







## DECLARACIÓN DE AUTORÍA Y ORIGINALIDAD

(Este documento debe acompañar al Trabajo Fin de Grado (TFG)/Trabajo Fin de Máster (TFM) cuando sea depositado para su evaluación).

D./D<sup>a</sup>. ZILIA MUÑOZ MARZO

con nº de DNI 73204684-P en aplicación de lo dispuesto en el art.

14 (Derechos de autor) del Acuerdo de 11 de septiembre de 2014, del Consejo de Gobierno, por el que se aprueba el Reglamento de los TFG y TFM de la Universidad de Zaragoza,

Declaro que el presente Trabajo de Fin de (Grado/Máster)  
INGENIERÍA ELECTRÓNICA Y AUTOMÁTICA, (Título del Trabajo)

RECONOCIMIENTO Y SEGUIMIENTO DE PERSONAS MEDIANTE UN SENSOR  
RGB-D EN UNA PLATAFORMA ROBÓTICA MÓVIL.

\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

es de mi autoría y es original, no habiéndose utilizado fuente sin ser citada debidamente.

Zaragoza, NOVIEMBRE DE 2016

Fdo: ZILIA MUÑOZ MARZO



# Reconocimiento Y Seguimiento De Personas Mediante Un Sensor RGB-D En Una Plataforma Robótica Móvil

## RESUMEN

Este proyecto presenta un sistema de detección y seguimiento de personas desde un robot móvil. El objetivo es aplicar este tipo de sistemas en entornos públicos donde el robot busque a personas a las cuales acercarse para ofrecer algún tipo de servicio o información.

Para la búsqueda de personas se ha diseñado un sistema de reconocimiento visual utilizando un sensor RGB-d. También se ha diseñado un sistema de seguimiento de esa persona seleccionada como objetivo, que consiste en acercarse a la misma. El reconocimiento se basa en las herramientas proporcionadas en las librerías de openCV. Este trabajo estudia distintas alternativas, y discute cuales se han elegido, porque y que cambios se han llevado a cabo en configuración. Después del reconocimiento de todas las personas que pueda haber en el campo de visión de nuestro robot, se debe elegir cuál de ellas se seguirá. Para ello se ha realizado un algoritmo para evaluar y ordenar las hipótesis detectadas. Por último, se seguirá a la persona seleccionada usando una plataforma robótica móvil a la que se le han proporcionado las coordenadas de la persona detectada en el punto anterior. El proceso de reconocimiento, elección y selección se repetirá hasta que el robot consigue acercarse a cierta persona, es decir, si la persona se está moviendo, el robot la seguirá.

Para poder alcanzar el objetivo deseado en primer lugar ha sido necesario familiarizarse con el entorno de trabajo openCV sobre eclipse, sobre ROS, para lo cual se han usado diferentes tutoriales proporcionados por el sitio web oficial de openCV. A continuación, se han tenido que evaluar los distintos reconocedores que posee openCV y sus parámetros, para evaluar el coste y calidad de los resultados de las distintas opciones. Estos experimentos se han realizado utilizando tanto secuencias públicas utilizadas en trabajos relacionados, como datos propios capturados para realizar pruebas de seguimiento en el entorno y con el robot disponible para este proyecto. También se han realizado experimentos de integración para comprobar el funcionamiento en tiempo real de todo el sistema con la plataforma robótica móvil en distintos escenarios.



# Índice

1. Introducción .....	1
1.1. Motivación Y Contexto.....	1
1.2. Objetivos Y Tareas.....	1
1.3. Trabajos Relacionados.....	3
1.3.1. Técnicas De Detección De Personas .....	3
1.3.2. Seguimiento o Tracking.....	5
1.3.3. Tecnologías Utilizadas.....	6
1.4. Organización Del Proyecto.....	7
2. Sistema De Detección Y Seguimiento .....	8
2.1. Resumen Del Sistema Realizado.....	8
2.2. Detección De Personas.....	9
2.3. Elección Y Seguimiento Del Target.....	10
3. Evaluación Y Experimentos .....	13
3.1. Evaluación del Sistema de Reconocimiento de Personas .....	14
3.1.1. Comparación de Métodos Disponibles en OpenCV .....	14
3.1.2. Variación de Parámetros en los Modelos Fullbody Y Upperbody.....	15
3.2. Evaluación Del Sistema De Elección y Seguimiento Del Objetivo .....	20
3.2.1. Elección del objetivo (persona).....	20
3.2.2. Error en la Estimación de Profundidad .....	21
3.3. Evaluación Del Sistema Completo .....	23
3.3.1. Experimentos de integración 1 .....	24
3.3.2. Experimentos Detección de Objetivo y Acercamiento Iterativo .....	25
3.3.3. Experimentos Finales .....	26
4. Conclusiones .....	28
4.1. Conclusiones del Prototipo y Trabajo Futuro.....	28
4.2. Conclusiones personales .....	28
Anexo I.....	29
Anexo II.....	30

Anexo III.....	31
Anexo IV .....	32
1. Evaluación Del Sistema de Reconocimiento de Personas .....	32
2. Evaluación Del Sistema de Elección Y Seguimiento Del Target .....	38
3. Evaluación Del Sistema Completo .....	45
Bibliografía.....	47

Figura 1. Robots móviles circulando por un museo (a). Plataforma móvil en guiado de museos (b). .....	1
Figura 2. Detección de personas en 2D en un entorno sencillo sin muchas oclusiones ni objetos móviles. 3	3
Figura 3. Detección de personas en 3D en entornos con un nivel alto de oclusiones y de densidad de elementos móviles. ....	3
Figura 4. Ejemplo de reconocimiento de caras con las librerías de OpenCV. ....	4
Figura 5. Ejemplo de aplicaciones de tracking. (a) seguimiento de coches. (b) Seguimiento de una línea trazada en el suelo. ....	5
Figura 6. TurtleBot utilizado en este experimento. ....	6
Figura 7. Cámara Asus Xtion pro Live .....	6
Figura 8. Diagrama de alto nivel de la aplicación desarrollada. ....	8
Figura 9. Diagrama que muestra el procedimiento que sigue la función de búsqueda de personas usada. 9	9
Figura 10. Ejemplo de rectángulos característicos utilizados para calcular las "Haar features". ....	9
Figura 11. Representación esquemática de la cascada de detección. Una serie de clasificadores son aplicados en cada sub-ventana. El clasificador inicial elimina un gran número de falsos positivos con muy poco tiempo de procesado, los siguientes eliminan más falsos positivos, pero necesitan más tiempo de cómputo. ....	9
Figura 12. Diagrama del algoritmo a seguir cuándo se quiere elegir la persona a seguir, en el caso (a) si se ha buscado por primera vez y en el caso (b) en la segunda y sucesivas veces. ....	11
Figura 13. Prioridad que sigue el algoritmo de elección de objetivo. Siguiendo el criterio 1 (primera vez que se realiza la elección), tienen mayor prioridad las características detectadas (A) y después la distancia al robot (B). Si se sigue el criterio 2 (segunda vez y posteriores que se realiza la elección), tiene mayor prioridad la distancia al robot (B) y después las características detectadas (A). ....	11
Figura 14. Diagrama de bloques en el que se muestra el proceso que se sigue el robot al acercarse al objetivo. La condición de parar el reconocimiento para moverse hacia las coordenadas es que las coordenadas (Coord.) sean distintas (!=) de cero. ....	12
Figura 15. Imágenes usadas para el experimento inicial. (a) Secuencia "epfl_lab". Espacio abierto y luminoso, y con poco solapamiento de gente. (b) Secuencia "epfl_corridor". Espacio estrecho, con poca luz y mucha densidad de gente. ....	14
Figura 16. Ejemplo de escenario fácil usado en los experimentos. Primera imagen del escenario, algunas imágenes intermedias y la última imagen del escenario "ArcFacil". ....	15
Figura 17. Gráficas que muestran la precisión y el recall de los métodos de reconocimiento en función del umbral de superposición. Sin distinción de distancia al robot "Depth 0". ....	16
Figura 18. Gráficas de la precisión y el recall en función del umbral de solapamiento. Con una diferenciación entre cerca y lejos en 2m "Depth2". ....	17
Figura 19. Gráficas de la Precisión y el Recall en función del parámetro mínimo de vecino requerido. ....	17
Figura 20. Gráficas de la precisión y el recall en función del factor de escala. ....	18
Figura 21. Precisión y recall en función del parámetro tamaño mínimo. ....	19

Figura 22. Precisión y recall en función del parámetro tamaño mínimo. ....	19
Figura 23. Gráfica que muestra la precisión y el número de positivos encontrados en los tres escenarios en función de la versión del algoritmo de elección que se ha usado. Cada punto de las líneas de regresión de los escenarios es una versión.....	21
Figura 24. Algunas imágenes de los resultados obtenidos con los diferentes escenarios y versiones. 1ª fila, escenario StillUZ con la versión 1 del algoritmo de elección. 2ª fila, escenario Still2UZ con la versión 2 del algoritmo. 3ª fila, escenario MoveUZ con la versión 3 del algoritmo. (A) primera imagen del experimento, (B) imágenes intermedias y (C) última imagen del experimento.....	21
Figura 25. Representación gráfica de un experimento realizado en cada escenario, (a) escenario 1, (b) escenario 2. ....	23
Figura 26. Algunas imágenes de la prueba 1 en el escenario 2. La primera imagen se corresponde al punto uno de los resultados y la última al punto 10, siendo las imágenes intermedias algunos puntos intermedios. Los cuadrados rojos y azules son los objetivos encontrados por el reconocedor y el punto verde señala el objetivo elegido para ser seguido. ....	23
Figura 27. Diagrama de bloques explicativo del método usado en los primeros experimentos con el sistema completo. ....	24
Figura 28. Imágenes de cuatro experimentos, uno en cada línea siguiendo el siguiente orden (Prueba1, Prueba 3, Prueba 4 y Prueba 8). Se usan tres escenarios: escenario 1: fila 1, escenario 2: fila 2 y 4 y escenario 3: fila 3. En cada experimento se puede ver la posición inicial del robot en rojo y la persona en azul (A), en medio imágenes intermedias en las que el robot se acerca a la persona (B), (C) y al final la posición en la que acaban el robot y la persona (D).....	24
Figura 29. Diagrama de bloques explicativo del método usado en los experimentos de detección y acercamiento iterativos. ....	25
Figura 30. Imágenes tomadas por el robot en los experimentos, que muestran los sucesivos reconocimientos que se han realizado en el experimento. Los cuadrados rojos son los objetivos encontrados por el reconocedor y el punto verde es el objetivo elegido para acercarse a él. Cada fila se corresponde a un experimento, dónde se ven la primera posición de la persona (A), las posiciones mientras el robot se acerca a la persona (B), (C) y la última posición (D). Estos experimentos se han realizado en el laboratorio, pero cambiando la posición del robot en el mismo.....	25
Figura 31. Diagrama de bloques explicativo del método usado en los experimentos finales. ....	26
Figura 32. Imágenes tomadas por el robot en los experimentos finales, que muestran los sucesivos reconocimientos automáticos que se han realizado en el experimento. Los cuadrados rojos son los objetivos encontrados por el reconocedor y el punto verde es el objetivo elegido para acercarse a él. Cada fila se corresponde a un experimento, dónde se ven la primera posición de la persona (A), las posiciones mientras el robot se acerca a la persona (B), (C) y la última posición (D). Estos experimentos se han realizado en dos escenarios diferentes, los primeros cuatro experimentos se han realizado en el laboratorio y el último en un pasillo.....	27

Figura 33. Diagrama de bloques que muestra de qué forma se usa ROS en este trabajo.....	29
Figura 34. Partes que componen la plataforma robótica TurtleBot.....	30
Figura 35. Conexiones de la base del TurtleBot (Kobuki base).....	30
Figura 36. Cámara Asus Xtion pro Live.....	31
Figura 37. Imágenes pertenecientes al escenario Still, en el cuál la cámara permanece quieta.....	32
Figura 38. Imágenes pertenecientes al escenario Translation, en el cuál la cámara se mueve describiendo una línea recta.....	32
Figura 39. Imágenes pertenecientes al escenario Arc, en el cuál la cámara describe una trayectoria en forma de arco.....	33
Figura 40. Imágenes pertenecientes al escenario Rotation, en el cuál la cámara rota sobre sí misma. ....	33
Figura 41. Precisión y recall en función del porcentaje de acierto, sin distinción de cerca y lejos.....	34
Figura 42. Precisión y recall en función del porcentaje de acierto, haciendo distinción de cerca y lejos en dos metros.....	34
Figura 43. Precisión y recall en función del porcentaje de acierto, sin distinción de cerca y lejos.....	35
Figura 44. Precisión y recall en función del porcentaje de acierto, haciendo distinción de cerca y lejos en dos metros.....	35
Figura 45. Precisión y recall en función del porcentaje de acierto, sin distinción de cerca y lejos.....	36
Figura 46. Precisión y recall en función del porcentaje de acierto, haciendo distinción de cerca y lejos en dos metros.....	36
Figura 47. Precisión y Recall en función del tamaño mínimo.....	37
Figura 48. Precisión y Recall en función del tamaño mínimo.....	37
Figura 49. Diagrama de bloques de la versión 1 del algoritmo de elección de objetivo.....	38
Figura 50. Diagrama de bloques de la versión 2 del algoritmo de elección de objetivo.....	38
Figura 51. Diagrama de bloques de la versión 3 del algoritmo de elección de objetivo.....	39
Figura 52. Diagrama de bloques de la versión 4 del algoritmo de elección de objetivo.....	40
Figura 53. Imágenes de los experimentos evaluados con la versión 1. En cada línea hay un experimento, (StillUZ, StillUZ2 y MoveUZ). Los cuadrados azul y rojo son los objetivos encontrados y el punto verde señala el objetivo elegido.....	40
Figura 54. Imágenes de los experimentos evaluados con la versión 2. En cada línea hay un experimento, (StillUZ, StillUZ2 y MoveUZ). Los cuadrados azul y rojo son los objetivos encontrados y el punto verde señala el objetivo elegido.....	41
Figura 55. Imágenes de los experimentos evaluados con la versión 3. En cada línea hay un experimento, (StillUZ, StillUZ2 y MoveUZ). Los cuadrados azul y rojo son los objetivos encontrados y el punto verde señala el objetivo elegido.....	41
Figura 56. Representación gráfica de la prueba 2 en el escenario 2.....	42
Figura 57. Representación gráfica de la prueba 3 en el escenario 2.....	42

Figura 58. Algunas fotos del experimento 1 en el escenario 1. La primera imagen se corresponde al primer punto del experimento y la última al último punto del experimento, el resto de imágenes son algunos puntos intermedios. Los cuadrados rojos y azules corresponden a los objetivos encontrados y el punto verde señala el objetivo elegido.....	43
Figura 59. Algunas fotos del experimento 1 en el escenario 2. La primera imagen se corresponde al primer punto del experimento y la última al último punto del experimento, el resto de imágenes son algunos puntos intermedios. Los cuadrados rojos y azules corresponden a los objetivos encontrados y el punto verde señala el objetivo elegido.....	43
Figura 60. Algunas fotos del experimento 2 en el escenario 2. La primera imagen se corresponde al primer punto del experimento y la última al último punto del experimento, el resto de imágenes son algunos puntos intermedios. Los cuadrados rojos y azules corresponden a los objetivos encontrados y el punto verde señala el objetivo elegido.....	44
Figura 61. Algunas fotos del experimento 3 en el escenario 2. La primera imagen se corresponde al primer punto del experimento y la última al último punto del experimento, el resto de imágenes son algunos puntos intermedios. Los cuadrados rojos y azules corresponden a los objetivos encontrados y el punto verde señala el objetivo elegido.....	44
Figura 62. Imágenes de los vídeos grabados en los primeros experimentos con el sistema completo. Cada línea es un experimento, dónde se muestra la posición inicial del robot y la persona, luego dos posiciones intermedias y la posición final del robot y la persona.....	45
Figura 63. Cada línea corresponde con un experimento en el que se ha realizado un reconocimiento y acercamiento iterativo. En la primera imagen se muestra la posición inicial de la persona y el robot en el experimento, a continuación, dos posiciones intermedias en las que el robot se ha acercado poco a poco a la persona y por último las posiciones finales en el experimento. ....	46
Figura 64. Imágenes de los vídeos grabados en los últimos experimentos con la plataforma real. Cada línea corresponde con un experimento. Las imágenes están ordenadas de la siguiente forma, posición inicial del robot y la persona, posiciones intermedias en las que se han realizado reconocimiento y seguimiento y, por último, posición final del robot y la persona.....	46

Tabla 1. Diagrama de Gant.....	2
Tabla 2. Resultados obtenidos por los métodos de reconocimiento en función del umbral de superposición "OverLapTH", en el escenario Arc. Sin hacer distinción de distancia al robot. ....	16
Tabla 3. Resultados obtenidos por los métodos de reconocimiento en función del umbral de superposición "OverLapTH", en el escenario Arc. Con una diferenciación de cerca y lejos en 2m.....	16
Tabla 4. Resultados obtenidos por los métodos de reconocimiento en función del parámetro mínimo de vecinos requerido.....	17
Tabla 5. Resultados obtenidos por los métodos de reconocimiento en función del valor que toma el parámetro factor de escala. ....	18
Tabla 6. Resultados cuándo el ancho y el alto del parámetro tamaño mínimo se cambian a la vez. ....	19
Tabla 7. Resultados cuándo sólo se cambia el alto del parámetro tamaño mínimo. ....	19
Tabla 8. Resultados del número de positivos encontrados por el algoritmo y del tanto por ciento de acierto en función del escenario y de la versión del algoritmo de elección. ....	20
Tabla 9. Resultados del error en la medida de distancia en el experimento 1 en el escenario 1. ....	22
Tabla 10. Resultados del error en la medida de distancia en el experimento 1 en el escenario 2. ....	22
Tabla 11. Especificaciones técnicas cámara Asus xtion pro Live.....	31
Tabla 12. Resultados con diferentes porcentajes de acierto sin hacer distinción de cerca y lejos. ....	34
Tabla 13. Resultados con diferentes porcentajes de acierto haciendo distinción de cerca y lejos en dos metros.....	34
Tabla 14. Resultados con diferentes porcentajes de acierto sin hacer distinción de cerca y lejos. ....	35
Tabla 15. Resultados con diferentes porcentajes de acierto haciendo distinción de cerca y lejos en dos metros.....	35
Tabla 16. Resultados con diferentes porcentajes de acierto sin hacer distinción de cerca y lejos. ....	36
Tabla 17. Resultados con diferentes porcentajes de acierto haciendo distinción de cerca y lejos en dos metros.....	36
Tabla 18. Resultados de cambiar el tamaño mínimo en el método Fullbody. Cambiando ancho y alto a la vez. ....	37
Tabla 19. Resultados de cambiar el tamaño mínimo en el método Fullbody. Cambiando solo el alto. ....	37
Tabla 20. Resultados del error de distancia en cada punto de la segunda prueba en el escenario 2. ....	41
Tabla 21. Resultados del error de distancia en cada punto de la tercera prueba en el escenario 2.....	42



# 1. Introducción

## 1.1. Motivación Y Contexto

La motivación que ha llevado a desarrollar este proyecto ha sido la creciente demanda de automatizar distintos servicios, y la posible necesidad de una atención al cliente que no tenga que recaer por completo en un recurso humano, sino que pueda ser automatizada y por tanto llevada a cabo por una máquina con la suficiente autonomía. Con este motivo se pretende desarrollar un robot que sea capaz de localizar una persona, determinar su posición respecto de sí mismo y avanzar hasta llegar a ella, de esta manera podrá ofrecerle ciertos servicios. Se pretende captar la atención del cliente o usuario de cualquier empresa o edificio público, desde una plataforma móvil que opere de manera autónoma en un espacio abierto, como podría ser un hall o una sala de museo, tal como se muestra en la Figura 1.



