasta llegar a la versión definitiva para posicionamiento en entornos complejos. Finalmente, el capítulo presenta una extensión del método orientado a predicción de movimientos.

El Capítulo 5 muestra los distintos experimentos que se han llevado a cabo con el sistema de posicionamiento.

El Capítulo 6 presenta una aplicación concreta vinculada al sistema de posicionamiento que lo integra con Europeana, base de conocimiento de obras de arte europeas.

Finalmente, el Capítulo 7 realiza una discusión sobre los resultados obtenidos, muestra las publicaciones realizadas y presenta futuras investigaciones.

Capítulo 2

Estado del arte

En este capítulo se aborda el estado del arte de las distintas tecnologías estudiadas y utilizadas en esta tesis doctoral. Se presentan: el estudio de los sistemas de posicionamiento en interiores (Sección 2.1), la predicción de movimientos (Sección 2.2) y finalmente, los sistemas semánticos de posicionamiento (Sección 2.3).

2.1. Sistemas de posicionamiento en interiores

Como otros sistemas de navegación, los IPS son empleados para estimar la posición de una persona o un objeto en entornos interiores. Una primera taxonomía fue presentada a principios de los años 2000 por Hightower y Borriello [8]. En ella, se establece una clasificación de sistemas de localización para computación ubicua considerando factores como: precisión, escala, coste, limitaciones, posición (física o simbólica, absoluta o relativa), la necesidad de servidores externos, y finalmente la necesidad de reconocer objetos.

Una segunda clasificación se presenta por Koyuncu y Yang en [9], donde los IPS se organizan de acuerdo a la tecnología empleada para su desarrollo. Siguiendo este planteamiento, los autores sugieren cuatro grupos diferentes:

- IPSs basados en el cálculo de distancias mediante infrarrojos o ultrasonidos, que normalmente requieren altos costos de instalación.
- Sistemas basados en RSSI mediante tecnología WLAN o RFID.

- Sistemas basados en visión por ordenador o basados en la combinación de tecnologías como RFID o WLAN RSSI.
- Tecnologías inerciales, como giroscopios o acelerómetros.

En lo relativo a la localización de personas, estos sistemas de localización también se pueden clasificar en cuatro grupos diferentes, tal como indica Fallah et al. en [10]:

- *Dead-reckoning*, donde la posición de las personas se obtiene a partir de una posición previamente estimada o conocida mediante el uso de sensores tales como: acelerómetros, magnetómetros, brújulas y giroscopios, o utilizando el patrón de marcha de una persona.
- *Direct sensing*, que determina la ubicación de las personas a través de la detección de identificadores o etiquetas instalados en el entorno, como infrarrojos (IR), ultrasonidos (USID), bluetooth *beacons* o códigos de barras. La detección se realiza de manera directa, llevando la persona un lector que indica la posición, inmediatamente, después de detectar un identificador.
- El tercer grupo utiliza triangulación por medio de infrarrojos, ultrasonidos o RFID. Se diferencia del anterior grupo porque utiliza simultáneamente la información procedente de varios sensores.
- Y finalmente, el *reconocimiento de patrones* utiliza datos de uno o más sensores llevados por una persona y compara los datos obtenidos con datos previamente recogidos y almacenados en un mapa. Tanto la visión por ordenador como la técnica del WPS *fingerprinting* son ejemplos de reconocimiento de patrones.

Finalmente, en [11] se evalúan distintos métodos de localización basados en WiFi. En dicho trabajo, los métodos de posicionamiento en interiores se dividen en tres grandes grupos:

- *Fingerprinting*: en los cuales se utilizan señales obtenidas por dispositivos móviles tales como WiFi, sonido ambiental, luz, campos magnéticos, etc. Su principal limitación es la necesidad de crear una base de datos inicial, siendo poco adaptable a nuevos cambios. Esta base de datos permite la localización de

personas u objetos mediante la búsqueda de las señales, más similares a las obtenidas, por un dispositivo móvil portado por la persona o montado en el objeto.

- *Crowdsensing*: es una ampliación de *fingerprinting* que recolecta continuamente los datos obtenidos por los teléfonos móviles, llevados por las personas, con el fin de actualizar la base de datos de posicionamiento.
- *Dead Reckoning Systems*: utilizan datos de los acelerómetros instalados en diversos dispositivos, tales como teléfonos móviles, para obtener el desplazamiento inercial de una persona, así como del sensor de campo magnético para obtener su dirección.

Como se puede extraer de los párrafos anteriores, existe una amplia variedad de tecnologías utilizadas para obtener posicionamiento en interiores. En la Figura 2.1 se muestran algunos ejemplos.

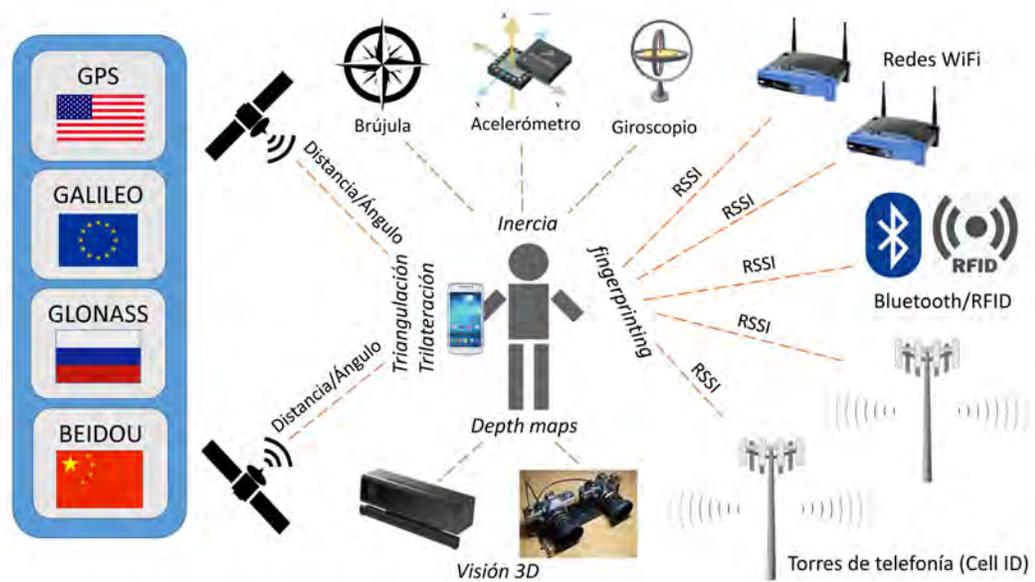


Figura 2.1: Tecnologías utilizadas en posicionamiento en interiores

En las secciones siguientes se detallan algunas de estas tecnologías, en las que se basa el sistema de navegación propuesto en esta tesis, y se presentan trabajos desarrollados en torno a estos dispositivos.

2.1.1. Posicionamiento mediante satélites

Una de las tecnologías más utilizadas para el posicionamiento de objetos y personas es la basada en el uso de señales recibidas de satélites, conocida como Sistema Global de Navegación por Satélite (GNSS) [12].

El Sistema de Posicionamiento Global (GPS) [13] funciona mediante una red de 24 satélites, en órbita sobre la Tierra, que operan a 20200 km de altura cubriendo trayectorias sincronizadas sobre la superficie terrestre. Los receptores necesitan un mínimo de tres satélites de la red para calcular su posicionamiento. Dichos receptores reciben de cada satélite una señal indicando la identificación y la hora del reloj de cada uno de ellos. En base a estas señales, el dispositivo receptor sincroniza el reloj del GPS y calcula el tiempo que tardan en llegar las señales al equipo. A partir de ese tiempo de llegada, *Time of Arrival* (TOA), el receptor calcula la distancia al satélite. Conocidas las distancias, se determina fácilmente la propia posición relativa respecto a los satélites mediante el método de trilateración [14].

Otros sistemas utilizados [12] son: GLONASS (del ruso *Globalnaya Navigatsionnaya Sputnikovaya Sistema*), desarrollado por la Unión Soviética, siendo hoy administrado por la Federación Rusa; BeiDou, el sistema de China; y Galileo, sistema abierto desarrollado por la Unión Europea. A diferencia de los sistemas GPS, GLONASS, y Galileo, que utilizan satélites en órbitas bajas y ofrecen servicio global, BeiDou utiliza satélites en órbita geoestacionaria. Esto implica que, aunque no requiera una gran constelación de satélites, su área de cobertura se reduzca a China. Otra diferencia de BeiDou, con respecto a los otros sistemas mencionados, es que calcula las coordenadas únicamente con dos satélites y una estación en tierra, por lo que necesita enviar una señal desde el dispositivo remoto, algo que no es necesario en GPS, GLONASS o Galileo.

Galileo es interoperable con los sistemas GPS y GLONASS. Esto quiere decir que el usuario puede calcular su posición con un receptor que, utiliza satélites de distintas constelaciones. Al ofrecer dos frecuencias en su versión estándar, Galileo ofrece una precisión del orden de 1 metro para el sistema gratuito, y hasta 1 cm en el de pago.

Aunque este tipo de tecnologías ofrecen resultados buenos en exteriores, no proporcionan mucha precisión en interiores [15], siendo Galileo el que mejores prestacio-

nes ofrece en estos entornos [16]. Esto lleva a buscar otras técnicas más precisas de posicionamiento dentro de edificios, como son las basadas en GSM (Sección 2.1.2), WiFi (Sección 2.1.3), en RFID (Sección 2.1.4), en visión artificial (Sección 2.1.5), o en la combinación de diversas tecnologías (Sección 2.1.6).

2.1.2. Posicionamiento mediante torres de telefonía móvil (GSM)

El *Global System for Mobile Communications* (GSM) utiliza torres de telefonía para establecer la comunicación entre los usuarios. Entre las tecnologías más utilizadas, para el posicionamiento de objetos y personas mediante este sistema, está la basada en el uso de la señal recibida de las torres de telefonía (RSSI).

Ibrahim y Youssef [17] presentan un sistema, denominado *CellSense*, que calcula la posición de teléfonos mediante un algoritmo probabilístico que hace uso de un mapa *fingerprint* creado con las señales RSSI de las distintas torres de telefonía.

En [18], los autores presentan un método de posicionamiento basado en GSM que consigue localizar a una persona con un error de precisión inferior a 4 metros cuando el usuario se encuentra navegando dentro de un edificio. Al igual que el método presentado en el párrafo anterior, utilizan un mapa *fingerprint* creado a partir de torres con alta relevancia para el usuario, es decir las 6 torres con señal más intensa utilizadas habitualmente en GSM.

Aunque este tipo de posicionamiento puede ser utilizado en aplicaciones que no requieren alta precisión en interiores, el posicionamiento basado en WiFi, mostrado en la sección siguiente, ofrece mejores resultados.

2.1.3. Sistemas de posicionamiento WiFi

Los Sistemas de Posicionamiento WiFi se basan principalmente en la técnica *fingerprinting* [19]. El objetivo de esta técnica es generar un mapa del entorno con las intensidades de señal WiFi, *Received Signal Strength Indication* (RSSI), registradas en diversos puntos del área bajo estudio. RSSI es una escala de referencia utilizada para medir el nivel de potencia, de las señales recibidas por un dispositivo, en una red inalámbrica (generalmente WiFi o telefonía móvil). El mapa producido se emplea

para obtener la posición de una persona u objeto en tiempo real, comparando los valores recibidos, desde el dispositivo portátil, con los almacenados en el mapa.

Según Martin et al. [20], esta estrategia puede estudiarse en profundidad mediante las siguientes técnicas: *Vecino más cercano en señal*, *Polígono más pequeño medio* y *Vecino más cercano en señal con medias de puntos de acceso*.

Una reciente comparación, entre los sistemas de posicionamiento en interiores basados en WiFi, ha sido presentada por He y Chan [21]. En dicho trabajo, los autores explican cómo utilizar patrones de señales temporales o espaciales, sensores de movimiento y cómo la colaboración activa de usuarios puede mejorar el posicionamiento. Además, se analizan los recientes avances en la reducción del tratamiento de datos *off-line* en *fingerprint*, la calibración de dispositivos heterogéneos para la recopilación de señales y las mejoras de la eficiencia energética para los *smartphones*.

Adicionalmente, Quan et al. [22] muestran cómo los WPS basados en *fingerprint maps* funcionan mejor que determinadas técnicas basadas en triangulación, como RADAR [23], la cual registra y procesa la información de la intensidad de la señal en múltiples estaciones base y combina las mediciones empíricas con la modelación de propagación de la señal, para determinar la ubicación de las personas por medio de la triangulación. De una forma similar, Shin et al. muestran en [24] un algoritmo diferente, para posicionamiento de dispositivos Android mediante WiFi, como alternativa a otros algoritmos como RADAR [23].

En [22], y de acuerdo con las técnicas presentadas en [20], el posicionamiento mediante *fingerprint map* se lleva a cabo de dos maneras: en primer lugar, considerando el vecino más cercano, donde se calculan las distancias euclídeas entre la lectura RSSI en tiempo real y cada huella en distintos puntos de referencia del mapa para determinar la posición, y por otro lado utilizando la ubicación probabilística de Markov, donde los datos estadísticos de la huella se utilizan para adivinar la posición más probable. Los resultados mostrados indican que el enfoque de vecino más cercano funciona mejor que el de Markov. El método de triangulación proporciona peores resultados [2, 9, 25] debido a la presencia de paredes y obstáculos. Otros artículos han tratado de obtener mejores resultados mediante el uso de la lógica difusa [26] o filtros de partículas [27].

Considerando la distancia entre APs y receptores, Kornuta et al. [28] analizan la

atenuación de la señal producida cuando los APs están lejos del receptor, o hay paredes u obstáculos a lo largo del camino. Por otro lado, diversos filtros son estudiados en [29] para atenuar el ruido de RSSI.

Mediante una técnica más compleja, Cáceres et al. [30] plantean un sistema para localización de vehículos, tanto en interiores como en exteriores, utilizando una red neuronal entrenada a partir del *fingerprint* WiFi obtenido y utilizando filtros para reducir el ruido de las señales. Los autores de [31] también analizan un modelo de red neuronal válida para implementar el problema de localización mediante RSSI *fingerprinting*.

Finalmente, y como un uso adicional de la técnica *fingerprinting*, Husen y Lee [32] proponen la utilización de un *fingerprint map* para obtener la orientación de una persona.

De los trabajos presentados en este apartado, se puede extraer que los sistemas basados en WiFi *fingerprint* no ofrecen resultados precisos. Como se indica en [33], el error usual de WPS al posicionar personas dentro de un entorno es de 2 metros. Este valor resulta impreciso en un posicionamiento en interiores, a pesar del uso de unidades de medición inercial (IMU).

2.1.4. Sistemas de posicionamiento RFID

Existen diversas técnicas que hacen uso de RFID para la localización y seguimiento de personas u objetos dentro de edificios, tal como se presenta en [34], donde se utilizan etiquetas RFID fijas en el suelo gracias a las cuales los usuarios pueden conocer sus posiciones mediante un lector RFID que llevan consigo.

Las técnicas basadas en RFID presentan varias desventajas, tales como interferencias con materiales y dispositivos. Además, no proporcionan resultados de localización demasiado precisos. Estos inconvenientes, entre otros, hacen necesario encontrar una solución alternativa que ofrezca mejores resultados en términos de precisión.

2.1.5. Posicionamiento mediante visión artificial

Las tecnologías basadas en visión artificial, como las que hacen uso de vídeo e imágenes RGB o las que utilizan sensores RGB-D, son un sistema alternativo para

estimar la posición de personas u objetos en diferentes tipos de entornos. Una de las principales ventajas es que son soluciones sencillas y económicas. Además, ofrecen una alta precisión de posicionamiento. En [35], se presenta un estudio de un sistema de posicionamiento óptico en interiores. En este trabajo, los autores clasifican las alternativas existentes de acuerdo con la referencia utilizada, para determinar la posición de las personas en la escena, como por ejemplo modelos 3D, imágenes, marcadores codificados o patrones proyectados.

Entre los métodos que utilizan imágenes, RGB, Tsai et al. [36] utilizan cámaras de vigilancia normales para capturar vídeo. A partir de ese vídeo, el sistema extrae los objetos de primer plano utilizando un modelo de fondo. Los objetos obtenidos se etiquetan y, después de una Transformación Lineal Directa 2D, se muestran en una vista aérea. Esta transformación se realiza utilizando las coordenadas de la cámara y las coordenadas del punto de vista aéreo. Aunque este sistema proporciona información sobre la posición de una persona, no permite obtener su identificación.

Kuo et al. [37] presentan un sistema basado en luces LED que han sido colocadas en el entorno y actúan como balizas. Estas luces transmiten sus identificadores y localizaciones codificadas en pulsos ópticos imperceptibles para el hombre. La cámara de los teléfonos extrae la identificación y ubicación de las luces de la imagen y determina la ubicación y orientación del teléfono con respecto a las luces. Siguiendo una técnica más compleja, Werner et al. [38] han desarrollado un sistema, llamado DeepMoVIPS, que utiliza redes neuronales convolucionales para la geolocalización simbólica en interiores. Esta técnica se basa en imágenes obtenidas por *smartphones*, a partir de las cuales se extraen una serie de características, para cada imagen, mediante convolución y agrupación y, posteriormente, clasificación. El proceso de ubicación se lleva a cabo después de un período de entrenamiento, y es afectado por los cambios en el escenario (por ejemplo, por la presencia de personas).

Entre los dispositivos basados en visión artificial cabe destacar los sensores RGB-D, cada vez más avanzados durante los últimos años. Se han desarrollado sistemas de posicionamiento, como OpenPTrack [4], que crea una solución escalable y multi-cámara, para el seguimiento de personas. Y otro enfoque, el método *Kinect Positioning System* (KPS), se presenta en [5] con el objetivo de obtener la posición de una persona. Este sistema tiene la limitación de no conocer la identidad de la persona

monitorizada a no ser que lleve algún distintivo visual. Por este motivo, en [3], los autores introducen una metodología, para rastrear a los trabajadores de la construcción, usando simultáneamente técnicas de procesamiento de imágenes e información de profundidad generada a partir de un sensor Kinect. Este sistema de seguimiento obtiene las ubicaciones de los obreros y sus movimientos en un área de trabajo cubierta. El estudio considera diferentes tipos de cascos coloreados codificados como una clave para diferenciar al personal. Por otro lado, Saputra et al. [6] presentan una aplicación de seguimiento humano en interiores con dos cámaras de profundidad.

En algunos trabajos los propios usuarios portan el sensor RGB-D. Así por ejemplo, en [39], los autores utilizan un sensor Kinect llevado por una persona para calcular la trayectoria del usuario. En el trabajo se evalúan distintos métodos SLAM (*Simultaneous localization and mapping*) implementados con Kinect mediante algoritmos como SIFT, SURF, RANSAC, ICP, o TORO [40]. Sin embargo, como en los enfoques mencionados anteriormente, estas dos últimas soluciones requieren una participación directa de los usuarios. Además, la presencia de otras personas en las imágenes o cambios en el escenario puede afectar el resultado.

Takizawa et al. [41] proponen un *smart-cane* para los discapacitados visuales que, con la ayuda de un sensor Kinect, permite localizar objetos.

Algunos sensores RGB-D, como Kinect, permiten obtener el esqueleto de las personas que se encuentran en su campo de visión. El esqueleto incluye las coordenadas de distintas articulaciones del cuerpo, como son: el cuello, los tobillos, las rodillas, la cadera, los codos o las muñecas. De entre los trabajos que hacen uso de los esqueletos, cabe destacar el de Sevrin et al. [42], que propone un modelo para fusionar trayectorias Kinect, similar al método utilizado por el sistema de posicionamiento presentado en esta tesis. Teniendo en cuenta las coordenadas globales en el mismo *time stamp*, el sistema es capaz de determinar la distancia entre dos esqueletos para discernir entre diferentes personas. Sin embargo, esa solución no recupera datos de identificación de los *smartphones* y, por lo tanto, no es capaz de proporcionar al usuario final su posición.

Adicionalmente, en [43], se plantea la utilización del esqueleto de una persona y los mapas de profundidad, devueltos por la Kinect, como un medio de interacción con una pizarra virtual situada a varios metros de la real. Los autores de [44] plantean

la relación entre las coordenadas de las articulaciones, del esqueleto de una persona, devueltas por la Kinect y las coordenadas de un manipulador electro-hidráulico. En el artículo buscan un modelo para obtener las posiciones de las articulaciones del manipulador, a partir de las articulaciones del esqueleto humano. En [1] se estudia la extracción de distintas personas solapadas de una escena, mediante el uso de tres sensores Kinect. Y también en esa misma línea, Ye et al. [45] han desarrollado un método para detectar e identificar a varias personas, que están ocluidas por otras en una escena.

Una de las desventajas de los sistemas basados en visión artificial es la complejidad de identificar a personas en el escenario. Tal y como se aprecia en los trabajos presentados previamente, los sistemas tienen que diferenciar a dichas personas por medio de técnicas adicionales, tales como la detección de rostros en tiempo real [46–48], o por medio de códigos de color o patrones [3]. Sin embargo, ambas técnicas plantean problemas con oclusiones y la necesidad de involucrar directamente a los usuarios en el sistema.

El método propuesto en esta tesis hace uso de la precisión de los sistemas de visión artificial sin la necesidad de que los usuarios porten algún distintivo especial. Como se explicará en las siguientes secciones, la razón de utilizar sensores RGB-D es aumentar la precisión del posicionamiento basado en WiFi *fingerprint*, utilizando esta segunda técnica para identificar a los usuarios.

2.1.6. Sistemas de posicionamiento combinado

Como han afirmado numerosos autores, la combinación de múltiples tecnologías mejora la eficiencia de las WPS tradicionales [21, 49, 50]. Los sistemas híbridos combinan WPS con otras tecnologías, como GSM [51], bluetooth [52], o RFID [53, 54].

También, las Unidades de Medición Inercial (IMU, del inglés *Inertial Measuring Units*), se combinan con técnicas basadas en visión artificial para ubicar a personas en entornos complejos. Como se muestra en [55], los autores implementan un sistema de odometría visual inercial para dispositivos móviles. Ese método utiliza las características extraídas de imágenes consecutivas para obtener la posición relativa y ajustar la posición y orientación final obtenidas por los sensores móviles

(giroscopio, acelerómetro, magnético). Los autores utilizan imágenes tomadas por el propio dispositivo móvil. El trabajo [56] estudia la combinación de WiFi y *Sistemas de Navegación Inercial* (INS) con el fin de obtener la trayectoria de personas. Se utilizan tres sensores: giroscopio, acelerómetro y un sensor de presión atmosférica. Además, se utilizan filtros de partículas y Kalman con el fin de reducir el ruido del posicionamiento.

Algunos sistemas, como la API de geolocalización de *Google Maps* [57], utilizan información de dos tecnologías diferentes (GSM y WiFi) para obtener la posición de usuarios por medio de mapas *fingerprint*. En la línea de utilizar mapas *fingerprint* globales, Wilk et al. [58] plantean un método de actualización continua del mapa, mediante *crowdsourced* WiFi, utilizando procesamiento *off-line* de las trayectorias seguidas por los usuarios.

Desde el punto de vista de la visión artificial, Mirowski et al. [59] analizan cómo generar un *fingerprint map* con un sensor RGB-D montado en un robot. La utilización de Kinect permite generar un mapa en tiempo real y grabar, en los puntos por donde pasa el robot, las mediciones de intensidad RSSI a distintos puntos WiFi. También en [60], un robot se localiza por medio de tres sistemas diferentes: un telémetro láser, una cámara de profundidad y los valores RSSI. Cada sistema se utiliza independientemente de acuerdo con la zona donde se encuentra el robot. Sin embargo, ninguna de estas dos soluciones explota las diferentes tecnologías simultáneamente. En el primer trabajo, los sensores RGB-D se utilizan para obtener la posición real, y en el segundo, los autores solo utilizan un sistema a la vez.

En [54], los autores proponen combinar la identificación y el posicionamiento basados en RFID con el sensor Kinect, para obtener la posición precisa de una persona dentro de un entorno. En este caso, un lector RFID se encuentra fijo en la habitación. Cada usuario lleva su propia etiqueta RFID, que permite identificarle, mientras que el sensor Kinect obtiene los esqueletos de las personas de cara a mejorar la precisión de localización. Cada esqueleto se compone de las coordenadas de las diferentes articulaciones de una persona: cuello, hombros, codos, rodillas, etc.

Los sistemas combinados ofrecen resultados más precisos, pero normalmente requieren dispositivos especiales. Como se mencionó anteriormente, una nueva solución para el posicionamiento en interiores se detalla en esta tesis. Se han combinado dos

técnicas de naturaleza diferente (sensores WPS y RGB-D) en un sistema único para mejorar la precisión de trabajos anteriores. A diferencia de otros trabajos, en este nuevo sistema los usuarios solo tienen que portar teléfonos móviles, ampliamente utilizados. Las cámaras Kinect utilizadas son económicas y permiten precisar la posición de las personas sin la necesidad de que lleven distintivos especiales. Como se muestran en capítulos posteriores, se han realizado experimentos en interiores habitados con resultados prometedores.

2.2. Predicción de movimientos

La predicción de los movimientos de personas es útil para el desarrollo de entornos inteligentes, donde se busca la reducción del consumo energético y la mejor optimización de recursos [61, 62]. Conocida la identidad de los usuarios, el entorno inteligente podría encender/apagar luces o controlar sistemas de calefacción o aire acondicionado y otros dispositivos.

En los últimos años se han desarrollado varios trabajos en el campo de la predicción en movilidad. La mayoría de autores [63, 64] consideran la idea de los patrones de movimientos, donde los movimientos de las personas son modelados como una repetición de algunos patrones de movimiento básicos. En [63], un algoritmo de predicción de movimiento móvil (MMP), basado en patrones y reconocimiento del movimiento, devuelve las próximas posiciones de las personas. Otras técnicas de predicción en esta línea hacen uso de regresión [65], el filtro de Kalman [66], o las redes neuronales [67]. Una desventaja de estos métodos se encuentra en los cambios abruptos en la trayectoria.

En [68], los autores exploran un nuevo algoritmo jerárquico de predicción de localización (HLP), que es capaz de tratar con movimientos aleatorios. Se compone de un algoritmo de comparación de patrones, que extrae cualquier patrón de movimiento regular existente y estima la dirección global entre celdas (creada en una red ATM inalámbrica). Después de esto, usa un filtro de Kalman de autoaprendizaje extendido, que con movimientos aleatorios, predice la siguiente celda. Otro trabajo, [69], utiliza un filtro robusto extendido de Kalman (REKF), para estimar la siguiente estación base móvil en la que se conectará el teléfono móvil del usuario. Los experimentos que

se desarrollan con la trayectoria de un coche, mejoran la confiabilidad de la conexión y la eficiencia del ancho de banda del sistema subyacente.

La minería de datos también se ha utilizado en la predicción del movimiento. Algunos autores, como [70], proponen un algoritmo de tres fases que, primero obtiene patrones de movilidad de las trayectorias de una persona, mediante la minería de datos, segundo extrae las reglas de movilidad de estos patrones y, finalmente, calcula las predicciones usando estas reglas.

En computación ubicua [71], las estrategias activas predicen el movimiento de una persona, a través de trayectorias y posiciones, para proporcionar una información de ubicación y contexto más rica. Este trabajo presenta dos modelos para la predicción de la trayectoria: un modelo basado en la probabilidad y otro en el aprendizaje. El modelo basado en la probabilidad almacena trayectorias históricas, a largo plazo, para el cálculo de probabilidad y la predicción de trayectoria.

Finalmente, en [72], los autores proponen un modelo mixto de cadena de Markov (MMM) que considera la personalidad del peatón como un parámetro inobservable. También consideran el estado anterior del peatón. Sus experimentos se han realizado en un centro comercial, donde han demostrado la más alta exactitud de predicción de su método.

2.3. Sistemas semánticos de posicionamiento

Los sistemas modernos de posicionamiento en interiores (IPS) ofrecen a los usuarios la posibilidad de interactuar con su entorno. Estos sistemas permiten obtener la posición de personas u objetos dentro de un edificio. En una empresa, los empleados pueden tener acceso a áreas particulares o habilitar dispositivos concretos. En un museo, los usuarios pueden obtener información sobre piezas y obtener el camino óptimo a otra parte del museo.

En los últimos años, se han presentado algunos trabajos que muestran una visión semántica del problema de navegación en interiores. En [73], los autores proponen una *Ontología de Navegación Interior* (INO), que soporta la búsqueda de rutas y las tareas de presentación de un sistema de navegación. Más recientemente, una versión extendida de INO es utilizada en [74], dando soporte a la realidad aumentada para

dar una experiencia más rica a los usuarios. Otra visión del enrutamiento en interiores para individuos es dada por [75], donde los autores proponen otra ontología (ONALIN) que provee una ruta de búsqueda de individuos con varias necesidades y preferencias. Los autores analizan diferentes casos de usuarios: discapacitados visuales, parapléjicos, ancianos o alguien con una discapacidad física temporal.

El trabajo [76] presenta un servicio basado en localización (LBS), que permite obtener la ruta a un destino, así como una ontología, la cual enriquece el posicionamiento estándar porque los usuarios pueden compartir, gestionar y consultar datos semánticamente. Esto proporciona información sobre lugares relevantes.

OGC GeoSPARQL [77] ofrece soporte para representar y consultar datos geoespaciales en la web semántica. Sin embargo, está más enfocado en representar topología geométrica, como polígonos, en lugar de ser utilizado como una herramienta con el propósito de posicionamiento.

Los autores de [78] proporcionan un enfoque de localización y navegación basado en la propiedad topológica, de la conectividad entre las entidades espaciales, y la información proporcionada por el usuario. El espacio interior es modelado como superficies (pisos) con contenedores (salas, pasajes corredores y conectores), portales y barreras. Esta ontología también se presenta en [79, 80].

En [81], una ontología se utiliza para representar un entorno interior y los objetos que aparecen. La ontología proporciona información para los robots de servicio con el fin de identificar objetos, aunque puedan estar ocluidos. Se utiliza un mapa métrico para instanciar datos sobre áreas vacías y ocupadas en el espacio. El mapa topológico incluye nodos enlazados a los objetos usando bordes de objeto. El mapa semántico vincula el mapa topológico con la representación de los objetos en la sala o edificio apropiado.

El trabajo [82] presenta una extensión del esquema *Open Street Map* (OSM) para ambientes interiores. OSM proporciona información sobre paisajes, redes de calles y edificios. Pero la información de los edificios está relacionada con el espacio exterior. La ontología extiende OSM incluyendo datos para las partes internas de un edificio: salas, ventanas, puertas, posiciones, tamaños, materiales, etc.

2.3.1. Ingeniería del conocimiento

En el campo de la ingeniería del conocimiento, el término ontología [83, 84] significa una especificación de una conceptualización. Una ontología es la definición y clasificación de conceptos y entidades, y las relaciones entre ellos. El conocimiento de un dominio está representado en un formalismo declarativo. El conjunto de objetos que se pueden representar se denomina universo del discurso.

Este conjunto de objetos, y las relaciones descriptibles entre ellos, se reflejan en el vocabulario representacional con el que un programa basado en el conocimiento, representa el conocimiento. Las ontologías utilizan entidades en el universo del discurso (por ejemplo, clases, relaciones, funciones u otros objetos) y axiomas formales, que limitan la interpretación y el uso bien formado de estos términos.

Las ontologías pueden representarse en OWL. El *W3C Web Ontology Language* (OWL) [85] es un lenguaje Web semántico diseñado para representar el conocimiento sobre cosas, grupos de cosas y relaciones entre las cosas. OWL es un lenguaje basado en lógica computacional tal que el conocimiento expresado en OWL puede ser explotado por programas informáticos. Los documentos de OWL pueden ser publicados en la *World Wide Web* y pueden referirse o ser referidos por otras ontologías de OWL, siendo OWL parte del conjunto de tecnologías del W3C, que incluye RDF, RDFS, SPARQL, etc.

El marco de descripción de recursos, *Resource Description Framework* (RDF), [86, 87], es una especificación del W3C, originalmente diseñada como un modelo de datos de metadatos, para representar información sobre recursos en la Web. Proporciona una estructura (marco) para describir las cosas identificadas (recursos). RDF también se puede usar para representar información sobre cosas que se pueden identificar en la Web, incluso cuando no se pueden recuperar directamente en la Web (por ejemplo, un libro o una persona). Los *Identificadores de Recursos Uniformes* (URI) son similares a las URL, pero pueden no representar una página web real. RDF es flexible en términos de cómo se pueden explorar las relaciones de datos, y es eficiente porque los datos se pueden leer rápidamente. No sigue una estructura lineal, como las bases de datos tradicionales, y no es jerárquico como XML.

Capítulo 3

Tecnologías

En este capítulo se abordan las tecnologías utilizadas, comenzando por los sensores-cámaras RGB-D (Sección 3.1), y continuando por los teléfonos basados en Android (Sección 3.2) y los puntos de acceso WiFi (Sección 3.3).

3.1. Sensores RGB-D

Los sensores o cámaras RGB-D son dispositivos que permiten obtener imágenes o mapas de profundidad, los cuales son matrices bidimensionales que contienen valores asociados a las distancias existentes entre la cámara y los objetos representados en el mapa.

Para explicar el funcionamiento de estos sensores, nos basaremos en el dispositivo Kinect 360, gracias a que los análisis llevados a cabo del sensor [88] y de las patentes de PrimeSense [89] han desvelado bastante sobre su funcionamiento. El *hardware* de Kinect 360 está basado en los diseños de la empresa PrimeSense. Como se detalla en los siguientes párrafos, Kinect 360 posee distintos componentes que lo hacen muy interesante desde el punto de vista de la visión artificial (ver Figura 3.1).

Este dispositivo utiliza un emisor láser de luz infrarroja (IR) estructurada, capaz de generar un patrón aleatorio de luz para cada punto de la escena tridimensional. Esto se lleva a cabo utilizando un difusor y un elemento difractor de luz infrarroja. El láser emite un único rayo que se divide formando el patrón, que es constante y se proyecta en la escena. Dicha proyección, generada por el emisor de puntos, es recibida

por una cámara CMOS IR (del inglés *complementary metal-oxide-semiconductor infrared*) que observa la escena (ver Figura 3.2).

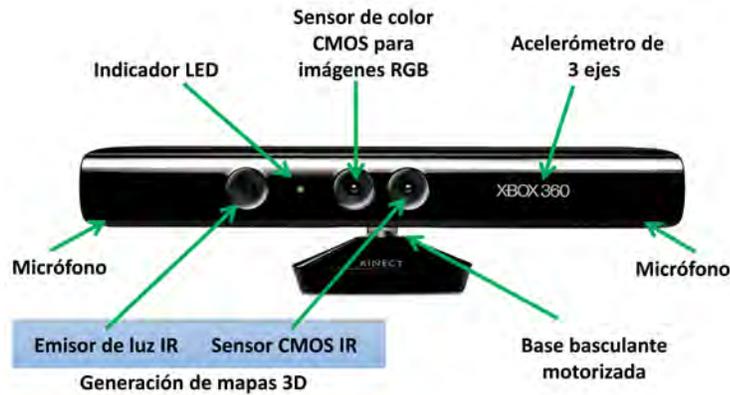


Figura 3.1: Partes de Kinect 360

La calibración [90] entre el proyector y la cámara debe ser conocida y es grabada en el momento de la construcción del dispositivo. Para llevar a cabo dicha calibración se toman un conjunto de imágenes de referencia, en distintas localizaciones, para ser almacenadas en memoria.

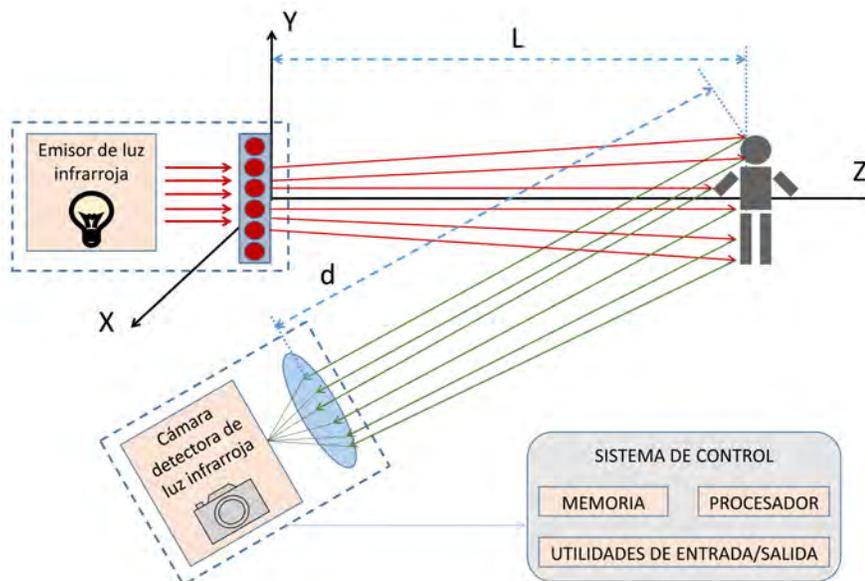


Figura 3.2: Esquema de funcionamiento del emisor/cámara IR de Kinect 360

El cálculo del mapa de profundidad, o mapa de las distancias del sensor a cada punto de la escena 3D, se realiza por medio de la triangulación (ver Figura 3.3) de cada punto emitido desde el emisor, q_j o q_{j+1} , y el punto recibido en el receptor, e_i o e_{i+1} , mediante la imagen observada (mota).