(a)



(b)

Figura 1. Robots móviles circulando por un museo (a). Plataforma móvil en guiado de museos (b).

Este proyecto se ha realizado dentro del laboratorio de Robótica, Percepción y Tiempo Real del Instituto de Investigación en Ingeniería de Aragón (I3A), Universidad de Zaragoza. Este proyecto ha hecho uso del equipamiento proporcionado por el proyecto CESAR (Centro de Supercomputación de Aragón), de la Universidad de Zaragoza.

## 1.2. Objetivos Y Tareas

El objetivo general de este proyecto es desarrollar un sistema en el que un robot de servicio desarrolle tareas de guiado y atención a los visitantes en el entorno de un museo u otros edificios públicos. En concreto, se quiere desarrollar un sistema de reconocimiento y seguimiento de personas utilizando un sensor RGB-D (visión y profundidad), montado en un robot.

Para conseguir este objetivo, y cubriendo los objetivos planteados al inicio de este proyecto se han llevado a cabo las siguientes tareas:

- Estudio, instalación y familiarización con entorno de OpenCV y C++ sobre Eclipse y Ubuntu OS.

- Estudio, instalación y familiarización del entorno ROS, entorno utilizado para gestión de plataformas robóticas y sensores relacionados. Incluyendo utilización de OpenCV sobre ROS.
- Evaluación de algoritmos incluidos en OpenCV para detección de personas, caras o partes del cuerpo en los casos de uso de interés para el proyecto.
- Estudio de otros trabajos relacionados sobre los problemas de reconocimiento y seguimiento basados en visión en entornos de interior.
- Implementación o adaptación de los métodos estudiados para usarlos como interfaz entre el sensor RGB-D (Asus) y el robot disponible (a través del sistema ROS).
- Evaluación de una aplicación que use el método propuesto para probar reconocimiento y seguimiento en el laboratorio.
- Integración y pruebas de la aplicación final en una plataforma robótica real.
- Documentación de actividades, evaluaciones, "manual" de uso de las demostraciones preparadas, para permitir el uso de los resultados de este proyecto en futuros eventos.

La organización temporal de las tareas que se han llevado a cabo en este trabajo es la señalada en el diagrama temporal que se muestra en la Tabla 1.

Tabla 1. Diagrama de Gant.

	FEBRERO	MARZO	ABRIL	MAYO	JUNIO	SEPTIEMBRE	OCTUBRE
Estudio, instalación y familiarización con entorno de OpenCV y C++ sobre Eclipse y Ubuntu OS.	■						
Estudio, instalación y familiarización del entorno ROS y sensores relacionados.	■						
Evaluación de algoritmos incluidos en OpenCV para detección de personas, caras, etc.	■						
Estudio de trabajos previos sobre detección y seguimiento para la comparación de los mismos.		■					
Implementación o adaptación de los métodos estudiados para usarlos como interfaz cámara, robot.			■	■			
Aplicación que use el método propuesto para probar reconocimiento y seguimiento en el laboratorio.			■	■			
Demostración de la aplicación final en una plataforma robótica real.					■	■	
Documentación de actividades, evaluaciones, manual de uso de las demostraciones preparadas.		■	■	■	■	■	■

## 1.3. Trabajos Relacionados

En esta sección se exponen brevemente los temas y trabajos relacionados con los principales bloques de este proyecto. Mientras que la tarea de reconocimiento de personas a realizar en este proyecto se parece en gran medida a los trabajos previos estudiados, la tarea de seguimiento es bastante diferente.

### 1.3.1. Técnicas De Detección De Personas

En el campo del reconocimiento visual automático, más concretamente en la detección de personas, hay una inmensa variedad de escritos y trabajos, como resultan demasiados para estudiarlos todos o incluso nombrarlos todos, se han seleccionado algunos de los más relevantes [2], [3], [4] que servirán de comparativa para este trabajo. Dos de estos trabajos [2], [3] consisten en la detección y seguimiento de personas con una cámara RGB-D, la cual se explica en detalle en el Anexo III, igual que la utilizada en este trabajo. El tercero [4] se centra en la detección, pero haciendo hincapié en la detección robusta ante obstrucciones, es decir cuándo haya muchas personas delante de la cámara. La Figura 2 y la Figura 3 muestran dos posibles escenarios de detección de personas. La Figura 2 muestra un caso más sencillo, donde se detecta a la persona en 2D, en un entorno con poco nivel de oclusiones, con lo cual un sistema no muy robusto puede funcionar correctamente. La Figura 3, sin embargo, muestra una situación de detección más detallada, en 3D, dónde es necesario un algoritmo más robusto debido a una alta densidad de personas en la escena, con muchas oclusiones.

En este trabajo, se ha planteado que no es necesario para nuestra aplicación realizar una detección detallada de las distintas personas (como en el ejemplo de la Figura 3) en un escenario en el que haya muchas personas juntas, ya que lo que se quiere es que el robot se acerque al grupo de personas, pero no resulta de mucha importancia qué persona en concreto haya reconocido.



Figura 2. Detección de personas en 2D en un entorno sencillo sin muchas oclusiones ni objetos móviles



Figura 3. Detección de personas en 3D en entornos con un nivel alto de oclusiones y de densidad de elementos móviles.

Resulta muy importante que los estudios relacionados elegidos para comparar nuestro sistema [2], [3], [4] tengan disponibles datos con los que poder trabajar y compararse de manera objetiva. Así, los

primeros experimentos de reconocimiento de personas se realizan con las imágenes proporcionadas por uno de los trabajos [2] y se comparan nuestros resultados con los resultados ideales, el producto de dicha comparación se expondrá más adelante en el capítulo 3, *Evaluación Y Experimentos*.

Estos estudios que se han tomado como referencia discuten y comparan a su vez muchos otros trabajos sobre el tema, y evalúan diferentes métodos para el reconocimiento de personas [5], [6], [7], [8], [9]. En este trabajo se ha decidido que el método que se va a usar para llevar a cabo la detección de personas en una sala es el método que viene incorporado en las librerías de openCV [1], que está basado en el conocido método de Viola-Jones [10] que usa clasificadores en cadena. Este clasificador se explicará a fondo en la sección 2.2, *Detección De Personas*.



Figura 4. Ejemplo de reconocimiento de caras con las librerías de OpenCV.

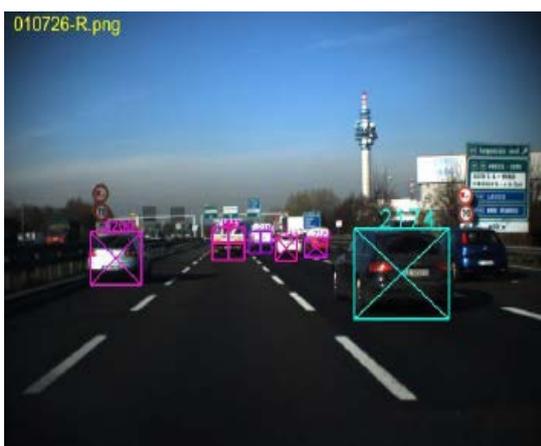
Se ha escogido este método porque al estar incorporado en las librerías de openCV resulta más sencillo de integrar en el código de manejo y comunicación con el robot. Además, los experimentos preliminares de precisión y *recall* del clasificador obtenidos fueron aceptables para los niveles de precisión que requiere la tarea a realizar.

Dentro de la librería de openCV, existen numerosos clasificadores pre-entrenados para detectar diferentes partes del cuerpo de una persona, por ejemplo, ojos, boca, cara, parte de arriba del cuerpo, cuerpo entero, etc. Todos basados en el mismo tipo de clasificador, entrenados para reconocer distintos tipos de patrones. En la Figura 4 se puede observar un ejemplo de los resultados obtenidos con estos reconocedores (reconocedor *frontalface* y *eye* de openCV). Se ha experimentado con varios de estos modelos, para determinar cuáles se adaptan mejor al objetivo planteado en este proyecto. Finalmente, se ha decidido usar los modelos de openCV llamados *frontalface\_default*, *fullbody* y *upperbody*, para combinar los resultados de varios detectores, como se explica en la sección 3.1, *Evaluación Del Sistema De Reconocimiento*. Y así obtener unos resultados más robustos.

### 1.3.2. Seguimiento o *Tracking*

En el campo del seguimiento, o *tracking*, y en particular del seguimiento de personas, también existen innumerables estudios y trabajos previos en la literatura, los cuales consisten, en su gran mayoría, en el *tracking* de un objetivo concreto, es decir, deben seguir con precisión un objetivo móvil, por ejemplo, un coche [11], una persona [3] o un camino previamente trazado [12]. En la Figura 5 se pueden ver dos de estos ejemplos nombrados.

Hay muchas técnicas distintas para realizar el seguimiento autónomo de un objeto en movimiento. Se pueden utilizar sensores de rango, sensores que midan cantidad de luz reflejada o cámaras. Este último caso es el más cercano a este proyecto. Un ejemplo muy común en el seguimiento con cámaras es el seguimiento preciso de un coche (Figura 5 a), en este tipo de seguimiento se suelen usar histogramas de color para diferenciar el coche que se quiere seguir de los demás. Un ejemplo del uso de un sensor que mide cantidad de luz reflejada, para realizar seguimiento, sería el seguimiento de un camino previamente trazado, una línea, por ejemplo (Figura 5 b). Si la luz se refleja, el robot está fuera de la línea, si no, está sobre la línea.



(a)



(b)

Figura 5. Ejemplo de aplicaciones de *tracking*. (a) seguimiento de coches. (b) Seguimiento de una línea trazada en el suelo.

En la mayoría de estos trabajos previos sobre seguimiento visual, el objetivo principal consiste en elegir un objeto y seguirlo con la mayor precisión posible. Sin embargo, en este proyecto no es crítico que el robot siga exactamente la trayectoria de la persona, sino que elija a un objetivo y se acerque. Por lo tanto, se ha decidido implementar una versión de seguimiento más sencilla que ha sido diseñada desde cero, como se explica en la sección 2.3, *Elección Y Seguimiento Del Target*. En este algoritmo el punto clave es cómo elegir el objetivo a seguir, y después de un primer acercamiento a una persona o un grupo de personas de la sala, el robot continúa acercándose a la persona que se encuentre más cerca suyo.

### 1.3.3. Tecnologías Utilizadas

Esta sección resume las tecnologías y entorno de desarrollo utilizado en este proyecto.

En primer lugar, se ha usado ROS (Robot Operating System) [13] que es un marco de referencia para escribir software para robots. El cuál proporciona librerías y herramientas para ayudar a la creación de aplicaciones robóticas, todo de forma gratuita y bajo un marco de total colaboración. Se explicará más ampliamente en que consiste ROS y cómo se ha usado en el Anexo I.

En segundo lugar, se ha usado cómo plataforma móvil el robot llamado TurtleBot [14], en la Figura 6 se puede ver el robot usado en nuestro experimento. Es una plataforma móvil de bajo coste para desarrolladores, que puede estar equipada con distintos sensores. En el Anexo II se explicará detalladamente las partes y funciones de este robot.



*Figura 6.* TurtleBot utilizado en este experimento.

Por último se usa un sensor RGB-D, que capta imágenes en color (RGB) y además posee un sensor que mide la distancia a la que se encuentran los objetos de la escena, en concreto que mide la distancia a la que se encuentra cada pixel de la imagen RGB. La cámara utilizada en este proyecto es de la marca Asus. En la Figura 7 se puede ver una foto de esta cámara y en el anexo III se incluyen más detalles de cómo funciona.



*Figura 7.* Cámara Asus Xtion pro Live

## 1.4. Organización Del Proyecto

En los siguientes capítulos de esta memoria se tratará el trabajo realizado estructurado de la siguiente manera.

En primer lugar, en la sección 2.1. *Resumen Del Sistema Realizado*, se explica brevemente la aplicación completa y los pasos necesarios para realizar dicha aplicación.

A continuación, en la sección 2.2. *Detección De Personas*, se explican los algoritmos usados para la detección de persona, dichos algoritmos son los incorporados en openCV y se procederá por tanto a la explicación de la documentación proporcionada por los desarrolladores de estos algoritmos para poder entender de esa forma como se reconoce a una persona. También se explicarán alguno de los cambios realizados después de realizar los experimentos pertinentes.

Después, en la sección 2.3. *Elección Y Seguimiento Del Target*, se procederá a la explicación del sistema de seguimiento realizado, este sistema no se basa en ningún otro algoritmo por lo tanto se explicarán las bases del algoritmo desarrollado íntegramente.

Luego, en el capítulo 3. *Evaluación Y Experimentos*, se evaluarán los experimentos realizados en las distintas partes del trabajo. Primero, el sistema de reconocimiento de personas cambiando y ajustando los distintos parámetros, así como la forma de aplicarlos. En segundo lugar, el sistema de seguimiento desarrollado íntegramente y en tercer lugar el sistema completo aunando ambos sistemas.

Por último, en el capítulo 4. *Conclusiones*, se redactarán las conclusiones dónde se pondrá de manifiesto las posibles limitaciones de este proyecto y a que se deben. Y si se pudieran realizar mejoras en el futuro y de qué manera se llevarían a cabo.

## 2. Sistema De Detección Y Seguimiento

En este capítulo se explica el sistema completo desarrollada en este trabajo.

### 2.1. Resumen Del Sistema Realizado

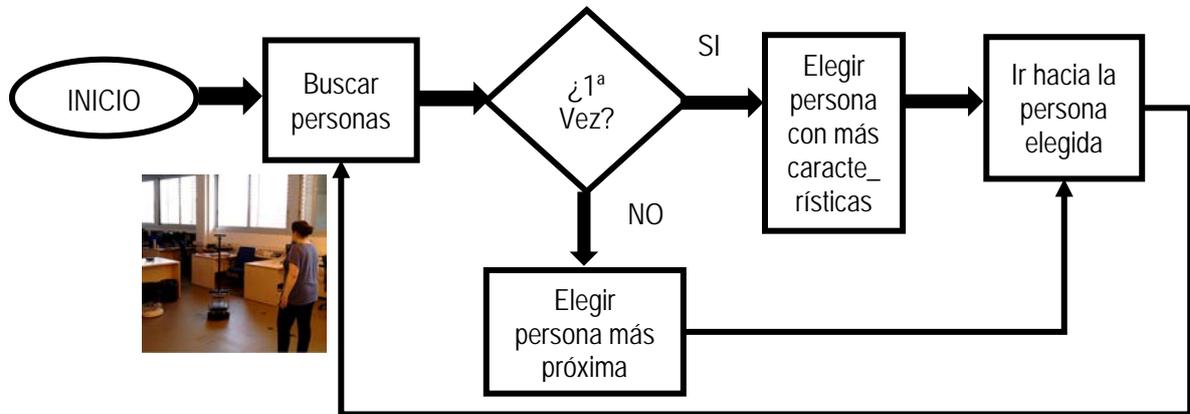


Figura 8. Diagrama de alto nivel de la aplicación desarrollada.

La Figura 8 muestra en un solo vistazo el funcionamiento del sistema de detección y seguimiento/acercamiento diseñado, que consta de los siguientes pasos:

1. En primer lugar, se buscan personas usando el reconocedor implementado.
2. A continuación, el sistema puede ejecutar dos posibilidades, dependiendo si es la primera vez que se ha ordenado la búsqueda de una persona o si no lo es:

2.1. En el caso de que sea la primera vez se elegirá la persona que más características encontradas tenga de todas las que se han encontrado, se ha decidido hacerlo de esta forma porque cuántas más características se encuentran más probabilidades hay de que sea una persona.

2.2. Si no es la primera vez que se ha realizado una búsqueda se elige a la persona que se encuentra más próxima, ya que el objetivo es acercarse a una persona, estas segunda y posteriores iteraciones se realizan para asegurarnos de que la persona no se ha movido desde que se ha visto la vez anterior.

3. Por último, se va hacia la persona, es decir el robot se mueve hacia las coordenadas que se le ha ordenado, cuándo llega a su destino se vuelve a ordenar que busque a una persona y se repite el bucle.

En las secciones posteriores se explicarán con mayor profundidad los distintos bloques de este sistema.

## 2.2. Detección De Personas

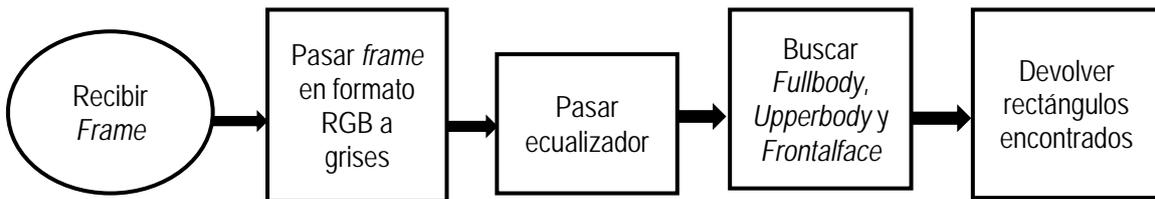


Figura 9. Diagrama que muestra el procedimiento que sigue la función de búsqueda de personas usada.

En esta sección se va a explicar el procedimiento diseñado para detectar una persona en una imagen, así como los detalles técnicos que explican dicho procedimiento. En la Figura 9 se muestran los pasos que sigue el sistema de reconocimiento, los cuáles se explicarán a continuación.

Esta parte del proyecto se ha basado en gran medida en algoritmos existentes que se han ajustado posteriormente para que cumplan el objetivo propuesto. Como se ha comentado anteriormente, el detector utilizado como base en este proyecto es una implementación de la librería OpenCV del reconocedor de Viola-Jones [10]. Este reconocedor está basado en las características de Haar (*Haar-like features*). Estas características son una adaptación de las funciones de Haar que usaron trabajos anteriores, como Papageorgiou [15] para detección de objetos en imágenes estáticas de escenas desordenadas. Las Haar-like features se utilizan en muchos trabajos relacionados, y podemos encontrar muchas aplicaciones o extensiones de las mismas, como por ejemplo en Lienhart [17], para la detección con características de imágenes rotadas. Las *Haar-like features* se basan en sumas y diferencias de los píxeles dentro de áreas rectangulares. El trabajo de Viola-Jones propone tres tipos (*two-rectangle*, *three-rectangle* y *four-rectangle*) que se pueden ver en la Figura 10. El valor que se le da a la característica llamada *two-rectangle* (Figura 10 A y B), es la diferencia de la suma de los píxeles dentro de dos regiones rectangulares, estas regiones tienen el mismo tamaño y forma y son adyacentes horizontal o verticalmente. La característica llamada *three-rectangle* (Figura 10 C), calcula la suma dentro de dos rectángulos exteriores, menos la suma en un rectángulo central. Por último, la característica llamada *four-rectangle* (Figura 10 D), calcula la diferencia entre pares de rectángulos en la diagonal.

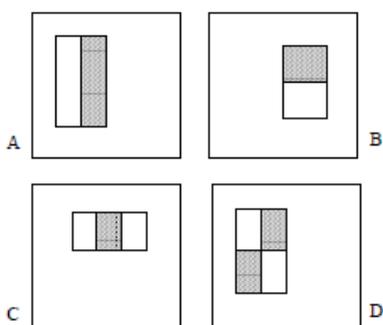


Figura 10. Ejemplo de rectángulos característicos utilizados para calcular las "Haar features".