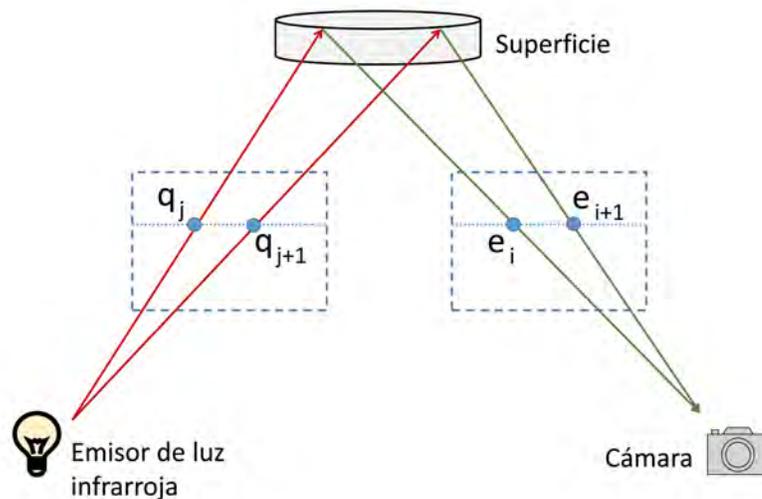


Figura 3.3: Triangulación para obtener mapa de profundidad con Kinect 360

De entre todos estos sensores, se ha utilizado Kinect v2 por su bajo coste y por la implementación de la librería de extracción, en los mapas de profundidad generados, de las coordenadas correspondientes a la posición de cada una de las articulaciones, que definen el esqueleto de una persona. Cabe destacar que esta librería permite obtener las coordenadas correspondientes a los esqueletos de varias personas que han sido detectadas, simultáneamente, por el dispositivo [91, 92].

Existen distintos sensores RGB-D en el mercado, entre los que podemos mencionar: Intel Real Sense [93] (ver Figura 3.4a), Asus Xtion PRO Live [94] (ver Figura 3.4b), Structure Sensor [95] (ver Figura 3.4c), Peregrine 3D LIDAR Vision System [96] (ver Figura 3.4d), IFMO 3D [97] (ver Figura 3.4e), o Kinect v2 [98] (ver Figura 3.4f).



(a) Intel Real Sense



(b) Asus Xtion PRO Live



(c) Structure Sensor 3D



(d) Peregrine 3D



(e) IFMO 3D



(f) Kinect 2.0

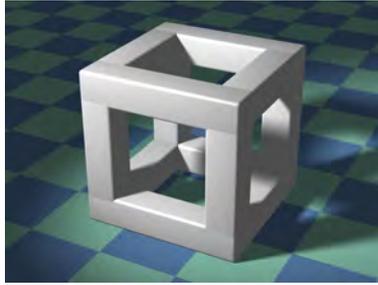
Figura 3.4: Sensores RGB-D

3.1.1. Mapas de profundidad

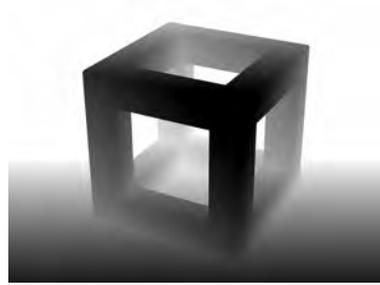
Como se ha mencionado anteriormente, un mapa de profundidad es, simplemente, una matriz bidimensional cuyas celdas están pobladas de distancias. Se puede asemejar, en cierto modo, a una fotografía en escala de grises, donde cada uno de los píxeles toma un valor dependiendo de cuán oscura o clara es la zona representada en esa diminuta porción de imagen. Así, las distancias almacenadas en cada una de las celdas del mapa de profundidad, pueden dar lugar a zonas más o menos claras si dicho mapa es representado como una imagen en escala de grises.

Esta representación permite hacerse una idea rápida de la información almacenada en el mapa de profundidad. Por ejemplo, en la Figura 3.5, se ve una imagen RGB de un cubo (Figura 3.5a) junto al mapa de profundidad en escala de grises (Figura 3.5b). En el mapa, los puntos más próximos al espectador tienen colores más cercanos al negro, mientras que los puntos más lejanos tienen colores más cercanos al blanco.

Si se conocen los parámetros del dispositivo, con el que se ha obtenido el mapa de profundidad (ángulo de visión, tipo de lente, etc.), es posible construir a partir



(a) Imagen RGB



(b) Mapa de profundidad

Figura 3.5: Imagen RGB y mapa de profundidad. Por Dominic Alves (2010)

de dicho mapa una representación tridimensional de la escena.

El sensor Kinect 360 proporciona mapas de disparidad [89], que no representan, exactamente, el mismo concepto que los mapas de profundidad. Sin embargo, al ser mapas normalizados, su comportamiento es similar a los mapas de profundidad.

Los mapas generados por Kinect 360 presentan una resolución de disparidad de 11 bits (2048 valores) y unas dimensiones de 640x480 píxeles. Para interpretar correctamente los valores de disparidad, estudios detallados del proceso de calibración de la Kinect 360 [99, 100] proporcionan la Ecuación 3.1 como la relación entre la distancia real y la disparidad normalizada,

$$p = \frac{1}{3.3309495161 - 0.0030711016 \cdot d'} \quad (3.1)$$

donde p representa la distancia en metros a un punto de la matriz devuelta y d' representa la disparidad normalizada devuelta por Kinect 360.

El trabajo [100] realiza un estudio sobre la obtención de mapas de profundidad del sensor Kinect 360. Los autores analizan el funcionamiento, la calibración y la desviación a distintas distancias.

3.1.2. Kinect para Windows

Se han desarrollado dos dispositivos Kinect: Kinect 360 y Kinect One v2, o simplemente Kinect 2.

Kinect 360

Kinect para Xbox 360 (ver Figura 3.6a), o simplemente Kinect, es un controlador de juego libre y entretenimiento desarrollado por Microsoft para la videoconsola Xbox 360, y desde junio del 2011 para PC a través de Windows 7 y Windows 8. Kinect permite a los usuarios controlar e interactuar con la consola, sin necesidad de tener contacto físico con un controlador de videojuegos tradicional, mediante una interfaz natural de usuario que reconoce gestos, comandos de voz, objetos e imágenes. El dispositivo tiene como objetivo primordial aumentar el uso de la consola Xbox, más allá de la base de jugadores que posee en la actualidad. Kinect ha competido con los sistemas Wiimote con Wii Motion Plus y PlayStation Move, que también controlan el movimiento para las consolas Wii y PlayStation 3/4, respectivamente.

Kinect 360 utiliza una cámara CMOS RGB para obtención de las imágenes, que permite obtener un flujo continuo para grabación y tratamiento de vídeo, con un *frame rate* que varía entre 12 y 30 fps, dependiendo de la resolución del flujo de vídeo.

En la generación del mapa de profundidad, el tamaño y forma de las motas recibidas depende de la distancia y orientación del sensor. Kinect utiliza tres tamaños diferentes de motas para tres regiones de distancia diferentes:

- En la primera región se obtiene un mapa de profundidad de alta precisión para objetos cercanos (0.8 a 1.2 metros).
- En la segunda región se obtiene un mapa de profundidad de media precisión para objetos a media distancia (1.2 a 2.0 metros).
- En la tercera región se obtiene un mapa de profundidad de baja precisión para objetos lejanos (2.0 a 3.5 metros).

Kinect One v2

Kinect 2 (ver Figura 3.6b) ofrece una resolución de mapa de profundidad de 512x424 píxeles. La resolución de las imágenes es de 1920 x 1080 píxeles con un refresco de 30Hz, mucho más alto que la resolución y refresco de Kinect 360 (640 x 480 píxeles a 30Hz). En Kinect 360, los mapas de profundidad tenían una resolución de



Figura 3.6: Sensores Microsoft Kinect.

640x480 píxeles. Aunque a simple vista parece que la resolución es mayor, existen diferencias significativas. Por ejemplo, la cámara de profundidad de Kinect 360 es una cámara virtual creada por la combinación de imágenes de la cámara real de infrarrojos (IR), cuya resolución es 1280x1024, con patrones de luz generados por el emisor de infrarrojos. Los patrones de luz emitidos son puntos espaciados que hacen que, aunque la resolución de la cámara IR sea de 1280x1024, podrían situarse como máximo en 640x512 posiciones. Teniendo en cuenta que el emisor IR debe crear además un patrón 2D de puntos reconocibles por la cámara IR, se puede decir que, aunque la resolución devuelta es de 640x480, muchos valores no son reales, obteniéndose mediante interpolación. Esto hace que la efectividad de los mapas de profundidad de Kinect 360 sea tal que solo 1 de cada 20 píxeles del mapa de profundidad tienen un valor real. Por este motivo es difícil detectar objetos pequeños con Kinect 360, y más concretamente si se encuentran entre dos puntos del patrón IR. Entre los problemas de Kinect 360 se encuentran el reconocimiento de dedos o caras, algo necesario en las modernas interfaces de interacción natural 3D.

Kinect 2 utiliza una cámara real de resolución 512x424 píxeles del tipo *time-of-flight*. Aunque su resolución es inferior a la de Kinect 360, su efectividad puede ser hasta diez veces superior. Las cámaras basadas en tiempo de vuelo disponen de un emisor, que emite un pulso de luz durante un período concreto. La cámara es capaz de calcular la distancia en función del tiempo de retorno de dicho pulso de luz.

Las cámaras de luz estructurada, como la usada en Kinect 360, presentan el inconveniente de que para medir un punto de la superficie, ese punto tiene que ser visible tanto desde el emisor del patrón como desde la cámara IR. En las cámaras basadas en tiempo de vuelo, el emisor y el receptor podrían estar incluso detrás de la misma lente, al no ser necesario un cálculo basado en la distancia entre el emisor

y la cámara, por lo que no existe prácticamente el problema de que el emisor y la cámara vean puntos no visibles entre ellos [101].

Cabe mencionar que el sensor Kinect 2 necesita un ancho de banda alto para transferir datos, ya que genera 30MB por segundo. Como se recomienda en [102], cada dispositivo requiere su propio concentrador USB 3.0. El uso de dos o más dispositivos conectados al mismo equipo, requiere tarjetas PCI adicionales, ya que la mayoría de los ordenadores actuales solo tienen un puerto USB 3.0 (o dos de ellos conectados al mismo concentrador).

Primeros pasos con Kinect

Existen distintas formas de trabajar con Kinect 360 o Kinect 2.0. Por ejemplo, Microsoft ha desarrollado un *Software Development Kit* (SDK) [88] que puede utilizarse con Visual Studio para desarrollar aplicaciones que hacen uso de los distintos sensores de Microsoft. Por otro lado, la librería Java de código abierto denominada J4K [91], permite enlazar Java con el SDK de Microsoft. Esto permite desarrollar software Java e integrar *plug-ins* de otras librerías como FIJI, para tratamiento y análisis de imágenes. J4K se comunica con la librería nativa de Windows, la cual gestiona los flujos de datos de profundidad, color, infrarrojos y *streams* de tipo esqueleto y los pasa a través del Java Native Interface (JNI).

J4K es compatible con todos los dispositivos Kinect (Kinect for Windows, Kinect 360, o Kinect 2 para Xbox One) y permite controlar múltiples dispositivos a la vez desde una aplicación. Esto último es interesante si se tienen varios dispositivos Kinect funcionando a la vez y conectados a las entradas USB del ordenador, independientemente del modelo Kinect que sean (Kinect 360, Kinect 2, etc.). Además, J4K incluye distintas librerías que convierten los mapas de profundidad, esqueletos o imágenes en distintos objetos Java de fácil utilización.

Para utilizar la librería J4K es necesario extender la clase J4KSDK, donde se define el constructor *Kinect()* para poder llevar a cabo acciones iniciales, incluyendo una llamada al constructor de la clase J4KSDK para poder inicializar el sensor. En el constructor también se crea un objeto de tipo *VideoFrame* para poder leer el último *frame* de vídeo recibido por el sensor en un momento dado. También es necesario implementar el interface de los tres métodos que se llamarán automáticamente cada

vez que un nuevo mapa de profundidad, imagen de vídeo o esqueleto sea recibido por el dispositivo Kinect. Dentro de cada uno de los métodos, se llevan a cabo las distintas operaciones que necesite la aplicación. Así, por ejemplo, el método *onDepthFrameEvent* es invocado cada vez que el sensor obtiene un mapa de profundidad, el método *onColorFrameEvent* es invocado cada vez que se obtiene una imagen RGB-D, y el método *onSkeletonFrameEvent* es invocado cada vez que el sensor obtiene un conjunto de esqueletos.

Un posterior procesamiento de los mapas de profundidad y las imágenes RGB-D proporcionadas por Kinect, mediante técnicas que no son objeto de esta tesis, permite generar nubes de puntos complejas, como la que se aprecia en la Figura 3.7.

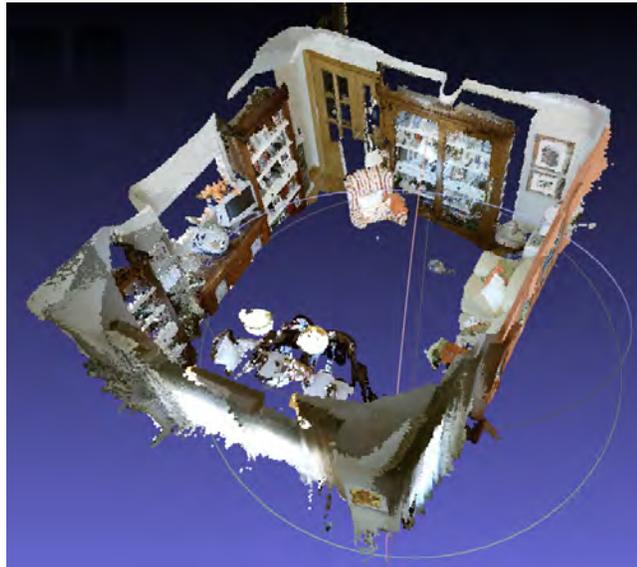


Figura 3.7: Nube de puntos, correspondiente a un entorno habitado, generada a partir de los datos obtenidos por un dispositivo Kinect 2

Obtención de esqueletos con Kinect

Kinect v2 permite extraer hasta 6 esqueletos simultáneamente del mapa de profundidad en un rango aproximado de 5 metros. La detección de esqueletos en Kinect es un aspecto clave que se ha utilizado para el desarrollo de videojuegos Xbox.

Los esqueletos se obtienen por medio de las técnicas presentadas en [91, 92]. En dichos trabajos, los autores proponen algoritmos para predecir, con rapidez y

precisión, las posiciones 3D de las articulaciones del cuerpo, a partir de mapas de profundidad.

Cada esqueleto devuelto por Kinect ofrece las coordenadas de distintas articulaciones del cuerpo humano, tales como cuello, hombros, manos, rodillas. La Figura 3.8 muestra las distintas articulaciones devueltas para cada esqueleto por el sensor Kinect v2.

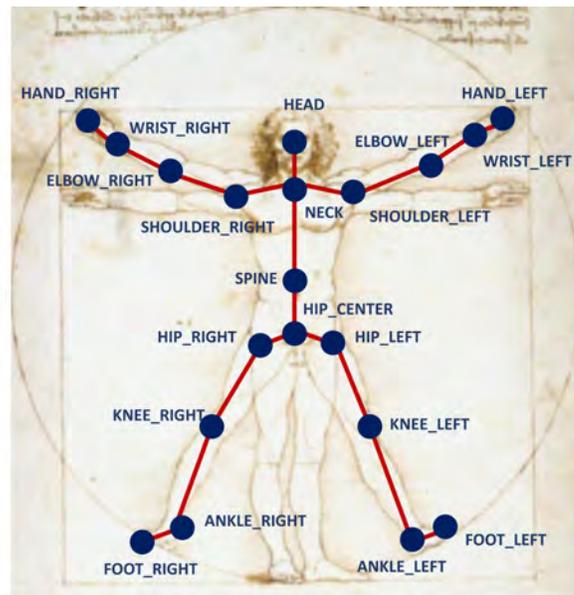


Figura 3.8: Articulaciones devueltas para cada esqueleto por el sensor Kinect v2

La librería J4K permite obtener los esqueletos directamente, sin necesidad de extraerlos del mapa de profundidad. Como se ha mencionado anteriormente, el método *onSkeletonFrameEvent* es invocado cada vez que el sensor obtiene un conjunto de esqueletos. Cada esqueleto se introduce en un objeto *Skeleton*, que permite extraer las coordenadas de las articulaciones de una manera directa.

3.2. Teléfonos móviles basados en Android

Los teléfonos móviles utilizaban originalmente sistemas propios. Desde comienzos de siglo, las compañías tecnológicas trataron de crear sistemas universales compatibles con distintos dispositivos. Android es un sistema operativo basado en Linux,

que está diseñado principalmente para dispositivos móviles con pantalla táctil, como: teléfonos inteligentes, *tablets*, relojes inteligentes, televisores, automóviles e incluso en ciertos electrodomésticos [103].

Android ofrece muchas ventajas a los usuarios, al unificar la forma de interactuar con el sistema en distintos dispositivos y al permitir utilizar aplicaciones genéricas. Además, ofrece ventajas a los desarrolladores, como la creación de contenido genérico no ligado a una plataforma. Android proporciona diversas librerías, como por ejemplo la librería destinada a localización *android.location* [104], utilizada posteriormente, en el Capítulo 6, para obtener la posición de un dispositivo.

Desde el punto de vista del desarrollo, Android ofrece la posibilidad de utilizar su propio entorno de desarrollo, denominado Android Studio [105], pero también permite trabajar con otros entornos como NetBeans o Eclipse.

3.3. Puntos de acceso WiFi

Los señalizadores, balizas, o *beacons*, son los dispositivos que emiten la señal que permite obtener la distancia a los dispositivos móviles. De entre los distintos tipos de señalizadores, los *routers* WiFi están muy extendidos, son económicos y permiten acceder a una red LAN (del inglés *Local Area Network*).

De entre los *routers* WiFi, un modelo interesante es el Linksys Router WiFi 54Mbps WRT54GL (ver Figura 3.9). Es un *router* Wireless-G, estándar de red inalámbrica de 54 Mbps, que proporciona una velocidad casi 5 veces superior que los populares productos Wireless-B (802.11b).



Figura 3.9: Router Linksys WIFI 54 Mbps WRT54GL

Algunas características del Linksys WRT54GL son:

- Firmware Linux actualizable.
- Conexión: Ethernet - Fast-Ethernet, Interfaz de Red: 4 puertos RJ-45 10/100 y 1 puerto RJ-45 WAN 10/100.
- Dos antenas exteriores para mejor alcance WiFi.
- WiFi con banda de frecuencia: 2.4 GHz. Conexión de redes: IEEE 802.11b, IEEE 802.11g. Velocidad de transferencia de datos: 54 Mbps.
- Número de canales seleccionables: 13 canales.
- 64 MB de RAM.
- Alta seguridad: cifrado TKIP y AES, filtrado de direcciones MAC inalámbricas, potente firewall SPI.

Capítulo 4

Análisis de métodos utilizados

En el presente capítulo se abordan los distintos métodos de gestión y procesamiento de datos utilizados durante las investigaciones. En primer lugar (Sección 4.1), se presenta el sistema de posicionamiento combinado mediante WPS y visión artificial, para su uso en una sala. A continuación, se generaliza dicho método a entornos complejos (Sección 4.2). Finalmente, se presentan los métodos seguidos en predicción de movimientos (Sección 4.3).

4.1. Sistema de posicionamiento inalámbrico (WPS) asistido por RGB-D

El objetivo del sistema propuesto, en este apartado, es mejorar la precisión del posicionamiento de las personas dentro de una habitación. Para ello, consideremos un escenario como el ilustrado en la Figura 4.1, que representa el marco genérico del sistema. Se supone que una o varias personas se mueven libremente en una zona de trabajo rectangular, portando su propio dispositivo portátil. Cada dispositivo recibe señal inalámbrica de uno o más puntos ubicados estratégicamente en el área de trabajo. Un sensor RGB-D se coloca de tal manera que la mayor parte del espacio de trabajo esté cubierto. Este dispositivo proporciona un *depth map* y una imagen en color de la escena, que se utilizan para identificar los esqueletos 3D de los usuarios.

Los esqueletos se obtienen por medio de las técnicas presentadas en [91, 92], donde los autores proponen nuevos algoritmos para predecir, con rapidez y precisión, las

posiciones 3D de las articulaciones del cuerpo, a partir de mapas de profundidad. Estos métodos forman un componente central de la plataforma de juegos Kinect.

A partir de estos esqueletos, las coordenadas del cuello se extraen con el objetivo de posicionar a las personas en el entorno. Esta parte del cuerpo se elige al ser menos propensa a oclusiones, por parte de los diversos elementos presentes en el escenario, que el resto de articulaciones. Por último, se utiliza un servidor para controlar el proceso global.

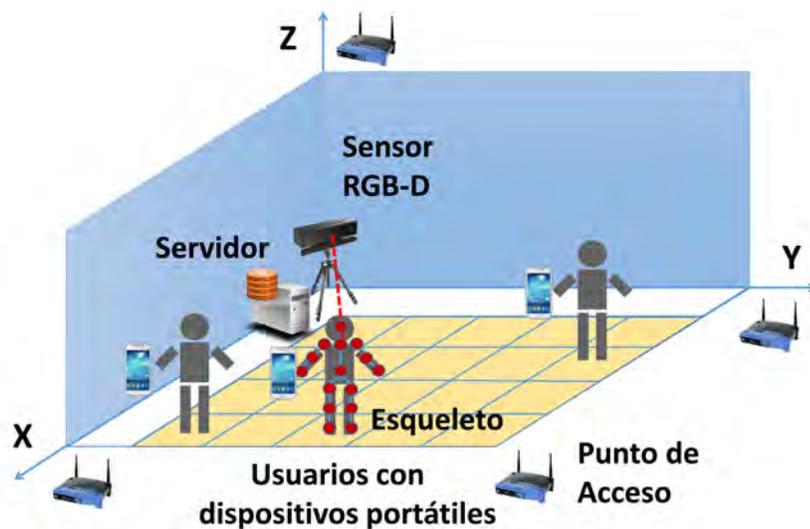


Figura 4.1: Componentes del sistema

4.1.1. Descripción del funcionamiento del sistema

El sistema desarrollado está dividido en dos etapas principales: *Aprendizaje* y *Ejecución*.

Etapas de Aprendizaje

El objetivo principal de esta etapa es crear, para el área de trabajo seleccionada, una base de datos con la información procesada procedente de las dos tecnologías: WPS y RGB-D. Durante esta etapa se crea el mapa de huellas, *fingerprint map*, asociado a un usuario. Para ello, se registran simultáneamente los valores RSSI obtenidos por su dispositivo portátil y las coordenadas de su esqueleto. El usuario se

mueve solo alrededor de la sala, con el fin de emparejar cada escaneo RSSI con cada posición de esqueleto. Esta tarea se realiza en tres pasos diferentes: *Escaneo WiFi*, *Escaneo RGB-D* y *Almacenamiento*.

Durante el *Escaneo WiFi*, el dispositivo portátil obtiene valores RSSI para cada AP (*Access Point*) y los envía al servidor. Cuando se reciben los valores RSSI, se inicia el *Escaneo RGB-D*. Este proceso devuelve el esqueleto de la persona detectada en la habitación. El sistema guarda automáticamente los datos RSSI y, además, las coordenadas del esqueleto del usuario. Estos datos representan el *fingerprint map* completo. La Figura 4.2 muestra el diagrama del sistema.

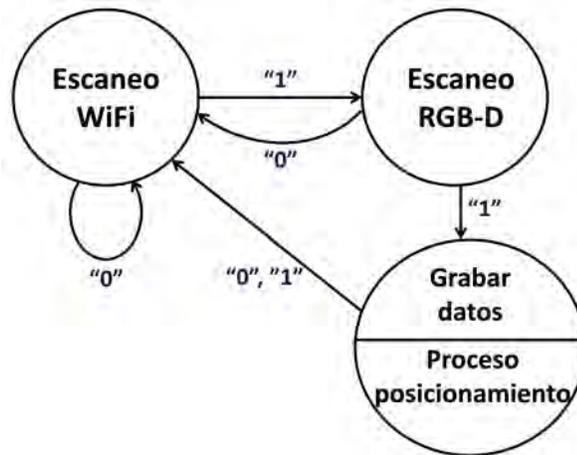


Figura 4.2: Diagrama del sistema durante los estados de Aprendizaje/Ejecución donde los valores 1 y 0 representan estado de ejecución satisfactorio/insatisfactorio respectivamente. “Guardar datos” es el tercer estado durante la fase de Aprendizaje mientras que “Proceso de posicionamiento” es el tercer paso durante la fase de Ejecución

Para simplificar el proceso de posicionamiento, sin pérdida significativa de precisión, se realizan otras tareas esenciales al final de esta etapa: el suelo del espacio de trabajo se divide en celdas rectangulares y los datos RSSI se agrupan en cada celda, utilizando la posición de celda del esqueleto.

La división en celdas se produce utilizando las coordenadas máximas y mínimas de x e y (ver Figura 4.1). Las coordenadas del esqueleto (x, y) proporcionan la celda donde se ubica el usuario (c_x, c_y) , de acuerdo con la Ecuación 4.1.

$$\begin{cases} c_x = n_x \cdot \frac{x - \text{mín}_x}{\text{máx}_x - \text{mín}_x} \\ c_y = n_y \cdot \frac{y - \text{mín}_y}{\text{máx}_y - \text{mín}_y} \end{cases} \quad (4.1)$$

Donde n_x y n_y representan el número de celdas en cada eje mientras que, las variables máx_x , mín_x , máx_y y mín_y representan los valores extremos de cada eje (obtenidos del *fingerprnt map*). Hay que tener en cuenta que las coordenadas en Z no se consideran, ya que la posición del usuario se estima en 2D, y Z representa la altura de la persona respecto al sensor.

Para cada una de las celdas, se crea un vector RSSI, emparejando cada componente con el centroide de todas las mediciones de RSSI para un determinado AP (como se muestra en el ejemplo de la Figura 4.3).

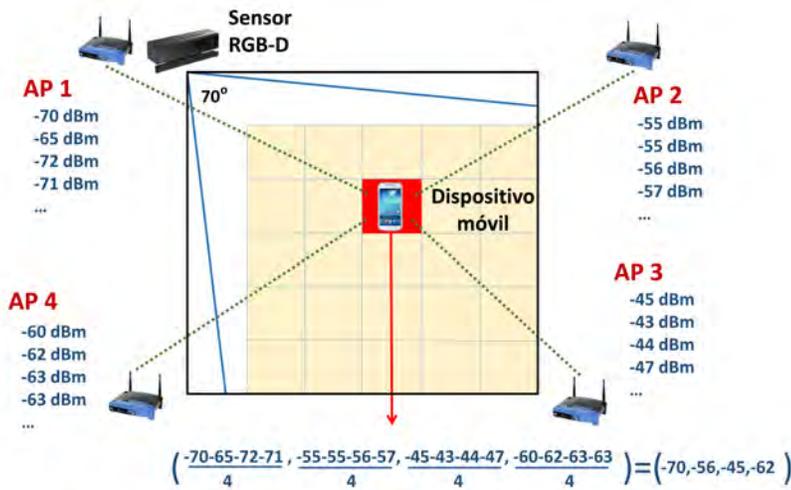


Figura 4.3: Obteniendo un vector RSSI para una celda (ejemplo con 4 APs)

Etapa de Ejecución

Esta etapa representa la forma normal de funcionamiento del sistema. Se realiza mediante los tres pasos diferentes que se muestran en la Figura 4.2, y considera que varios usuarios se están moviendo por la habitación.

Mientras se ejecuta el *Escaneo WiFi*, cada usuario envía de forma síncrona sus valores RSSI al servidor web. Cuando se reciben estos datos, el *Escaneo RGB-D*

comienza, con el objetivo de obtener los esqueletos de personas detectadas en la habitación. Por último, el proceso de posicionamiento calcula la posición de cada usuario combinando ambos datos, de tal manera que cada esqueleto se vincule a cada exploración RSSI.

En el proceso de posicionamiento se ejecutan diferentes algoritmos, dependiendo del modo de funcionamiento del sistema, pasando desde el modo básico más simple, en el que solo se aplica el método WPS, al más sofisticado, donde ambos tipos de sensores se combinan de modo que cada esqueleto se vincule a cada exploración RSSI.

El sistema almacena las diferentes mediciones RSSI recibidas desde el *Escaneo WiFi* en una tabla (ver Tabla 4.1), y las coordenadas del esqueleto obtenidas del *Escaneo RGB-D* en otra (ver Tabla 4.2). Nótese que los usuarios A y B de la Tabla 4.1 no están relacionados con los usuarios M y N de la Tabla 4.2. Durante el proceso de posicionamiento, el sistema podrá decidir si A corresponde a M o N y, de la misma manera, si B corresponde a M o N. Las coordenadas de esqueleto y los datos RSSI están emparejados por una marca de tiempo (*time stamp*).

<i>Time Stamp</i>	Usuario	BSSID	SSID	RSSI (dBm)
1 (8:31:57)	A	dc:53:7c:25:2c:36	Router 1	-53
1 (8:31:57)	A	84:9c:a6:fe:0e:34	Router 2	-46
1 (8:31:57)	A	4c:54:99:df:9e:ec	Router 3	-93
1 (8:31:57)	B	dc:53:7c:25:2c:36	Router 1	-74
1 (8:31:57)	B	84:9c:a6:fe:0e:34	Router 2	-40
1 (8:31:57)	B	4c:54:99:df:9e:ec	Router 3	-73
2 (8:31:58)	A	dc:53:7c:25:2c:36	Router 1	-52
2 (8:31:58)	A	84:9c:a6:fe:0e:34	Router 2	-51
2 (8:31:58)	A	4c:54:99:df:9e:ec	Router 3	-89
2 (8:31:58)	B	dc:53:7c:25:2c:36	Router 1	-73
2 (8:31:58)	B	84:9c:a6:fe:0e:34	Router 2	-41
2 (8:31:58)	B	4c:54:99:df:9e:ec	Router 3	-72

Tabla 4.1: Ejemplo de Tabla RSSI

<i>Time Stamp</i>	Usuario	X	Y	Z
1 (8:31:57)	M	-0.299	0.036	1.237
2 (8:31:58)	N	0.230	-0.025	1.739

Tabla 4.2: Ejemplo de coordenadas de esqueleto (en metros)

La estructura de los datos RSSI registrados contiene el Identificador de Conjunto de Servicio Básico, *Basic Service Set Identifier* (BSSID), el Identificador de Conjunto de Servicio, *Service Set Identifier* (SSID), el valor RSSI y la marca de tiempo (*time stamp*). BSSID está formado por el Control de Acceso a Medios, *Media Access Control* (MAC), de cada AP. SSID corresponde al nombre utilizado por los APs.

El BSSID se utiliza en lugar del SSID, ya que el SSID es informativo y diferentes APs pueden tener el mismo nombre en la red WLAN. Los datos RSSI, SSID y BSSID son recogidos por los dispositivos portátiles que utilizan la capa 802.11. Al mismo tiempo, los dispositivos portátiles deben establecer una conexión a alguna red accesible, como puede ser una red WiFi o una red de datos inalámbrica de telefonía (3G, 4G, etc.), con el objetivo de comunicar los datos RSSI. Los dispositivos envían datos, a través del protocolo SOAP y la capa de aplicación, a un servidor web. Este servidor web tiene acceso a la cámara RGB-D, aunque no necesariamente tiene que estar en la misma red que los dispositivos, ya que los servicios web que proporciona están disponibles en Internet, y son accedidos por los dispositivos móviles utilizando la conexión de red que hayan establecido.

Diferentes entradas de datos RSSI, así como de datos de esqueletos, se pueden producir de forma síncrona en el mismo *time stamp*, como se puede observar en la Tabla 4.1 y Tabla 4.2.

Como se mencionó anteriormente, se consideran tres modos de funcionamiento diferentes: *Modo Básico*, *Modo Mejorado sin Trayectoria* y *Modo Mejorado con Trayectoria*. Sus respectivas características se analizan en los siguientes párrafos.

Modo Básico: Solo WPS

En este modo, las mediciones RSSI se obtienen de dispositivos portátiles (es decir, teléfonos) y se comparan con los valores almacenados en la base de datos *fingerprint*. Para ello, se calcula un error basado en la distancia euclídea entre el vector RSSI medido y los vectores RSSI del centroide de cada celda. La celda WPS estimada es la que se corresponde con el error más bajo. La Ecuación 4.2 muestra cómo se obtiene este error a partir de dos vectores RSSI: el primero leído por el dispositivo portátil y el segundo correspondiente al centroide de cada celda. Cada

vector tiene n componentes correspondientes a cada AP. La variable $v_{p,\text{new}}$ representa el componente del vector para un AP_p donde se encuentra el usuario, mientras que v_{p,c_x,c_y} , representa el componente del vector centroide para ese AP_p en cada celda (c_x, c_y) .

$$e_{c_x,c_y} = \frac{1}{n} \cdot \sqrt{\sum_{p \in \text{AP set}} (v_{p,\text{new}} - v_{p,c_x,c_y})^2} \quad (4.2)$$

Modo Mejorado sin considerar Trayectoria de Usuario: combinando WPS con depth maps

En este modo, la información proporcionada por los *depth maps* ayuda a determinar en qué celda se encuentra el usuario. Además, es útil para aclarar su posición exacta. La combinación de ambos métodos mejora el posicionamiento en interiores de una manera sencilla.

Se estudian dos casos diferentes, dependiendo del número de usuarios dentro de la sala: si solo hay un usuario en la sala, el *depth map* permite obtener una correspondencia unívoca entre la posición del móvil y la del esqueleto. El dispositivo proporciona la identificación correcta del usuario.

Cuando hay dos o más usuarios, como se muestra en la Figura 4.4, se obtienen varios esqueletos. Al mismo tiempo, el móvil envía un grupo de mediciones RSSI al servidor. Inicialmente, el sistema no sabe qué esqueleto está vinculado a cada grupo de datos RSSI. El método propuesto calcula la distancia euclídea entre los datos RSSI, enviados por los teléfonos, y el centroide RSSI de la celda donde se encuentra cada esqueleto. Entonces, el sistema busca la mejor combinación entre cada esqueleto y las medidas de RSSI obtenidas de cada dispositivo portátil. Esto ocurre cuando la suma de las distancias euclidianas alcanza el mínimo, como se muestra en la Ecuación 4.3.

$$\min \sum_{i,j} s_{ij} \cdot d_{E_{ij}} \wedge s_{ij} \in \{0, 1\} \wedge \sum_x s_{xj} = 1 \wedge \sum_y s_{iy} = 1 \quad (4.3)$$

donde s_{ij} representa el emparejamiento entre cada esqueleto y un dispositivo móvil, y puede tomar los valores 0 y 1. $d_{E_{ij}}$ representa la distancia euclídea entre el esqueleto i y la posición del dispositivo j , calculada a partir del mapa WPS.

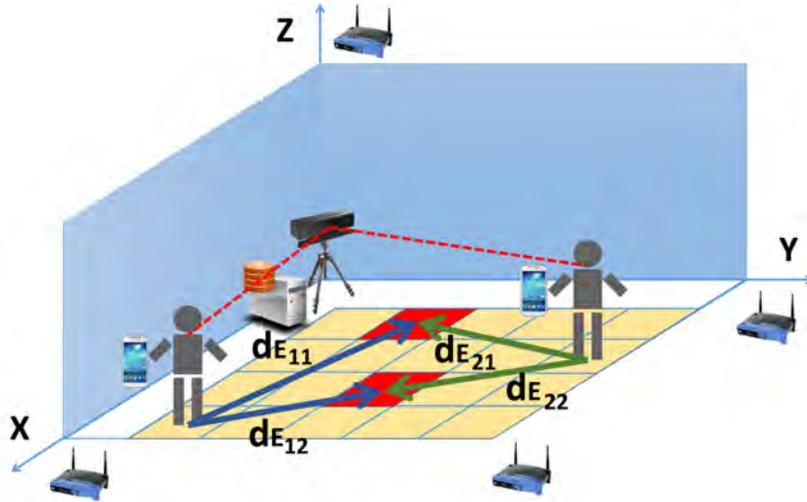


Figura 4.4: Esquema de dos personas en la habitación

Modo Mejorado con Trayectoria: considerando la trayectoria del usuario con WPS y depth maps

En este modo, la combinación de los *depth maps* y WPS permite también obtener dos trayectorias diferentes. La ruta del usuario con WPS representa las celdas que el usuario ha visitado previamente, según datos del sistema. La ruta del usuario, en el mapa de profundidad, está formada por un grupo de esqueletos, obtenidos cada uno de ellos en un instante de tiempo (*time stamp*) diferente. Ambas trayectorias (WPS y esqueletos) se sincronizan con sus marcas de tiempo, por lo que, cada instante de tiempo tiene una colección de esqueletos junto con otra colección de valores RSSI, de todos los usuarios.

Cuando dos o más usuarios están simultáneamente en la sala, el sistema inicialmente no sabe qué esqueleto está vinculado a cada dispositivo móvil. Sin embargo, puede calcularlo de acuerdo con una extensión de la Ecuación 4.3, como se explica a

continuación.

De acuerdo con [106], la distancia euclídea sincronizada, del inglés *Synchronized Euclidean Distance*, representa un método para calcular la distancia entre conjuntos de puntos en *time stamps* idénticos. Si se obtienen dos trayectorias (WPS y RGB-D), con puntos diferentes obtenidos en las mismas marcas de tiempo, el error total se mide como la suma de las distancias euclídeas entre cada par de puntos tomados en el mismo *time stamp*, como se muestra en la Ecuación 4.4.

$$e = \sum_{t=1}^m d_E(P(t), P'(t)) \quad (4.4)$$

La Figura 4.5 muestra las trayectorias WPS y de esqueleto de 2 usuarios en 4 *time stamps*, donde $P_i(t)$ representa la posición WPS del usuario i en el *time stamp* t , y $P'_j(t)$ representa la posición del esqueleto del usuario j en dicho *time stamp*. Aunque la Figura 4.5 representa la trayectoria del usuario 1 ($P_1(t)$, $P'_1(t)$) y la trayectoria del usuario 2 ($P_2(t)$, $P'_2(t)$), las trayectorias de cada usuario no están enlazadas. El sistema inicialmente no sabe si $P_1(t)$ está asociado a $P'_1(t)$ o $P'_2(t)$, ni tampoco si $P_2(t)$ está asociado a $P'_1(t)$ o $P'_2(t)$.

Para resolver este problema, se utiliza la Ecuación 4.5.

$$\min \sum_{i,j} s_{ij} \cdot \sum_{t=1}^m d_E(P_i(t), P'_j(t)) \wedge s_{ij} \in \{0, 1\} \wedge \sum_x s_{xj} = 1 \wedge \sum_y s_{iy} = 1 \quad (4.5)$$

La Expresión 4.5 tiene en cuenta la distancia euclídea sincronizada, calculando la suma de distancias entre cada par de puntos (trayectorias WPS y de esqueleto) y buscando por la mejor combinación entre las trayectorias WPS y las trayectorias de los esqueletos, que minimiza la suma de todas las distancias de todas las trayectorias. La Figura 4.5 muestra todas las distancias euclídeas sincronizadas, que son calculadas para 4 *time stamps*, donde hay dos usuarios con diferente trayectoria de esqueleto y WPS, respectivamente.

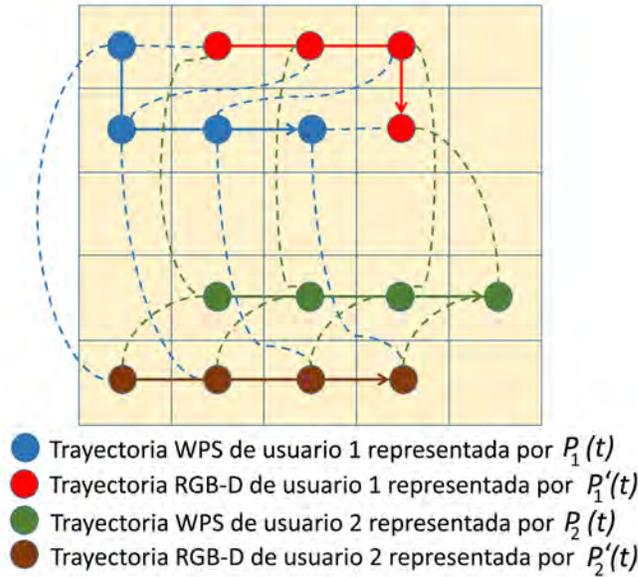


Figura 4.5: Ejemplo de trayectorias de $n = 2$ usuarios en $m = 4$ *time stamps*, incluyendo las distancias euclídeas entre la combinación de $P_i(t)$ y $P'_j(t)$

4.2. Posicionamiento WPS mediante RGB-D en entornos complejos

El objetivo principal del sistema, en este apartado, es identificar y estimar la posición de los usuarios en escenarios complejos habitados, donde varias personas navegan simultáneamente. Ambas tecnologías, WPS y cámaras de profundidad, están conectadas a un servidor central como se muestra en la Figura 4.6. Se sigue la estrategia presentada en la sección anterior, incluyendo mayor número de dispositivos RGB-D, y evaluando los movimientos de usuarios en un edificio completo. El WPS hace uso de varios APs (ej., *routers*) ubicados alrededor del entorno, que están al alcance de los teléfonos móviles de los usuarios. Los teléfonos miden el nivel de las señales recibidas y envían los datos a un servidor web. Simultáneamente, cada sensor RGB-D registra una parte del escenario, mediante una cámara y un sistema de proyección infrarroja, obteniendo mapas de profundidad. Dichos mapas se procesan para identificar los esqueletos de las personas y afinar los resultados obtenidos por el WPS, proporcionando valores más precisos para el posicionamiento. Finalmente, el

servidor central obtiene las posiciones de las personas y las devuelve a los teléfonos móviles.

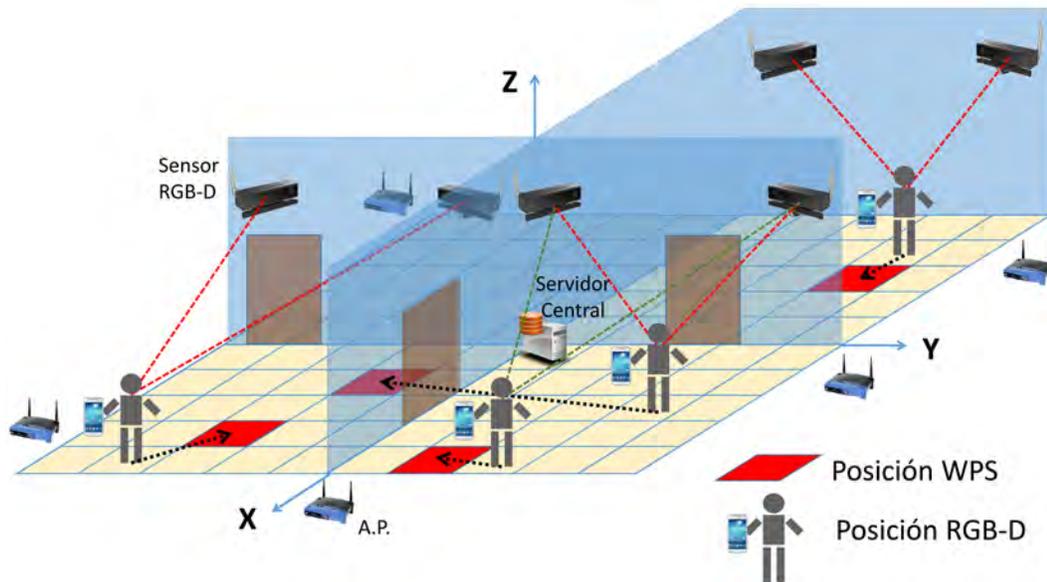


Figura 4.6: Escenario del sistema

El proceso de trabajo del sistema desarrollado se divide en dos fases principales: *Fase de Aprendizaje* y *Fase de Ejecución*, las cuales se detallan a continuación.

4.2.1. Fase de Aprendizaje

Durante la Fase de Aprendizaje, un usuario se mueve libremente en el entorno. La Figura 4.7 muestra, con un diagrama de actividades, los diferentes pasos durante el proceso. Los valores RSSI de los teléfonos, junto con su posición estimada por los sensores de profundidad, son grabadas en el *fingerprint map*.

A partir de los mapas de profundidad, se extraen las articulaciones del cuerpo y se consideran las coordenadas correspondientes al cuello (x_n, y_n, z_n) , como se muestra en la Tabla 4.3. Además, un Sistema de Coordenadas Universal (UCS, del inglés *Universal Coordinate System*) se ha definido para el escenario completo. Al igual que en el caso anterior (Sección 4.1), los esqueletos se obtienen mediante las técnicas presentadas en [91, 92].

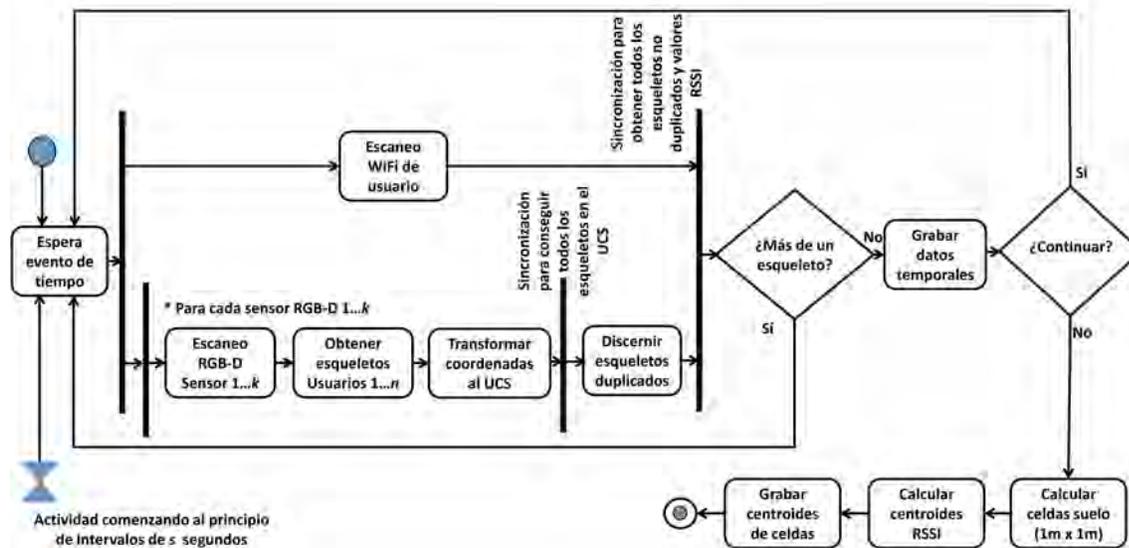


Figura 4.7: Diagrama de actividades durante la Fase de Aprendizaje

<i>Time Stamp</i>	RGB-D Sensor	x_n	y_n	z_n
1478460903017	E	-0.307	0.023	1.048
1478460975826	E	0.502	-0.022	1.895
1478460985055	E	0.476	-0.018	1.802
1478460993665	E	0.328	-0.014	1.702
1478461036995	E	0.063	0.007	1.400
1478461042276	E	0.069	0.008	1.384
1478461053002	E	-0.046	0.019	1.138
1478461204741	E	0.017	0.002	1.466
1478461213664	E	-0.003	0.008	1.361
1478461221343	E	-0.218	0.025	1.007
1478461231829	E	0.301	-0.016	1.837

Tabla 4.3: Ejemplo de coordenadas del cuello de un usuario obtenidas por un sensor RGB-D denominado E para diferentes *time stamps* (en metros)

Por otro lado, la estructura de los datos RSSI registrados contiene: el BSSID, el SSID, la intensidad RSSI y la marca de tiempo o *time stamp*. El BSSID está formado por el MAC de cada AP. Como se muestra en la Tabla 4.4, el SSID corresponde al nombre utilizado por los APs.

<i>Time Stamp</i>	BSSID	SSID	RSSI (dBm)
1478460903017	00:22:2d:02:70:8c	ROUTER_1	-49
	00:80:5a:47:13:a8	ROUTER_2	-59
	02:53:7c:25:2c:37	ROUTER_3	-69
	84:9c:a6:fe:0e:34	ROUTER_4	-50
	8c:0c:a3:45:38:5f	ROUTER_5	-85
	b2:46:fc:78:e1:a8	ROUTER_6	-89
	b4:75:0e:96:c3:9d	ROUTER_7	-88
	dc:53:7c:25:2c:36	ROUTER_8	-68
	f8:8e:85:29:21:12	ROUTER_9	-80
1478460975826	00:22:2d:02:70:8c	ROUTER_1	-49
	00:80:5a:47:13:a8	ROUTER_2	-75
	02:53:7c:25:2c:37	ROUTER_3	-55
	84:9c:a6:fe:0e:34	ROUTER_4	-49
	8c:0c:a3:45:38:5f	ROUTER_5	-86
	b2:46:fc:78:e1:a8	ROUTER_6	-81
	b4:75:0e:96:c3:9d	ROUTER_7	-89
	dc:53:7c:25:2c:36	ROUTER_8	-55
	f8:8e:85:29:21:12	ROUTER_9	-81

Tabla 4.4: Ejemplo de tabla RSSI por *Time Stamp*

Durante el escaneo WiFi, los dispositivos portátiles obtienen los valores RSSI para cada AP y envían los datos al servidor. Simultáneamente, se inicia el escaneo RGB-D, en cada uno de los k sensores diferentes. Este proceso devuelve las coordenadas, anteriormente mencionadas (x_n, y_n, z_n) , de la persona detectada en la sala. Estas coordenadas se transforman al UCS general. Si hay esqueletos duplicados, obtenidos por dos o más sensores RGB-D, no se consideran. Para determinar si los esqueletos están duplicados se considera la distancia euclídea entre las coordenadas de dos usuarios, obtenidas por diferentes sensores. Si la distancia está por debajo de un umbral (50 cm), solo se considera que un usuario está en esa posición. A continuación, el sistema guarda los datos de RSSI y la posición del usuario en el *fingerprint map*.

Cuando el usuario detiene el bucle de la *Fase de Aprendizaje* (ver Figura 4.7), con el objetivo de simplificar el proceso de posicionamiento, sin pérdida significativa de precisión, otra tarea esencial se lleva a cabo al final de esta etapa: el escenario completo se divide en celdas de 1 x 1 metro y, los datos RSSI son agrupados en cada celda, utilizando la posición del esqueleto. Se crea un vector RSSI para cada celda, donde cada componente representa el centroide de las mediciones RSSI de

un determinado AP. Los escaneos RSSI se agrupan de acuerdo con la distancia entre su esqueleto original asociado y las coordenadas del centro de cada celda. De esta manera se reduce el tamaño del *fingerprint map*, mejorando el rendimiento del sistema (ver Tabla 4.5).

X_{UCS}	Y_{UCS}	Valores RSSI	
		BSSID	RSSI (dBm)
3.5 metros	4.5 metros	50:7e:5d:43:15:ca	-73.00
		dc:53:7c:25:2c:36	-53.13
		f8:8e:85:29:21:12	-78.25
		00:80:5a:47:13:a8	-60.31
		02:53:7c:25:2c:37	-53.69
		b2:46:fc:78:e1:a8	-84.80
		b4:75:0e:96:c3:9d	-86.67
		00:22:2d:02:70:8c	-63.88
		8c:0c:a3:45:38:5f	-88.69
		f8:8e:85:6b:ee:6a	-62.31
		84:9c:a6:fe:0e:34	-46.94

Tabla 4.5: Ejemplo de valores para el *fingerprint map* reducido en la celda (3, 4)

Hay que tener en cuenta que, en esta etapa, solo un usuario está registrado en el *fingerprint map* por cada cámara RGB-D. Además, se almacenan los valores RSSI obtenidos por su dispositivo portátil y las coordenadas de su esqueleto. Sin embargo, si varios usuarios están involucrados en la creación del *fingerprint map*, el sistema registrará los datos de posicionamiento, para cada usuario, siempre que estén situados en salas diferentes. Esto implica que una cámara RGB-D no detectará más de un esqueleto y por lo tanto, no habrá conflicto entre esqueletos y valores de WPS. El sistema también podría ser entrenado usando marcadores para más de un usuario. Por ejemplo, podría identificar el esqueleto de cada usuario en función del color de la camisa que lleva. De una manera similar, en el trabajo [3] se rastrea a los trabajadores de la construcción considerando cascos de construcción, de diferentes colores, codificados como una clave para diferenciar al personal.

4.2.2. Fase de Ejecución

Esta etapa representa el modo de trabajo habitual del sistema. Como se muestra en la Figura 4.8, el proceso de obtener la posición de los usuarios se inicia cada s segundos. El sistema considera que n usuarios se mueven libremente alrededor del escenario y, de forma síncrona, envían sus valores RSSI al servidor web central. Simultáneamente, se ejecuta el proceso de escaneo RGB-D con el fin de obtener los esqueletos de personas detectadas en el escenario completo. Las coordenadas devueltas por los sensores RGB-D se transforman al UCS, y los esqueletos duplicados obtenidos por dos o más sensores RGB-D se descartan. Finalmente, después de m *time stamps*, si el número de trayectorias de WPS y el número de esqueletos coinciden (es decir, igual a n usuarios), el IPS estima la posición de cada usuario. En caso contrario, el sistema continúa obteniendo esqueletos y datos RSSI hasta que, coincida el número de trayectorias obtenidas por ambos métodos.

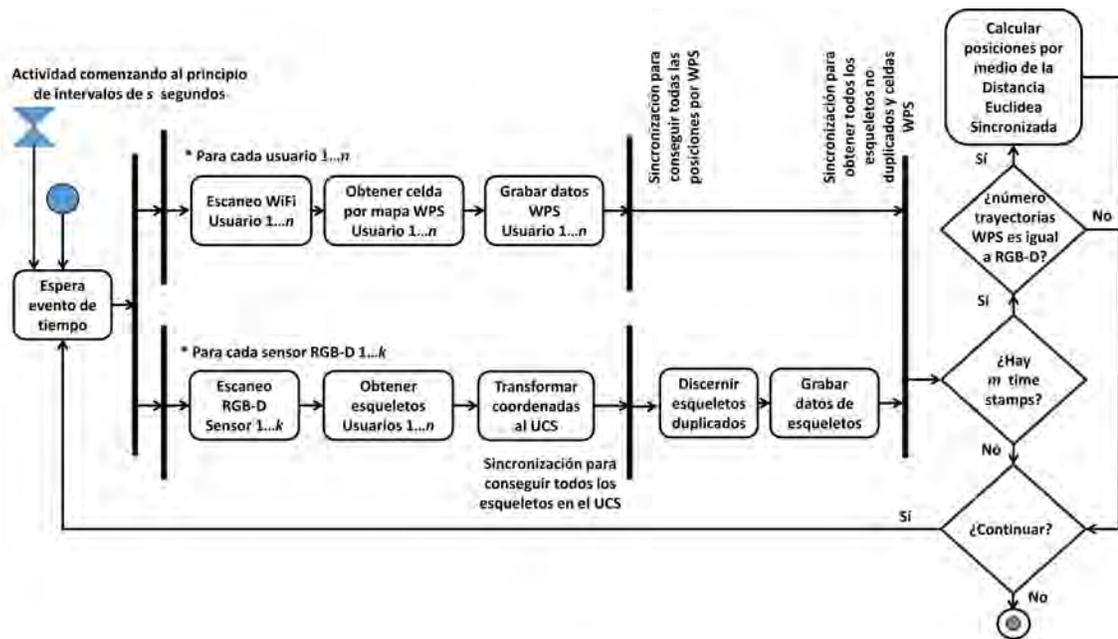


Figura 4.8: Diagrama de actividades durante la Fase de Ejecución

Como se mencionó anteriormente, cada usuario lleva un *smartphone* con una aplicación que envía de forma síncrona y periódica sus valores RSSI al servidor central, a través de un servicio web. Este servidor web aloja la base de datos completa,

registrando los valores RSSI. Al mismo tiempo, este servidor central crea varios subprocesos para llamar a diferentes servicios web, uno para cada sensor RGB-D, siguiendo el diagrama de la Figura 4.9. Además, cada dispositivo RGB-D está conectado a un servidor web diferente que envía los datos al servidor central. Finalmente, este servidor central calcula el resultado de la posición que combina resultados WPS y RGB-D para cada *time stamp*.

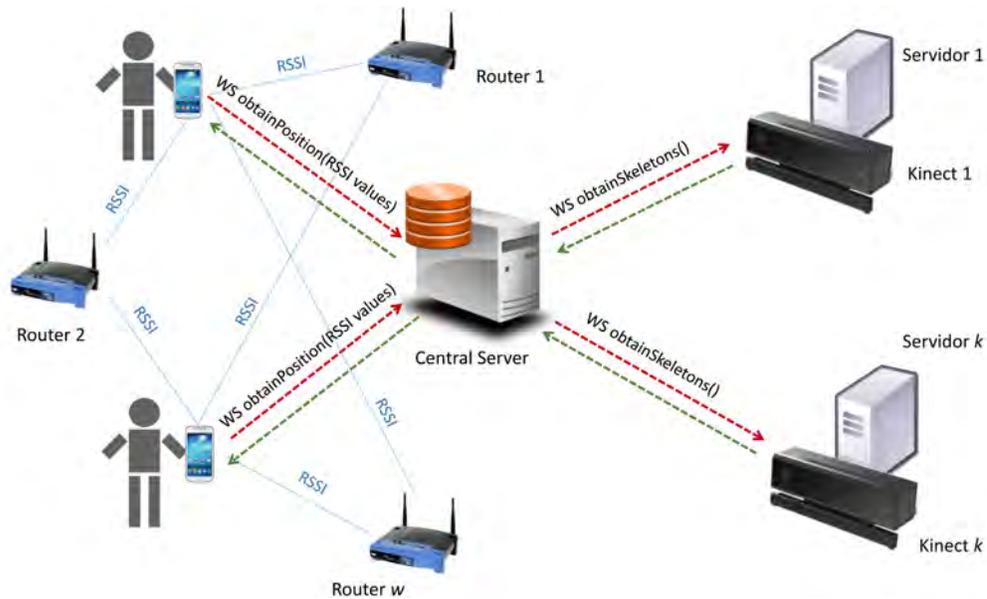


Figura 4.9: Esquema de las llamadas a *web services*

Hay que tener en cuenta que, la presencia de varias personas que se mueven en un entorno amplio implica requisitos adicionales. Por ejemplo, en algunos casos se rastrea la trayectoria de los usuarios por medio de más de un sensor RGB-D, obteniéndose un conjunto de coordenadas de usuario 3D para cada *time stamp*. Estos valores (es decir, coordenadas de esqueletos), que pertenecen a la misma persona, se sintetizan y transforman al mismo sistema de coordenadas, tal como se propone en [107]. En el caso de que la distancia euclídea entre las coordenadas de dos usuarios, obtenidas por diferentes sensores, esté por debajo de un umbral (50 cm ha sido considerado como el cilindro imaginario donde las personas encajan mientras están de pie), solo se considera que un usuario está en esa posición. Además, si un usuario es detectado por dos o más sensores RGB-D, todos ellos rastrean sus movimientos y,

por lo tanto, sus trayectorias. Esto es especialmente útil cuando las personas salen de una habitación y otro RGB-D identifica sus movimientos.

Algoritmo de emparejamiento de posiciones

Un aspecto importante del método propuesto es cómo asociar los datos de posicionamiento procedentes del WPS, con los datos de posicionamiento procedentes de los sensores RGB-D. Como se ha explicado anteriormente, existen datos, tanto de dispositivos WPS como RGB-D, correspondientes a varios usuarios que se han registrado al mismo tiempo. Es necesario, por tanto, combinar adecuadamente ambos tipos de datos para ofrecer información de posicionamiento correcta, y única, para cada usuario en cada *time stamp*. El algoritmo desarrollado para resolver este problema tiene en cuenta las siguientes consideraciones:

- Sea n el número de usuarios detectado y m el número de *time stamps* registrados en el experimento. Ambos sistemas de posicionamiento proporcionan sus correspondientes medidas de posición, para todos los usuarios en cada *time stamp*.
- Sea $P_i(t)$ la posición de un usuario i provista por el WPS en el *time stamp* t , y sea $P'_j(t)$ la posición del esqueleto de un usuario j provista por los sensores RGB-D en el *time stamp* t . Es importante tener en cuenta que la numeración de usuarios del WPS no es la misma que la de los sensores RGB-D, lo que significa que inicialmente $P_i(t)$ puede no estar asociado a $P'_i(t)$ sino a otro $P'_j(t)$. El problema consiste en determinar, para todos los *time stamp* considerados, cuál es el índice i asociado a cada índice j . La Figura 4.10 muestra un ejemplo de las trayectorias de tres usuarios diferentes con sus correspondientes caminos WPS y RGB-D para 6 *time stamps*.
- Sea s_{ij} el elemento binario de la *Matriz de Emparejamiento*, S , en la fila i , columna j . Dicho elemento empareja la trayectoria de un usuario i , provista por WPS, y la trayectoria de un usuario j , provista por el sistema de sensores RGB-D.
- La solución del algoritmo es encontrar la matriz S , válida para el conjunto completo de datos del experimento, tal que:

$$s_{ij} : i = 1 \dots n, j = 1 \dots n \wedge s_{ij} \in \{0, 1\} \wedge \sum_x s_{xj} = 1 \wedge \sum_y s_{iy} = 1$$

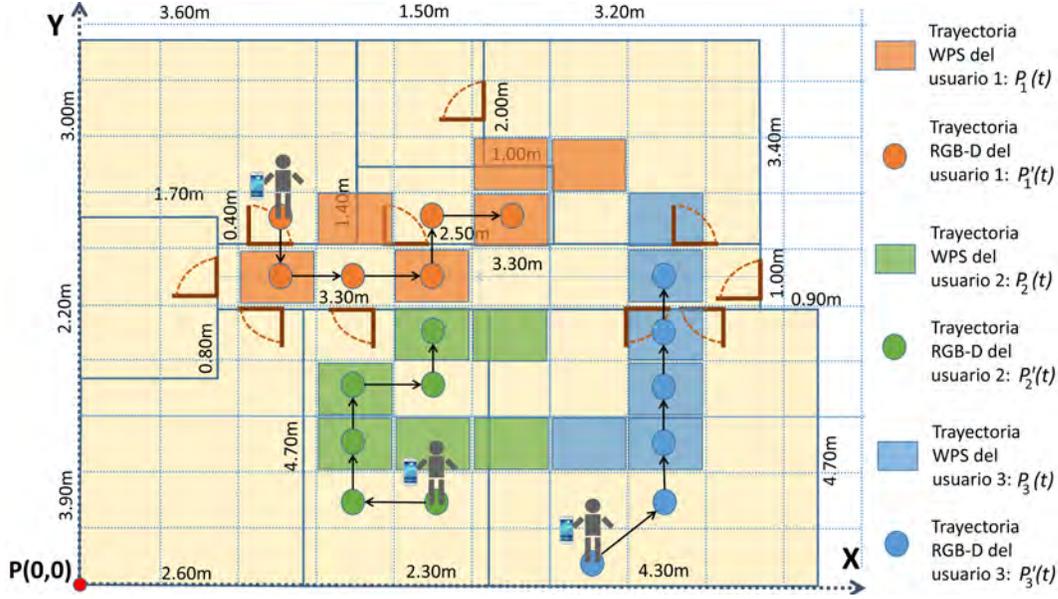


Figura 4.10: Celdas con trayectorias WPS y RGB-D durante la Fase de Ejecución

Para encontrar la matriz S , todos los m *time stamps* considerados deben ser tenidos en cuenta, de forma que el problema es una optimización de la combinación entre las trayectorias WPS y las de los esqueletos para todos los usuarios. La mejor combinación se define como la que consigue que la suma de distancias entre todas las trayectorias de usuarios sea mínima, como se muestra en la Ecuación 4.6.

$$\min \sum_{i,j} s_{ij} \cdot \sum_{t=1}^m d_E(P_i(t), P'_j(t)) \quad (4.6)$$

donde d_E representa la distancia euclídea entre un esqueleto y una celda WPS. Esta se mide como la distancia entre las coordenadas del cuello del esqueleto y el centroide de la celda WPS. Esta expresión (4.6) considera la distancia euclídea sincronizada [106], calculando la suma de distancias entre cada par de puntos WPS y de esquele-

to. El problema se resuelve utilizando programación lineal, y más concretamente el *Algoritmo Aditivo de Balas* [108]. Este método permite discernir qué trayectoria de un grupo es más similar a otra.

La solución permite relacionar cada usuario con sus posiciones de esqueleto en el UCS. Aunque algunas celdas estén situadas entre dos o más salas, el sistema obtiene la posición final de los usuarios según los sensores Kinect. Esto significa que, el sistema relaciona cada trayectoria de WPS con la trayectoria del esqueleto respectivo, pero usando las coordenadas de los esqueletos como las posiciones reales de los usuarios, y considerando la información WPS solo para obtener sus identificaciones.

Si dos o más sensores RGB-D detectan el mismo usuario, el usuario considerado estará situado en la misma posición del UCS (como se ha mencionado anteriormente, se considera un error de 0.5 metros para discernir dos usuarios). Una restricción del modelo implica que si un sensor RGB-D detecta z esqueletos, los resultados deben concluir que hay z usuarios en el rango de ese sensor. De lo contrario, el sistema podría detectar un usuario en la habitación adyacente, si su posición está cerca de la pared y esta restricción no se aplica. Esto ocurre porque el piso ha sido dividido en celdas en el UCS general, y algunas de estas celdas están, al mismo tiempo, en diferentes salas. Por último, el algoritmo considera que el número de trayectorias WPS debe ser igual al número de trayectorias de esqueletos.

Algoritmo de emparejamiento extendido

Una de las limitaciones del algoritmo presentado anteriormente es que, el número de usuarios detectados a través de las cámaras RGB-D debe ser similar al detectado utilizando WPS. En esta sección, se va a presentar una modificación del algoritmo para que éste sea válido en diversas situaciones, como puede ser el caso de personas no detectadas por los sensores RGB-D pero sí por WiFi, o personas detectadas por los sensores RGB-D pero que no portan teléfono móvil.

El algoritmo desarrollado para resolver este nuevo problema tiene en cuenta las siguientes consideraciones:

- Sea n_1 el número de usuarios detectado a través de WPS, y n_2 el número de usuarios detectados por las cámaras RGB-D.

- Sea m el número de *time stamps* registrados en el experimento. Ambos sistemas de posicionamiento proporcionan sus correspondientes medidas de posición para determinados usuarios en cada *time stamp* (para n_1 usuarios en WPS y n_2 usuarios en RGB-D).
- Sea $P_i(t)$ la posición de un usuario i provista por el WPS en el *time stamp* t , y sea $P'_j(t)$ la posición del esqueleto del usuario j provista por los sensores RGB-D en el *time stamp* t . Al igual que en el algoritmo anterior, hay que tener en cuenta que la numeración de usuarios de la WPS puede no ser la misma que la de los sensores RGB-D, lo que significa que inicialmente $P_i(t)$ puede no estar asociado a $P'_i(t)$ sino a otro $P'_j(t)$. El problema, también en este algoritmo, consiste en determinar para todos los *time stamp* considerados, cuál es el índice i asociado a cada índice j .
- Sea s_{ij} el elemento binario de la *Matriz de Emparejamiento*, S , en la fila i , columna j . Dicho elemento enlaza la trayectoria del usuario i , provista por WPS, y la trayectoria del usuario j , provista por el sistema de sensores RGB-D.
- La solución del algoritmo es encontrar la matriz S , válida para el conjunto completo de datos del experimento, tal que:

$$\text{si } n_1 > n_2, \left\{ \begin{array}{l} \forall j = 1, \dots, n_2 : \sum_{x=1}^{n_1} s_{xj} = 1 \\ \forall i = 1, \dots, n_1 : \sum_{y=1}^{n_2} s_{iy} \leq 1 \\ \sum_{x=1}^{n_1} \sum_{y=1}^{n_2} s_{xy} = n_2 \end{array} \right. \quad (4.7)$$

$$\text{si } n_1 = n_2 = n, \left\{ \begin{array}{l} \forall j = 1, \dots, n_2 : \sum_{x=1}^{n_1} s_{xj} = 1 \\ \forall i = 1, \dots, n_1 : \sum_{y=1}^{n_2} s_{iy} = 1 \\ \sum_{x=1}^{n_1} \sum_{y=1}^{n_2} s_{xy} = n \end{array} \right. \quad (4.8)$$

$$\text{si } n_2 > n_1, \begin{cases} \forall j = 1, \dots, n_2 : \sum_{x=1}^{n_1} s_{xj} \leq 1 \\ \forall i = 1, \dots, n_1 : \sum_{y=1}^{n_2} s_{iy} = 1 \\ \sum_{x=1}^{n_1} \sum_{y=1}^{n_2} s_{xy} = n_1 \end{cases} \quad (4.9)$$

El caso $n_1 = n_2 = n$ corresponde al algoritmo presentado anteriormente, ya que el número de usuarios de los dos sistemas es similar. Gráficamente, se puede ver en la Figura 4.11 una matriz S , donde $n_1 > n_2$. En la imagen se observa cómo los cálculos coinciden con las consideraciones indicadas.

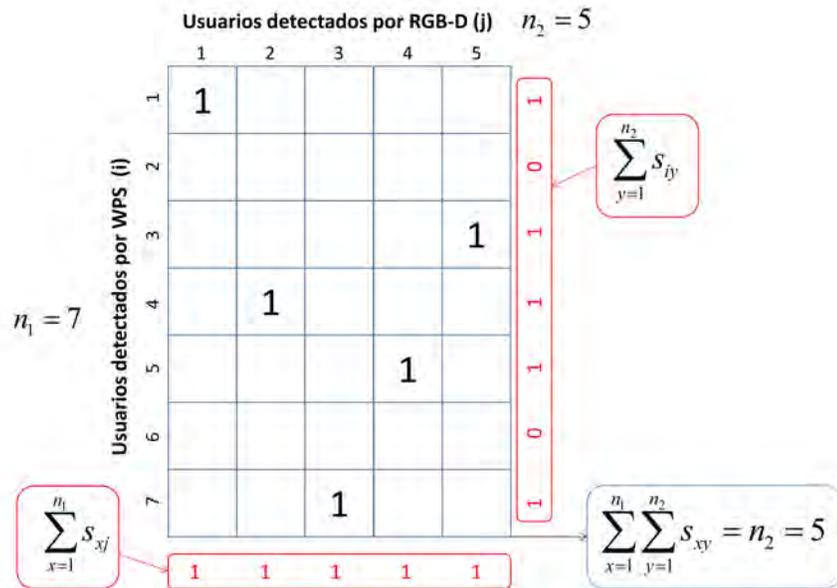


Figura 4.11: Ejemplo de matriz S para un distinto número de usuarios WPS y RGB-D.

Al igual que en el algoritmo anterior, para encontrar la matriz S , todos los m *time stamps* considerados deben ser tenidos en cuenta, de forma que el problema es una optimización de la combinación entre las trayectorias WPS y las de los esqueletos, para todos los usuarios. Al igual que en caso anterior, la mejor combinación es la que consigue que la suma de distancias entre todas las trayectorias de usuarios sea mínima, como se muestra de nuevo en la Ecuación 4.10, adaptada a este nuevo algoritmo.

$$\text{mín} \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} s_{ij} \cdot \sum_{t=1}^m d_E(P_i(t), P'_j(t)) \quad (4.10)$$

donde d_E representa la distancia euclídea entre un esqueleto y una celda WPS. Esta se mide como la distancia entre las coordenadas del cuello del esqueleto y el centroide de la celda WPS. El algoritmo nuevamente se resuelve utilizando el *Algoritmo Aditivo de Balas* [108].

Al haber un distinto número de usuarios detectados por WPS y RGB-D, algún usuario no será enlazado. Esto podría ocurrir, por ejemplo, si un usuario no lleva consigo un móvil pero es detectado por las cámaras. Como el sistema busca la combinación óptima, se minimizarán los posibles falsos positivos.

4.2.3. Transformación de coordenadas bajo un único UCS

Como se ha mencionado anteriormente, el sistema generalizado de posicionamiento requiere un sistema universal de coordenadas (UCS). Las coordenadas obtenidas por los sensores RGB-D de las diferentes salas se transforman en un único UCS, proceso que se realiza antes de obtener posiciones y trayectorias de los usuarios.

Las coordenadas de los esqueletos obtenidos representan en el eje X_{RGB-D} y Z_{RGB-D} la posición en el plano respecto al sensor, mientras que Y_{RGB-D} representa la altura (ver Figura 4.12).

Se considera un UCS único para todos los sensores. Por esta razón, todas las coordenadas obtenidas por los sensores RGB-D tienen que ser transformadas a ese UCS. Debido a que solo se consideran coordenadas 2D en este sistema de posicionamiento, el eje Y_{RGB-D} se ignora mientras que los ejes X_{RGB-D} y Z_{RGB-D} corresponden a los ejes X'_{RGB-D} e Y'_{RGB-D} por medio de una transformación, como se verá posteriormente en la Sección 5.4.2 del Capítulo 5. El eje Y_{RGB-D} se ignora porque representa la altura del usuario.

Para transformar las coordenadas obtenidas por cada cámara en el UCS, es necesario utilizar el ángulo de la cámara y la distancia al punto central universal $P(0, 0)$. En la Subsección 5.4.2 del Capítulo 5 se muestra el proceso de transformación de dichas coordenadas.