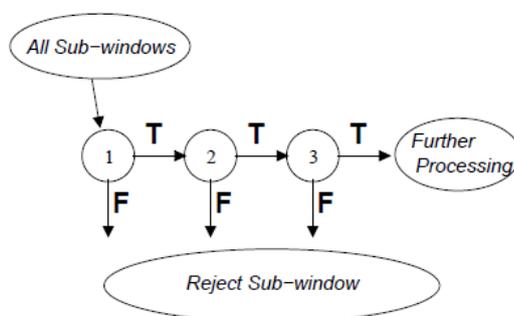


Figura 11. Representación esquemática de la cascada de detección. Una serie de clasificadores son aplicados en cada sub-ventana. El clasificador inicial elimina un gran número de falsos positivos con muy poco tiempo de procesado, los siguientes eliminan más falsos positivos, pero necesitan más tiempo de cómputo.

La forma general del proceso de detección que usa el sistema de Viola-Jones [10], es una serie de clasificadores en "cascada" (Figura 11). Los resultados positivos del primer clasificador desencadenan la evaluación de un segundo clasificador, el cuál ha sido también ajustado para alcanzar tasas de detección muy altas. Los resultados positivos del segundo clasificador activan la evaluación del tercer clasificador y así consecutivamente. Un resultado negativo en cualquier punto lleva a la inmediata desestimación de la sub-ventana. Las etapas en la cascada son construidas por entrenadores de clasificadores usando AdaBoost [16] y después se ajustan los umbrales para minimizar los falsos negativos.

Para que este método sea eficiente es necesario pasar las imágenes de formato RGB a escala de grises, utilizando una representación intermedia que se llama imagen integral. Aunque puede resultar menos preciso que otros sistemas de reconocimiento, por ser algo más tosco en su reconocimiento, cuando se usa la imagen integral, la eficiencia del método suple ampliamente su limitada flexibilidad.

Antes de realizar la detección, hay que realizar un filtrado y normalizado de la imagen, para minimizar el efecto de las diferentes condiciones de iluminación. Esta normalización se realizará usando la desviación estándar, calculada en la Ecuación 1, donde  $\sigma$  es la desviación estándar,  $m$  es la media y  $x$  es el valor del pixel dentro de la sub-ventana. La variación media de una sub-ventana de una imagen, puede ser calculada rápidamente usando la representación integral de las imágenes.

$$\sigma^2 = m^2 - \frac{1}{N} \sum x^2$$

*Ecuación 1.* Cálculo de la variación en una sub-ventana de una imagen.

Por último, se busca en la imagen personas usando los métodos *Fullbody*, *Upperbody* y *Face*, los cuáles buscan respectivamente cuerpo entero, parte de arriba del cuerpo y cara.

## 2.3. Elección Y Seguimiento Del Target

Este capítulo detalla el proceso diseñado para seleccionar el objetivo, o *target*, de entre todos los posible. Y el proceso para llevar a cabo el seguimiento de dicho objetivo. Este algoritmo se ha realizado desde cero y por lo tanto se va a explicar desde cero.

En el diagrama de la Figura 12 se pueden ver las pautas que sigue el algoritmo para seleccionar la persona a la que seguir o acercarse (objetivo). En el caso a) si es la primera vez que se ha mandado la orden de buscar persona. Y en el caso b) si es la segunda vez y sucesivas.

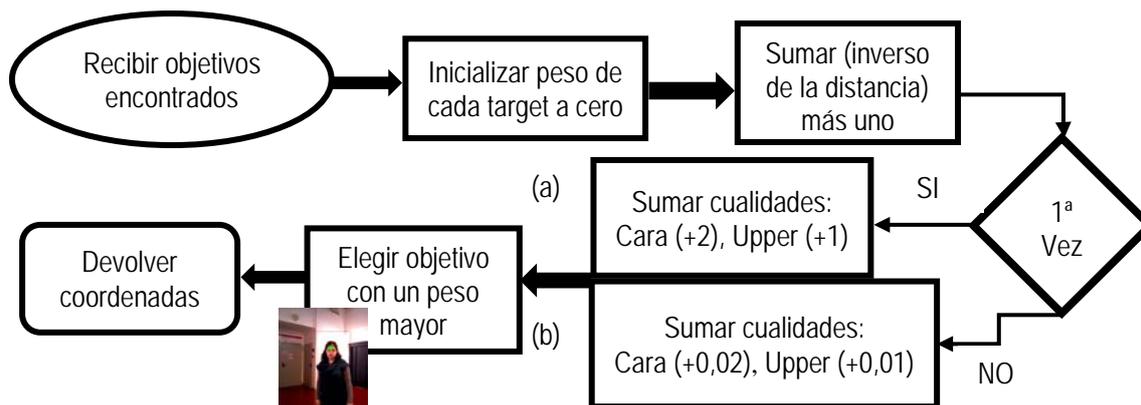


Figura 12. Diagrama del algoritmo a seguir cuando se quiere elegir la persona a seguir, en el caso (a) si se ha buscado por primera vez y en el caso (b) en la segunda y sucesivas veces.

Para decidir a qué objetivo de todos los que se han reconocido se va a seguir, se ha desarrollado un sistema que asigna prioridades a cada objetivo según los resultados, obtenidos combinando los distintos detectores, evaluados en cada imagen (Figura 13). Siguiendo este proceso, se puede saber si un posible objetivo está cerca o lejos o si tiene más o menos características relevantes detectadas. Los criterios que se evalúan a lo largo de este proceso son detallados a continuación.

(A) Upperbody	Upperbody + Frontalface	Fullbody	Fullbody + Upperbody	Fullbody + Frontalface	Fullbody + Frontalface +Upperbody
Menor prioridad					Mayor prioridad
(B)	Cerca		Lejos		

Figura 13. Prioridad que sigue el algoritmo de elección de objetivo. Siguiendo el criterio 1 (primera vez que se realiza la elección), tienen mayor prioridad las características detectadas (A) y después la distancia al robot (B). Si se sigue el criterio 2 (segunda vez y posteriores que se realiza la elección), tiene mayor prioridad la distancia al robot (B) y después las características detectadas (A).

- Primero, este sistema empieza otorgando a cada objetivo la puntuación equivalente al inverso de la distancia al robot más uno, con este sistema la puntuación inicial de cada objetivo queda ponderada entre cero y uno, dando mayor prioridad a los objetivos que se encuentran más cerca del robot.
- En segundo lugar:
  - Si es la primera vez que se ha realizado el reconocimiento, se suman dos puntos si se ha encontrado una cara en el mismo lugar que el cuerpo y/o uno si se ha encontrado un cuerpo medio a la vez que el cuerpo entero. Criterio 1 Figura 13 (A), tienen mayor prioridad los objetivos que más características tengan.
  - Si es la segunda vez o posteriores que se realiza el reconocimiento, se suman 0.02 si se ha encontrado cara y 0.01 si se ha encontrado medio cuerpo con el cuerpo entero. Criterio 2 Figura 13 (B), tienen mayor prioridad los objetivos que más cerca estén.

Se tienen en cuenta tanto los resultados del detector *Fullbody* (cuerpo entero) como del detector *Upperbody* (medio cuerpo), pero a la hora de elegir un objetivo tienen prioridad las detecciones del primero, ya que presenta una precisión mayor en los experimentos realizados.

Después de elegir el objetivo a seguir se toman las coordenadas de dicho objetivo (x e y respecto de la cámara de RGB y la profundidad dada por el sensor) y se transforman a coordenadas de robot (este procedimiento se muestra en la Ecuación 2). En el capítulo 3, se muestra un experimento en el que se calcula el error de profundidad medio que comete esta ecuación. Por último, se usan la profundidad y la coordenada x para mandárselas al robot como coordenadas finales del objetivo.

$$z = d$$

$$x = (u - cx) * z / fx$$

$$y = (v - cy) * z / fy$$

Ecuación 2. Ecuaciones para pasar de coordenadas en la imagen a coordenadas del robot.

En estas ecuaciones, 'z', 'x' e 'y' son las coordenadas que usa el robot para ir a su objetivo. Por otra parte 'u', 'v' y 'd' son respectivamente las coordenadas del punto en la imagen y la profundidad de dicho punto. Por último, 'cx', 'cy', 'fx', y 'fy' son respectivamente el centro óptico por defecto y la longitud focal por defecto en x e y.

Se ha aplicado un *offset*, ajustado experimentalmente, a las coordenadas del objetivo o *goal* que debe alcanzar el robot, porque se ha comprobado que en muchos casos este objetivo es demasiado preciso y se sitúa encima de la persona, por lo tanto, el robot no es capaz de alcanzar el objetivo.

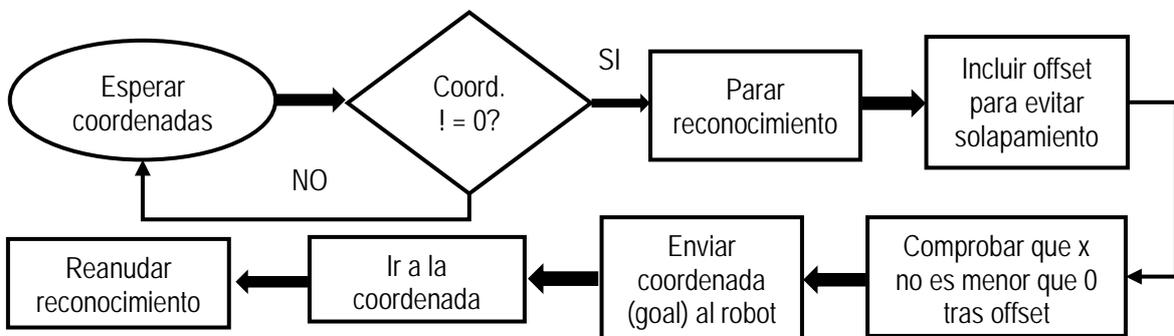


Figura 14. Diagrama de bloques en el que se muestra el proceso que se sigue el robot al acercarse al objetivo. La condición de parar el reconocimiento para moverse hacia las coordenadas es que las coordenadas (*Coord.*) sean distintas (*!=*) de cero.

Una vez que se ha elegido un objetivo y se ha mandado su ubicación al robot se procede al seguimiento (Figura 14) de dicha persona, primero se espera a que el robot alcance la primera posición proporcionada, después se vuelve a buscar una persona, utilizando esta vez el criterio 2 (Figura 13 (B)) del algoritmo de elección, y de nuevo se envían las coordenadas transformadas al robot para que vaya a la segunda ubicación. Este procedimiento se repetiría hasta que se apague el sistema.

### 3. Evaluación Y Experimentos

En este capítulo se explicarán los experimentos realizados, el objetivo de cada experimento, una descripción detallada de los datos usados y los parámetros que se utilizan, los resultados obtenidos y las conclusiones que se sacan de cada experimento.

Se han organizado los experimentos en tres apartados, experimentación con el sistema de reconocimiento, sección 3.1, experimentación con el sistema de elección y seguimiento del *target*, sección 3.2, y por último experimentación en una plataforma real de la aplicación completa, sección 3.3.

En los experimentos detallados a continuación se utilizan las siguientes medidas de rendimiento:

- **Superposición (Overlap).**

$$Inter = CuadradoReal \cap ObjetivoEncontrado$$

$$Union = CuadradoReal \cup ObjetivoEncontrado$$

$$Overlap = Inter / Union$$

$$Si\ Overlap > OverlapTH\ entonces\ positivo$$

Ecuación 3. Ecuaciones para saber si se ha encontrado un objetivo real, calculando la superposición entre los objetivos encontrados y las personas reales.

En las ecuaciones 3 se calculan la intersección y la unión de los objetivos encontrados con los objetivos reales. Con estos parámetros se calcula la superposición que hay entre ellos (*Overlap*) y así se decide si el objetivo encontrado es realmente una persona usando el umbral de superposición que en las tablas se referenciará como "OverlapTH".

- **Precisión.** Medida de rendimiento que indica que porcentaje de los objetivos encontrados son realmente personas.

$$Precisión = TP / (TP + FP)$$

Ecuación 4. Ecuación para calcular la precisión de un método.

- **Recall.** Medida de rendimiento que indica que porcentaje de personas reales han sido encontradas.

$$Recall = TP / (TP + FN)$$

Ecuación 5. Ecuación para calcular el *recall* de un método.

Los componentes de estas dos ecuaciones son *TP*, verdaderos positivos (true positives), objetivos encontrados que realmente son personas. *FP*, falsos positivos (false positives), objetivos encontrados que no son una persona. Y *FN*, falsos negativos (false negatives), personas reales que no se han encontrado.

## 3.1. Evaluación del Sistema de Reconocimiento de Personas

### 3.1.1. Comparación de Métodos Disponibles en OpenCV

El **objetivo** de este experimento es evaluar las funciones y modelos pre-entrenados disponibles en la biblioteca de openCV para detección de personas en varios escenarios relacionados con nuestra aplicación. Se evalúan cuatro modelos disponibles (*Fullbody*, *Upperbody*, *Frontalface*, y *mes-upperbody*) que reconocen distintas partes de una persona (cuerpo entero, la parte superior de un cuerpo, la cara y medio cuerpo, respectivamente). Se considera que pueden ser de utilidad para combinarlos y reconocer una persona de manera robusta en un espacio cerrado. Se han probado en diferentes escenarios para llegar a saber qué modelos funcionan mejor en cada escenario y así elegir con que modelos construir nuestro sistema.

Los **datos** utilizados en este experimento son imágenes obtenidas del conjunto de datos utilizados en trabajos relacionados [5], y disponibles para investigación<sup>1</sup>. Para este experimento inicial se han escogido seis imágenes, tres de un escenario y tres de otro, porque se pretendía que la evaluación fuera rápida al ser un experimento preliminar. Además, se han escogido las imágenes para que tengan diferentes grados de dificultad. Podemos ver estas imágenes en la Figura 15. Se han elegido estos dos escenarios, para evaluar los modelos disponibles en distintas situaciones, ya que "*epfl\_lab*" (Figura 15.a) tiene unas condiciones más favorables para el reconocedor que "*epfl\_corridor*" (Figura 15.b) porque hay más luz y la gente está menos junta.



Figura 15. Imágenes usadas para el experimento inicial. (a) Secuencia "*epfl\_lab*". Espacio abierto y luminoso, y con poco solapamiento de gente. (b) Secuencia "*epfl\_corridor*". Espacio estrecho, con poca luz y mucha densidad de gente.

En los experimentos los cuatro modelos obtienen **resultados** parecidos. Reconocen a todas o casi todas las personas del escenario si el escenario es fácil (luminoso y sin solapamiento de gente, secuencia "*epfl\_lab*"). Pero no reconocen a nadie o casi nadie si el escenario es difícil (poca luz y gente muy junta, "*epfl\_corridor*"). Tan sólo uno de los cuatro tiene resultados algo peores que los otros, el

<sup>1</sup> <http://cvlab.epfl.ch/data/rgb-d-pedestrian>

modelo *mes-upperbody* sólo reconoce a las personas si éstas están de frente, nunca si se encuentran de lado o de espaldas.

En vista de los resultados obtenidos se ha decidido no usar el modelo *mes-upperbody*, porque se pretende que la plataforma móvil se acerque también a personas que estén de espaldas y dicho modelo no ha reconocido a ninguna persona de espaldas o de lado en los experimentos preliminares realizados. Los otros tres modelos se van a combinar, con el objetivo de que el sistema sea más preciso y robusto.

### 3.1.2. Variación de Parámetros en los Modelos *Fullbody* Y *Upperbody*

El **objetivo** de este experimento es evaluar los modelos elegidos anteriormente en función de sus parámetros de configuración. De este modo se puede ver qué ajuste de estos parámetros es más adecuado en nuestro caso.

Los parámetros que se quiere ajustar son: "profundidad" a la que se encuentra la persona, "mínimo de vecinos requeridos", "factor de escala" y "tamaño" mínimo y máximo de la persona encontrada. Los **datos** utilizados son los escenarios del *dataset* de un trabajo relacionado [2], que están ordenados de más sencillos a más difíciles (*Still*, *Translation*, *Arc*, *Rotation*), siendo el más fácil con el robot parado (*Still*) y el más difícil con el robot rotando (*Rotation*). Además, se han extraído las primeras imágenes de cada escenario para diseñar otros experimentos en los que sea más sencillo el reconocimiento (Figura 16), ya que hay una sola persona. En el anexo 4.1 se han añadido ejemplos de los demás escenarios. Estos experimentos se evalúan mediante la precisión y el *recall* que obtienen los modelos para detectar personas. Los resultados se analizan en más detalle a continuación.



Figura 16. Ejemplo de escenario fácil usado en los experimentos. Primera imagen del escenario, algunas imágenes intermedias y la última imagen del escenario "ArcFacil".

**Resultados** analizando cada uno de los parámetros de interés:

- Profundidad

Los modelos *FullBody* y *UpperBody* reconocen respectivamente el cuerpo entero y la parte superior del cuerpo de una persona. Con este experimento se quiere decidir si usar uno u otro en función de la distancia a la que se encuentre la persona.

A continuación, se muestran los **resultados** con el escenario *Arc* sin distinción de cerca o lejos (Tabla 2 y Figura 17) y con distinción de cerca y lejos en 2m (Tabla 3 y Figura 18). En el anexo 4.1 se adjuntan los resultados con el resto de escenarios.

En las tablas y gráficas, el parámetro "OverLapTH" es el tanto por ciento de solapamiento que hay entre la persona encontrada por nosotros y la persona real. En este experimento ese parámetro se ha usado para comprobar cómo cambian la precisión y el *recall* en función de con qué porcentaje de acierto se toma por buena una imagen.

OverlapTH	FullBody		UpperBody	
	Precision	Recall	Precision	Recall
0,50	0,531	0,024	0,682	0,157
0,55	0,448	0,020	0,658	0,151
0,60	0,400	0,018	0,626	0,144
0,65	0,297	0,013	0,585	0,135
0,70	0,207	0,009	0,542	0,125
0,75	0,103	0,005	0,495	0,114

Tabla 2. Resultados obtenidos por los métodos de reconocimiento en función del umbral de superposición "OverLapTH", en el escenario Arc. Sin hacer distinción de distancia al robot.

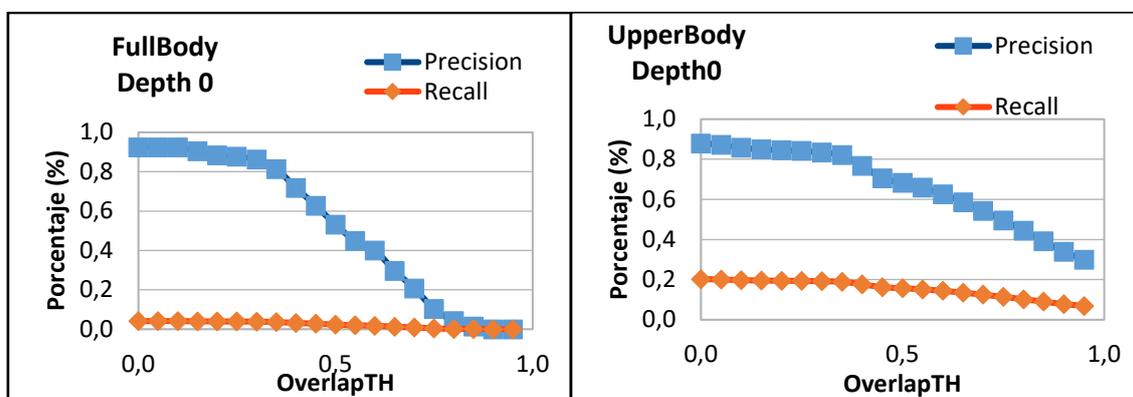


Figura 17. Gráficas que muestran la precisión y el *recall* de los métodos de reconocimiento en función del umbral de superposición. Sin distinción de distancia al robot "Depth 0".

OverlapTH	FullBody				UpperBody			
	Cerca		Lejos		Cerca		Lejos	
	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall
0,5	0,200	0,005	0,565	0,033	0,562	0,227	0,486	0,174
0,55	0,200	0,005	0,393	0,023	0,537	0,218	0,435	0,155
0,6	0,200	0,005	0,286	0,017	0,518	0,211	0,438	0,157
0,65	0,120	0,003	0,280	0,016	0,475	0,192	0,467	0,167
0,7	0,200	0,005	0,179	0,010	0,544	0,219	0,424	0,152
0,75	0,120	0,003	0,119	0,007	0,525	0,213	0,287	0,103

Tabla 3. Resultados obtenidos por los métodos de reconocimiento en función del umbral de superposición "OverLapTH", en el escenario Arc. Con una diferenciación de cerca y lejos en 2m

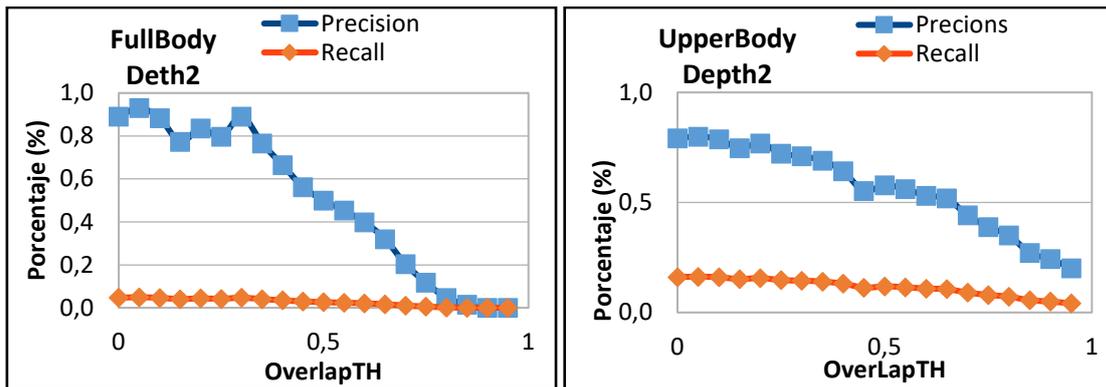


Figura 18. Gráficas de la precisión y el *recall* en función del umbral de solapamiento. Con una diferenciación entre cerca y lejos en 2m "Depth2".

Cómo se muestra en la Tabla 3 el modelo *FullBody* obtiene mejores resultados de lejos, mientras que el modelo *UpperBody* obtiene mejores resultados de cerca. Por lo tanto, se ha decidido que en los experimentos posteriores se va a usar la distancia 2 m para separar lo que es cerca de lo que es lejos.

- Mínimo número de vecinos requerido (*minNeig*)

Para este experimento se ha usado sólo el escenario "ArcFacil" (Figura 16), por ser suficientemente representativo. Se pueden ver los **resultados** obtenidos en este experimento en la Tabla 4 y la Figura 19.

minNeig	FullBody				UpperBody			
	Cerca		Lejos		Cerca		Lejos	
	Precisión	Recall	Precisión	Recall	Precisión	Recall	Precisión	Recall
2	0,00	0,00	0,69	0,32	1,00	0,07	0,00	0,00
3	0,00	0,00	0,74	0,23	1,00	0,04	0,00	0,00
4	1,00	0,00	0,73	0,16	1,00	0,02	0,00	0,00
5	1,00	0,00	0,88	0,10	1,00	0,00	0,00	0,00
6	1,00	0,00	1,00	0,03	1,00	0,00	0,00	0,00
7	1,00	0,00	1,00	0,01	1,00	0,00	0,00	0,00
8	1,00	0,00	1,00	0,00	1,00	0,00	0,00	0,00
9	1,00	0,00	1,00	0,00	1,00	0,00	1,00	0,00

Tabla 4. Resultados obtenidos por los métodos de reconocimiento en función del parámetro mínimo de vecinos requerido.

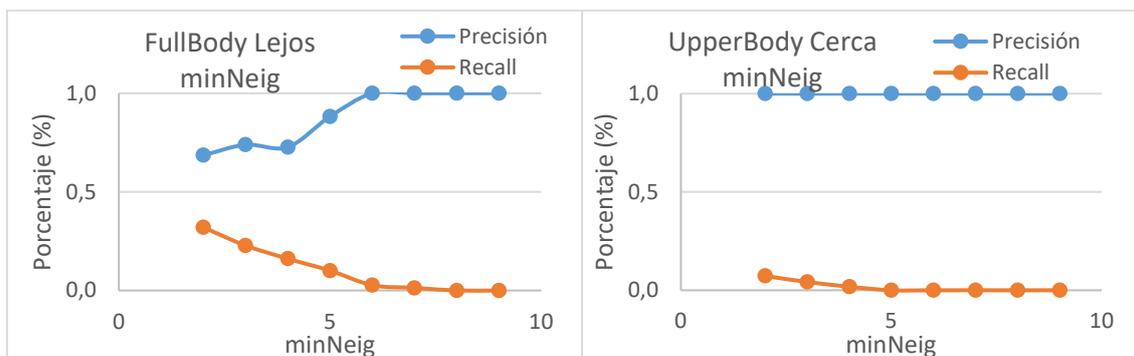


Figura 19. Gráficas de la Precisión y el *Recall* en función del parámetro mínimo de vecino requerido.

El parámetro mínimo de vecinos requerido (minNeig) se ha dejado con un valor de 2, porque con otros valores el *recall* es demasiado bajo, es decir reconoce a muy pocas personas. Aunque las que encuentre sean muy probablemente personas, se corre el peligro de no encontrar a nadie.

- Factor de escala (ScaleFactor)

Al igual que en el experimento anterior, sólo se ha usado el escenario "ArcFacil". Los resultados obtenidos se muestran en la Tabla 5 y la Figura 20.

ScaleFactor	FullBody				UpperBody			
	Cerca		Lejos		Cerca		Lejos	
	Precisión	Recall	Precisión	Recall	Precisión	Recall	Precisión	Recall
1,1	0,00	0,00	0,69	0,32	1,00	0,07	0,00	0,00
1,2	1,00	0,00	0,75	0,10	1,00	0,01	0,00	0,00
1,3	1,00	0,00	0,75	0,02	1,00	0,02	0,00	0,00
1,4	1,00	0,00	0,25	0,01	1,00	0,01	0,00	0,00
1,5	1,00	0,00	0,20	0,01	1,00	0,00	0,00	0,00
1,6	1,00	0,00	0,33	0,01	1,00	0,00	0,00	0,00
1,7	1,00	0,00	1,00	0,09	1,00	0,00	0,00	0,00
1,8	1,00	0,00	1,00	0,00	1,00	0,00	0,00	0,00
1,9	1,00	0,00	0,50	0,02	1,00	0,00	0,00	0,00

Tabla 5. Resultados obtenidos por los métodos de reconocimiento en función del valor que toma el parámetro factor de escala.

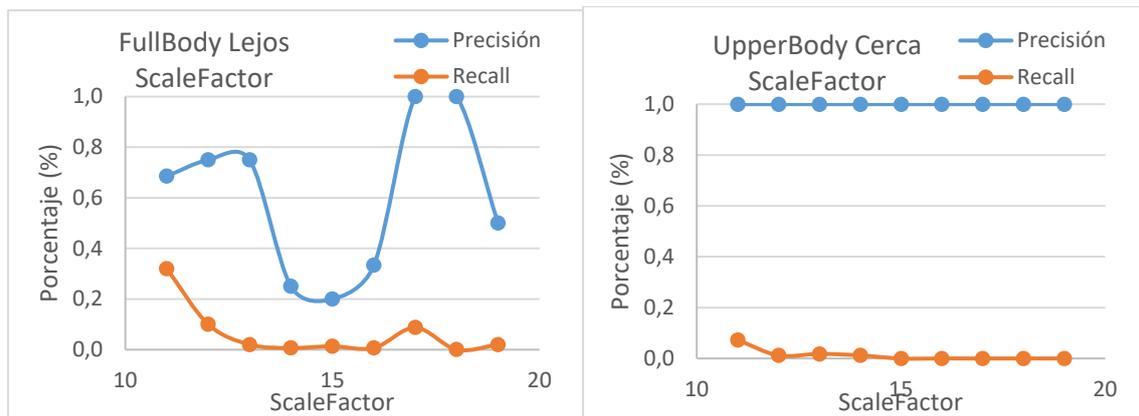


Figura 20. Gráficas de la precisión y el *recall* en función del factor de escala.

Tras ver los resultados obtenidos el parámetro de escala seleccionado es de 1,1 ya que con otros valores el *recall* es demasiado bajo.

- Tamaño mínimo y máximo de la persona encontrada

Este parámetro se refiere a qué tamaño tiene la persona encontrada en la imagen. Si la persona está lejos de la cámara será pequeña en la imagen, por otra parte, si la persona está cerca de la cámara será grande en la imagen.

Se han realizado pruebas con diferentes tamaños mínimos para determinar el tamaño adecuado para el modelo *FullBody*.

Se muestran los **resultados** obtenidos en el escenario *Arc* en las tablas siguientes, en el anexo 4.1 se adjuntan los resultados en el escenario *Still*. La Tabla 6 y la Figura 21 muestran los datos cuándo ambos parámetros del tamaño, (ancho y alto) se cambian a la vez. Y la Tabla 7 y la Figura 22 muestran los resultados cuándo el ancho se deja en 50 y el alto cambia.

minSize		FullBody			
		Cerca		Lejos	
Ancho	Alto	Precision	Recall	Precision	Recall
50	50	0,00	0,00	0,45	0,23
75	75	0,00	0,00	0,55	0,23
100	100	0,00	0,00	0,75	0,22
125	125	1,00	0,00	0,98	0,19
150	150	1,00	0,00	0,75	0,01
175	175	1,00	0,00	1,00	0,01
200	200	1,00	0,00	1,00	0,00
225	225	1,00	0,00	1,00	0,00
250	250	1,00	0,00	1,00	0,00

Tabla 6. Resultados cuándo el ancho y el alto del parámetro tamaño mínimo se cambian a la vez.

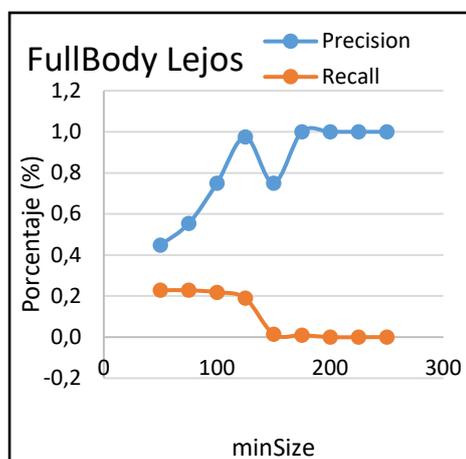


Figura 21. Precisión y recall en función del parámetro tamaño mínimo.

minSize		FullBody			
		Cerca		Lejos	
Ancho	Alto	Precision	Recall	Precision	Recall
50	50	0,00	0,00	0,45	0,23
50	75	0,00	0,00	0,45	0,23
50	100	0,00	0,00	0,45	0,23
50	125	0,00	0,00	0,55	0,23
50	150	0,00	0,00	0,55	0,23
50	175	0,00	0,00	0,69	0,23
50	200	0,00	0,00	0,75	0,22
50	225	1,00	0,00	0,94	0,22
50	250	1,00	0,00	0,98	0,19

Tabla 7. Resultados cuándo sólo se cambia el alto del parámetro tamaño mínimo.

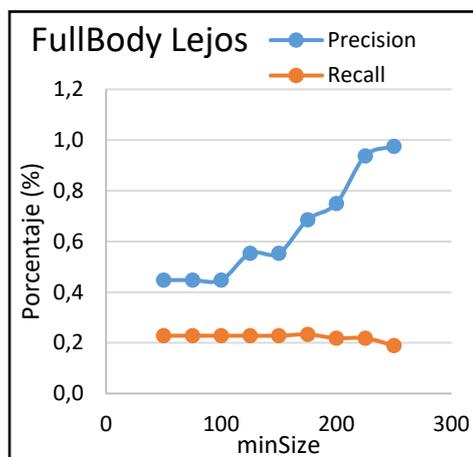


Figura 22. Precisión y recall en función del parámetro tamaño mínimo.

En vista de los resultados se ha decidido que el tamaño mínimo de las personas que encuentre el modelo *FullBody* (cuerpo entero) sea, 50-200 (50 de ancho y 200 de alto). Al modelo *UpperBody* (medio cuerpo) se le ha limitado el tamaño mínimo a 50-50 (50 de ancho y 50 de alto), porque es el mismo ancho que en el modelo *FullBody* y se considera que la parte superior de una persona podría ser cuadrada. En ambos casos el tamaño máximo no se ha limitado porque si la persona está muy cerca de la cámara puede ser muy grande, si el tamaño máximo se limitara se correría el peligro de no reconocer a algunas personas.

## 3.2. Evaluación Del Sistema De Elección y Seguimiento Del Objetivo

### 3.2.1. Elección del objetivo (persona)

El **objetivo** de este experimento es probar diferentes estrategias para seleccionar la persona a seguir (objetivo) entre todas las detectadas, y elegir cuál de estos métodos es el más apropiado.

Se han probado cuatro versiones distintas del método con tres escenarios diferentes. Estos escenarios se han grabado en el laboratorio de Robótica del I3A. Dos se han grabado con el robot parado (*StillUZ*, *Still2UZ*) y el otro con el robot moviéndose (*MoveUZ*) para poder probar el método en diferentes escenarios.

Las diferentes versiones del algoritmo de elección consisten en combinar o no los dos métodos principales, *Fullbody* y *Upperbody*, con la distancia a la que se encuentra la persona y el método *Frontalface*. En el Anexo 4.2, se adjuntan diagramas de bloques explicativos de las cuatro versiones.

- Versión 1. Se eligen primero las personas reconocidas con el método *Fullbody* y después las reconocidas con el método *Upperbody*, sin limitar la distancia a la que reconocen uno u otro y sin tener en cuenta si se ha encontrado o no una cara con el método *Frontalface* junto al cuerpo.
- Versión 2. En esta versión se limita la distancia a la que pueden reconocer los métodos *Fullbody* y *Upperbody*. Además, se añade un nivel de preferencia respecto a la versión anterior, si se ha encontrado cara dentro o cerca de cualquiera de los dos métodos principales se elige antes que el método principal sin cara.
- Versión 3. En la versión tres además de comprobar si hay cara cerca de los métodos principales, se comprobará si se ha encontrado un *Upperbody* y un *Fullbody* juntos, en cuyo caso se elegirá esa persona.
- Versión 4. En la versión cuatro, a diferencia de en las tres anteriores, se prioriza también en función de la distancia a la que se encuentra la persona. Si se encuentran dos personas con el mismo número de atributos (cara, cuerpo entero, medio cuerpo) se elige a la que esté más cerca.

A continuación, se muestran los **resultados** obtenidos en la Tabla 8 y en la Figura 23. En la Figura 24 se muestran algunas imágenes de los experimentos realizados.

	Still		Still2		Move	
	Positivos	Precisión	Positivos	Precisión	Positivos	Precisión
Version1	36	0,667	59	0,855	34	0,410
Version2	29	0,935	57	0,966	34	0,708
Version3	29	0,935	58	0,967	36	0,720
Version4	24	1,000	49	1,000	32	1,000

Tabla 8. Resultados del número de positivos encontrados por el algoritmo y del tanto por ciento de acierto en función del escenario y de la versión del algoritmo de elección.

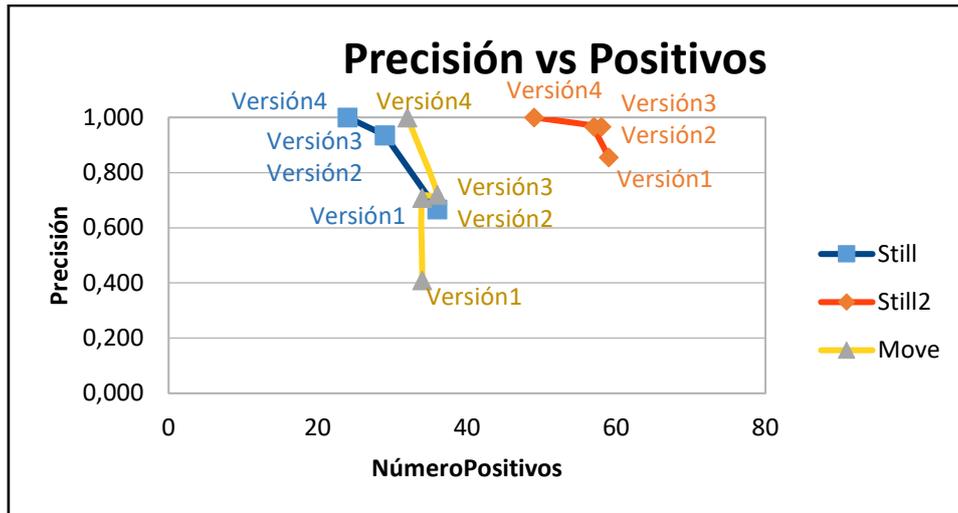


Figura 23. Gráfica que muestra la precisión y el número de positivos encontrados en los tres escenarios en función de la versión del algoritmo de elección que se ha usado. Cada punto de las líneas de regresión de los escenarios es una versión.

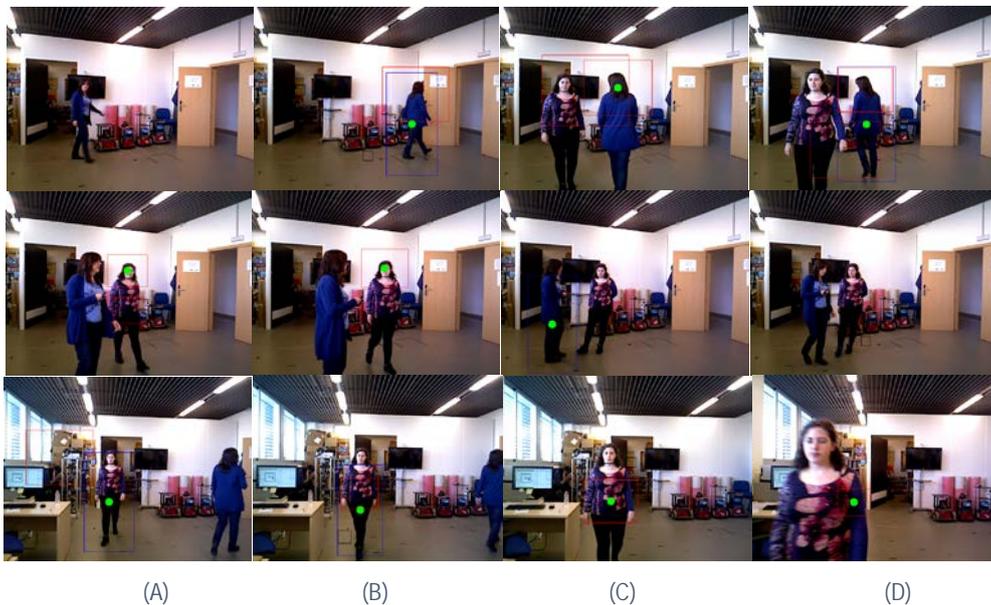


Figura 24. Algunas imágenes de los resultados obtenidos con los diferentes escenarios y versiones. 1ª fila, escenario *StillUZ* con la versión 1 del algoritmo de elección. 2ª fila, escenario *Still2UZ* con la versión 2 del algoritmo. 3ª fila, escenario *MoveUZ* con la versión 3 del algoritmo. (A) primera imagen del experimento, (B) imágenes intermedias y (C) última imagen del experimento.