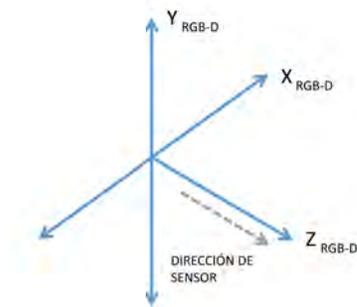


Figura 4.12: Los ejes X_{RGB-D} , Y_{RGB-D} , y Z_{RGB-D} respecto a la dirección del sensor RGB-D

4.3. Predicción de movimientos

Dado que la trayectoria del esqueleto se conoce, a partir de los sensores RGB-D, es posible predecir los movimientos del usuario. Esta predicción es útil para estimar hacia dónde va el usuario y, por lo tanto, que sea capaz de ejecutar diversas acciones, como: abrir o cerrar una puerta, encender o apagar una luz, o bien activar o desactivar el calentador, según las preferencias del usuario de una manera proactiva. Aunque los movimientos de las personas a veces parecen aleatorios, varias técnicas pueden ser utilizadas para estimar predicciones. Estos métodos se basan en la trayectoria del individuo, teniendo en cuenta los últimos puntos donde ha estado. Posibles técnicas a emplear incluyen la regresión [65] o el filtro de Kalman [66]. Otras utilizan métodos de aprendizaje automático [109], el conocimiento sobre los movimientos del usuario a través de redes neuronales [67, 110], modelos de probabilidad bayesianos [111, 112], modelos de Markov [113], o considerando patrones de movimiento [63, 64, 68].

La regresión es una técnica que permite obtener una función para aproximarse a los últimos puntos de una trayectoria. Siguiendo esta función, es posible predecir las posiciones siguientes. La trayectoria del usuario puede definirse como un conjunto de puntos $P(x, y)$, donde cada eje puede expresarse como una función de tiempo: $X(t)$ e $Y(t)$. La regresión se aproxima a cada función, aplicada para cada eje $X(t)$ e $Y(t)$, combinando los valores predictivos como un nuevo punto $P(X(t'), Y(t'))$, donde t' es la siguiente marca de tiempo.

La regresión lineal simple ajusta los datos a una función con la forma de la Ecuación 4.11, mientras que la regresión polinómica con grado $n = 2$ ajusta los datos a la Ecuación 4.12 y la regresión polinómica con el grado $n = 3$ se ajusta a la Ecuación 4.13.

$$y = a + bx \quad (4.11)$$

$$y = a + bx + cx^2 \quad (4.12)$$

$$y = a + bx + cx^2 + dx^3 \quad (4.13)$$

Por otro lado, el filtro de Kalman y el filtro extendido de Kalman [66] son técnicas que permiten estimar la siguiente ubicación de un objeto o persona a partir de su trayectoria. Este filtro considera el ruido y otras inexactitudes y ayuda a rectificar las señales, dando la posibilidad de obtener una predicción. El filtro de Kalman se implementa mediante un algoritmo recursivo de dos fases: durante la primera fase el algoritmo predice el estado del sistema, mientras que en la segunda fase utiliza las mediciones de ruido de cara a ajustar la estimación del estado del sistema. Para la estimación utiliza series temporales mediante estadística bayesiana [114] y calcula una distribución de probabilidad conjunta, sobre las variables, para cada marca de tiempo. El vector de estado inicial incluye las coordenadas del usuario y el desplazamiento entre marcas de tiempo consecutivas.

De entre los distintos métodos, las redes neuronales ofrecen buenos resultados en predicción de movimientos [115]. Son modelos computacionales que intentan imitar el comportamiento, observado en los axones de las neuronas, de los cerebros biológicos. Para ello, implementan un conjunto de neuronas, denominadas perceptrones, situadas en distintas capas y que se conectan entre sí a través de un conjunto de funciones de activación. El principal objetivo de las redes neuronales es clasificar un vector en una clase minimizando el posible error mediante un entrenamiento supervisado. A partir de un conjunto de valores de entrada, la red devuelve una salida válida para clasificar al grupo.

Las redes neuronales de propagación hacia atrás tienen una fase de aprendizaje, en la cual se configuran los distintos pesos y valores necesarios para la operación normal de la red, y una fase de clasificación, donde a partir de una entrada específica, se genera una salida concreta. Como se observa en la Figura 4.13, de una red neuronal de dos capas ocultas, cada perceptrón de las capas intermedias se conecta con todos los perceptrones de la capa anterior y todos los perceptrones de la capa siguiente.

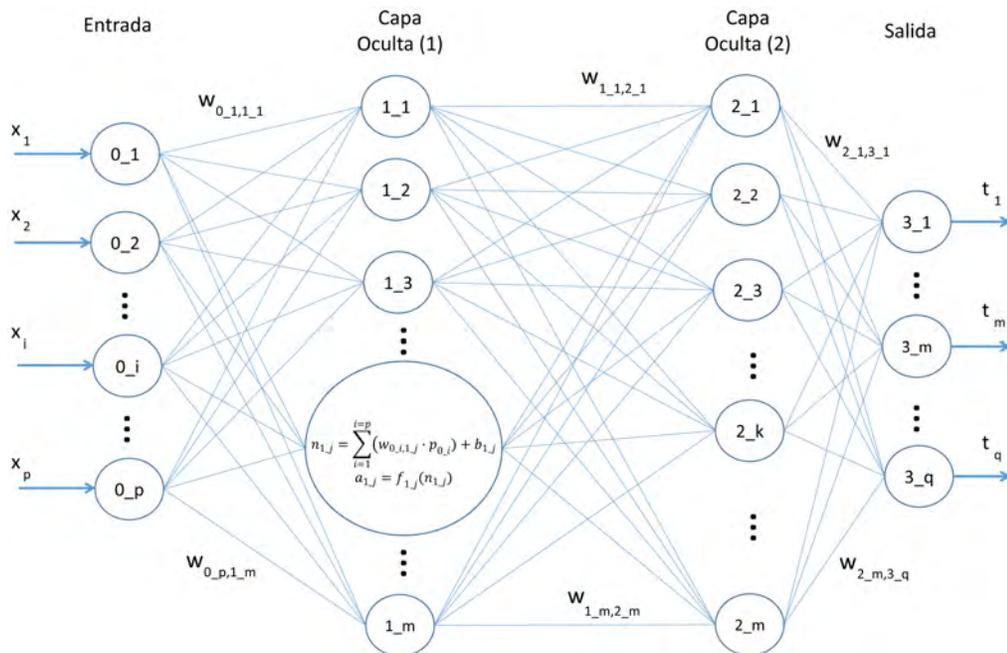


Figura 4.13: Red neuronal multicapa.

Durante la fase de aprendizaje, una vez que se ha aplicado un patrón a la entrada de la red, como estímulo, éste se propaga desde la primera capa a través de las capas superiores de la red, hasta generar una salida. La señal obtenida se compara con la esperada y se calcula una señal de error para cada una de las salidas. Las salidas de error se propagan hacia atrás, calculando el error asociado a cada perceptrón. Los perceptrones de las capas ocultas solo reciben una fracción de la señal total del error, calculada en función de la contribución relativa que haya aportado al error total. Una vez calculados los errores de cada perceptrón, se recalculan los pesos de la red. A medida que se entrena la red, los perceptrones de las capas intermedias se organizan entre sí, aprendiendo a reconocer distintas características del espacio total

de entrada. Después del entrenamiento, cuando se les presente un patrón arbitrario de entrada que contenga ruido o que esté incompleto, los perceptrones de las capas ocultas de la red responderán con una salida activa si, la nueva entrada contiene un patrón que se asemeje a aquella característica que los perceptrones individuales hayan aprendido a reconocer durante su entrenamiento.

Durante el proceso de propagación del estímulo de entrada, el valor de cada perceptrón se calcula a través de la Ecuación 4.14, donde x_i representa el valor de cada perceptrón de la capa anterior, w_{ij} el peso asociado entre el perceptrón i y el j , b representa un valor constante de ganancia asignado a cada perceptrón, y p el número total de perceptrones de la capa anterior. Al valor obtenido se le aplica una función de activación, representada en la Ecuación 4.15 por $f(n)$. La función de activación utilizada en esta tesis ha sido la sigmoidea, la cual hace que el perceptrón tome valores entre 0 y 1.

$$n_j = \sum_{i=1}^{i=p} (w_{ij} \cdot x_i) + b_j \quad (4.14)$$

$$a_j = f_j(n_j) = \frac{1}{1 + \exp(n_j)} = \frac{1}{1 + \exp\left(\sum_{i=1}^{i=p} (w_{ij} \cdot x_i) + b_j\right)} \quad (4.15)$$

El cálculo del error de cada perceptrón de salida se representa por la Ecuación 4.16, donde t_i representa la salida obtenida y o_i la salida esperada.

$$\delta_i = t_i - o_i \quad (4.16)$$

El error de cada perceptrón de las capas anteriores a la de salida, durante la fase de aprendizaje, se obtiene con la Ecuación 4.17, donde w_{ji} representa el peso entre el perceptrón calculado j y el perceptrón de la capa siguiente i , δ_i representa el error de dicho perceptrón, y q indica el número total de perceptrones de la siguiente capa.

$$\delta_j = \sum_{i=1}^{i=q} (w_{ji} \cdot \delta_i) \quad (4.17)$$

Finalmente, el recálculo de los pesos se realiza según la Ecuación 4.18, donde w_{ij} es el peso anterior, η es el coeficiente de velocidad de aprendizaje de la red neuronal, δ_j es el error calculado en el perceptrón j , $df(n)$ es la derivación de la función de activación sigmoidea, y finalmente x_i es el valor del perceptrón i .

$$w'_{ij} = w_{ij} + \eta \cdot \delta_j \cdot \frac{df(n)}{dn} x_i \quad (4.18)$$

4.3.1. Red neuronal para problema de posicionamiento

Para el problema de posicionamiento, se utiliza una red neuronal basada en un perceptrón multicapa (ver Figura 4.14), donde hay una capa de entrada con 6 neuronas, una capa oculta con 13 neuronas y una capa de salida con 2 neuronas. El suelo del escenario se divide en celdas, y la capa de entrada de la red representa los desplazamientos de celdas producidos en cada eje, entre dos marcas de tiempo consecutivas, pudiendo tomar valores 0 o 1. La capa de salida representa los desplazamientos entre las dos últimas posiciones durante el aprendizaje, y representa el desplazamiento entre la última posición y la predicción de la próxima ubicación, durante la fase de operación de la red.

Para el diseño de la red se utiliza una capa oculta, siguiendo la regla de la pirámide geométrica para el cálculo del número de neuronas de la capa oculta, donde se utiliza la Ecuación 4.19, siendo n el número de neuronas de entrada, m el número de neuronas de salida, y h el número de neuronas de la capa oculta. h se ha incrementado en una neurona por su mejora en los resultados. La red neuronal emplea una función de activación sigmoidea bipolar y la regla de aprendizaje se basa en la retropropagación con momentos [116].

$$h = m \cdot n \quad (4.19)$$

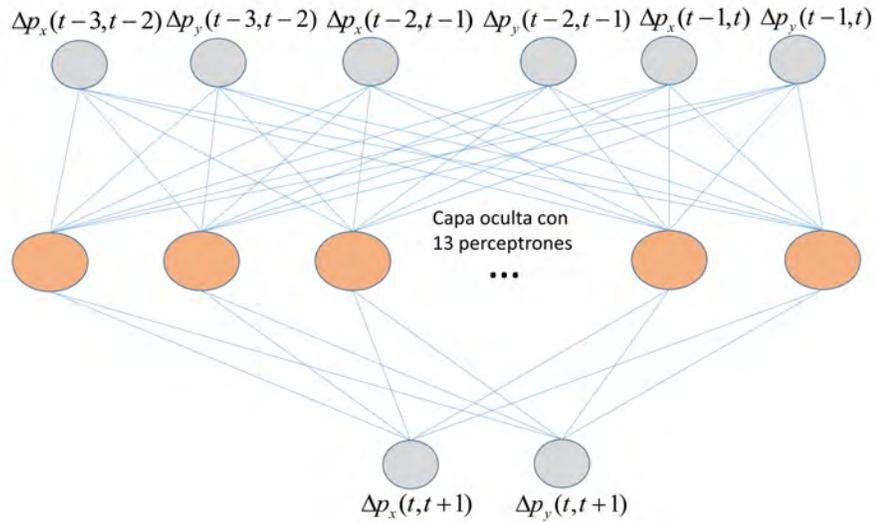


Figura 4.14: Red neuronal para la predicción de posicionamiento

Durante la etapa de aprendizaje, la capa de entrada de la red representa los desplazamientos entre las últimas posiciones de usuario, para cada eje X e Y . La capa de salida de la red representa el desplazamiento entre las dos últimas posiciones del usuario. Si se consideran cinco marcas de tiempo para el aprendizaje, las neuronas de entrada reciben: $\Delta p_x(t-4, t-3)$, $\Delta p_y(t-4, t-3)$, $\Delta p_x(t-3, t-2)$, $\Delta p_y(t-3, t-2)$, $\Delta p_x(t-2, t-1)$ y $\Delta p_y(t-2, t-1)$, mientras que las de salida son: $\Delta p_x(t-1, t)$ y $\Delta p_y(t-1, t)$. Durante la etapa de operación, la entrada de la red consiste en los desplazamientos entre las cuatro últimas posiciones de usuario (cuatro marcas de tiempo), mientras que la salida devuelve el desplazamiento previsto del usuario. Ese desplazamiento se añade a la última posición del usuario, para obtener la siguiente predicción de la posición del mismo.

Capítulo 5

Experimentación

En este capítulo se detallan los distintos experimentos realizados, cuyo objetivo es evaluar el comportamiento del sistema de posicionamiento desarrollado. En la Sección 5.1 se analiza la variabilidad de los datos RSSI, producidos por los teléfonos móviles, así como los datos de posición devueltos por el dispositivo Kinect. A continuación, en la Sección 5.2, se presentan los experimentos realizados con posicionamiento WPS, sin el uso de cámaras RGB-D. Después, se presentan los experimentos con el método de posicionamiento desarrollado, combinando datos WPS y Kinect, tanto para ubicación en una habitación (Sección 5.3), como para posicionamiento en entornos más complejos, formados por habitaciones adyacentes conectadas por un pasillo (Sección 5.4). Finalmente, se muestran los experimentos realizados en predicción de movimientos de personas (Sección 5.5).

5.1. Análisis de datos de posicionamiento

En primer lugar, se ha llevado a cabo un análisis que ha consistido en determinar las fluctuaciones de la medición RSSI, procedente de distintos puntos de acceso Wi-Fi, posicionándose el usuario en una única localización. Este análisis ha permitido comprobar la variabilidad de las muestras y ver si las mediciones RSSI son estables a lo largo del tiempo. Para ello, se ha desarrollado una aplicación móvil, con el objetivo de analizar cómo se comportan los datos RSSI obtenidos por un teléfono.

La aplicación obtiene, cada 2 segundos, los valores RSSI correspondientes a todos

los puntos de acceso WiFi. Con cada escaneo, u obtención de datos realizada, se hace una llamada a un servicio web alojado en un servidor, que graba los datos con el fin de analizarlos posteriormente, como se muestra en la Figura 5.1. Cuando se pulsa en el botón *Salir*, o se alcanza un número de mediciones determinado, la aplicación finaliza.

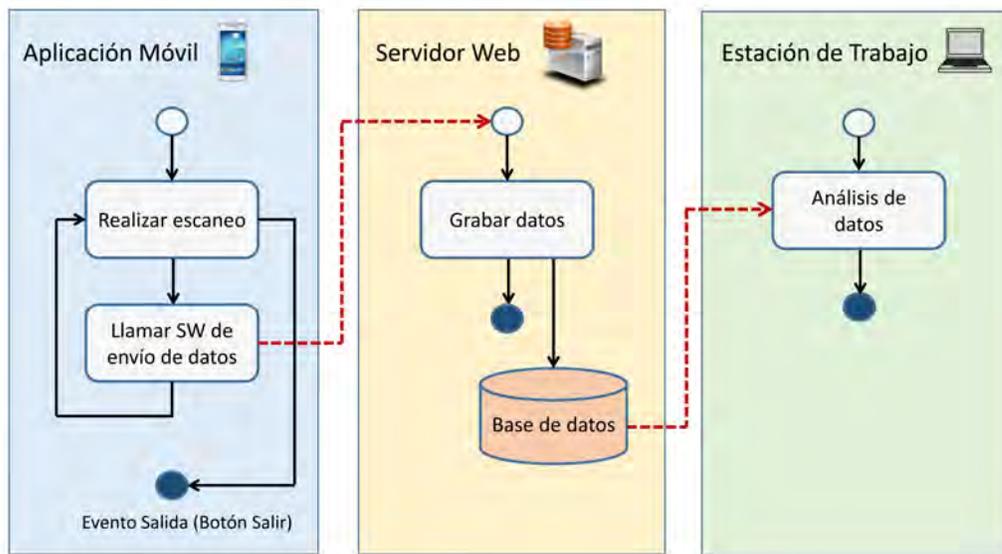


Figura 5.1: Esquema de aplicación desarrollada

Se han realizado 1000 medidas con el teléfono ubicado en una posición determinada, las cuales han producido una tabla de 11000 registros, almacenando en cada uno, un valor RSSI para un AP en un instante de tiempo. En cada medición se registran múltiples datos correspondientes a cada AP, y la señal de puntos de acceso lejanos puede no detectarse, por lo que si no hay señal para un AP, se deja el valor de intensidad en blanco en el campo de la tabla. La estructura de los registros incluye el BSSID, el SSID, el valor RSSI y la fecha. De entre los parámetros mencionados anteriormente, el BSSID se crea a partir del MAC (*Media Access Control*) de cada AP inalámbrico. Su tamaño es de 48 bits (6 bloques hexadecimales). Por otro lado, el SSID se corresponde al nombre utilizado por el propio AP (ver Tabla 5.1). De entre los valores registrados, el indicador de fuerza de la señal recibida, o RSSI, es una escala de referencia, en la cual un nivel de 0dBm es igual a 1 mW (milivatio),

que sirve para medir el nivel de potencia de las señal recibida por un dispositivo de un punto de acceso, como puede ser un *router* WiFi. La escala tiene el valor 0 como centro; representado como 0 RSSI, ó 0 dBm. Aunque teóricamente pueden medirse valores positivos, generalmente la escala se expresa dentro de valores negativos; cuanto más negativo es un valor, mayor es la pérdida de señal. Los valores RSSI pueden interpretarse como se muestra en la Tabla 5.2.

BSSID	SSID	Valor RSSI (dBm)	Fecha	Hora
dc:53:7c:25:2c:36	NUBE	-47	03/05/2015	07:56:02
02:53:7c:25:2c:37	_AUTO_ONOWiFi	-47	03/05/2015	07:56:02
3e:d9:54:28:4f:53	JAZZTEL_AF89	-67	03/05/2015	07:56:02
f8:8e:85:fd:fa:72	MOVISTAR_FA71	-87	03/05/2015	07:56:02
e0:91:53:5f:1e:85	WLAN_95	-84	03/05/2015	07:56:02

Tabla 5.1: Mediciones de registros RSSI

Señal RSSI	Calidad esperada	Descripción
0dBm	100 %	Señal ideal, difícil de lograr en la práctica. Un nivel de 0dBm es igual a 1 mW (milivatio).
-40dBm a -60dBm	100 % a 80 %	Señal idónea con tasas de transferencia estables.
-60dBm	80 %	Enlace bueno. Ajustando la potencia de transmisión (TX) y <i>basic rates</i> se puede lograr una conexión estable al 80 %.
-70dBm	60 %	Enlace normal-bajo. Es una señal medianamente buena, aunque se pueden sufrir problemas con lluvia y viento.
-80dBm	40 %	Es la señal mínima aceptable para establecer la conexión; pueden ocurrir caídas, que se traducen en corte de comunicación (pérdida de llamada, pérdida de datos, mensajes (SMS) corruptos o ilegibles).
-90dBm a -100dBm	20 % a 0 %	Es prácticamente imposible establecer conexión

Tabla 5.2: Interpretación de los valores RSSI

5.1.1. Datos RSSI de un punto de acceso próximo

El primer análisis ha consistido en el estudio de la señal RSSI recibida por el teléfono móvil, de un AP concreto cercano. En este caso, se ha empleado el punto de acceso denominado NUBE, un punto situado a 3 metros de distancia, en la misma habitación que el móvil. Al ser un AP próximo al móvil, se han obtenido valores RSSI altos para cada uno de los 1000 escaneos realizados. A partir de los datos RSSI obtenidos, se ha estimado una función de densidad de probabilidad, como se muestra en la Figura 5.2, en la que se agrupan el número de datos RSSI obtenidos, para cada señal, en una clase de un histograma, el cual permite distinguir claramente la concentración de valores tomados por cada intensidad medida.

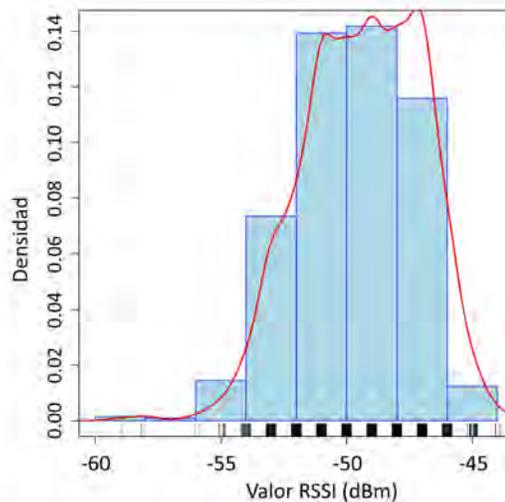


Figura 5.2: Estimación de la densidad de probabilidad RSSI (dBm) sobre NUBE en una localización concreta

A continuación, se ha obtenido la estimación de densidad de RSSI sobre una función de distribución normal, generada automáticamente. A efectos prácticos, la obtención de la distribución normal, según la Expresión 5.1, permite simular el comportamiento de la señal RSSI ante un único AP y desde una misma posición. Dicha distribución normal ha seguido la Expresión 5.2 en el caso concreto, con un valor medio calculado para las 1000 muestras de $\mu = 49.35$ dBm y una desviación estándar de $\sigma = 2.33$. Esta simulación de datos RSSI permite realizar un despliegue simulado de APs.

$$f_N(\mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2} \quad (5.1)$$

$$f_N = \frac{1}{2.33\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-(-49.35)}{2.33}\right)^2} \quad (5.2)$$

La Figura 5.3 muestra la estimación simulada de la densidad de probabilidad, donde se han generado distintos valores RSSI utilizando la distribución normal presentada. Comparada con la distribución real, los datos se agrupan de una manera más uniforme respecto a la media.

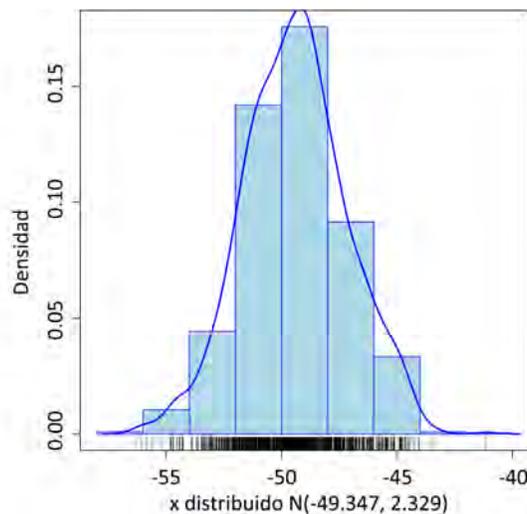


Figura 5.3: Estimación simulada de la densidad de probabilidad RSSI en una localización concreta para un *router* próximo

Las conclusiones que se pueden extraer de este análisis son:

- Sin tener en cuenta el ruido, cuanto más cerca se encuentra el móvil al punto de acceso WiFi, menor es el valor absoluto de la intensidad RSSI. Por ejemplo, un AP con un RSSI de -45dBm está más próximo al usuario que otro con -90dBm.
- En un punto de acceso próximo al móvil, los datos siguen una distribución normal, produciéndose pocos *outliers* o elementos atípicos.

- Debido a que la distribución de datos obtenidos puede aproximarse o ajustarse mediante una ecuación, es factible generar un simulador que represente la intensidad RSSI recibida por un dispositivo móvil.

5.1.2. Datos RSSI de un punto de acceso lejano

El segundo análisis ha consistido en el estudio de la señal RSSI recibida por el teléfono móvil, de un AP concreto lejano. En este caso, se ha empleado el punto de acceso denominado MOVISTAR_FA71, un punto situado a 25 metros de distancia, en otro piso diferente al del teléfono móvil. Para este AP también se han registrado 1000 valores RSSI, si bien su intensidad ha sido inferior a la obtenida para NUBE, debido a la distancia y a la existencia de muros. De forma similar al caso anterior, se ha estimado una función de densidad de probabilidad, como se muestra en la Figura 5.4, en la que se agrupan el número de datos RSSI obtenidos para cada señal en una clase del histograma. Para la generación del mencionado histograma, se han creado 10 clases, de las cuales 2 se representan a la izquierda del gráfico y muy alejadas del centro al reunir valores atípicos, al salirse de los datos habituales para un AP alejado según los valores reflejados en la Tabla 5.2.

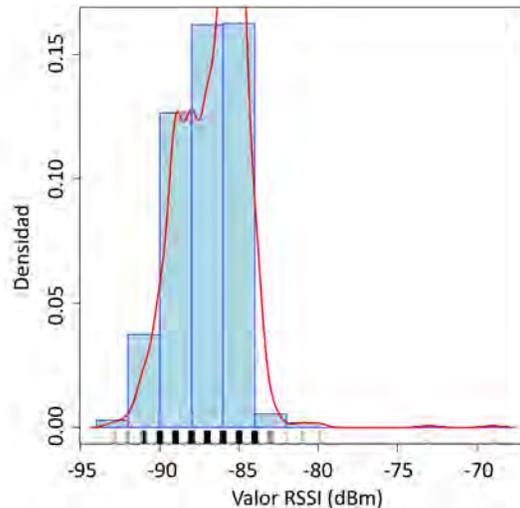


Figura 5.4: Estimación de la densidad RSSI (dBm) sobre MOVISTAR_FA71 en una localización concreta

El siguiente paso ha consistido en obtener la estimación de densidad de RSSI sobre

una función de distribución normal, generada automáticamente. Como en el caso anterior, la obtención de la distribución normal permite simular el comportamiento de la señal RSSI ante un único AP lejano y desde una misma posición. Dicha distribución normal ha seguido la Expresión 5.3, con un valor medio calculado para las 1000 muestras de $\mu = -86.63$ dBm y una desviación estándar de $\sigma = 2.07$. La Figura 5.5 muestra la estimación simulada de la densidad de probabilidad. Al igual que en el caso anterior, se han generado distintos valores RSSI utilizando la distribución normal presentada. Comparada con la distribución real, los datos se agrupan de una manera más uniforme respecto a la media.

$$f_N = \frac{1}{2.07\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-(-86.64)}{2.07}\right)^2} \quad (5.3)$$

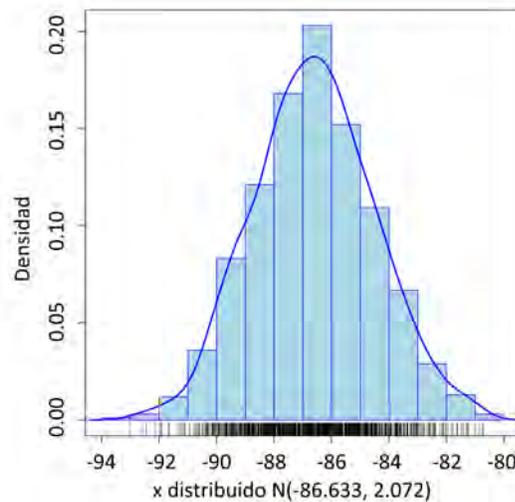


Figura 5.5: Estimación simulada de la densidad de probabilidad RSSI en una localización concreta para un *router* lejano

Las conclusiones que se pueden extraer de este análisis son:

- El comportamiento de un punto de acceso lejano también sigue una distribución normal, lo que permite simular su comportamiento. En función de qué tipo de AP se quiere simular, es necesario tener en cuenta la media y la desviación según Tabla 5.3.

Punto de Acceso	RSSI media (dBm)	Calidad	Distribución Normal Simulada
Próximo (3m)	-49.35	100 %	N(49.35, 2.33)
Media distancia (15m)	-68	64 %	N(68, 2.2)
Distante (25m)	-86.63	27 %	N(86.63, 2.07)

Tabla 5.3: Interpretación de los valores RSSI

- La dispersión entre los datos en un punto de acceso lejano es mayor que en uno cercano, como se muestra en la Figura 5.6, donde las cajas se representan entre el valor mínimo y máximo registrados de señal RSSI, la línea gruesa de cada caja representa la mediana de los datos, y los círculos alejados de la caja representan valores atípicos.

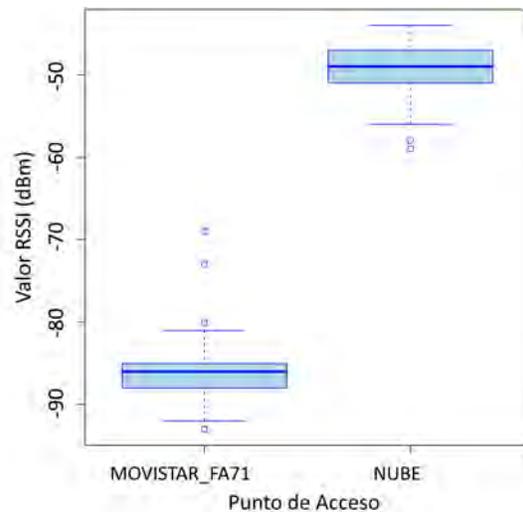


Figura 5.6: Diagrama de cajas de la distribución de los datos

5.1.3. Análisis de datos Kinect

De la misma manera que se ha realizado un análisis de la señal RSSI recibida por los teléfonos móviles, en esta sección se presenta el análisis realizado sobre los datos obtenidos de un sensor Kinect. El análisis ha consistido en determinar las fluctuaciones de la medición de las coordenadas, de un punto del esqueleto devueltas por el sensor Kinect, de una persona que se encontraba en una posición fija de la

sala. Este análisis ha permitido comprobar la variabilidad de las muestras y ver si dichas mediciones Kinect son estables en el tiempo.

Se han realizado 50 mediciones desde el mismo punto, las cuales se han almacenado en una tabla que incluye las tres coordenadas de posición, del cuello del esqueleto, generadas por la Kinect respecto al propio sensor (como se muestra en la Figura 5.7).

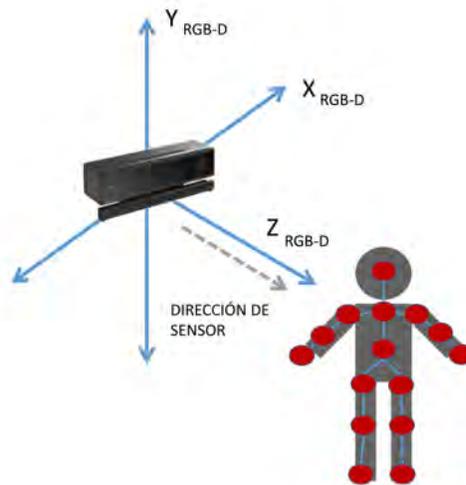


Figura 5.7: Sistema de coordenadas, tomando el sensor Kinect como origen

Como en casos anteriores, para el estudio de la intensidad de la señal RSSI, se ha procedido a obtener una estimación de la densidad de probabilidad de las mediciones. Para cada eje, se han agrupado las mediciones en distintas clases de un histograma, como se puede ver en la Figura 5.8.

Los datos analizados muestran que las mediciones obtenidas oscilan entre ± 3 centímetros, lo cual representa una variación aceptable dentro de las necesidades del sistema de posicionamiento desarrollado, donde los datos de dos esqueletos obtenidos por dos sensores diferentes se consideraban el mismo si la distancia en el UCS era inferior a 50 cm.

Se ha podido establecer que los datos obtenidos mediante el dispositivo Kinect siguen una distribución normal, donde se utiliza la Expresión 5.4 para el eje X , la Expresión 5.5 para el eje Y , y la Expresión 5.6 para el eje Z .

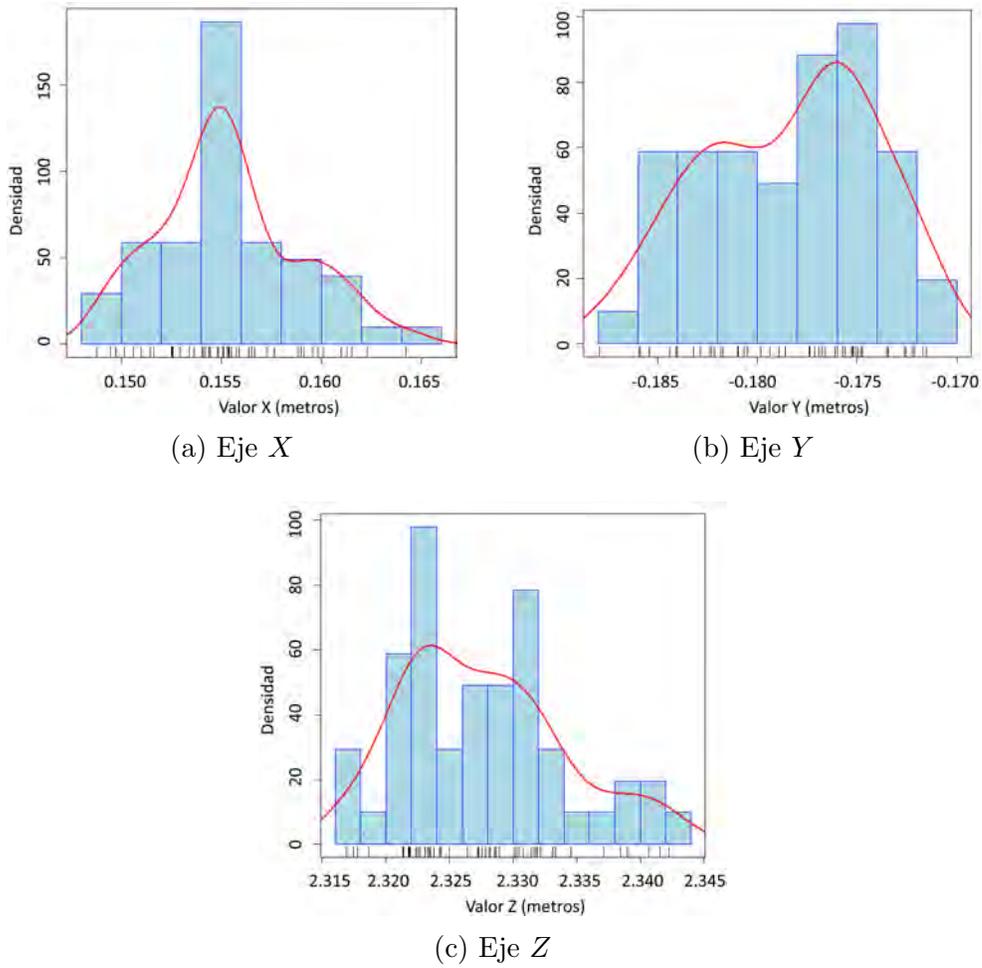


Figura 5.8: Estimación de la densidad de probabilidad de los 3 ejes Kinect

$$f_{XN} = \frac{1}{0.004\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-0.155}{0.004}\right)^2} \quad (5.4)$$

$$f_{YN} = \frac{1}{0.004\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x+0.178}{0.004}\right)^2} \quad (5.5)$$

$$f_{ZN} = \frac{1}{0.006\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-2.328}{0.006}\right)^2} \quad (5.6)$$

Las Figura 5.9 muestra las 3 distribuciones de probabilidad.