Cómo se ve en la Tabla 8 en la última versión se obtienen muy buenos resultados, pero se encuentran bastantes menos personas que en las otras versiones, en los tres escenarios. Por esto la decisión que se ha tomado ha sido usar la última versión, pero ampliando el rango de distancia en la que los métodos *Fullbody* y *Upperbody* son válidos.

### 3.2.2. Error en la Estimación de Profundidad

El **objetivo** de este experimento es comprobar que error se comete al medir la distancia o profundidad a la que se encuentra la persona. Este error se deberá a que el sensor toma la distancia en diagonal, porque el centro de la persona no está a la misma altura que el sensor, y no se realizan los cambios necesarios para calcular la distancia en línea recta.

Para realizar este experimento se han grabado un conjunto de **datos** de dos escenarios en el laboratorio de Robótica I3A, colocando 10 marcas en el suelo. Con anterioridad se ha medido la distancia de las marcas a la posición del robot y, por último, si se ha devuelto más de una distancia para un mismo punto, se ha realizado la media de todas las distancias devueltas.

En la Tabla 9 y la Tabla 10 se muestran los **resultados** obtenidos en dos de los cuatro experimentos realizado, el resto de resultados se adjunta en el anexo 4.2. En la Figura 25 se muestran las primeras pruebas en los dos escenarios en los que se han realizado los experimentos. En las tablas siguientes la distancia real, es la medida que se ha tomado sobre el escenario real y la distancia obtenida, es la media de las distancias dadas por el algoritmo para cada punto, si no hay distancia significa que el algoritmo no ha encontrado a la persona en ese punto.

En la Figura 25 se muestran los resultados gráficos de los dos experimentos expuestos. En ambos casos el robot y el sensor están situados en el punto (0, 0).

La Figura 26 contiene algunas imágenes pertenecientes al primer experimento en el segundo escenario. En el anexo 4.2 se muestran algunas imágenes del resto de experimentos.

MARCA	Distancia Real (m)	Distancia Obtenida (m)	ERROR
1	5,4778	-	-
2	4,4331	3,8283	0,6048
3	3,6225	3,4821	0,1404
4	3,9836	3,6550	0,3286
5	3,6456	3,2517	0,3939
6	4,3916	3,8190	0,5727
7	2,0307	1,8796	0,1511
8	2,1832	-	-
9	2,6112	2,2995	0,3118
10	1,2088	1,1788	0,0300

Tabla 9. Resultados del error en la medida de distancia en el experimento 1 en el escenario 1.

MARCA	Distancia Real (m)	Distancia Obtenida (m)	ERROR
1	3,3314	-	-
2	4,1362	3,3868	0,7494
3	3,6807	3,3117	0,3690
4	3,9696	-	-
5	3,7065	3,1407	0,5658
6	3,0203	2,7550	0,2653
7	2,3418	-	-
8	2,5300	2,1737	0,3563
9	2,0270	1,8478	0,1792
10	1,2928	-	-

Tabla 10. Resultados del error en la medida de distancia en el experimento 1 en el escenario 2.

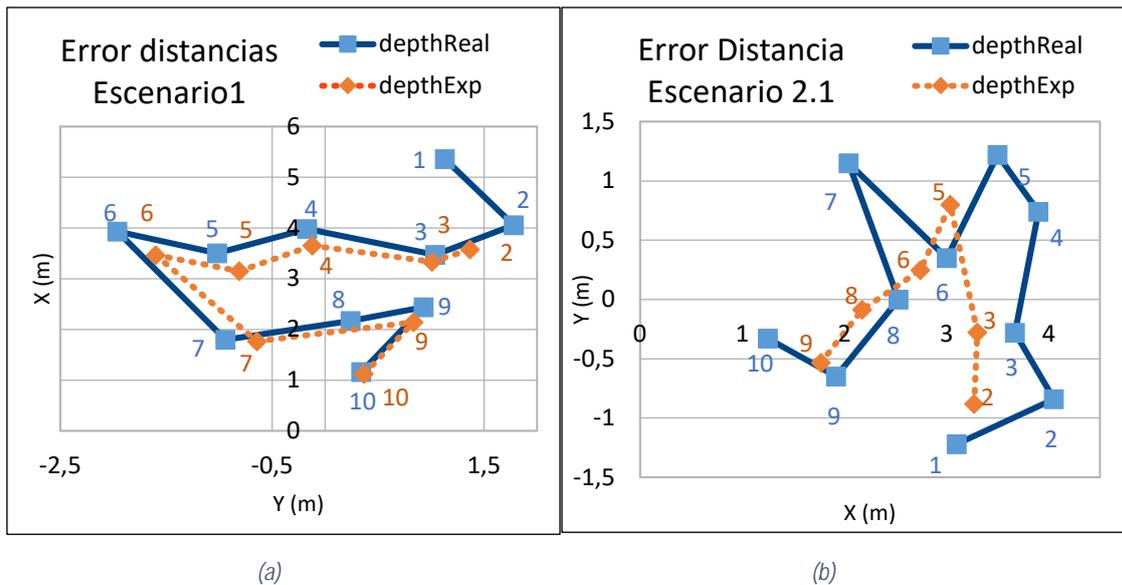


Figura 25. Representación gráfica de un experimento realizado en cada escenario, (a) escenario 1, (b) escenario 2.

Cómo muestran las tablas y los gráficos, los errores de distancia no son muy significativos en ningún caso. En el escenario 1 se reconoce mejor a la persona, esto se debe al tipo de escenario. El escenario 2 (Figura 26) está delante de unas ventanas y al recibir el sensor la luz de frente, el reconocimiento es peor.



Figura 26. Algunas imágenes de la prueba 1 en el escenario 2. La primera imagen se corresponde al punto uno de los resultados y la última al punto 10, siendo las imágenes intermedias algunos puntos intermedios. Los cuadrados rojos y azules son los objetivos encontrados por el reconocedor y el punto verde señala el objetivo elegido para ser seguido.

### 3.3. Evaluación Del Sistema Completo

El **objetivo** de este experimento es probar el sistema completo en una plataforma robótica móvil, y ajustar el algoritmo final para que el reconocimiento de personas y el acercamiento a ellas sea lo más óptimo posible.

### 3.3.1. Experimentos de integración 1



Figura 27. Diagrama de bloques explicativo del método usado en los primeros experimentos con el sistema completo.

En estos experimentos iniciales se ha programado el sistema para que empiece a buscar personas cuándo se le dé la orden, y una vez haya reconocido a alguien, vaya a las coordenadas de la persona, y pare. En la Figura 27 se muestra este proceso. Los experimentos se han realizado en el laboratorio, con distintas orientaciones, para tener varios escenarios. En la Figura 28 se muestran algunas imágenes de vídeos realizados durante los experimentos grabados, donde se ve al robot marcado en rojo y la persona a la que se acerca marcada en azul, en tres escenarios distintos.



Figura 28. Imágenes de cuatro experimentos, uno en cada línea siguiendo el siguiente orden (Prueba1, Prueba 3, Prueba 4 y Prueba 8). Se usan tres escenarios: escenario 1: fila 1, escenario 2: fila 2 y 4 y escenario 3: fila 3. En cada experimento se puede ver la posición inicial del robot en rojo y la persona en azul (A), en medio imágenes intermedias en las que el robot se acerca a la persona (B), (C) y al final la posición en la que acaban el robot y la persona (D).

En el primer experimento se comprobó que el robot encontraba a la persona, pero que era incapaz de llegar hasta las coordenadas de dicha persona. Esto se debe a que las coordenadas eran demasiado exactas y el robot no puede llegar al lugar dónde hay un obstáculo (la persona). Por esto se decide poner un offset, que garantice que las coordenadas a las que debe llegar el robot no son las coordenadas de un obstáculo.

En los experimentos posteriores se comprueba que el robot llega bien a la persona, aunque en algún caso tarda más de lo deseado en encontrar a dicha persona. En uno de los escenarios (primera fila

de la Figura 29) el reconocimiento es especialmente lento, esto se debe al tipo de luz del escenario, las ventanas están delante de la cámara y provocan reflejos.

De los diez experimentos realizados, dos han sido fallidos, en uno se dirige hacia un punto donde no hay ninguna persona (cuarta fila Figura 29) y en otro es incapaz de dirigirse a las coordenadas que le han mandado (primera prueba).

### 3.3.2. Experimentos Detección de Objetivo y Acercamiento Iterativo

A continuación, se han hecho experimentos con el sistema programado igual que antes. Pero en este caso cada vez que se aproxima a un objetivo se le ordena que inicie una nueva búsqueda. En la Figura 29 se muestra un diagrama del algoritmo con el que se realizan estos experimentos.

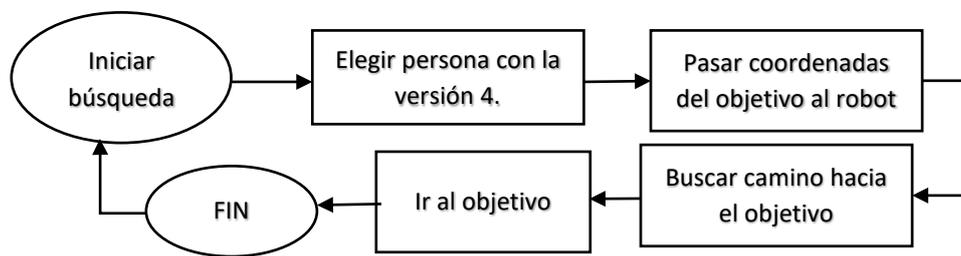


Figura 29. Diagrama de bloques explicativo del método usado en los experimentos de detección y acercamiento iterativos.

En la Figura 30 se muestran imágenes que ha grabado el robot en algunos experimentos de detección y acercamiento iterativo de forma manual. En el anexo 4.3 se han adjuntado algunas imágenes de los vídeos de estos y otros experimentos.

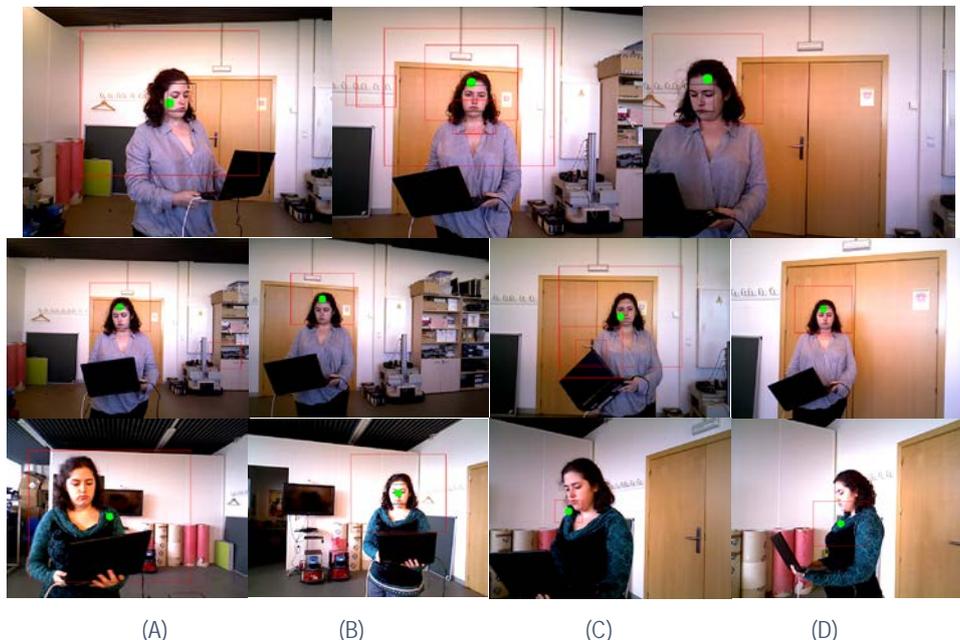


Figura 30. Imágenes tomadas por el robot en los experimentos, que muestran los sucesivos reconocimientos que se han realizado en el experimento. Los cuadrados rojos son los objetivos encontrados por el reconocedor y el punto verde es el objetivo elegido para acercarse a él. Cada fila se corresponde a un experimento, donde se ven la primera posición de la persona (A), las posiciones mientras el robot se acerca a la persona (B), (C) y la última posición (D). Estos experimentos se han realizado en el laboratorio, pero cambiando la posición del robot en el mismo.

En estos experimentos intermedios se comprueba que el robot es capaz de acercarse sucesivamente a una persona, aunque, lentamente debido a la necesidad de ordenarle que busque persona cada vez que llega a unas coordenadas. Además, en estos experimentos no se ha usado la segunda forma de elección y esto podría provocar que el robot cambie de persona si se cruza alguien por detrás.

### 3.3.3. Experimentos Finales

Por último, se han realizado unos experimentos que realizan un seguimiento continuado. Es decir, el sistema empieza a buscar persona cuándo se le da la orden y deja de buscar cuándo reconoce a alguien, igual que en los anteriores. Sin embargo, cuándo la plataforma móvil llega a su objetivo el sistema comprueba donde está la persona otra vez, para que, si se ha desplazado, el robot se siga acercando a ella. En la Figura 31 se muestra el diagrama del algoritmo con el que se han realizado estos experimentos.

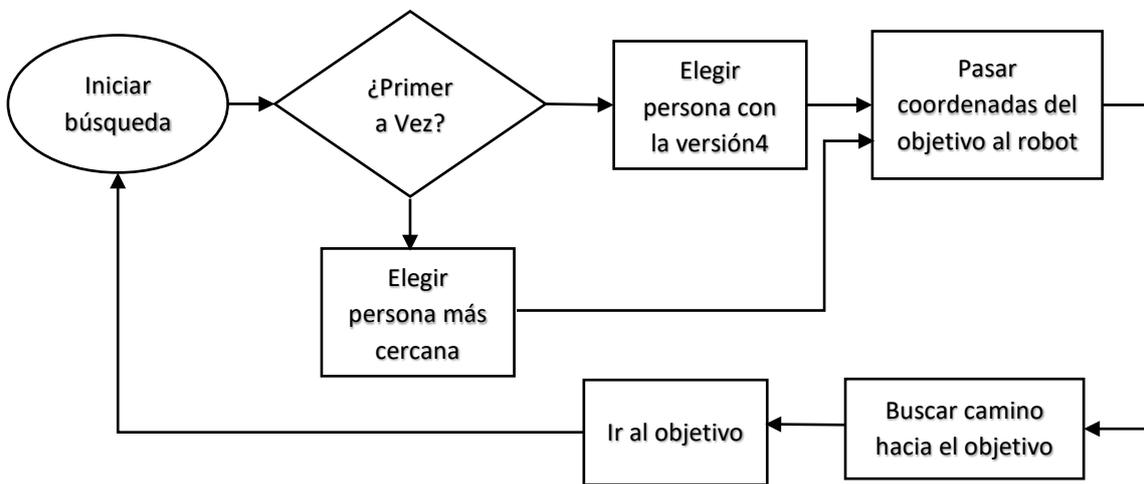


Figura 31. Diagrama de bloques explicativo del método usado en los experimentos finales.

En estos experimentos, se ha modificado el sistema para que cuando detecta a una persona diferencie si es la primera vez que busca persona o no, es decir, si acaba de arrancar, o si ya había realizado un primer seguimiento. La primera vez se elige el objetivo que más atributos encontrados tenga, la segunda y posteriores se elige el objetivo que más cerca del robot este, tal como se explica en la sección 2.3, *Elección Y Seguimiento Del Target*. En la Figura 32 se muestran imágenes grabadas por el robot en algunos de los experimentos finales. En el anexo 4.3 se adjuntan imágenes de los vídeos grabados en los experimentos finales.

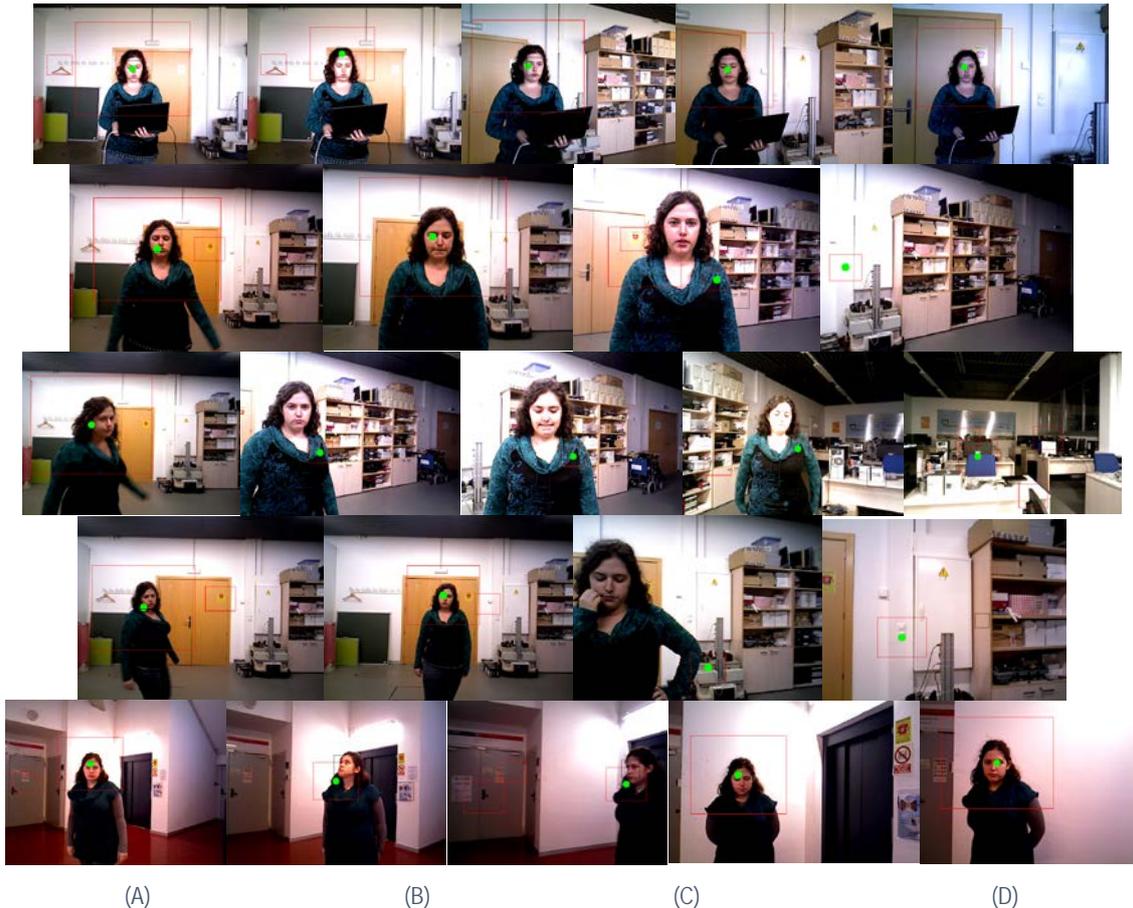


Figura 32. Imágenes tomadas por el robot en los experimentos finales, que muestran los sucesivos reconocimientos automáticos que se han realizado en el experimento. Los cuadrados rojos son los objetivos encontrados por el reconocedor y el punto verde es el objetivo elegido para acercarse a él. Cada fila se corresponde a un experimento, dónde se ven la primera posición de la persona (A), las posiciones mientras el robot se acerca a la persona (B), (C) y la última posición (D). Estos experimentos se han realizado en dos escenarios diferentes, los primeros cuatro experimentos se han realizado en el laboratorio y el último en un pasillo.