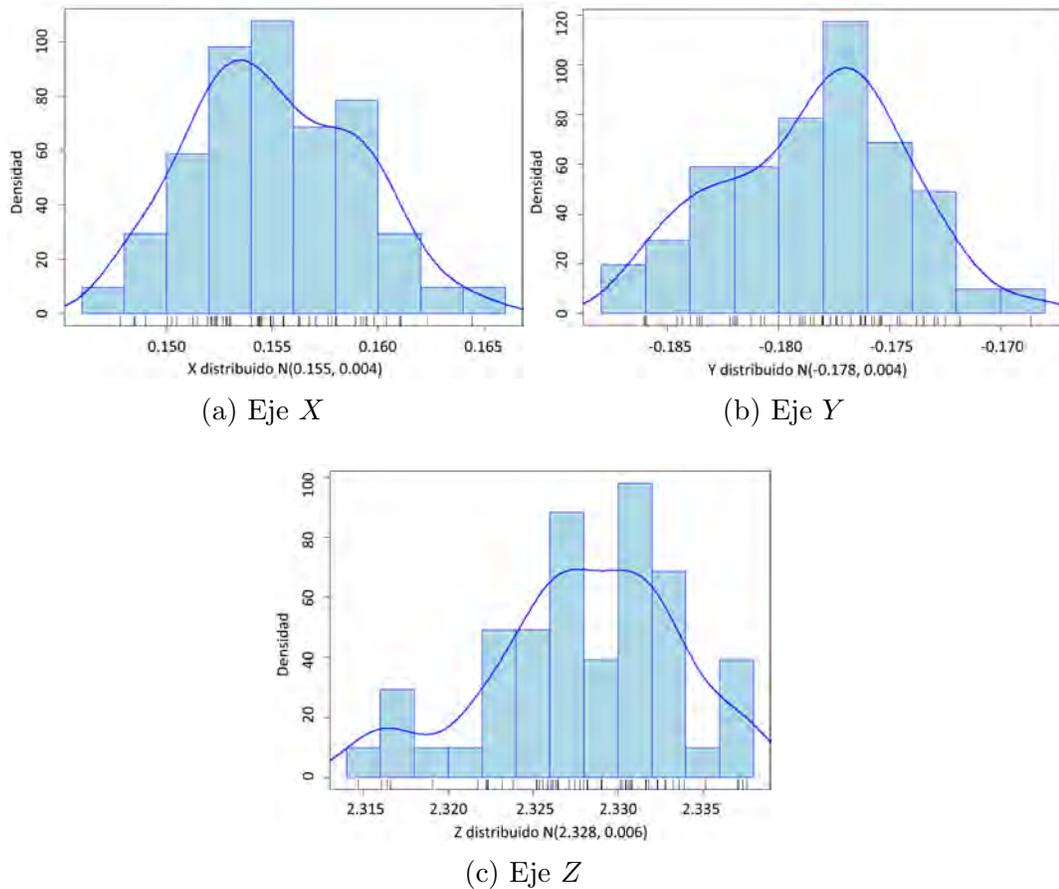


Figura 5.9: Estimación simulada de la densidad de probabilidad de los 3 ejes Kinect en una posición concreta

5.2. Experimentación de posicionamiento con WPS

Para evaluar el posicionamiento de personas mediante WPS, se han realizado dos experimentos: el primero ha consistido en determinar mediante WPS la habitación

donde se encontraba una persona con su teléfono móvil; el segundo ha consistido en determinar, con ayuda del sensor Kinect, la celda virtual en la que se encontraba la persona dentro de la habitación. Estos experimentos han permitido evaluar el *Modo Básico* de posicionamiento, explicado en la Sección 4.1 del capítulo anterior.

5.2.1. Determinación de la habitación donde se encuentra la persona

En un escenario compuesto por 9 habitaciones, según el esquema de la Figura 5.10, se ha realizado un experimento para detectar la habitación en la que se encontraba una determinada persona. Para ello, se ha realizado un mapa *fingerprint* de mediciones RSSI.

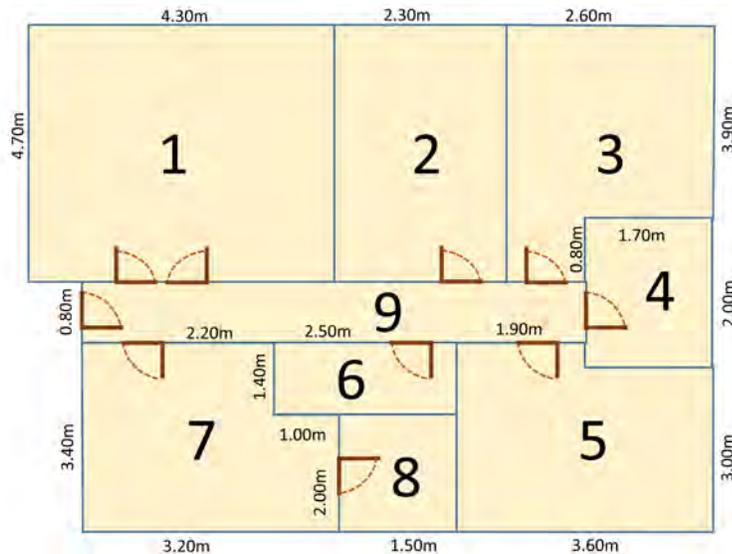


Figura 5.10: Esquema de entorno para detección de habitación

En total, se han tomado 525 mediciones, repartidas de manera uniforme entre las distintas salas. Una persona se ha desplazado por distintos puntos de las habitaciones, utilizando una aplicación móvil que le ha permitido introducir manualmente el número de habitación y obtener los valores RSSI. Cada vez que la persona ha pulsado en el botón de *Realizar escaneo*, se han grabado las mediciones RSSI obtenidas y el número de habitación en una base de datos, alojada en el propio teléfono.

Una vez obtenidos los datos del mapa, se han generado 47 mediciones para probar el posicionamiento, realizadas en distintos puntos de cada habitación. Estas mediciones se han obtenido utilizando la misma aplicación.

La técnica utilizada calcula la distancia euclídea entre los valores RSSI obtenidos en una posición y los valores RSSI obtenidos en las distintas posiciones del mapa. Se han probado dos métodos: búsqueda de la posición más cercana entre todas las almacenadas en el mapa, y comparativa de una medición con los centroides de cada sala. Además, en el caso de la comparativa con todas las posiciones del mapa, se ha considerado el caso de APs con una intensidad de señal RSSI superior a un valor dado (-88 dBm).

Los resultados obtenidos se pueden ver en la Figura 5.11. El mejor comportamiento se ha producido cuando se han considerado solo los AP, con una intensidad de señal superior a -88dBm, y la detección se ha realizado buscando la medición más cercana entre todas las almacenadas en el mapa *fingerprint*. El resultado indica que se detecta correctamente la sala donde se encuentra el usuario en un 87.23% de los casos. De entre las distintas pruebas, el caso que ofrece peores resultados ha sido el de la habitación 9, ya que como se aprecia en la Figura 5.10, la sala se distribuye a lo largo de todo el escenario y sus señales RSSI pueden ser, en determinados puntos, parecidas a las de otras habitaciones contiguas.

Cabe destacar que la utilización del centroide de la sala no ofrece resultados satisfactorios, ya que debido a la forma de las habitaciones, es posible que un grupo de señales se encuentre más cerca del centroide de otra sala diferente. Para ubicar al usuario de una forma más precisa, el escenario se ha dividido en celdas como se muestra en la siguiente sección.

Estancia donde se realiza medición	Estancia estimada por WPS (mapa completo)	Distancia euclídea (mapa completo)	Estancia estimada por WPS (centroide)	Distancia euclídea (centroide)	Estancia estimada por WPS (mapa completo), >88dBm	Distancia euclídea (mapa completo) >88dBm	%detección correcta/ estancia (mapa completo)	%detección correcta/ estancia (centroide)	%detección correctas/ estancia (mapa completo) >88dBm	%detección correcta (mapa completo)	%detección correctas (centroide)	%detección correcta (mapa completo) >88dBm
1	1	15.32	1	21.10	1	11.18						
1	1	15.77	1	14.86	1	13.67						
1	1	18.62	1	18.85	1	13.37						
1	1	13.67	1	16.72	1	13.07	100%	100%	100%			
1	1	15.45	1	16.87	1	15.0						
1	1	13.63	1	16.62	1	11.74						
2	2	18.57	2	13.16	2	9.43						
2	2	56.78	2	13.72	3	7.68	75%	75%	75%			
2	3	12.52	3	15.34	2	6.55						
2	2	10.24	2	14.81	2	10.19						
3	3	58.05	1	43.51	3	11.35						
3	3	58.40	1	48.38	3	13.85	100%	50%	100%			
3	3	8.12	3	11.52	3	7.34						
3	3	7.34	3	13.44	3	4.47						
4	8	41.01	5	28.65	9	11.18						
4	5	21.37	5	26.21	4	14.76						
4	4	7.14	4	13.18	4	6.85	60%	60%	80%			
4	4	8.60	4	10.28	4	5.29						
4	4	12.04	4	14.37	4	8.71						
5	5	101.78	5	71.53	5	9.05						
5	5	101.85	5	71.51	5	10.81	100%	100%	100%			
5	5	57.89	5	38.22	5	8.88						
5	5	60.03	5	39.28	5	10.81						
6	6	12.24	6	18.05	6	9.79				85%	68%	87%
6	6	11.61	6	8.64	6	9.32						
6	6	10.29	6	10.85	6	9.16	100%	100%	100%			
6	6	13.11	6	8.86	6	7.74						
6	6	10.34	6	8.92	6	8.36						
6	6	9.48	6	8.35	6	6.24						
7	7	17.63	7	19.19	7	16.37						
7	7	56.72	5	34.33	7	10.48						
7	7	16.30	5	37.08	7	10.04	100%	17%	100%			
7	7	58.77	5	38.09	7	7.61						
7	7	61.21	5	64.68	7	11.04						
7	7	10.09	6	58.42	7	8.0						
8	8	99.26	8	64.14	8	10.90						
8	8	82.61	8	28.18	8	10.77	100%	100%	100%			
8	8	59.53	8	20.66	8	13.74						
8	8	18.94	8	20.11	8	13.0						
9	2	9.11	2	13.83	3	7.14						
9	9	11.18	2	22.45	9	7.93						
9	9	7.61	9	12.58	9	4.0						
9	6	12.80	6	17.17	6	12.80	50%	38%	50%			
9	6	8.66	6	13.53	6	7.68						
9	9	9.21	9	5.86	9	9.21						
9	9	12.76	9	14.53	9	12.76						
9	4	11.31	4	17.14	4	9.94						

Figura 5.11: Porcentaje de éxito en la detección de la habitación donde se encuentra el usuario

5.2.2. Determinación de la celda donde se encuentra una persona

Para evaluar el comportamiento de WPS *fingerprinting* dentro de una habitación, dividida en celdas, se ha utilizado el sensor Kinect para obtener la posición de la persona y compararla con la obtenida mediante WPS. Como se muestra en la Figura 5.12, el suelo se ha dividido en celdas virtuales que han agrupado los valores RSSI, obtenidos cuando la persona se encontraba dentro de ellas. Se han creado 10x10 celdas de 0.37m de lado, que permiten ubicar a una persona sobre una celda concreta. Durante el experimento, 8 APs han estado accesibles.

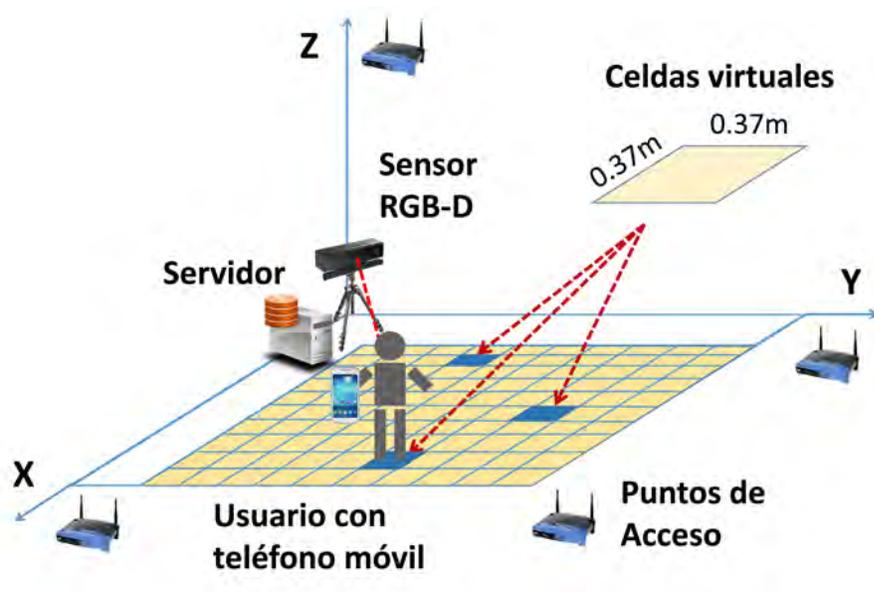


Figura 5.12: División del suelo de la habitación en celdas virtuales

En este experimento, como se muestra en la Figura 5.13, se ha desarrollado una aplicación que enviaba los valores de posicionamiento móvil RSSI a un servidor web, cada vez que el usuario pulsaba un botón de escaneo de datos. El servidor web ha estado conectado al sensor Kinect, y cada vez que recibía datos RSSI, obtenía los valores de posicionamiento devueltos por la Kinect, siendo estos valores la posición tridimensional del cuello de la persona.

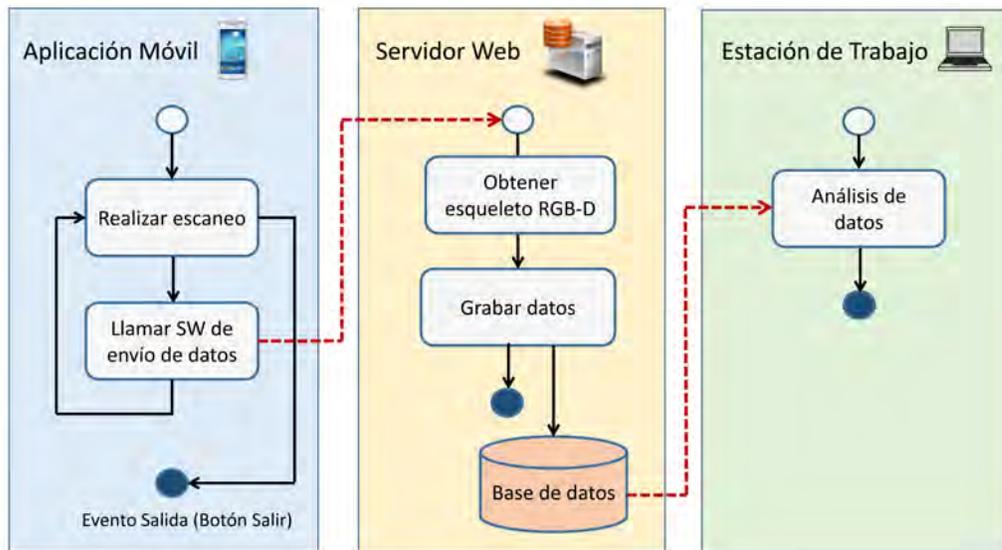


Figura 5.13: Esquema de aplicación desarrollada para determinación de la celda donde se encontraba una persona

La habitación donde se ha desarrollado el experimento tiene unas dimensiones de 4.7 x 4.3 metros. El campo de visión horizontal del sensor Kinect v2 es 70°, por lo que, como se muestra en la Figura 5.14, solo es capaz de detectar a las personas en una sección de la habitación. Esta sección tiene un tamaño de 3.71 x 3.71 metros. La persona bajo estudio, se ha movido por la habitación, registrando los distintos valores RSSI y RGB-D, y desplazándose desde puntos cercanos a la Kinect hasta los puntos más alejados del sensor en la habitación. A continuación, se han obtenido los valores máximos y mínimos de cada coordenada del cuello.

Para el análisis de datos, se han utilizado las mediciones de un único esqueleto. De las tres coordenadas calculadas por la cámara RGB-D, se ha prescindido de y , que corresponde a la altura del cuello respecto a la Kinect. x y z representan las coordenadas en el plano horizontal XZ .

Durante una primera fase del experimento, se han grabado las mediciones correspondientes a los valores RSSI en una tabla, como se muestra a modo de ejemplo en la Tabla 5.4, y las mediciones correspondientes a las lecturas Kinect en otra, como se puede ver en la Tabla 5.5. Las dos tablas se cruzan por el número de medición, que se crea cada vez que el usuario ha pulsado en el botón de escaneo.

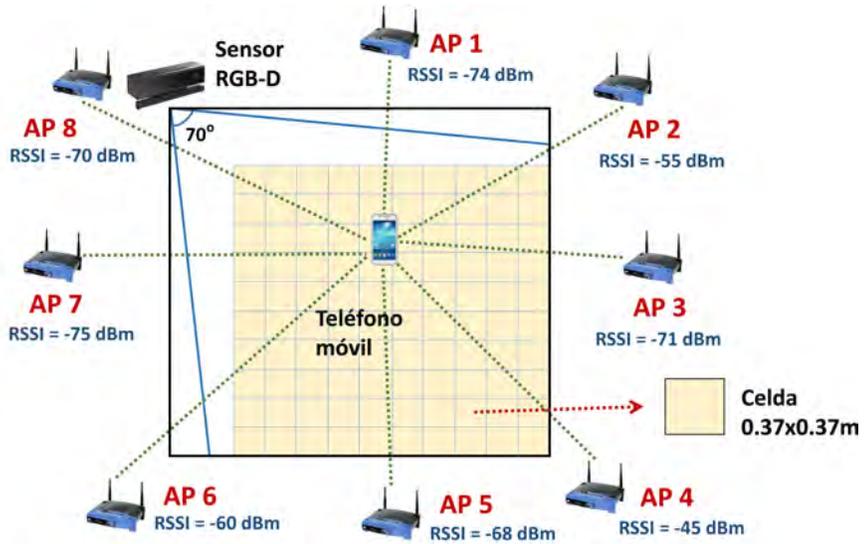


Figura 5.14: Esquema de la habitación donde se han realizado los experimentos

Medición	BSSID	SSID	Valor RSSI (dBm)	Fecha	Hora
1	dc:53:7c:25:2c:36	NUBE	-49	12/05/2015	12:35:37
1	02:53:7c:25:2c:37	_AUTO_ONOWiFi	-49	12/05/2015	12:35:37
1	3e:d9:54:28:4f:53	JAZZTEL_AF89	-69	12/05/2015	12:35:37
1	8c:0c:a3:35:6b:19	MOVISTAR_6B19	-85	12/05/2015	12:35:37
2	dc:53:7c:25:2c:36	NUBE	-45	12/05/2015	12:35:41
2	02:53:7c:25:2c:37	_AUTO_ONOWiFi	-45	12/05/2015	12:35:41
2	3e:d9:54:28:4f:53	JAZZTEL_AF89	-64	12/05/2015	12:35:41
2	8c:0c:a3:35:6b:19	MOVISTAR_6B19	-84	12/05/2015	12:35:41

Tabla 5.4: Ejemplo de mediciones de registros RSSI

Medición	X	Y	Z
1	0.102	0.008	1.138
2	0.098	0.003	1.160

Tabla 5.5: Ejemplo de mediciones de registros Kinect (en metros)

A partir de los valores máximos y mínimos de x y z , se ha creado una matriz de posicionamiento dividida en 10x10 celdas (ver Figura 5.13). En cada celda se han grabado las distintas mediciones RSSI obtenidas por el dispositivo móvil, de acuerdo

al número de celda calculado a partir de las coordenadas devueltas por el sensor Kinect, según la Expresión 5.7, donde $E(x)$ representa la función parte entera.

$$\begin{cases} C_X = E(10 \cdot \frac{X - \min_X}{\max_X - \min_X}) \\ C_Z = E(10 \cdot \frac{Z - \min_Z}{\max_Z - \min_Z}) \end{cases} \quad (5.7)$$

De todas las mediciones RSSI que se han obtenido dentro de cada celda, se ha calculado el centroide para cada grupo de valores RSSI devueltos por cada AP. Cada celda tiene, por lo tanto, un conjunto de valores RSSI asociados correspondientes a cada AP, como se muestra en la Tabla 5.6 para la celda $C_x = 4$, $C_z = 5$.

C_x	C_z	BSSID	SSID	Centroide RSSI (dBm)
4	5	dc:53:7c:25:2c:36	NUBE	-49
4	5	02:53:7c:25:2c:37	_AUTO_ONOWiFi	-49
4	5	3e:d9:54:28:4f:53	JAZZTEL_AF89	-69
4	5	8c:0c:a3:35:6b:19	MOVISTAR_6B19	-85
4	5	e0:91:53:5f:1e:85	WLAN_95	-83
4	5	9c:80:df:b9:3a:b7	JUAN	-86
4	5	50:7e:5d:43:15:ca	Orange15C8	-85

Tabla 5.6: Centroides RSSI para una celda

Durante una segunda fase del experimento, el teléfono móvil ha obtenido las mediciones RSSI y se ha buscado la celda en la que debía encontrarse según la base de datos. Para ello, se ha calculado el error de posicionamiento como la distancia euclídea entre los vectores RSSI de la nueva medición y del centroide de cada celda, según la Expresión 5.8, donde $rssip$ representa la señal recibida de un AP p , y $rssip_{C_x, C_z}$ representa el centroide, previamente calculado de la señal recibida de un AP p para la celda C_x, C_z . La celda estimada es en la que se produce menor error.

$$error_{C_x, C_z} = \frac{1}{\text{total_APs}} \cdot \sqrt{\sum_{p \in \text{APs}} (rssip - rssip_{C_x, C_z})^2} \quad (5.8)$$

Los datos obtenidos a partir del sensor Kinect han permitido obtener la celda RGB-D donde se encontraba la persona, y compararla con la celda obtenida mediante

WPS. Se ha calculado el error producido entre la obtención de la celda WPS y la celda RGB-D, de acuerdo al sensor Kinect. La Tabla 5.7 recoge las pruebas de localización, donde los resultados obtenidos mediante posicionamiento *fingerprinting* no ofrecen resultados precisos a priori, de acuerdo a las celdas obtenidas por RGB-D. En la Figura 5.15 se muestran, para cada posición considerada, flechas desde la celda obtenida por WPS hasta la celda RGB-D. Cada flecha tiene un indicador, referente al número de posicionamiento considerado, y un color distinto en función del error entre las dos tecnologías.

Posicionamiento	Celda WPS (mediante <i>fingerprinting</i>)	Celda RGB-D (mediante Kinect)	Error (metros)
Posicionamiento 1	C(6,2)	C(4,5)	1.33
Posicionamiento 2	C(5,1)	C(8,9)	3.16
Posicionamiento 3	C(3,5)	C(4,1)	1.53
Posicionamiento 4	C(9,5)	C(5,1)	2.09
Posicionamiento 5	C(9,5)	C(5,1)	2.09
Posicionamiento 6	C(10,10)	C(7,7)	1.57
Posicionamiento 7	C(2,5)	C(9,10)	3.18
Posicionamiento 8	C(2,5)	C(5,1)	1.85
Posicionamiento 9	C(3,3)	C(3,3)	0.00
Posicionamiento 10	C(4,2)	C(5,1)	0.52

Tabla 5.7: Posiciones mediante *fingerprinting*

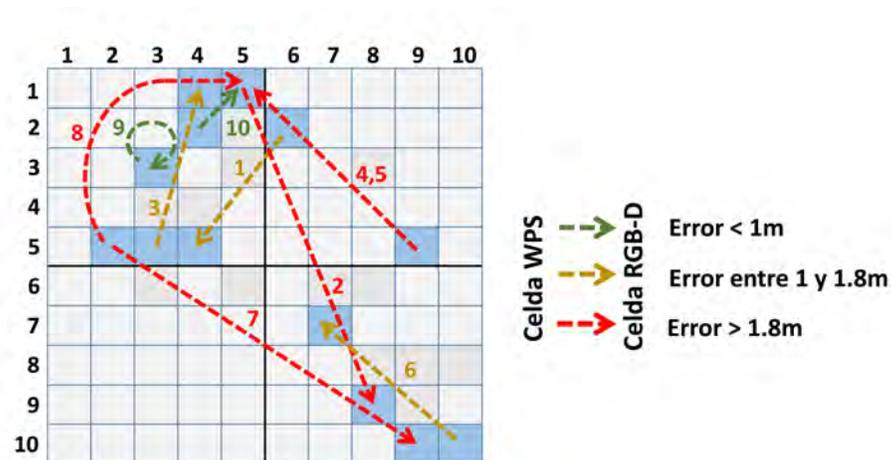


Figura 5.15: Diferencia entre posicionamiento WPS y RGB-D

Como conclusión al análisis de datos Kinect y RSSI, hay que mencionar que los datos Kinect tienen menor variabilidad que los datos RSSI, en los cuales se presentan algunos valores atípicos. Además, el posicionamiento WiFi mediante *fingerprinting* no ofrece resultados precisos, como se muestra en la Tabla 5.7, ya que una persona puede ser detectada incluso en otra habitación distinta de la que se encuentra. Los sensores RGB-D permiten mejorar el posicionamiento WiFi, como se muestra en el siguiente apartado.

5.3. Experimentos de posicionamiento WPS con RGB-D en una sala

Con el objetivo de evaluar la combinación de WPS con RGB-D, se han llevado a cabo varios experimentos en la habitación presentada en el apartado anterior, una habitación de 4.7 x 4.3 metros, donde 8 APs fueron posicionados en distintas ubicaciones del edificio (ver Figura 5.14). A diferencia del caso anterior, varios usuarios han participado en el proceso utilizando teléfonos móviles.



Figura 5.16: Sensor Kinect montado en una plataforma de madera

Adicionalmente, en estos experimentos se ha empleado un sensor RGB-D de tipo Kinect v2. Como en el caso anterior, el campo de visión horizontal de este sensor

RGB-D es 70° , por lo que, solo es capaz de detectar a las personas en una sección de 3.71×3.71 metros. Como se indica en la siguiente sección (Sección 5.4), para cubrir una habitación completa se necesitan dos o más sensores RGB-D. Los sensores utilizados en los experimentos se han montado en trípodes, sobre plataformas de madera, como se observa en la Figura 5.16.

Durante la etapa de aprendizaje, un usuario ha generado el mapa *fingerprint* y la correspondencia con los esqueletos. El usuario se ha movido alrededor de la sala para producir 1000 mediciones diferentes.

Al final de la etapa de aprendizaje, el mapa se ha dividido en 25 celdas (5×5 celdas cuadradas de 0.74 metros de lado) y se han calculado los centroides RSSI para cada celda, donde los escaneos RSSI se han agrupado de acuerdo a la distancia entre su esqueleto asociado original y el centro de cada celda.

En este experimento se han considerado dos modos de funcionamiento del sistema, durante la etapa de posicionamiento, como se ha explicado en la Sección 4.1 del capítulo anterior: un *Modo Mejorado sin Trayectoria*, donde no se considera la trayectoria que las personas han seguido por la habitación; y un *Modo Mejorado con Trayectoria*, donde se consideran las trayectorias WPS y RGB-D, como se muestra en la Figura 5.17.

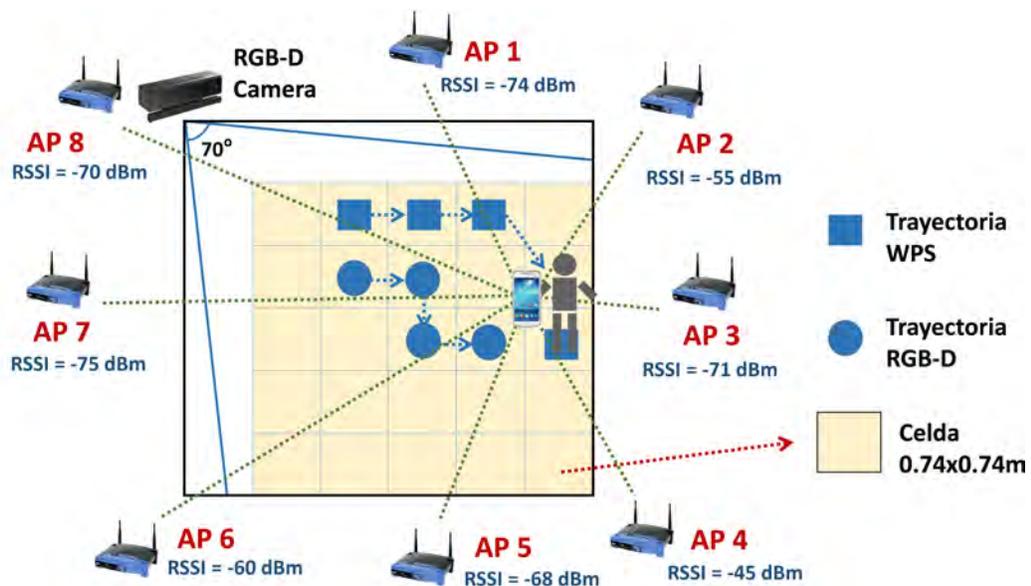


Figura 5.17: Trayectorias en la habitación donde se han realizado los experimentos

Para probar el sistema, se han realizado 250 pruebas, considerando 2 y 3 usuarios en la habitación, en diferentes experimentos. Un usuario ha grabado distintas trayectorias, como se muestra en la Figura 5.18, moviéndose por la habitación y almacenando los datos WPS y RGB-D para cada una de las trayectorias. Posteriormente, se han combinado las trayectorias en grupos de 2 ó 3 de cara a simular una situación con 2 y 3 usuarios respectivamente. Cuando no se utilizan las trayectorias de los usuarios, en el *Modo Mejorado sin Trayectoria*, los resultados muestran que se detectan correctamente dos usuarios en el 73% de los casos, y tres en el 46% de los casos, representando dicho porcentaje el emparejamiento correcto entre rutas WPS y RGB-D. Cabe destacar que la mayoría de los errores se producen cuando los usuarios están en la misma celda. Cuando se tienen en cuenta las trayectorias de los usuarios, en el *Modo Mejorado con Trayectoria*, los resultados mejoran considerablemente, como se muestra en la Tabla 5.8. Para el caso en el que se utilizan las trayectorias de los usuarios se han considerado mediciones en 4 *time stamps*.

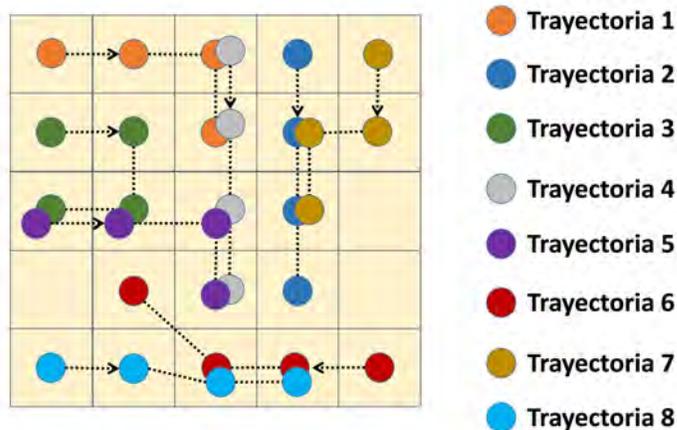


Figura 5.18: Algunas de las trayectorias seguidas por los usuarios

Uso de trayectoria	Éxitos (2 usuarios)	Éxitos (3 usuarios)
No	73 %	46 %
Sí	89 %	71 %

Tabla 5.8: Resultados de experimento WPS con RGB-D

Los resultados muestran que, el mejor rendimiento se obtiene cuando los usuarios están en diferentes celdas. Cuando el número de usuarios aumenta, el rendimiento es menor porque hay más trayectorias de esqueleto para cada trayectoria de WPS, dentro de un espacio pequeño (4.7 m x 4.3 m). A diferencia del caso anterior, se han utilizado celdas de mayor tamaño para reducir el error de posicionamiento mediante WPS ya que, como se vio anteriormente, era alto para celdas de 0.37 metros de lado.

5.4. Experimentación de posicionamiento WPS con RGB-D en entornos complejos

Para los experimentos en entornos complejos también se han utilizado sensores Kinect v2. Los experimentos se han llevado a cabo en una oficina de $80m^2$, con 8 habitaciones y un pasillo central, donde 20 cámaras RGB-D, Kinect v2, han sido desplegadas cubriendo todos los puntos accesibles del espacio (ver Figura 5.19 donde las cámaras son representadas con círculos etiquetados). Los experimentos se han desarrollado teniendo en cuenta los ejes X e Y ya que todas las salas están situadas en el mismo piso. Como se indica posteriormente en la Subsección 5.4.1, este despliegue se ha realizado de manera simulada. De esta forma, a partir de 2 sensores RGB-D disponibles que se han movido entre las distintas habitaciones, se han simulado 20 sensores en los experimentos. Además, la transformación de las coordenadas devueltas por la Kinect al UCS se lleva a cabo a través de las técnicas presentadas en la Subsección 5.4.2.

En cuanto al sistema de posicionamiento inalámbrico, hay que indicar que los usuarios han utilizado *smartphones* Lenovo A936, con sistema operativo Android 4.4 Kitkat. Se han instalado varios *routers* de diferentes tipos: Conceptronic C54BRS4, SMC7908A, Linksys WRT54G, etc. Todos estos *routers* se pueden encontrar habitualmente en instalaciones domésticas o profesionales. Un servidor Apache Tomcat 8.0.27 se ha instalado en el servidor central y en los servidores conectados a cada sensor RGB-D.

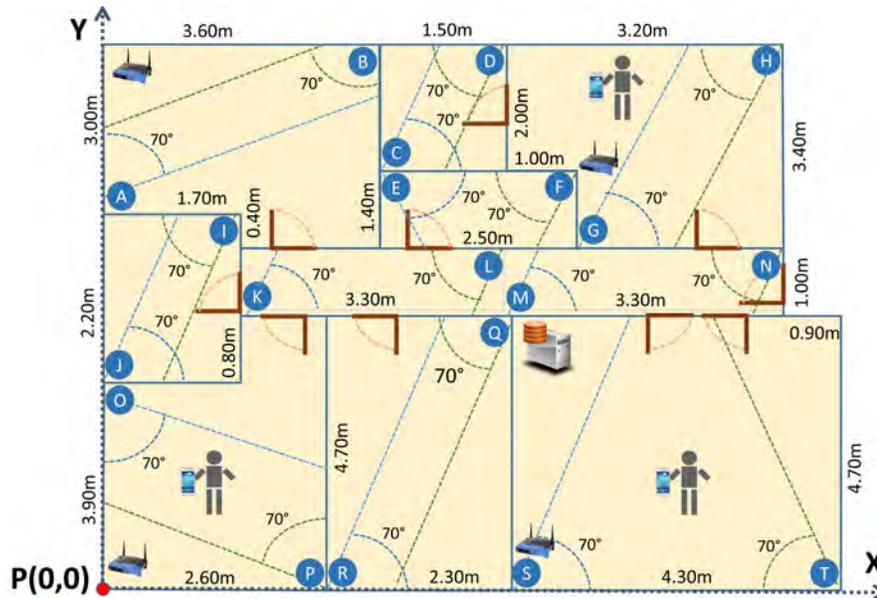


Figura 5.19: Plano de la oficina donde se han realizado los experimentos

Los teléfonos de los usuarios han establecido una conexión a la red WiFi local, y aunque todos los experimentos se han llevado a cabo en dicha red, la conexión entre los *smartphones* y el servidor central puede realizarse también por medio de la red de telefonía móvil (3G, 4G, etc.). Los teléfonos han obtenido las intensidades RSSI de cada AP sin necesidad de estar conectados a una red, pero han necesitado una conexión de datos para comunicarse con el servidor central. Una vez obtenidos los datos RSSI, estos se han enviado de forma sincronizada al servidor web central, a través del protocolo SOAP [117] y utilizando la red local. Al mismo tiempo, este servidor central ha iniciado 20 hilos de ejecución diferentes, uno por cada sensor RGB-D, como se muestra en la Figura 5.20, cada uno de los cuales ha llamado a un servicio web diferente para obtener los esqueletos. Estos servicios web han sido alojados en los servidores respectivos conectados a cada Kinect asociada. Cuando estos hilos han terminado, la información completa se ha guardado y procesado. La utilización de los hilos de ejecución permite realizar en paralelo el proceso de obtención de todos los esqueletos en todos los sensores RGB-D.

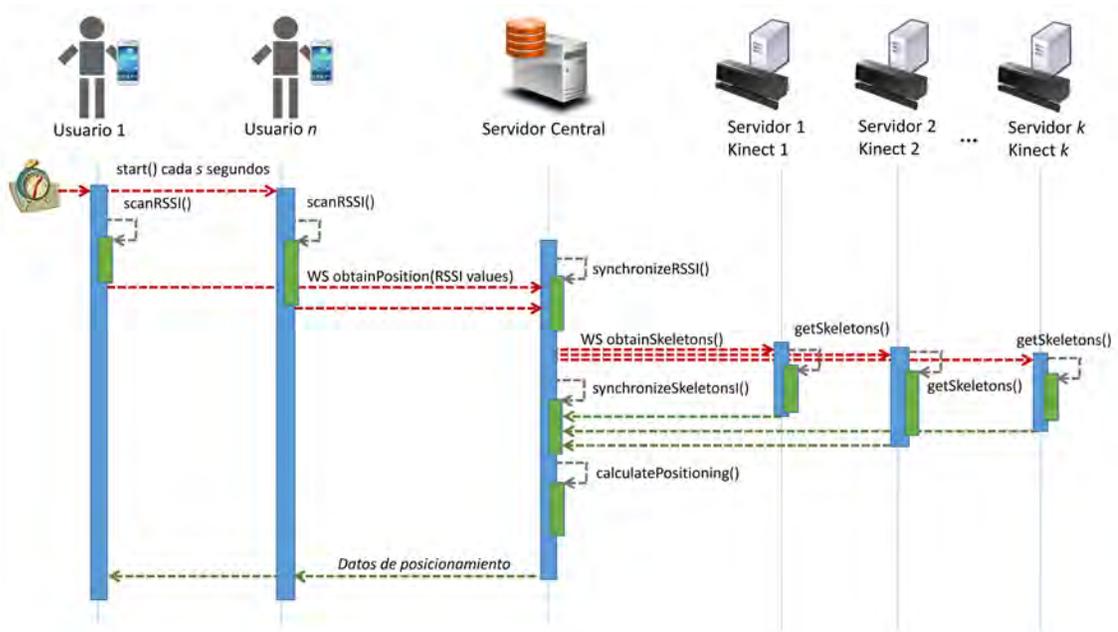


Figura 5.20: Diagrama de secuencia del sistema

Para llevar a cabo el experimento, se ha desarrollado una aplicación Android, con los modos de aprendizaje y de ejecución, que rastrea la posición del usuario mediante WPS y el IPS desarrollado (ver Figura 5.21). Durante el modo de aprendizaje, la aplicación envía datos RSSI al servidor central, para generar el mapa. Durante el modo de ejecución, la aplicación envía datos RSSI al servidor y recibe datos de posición que muestra en pantalla. En pantalla muestra la celda WPS y la posición RGB-D, que el sistema de posicionamiento ha obtenido para el usuario, junto con los cálculos de la distancia euclídea sincronizada y el número de *time stamps* considerados.