Cómo muestran las imágenes en la mayoría de los casos el robot reconoce a la persona y va hacia ella, pero hay alguna ocasión en la que el reconocedor falla y el robot se dirige hacia un punto que no es una persona. El primer experimento y el último se acaban con el robot al lado de la persona. Sin embargo, al final de los otros tres experimentos el robot reconoce cómo persona parte del mobiliario. El algoritmo final resulta satisfactorio ya que es capaz de seguir a una persona en un espacio cerrado. En estos experimentos finales el seguimiento de la persona, una vez que se ha acercado a ella por primera vez, resulta bastante rápido, en ocasiones tanto que el robot no llega a moverse porque la persona no se ha movido. Además, al haberse implementado un segundo modo para la elección de persona en la segunda y sucesivas veces es más probable que el robot siga a la misma persona. Por otra parte, al estar la persona cerca de la cámara el reconocimiento es más eficiente.

## 4. Conclusiones

### 4.1. Conclusiones del Prototipo y Trabajo Futuro

En este trabajo se ha diseñado y construido un sistema en el que una plataforma móvil robótica se acerca a una persona, buscando dónde está y navegando hacia sus coordenadas. El sistema se sigue acercando si cuando llega al objetivo, detecta que la persona se ha movido y ahora está en otro sitio.

El resultado final es satisfactorio, tal y como se ha visto en el capítulo 3, y sirve como prototipo para sobre el construir un robot guía en una aplicación real, por ejemplo, en un museo. Si se le colocara una pantalla a la plataforma móvil en su parte superior, esta podría informar al usuario al cual se acerca, o incluso interactuar con él a través una vez se hubiera acercado a él.

La aplicación final implementada usa tres métodos incluidos en las librerías de openCV llamadas "Frontalface", "Upperface" y "Fullbody", que reconocen respectivamente la cara, la parte de arriba del cuerpo y el cuerpo entero de una persona. Se ha decidido usar este sistema porque los tres métodos se complementan entre ellos, el Upperbody reconociendo a las personas que están más cerca, el Fullbody reconociendo a las personas que están lejos y el método Frontalface para añadir precisión al reconocimiento.

El prototipo tiene aspectos que se podría mejorar de cara a utilizar en una aplicación real final. En la parte de reconocimiento de persona, podría implementarse un reconocedor más robusto, con mayor precisión, para que la plataforma móvil nunca se moviera hacia un sitio en el que no hubiera persona. En el punto de elección de persona a la que seguir, se podría haber desarrollado un algoritmo que fuera más parecido al *tracking* que aparece en los trabajos de la literatura reciente. Si la segunda vez y sucesivas, que se ordena la búsqueda de persona, se hubiera usado un método basado en histogramas de color para la elección de la persona a seguir, habría sido más seguro que la plataforma móvil estaba siguiendo a la misma persona que antes.

### 4.2. Conclusiones personales

Con este trabajo he aprendido a manejar el lenguaje de programación basado en C++ que usa OpenCV sobre eclipse y ROS y a desenvolverme en un sistema operativo basado en el *kernel* de Linux. Además del manejo de una cámara con sensor RGB-D y una plataforma móvil robótica (TurtleBot). En la parte de visión por computador he aprendido diferentes técnicas para el manejo de imágenes en OpenCV y el reconocimiento de personas en esas imágenes.

Este trabajo me ha aportado experiencia en el campo de la robótica y de visión por computador y, además, aprender la metodología de trabajo de un proyecto de experimentación en el cuál se debe llevar un seguimiento exhaustivo y formal de los pasos realizados y resultados obtenidos.

# Anexo I

ROS (*Robot Operating System*)<sup>2</sup> provee librerías y herramientas para ayudar a los desarrolladores de software a crear aplicaciones para robots. ROS provee abstracción de hardware, controladores de dispositivos, librerías, herramientas de visualización, comunicación por mensajes, administración de paquetes y más. ROS está bajo la licencia open source, BSD.

ROS ha sido creado para facilitar a una colaboración total entre desarrolladores de software para robótica. Por ejemplo, un laboratorio podría tener expertos en "mapeado" de entornos cerrados, y podría contribuir a sistemas mundiales para producir mapas. ROS ha sido diseñado específicamente para grupos de trabajo como estos de colaboración y construcción entre ellos.

En la Figura 33 se muestra de forma esquemática el sistema que se usa en este trabajo para comunicar las órdenes del ordenador con el ordenador, usando ROS como interfaz.

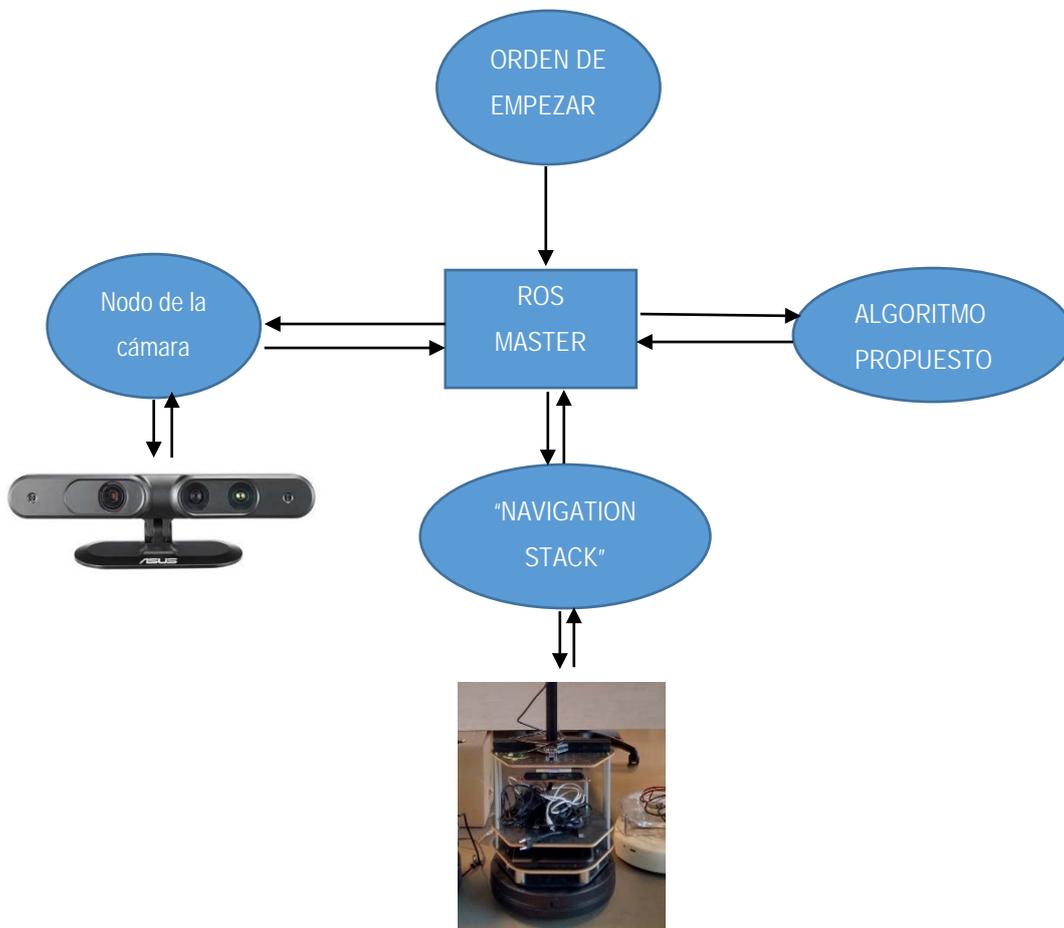


Figura 33. Diagrama de bloques que muestra de qué forma se usa ROS en este trabajo.

<sup>2</sup> <http://www.ros.org>

## Anexo II

La plataforma móvil robótica que se ha usado en este proyecto es el robot llamado TurtleBot<sup>3</sup>. Este robot es un kit personal de bajo coste con software libre para desarrollar aplicaciones robóticas.

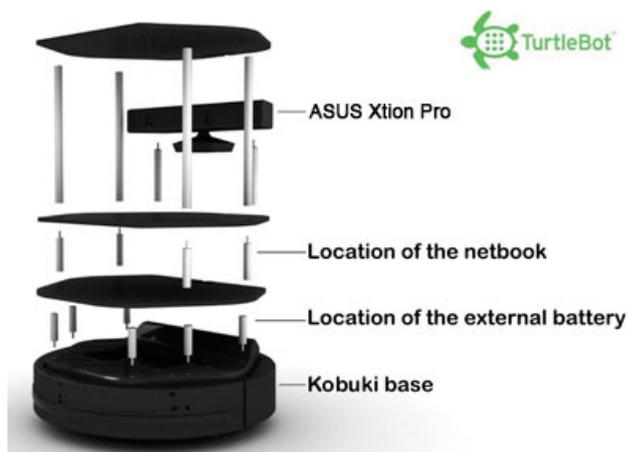


Figura 34. Partes que componen la plataforma robótica TurtleBot.

En la Figura 34 se muestran las partes de las que se compone el TurtleBot. Se basa en el robot aspirador de la marca iRobot llamado Roomba, esta base se llama Kobuki base (Figura 35). Después, está formada por una estructura de barras y plataformas, que se usan para dejar el portátil con el algoritmo de funcionamiento que se conecta a la base para mandar las órdenes al robot. Esta estructura también servirá para situar un sensor ASUS Xtion Pro algo más levantada que el nivel del suelo.



Figura 35. Conexiones de la base del TurtleBot (Kobuki base).

En la Figura 35 se muestran los conectores de los que se compone la base del TurtleBot. Se compone de un conector usb para conectarlo al portátil y transmitirle las órdenes a través de la plataforma ROS.

---

<sup>3</sup> <http://www.turtlebot.com/>

## Anexo III

En este proyecto se ha usado una cámara de la marca Asus llamada Xtion PRO LIVE, que consta de un sensor RGB y un sensor de profundidad (Figura 4), esta cámara se ha usado para captar imágenes en formato de color (RGB) y para saber también la profundidad de cada pixel de la imagen captada. El emisor de infrarrojos genera una cuadrícula que envía a la escena y el receptor recibe esa cuadrícula, sabiendo el patrón que sigue, para calcular la profundidad de cada pixel de la escena.



Figura 36. Cámara Asus Xtion pro Live

En la Figura 36 se muestran los sensores que posee la cámara. A la izquierda de la figura el emisor de infrarrojos y a la derecha los receptores de luz roja y luz infrarroja, para captar la escena en color y para captar las profundidades de la escena respectivamente.

Sensor	RGB y profundidad
Profundidad del tamaño de la imagen	VGA (640x480): 30fps QVGA (320x240): 60fps
Resolución	SXGVA (1280*1024)
Campo de visión	58°H, 45°V, 70°D (Horizontal, Vertical, Diagonal)
Distancia de uso	Entre 0,8 m y 3,5 m

Tabla 11. Especificaciones técnicas cámara Asus Xtion pro Live

En la Tabla 11 se muestran algunos parámetros del sensor usado. En primer lugar, la resolución que tendrá la imagen captada por el sensor y su tamaño en pixeles. A continuación, el campo de visión del sensor, cuál es la escena que capta el sensor. Y, por último, el sensor de profundidad tiene un rango de 0,8 m a 3,5 metros si algún objeto de la imagen se encuentra fuera de este rango, su profundidad se ajustará al mínimo y máximo en función de por dónde haya sobrepasado los límites.

## Anexo IV

En este anexo se adjuntarán todos los resultados, datos o aclaraciones que no se han introducido en el capítulo 3 para que este no fuera demasiado extenso.

### 1. Evaluación Del Sistema de Reconocimiento de Personas

- Imágenes sacadas de los escenarios usados en estos experimentos.
  - ❖ Still:



Figura 37. Imágenes pertenecientes al escenario Still, en el cuál la cámara permanece quieta.

- ❖ Translation:



Figura 38. Imágenes pertenecientes al escenario Translation, en el cuál la cámara se mueve describiendo una línea recta.

❖ Arc:



Figura 39. Imágenes pertenecientes al escenario Arc, en el cuál la cámara describe una trayectoria en forma de arco.

❖ Rotation:



Figura 40. Imágenes pertenecientes al escenario Rotation, en el cuál la cámara rota sobre sí misma.

- Resultados del experimento variación del parámetro distancia en los métodos de reconocimiento.

❖ Still:

OverLapTH	FullBody		UpperBody	
	Precision	Recall	Precision	Recall
0,50	0,710	0,035	0,893	0,332
0,55	0,601	0,030	0,873	0,324
0,60	0,482	0,024	0,842	0,313
0,65	0,342	0,017	0,801	0,297
0,70	0,207	0,010	0,748	0,278
0,75	0,124	0,006	0,670	0,249

Tabla 12. Resultados con diferentes porcentajes de acierto sin hacer distinción de cerca y lejos.

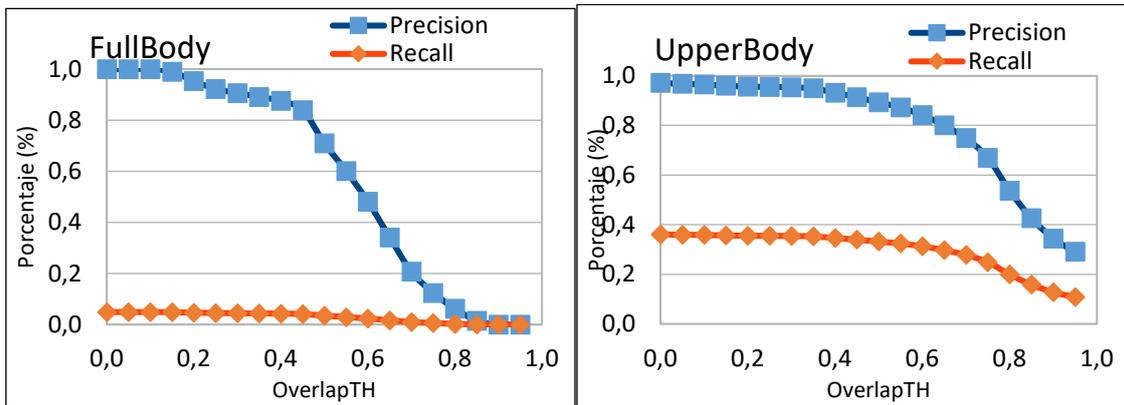


Figura 41. Precisión y recall en función del porcentaje de acierto, sin distinción de cerca y lejos.

OverLapTH	FullBody				UpperBody			
	Cerca		Lejos		Cerca		Lejos	
	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall
0,5	0,200	0,005	0,565	0,033	0,562	0,227	0,486	0,174
0,55	0,200	0,005	0,393	0,023	0,537	0,218	0,435	0,155
0,6	0,200	0,005	0,286	0,017	0,518	0,211	0,438	0,157
0,65	0,120	0,003	0,280	0,016	0,475	0,192	0,467	0,167
0,7	0,200	0,005	0,179	0,010	0,544	0,219	0,424	0,152
0,75	0,120	0,003	0,119	0,007	0,525	0,213	0,287	0,103

Tabla 13. Resultados con diferentes porcentajes de acierto haciendo distinción de cerca y lejos en dos metros.

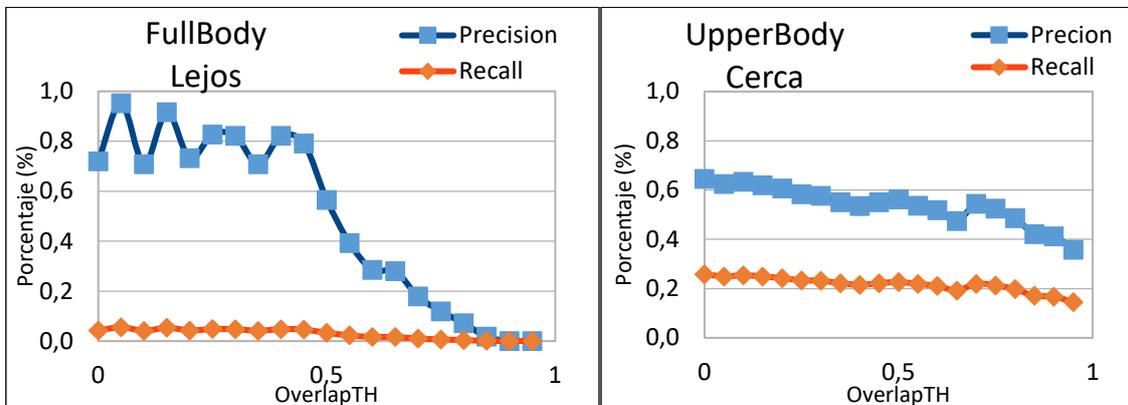


Figura 42. Precisión y recall en función del porcentaje de acierto, haciendo distinción de cerca y lejos en dos metros.

❖ Translation:

OverLapTH	FullBody		UpperBody	
	Precision	Recall	Precision	Recall
0,50	0,751	0,036	0,628	0,281
0,55	0,608	0,029	0,614	0,275
0,60	0,365	0,018	0,595	0,266
0,65	0,221	0,011	0,577	0,258
0,70	0,160	0,008	0,545	0,244
0,75	0,110	0,005	0,509	0,228

Tabla 14. Resultados con diferentes porcentajes de acierto sin hacer distinción de cerca y lejos.

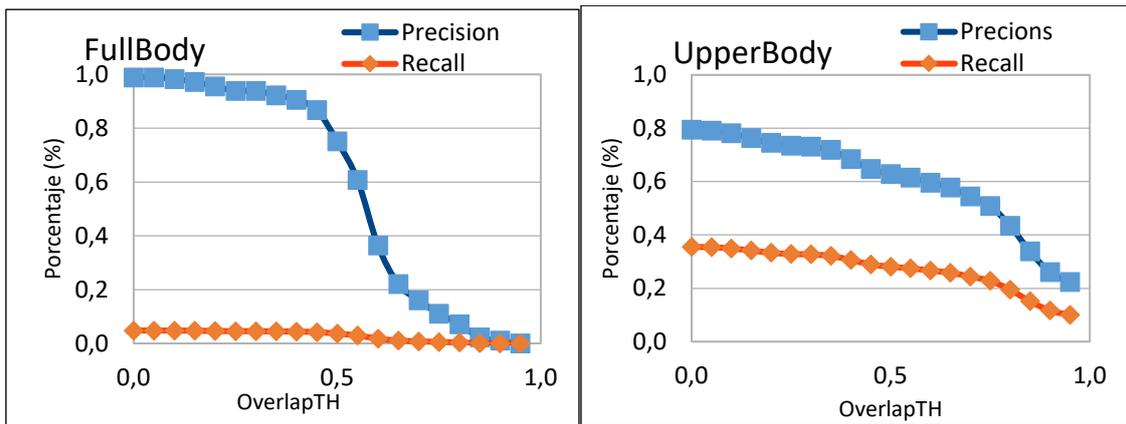


Figura 43. Precisión y recall en función del porcentaje de acierto, sin distinción de cerca y lejos.

OverLapTH	FullBody				UpperBody			
	Cerca		Lejos		Cerca		Lejos	
	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall
0,50	0,125	0,002	0,467	0,030	0,345	0,139	0,387	0,179
0,55	0,063	0,001	0,461	0,029	0,272	0,110	0,413	0,190
0,60	0,000	0,000	0,279	0,018	0,336	0,137	0,318	0,147
0,65	0,000	0,000	0,224	0,014	0,347	0,141	0,293	0,135
0,70	0,000	0,000	0,176	0,011	0,332	0,135	0,279	0,129
0,75	0,000	0,000	0,121	0,008	0,302	0,123	0,250	0,115

Tabla 15. Resultados con diferentes porcentajes de acierto haciendo distinción de cerca y lejos en dos metros.