El rendimiento del sistema ha sido evaluado por un grupo de hasta 20 usuarios que han sido rastreados simultáneamente en los experimentos, los cuales han recorrido más de 50 trayectorias diferentes. Como se ilustra, a modo de ejemplo, en la Figura 5.22, se han grabado diez marcas de tiempo para cada trayectoria, con un intervalo de 2 segundos entre mediciones consecutivas.

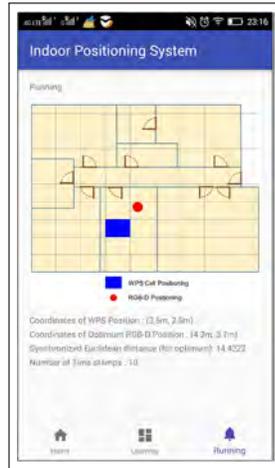


Figura 5.21: Captura de la aplicación desarrollada para dispositivos Android

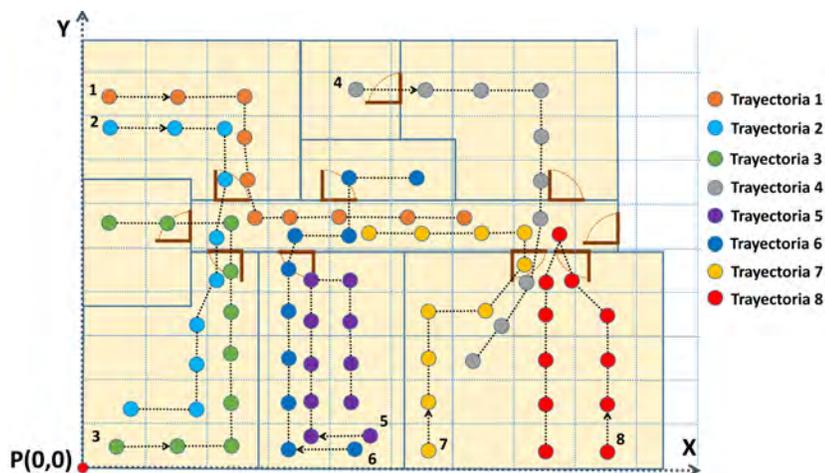


Figura 5.22: Algunas de las trayectorias seguidas por los usuarios

La sincronización entre los teléfonos y el servidor central se ha realizado con un servicio externo de sincronización de tiempo (*time.nist.gov*), mediante el protocolo de tiempo de red, *Network Time Protocol* (NTP). Se han considerado intervalos de 2 segundos entre los *time stamps* para obtener las mediciones y esqueletos RSSI. Cada servidor conectado a una Kinect se sincroniza con el servidor central mediante un servicio interno de sincronización de tiempo. El servidor central actúa como controlador de dominio y representa la fuente de tiempo autoritativa para el resto de equipos de la subred interna.

Con el fin de evaluar el sistema para varios usuarios, se han registrado diversas trayectorias y se han combinado para completar, más de un millón de posibles situaciones aleatorias por número de usuarios.

Los siguientes apartados muestran las distintas herramientas o métodos que se han utilizado durante los experimentos de posicionamiento en entornos complejos, presentando: el despliegue simulado de sensores (Subsección 5.4.1), la transformación de las coordenadas de cada sensor al UCS (Subsección 5.4.2), el cálculo de la distancia euclídea sincronizada (Subsección 5.4.3) y, finalmente, los resultados obtenidos (Subsección 5.4.4).

5.4.1. Despliegue simulado de sensores y trayectorias

Para la realización de los experimentos se ha contado con dos sensores RGB-D. Por este motivo, se han tenido que ir moviendo de habitación en habitación hasta completar el despliegue de 20 cámaras. En cada habitación se han grabado previamente los valores RSSI y las posiciones Kinect para generar el mapa *fingerprint* de las señales WiFi, así como un conjunto de trayectorias posibles. En la Figura 5.23, se pueden ver las trayectorias generadas en cada habitación, donde cada punto corresponde a una marca de tiempo donde se ha efectuado la medición. Esto es posible ya que, las mediciones han registrado los datos cuando el usuario presionaba un botón de la aplicación móvil. Posteriormente, las trayectorias se han combinado para generar trayectorias con al menos 10 marcas de tiempo, ya que en el experimento se consideran trayectorias con 10 *time stamps*.

La Tabla 5.9 muestra las distintas combinaciones de trayectorias individuales que se han realizado. Las trayectorias están compuestas de las mediciones de los sensores Kinect y de los valores WPS, los cuales sirven para la realización de las posteriores evaluaciones. Cuando en la trayectoria se muestra un número entre paréntesis, por ejemplo $E(2)(3)$, quiere decir que se seleccionan las mediciones 2 y 3 de la trayectoria referida, para generar la nueva. Así, en el caso $D \rightarrow E(2)(3) \rightarrow Inv(F)$, se indica que la nueva trayectoria estará formada por 6 *time stamps* de D , 2 *time stamps* de E (el 2 y el 3) y, finalmente, 2 *time stamps* de F invertida. Como la trayectoria F se ha generado en sentido inverso (ver la flecha de F), será necesario invertir el orden de los *time stamps* y seleccionar las 2 primeras mediciones para completar 10 *time*

stamps en total. La función $Inv(x)$ refleja la inversión de la trayectoria.

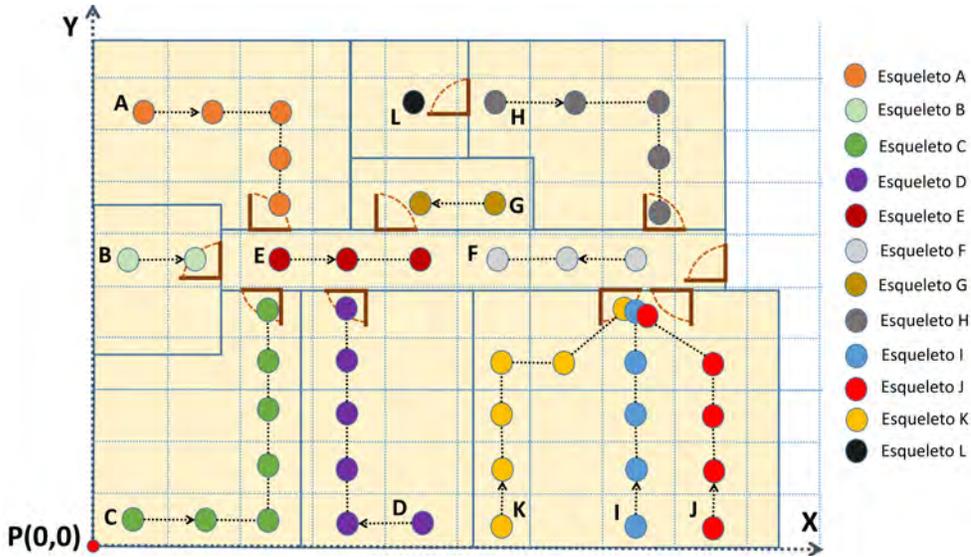


Figura 5.23: Trayectorias generadas durante la simulación de los experimentos

$C \rightarrow E$	$Inv(E) \rightarrow Inv(C)$	$C \rightarrow E(1) \rightarrow Inv(A)$
$A \rightarrow E(1) \rightarrow Inv(C)$	$A \rightarrow E \rightarrow Inv(G)$	$G \rightarrow Inv(E) \rightarrow Inv(A)$
$K \rightarrow F \rightarrow Inv(E)$	$E \rightarrow Inv(F) \rightarrow Inv(K)$	$I \rightarrow F \rightarrow Inv(E)$
$E \rightarrow Inv(F) \rightarrow Inv(I)$	$J \rightarrow F \rightarrow Inv(E)$	$E \rightarrow Inv(F) \rightarrow Inv(J)$
$H \rightarrow F \rightarrow Inv(E)$	$E \rightarrow Inv(F) \rightarrow Inv(H)$	$F \rightarrow Inv(E) \rightarrow Inv(A)$
$A \rightarrow E \rightarrow Inv(F)$	$B \rightarrow E \rightarrow Inv(F) \rightarrow Inv(H)$	$L \rightarrow H \rightarrow F \rightarrow Inv(E)$
$D \rightarrow E(2)(3) \rightarrow Inv(F)$	$B \rightarrow E(1)(2) \rightarrow Inv(D)$	$L \rightarrow H \rightarrow F(1) \rightarrow Inv(K)$
$L \rightarrow H \rightarrow F(1) \rightarrow Inv(J)$	$I \rightarrow F(1) \rightarrow Inv(H)$	$H \rightarrow F \rightarrow E(3) \rightarrow Inv(G)$
$F \rightarrow E(3)(2) \rightarrow Inv(D)$	$G \rightarrow Inv(E) \rightarrow Inv(C)$	$C \rightarrow E \rightarrow Inv(G)$
$C \rightarrow E(1)(2) \rightarrow Inv(D)$	$D \rightarrow E(2)(1) \rightarrow Inv(C)$	$B \rightarrow E(1) \rightarrow Inv(C)$
$C \rightarrow E(1) \rightarrow Inv(B)$	$B \rightarrow E(1) \rightarrow Inv(A) \rightarrow A$	$A \rightarrow E(1) \rightarrow Inv(B) \rightarrow B$
$A(4)(5) \rightarrow E \rightarrow Inv(F) \rightarrow H(5)(4)$	$B \rightarrow E \rightarrow Inv(E) \rightarrow Inv(B)$	$K \rightarrow Inv(J)$
$J \rightarrow Inv(K)$	$A \rightarrow E \rightarrow Inv(E) \rightarrow Inv(A)$	$D \rightarrow E(2)(1) \rightarrow Inv(A)$
$C \rightarrow Inv(C)$	$D \rightarrow Inv(D)$	$A \rightarrow Inv(A)$
$H \rightarrow Inv(H)$	$G \rightarrow E(3) \rightarrow Inv(F) \rightarrow Inv(K)$	$G \rightarrow E(3) \rightarrow Inv(F) \rightarrow Inv(H)$
$G \rightarrow E(3) \rightarrow Inv(F) \rightarrow Inv(J)$	$C \rightarrow Inv(C)$	$D \rightarrow Inv(D)$
$A \rightarrow Inv(A)$	$H \rightarrow Inv(H)$	

Tabla 5.9: Composición de las distintas trayectorias

A partir de las 50 trayectorias generadas, es posible realizar experimentos combinándolas de distintas maneras. Así, por ejemplo, para evaluar el posicionamiento

de dos usuarios, se combinan y evalúan aleatoriamente una trayectoria WPS, con sus datos RGB-D, con otra distinta, con sus datos WPS y RGB-D, para ver si los resultados del sistema emparejan correctamente la trayectoria WPS con su trayectoria RGB-D. Esto es posible ya que al grabar manualmente los *time stamps*, se tiene asociada cada medición WPS con su esqueleto.

Si hay un conjunto de n elementos distintos, y se quiere calcular todos los posibles subconjuntos de dicho conjunto que tengan m elementos, sin importar el orden de dichos m elementos, se dice que hay C combinaciones de n elementos tomados de m en m (o bien número combinatorio n sobre m). La Expresión 5.9 muestra el cálculo del número combinatorio n sobre m .

$$C = \binom{n}{m} = \frac{n!}{m!(n-m)!} \quad (5.9)$$

En el caso de combinaciones de 2 usuarios sobre las 50 trayectorias distintas, el número total de posibilidades es de 1,225. Cuando 10 usuarios se encuentran en el entorno bajo estudio, se obtiene un número combinatorio de 50 sobre 10, que resulta en $1.02 \cdot 10^{10}$ posibilidades. Finalmente, cuando 20 usuarios se encuentran simultáneamente, se obtiene un número combinatorio de 50 sobre 20, que resulta en $4.71 \cdot 10^{13}$ posibilidades. En los experimentos se han evaluado situaciones hasta con 20 usuarios en el entorno bajo estudio. Como se puede apreciar en todos los casos, el número de combinaciones es muy alto, por lo que se ha establecido un límite de cálculo de 1 millón de combinaciones, para lo cual ha sido necesario un día de cómputo para obtener todos los resultados. Las combinaciones utilizadas se han generado de manera aleatoria y sin repeticiones.

5.4.2. Ubicación de los sensores en el UCS

Cada sensor RGB-D tiene una pose diferente (posición y orientación) con respecto al UCS. Como se ha mencionado anteriormente, en todos los experimentos realizados con cámaras RGB-D se han utilizado sensores de tipo Kinect v2, los cuales tienen un ángulo limitado de trabajo de 70° . Para obtener un máximo rango de trabajo de varios sensores en una sala, se han orientado los sensores de manera que cubran toda

la superficie, como se muestra en la Figura 5.24.

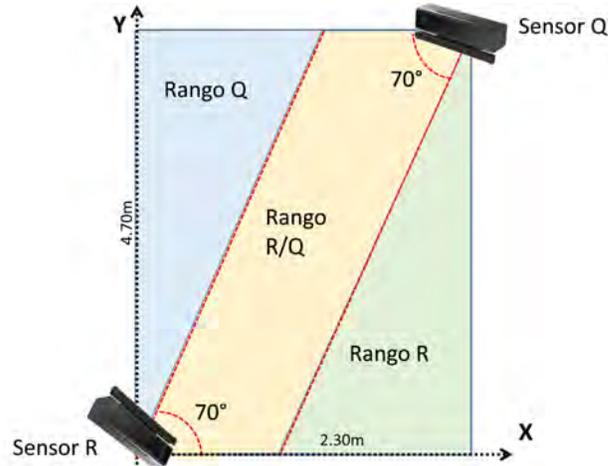


Figura 5.24: Ubicación de los sensores y rango de trabajo dentro de una habitación

En la Figura 5.25, los valores X_{POS} , Y_{POS} representan la posición del sensor respecto de la esquina. Kinect v2 tiene 25 cm de ancho. Una matriz de rotación permite girar las coordenadas obtenidas por el sensor RGB-D respecto al UCS, rotando el ángulo ω en el sentido de las agujas del reloj. El ángulo ω se calcula a partir de ε , obtenido en la Ecuación 5.10. Ambos ángulos pertenecen a un triángulo, con otro ángulo de 90° , por lo que el valor de ω es el mostrado en la Ecuación 5.11.

$$\varepsilon = \arcsin\left(\frac{X_{POS}}{25}\right) \quad (5.10)$$

$$\omega = 90^\circ - \varepsilon = 90^\circ - \arcsin\left(\frac{X_{POS}}{25}\right) \quad (5.11)$$

La matriz de rotación general en dos dimensiones viene determinada por la Expresión 5.12. El ángulo de giro que considera es en contra de las agujas del reloj.

$$R(\mu) = \begin{bmatrix} \cos \mu & -\sin \mu \\ \sin \mu & \cos \mu \end{bmatrix} \quad (5.12)$$

en la misma unidad que se utiliza en el UCS, es decir metros, no es necesario realizar un escalado. La Tabla 5.10 muestra los valores de posición de los sensores Kinect en la sala, X_{POS} e Y_{POS} , así como la distancia de la esquina donde se encuentra el sensor respecto al UCS, y los valores del vector de traslación, C_X y C_Y .

$$\begin{cases} X_{UCS} = X_{CORNER} + X'_{RGB-D} \\ Y_{UCS} = Y_{CORNER} + Y'_{RGB-D} \end{cases} \quad (5.15)$$

$$\begin{aligned} \begin{bmatrix} X_{UCS} \\ Y_{UCS} \end{bmatrix} &= \begin{bmatrix} \cos(-\omega) & -\sin(-\omega) \\ \sin(-\omega) & \cos(-\omega) \end{bmatrix} \begin{pmatrix} X_{RGB-D} \\ Z_{RGB-D} \end{pmatrix} \\ &+ \begin{bmatrix} C_X \\ C_Y \end{bmatrix} + \begin{bmatrix} X_{CORNER} \\ Y_{CORNER} \end{bmatrix} \end{aligned} \quad (5.16)$$

<i>Sensor</i> <i>RGB-D</i>	<i>Distancia</i> X_{CORNER}	<i>Distancia</i> Y_{CORNER}	<i>Desplazamiento</i> X_{POS}	<i>Desplazamiento</i> Y_{POS}	C_X	C_Y
A	0.00	6.10	0.21	0.14	0.07	0.10
B	3.60	9.10	-0.22	-0.13	0.06	0.11
C	3.60	7.10	0.13	0.21	0.06	0.11
D	5.10	9.10	-0.13	-0.21	0.06	0.11
E	3.60	7.10	0.13	-0.21	0.07	0.11
F	6.10	7.10	-0.13	-0.21	0.07	0.11
G	6.10	5.70	0.13	0.21	0.07	0.11
H	8.30	9.10	-0.14	-0.21	0.07	0.10
I	1.70	6.10	-0.13	-0.21	0.06	0.11
J	0.00	3.90	0.13	0.21	0.06	0.11
K	1.70	4.70	0.14	0.21	0.07	0.10
L	5.00	5.70	-0.14	-0.21	0.07	0.10
M	5.00	4.70	0.14	0.21	0.07	0.10
N	8.30	5.70	-0.14	-0.21	0.07	0.10
O	0.00	3.90	0.21	-0.14	0.07	0.10
P	2.60	0.00	-0.21	0.14	0.07	0.10
Q	4.90	4.70	-0.13	-0.21	0.07	0.11
R	2.60	0.00	0.13	0.21	0.07	0.11
S	4.90	0.00	0.14	0.21	0.07	0.10
T	9.20	0.00	-0.14	0.21	0.07	0.10

Tabla 5.10: Tabla de parámetros de emplazamiento de los sensores Kinect (en metros)

5.4.3. Cálculo de la distancia euclídea sincronizada

Como se muestra en la Sección 4.2.2 del Capítulo 4, diferentes usuarios tienen sus rutas correspondientes de WPS, $P_i(t)$, y RGB-D, $P'_j(t)$, para varios *time stamps*. Aunque inicialmente no es posible discernir qué trayectoria de RGB-D corresponde a cada trayectoria WPS, este emparejamiento de trayectorias se obtiene buscando la menor distancia euclídea sincronizada, que es la suma de las distancias entre pares de puntos, en *time stamps* idénticos.

La Expresión 5.17, correspondiente a la distancia euclídea sincronizada [106], se ha resuelto mediante programación lineal con el *Algoritmo Aditivo de Balas* [108]. En estos experimentos ha habido hasta 20 usuarios ($n=20$), y se han utilizado los diez últimos *time stamps* consecutivos para cada cálculo de posición ($m=10$). Para la resolución del problema, se han considerado las restricciones mostradas en la Expresión 5.18.

$$\text{mín} \left[\begin{array}{l} s_{11} \cdot \sum_{t=1}^m d_E(P_1(t), P'_1(t)) + \dots + s_{1n} \cdot \sum_{t=1}^m d_E(P_1(t), P'_n(t)) + \\ \dots \\ + s_{n1} \cdot \sum_{t=1}^m d_E(P_n(t), P'_1(t)) + \dots + s_{nn} \cdot \sum_{t=1}^m d_E(P_n(t), P'_n(t)) \end{array} \right] \quad (5.17)$$

$$\text{restricciones} \left\{ \begin{array}{l} s_{11} \in (0, 1) \\ \dots \\ s_{1n} \in (0, 1) \\ \dots \\ s_{n1} \in (0, 1) \\ \dots \\ s_{nn} \in (0, 1) \\ s_{11} + s_{12} + \dots + s_{1n} = 1 \\ \dots \\ s_{n1} + s_{n2} + \dots + s_{nn} = 1 \end{array} \right. \quad (5.18)$$

Las distancias euclídeas son valores fijos, calculados como la distancia entre las

coordenadas del cuello del esqueleto obtenido por la Kinect, en el UCS, y el centroide de la celda WPS donde el usuario ha sido detectado de acuerdo con el posicionamiento WPS. En la Expresión 5.19 se muestra el cálculo de la distancia, donde las coordenadas del cuello de un usuario i en un *time stamp* t están determinadas por $(x_{UCS,i}(t), y_{UCS,i}(t))$, mientras que las coordenadas del centroide de la celda WPS, donde el usuario j ha sido detectado, están determinadas por $(x_{WPS,j}(t), y_{WPS,j}(t))$.

$$d_E(P_i(t), P'_j(t)) = \sqrt{(x_{UCS,i}(t) - x_{WPS,j}(t))^2 + (y_{UCS,i}(t) - y_{WPS,j}(t))^2} \quad (5.19)$$

A su vez, la celda WPS se obtiene calculando las distancias euclídeas entre los valores RSSI, obtenidos por el teléfono móvil, y los vectores RSSI almacenados en el mapa *fingerprint*. Durante la etapa de aprendizaje, el mapa se reduce para obtener un vector RSSI por celda. De esta manera, un vector RSSI está asociado al centroide de la celda y el sistema funciona más rápido durante la etapa normal de posicionamiento.

5.4.4. Resultados

Los experimentos de posicionamiento WPS con RGB-D, en entornos complejos, se han llevado a cabo considerando 10 *time stamps* consecutivos (separados entre sí 2 segundos). Como se muestra en la Tabla 5.11, los experimentos con un menor número de *time stamps* no ofrecen buenos resultados para la comparación entre trayectorias RGB-D y WPS, especialmente cuando el número de usuarios aumenta. Un experimento es calificado de exitoso, cuando todos los usuarios considerados han sido emparejados correctamente entre WPS y RGB-D, devolviendo su posición según el sensor Kinect.

Cuantos más *time stamps* son considerados en la trayectoria, mejores son los resultados obtenidos. El sistema es capaz de localizar e identificar con precisión hasta 10 usuarios en un 95 % de los casos, y hasta 15 usuarios en un 95 % de las situaciones.

<i>Time stamps</i>	1	2	3	4	5	6	7	8	9	10
1	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
2	82.24	85.55	90.37	92.98	95.10	95.96	97.80	98.24	99.59	99.84
3	56.53	60.95	72.87	80.44	86.30	88.12	94.36	95.06	98.74	99.62
4	30.89	38.59	52.85	64.81	73.03	77.52	87.92	90.22	97.54	99.08
5	15.04	20.20	32.35	49.21	59.31	65.39	80.72	84.72	96.00	97.96
6	5.32	8.12	17.14	34.11	45.45	53.87	73.07	77.44	94.14	97.84
7	3.64	6.20	14.66	31.13	44.89	53.17	72.01	75.04	93.70	97.32
8	0.94	2.04	5.64	18.88	31.79	39.73	61.55	65.75	91.04	95.98
9	0.50	1.06	2.86	13.64	24.86	32.81	57.69	61.61	89.48	95.04
10	0.50	1.02	2.98	13.22	25.59	33.55	56.87	60.69	88.96	94.64
11	0.18	0.34	1.92	10.04	23.84	30.77	54.21	55.47	87.32	93.48
12	0.14	0.16	1.28	8.42	20.84	27.69	52.11	52.73	86.06	92.50
13	0.04	0.14	0.48	6.34	17.38	23.20	48.17	48.35	85.22	91.20
14	0.00	0.08	0.30	4.70	15.46	21.62	45.05	44.03	83.20	90.08
15	0.00	0.04	0.14	2.88	11.30	17.36	40.01	37.41	81.00	87.74
16	0.00	0.00	0.04	1.40	7.10	12.04	32.89	32.41	77.42	87.40
17	0.00	0.00	0.00	0.66	3.82	7.28	26.83	25.93	73.55	84.36
18	0.00	0.00	0.00	0.54	2.86	5.36	22.38	22.56	71.45	82.74
19	0.00	0.00	0.02	0.10	2.22	4.30	21.42	20.00	68.91	81.18
20	0.00	0.00	0.00	0.10	0.52	1.80	12.98	13.72	63.75	78.00

Tabla 5.11: Porcentaje de éxito en función del número de usuarios y *time stamps*

En cuanto al error cometido en el posicionamiento, varios autores [21, 49, 50] se refieren a este valor como la distancia entre la posición obtenida del usuario y la real. Considerando un error de posicionamiento de 0.2m, para usuarios satisfactoriamente localizados (como se ha visto anteriormente, la Kinect ofrece un error de 0.03m), y 2.0m para el resto (error con WPS), se puede calcular un error medio, mostrado en la Tabla 5.12.

Número de usuarios	1	5	10	15	20
1 <i>time stamp</i>	0.2	1.73	1.99	2.00	2.00
2 <i>time stamps</i>	0.2	1.64	1.98	2.00	2.00
3 <i>time stamps</i>	0.2	1.42	1.95	2.00	2.00
4 <i>time stamps</i>	0.2	1.11	1.76	1.95	2.00
5 <i>time stamps</i>	0.2	0.93	1.54	1.80	1.99
6 <i>time stamps</i>	0.2	0.82	1.40	1.69	1.97
7 <i>time stamps</i>	0.2	0.55	0.98	1.28	1.77
8 <i>time stamps</i>	0.2	0.47	0.91	1.33	1.75
9 <i>time stamps</i>	0.2	0.27	0.40	0.54	0.85
10 <i>time stamps</i>	0.2	0.23	0.29	0.42	0.59

Tabla 5.12: Error de distancia medio (metros) para diferente número de usuarios y *time stamps*

Algoritmo de emparejamiento extendido

Cuando el número de personas detectado por las cámaras RGB-D no es igual al número de usuarios detectados por WPS, es necesario aplicar el *Algoritmo de emparejamiento extendido*. Este algoritmo permite obtener el emparejamiento óptimo de rutas WPS con RGB-D para aquellos usuarios detectados mediante ambas tecnologías. El resto de usuarios son posicionados de acuerdo a la ubicación devuelta por el método que los detecta, WPS o RGB-D.

En los experimentos llevados a cabo, se muestra en la Tabla 5.13 el éxito al emparejar correctamente las rutas WPS con las RGB-D. Para este experimento, se han utilizado rutas con 10 *time stamps*, y se ha variado el número de usuarios detectados por WPS y RGB-D entre 6 y 20.

RGB-D \ WPS	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
6	97.84	94.76	93.00	91.00	89.12	85.94	83.42	81.70	79.08	78.26	74.95	72.43	71.87	69.61	67.43
7	94.50	97.32	94.30	91.50	88.64	86.38	83.96	81.24	78.90	76.84	73.69	72.23	69.99	68.09	65.01
8	91.72	93.66	95.98	93.34	90.62	86.82	83.90	80.82	76.84	74.05	72.99	70.03	67.41	65.95	61.01
9	89.36	90.42	92.92	95.04	91.78	88.68	85.60	81.06	78.28	75.08	72.13	68.61	65.45	62.93	61.83
10	86.80	86.66	87.62	90.34	94.64	90.50	85.44	82.26	78.72	75.88	72.39	68.31	65.23	62.35	60.65
11	84.56	85.06	84.68	86.98	89.70	93.48	88.56	83.62	80.48	76.20	73.35	69.33	67.37	63.59	62.25
12	81.58	81.70	81.58	83.64	84.72	87.88	92.50	88.30	82.96	78.82	74.85	71.61	68.69	65.43	60.79
13	79.20	79.40	79.56	79.66	81.14	83.44	85.94	91.20	86.10	80.46	77.48	73.29	69.57	66.07	62.91
14	76.74	77.18	75.08	75.00	77.76	77.98	80.88	84.82	90.08	85.42	80.44	75.00	70.77	67.53	61.87
15	75.12	75.20	73.33	72.89	72.59	75.04	75.68	78.86	83.98	87.74	82.32	77.56	73.33	68.13	62.99
16	75.60	72.87	70.67	70.17	69.45	70.99	71.61	73.79	76.84	81.36	87.40	80.66	75.78	69.81	65.65
17	73.39	71.79	68.17	66.83	66.49	66.65	66.73	67.91	71.49	75.86	79.78	84.36	78.82	73.45	68.83
18	69.73	68.29	65.41	63.91	63.15	63.27	63.17	63.25	66.09	68.37	73.37	77.54	82.74	77.58	71.31
19	69.35	67.35	64.21	62.61	59.55	58.85	58.33	60.97	62.83	64.01	66.45	70.27	74.95	81.18	74.57
20	67.13	65.33	62.05	59.37	58.01	56.73	57.33	58.99	57.53	60.61	62.27	63.83	68.01	73.29	78.00

Tabla 5.13: Éxito al emparejar correctamente usuarios WPS con RGB-D para $m = 10$ *time stamps*

5.5. Experimentos en predicción de movimientos

Se han realizado varios experimentos, con la finalidad de predecir los movimientos de los usuarios del sistema. Para ello, se han aplicado las distintas técnicas presentadas en la Sección 4.3 del Capítulo 4: regresión lineal y polinómica, filtro de Kalman y redes neuronales.

Los experimentos han consistido en determinar, para una persona que se ha estado moviendo en el interior de una habitación, la siguiente posición que va a ocupar. Para simplificar el proceso de posicionamiento y predicción, el piso del área de trabajo se ha dividido en celdas, como se ha explicado en secciones anteriores. Después de obtener el mapa *fingerprint*, se han creado 25 celdas (5 x 5 celdas cuadradas de 0.74 m de lado). La división en celdas permite obtener fácilmente la posición del usuario y se utiliza para el proceso de predicción. De esta manera, el sistema predice en qué celda se situará el usuario durante la siguiente marca de tiempo. Para los experimentos, se han considerado las cuatro últimas posiciones en las que el usuario ha estado situado, obtenidas en las últimas cuatro marcas de tiempo. Dichas posiciones se han obtenido mediante las coordenadas del esqueleto devueltas por el sensor Kinect.

- Las primeras técnicas consideradas han sido las relacionadas con la regresión. Como se indica en la Sección 4.3, cada eje ha sido considerado independientemente para el proceso de regresión.
- La segunda técnica empleada ha sido el filtro de Kalman, donde el vector de estado inicial incluye las coordenadas del usuario y el desplazamiento, entre posiciones de usuario consecutivas, en varias marcas de tiempo.
- Para el experimento con redes neuronales, se ha utilizado la red presentada en la Sección 4.3.1. Se han registrado 200 coordenadas de usuario para entrenar a la red, que consisten en varios patrones de movimientos que los usuarios han realizado. Con ellos, se han añadido otros patrones, de movimientos preestablecidos, a la base de datos para el proceso de aprendizaje. Los resultados de los experimentos se muestran usando la división en celdas, creada durante el experimento de posicionamiento WPS con RGB-D en una habitación.

La Figura 5.26 muestra cinco situaciones en las que se representan varias trayectorias. Las flechas con líneas discontinuas señalan las predicciones obtenidas con los diferentes métodos. Las celdas en verde más claro representan la posición real del usuario en el tiempo de predicción. En algunos casos, los métodos obtienen una predicción fuera del área de trabajo. La regresión polinómica con grado $n = 3$ produce valores atípicos mientras que los otros métodos producen predicciones más precisas. En estas cinco situaciones, las redes neuronales ofrecen los mejores resultados.

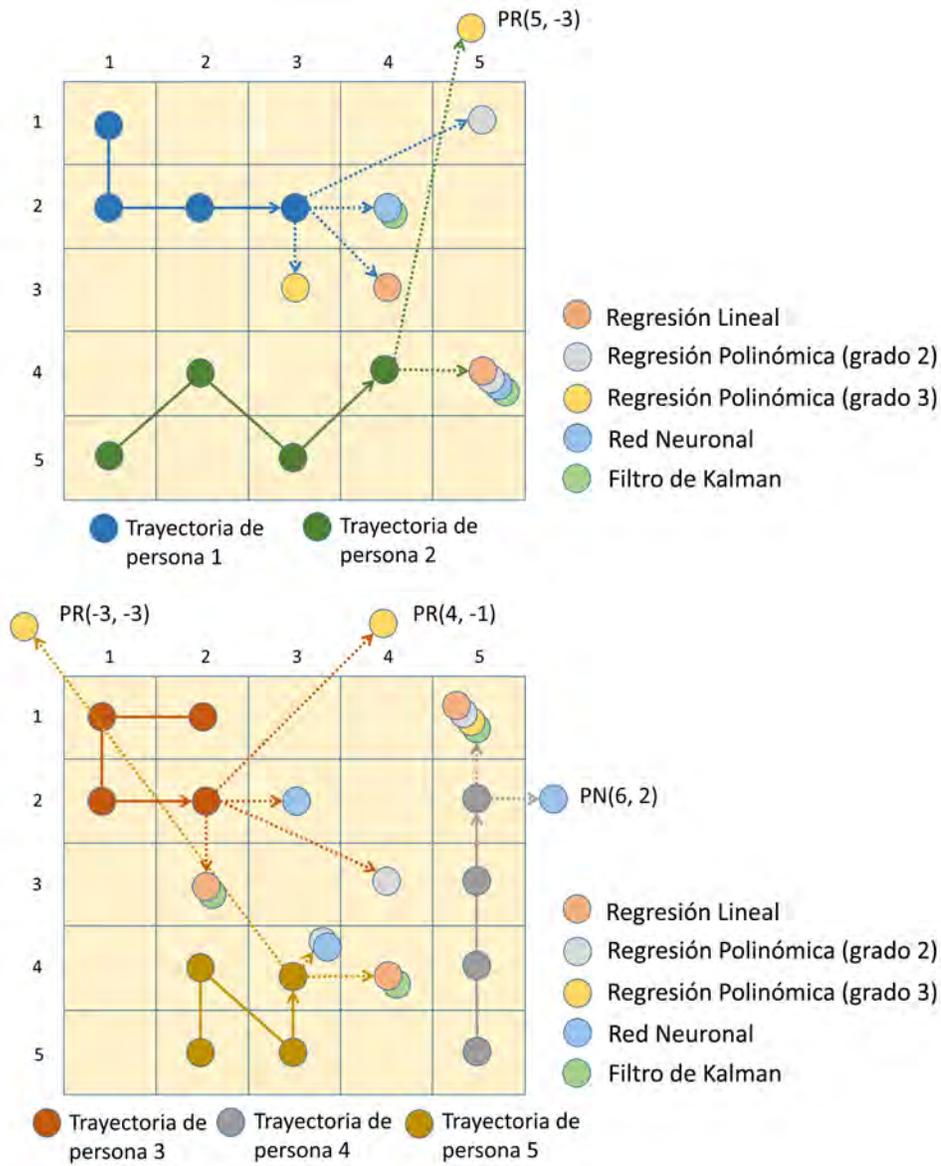


Figura 5.26: Resultados de la predicción de trayectorias de los esqueletos

Si hay obstáculos estáticos en el espacio de trabajo que impiden el movimiento del usuario, como una mesa o un armario, las redes neuronales ofrecen mejores predicciones porque solo aprenden movimientos reales posibles. Durante el aprendizaje, los usuarios evitan los obstáculos. El filtro de Kalman y la regresión no tienen en cuenta los obstáculos y no son capaces de tratar con ellos.

Cuando hay obstáculos dinámicos en el espacio de trabajo, como una silla o una caja, ninguno de estos métodos los evita. Los resultados de los experimentos se muestran en la Tabla 5.14.

Técnica	Éxitos
Regresión lineal simple	76 %
Regresión polinómica (grado $n = 2$)	80 %
Regresión polinómica (grado $n = 3$)	36 %
Filtro de Kalman	84 %
Red Neuronal	92 %

Tabla 5.14: Éxitos en la predicción de movimientos en interiores.

Capítulo 6

Integración del IPS con la plataforma Europea

En este capítulo se presenta el diseño de una aplicación, donde se pone de manifiesto la utilidad del sistema de posicionamiento desarrollado. Esta aplicación permite ubicar a los usuarios dentro de museos y mostrar información sobre las obras de arte que tienen enfrente de ellos, sin necesidad de teclear el nombre de la obra o pulsar algún botón de búsqueda en un dispositivo multimedia. La utilización de Europeana, la plataforma digital de la Unión Europea para el patrimonio cultural, con una ontología concebida para enlazar el IPS desarrollado con dicha plataforma, permite explotar la información disponible dentro de la aplicación.