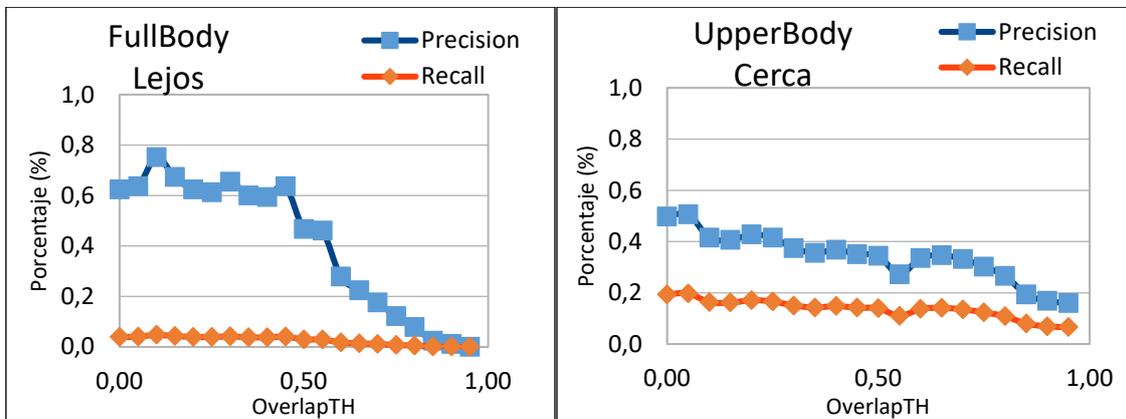


Figura 44. Precisión y recall en función del porcentaje de acierto, haciendo distinción de cerca y lejos en dos metros.

❖ Rotation:

OverLapTH	FullBody		UpperBody	
	Precision	Recall	Precision	Recall
0,50	0,586	0,032	0,560	0,178
0,55	0,452	0,025	0,524	0,166
0,60	0,329	0,018	0,493	0,156
0,65	0,238	0,013	0,449	0,142
0,70	0,152	0,008	0,406	0,129
0,75	0,090	0,005	0,337	0,107

Tabla 16. Resultados con diferentes porcentajes de acierto sin hacer distinción de cerca y lejos.

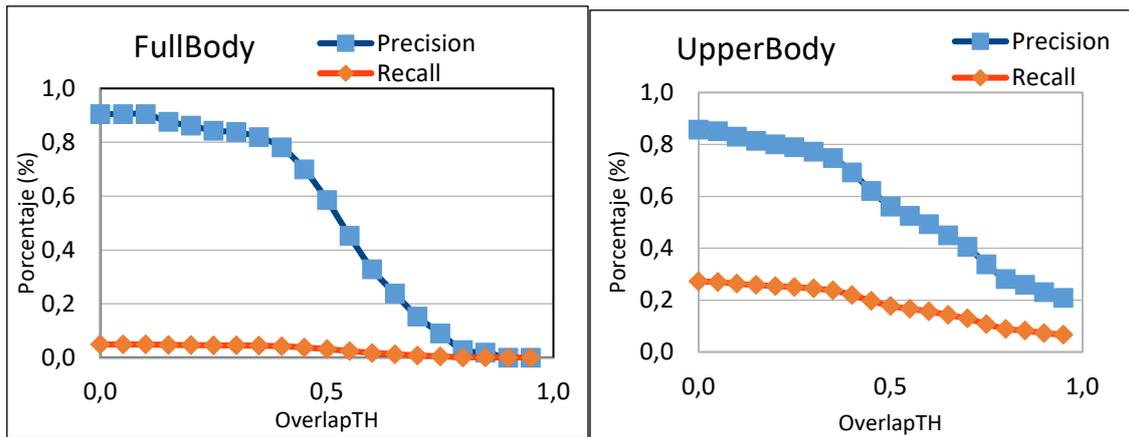


Figura 45. Precisión y recall en función del porcentaje de acierto, sin distinción de cerca y lejos.

OverLapTH	FullBody				UpperBody			
	Cerca		Lejos		Cerca		Lejos	
	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall
0,50	0,053	0,001	0,455	0,030	0,496	0,130	0,378	0,127
0,55	0,000	0,000	0,330	0,022	0,500	0,131	0,335	0,113
0,60	0,000	0,000	0,361	0,024	0,402	0,105	0,356	0,120
0,65	0,000	0,000	0,215	0,014	0,429	0,112	0,293	0,099
0,70	0,000	0,000	0,136	0,009	0,383	0,100	0,263	0,089
0,75	0,000	0,000	0,089	0,006	0,402	0,105	0,190	0,064

Tabla 17. Resultados con diferentes porcentajes de acierto haciendo distinción de cerca y lejos en dos metros.

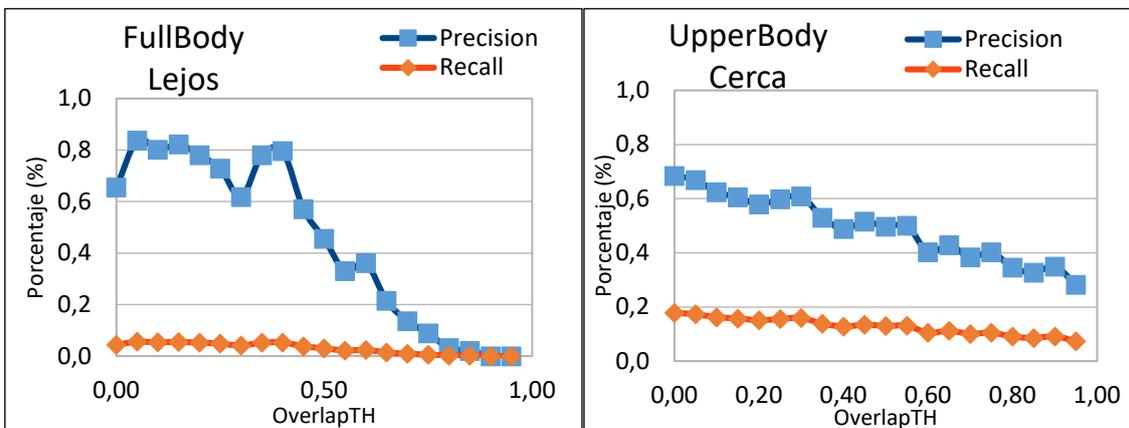


Figura 46. Precisión y recall en función del porcentaje de acierto, haciendo distinción de cerca y lejos en dos metros.

- Resultados de variar el parámetro tamaño mínimo en el método *Fullbody*.

- ❖ Still

minSize		FullBody			
minSize		Cerca		Lejos	
Ancho	Alto	Precision	Recall	Precision	Recall
50	50	0,000	0,000	0,638	0,208
75	75	1,000	0,000	0,662	0,208
100	100	1,000	0,000	0,800	0,229
125	125	1,000	0,000	1,000	0,135
150	150	1,000	0,000	0,875	0,057
175	175	1,000	0,000	1,000	0,033
200	200	1,000	0,000	1,000	0,000
225	225	1,000	0,000	1,000	0,000
250	250	1,000	0,000	1,000	0,000

Tabla 18. Resultados de cambiar el tamaño mínimo en el método *Fullbody*. Cambiando ancho y alto a la vez.

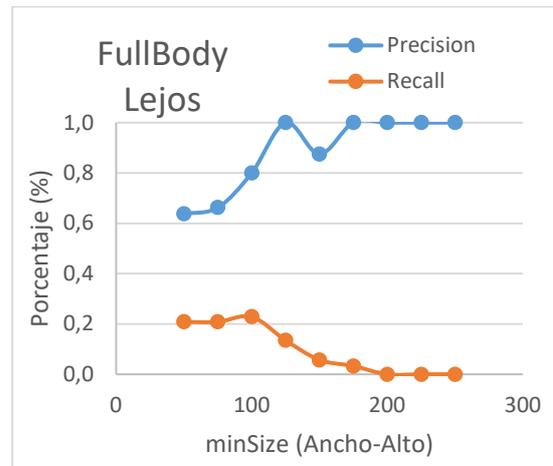


Figura 47. Precisión y *Recall* en función del tamaño mínimo.

minSize		FullBody			
minSize		Cerca		Lejos	
Ancho	Alto	Precision	Recall	Precision	Recall
50	50	0,000	0,000	0,638	0,208
50	75	0,000	0,000	0,638	0,208
50	100	0,000	0,000	0,638	0,208
50	125	0,000	0,000	0,646	0,208
50	150	1,000	0,000	0,662	0,208
50	175	1,000	0,000	0,730	0,220
50	200	1,000	0,000	0,800	0,229
50	225	1,000	0,000	0,912	0,212
50	250	1,000	0,000	1,000	0,135

Tabla 19. Resultados de cambiar el tamaño mínimo en el método *Fullbody*. Cambiando solo el alto.

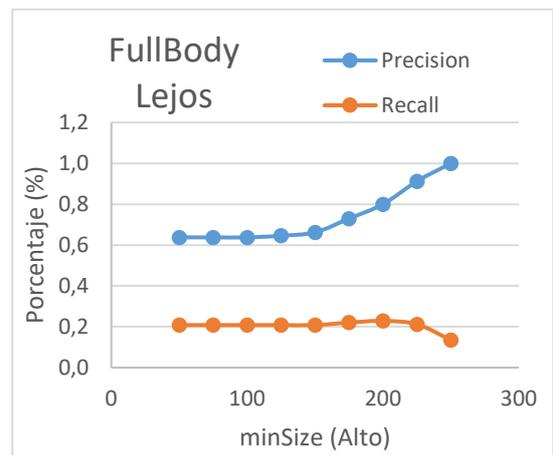


Figura 48. Precisión y *Recall* en función del tamaño mínimo.

## 2. Evaluación Del Sistema de Elección Y Seguimiento Del Target

- Diagramas de bloques de las versiones del algoritmo de elección.

❖ Versión 1

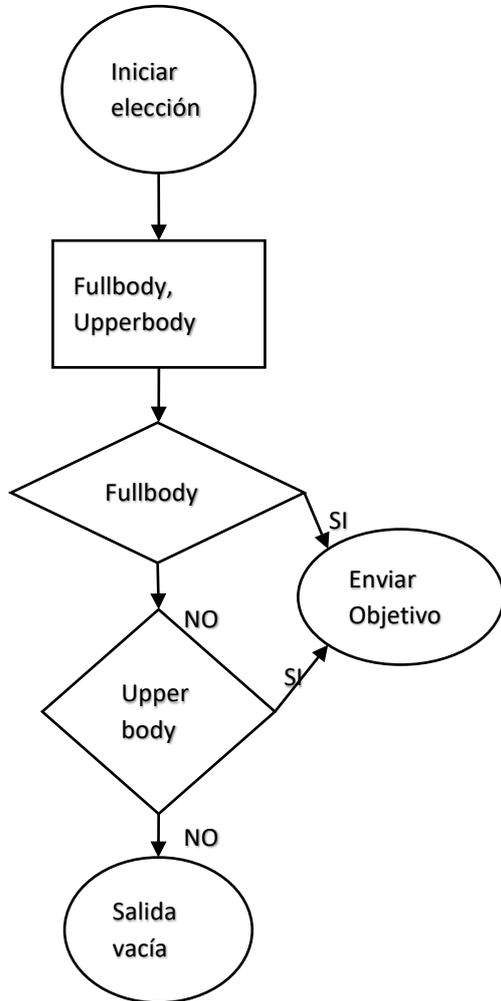


Figura 49. Diagrama de bloques de la versión 1 del algoritmo de elección de objetivo.

❖ Versión 2

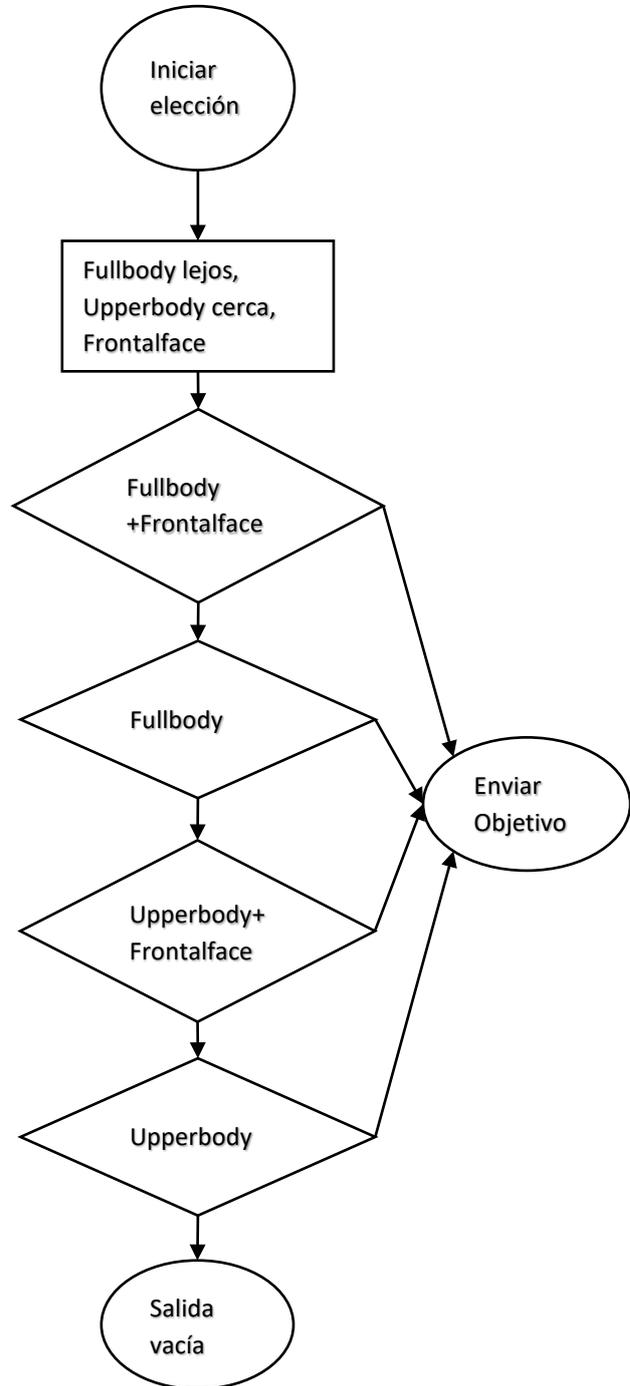


Figura 50. Diagrama de bloques de la versión 2 del algoritmo de elección de objetivo.

❖ Versión 3

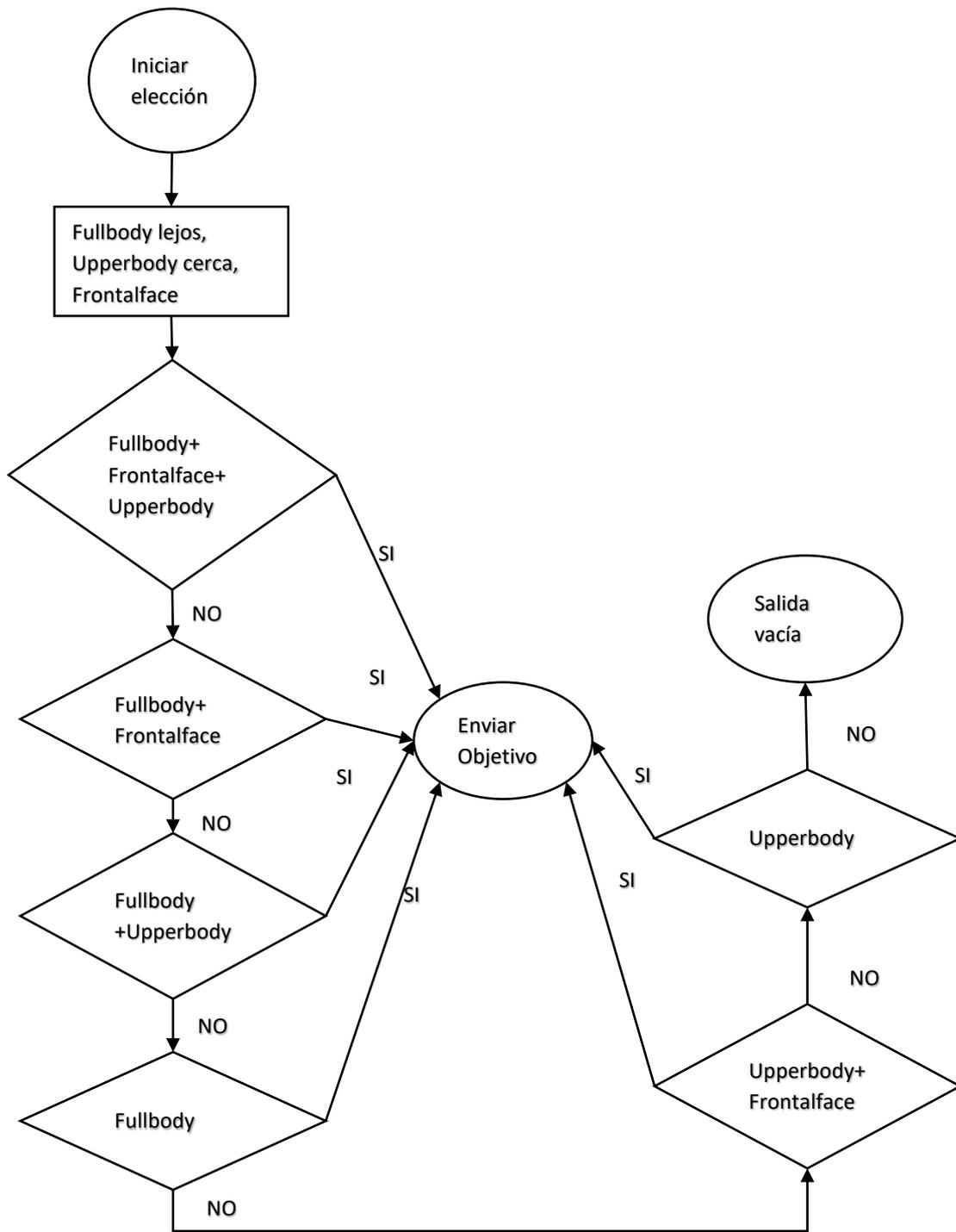


Figura 51. Diagrama de bloques de la versión 3 del algoritmo de elección de objetivo.

❖ Versión 4

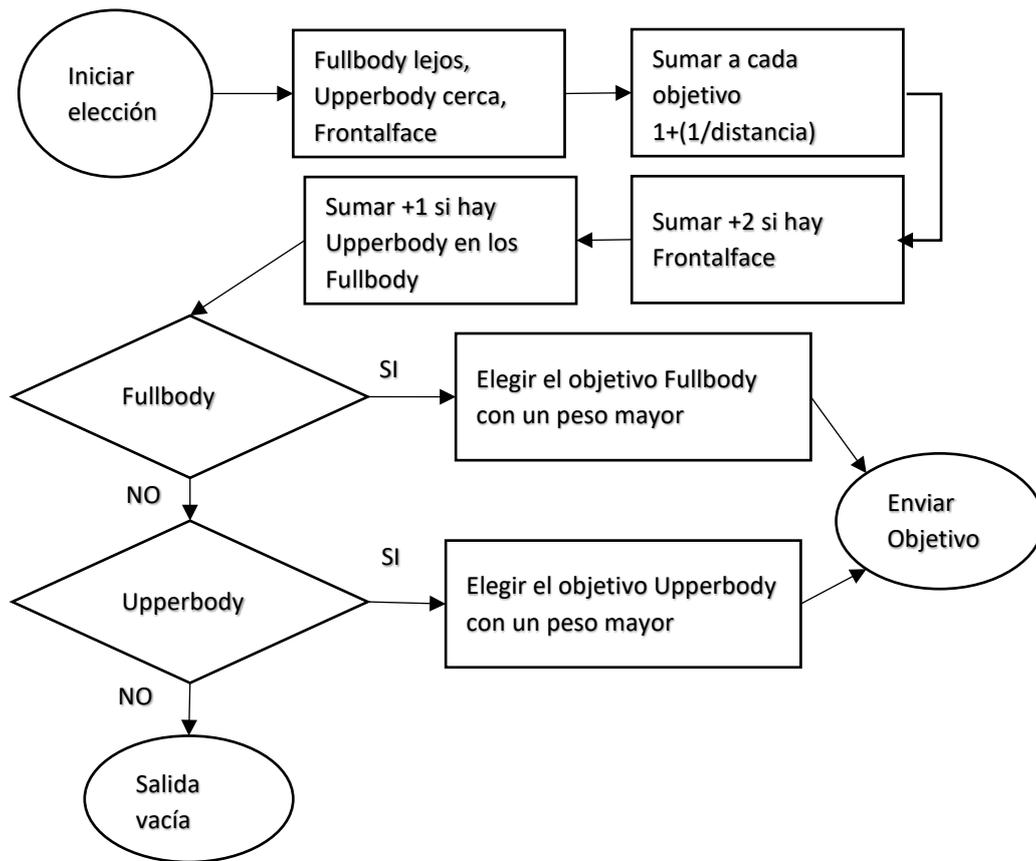


Figura 52. Diagrama de bloques de la versión 4 del algoritmo de elección de objetivo.