Un usuario se mueve libremente por un museo portando un teléfono móvil con la aplicación en ejecución. El IPS desarrollado o las API de Google proporcionan la posición del usuario, la cual permite calcular la distancia a los objetos. Cuando la persona se encuentra cerca de una obra, la aplicación móvil ofrece información sobre los Objetos del Patrimonio Cultural (CHO, del inglés *Cultural Heritage Objects*), recuperando dichos datos de Europeana. Esta información incluye imágenes de la obra, datos de nombre, autor, año, material, etc. Esta aplicación permite a los usuarios conocer los detalles de las obras de un museo, de una manera rápida y sencilla. La utilización de una ontología, para conectar el IPS con Europeana, facilita un comportamiento dinámico, añadiendo nuevas funcionalidades de una manera simple sin tener que modificar el modelo de datos.

Aunque se han realizado diferentes aproximaciones semánticas, en relación con el problema de encontrar el camino óptimo dentro de un escenario con IPSs [73–75], o estudiar las tareas del usuario en posiciones concretas [76], en este capítulo se muestra un nuevo enfoque que integra el posicionamiento en interiores con la visión semántica de los objetos culturales. La nueva ontología desarrollada enriquece la experiencia del usuario y ofrece una nueva forma de disfrutar del arte. El uso de ontologías es particularmente eficaz porque el sistema es flexible para establecer las relaciones existentes, los valores conocidos, y para inferir el conocimiento necesario en base a dichas relaciones y valores. En el caso de la aplicación diseñada, dicho conocimiento son los datos de las obras de arte.

A lo largo del capítulo se hace una breve introducción a Europeana (Sección 6.1), se muestra cómo se utilizan los datos de Europeana con la ontología desarrollada y se integran con un sistema de posicionamiento (Sección 6.2), se presenta la aplicación móvil desarrollada (Sección 6.3) y, finalmente, se muestran los experimentos realizados dentro de un museo (Sección 6.4).

6.1. Europeana

Europeana [118] es la plataforma digital de la Unión Europea para el patrimonio cultural, que da acceso a diferentes tipos de contenidos en varias instituciones. Más de 3.000 instituciones de toda Europa han contribuido a la plataforma. Las colecciones permiten a los usuarios explorar el patrimonio cultural y científico de Europa desde la prehistoria hasta la actualidad. Los objetos digitales en Europeana se almacenan en la institución cultural y se alojan en sus redes. Europeana recopila información contextual, o metadatos, sobre los ítems. Los datos abiertos vinculados (*Linked Open Data*) de Europeana (data.europeana.eu) contienen metadatos sobre aproximadamente 2.4 millones de textos, imágenes, vídeos y sonidos.

Europeana es un proyecto desarrollado por la Comisión Europea para incentivar, a los estados miembros, a digitalizar y mejorar la preservación digital en Europa. El portal ofrece acceso gratuito a una amplia gama de contenidos digitales. La plataforma tiene varias formas de recuperar datos: mediante REST API Standard sobre HTTP, que devuelve datos JSON, o por *Annotations* REST API, que

devuelve JSON-LD; mediante recolección de datos OAI-PMH a través del protocolo OAI-PMH; mediante consultas de datos abiertos vinculados (*Linked Open Data Queries*), a través de SPARQL. Los datos también se pueden enlazar con fuentes externas de información, tales como el *Swedish Cultural Heritage Aggregator* (SOCH), GeoNames, GEMET thesaurus o DBPedia.

En lugar de desarrollar un sistema ontológico completo, para reflejar el lugar donde se encuentra cada obra, se ha planteado utilizar Europeana como fuente de CHO, que proporciona información global sobre la ubicación de los objetos.

6.1.1. Consultas contra Europeana

En este trabajo, se han utilizado consultas SPARQL para recuperar información sobre las obras. Por ejemplo, si se busca información sobre el famoso cuadro de Diego Velázquez, *Las Meninas*, una consulta SPARQL proporciona los resultados mostrados en la Tabla 6.1.

CHO	http://data.europeana.eu/proxy/provider/2059101/MP_P01174
title	"Las meninas, o La familia de Felipe IV"@es
creator	"Velázquez, Diego Rodríguez de Silva y"
date	"1656"@es
type	"Pintura"
subject	"Retrato"
format	"Lienzo"@es
description	"Las meninas, o La familia de Felipe IV"
publisher	"Museo Nacional del Prado"
extent	"Alto: 318 cm.; Ancho: 276 cm."@es
latitude	"40.41369"
longitude	"-3.69241"
mediaURL	" https://www.museodelprado.es/uploads/tx_gbobras/P01174.jpg "

Tabla 6.1: Resultados para la consulta de Las Meninas

Las consultas a través de Europeana permiten obtener la longitud y la latitud de los edificios, como el Museo del Prado, pero no la posición de una pieza concreta dentro del museo. Como se detalla en la Sección 6.2, se necesita crear otra relación,

mediante una nueva ontología, que dé la sala concreta y la posición de una obra maestra, para ser utilizada con el sistema desarrollado. La nueva ontología no es solo una relación entre un objeto CHO y su posición en un museo en particular, sino que es capaz de proporcionar más información interrelacionada con los diferentes casos.

6.2. Nueva ontología para integrar el IPS con Europea

Una ontología es una especificación de una conceptualización [83, 84], que define y clasifica conceptos y entidades, y las relaciones entre ellos. Las ontologías se componen de entidades (por ejemplo, clases, relaciones, funciones u otros objetos) y axiomas formales que limitan la interpretación y el uso bien formado de estos términos.

Las clases proporcionan un mecanismo de abstracción para agrupar recursos con características similares. En este caso en particular, se han predefinido dos identificadores de clase en el lenguaje Web semántico OWL: las clases *Thing* y *No Thing*. La extensión de *Thing* es el conjunto de todos los individuos, mientras que para *No Thing* es el conjunto vacío. En consecuencia, cada clase OWL es una subclase de *Thing* [85]. Los individuos, en la extensión de la clase, se denominan las instancias de la clase. Si una clase se define como una subclase, el conjunto de individuos que acepta debe ser un subconjunto, de aquellos individuos en la clase padre.

OWL distingue dos categorías principales de propiedades que se definen en una ontología: propiedades de objeto, que vinculan individuos a individuos; y las propiedades de tipo de datos, que vinculan a los individuos con los valores de los datos.

Por otro lado, dominio y rango son axiomas utilizados en el proceso de inferencia, que es el proceso de deducción que permite obtener conocimiento a partir de los datos conocidos y las relaciones existentes en la ontología. Ambos se definen como propiedades integradas. Un axioma de dominio vincula una propiedad a una descripción de clase, e indica que los sujetos de tales declaraciones de propiedad deben pertenecer a la extensión de la clase indicada. Un axioma de rango vincula una propiedad a una descripción de clase o a un rango de datos, indicando que los valores de esta propiedad deben pertenecer a la extensión de la clase, o a valores de datos

en el rango especificado.

En esta sección se presenta la ontología propuesta (Subsección 6.2.1), mostrando ejemplos a partir de datos obtenidos del Museo del Prado, que incluye cerca de 90 obras maestras en Europeana, representando a artistas como Velázquez o Goya. A continuación, se muestra cómo se integra dicha ontología con Europeana (Subsección 6.2.2), y cómo se integra la ontología con distintos tipos de IPSs (Subsección 6.2.3).

6.2.1. Ontología propuesta

La ontología representada en la Figura 6.1 ha sido concebida para dar soporte a la integración entre diferentes IPS, incluyendo el desarrollado en capítulos anteriores o los sistemas de posicionamiento de Google. Como se detalla a continuación, esta ontología permite recuperar distintos tipos de información.

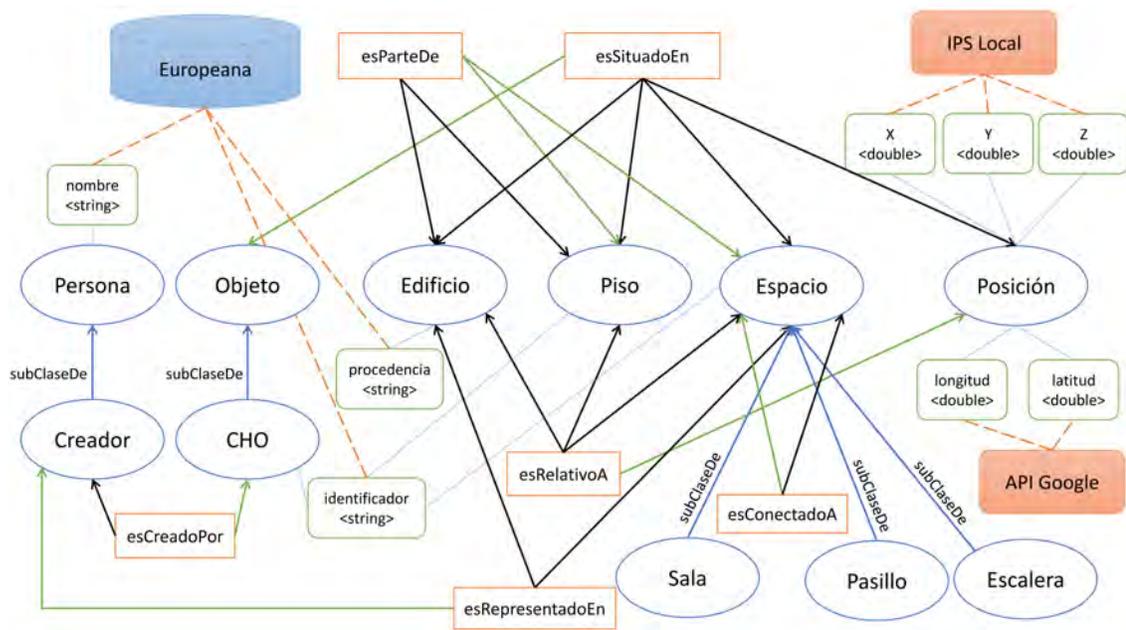


Figura 6.1: Ontología implementada

Hay seis subclases generales de *Thing*: *Persona*, *Objeto*, *Edificio*, *Piso*, *Sala* y *Posición*. Al mismo tiempo, *Persona* tiene una subclase *Creador* y *Objeto* tiene una subclase *CHO* (*Cultural Heritage Object*). Los axiomas de dominio se muestran con

flechas verdes en la Figura 6.1, mientras que los axiomas de rango se indican con flechas negras.

La *Persona* se refiere a todo tipo de individuos, pero el *Creador* solo se refiere a los creadores recuperados principalmente de Europeana. La propiedad de datos “nombre” del tipo $\langle string \rangle$ acepta valores como “Velázquez, Diego Rodríguez de Silva y”.

Objeto se refiere a todo tipo de objetos que se pueden ubicar en algún lugar: un ordenador o una mesa en una sala, una pantalla o TV, una pintura en una pared, etc. *CHO* son objetos particulares que tienen un identificador para enlazar con Europeana. La propiedad de datos “identificador”, de tipo $\langle string \rangle$ acepta valores como “http://data.europeana.eu/proxy/provider/2059101/MP_P01174”, identificador de Europeana.

Un *Edificio* es el lugar relacionado con el editor en Europeana. Tiene la propiedad de datos “procedencia” de tipo $\langle string \rangle$ que acepta valores como “Museo Nacional del Prado”. Como se ha visto anteriormente, el uso de consultas SPARQL contra Europeana permite obtener la longitud y la latitud del edificio. También hay clases para *Piso* y *Sala*.

La clase *Posición* indica la posición de un objeto con referencia a una *Sala*, un *Piso* o un *Edificio*. Contiene tres propiedades de datos: “X”, “Y”, “Z”, de tipo $\langle double \rangle$, que mediante la propiedad de objeto “esRelativoA” se indica que son relativas a una sala, piso o edificio. Las otras dos propiedades de datos, “longitud” y “latitud”, de tipo $\langle double \rangle$, no son relativas a un edificio, piso o edificio. Estos valores se utilizan para posicionarse con las API de Google, como se explica más adelante. La propiedad de objeto “esParteDe” está relacionada con el lugar donde se encuentran las salas o los pisos. Una sala es parte de un piso y también de un edificio y, al mismo tiempo, un piso es parte de un edificio.

Esta ontología permite considerar varias posibilidades dependiendo de las necesidades de la aplicación móvil. Por ejemplo, una posición “posición1” puede ser relativa a un piso “piso1” y otra posición “posición2” puede ser relativa a una sala “sala1”, pero ambas pueden estar relacionadas con el mismo objeto. “objeto1” puede estar situado en “posición1”, pero también en “posición2”. El sistema es flexible y permitiría incluir relaciones como “objeto1” está situado en “piso1”.

Piso y *Sala* también tienen identificadores para conocer su nombre particular. Por ejemplo, la sala con el identificador “12”, en el Museo del Prado, es en la que está situada la pintura “Las meninas”. Esta sala es parte del piso con el identificador “Primer piso”. Otra propiedad del objeto consiste en “esCreadoPor”. Un *CHO* es creado por un *Creador*. El *CHO* con el identificador “http://data.europeana.eu/proxy / provider / 2059101 / MP_P01174”, que se refiere a “Las meninas”, ha sido creado por el creador, cuya persona tiene el nombre “Velázquez, Diego Rodríguez de Silva y ”.

La propiedad del objeto “esRepresentadoEn” indica que un creador tiene ilustraciones en una sala o edificio en particular. Por ejemplo, el creador cuya persona tiene el nombre de “Velázquez, Diego Rodríguez de Silva y” está representado en la sala con el identificador “12”, que es parte del piso con el identificador “Primer piso”, que forma parte del edificio con la “procedencia”: “Museo Nacional del Prado”. Al mismo tiempo, el creador está representado en el edificio, ya que hay varias salas con sus obras de arte.

Esta ontología es capaz de representar múltiples combinaciones y permite vincularse con Europeana y otras fuentes de conocimiento. Dependiendo de los diferentes casos, es posible utilizar algunas clases, algunas propiedades de datos o algunas propiedades del objeto.

El código 6.1 muestra diferentes instancias y relaciones creadas para el cuadro “Las Meninas” en la ontología desarrollada.

Listado 6.1: Añadiendo “Las Meninas” a la base de conocimiento

```

creador1 esInstanciaDe Persona
creador1.nombre = "Velázquez, Diego Rodríguez de Silva y"
pintura1 esInstanciaDe CHO
pintura1.identificador = "http://data.europeana.eu/proxy/
                        provider/2059101/MP\_P01174"
pintura1 esCreadoPor creador1
edificio1 esInstanciaDe Edificio
edificio1.procedencia = "Museo Nacional del Prado"
piso1 esInstanciaDe Piso
piso1.identificador = "Primer piso"
piso1 esParteDe edificio1
sala12 esInstanciaDe Sala
sala12.identificador = "12"
sala12 esParteDe edificio1
sala12 esParteDe piso1

```

```

creador1 esRepresentadoEn sala12
posición1 esInstanciaDe Posición
posición1.X = 7.50
posición1.Y = 25.00
posición1 esRelativoA sala12
pintura1 esSituadoEn posición1
pintura1 esSituadoEn sala12
pintura1 esSituadoEn piso1
pintura1 esSituadoEn edificio1

```

6.2.2. Integración de la ontología con Europeana

Con el fin de recuperar la información sobre un conjunto de obras de arte próximas, se ha creado la Consulta 6.2, que obtiene la obra más cercana respecto al usuario, filtrando las piezas situadas a menos de 5 metros. Este cálculo se realiza por medio de la distancia euclídea, entre la posición del usuario y la posición de la obra de arte. La posición del usuario considerada es la obtenida por el IPS desarrollado.

Listado 6.2: Consulta para obtener el CHO más cercano combinando la presente ontología con Europeana

```

PREFIX mp: <http://www.amenofis.com/ontology/master_piece.owl#>
PREFIX dc: <http://purl.org/dc/elements/1.1/>
PREFIX dc_e: <http://purl.org/dc/terms/>
PREFIX edm: <http://www.europeana.eu/schemas/edm/>
PREFIX ore: <http://www.openarchives.org/ore/terms/>
PREFIX wgs84_pos: <http://www.w3.org/2003/01/geo/wgs84-pos#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX afn: <http://jena.hpl.hp.com/ARQ/function#>

SELECT ?CHO ?title ?creator ?date
(GROUP.CONCAT(?type ; SEPARATOR=" - ") AS ?type_final)
(GROUP.CONCAT(?subject ; SEPARATOR=" - ") AS ?subject_final)
(GROUP.CONCAT(?format ; SEPARATOR=" - ") AS ?format_final)
?description ?publisher ?extent ?latitude ?longitude ?mediaURL
?objetoCHO ?posicion ?X ?Y ?sala
(afn:sqrt( (POSICION_USUARIO.X - ?X) * (POSICION_USUARIO.X - ?X) +
(POSICION_USUARIO.Y - ?Y) * (POSICION_USUARIO.Y - ?Y) )
AS ?distance)

WHERE {
  ?objetoCHO mp:esSituadoEn ?posicion ;
  mp:identificador ?id.

```

```

?posicion mp:X ?X;
           mp:Y ?Y;
           mp:esRelativoA ?sala .

FILTER (
  afn:sqrt( (POSICION_USUARIO_X - ?X) *
            (POSICION_USUARIO_X - ?X) +
            (POSICION_USUARIO_Y - ?Y) *
            (POSICION_USUARIO_Y - ?Y) ) < 5 ).

SERVICE <http://sparql.europeana.eu> {
?CHO edm:type "IMAGE" ;
     ore:proxyIn ?proxy;
     dc:title ?title ;
     dc:identifier ?id ;
     dc:creator ?creator ;
     dc:date ?date ;
     dc:type ?type ;
     dc:subject ?subject ;
     dc:format ?format ;
     dc:description ?description ;
     dc:publisher ?publisher ;
     dc:e:extent ?extent ;
     edm:currentLocation ?location .
?location wgs84_pos:lat ?latitude ;
          wgs84_pos:long ?longitude .
?proxy edm:isShownBy ?mediaURL .
FILTER (regex(?publisher, "Prado")) }
} GROUP BY ?CHO ?title ?creator ?date ?description
?publisher ?extent ?latitude ?longitude ?mediaURL
?objetoCHO ?posicion ?X ?Y ?sala
ORDER BY DESC (?distance)
LIMIT 1

```

Para acceder al *endpoint* remoto de SPARQL de Europeana se hace uso de consultas federadas [119] (ver el enlace SERVICE dentro de la consulta). Se utilizan las funciones AQR [120] para implementar la raíz cuadrada y el cálculo de la distancia euclídea. Se ha creado un servidor Jena de SPARQL [121] para soportar la propia ontología, la cual se ha implementado utilizando Protégé 5.1.

6.2.3. Integración de IPSs con la ontología

La ontología presentada se puede conectar a dos tipos de IPSs diferentes: uno de ellos basado en posicionamiento global y otro orientado a posicionamiento local.

La diferencia entre ellos consiste en el tipo de coordenadas, globales o relativas a la Tierra, o locales y relativas a un edificio, espacio o sala.

Los sistemas de posicionamiento locales son más precisos en entornos interiores. Por esta razón, se puede utilizar un sistema de posicionamiento global, como Google API, para identificar si el usuario está en un edificio concreto, como los Museos Reales de Bellas Artes de Bélgica, el Museo del Prado o el Louvre. Entonces, el IPS local, como el presentado en capítulos anteriores, es utilizado para el posicionamiento interior devolviendo coordenadas más precisas.

El sistema presentado avisa al usuario cuando una obra de arte está a menos de una distancia concreta del usuario, por ejemplo 10 metros. La distancia se calcula por medio de la distancia euclídea entre la posición real del usuario y la posición de la obra de arte.

A continuación se presenta el posicionamiento mediante las APIs de Google y Android.

Estimación de la posición mediante las APIs de Google y Android

Android proporciona una API en `android.location` [104] que permite obtener la posición del usuario tanto en interiores como en exteriores. La API de Android devuelve la posición según tres métodos diferentes: GPS, ID de celda y WiFi. El sistema devuelve la posición a un objeto, del tipo *Location Listener*, que invoca un método cada vez que se recibe un nuevo objeto *Location*. La determinación de la mejor ubicación, si hay varios resultados de los diferentes métodos, se decide con la precisión, la velocidad y el estado de la batería. El movimiento del usuario también se toma en cuenta para volver a estimar la ubicación del usuario. El objeto *Location* proporciona información sobre la precisión y un método para calcular la distancia con otro objeto *Location*.

Por otro lado, Google proporciona una API de servicios de ubicación [122], parte de *Google Play Services*. De acuerdo con la API de localización de Android [123], la API de servicios de ubicación de Google proporciona una infraestructura más potente y de alto nivel, que controla automáticamente los proveedores de ubicación, el movimiento del usuario y la precisión de la ubicación.

Finalmente, la API de geolocalización de *Google Maps* [57] proporciona un servicio

web mediante HTTPS y mediante POST, el cual recibe como entrada una lista de torres de telefonía móvil y una lista de *routers* WiFi, con su intensidad de señal respectiva. El servicio devuelve la ubicación con longitud y latitud, y la precisión. Las comunicaciones con el servicio se establecen utilizando el formato JSON. Esta técnica requiere que el cliente obtenga datos sobre las torres de telefonía móvil y los *routers* WiFi. Respecto a las torres de telefonía móvil, la información que se tiene que enviar al servicio es: el identificador de celda, el código de área de la ubicación, el código del país móvil, el código de la red móvil y la intensidad de la señal. Respecto a las redes WiFi, el cliente tiene que enviar la dirección MAC y la intensidad de la señal.

Estos sistemas anteriores (Android Location API, Google Location API y Google Maps API) devuelven la ubicación del usuario tanto en interiores como en exteriores utilizando la longitud y la latitud, las cuales permiten calcular la distancia absoluta entre dos puntos. La fórmula de Haversine [124] (ver expresión 6.1 y siguientes) es un método para calcular la distancia entre dos localizaciones de una manera simple. Así mismo, las fórmulas de Vincenty [125] son dos métodos iterativos utilizados en la geodesia, para calcular la distancia entre dos puntos en la superficie de un esferoide.

$$R = Radius_{EARTH} \quad (6.1)$$

$$\Delta lat = lat_2 - lat_1 \quad (6.2)$$

$$\Delta long = long_2 - long_1 \quad (6.3)$$

$$a = \sin^2 \left(\frac{\Delta lat}{2} \right) + \cos(lat_1) \cdot \cos(lat_2) \cdot \sin^2 \left(\frac{\Delta long}{2} \right) \quad (6.4)$$

$$c = 2 \cdot \text{atan2} \left(\sqrt{a}, \sqrt{1-a} \right) \quad (6.5)$$

$$d = R \cdot c \quad (6.6)$$

La ontología creada es flexible y es capaz de asignar un valor de longitud y latitud a una instancia de *Posición*. Al mismo tiempo, es posible obtener la longitud y la latitud del edificio por medio de Europeana. Con el fin de combinar las API de Google con la ontología presentada, la aplicación móvil obtiene el museo, donde se encuentra el usuario, mediante las fórmulas de Vincenty, y posteriormente obtiene el

CHO más cercano por medio de la fórmula de Haversine.

Algunas bibliotecas, como [126], o el uso de Javascript embebido [127] permiten extender las funcionalidades de SPARQL para realizar cálculos con distancias. Así por ejemplo, la función `ldodds:Distance` calcula la distancia entre dos coordenadas geográficas basadas en longitud y latitud.

Finalmente, cabe destacar que la aplicación móvil permite obtener el posicionamiento en el museo, mediante el IPS presentado en capítulos anteriores (Sección 4.2).

6.3. Desarrollo de una aplicación móvil para integrar un IPS con la plataforma Europea

Se ha desarrollado una aplicación móvil con el objetivo de evaluar la integración de un IPS (Google API, o el IPS desarrollado) con la plataforma Europea. La aplicación tiene dos modos de funcionamiento: uno para registrar la posición de las obras de arte dentro de la ontología creada, y otro modo normal de detección donde se avisa al usuario cada vez que una obra es detectada. Antes de comenzar a utilizar el modo de detección, es necesario que un usuario registre las obras en la ontología.

En la Figura 6.2 se muestra el esquema de la aplicación desarrollada. En el modo de *Base de Datos* un usuario registra una obra de arte cada vez que pulsa el botón *Registrar*, introduciendo un identificador de Europea así como la procedencia de la obra. A continuación, la aplicación busca la ubicación donde se encuentra el usuario utilizando alguno de los IPSs. Finalmente, la aplicación registra los datos en la base de conocimiento, utilizando el identificador de Europea, la procedencia, y las coordenadas de ubicación.

En el modo de *Detección*, la aplicación obtiene la ubicación del usuario cada 2 segundos. A partir de dicha ubicación, realiza una búsqueda en Europea y la base de conocimiento generada, utilizando una consulta SPARQL federada, como se ha visto en la sección anterior. Este tipo de consultas permiten obtener información de dos fuentes de datos distintas. Finalmente, si se encuentra una obra de arte a menos de 5 metros, la aplicación se la muestra al usuario.

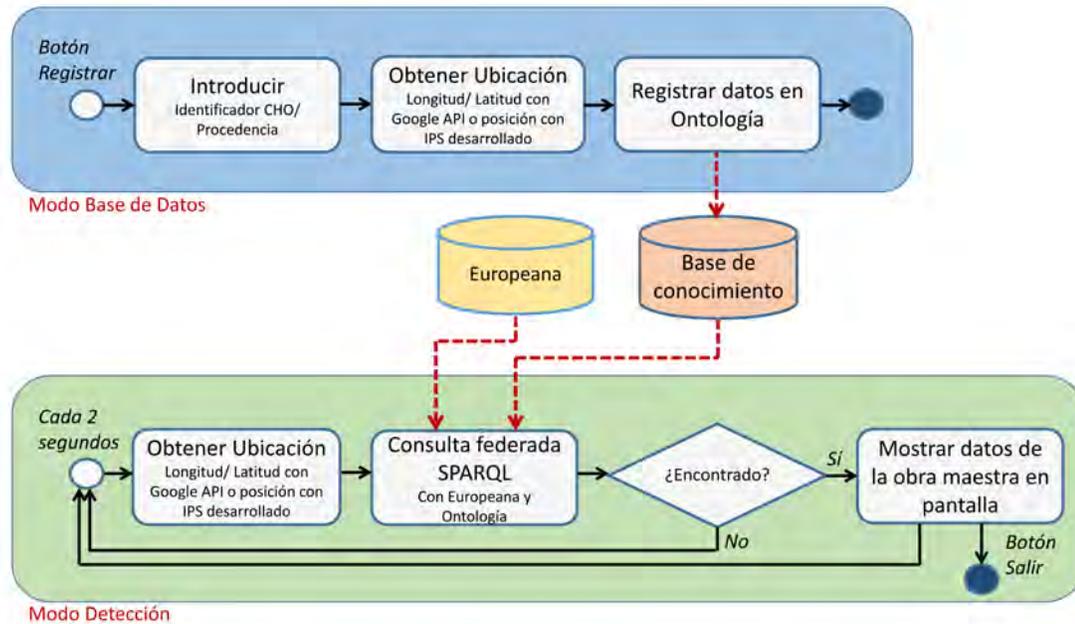


Figura 6.2: Esquema de aplicación desarrollada

6.3.1. Obtener el camino óptimo entre dos salas

La aplicación presentada permite también obtener el camino óptimo entre dos salas. Como se aprecia en la Figura 6.1, las salas están conectadas con otras salas, pasillos o escaleras. Tanto el algoritmo de Dijkstra [128] como el algoritmo A* [129] permiten obtener el camino más corto entre dos salas, las cuales son consideradas como nodos en un grafo. Los usuarios pueden seleccionar un CHO en el catálogo y el sistema calcula el camino óptimo desde su posición actual.

A modo de ejemplo, en la Figura 6.3, el algoritmo A* devuelve el camino óptimo para ir desde la sala “David” de los Museos Reales de Bellas Artes de Bélgica, donde se encuentra el cuadro “Marat Assassiné”, de Jacques-Louis David, a la sala “70_Weyden”, donde los usuarios pueden encontrar el cuadro “Déploration” de Weyden.

Las salas son subclases de *Espacio*, y en la Figura 6.3, el enlace entre ellas es representado por la propiedad de objeto “esConectadoA”, mostrado en la ontología previa (ver Figura 6.1).

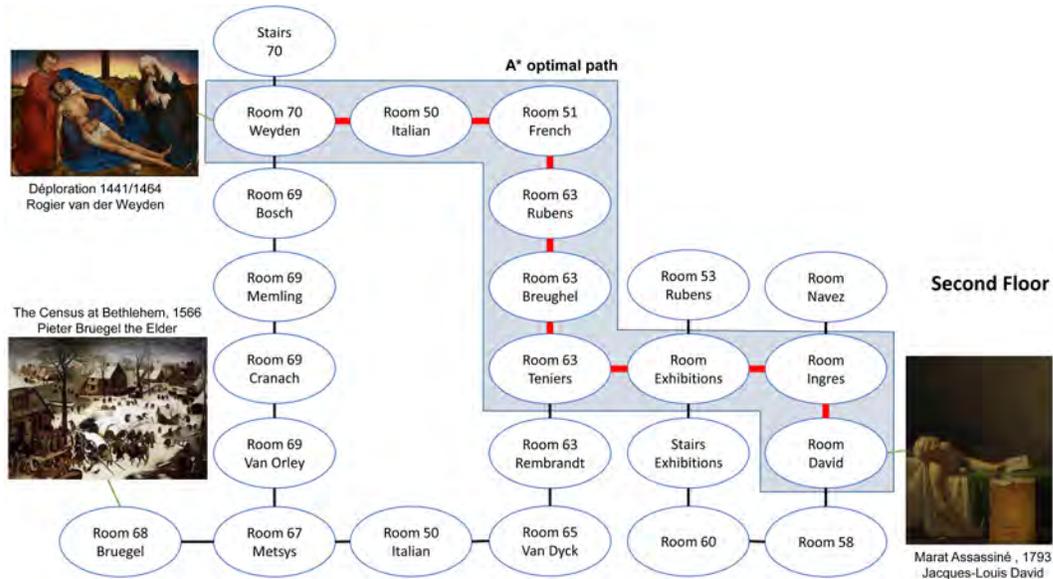


Figura 6.3: Obteniendo el camino óptimo entre dos salas en los Museos Reales

6.4. Experimentación

Con el objetivo de evaluar la ontología propuesta, se ha realizado un experimento en los Museos Reales de Bellas Artes de Bélgica, con más de 6000 piezas en Europea. En concreto, se han realizado los experimentos en el Museo de Pintura Antigua, uno de los museos que forman parte de los Museos Reales, y que está dedicado a pintores europeos de los siglos *XV* a *XVIII*.

El experimento ha tenido dos etapas, como se explica a continuación. Primero, un usuario se ha movido por el museo, registrando la localización de cada pintura de Europea, por medio de una aplicación Android desarrollada para este propósito (ver Figura 6.4a). La ubicación de cada obra maestra se ha obtenido utilizando la librería Google API y se ha almacenado en la base de datos. Este proceso solo se ha realizado una vez, y no es necesario repetirlo excepto en el caso de que las pinturas se cambien de ubicación en el museo. Cuatro pinturas han sido elegidas y sus posiciones se han grabado en la base de datos.

A continuación, otro usuario ha navegado por las habitaciones del museo, recuperando información de Europea con la aplicación Android instalada en el *smartphone*. Las cuatro pinturas seleccionadas se han visitado 5 veces, devolviendo la in-

formación correcta en la pantalla del teléfono, cada vez que el usuario se encontraba a menos de 5 metros de las pinturas. La Figura 6.4b muestra un ejemplo donde el sistema detecta el cuadro “The Census at Bethlehem” de Pieter Bruegel.



(a) Aplicación Android



(b) Detectando pintura

Figura 6.4: Experimentos desarrollados usando *Google Location Services API*.

Capítulo 7

Conclusiones

Esta tesis presenta un método para posicionamiento en interiores, en entornos complejos. Se presenta, inicialmente, un algoritmo que ha funcionado exclusivamente en una sala, utilizando un único sensor RGB-D para obtener los mapas de profundidad del entorno y, posteriormente, estimar la posición de las personas en la estancia con la ayuda del posicionamiento WiFi. De esta primera configuración, se extrae que el método propuesto permite detectar la posición correcta de dos personas, según el sensor RGB-D, en un 89% de los casos, y de tres personas en un 70%. A continuación, se presenta el método que permite obtener el posicionamiento dentro de un entorno más complejo. Se ha realizado un despliegue en parte simulado de 20 sensores RGB-D para obtener los esqueletos. La combinación de redes inalámbricas con esqueletos es un método simple y económico para aumentar el rendimiento de WPS en interiores. Cada esqueleto, que se obtiene a partir de mapas de profundidad, se compone de las coordenadas de las diferentes articulaciones de una persona: cuello, hombros, codos, rodillas, etc. Los experimentos desarrollados permiten ubicar a más de 10 personas correctamente, con un éxito superior al 95% en un escenario con múltiples salas, siendo el éxito la ubicación correcta de todas las personas, de acuerdo al posicionamiento de los sensores RGB-D.

La tesis también presenta el algoritmo extendido, para casos donde el número de personas detectadas mediante WPS y RGB-D no es igual. Esta situación puede ocurrir si una persona no lleva la aplicación móvil activa o si no es detectada por los sensores RGB-D. Los resultados indican que el éxito decrece cuanto mayor es la

diferencia entre el número de personas detectadas por cada sistema.

De acuerdo con la taxonomía establecida por [8], el sistema desarrollado obtiene una posición física con respecto a un sistema universal de coordenadas, opera reconociendo personas y procesa datos en servidores remotos. Considerando que los edificios están actualmente equipados con *routers*, la precisión, coste y limitaciones son establecidos por los sensores RGB-D utilizados. El sistema es escalable, permitiendo su uso en diferentes tipos de entornos. El sensor utilizado en los experimentos, Kinect v2, tiene un alcance limitado de unos pocos metros, por lo que es necesario utilizar más de un dispositivo. Otros dispositivos comerciales, de bajo coste, permiten obtener mapas de profundidad en rangos más amplios, siendo un ejemplo Peregrine 3D LIDAR Vision System [96], una cámara ligera capaz de capturar un mapa en 5 nanosegundos con la ayuda de un láser clase I. Este sistema puede funcionar con lentes de 60° y un rango de más de 1 km.

El método propuesto está abierto al *crowdsensing* [11], ya que es posible agregar conocimiento sin hacer un nuevo aprendizaje. Por lo tanto, cuando hay solo una persona en la sala, el sistema puede recalcular los centroides RSSI para cada una de las celdas en las que se divide el entorno, utilizando los nuevos datos obtenidos del móvil y el esqueleto (valores RSSI y esqueleto). Esta técnica ajusta los parámetros continuamente durante el funcionamiento del sistema.

Este método también es válido para identificar personas en sistemas como OpenPTrack [4], donde la gente es rastreada en tiempo real por medio de múltiples sensores RGB-D.

La tesis presenta un estudio sobre técnicas de predicción de los movimientos de personas utilizando el IPS presentado. Entre estas técnicas, las redes neuronales ofrecen resultados más precisos. Estas predicciones ofrecen una ubicación más rica que aumenta la conciencia del contexto. En la computación ubicua, las estrategias proactivas, como las predicciones del movimiento de los usuarios, ofrecen una solución para reducir el consumo de energía o para mejorar la experiencia de los usuarios.

7.1. Aplicaciones del método propuesto

Adicionalmente, esta tesis presenta a mayores una aplicación concreta del IPS desarrollado. En la aplicación, se ha desarrollado una nueva ontología para conectar el IPS y Europea, la plataforma digital de la UE para la gestión del patrimonio cultural. La ontología propuesta puede ser utilizada en un museo para mejorar la experiencia del usuario, y también para inferir conocimiento sobre objetos del patrimonio cultural utilizando el IPS, o los servicios ofrecidos por las distintas APIs de posicionamiento de Google.

El IPS presentado en esta tesis, requiere que el museo instale cámaras RGB-D y que los usuarios lleven teléfonos móviles con la aplicación desarrollada. El software puede recopilar la posición del usuario, las preferencias de acuerdo con su perfil, la orientación del esqueleto y otros datos con el fin de ofrecer información más precisa, definiendo rutas entre piezas de arte concretas. Los experimentos, detallados en el Capítulo 6 (Sección 6.1), se han llevado a cabo con datos obtenidos de los Museos Reales de Bélgica, que incluyen más de 6000 objetos. El camino óptimo, entre la posición del usuario y una pintura en el museo, se calcula utilizando dos tipos de algoritmos conocidos, Dijkstra o A*, y considera las salas como nodos de un grafo. Los experimentos se han llevado a cabo utilizando la API de localización desarrollada por Google. Se ha desarrollado una aplicación móvil y, aunque el sistema ha funcionado correctamente utilizando la señal WiFi y de telefonía móvil para las pinturas indicadas, la instalación de más puntos de acceso WiFi mejoraría el posicionamiento, al haber sido detectados pocos APs accesibles. El uso de ontologías es muy eficaz, ya que el sistema es flexible para establecer las relaciones o valores conocidos e inferir el conocimiento necesario.

7.2. Publicaciones

Los resultados obtenidos con la tesis han dado lugar a 3 artículos en publicaciones anglosajonas de alto impacto de los cuartiles Q1 y Q2 del JCR:

- “Indoor Positioning System using Depth Maps and Wireless Networks” [33] en publicación “Journal of Sensors”, Hindawi, Junio de 2016 (Factor de Impacto:

1.704 en SCI-JCR (Q2) en 2016), DOI: 10.1155/2016/2107872

- “A New Strategy for Navigation in Museums: Locating the User and Deciphering Egyptian Hieroglyphs” [130] en publicación “Sensors”, MDPI, Marzo de 2017 (Factor de Impacto: 2.677 en SCI-JCR (Q1) en 2016), DOI: 10.3390/s17030589
- “An Improved Indoor Positioning System Using RGB-D Cameras and Wireless Networks for Use in Complex Environments” [131] en publicación “Sensors”, MDPI, Octubre de 2017 (Factor de Impacto: 2.677 en SCI-JCR (Q1) en 2016), DOI: 10.3390/s17102391

Asimismo, se han presentado las siguientes publicaciones en congresos internacionales:

- “People Positioning System with Low Cost 3D Cameras and Wireless Devices for Indoor Environments” en el congreso “Robocity”, Madrid, Mayo de 2016.
- “Locating Multiple Camera Sensors and Wireless Access Points for a Generalized Indoor Positioning System” [132] publicado por “The Thirteenth International Conference on. Wireless and Mobile Communications”, Niza (Francia), ICWMC Julio de 2017.
- “A Semantic Approach to Enrich User Experience in Museums Through Indoor Positioning” [133] publicado por “International Conference on Ubiquitous Computing and Ambient Intelligence”, UCAMI, Filadelfia (EEUU), Noviembre de 2017.

Y también se ha conseguido la participación en otros congresos nacionales:

- “Localización de personas mediante cámaras RGB-D y redes inalámbricas” [134] publicado por el “Comité Nacional de Automática”, 2015. **Obtuvo el premio INFAIMON 2015, otorgado al mejor trabajo en visión artificial.** Congreso “Jornadas de Automática”, Bilbao, Septiembre de 2015.

- “Indoor Positioning Prediction System based on Wireless Networks and Depth Sensing Cameras” en el congreso “Jornadas de Automática”, Madrid, Septiembre de 2016.
- “Experiencias de Predicción para un Sistema de Posicionamiento de uso en Recintos de Interior basado en Cámaras 3D de Bajo Coste y Redes Inalámbricas” en las “Jornadas Nacionales de Robótica”, Valencia, Junio de 2017.
- “Mejoras en el algoritmo de posicionamiento en interiores mediante sensores RGB-D y redes WiFi” en el congreso “Jornadas de Automática”, Badajoz, Septiembre de 2018.

Finalmente, cabe destacar la aparición en diversos medios de prensa, como en el periódico *La Razón*, y televisión, como en el canal *Cuatro*, de la noticia sobre la publicación del artículo de traducción de jeroglíficos egipcios [130].

7.3. Trabajos futuros

Un trabajo futuro consistiría en utilizar cámaras RGB en lugar de cámaras RGB-D. Existen trabajos que estudian el posicionamiento de objetos a través de visión 3D estereoscópica en planos 2D [135, 136], trabajos que intentan reconstruir el posicionamiento 3D de individuos a través de métodos probabilísticos [137, 138], o de la trayectoria [139], o utilizando cámaras estáticas *Pan-Tilt-Zoom* (PTZ) [140], e incluso soluciones comerciales como TRACAB, utilizada en determinados deportes para seguir la trayectoria de los balones o jugadores [141]. Aunque los sistemas de múltiples cámaras normales ofrecen un error previsiblemente mayor que los obtenidos por cámaras RGB-D, sería interesante evaluar el sistema presentado en esta tesis con dichas tecnologías. El WPS podría mejorar el propio sistema de posicionamiento de cámaras normales.

Dentro de entornos industriales, se podría evaluar el uso del sistema de posicionamiento de cara a la mejora de la productividad y a la reducción de riesgos laborales.

También sería interesante evaluar la propia posición y orientación de los usuarios, para ofrecerles una mejor interacción con el sistema. En el sistema presentado de

conexión con Europeana, la orientación, por ejemplo, permitiría saber si el usuario está mirando una obra de arte concreta.

Dentro del problema de la predicción de movimientos, las cámaras RGB-D permiten obtener posibles obstáculos, siendo necesario combinar esta información con el sistema de predicción, para obtener una posición más precisa y evitar los posibles nuevos obstáculos estáticos y dinámicos.

Finalmente, aunque el emparejamiento de rutas presentado en este trabajo integra trayectorias de personas obtenidas mediante WPS y RGB-D, se plantea como trabajo futuro el estudio del algoritmo con rutas obtenidas mediante otros sistemas de posicionamiento, como pueden ser el GPS o las señales de las torres de telefonía móvil.

Glosario

- **AP:** Puntos de Acceso, del inglés *Access Point*. Habitualmente son *routers* WiFi.
- **BEIDOU:** Es el sistema de posicionamiento desarrollado por China. Utiliza satélites en órbita geostacionaria. Esto implica que el sistema no requiere una gran constelación de satélites. Esto limita su cobertura sobre la tierra a la zona visible por los satélites, China en este caso. BEIDOU calcula las coordenadas únicamente con dos satélites y una estación en tierra.
- **Crowdsensing:** es una ampliación de *fingerprinting* que recolecta continuamente los datos obtenidos por los teléfonos móviles que llevan los usuarios con el fin de actualizar la base de datos de posicionamiento.
- **Dead Reckoning Systems:** utiliza datos de los acelerómetros de los dispositivos para obtener el desplazamiento inercial así como el sensor de campo magnético para obtener la dirección.
- **Depth map:** Un mapa de profundidad es una matriz bidimensional que contiene valores de distancias. En todos los aspectos, un mapa de profundidad es equivalente a una imagen, aunque en vez de almacenar valores de color, almacena distancias.
- **Fingerprinting:** El objetivo de esta técnica es generar un mapa del entorno con las intensidades de señal WiFi, *Received Signal Strength Indication* (RSSI), registradas en diversos puntos del área.
- **Galileo:** Galileo es el sistema de posicionamiento desarrollado por la Unión Europea, siendo interoperable con los sistemas GPS y GLONASS. El usuario

puede calcular su posición con un receptor que utiliza satélites de distintas constelaciones. Al ofrecer dos frecuencias en su versión estándar, Galileo ofrece una precisión del orden de 1 metro para el sistema gratuito, y hasta 1 cm en el de pago. Galileo es un sistema que ofrece mayor precisión en el posicionamiento en interiores.

- **GLONASS:** Sistema Global de Navegación por Satélite, del inglés *Global Navigation Satellite System*. Es el sistema de posicionamiento desarrollado por la Unión Soviética, siendo hoy administrado por la Federación Rusa.
- **GNSS:** Sistema Global de Navegación por Satélite, del inglés *Global Navigation Satellite System*.
- **GPS:** Sistemas de Posicionamiento Global, del inglés *Global Positioning Systems*. Funciona mediante una red de 24 satélites en órbita sobre la Tierra. Estos satélites operan a 20200 km de altura, cubriendo trayectorias sincronizadas sobre la superficie de la Tierra.
- **IPS:** Sistemas de Posicionamiento en Interiores, del inglés *Indoor Positioning Systems*.
- **RGB:** Imagen Red-Green-Blue. El sistema RGB permite representar imágenes utilizando tres colores diferentes para cada punto: Rojo-Verde-Azul. La combinación de distintas intensidades de dichos colores permite generar el abanico completo de los colores que habitualmente muestra una imagen.
- **RGB-D:** Son imágenes formadas por la propia imagen RGB y el mapa de profundidad o *depth map*.

Bibliografía

- [1] Gabriel Deak, Kevin Curran, y Joan Condell. A survey of active and passive indoor localisation systems. *Computer Communications*, 35(16):1939–1954, 2012.
- [2] Hui Liu, Houshang Darabi, Pat Banerjee, y Jing Liu. Survey of wireless indoor positioning techniques and systems. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 37(6):1067–1080, 2007.
- [3] IP Tharindu Weerasinghe, Janaka Y Ruwanpura, Jeffrey E Boyd, y Ayman F Habib. Application of microsoft kinect sensor for tracking construction workers. In *Construction Research Congress*, pages 858–867, 2012.
- [4] Matteo Munaro, Filippo Basso, y Emanuele Menegatti. Opentrack: Open source multi-camera calibration and people tracking for rgb-d camera networks. *Robotics and Autonomous Systems*, 75:525–538, 2016.
- [5] Yoshiaki Nakano, Katsunobu Izutsu, Kiyoshi Tajitsu, Katsutoshi Kai, y Takeo Tatsumi. Kinect positioning system (kps) and its potential applications. In *International Conference on Indoor Positioning and Indoor Navigation*, volume 13, page 15th, 2012.
- [6] Muhamad Risqi Utama Saputra, W Widyawan, Guntur Dharma Putra, y Paulus Insap Santosa. Indoor human tracking application using multiple depth-cameras. In *Advanced Computer Science and Information Systems (ICACISIS), 2012 International Conference on*, pages 307–312. IEEE, 2012.
- [7] Kanad K Biswas y Saurav Kumar Basu. Gesture recognition using micro-

- soft kinect®. In *Automation, Robotics and Applications (ICARA), 2011 5th International Conference on*, pages 100–103. IEEE, 2011.
- [8] Jeffrey Hightower y Gaetano Borriello. Location systems for ubiquitous computing. *Computer*, 34(8):57–66, 2001.
- [9] Hakan Koyuncu y Shuang Hua Yang. A survey of indoor positioning and object locating systems. *IJCSNS International Journal of Computer Science and Network Security*, 10(5):121–128, 2010.
- [10] Navid Fallah, Ilias Apostolopoulos, Kostas Bekris, y Eelke Folmer. Indoor human navigation systems: A survey. *Interacting with Computers*, page iws010, 2013.
- [11] Kalyan Subbu, Chi Zhang, Jun Luo, y Athanasios Vasilakos. Analysis and status quo of smartphone-based indoor localization systems. *Wireless Communications, IEEE*, 21(4):106–112, 2014.
- [12] Bernhard Hofmann-Wellenhof, Herbert Lichtenegger, y Elmar Wasle. *GNSS—global navigation satellite systems: GPS, GLONASS, Galileo, and more*. Springer Science & Business Media, 2007.
- [13] Elliott Kaplan y Christopher Hegarty. *Understanding GPS: principles and applications*. Artech house, 2005.
- [14] Dimitris E Manolakis. Efficient solution and performance analysis of 3-d position estimation by trilateration. *IEEE Transactions on Aerospace and Electronic systems*, 32(4):1239–1248, 1996.
- [15] Mikkel Baun Kjærgaard, Henrik Blunck, Torben Godsk, Thomas Toftkjær, Dan Lund Christensen, y Kaj Grønbæk. Indoor positioning using gps revisited. In *International conference on pervasive computing*, pages 38–56. Springer, 2010.
- [16] Bo Yang, Yajun Xu, Jianjun Luo, y Yong Bao. Indoor positioning performance of galileo signal. In *Second International Conference on Spatial Information*

- Technology*, pages 679576–679576. International Society for Optics and Photonics, 2007.
- [17] Mohamed Ibrahim y Moustafa Youssef. Cellsense: An accurate energy-efficient gsm positioning system. *IEEE Transactions on Vehicular Technology*, 61(1): 286–296, 2012.
- [18] Alex Varshavsky, Eyal De Lara, Jeffrey Hightower, Anthony LaMarca, y Veljo Otsason. Gsm indoor localization. *Pervasive and Mobile Computing*, 3(6): 698–720, 2007.
- [19] Wei Liu, Yongle Chen, Yongping Xiong, Limin Sun, y Hongsong Zhu. Optimization of sampling cell size for fingerprint positioning. *International Journal of Distributed Sensor Networks*, 2014, 2014.
- [20] Eladio Martin, Oriol Vinyals, Gerald Friedland, y Ruzena Bajcsy. Precise indoor localization using smart phones. In *Proceedings of the international conference on Multimedia*, pages 787–790. ACM, 2010.
- [21] Suining He y S-H Gary Chan. Wi-fi fingerprint-based indoor positioning: Recent advances and comparisons. *IEEE Communications Surveys & Tutorials*, 18(1):466–490, 2016.
- [22] Michael Quan, Eduardo Navarro, y Benjamin Peuker. Wi-fi localization using rssi fingerprinting. 2010.
- [23] Paramvir Bahl y Venkata N Padmanabhan. RADAR: An in-building RF-based user location and tracking system. In *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 2, pages 775–784. Ieee, 2000.
- [24] Beom-Ju Shin, Kwang-Won Lee, Sun-Ho Choi, Joo-Yeon Kim, Woo Jin Lee, y Hyung Seok Kim. Indoor wifi positioning system for android-based smartphone. In *Intl Conference on Information and Communication Technology Convergence*, pages 319–320, 2010.

- [25] Yanying Gu, Anthony Lo, y Ignas Niemegeers. A survey of indoor positioning systems for wireless personal networks. *IEEE Communications surveys & tutorials*, 11(1):13–32, 2009.
- [26] Xiufang Feng, Zhanqiang Gao, Mian Yang, y Shibo Xiong. Fuzzy distance measuring based on rssi in wireless sensor network. In *Intelligent System and Knowledge Engineering, 2008. ISKE 2008. 3rd International Conference on*, volume 1, pages 395–400. IEEE, 2008.
- [27] Janja Svečko, Marko Malajner, y Dušan Gleich. Distance estimation using rssi and particle filter. *ISA transactions*, 55:275–285, 2015.
- [28] Carlos Kornuta, Nelson Acosta, y Juan Manuel Toloza. Posicionamiento wifi con variaciones de fingerprint. In *XVIII Congreso Argentino de Ciencias de la Computación*, 2013.
- [29] G Deak, Kevin Curran, y Joan Condell. Filters for rssi-based measurements in a device-free passive localisation scenario. *Image Processing & Communications*, 15:23–34, 2010.
- [30] Mauricio Cáceres, Francesco Sottile, y Maurizio A Spirito. Wlan-based real time vehicle locating system. In *Vehicular Technology Conference, 2009. VTC Spring 2009. IEEE 69th*, pages 1–5. IEEE, 2009.
- [31] Shiu Kumar y Seong Ro Lee. Localization with rssi values for wireless sensor networks: an artificial neural network approach. In *International Electronic Conference on Sensors and Applications*, volume 1. Multidisciplinary Digital Publishing Institute, 2014.
- [32] Mohd Nizam Husen y Sukhan Lee. Indoor human localization with orientation using wifi fingerprinting. In *Proceedings of the 8th International Conference on Ubiquitous Information Management and Communication*, page 109. ACM, 2014.
- [33] Jaime Duque Domingo, Carlos Cerrada, Enrique Valero, y JA Cerrada. Indoor positioning system using depth maps and wireless networks. *Journal of Sensors*, 2016, 2016.

- [34] Sang Yup Lee, Byung-Cheol Min, Dong Hoe Kim, Jae Seok Yoon, y Dong Han Kim. Passive rfid positioning system using rf power control. In *Robot Intelligence Technology and Applications 2012*, pages 845–853. Springer, 2013.
- [35] Rainer Mautz y Sebastian Tilch. Survey of optical indoor positioning systems. In *IPIN*, pages 1–7, 2011.
- [36] Tsung-Han Tsai, Chih-Hao Chang, y Shih-Wei Chen. Vision based indoor positioning for intelligent buildings. In *Intelligent Green Building and Smart Grid (IGBSG), 2016 2nd International Conference on*, pages 1–4. IEEE, 2016.
- [37] Ye-Sheng Kuo, Pat Pannuto, Ko-Jen Hsiao, y Prabal Dutta. Luxapose: Indoor positioning with mobile phones and visible light. In *Proceedings of the 20th annual international conference on Mobile computing and networking*, pages 447–458. ACM, 2014.
- [38] Martin Werner, Carsten Hahn, y Lorenz Schauer. Deepmovips: Visual indoor positioning using transfer learning. In *Indoor Positioning and Indoor Navigation (IPIN), 2016 International Conference on*, pages 1–7. IEEE, 2016.
- [39] Corina Kim Schindhelm. Evaluating slam approaches for microsoft kinect. In *Proc. 2011 The Eighth International Conference on Wireless and Mobile Communications (ICWMC 2012), Venice*, pages 402–407, 2012.
- [40] Giorgio Grisetti, Cyrill Stachniss, Slawomir Grzonka, y Wolfram Burgard. A tree parameterization for efficiently computing maximum likelihood maps using gradient descent. In *Robotics: Science and Systems*, pages 27–30, 2007.
- [41] Hiroyuki Takizawa, Satarou Yamaguchi, Masahiro Aoyagi, Nobuo Ezaki, y Seiya Mizuno. Kinect cane: object recognition aids for the visually impaired. In *Human System Interaction (HSI), 2013 The 6th International Conference on*, pages 473–478. IEEE, 2013.
- [42] Loïc Sevrin, Norbert Noury, Nacer Abouchi, Fabrice Jumel, Bertrand Massot, y Jacques Saraydaryan. Characterization of a multi-user indoor positioning system based on low cost depth vision (kinect) for monitoring human activity

- in a smart home. In *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 5003–5007. IEEE, 2015.
- [43] Pan Jing y Guan Ye-peng. Human-computer interaction using pointing gesture based on an adaptive virtual touch screen. *International Journal of Signal Processing, Image Processing and Pattern Recognition*, 6(4):81–92, 2013.
- [44] Jarosław Gośliński, Piotr Owczarek, y Dominik Rybarczyk. The use of kinect sensor to control manipulator with electrohydraulic servodrives. *Pomiary Automatyka Robotyka*, 17:481–486, 2013.
- [45] Genzhi Ye, Yebin Liu, Yue Deng, Nils Hasler, Xiangyang Ji, Qionghai Dai, y Christian Theobalt. Free-viewpoint video of human actors using multiple handheld kinects. *Cybernetics, IEEE Transactions on*, 43(5):1370–1382, 2013.
- [46] Paul Viola y Michael J Jones. Robust real-time face detection. *International journal of computer vision*, 57(2):137–154, 2004.
- [47] Gareth J Edwards, Timothy F Cootes, y Christopher J Taylor. Face recognition using active appearance models. In *European conference on computer vision*, pages 581–595. Springer, 1998.
- [48] Wenyi Zhao, Rama Chellappa, P Jonathon Phillips, y Azriel Rosenfeld. Face recognition: A literature survey. *ACM computing surveys (CSUR)*, 35(4):399–458, 2003.
- [49] Zhongliang Deng, Yanpei Yu, Xie Yuan, Neng Wan, y Lei Yang. Situation and development tendency of indoor positioning. *China Communications*, 10(3): 42–55, 2013.
- [50] Robert Harle. A survey of indoor inertial positioning systems for pedestrians. *IEEE Communications Surveys and Tutorials*, 15(3):1281–1293, 2013.
- [51] Juraj Machaj y Peter Brida. Impact of optimization algorithms on hybrid indoor positioning based on gsm and wi-fi signals. *Concurrency and Computation: Practice and Experience*, 2016.

- [52] Hui-Kai Su, Zi-Xiang Liao, Chih-Hsueh Lin, y Tse-Min Lin. A hybrid indoor-positioning mechanism based on bluetooth and wifi communications for smart mobile devices. In *2015 International Symposium on Bioelectronics and Bioinformatics (ISBB)*, pages 188–191, 2015.
- [53] Apostolia Papapostolou y Hakima Chaouchi. Integrating rfid and wlan for indoor positioning and ip movement detection. *Wireless Networks*, 18(7):861–879, 2012.
- [54] Ching-Sheng Wang y Chien-Liang Chen. Rfid-based and kinect-based indoor positioning system. In *Wireless Communications, Vehicular Technology, Information Theory and Aerospace & Electronic Systems (VITAE), 2014 4th International Conference on*, pages 1–4. IEEE, 2014.
- [55] Ronald Clark, Niki Trigoni, y Andrew Markham. Robust vision-based indoor localization. In *Proceedings of the 14th International Conference on Information Processing in Sensor Networks*, pages 378–379. ACM, 2015.
- [56] Frédéric Evennou y François Marx. Advanced integration of wifi and inertial navigation systems for indoor mobile positioning. *Eurasip journal on applied signal processing*, 2006:164–164, 2006.
- [57] Google Maps Geolocation API. Google Maps Application Programming Interface, . URL <http://developers.google.com/maps/documentation/geolocation/intro>.
- [58] Pawel Wilk, Jaroslaw Karciarz, y Jaroslaw Swiatek. Indoor radio map maintenance by automatic annotation of crowdsourced wi-fi fingerprints. In *Indoor Positioning and Indoor Navigation (IPIN), 2015 International Conference on*, pages 1–8. IEEE, 2015.
- [59] Piotr Mirowski, Ravishankar Palaniappan, y Tin Kam Ho. Depth camera slam on a low-cost wifi mapping robot. In *Technologies for Practical Robot Applications (TePRA), 2012 IEEE International Conference on*, pages 1–6. IEEE, 2012.

- [60] Joydeep Biswas y Manuela Veloso. Multi-sensor mobile robot localization for diverse environments. In *RoboCup 2013: Robot World Cup XVII*, pages 468–479. Springer, 2014.
- [61] AmirHosein GhaffarianHoseini, Nur Dalilah Dahlan, Umberto Berardi, Ali GhaffarianHoseini, y Nastaran Makaremi. The essence of future smart houses: From embedding ict to adapting to sustainability principles. *Renewable and Sustainable Energy Reviews*, 24:593–607, 2013.
- [62] Koen Kok, Stamatis Karnouskos, David Nestle, Aris Dimeas, Anke Weidlich, Cor Warmer, Philipp Strauss, Britta Buchholz, Stefan Drenkard, Nikos Hatzargyriou, et al. Smart houses for a smart grid. In *Electricity Distribution-Part 1, 2009. CIREN 2009. 20th International Conference and Exhibition on*, pages 1–4. IET, 2009.
- [63] George Y Liu y GQ Maguire. A predictive mobility management algorithm for wireless mobile computing and communications. In *Universal Personal Communications. 1995. Record., 1995 Fourth IEEE International Conference on*, pages 268–272. IEEE, 1995.
- [64] Sami Tabbane. An alternative strategy for location tracking. *Selected Areas in Communications, IEEE Journal on*, 13(5):880–892, 1995.
- [65] John Neter, Michael H Kutner, Christopher J Nachtsheim, y William Wasserman. *Applied linear statistical models*, volume 4. Irwin Chicago, 1996.
- [66] Romer Rosales y Stan Sclaroff. Improved tracking of multiple humans with trajectory prediction and occlusion modeling. In *IEEE CVPR workshop on the Interpretation of Visual Motion*, 1998.
- [67] Pierre Payeur, Hoang Le-Huy, y Clement M Gosselin. Trajectory prediction for moving objects using artificial neural networks. *Industrial Electronics, IEEE Transactions on*, 42(2):147–158, 1995.
- [68] Tong Liu, Paramvir Bahl, y Imrich Chlamtac. Mobility modeling, location tracking, and trajectory prediction in wireless atm networks. *Selected Areas in Communications, IEEE Journal on*, 16(6):922–936, 1998.

- [69] Pubudu N Pathirana, Andrey V Savkin, y Sanjay Jha. Location estimation and trajectory prediction for cellular networks with mobile base stations. *Vehicular Technology, IEEE Transactions on*, 53(6):1903–1913, 2004.
- [70] Gökhan Yavaş, Dimitrios Katsaros, Özgür Ulusoy, y Yannis Manolopoulos. A data mining approach for location prediction in mobile environments. *Data & Knowledge Engineering*, 54(2):121–146, 2005.
- [71] Xiong Liu y Hassan A Karimi. Location awareness through trajectory prediction. *Computers, Environment and Urban Systems*, 30(6):741–756, 2006.
- [72] Akinori Asahara, Kishiko Maruyama, Akiko Sato, y Kouichi Seto. Pedestrian-movement prediction based on mixed markov-chain model. In *Proceedings of the 19th ACM SIGSPATIAL international conference on advances in geographic information systems*, pages 25–33. ACM, 2011.
- [73] Vassileios Tsetsos, Christos Anagnostopoulos, Panayotis Kikiras, P Hasiotis, y Stathes Hadjiefthymiades. A human-centered semantic navigation system for indoor environments. In *Pervasive Services, 2005. ICPS'05. Proceedings. International Conference on*, pages 146–155. IEEE, 2005.
- [74] Tamás Matuszka, Gergő Gombos, y Attila Kiss. A new approach for indoor navigation using semantic webtechnologies and augmented reality. In *International Conference on Virtual, Augmented and Mixed Reality*, pages 202–210. Springer, 2013.
- [75] Patrick M Dudas, Mahsa Ghafourian, y Hassan A Karimi. ONALIN: Ontology and algorithm for indoor routing. In *Mobile Data Management: Systems, Services and Middleware, 2009. MDM'09. Tenth International Conference on*, pages 720–725. IEEE, 2009.
- [76] Kangjae Lee, Jiyeong Lee, y Mei-Po Kwan. Location-based service using ontology-based semantic queries: A study with a focus on indoor activities in a university context. *Computers, Environment and Urban Systems*, 62:41–52, 2017.

- [77] Open Geospatial Consortium et al. Ogc geosparql-a geographic query language for rdf data. *OGC Candidate Implementation Standard*, 2, 2012.
- [78] Gangani Geethika Wijewardena, Maria Vasardani, y Stephan Winter. Towards indoor localization and navigation independent of sensor based technologies. In *Proceedings of the Eighth ACM SIGSPATIAL International Workshop on Indoor Spatial Awareness*, pages 19–26. ACM, 2016.
- [79] Michael Worboys. Modeling indoor space. In *Proceedings of the 3rd ACM SIGSPATIAL International Workshop on Indoor Spatial Awareness*, pages 1–6. ACM, 2011.
- [80] Liping Yang y Michael Worboys. A navigation ontology for outdoor-indoor space:(work-in-progress). In *Proceedings of the 3rd ACM SIGSPATIAL international workshop on indoor spatial awareness*, pages 31–34. ACM, 2011.
- [81] Gi Hyun Lim, Il Hong Suh, y Hyowon Suh. Ontology-based unified robot knowledge for service robots in indoor environments. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 41(3):492–509, 2011.
- [82] Marcus Goetz y Alexander Zipf. Extending openstreetmap to indoor environments: bringing volunteered geographic information to the next level. *Urban and Regional Data Management: Udms Annual 2011*, pages 47–58, 2011.
- [83] Thomas R Gruber. Toward principles for the design of ontologies used for knowledge sharing. *International journal of human-computer studies*, 43(5-6): 907–928, 1995.
- [84] T Guber. A translational approach to portable ontologies. *Knowledge Acquisition*, 5(2):199–229, 1993.
- [85] World Wide Web Consortium et al. Owl 2 web ontology language document overview. 2012.
- [86] Franz Baader. *The description logic handbook: Theory, implementation and applications*. Cambridge university press, 2003.

- [87] Ora Lassila y Ralph R Swick. Resource description framework (rdf) model and syntax specification. 1999.
- [88] Jarrett Webb y James Ashley. *Beginning Kinect Programming with the Microsoft Kinect SDK*. Apress, 2012.
- [89] Barak Freedman, Alexander Shpunt, Meir Machline, y Yoel Arieli. Depth mapping using projected patterns, April 3 2012. US Patent 8,150,142.
- [90] Jacky CK Chow y Derek D Lichti. Photogrammetric bundle adjustment with self-calibration of the primesense 3d camera technology: Microsoft kinect. *IEEE Access*, 1:465–474, 2013.
- [91] Angelos Barmoutis. Tensor body: Real-time reconstruction of the human body and avatar synthesis from rgb-d. *Cybernetics, IEEE Transactions on*, 43 (5):1347–1356, 2013.
- [92] Jamie Shotton, Toby Sharp, Alex Kipman, Andrew Fitzgibbon, Mark Finocchio, Andrew Blake, Mat Cook, y Richard Moore. Real-time human pose recognition in parts from single depth images. *Communications of the ACM*, 56(1):116–124, 2013.
- [93] Intel Real Sense. Intel. URL [www.intel.es. https://www.intel.es/content/www/es/es/architecture-and-technology/realsense-overview.html](http://www.intel.es/content/www/es/es/architecture-and-technology/realsense-overview.html) (último acceso: 15 de Abril de 2018).
- [94] Asus Xtion PRO Live. Asus. URL [www.asus.com. https://www.asus.com/es/3D-Sensor/Xtion_PRO_LIVE/](http://www.asus.com/es/3D-Sensor/Xtion_PRO_LIVE/) (último acceso: 15 de Abril de 2018).
- [95] Structure Sensor. Occipital. URL occipital.com. https://structure.io/ (último acceso: 15 de Abril de 2018).
- [96] Peregrine 3D Flash LIDAR Vision System. Advanced Scientific Concepts. URL [www.advancedscientificconcepts.com. http://www.advancedscientificconcepts.com/products/Peregrine.html](http://www.advancedscientificconcepts.com/products/Peregrine.html) (último acceso: 12 de Mayo de 2016).

- [97] IFMO 3D. PMD Technologies. URL pmdtec.com. <https://pmdtec.com/ifmo3d/ifm-03D303-PMD-3D-Camera-e-15-n.pdf> (último acceso: 15 de Abril de 2018).
- [98] Kinect v2. Microsoft. URL microsoft.com. <https://developer.microsoft.com/en-us/windows/kinect> (último acceso: 15 de Abril de 2018).
- [99] Open Kinect. Openkinect project, 2011.
- [100] Kouros Khoshelham. Accuracy analysis of kinect depth data. In *ISPRS workshop laser scanning*, volume 38, page W12, 2011.
- [101] Carlo Dal Mutto, Pietro Zanuttigh, y Guido M Cortelazzo. *Time-of-flight cameras and Microsoft KinectTM*. Springer Science & Business Media, 2012.
- [102] Jeff Kramer, Nicolas Burrus, Florian Echtler, Herrera C Daniel, y Matt Parker. *Hacking the Kinect*, volume 268. Springer, 2012.
- [103] Stephen A Joseph, Ronald V Rieger, Danilo Micu, y Miguel Estrella. Android controlled oven, October 24 2013. US Patent App. 13/869,006.
- [104] Android Location API. Android Developers Application Programming Interface. URL <http://developer.android.com/reference/android/location/package-summary.html>.
- [105] Android Studio. Google. URL android.com. <https://developer.android.com/studio/index.html> (último acceso: 15 de Abril de 2018).
- [106] Catherine T Lawson, SS Ravi, y Jeong-Hyon Hwang. Compression and mining of gps trace data: New techniques and applications. Technical report, Technical Report. Region II University Transportation Research Center, 2011.
- [107] Kwok-Yun Yeung, Tsz-Ho Kwok, y Charlie CL Wang. Improved skeleton tracking by duplex kinects: A practical approach for real-time applications. *Journal of Computing and Information Science in Engineering*, 13(4):041007, 2013.
- [108] Egon Balas. An additive algorithm for solving linear programs with zero-one variables. *Operations Research*, 13(4):517–546, 1965.

- [109] Theodoros Anagnostopoulos, Christos Anagnostopoulos, Stathes Hadjiefthymiades, Miltos Kyriakakos, y Alexandros Kalousis. Predicting the location of mobile users: a machine learning approach. In *Proceedings of the 2009 international conference on Pervasive services*, pages 65–72. ACM, 2009.
- [110] Lucian Vintan, Arpad Gellert, Jan Petzold, y Theo Ungerer. Person movement prediction using neural networks. 2006.
- [111] Sherif Akoush y Ahmed Sameh. Mobile user movement prediction using bayesian learning for neural networks. In *Proceedings of the 2007 international conference on Wireless communications and mobile computing*, pages 191–196. ACM, 2007.
- [112] Jan Petzold, Andreas Pietzowski, Faruk Bagci, Wolfgang Trumler, y Theo Ungerer. Prediction of indoor movements using bayesian networks. In *International Symposium on Location-and Context-Awareness*, pages 211–222. Springer, 2005.
- [113] Arpad Gellert y Lucian Vintan. Person movement prediction using hidden markov models. *Studies in Informatics and control*, 15(1):17, 2006.
- [114] Robert Grover Brown, Patrick YC Hwang, et al. *Introduction to random signals and applied Kalman filtering*, volume 3. Wiley New York, 1992.
- [115] Jan Petzold, Faruk Bagci, Wolfgang Trumler, y Theo Ungerer. Comparison of different methods for next location prediction. In *European Conference on Parallel Processing*, pages 909–918. Springer, 2006.
- [116] Martin T Hagan, Howard B Demuth, Mark H Beale, et al. *Neural network design*, volume 20. Pws Pub. Boston, 1996.
- [117] Francisco Curbera, Matthew Duftler, Rania Khalaf, William Nagy, Nirmal Mukhi, y Sanjiva Weerawarana. Unraveling the web services web: an introduction to soap, wsdl, and uddi. *IEEE Internet computing*, 6(2):86–93, 2002.
- [118] Antoine Isaac y Bernhard Haslhofer. Europeana linked open data (data.europeana.eu). *Semantic Web*, 4(3):291–297, 2013.

- [119] Eric Prud'hommeaux, Carlos Buil-Aranda, et al. SPARQL 1.1 federated query. *W3C Recommendation*, 21, 2013.
- [120] Andy Seaborne. ARQ-A SPARQL Processor for Jena. *Obtained through the Internet: <http://jena.sourceforge.net/ARQ>*, 2010.
- [121] Apache Jena Server. Apache. URL <http://jena.apache.org>.
- [122] Google Play Services Location APIs. Google Play Services Application Programming Interface, . URL <http://developer.android.com/training/location/retrieve-current.html>.
- [123] Android Location Strategies. Android Developers Application Programming Interface. URL <http://developer.android.com/guide/topics/location/strategies.html>.
- [124] José de Mendoza et al. *Memoria sobre algunos métodos nuevos de calcular la longitud por las distancias lunares: y aplicación de su teórica a la solución de otros problemas de navegación*. En la Imprenta real, 1795.
- [125] Thaddeus Vincenty. Direct and inverse solutions of geodesics on the ellipsoid with application of nested equations. *Survey review*, 23(176):88–93, 1975.
- [126] Leigh Dodds. Sparql geo extensions, 2006.
- [127] Greg Williams. Extensible SPARQL Functions with Embedded Javascript. In *SFSW*, 2007.
- [128] Edsger W Dijkstra. A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1):269–271, 1959.
- [129] Peter E Hart, Nils J Nilsson, y Bertram Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE transactions on Systems Science and Cybernetics*, 4(2):100–107, 1968.
- [130] Jaime Duque-Domingo, Pedro Javier Herrera, Enrique Valero, y Carlos Cerrada. Deciphering egyptian hieroglyphs: Towards a new strategy for navigation in museums. *Sensors*, 17(3):589, 2017.

- [131] Jaime Duque Domingo, Carlos Cerrada, Enrique Valero, y Jose A Cerrada. An improved indoor positioning system using rgb-d cameras and wireless networks for use in complex environments. *Sensors*, 17(10):2391, 2017.
- [132] Jaime Duque Domingo, Carlos Cerrada, JA Cerrada, y Enrique Valero. Locating multiple camera sensors and wireless access points for a generalized indoor positioning system. *ICWMC 2017*, page 59, 2017.
- [133] Jaime Duque Domingo, Carlos Cerrada, Enrique Valero, y JA Cerrada. A semantic approach to enrich user experience in museums through indoor positioning. In *International Conference on Ubiquitous Computing and Ambient Intelligence*, pages 612–623. Springer, 2017.
- [134] Jaime Duque Domingo, Carlos Cerrada Somolinos, y Enrique Valero. Localización de personas mediante camaras rgb-d y redes inalámbricas.
- [135] Dimitar Valkov, Frank Steinicke, Gerd Bruder, y Klaus Hinrichs. 2d touching of 3d stereoscopic objects. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1353–1362. ACM, 2011.
- [136] Eric Marchand, Patrick Bouthemy, y François Chaumette. A 2d–3d model-based approach to real-time visual tracking. *Image and Vision Computing*, 19(13):941–955, 2001.
- [137] Hedvig Sidenbladh, Michael J Black, y David J Fleet. Stochastic tracking of 3d human figures using 2d image motion. In *European conference on computer vision*, pages 702–718. Springer, 2000.
- [138] Hui-Huang Hsu, Wei-Min Yang, y Timothy K Shih. People tracking in a multi-camera environment. In *Conference Anthology, IEEE*, pages 1–4. IEEE, 2013.
- [139] João Dias y Pedro Mendes Jorge. People tracking with multi-camera system. In *Proceedings of the 9th International Conference on Distributed Smart Cameras*, pages 181–186. ACM, 2015.

- [140] Scott Stillman, Rawesak Tanawongsuwan, y Irfan Essa. Tracking multiple people with multiple cameras. In *International Conference on Audio-and Video-based Biometric Person Authentication*, 1999.
- [141] Jiwon Lee, Do-Won Nam, JungSoo Lee, Sungwon Moon, Kyungill Kim, y Hwon Kim. A study on composition of context-based soccer analysis system. In *Advanced Communication Technology (ICACT), 2017 19th International Conference on*, pages 886–889. IEEE, 2017.