- Imágenes de los experimentos de elección de objetivo con distintas versiones.

❖ Versión 1



Figura 53. Imágenes de los experimentos evaluados con la versión 1. En cada línea hay un experimento, (StillUZ, StillUZ2 y MoveUZ). Los cuadrados azul y rojo son los objetivos encontrados y el punto verde señala el objetivo elegido.

❖ Versión 2



Figura 54. Imágenes de los experimentos evaluados con la versión 2. En cada línea hay un experimento, (StillUZ, StillUZ2 y MoveUZ). Los cuadrados azul y rojo son los objetivos encontrados y el punto verde señala el objetivo elegido.

❖ Versión 3



Figura 55. Imágenes de los experimentos evaluados con la versión 3. En cada línea hay un experimento, (StillUZ, StillUZ2 y MoveUZ). Los cuadrados azul y rojo son los objetivos encontrados y el punto verde señala el objetivo elegido.

- Resultados del experimento medición del error de distancia cometido por el sistema.

❖ Prueba 2 del escenario 2:

MARCA	distanciaReal	distanciaExp	ERROR
1	3,3314	-	-
2	4,1362	3,7657	0,3705
3	3,6807	3,2813	0,3994
4	3,9696	3,4941	0,4754
5	3,7065	3,3451	0,3615
6	3,0203	2,6673	0,3530
7	2,3418	-	-
8	2,5300	2,3041	0,2259
9	2,0270	-	-
10	1,2928	1,1302	0,1626

Tabla 20. Resultados del error de distancia en cada punto de la segunda prueba en el escenario 2.

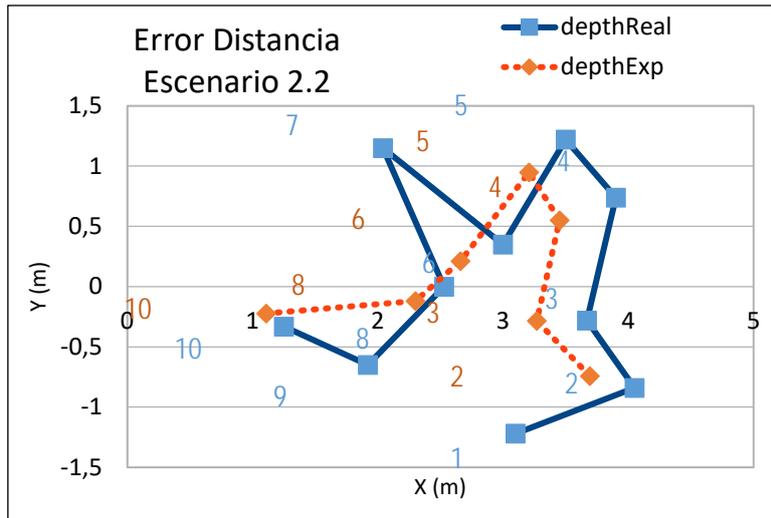


Figura 56. Representación gráfica de la prueba 2 en el escenario 2.

❖ Prueba 3 del escenario 2:

MARCA	distanciaReal	distanciaExp	ERROR
1	3,3314	-	-
2	4,1362	3,5553	0,5809
3	3,6807	3,2103	0,4704
4	3,9696	3,4892	0,4803
5	3,7065	3,1114	0,5952
6	3,0203	2,8196	0,2007
7	2,3418	2,1492	0,1926
8	2,5300	-	-
9	2,0270	1,9171	0,1100
10	1,2928	1,0444	0,2485

Tabla 21. Resultados del error de distancia en cada punto de la tercera prueba en el escenario 2.

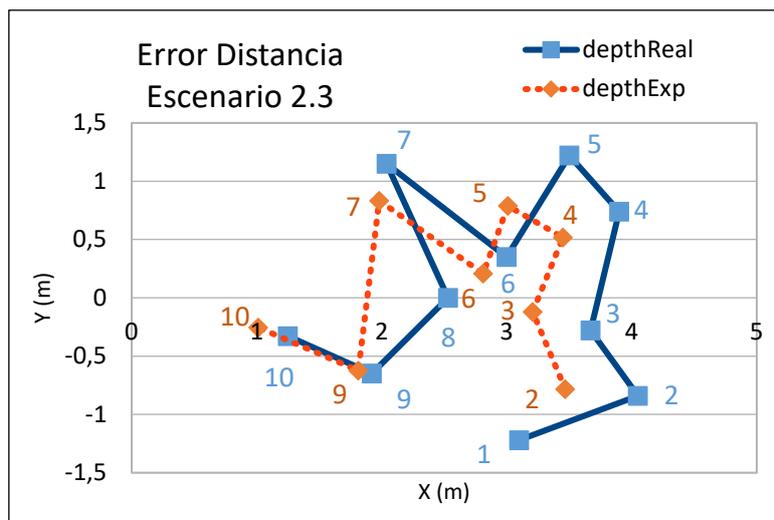


Figura 57. Representación gráfica de la prueba 3 en el escenario 2.

○ Imágenes de los experimentos de medición del error de distancia cometido por el sistema.

❖ Prueba1 en el escenario 1



Figura 58. Algunas fotos del experimento 1 en el escenario 1. La primera imagen se corresponde al primer punto del experimento y la última al último punto del experimento, el resto de imágenes son algunos puntos intermedios. Los cuadrados rojos y azules corresponden a los objetivos encontrados y el punto verde señala el objetivo elegido.

❖ Prueba 1 en el escenario 2

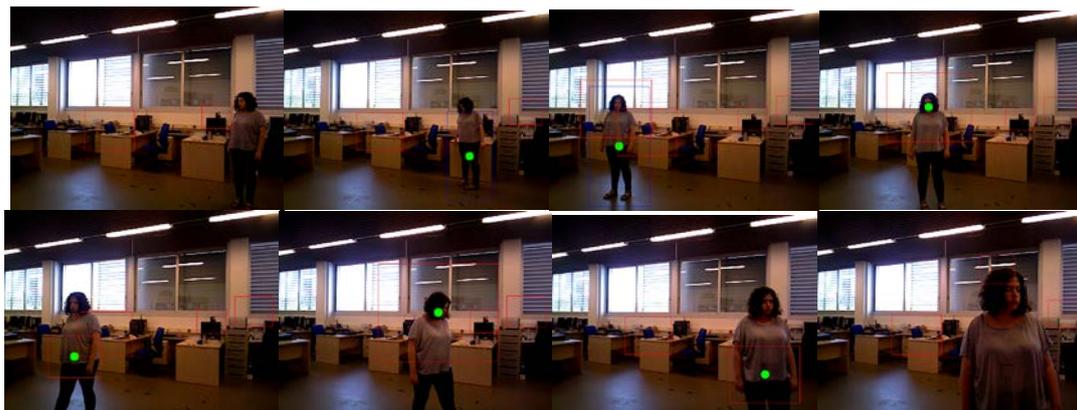


Figura 59. Algunas fotos del experimento 1 en el escenario 2. La primera imagen se corresponde al primer punto del experimento y la última al último punto del experimento, el resto de imágenes son algunos puntos intermedios. Los cuadrados rojos y azules corresponden a los objetivos encontrados y el punto verde señala el objetivo elegido.

❖ Prueba 2 en el escenario 2



Figura 60. Algunas fotos del experimento 2 en el escenario 2. La primera imagen se corresponde al primer punto del experimento y la última al último punto del experimento, el resto de imágenes son algunos puntos intermedios. Los cuadrados rojos y azules corresponden a los objetivos encontrados y el punto verde señala el objetivo elegido.

❖ Prueba 3 en el escenario 2



Figura 61. Algunas fotos del experimento 3 en el escenario 2. La primera imagen se corresponde al primer punto del experimento y la última al último punto del experimento, el resto de imágenes son algunos puntos intermedios. Los cuadrados rojos y azules corresponden a los objetivos encontrados y el punto verde señala el objetivo elegido.

### 3. Evaluación Del Sistema Completo

- Fotos pertenecientes a los vídeos grabados en los experimentos del sistema completo.
  - ❖ Experimentos de integración 1. Pruebas

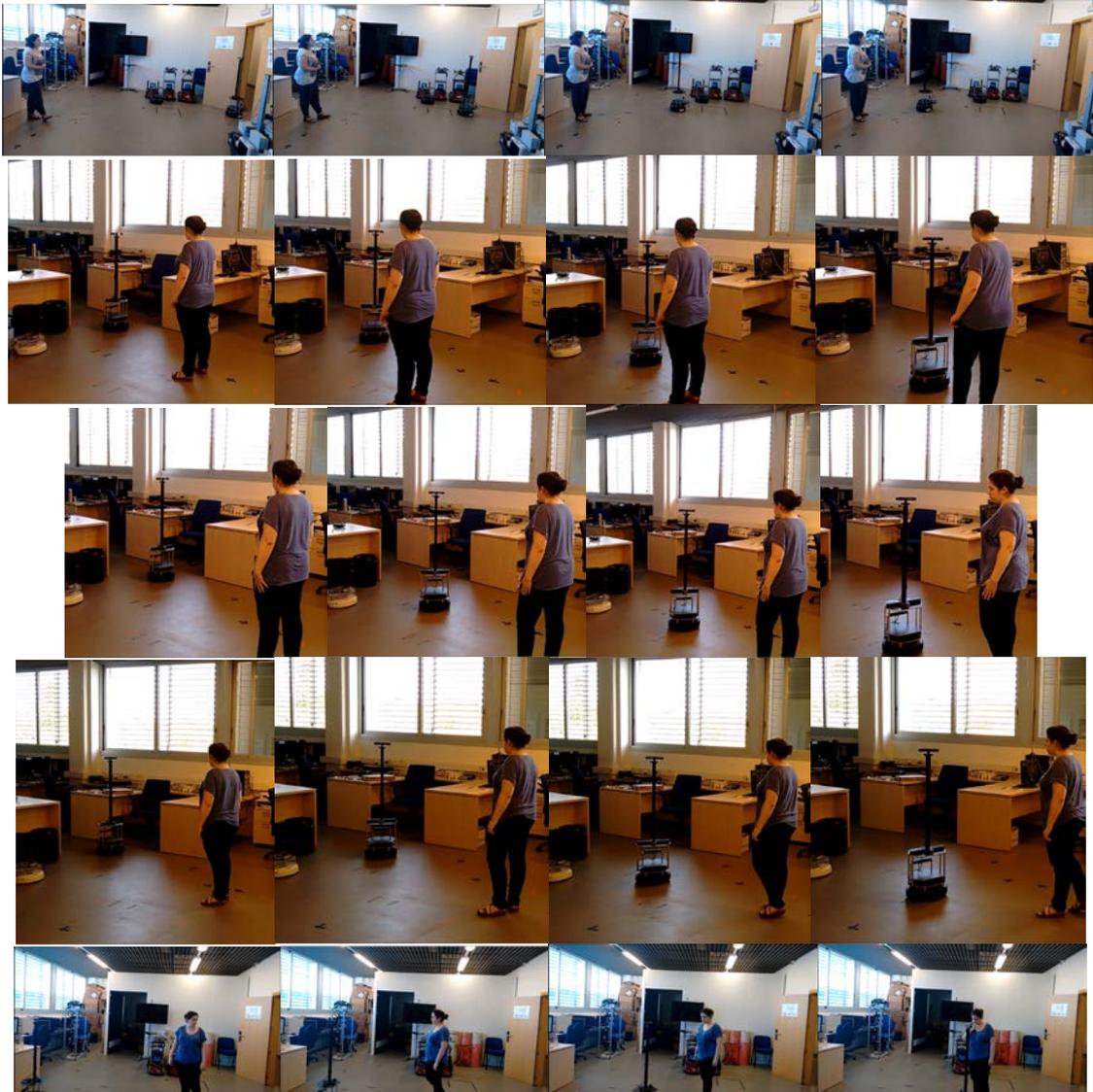


Figura 62. Imágenes de los vídeos grabados en los primeros experimentos con el sistema completo. Cada línea es un experimento, dónde se muestra la posición inicial del robot y la persona, luego dos posiciones intermedias y la posición final del robot y la persona.

❖ Experimentos detección de objetivo y acercamiento iterativo.



Figura 63. Cada línea corresponde con un experimento en el que se ha realizado un reconocimiento y acercamiento iterativo. En la primera imagen se muestra la posición inicial de la persona y el robot en el experimento, a continuación, dos posiciones intermedias en las que el robot se ha acercado poco a poco a la persona y por último las posiciones finales en el experimento.

❖ Experimentos finales.

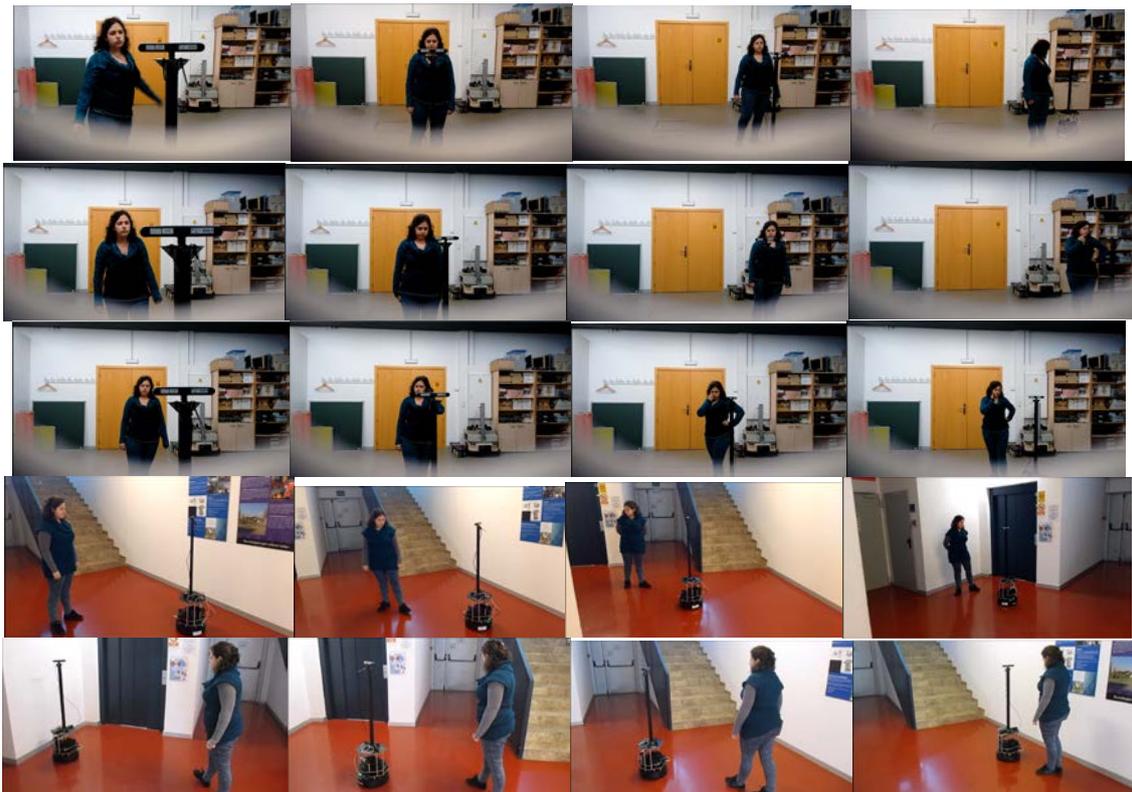


Figura 64. Imágenes de los vídeos grabados en los últimos experimentos con la plataforma real. Cada línea corresponde con un experimento. Las imágenes están ordenadas de la siguiente forma, posición inicial del robot y la persona, posiciones intermedias en las que se han realizado reconocimiento y seguimiento y, por último, posición final del robot y la persona.

# Bibliografía

- [1] Opencv.org. (2016). *OpenCV | OpenCV*. [online] Available at: <http://opencv.org/>. (2016).
- [2] Munaro, M. y Menegatti, E. Fast RGB-D people tracking for service robots. *Autonomous Robots*, 37(3), pp.227-242. (2014).
- [3] Omid Hosseini Jafari, Dennis Mitzel y Bastian Leibe. Real-Time RGB-D based People Detection and Tracking for Mobile Robots and Head-Worn Cameras. *Computer Vision Group, RWTH Aachen University, Germany*. (2014).
- [4] Timur Bagautdinov, François Fleuret y Pascal Fua. Probability Occupancy Maps for Occluded Depth Images. *École Polytechnique Fédérale de Lausanne (EPFL), Switzerland y IDIAP Research Institute, Switzerland*. (2015).
- [5] L. Spinello and K. O. Arras, "People detection in rgb-d data," en *iros 2011*, pp. 3838-3843. (2011)
- [6] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *CVPR*, (2005).
- [7] P. Felzenszwalb, B. Girshick, D. McAllester, and D. Ramanan, "Object Detection with Discriminatively Trained Part-Based Models," *PAMI*, vol. 32, no. 9, (2010).
- [8] M. Enzweiler, A. Eigenstetter, B. Schiele, and D. M. Gavrila, "Multi-Cue Pedestrian Classification with Partial Occlusion Handling," en *CVPR*, (2010).
- [9] P. Dollar, Z. Tu, P. Perona, and S. Belongie. Integral Channel Features. En *British Machine Vision Conference*, (2009).
- [10] Paul Viola y Michael Jones. Rapid Object Detection using a Boosted Cascade of Simple Features. *Accepted conference on computer vision and pattern recognition*. (2001)
- [11] Claudio Caraffi, Tomas Vojir, Jiri Trefny, Jan Sochman y Jiri Matas. A System for Real-time Detection and Tracking of Vehicles from a Single Car-mounted Camera. *Advanced Technology division, Toyota Motor Europe y The Center for Machine Perception, Department of Cybernetics, Faculty of Electrical Engineering, Czech Technical University in Prague, Czech Republic*. (2012)
- [12] M. S. Islam & M. A. Rahman. Design and Fabrication of Line Follower Robot. *Asian Journal of Applied Science and Engineering, Volume 2, No 2* (2013)
- [13] Wiki.ros.org. (2016). *Documentation - ROS Wiki*. [online] Accesible en: <http://wiki.ros.org/> (2016).
- [14] Turtlebot.com. (2016). *TurtleBot*. [online] Accesiblenen: <http://www.turtlebot.com/> (2016).
- [15] C. Papageorgiou, M. Oren, y T. Poggio. A general framework for object detection. In *International Conference on Computer Vision*, (1998).
- [16] Yoav Freund y Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. In *Computational Learning Theory: Eurocolt '95*, pages 23–37. Springer-Verlag, (1995).
- [17] Lienhart, R. and Maydt, J., "An extended set of Haar-like features for rapid object detection", *ICIP02*, pp. I: 900–903, (2002)

